

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Uma Pesquisa Qualitativa das Práticas de Testes de Software no Contexto das Aplicações Móveis

Achiles Pedro da Cunha Luciano

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande -
Campus I como parte dos requisitos necessários para obtenção do grau
de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Engenharia de Software

Wilkerson de Lucena Andrade e Dalton Dario Serey Guerrero
(Orientadores)

Campina Grande, Paraíba, Brasil

©Achiles Pedro da Cunha Luciano, 01/06/2017

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

S483p Luciano, Achiles Pedro da Cunha.
Uma pesquisa qualitativa das práticas de testes de software no contexto das aplicações móveis / Achiles Pedro da Cunha Luciano. – Campina Grande, 2017.
87 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2017.
"Orientação: Prof. Dr. Wilkerson de Lucena Andrade, Prof. Dr. Dalton Dario Serey Guerrero".
Referências.

1. Ciência da Computação. 2. Engenharia de Software. I. Andrade, Wilkerson de Lucena. II. Guerrero, Dalton Dario Serey. III. Título.

CDU 004.41(043)

Resumo

Contexto: O desenvolvimento de aplicações móveis cresce, acompanhando as constantes evoluções do hardware dos dispositivos. Com esta evolução, mais e mais recursos são adicionados e integrados à vida diária dos usuários. Contudo, a prática de testes de *software* utilizada pela comunidade para garantir o correto funcionamento das aplicações, incluindo a verificação de aspectos de segurança, portabilidade e compatibilidade, continua sendo um assunto pouco difundido. **Objetivo:** O objetivo deste estudo é compreender como as equipes lidam com a prática de testes de *software* aplicada ao contexto móvel, a relação cotidiana dos desenvolvedores com a literatura científica, a automação de testes e os desafios encontrados neste contexto. **Método:** Foi elaborado um estudo qualitativo, baseado na metodologia de Análise Temática, em três etapas: um *survey*, de caráter exploratório, que obteve 44 respostas de profissionais da área, seguido de uma seção de entrevistas com 12 profissionais para detalhar as questões levantadas durante o *survey* e um novo *survey*, de caráter confirmatório, que obteve 54 respostas e permitiu aos respondentes avaliar as principais conclusões das etapas anteriores. **Resultados:** Nossos resultados indicam a preferência dos desenvolvedores por testes manuais em detrimento da automação e o uso de testes funcionais automáticos de performance em alternativa, a escassez de referências para criação dos cenários de teste e os meios utilizados pelos profissionais para lidar com desafios como a fragmentação.

Abstract

Context: Mobile Web Application development increase, followed by often hardware evolutions from devices. With this development, more and more features has been added in a daily life of users. However, the practices of software testing used by developers to ensure the correct execution flow of the application, including safety aspects, portability and compatibility, remains a widespread issue. **Objective:** This study addresses how to understand teams and how they apply the tests in the development phase. **Method:** There was elaborated a qualitative study, based on Tematic Analisis methodology, in three stages: a exploratory survey, that collected 44 answers from professionals, followed by an interview with 12 professionals to detail the answers collected during the exploratory survey and a new confirmatory survey, that collected 54 answers and allowed the professionals to evaluate the main conclusions from previous stages. **Results:** Our results shown a preference from developers for manual testing over automation and use of automatic functional tests instead, the lack of references to create new test scenarios and the method used by professionals to deal with challenges such as fragmentation.

Agradecimentos

À minha mãe, **Jaceli**, pela dedicação, amor e todo suporte que me ofereceu. Por cultivar em mim o gosto pela leitura e conhecimento. Pelo apoio incondicional nos momentos mais difíceis. Sem este apoio seria impossível continuar.

Ao meu pai, **Aderaldo**, por seu exemplo, conselhos e inspiração necessária para desempenhar melhor o papel de pesquisador, profissional e ser humano.

À minha esposa, **Ana Renalle**, que acredita em mim antes que eu mesmo acredite. Pelo companheirismo, amor e parceria que compartilha comigo todos os dias.

À minha família e amigos, por serem um alicerce e fonte de inspiração nos estudos e na vida.

Ao amigo e irmão, **Dalton Jorge**, pelos conselhos, ensinamentos e exemplo de perseverança e responsabilidade.

Ao amigo **Alysson Milanez**, por ter compartilhado seu conhecimento e experiência por tantas vezes.

Aos meus orientadores, **Wilkerson Andrade** e **Dalton Serey**, por toda ajuda, inspiração, orientação, paciência e estímulo antes e durante o mestrado.

Aos colegas de trabalho, do projeto **Ingenico FT**, pelo imenso crescimento profissional e acadêmico que compartilham no dia a dia no laboratório.

Aos professores da banca examinadora, **Roberta Coelho** e **Adalberto Cajueiro**, pela disposição e contribuições ao avaliar este trabalho.

Aos professores, estudantes e colegas que fazem parte do SPLAB, e que, de alguma forma ajudaram na construção deste trabalho e de tantos outros que nascem a partir deste laboratório.

Conteúdo

1	Introdução	1
1.1	Objetivos	2
1.2	Metodologia	3
1.3	Relevância	4
1.4	Estrutura da Dissertação	4
2	Fundamentação Teórica	6
2.1	<i>Mobile Apps</i>	6
2.2	Teste de <i>Software</i>	7
2.2.1	Tipos e Níveis de Teste de <i>Software</i>	8
2.2.2	Testes em Aplicativos Móveis	9
2.3	Pesquisa Qualitativa e Análise Temática	10
2.4	Considerações Finais	11
3	Trabalhos Relacionados	12
4	Descrição do Método	18
4.1	Planejamento da Pesquisa	18
4.1.1	Estudo Piloto	20
4.1.2	<i>Snowball Sampling</i>	21
4.2	Execução do Estudo	23
4.3	Considerações Finais	24
5	Resultados	25
5.1	Survey Exploratório	25

5.1.1	Dados profissionais dos participantes	25
5.1.2	Papéis desempenhados pelos participantes	27
5.1.3	Automação de Testes	29
5.1.4	Ferramentas de Testes	30
5.1.5	Literatura	30
5.2	Entrevistas	32
5.2.1	Codificação	33
5.2.2	Experiência dos participantes	35
5.2.3	Testes em <i>Apps</i>	35
5.2.4	Relações com a literatura	37
5.2.5	Documentação	38
5.2.6	Desafios Encontrados	38
5.2.7	Considerações Finais	41
5.3	<i>Survey</i> Confirmatório	42
5.3.1	Perfil dos Participantes	42
5.3.2	Frases e Avaliações	44
5.4	Ameaças à validade	56
5.5	Considerações Finais	57
6	Discussão	58
6.1	<i>RQ1: Como os desenvolvedores lidam com a automação dos testes em aplicações móveis?</i>	58
6.2	<i>RQ2: Quais os tipos de testes automáticos mais utilizados pela comunidade?</i>	60
6.3	<i>RQ3: Qual o grau de satisfação dos participantes com as atuais práticas de teste?</i>	61
6.4	<i>RQ4: Como é a relação entre a literatura e a prática de testes?</i>	62
6.5	<i>RQ5: Como os desenvolvedores lidam com os desafios encontrados?</i>	63
7	Conclusões	65
7.1	Contribuições	66
7.2	Trabalhos Futuros	66

A Roteiro do <i>Survey</i> Exploratório	73
B Roteiro das Entrevistas	77
C Termo de Consentimento da Entrevista	81
D Roteiro do <i>Survey</i> Confirmatório	83

Lista de Símbolos

URL - *Uniform Resource Locator*

HTML5 - *Hypertext Markup Language Versão 5*

SLOC - *Source Lines of Code*

Lista de Figuras

4.1	Snowball Sampling Linear	22
4.2	Snowball Sampling Exponencial Não-Discriminante	22
4.3	Snowball Sampling Exponencial Discriminante	23
5.1	Cargos desempenhados pelos participantes	26
5.2	Experiência dos participantes em diferentes plataformas móveis	27
5.3	Equipes Separadas para testes	28
5.5	Ferramentas utilizadas pelos respondentes	31
5.6	Tipo de literatura pesquisada pelos participantes	31
5.7	Leitura dos textos e início da codificação	34
5.8	Exemplo da organização de um tema	34
5.9	Formação dos participantes - <i>Survey Confirmatório</i>	42
5.10	Vinculação profissional dos participantes - <i>Survey Confirmatório</i>	43
5.11	Tempo de Experiência dos Participantes - <i>Survey Confirmatório</i>	44
5.12	Respostas dos Participantes à Pesquisa (Likert)	45
5.13	Respostas dos Participantes sobre a satisfação com a prática de testes	46
5.14	Respostas dos Participantes sobre automação dos testes	47
5.15	Respostas dos Participantes sobre os tipos de testes utilizados no contexto móvel	49
5.16	Respostas dos Participantes sobre a literatura consultada	50
5.17	Respostas dos Participantes sobre a documentação e cenários de teste	53
5.18	Respostas dos Participantes sobre a fragmentação	54

Lista de Tabelas

2.1	Tabela de Características dos <i>apps</i> e impacto nos testes	9
3.1	Número de <i>papers</i> cujo título continha a <i>string</i> "mobil"	13
3.2	Distribuição 2010-2012	13
4.1	<i>Apps Open-source</i> Visitados	19
5.1	Seções das Entrevistas e Exemplos de Perguntas Presentes no Roteiro	33
5.2	Experiência dos Participantes	36

Lista de Códigos Fonte

5.1	testes em R	46
5.2	Teste de Wilcoxon Aplicado à afirmação AF3	48

Capítulo 1

Introdução

Dados da União Internacional de Telecomunicações apontam que a quantidade de dispositivos de telefonia móvel no mundo já está próxima da quantidade de seres humanos que habitam o planeta [1]. Além da imensa quantidade, os aparelhos atuais integram recursos como câmera, *bluetooth*, acelerômetro, acesso à Internet, *gps*, etc. A quantidade de aplicações desenvolvidas para estes sistemas também cresceu e se desenvolveu de acordo com as soluções disponíveis [17].

De acordo com o *Stack OverFlow Developer Survey 2017* [2], a plataforma *Android* é a terceira mais utilizada pelos desenvolvedores participantes, na lista estavam inclusas também plataformas *desktop*. Ainda de acordo com a mesma pesquisa, entre os sistemas operacionais utilizados no contexto móvel, o *Android* contava com cerca de 65% do *market share*.

Os aplicativos móveis, ou simplesmente *apps*, possuem características particulares e diferenciadas das aplicações *web* e *desktop* [9]. O ambiente *mobile* lida com suas próprias limitações como recursos de *hardware* limitados e fonte de energia finita [39], ao mesmo tempo em que possui uma grande variedade de conexões com a rede (3g, 4g, *Wi-Fi*, *WiMAX*) e sensibilidade ao contexto [32].

Autores como Gao *et al.* [17], Masi *et al.* [30] e Haller [21] reconhecem tipos de aplicativos que se distinguem pelas tecnologias empregadas em seu desenvolvimento: aplicações Nativas, *Web Apps* ou *Launcher Apps* e Híbridas. Os aplicativos nativos são desenvolvidos direcionados para uma plataforma específica, *e. g.* *Android* ou *IOS* [8], enquanto que os aplicativos *web* e híbridos são desenvolvidos em *HTML5* e podem não ser capazes de acessar todos os recursos do dispositivo [20].

Outra característica apontada pelos autores é a *fragmentação* [44], que representa a diversidade de dispositivos e sistemas operacionais existentes e parece ser um desafio para os desenvolvedores. A fragmentação externa se refere às diferenças entre sistemas operacionais, enquanto que a fragmentação interna remete às constantes modificações nas versões de um mesmo sistema operacional que, por vezes, altera a forma como as funcionalidades devem ser acessadas. [25]

Os atributos diferenciados dos aplicativos móveis afetam diretamente as atividades de qualidade de software e necessitam de abordagens específicas para identificar *bugs* nos *apps* [17]. Apesar dos estudos que relacionam os tipos de testes a serem utilizados em aplicativos móveis e as ferramentas disponíveis para automação dos testes, como o *AppDoctor* [23], o *MonkeyRunner* e o *MobiGuitar* [4], ainda não há um consenso sobre qual tipo de teste ou conjunto de testes suportados pelos *frameworks* disponíveis apresenta melhor eficácia na detecção de falhas nos *apps*, restando assim, desafios a serem enfrentados [32]. Além disso, há poucos estudos qualitativos na área e que focam no estudo da prática real dos testes pelos desenvolvedores [25].

Neste Contexto, esta pesquisa pretende, por meio de um estudo qualitativo, investigar a prática dos desenvolvedores de aplicações móveis e apresentar como estão sendo planejadas e executadas as atividades de teste, quais as ferramentas utilizadas e as técnicas comuns de validação e verificação do software produzido.

1.1 Objetivos

O objetivo geral deste trabalho é realizar uma pesquisa de caráter exploratório a fim de compreender as práticas de teste de software utilizadas no contexto de aplicações para dispositivos móveis (*apps*) no Brasil.

Sendo assim, ao invés de iniciar o estudo com hipóteses pré-determinadas, que seriam testadas e avaliadas por meio de experimentos e análise, a preocupação deste estudo está em investigar as atividades de teste, incluindo a sua automação, realizadas por equipes de desenvolvimento, buscando responder às seguintes perguntas de pesquisa:

RQ1: *Como os desenvolvedores lidam com a automação dos testes de aplicações móveis?*

RQ2: *Quais os tipos de testes automáticos mais utilizados pela comunidade?*

RQ3: *Qual o grau de satisfação dos participantes, em geral, com as atuais práticas de teste?*

RQ4: *Como é a relação entre a literatura e a prática de testes?*

RQ5: *Como os desenvolvedores lidam com os desafios encontrados?*

1.2 Metodologia

A metodologia utilizada na pesquisa contou com as seguintes etapas:

1. **Survey Exploratório:** Esta etapa consistiu na elaboração de um questionário, disponibilizado *online*, para profissionais e pesquisadores da área de desenvolvimento *mobile*. O planejamento desta etapa foi realizado através de um roteiro, construído a partir de um estudo prévio da literatura na área e guiado pelas perguntas de pesquisa. O método de *Snowball Sampling* [19] foi utilizado para auxiliar no recrutamento de novos participantes e, ao todo, 44 respostas foram coletadas.
2. **Entrevistas:** Após a obtenção das respostas na etapa anterior, foram elaboradas entrevistas com participantes voluntários. Como a quantidade de respostas foi pequena, não foi necessário elaborar critérios de seleção. Ao todo, 12 entrevistas foram realizadas.
3. **Análise Temática:** Durante esta etapa, as entrevistas foram transcritas e analisadas para identificar afirmações plausíveis sobre o contexto de práticas de teste de software aplicadas aos *apps*. Estas afirmações foram essenciais para a elaboração da próxima etapa do estudo.
4. **Survey Confirmatório:** A fim de buscar a avaliação das afirmações levantadas na etapa de entrevistas, foi elaborado um novo *survey*, de caráter confirmatório. O formato das perguntas foi construído de forma que as afirmações foram expostas ao participante, que, por sua vez, respondia o grau de concordância. O questionário foi enviado para profissionais da área e coletou mais 53 respostas.

1.3 Relevância

A principal contribuição deste trabalho é a descrição de um cenário recente, em crescente interesse pela indústria e pela academia [9] e cuja atividade de testes ainda parece utilizar métodos herdados das aplicações *web* e *desktop*.

A identificação das práticas de testes de aplicações móveis em diferentes projetos de software pode ajudar a identificar oportunidades de pesquisa e desenvolvimento bem como auxiliar os desenvolvedores no oferecimento de tecnologias que possam atendê-los.

Os resultados obtidos com este trabalho também incorporam elementos de análise qualitativa dos dados, somando forças ao interesse de uso desta metodologia em diferentes estudos e pesquisas.

1.4 Estrutura da Dissertação

Este trabalho tem seus capítulos estruturados conforme descrito abaixo:

Capítulo 2 - Fundamentação Teórica: Este capítulo apresenta os principais conceitos que fundamentam este trabalho. São contextualizados o conceito de *mobile apps* e contextualizada a prática de teste de software aplicada às aplicações móveis. Também é explicada a utilização de pesquisas qualitativas em Ciência da Computação e Análise Temática.

Capítulo 3 - Trabalhos Relacionados: Este capítulo apresenta trabalhos que relacionam o estudo e descrição de ferramentas para testes de software no contexto móvel, assim como trabalhos que utilizaram o método qualitativo de análise de dados em Ciência da Computação. Além disso, são retratadas as semelhanças e diferenças dos trabalhos relacionados com o presente estudo.

Capítulo 4 - Descrição do Método: O capítulo descreve as questões de pesquisa utilizadas para nortear os objetivos da pesquisa e a descrição dos passos metodológicos aplicados para conduzir o estudo.

Capítulo 5 - Resultados: Neste capítulo são apresentados os resultados obtidos por meio dos *surveys* e entrevistas realizados durante as etapas da metodologia. Número de

respondentes, quantidade de respostas, a caracterização do perfil dos participantes e as afirmações levantadas. Também são explicitadas as limitações do estudo e as ameaças à validade.

Capítulo 6 - Discussão: Neste capítulo são discutidos os resultados obtidos, as principais afirmações do trabalho e a interpretação dos dados de acordo com as perguntas de pesquisa.

Capítulo 7 - Conclusões: Traz à tona uma breve discussão sobre os resultados obtidos e as perspectivas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo provê o alicerce teórico necessário à compreensão dos temas abordados por este trabalho. São apresentados os principais conceitos relacionados ao Teste de Software, Aplicações Móveis (*Apps*), Pesquisa Qualitativa e Análise Temática.

2.1 *Mobile Apps*

Uma aplicação móvel ou simplesmente *app* [9] é um software desenvolvido para funcionar em dispositivos móveis tais como *Smartphones* e *Tablets* [27]. Este ambiente possui restrições comuns ao contexto de computação móvel, como recursos computacionais limitados, vulnerabilidades em segurança, performance, variabilidade de confiabilidade, e fonte de energia finita [39] e pode ser equipado com sensores que possibilitam às aplicações a característica de sensibilidade ao contexto [32].

As aplicações construídas podem ser divididas em três tipos: Nativas, Web Apps e Híbridas [8]. Os aplicativos **nativos** são desenvolvidos direcionados para uma plataforma específica (ex: Android ou IOS) e podem acessar diretamente os recursos fornecidos pelo aparelho, fornecendo confiabilidade e bom desempenho [25]. Eles também requerem um grau maior de experiência do desenvolvedor do que os outros tipos de aplicações [44]. Os **Web Apps** são, na realidade, sites que se adequam à usabilidade diferenciada de um dispositivo móvel, são desenvolvidos em HTML 5 e direcionam o usuário para uma determinada URL, e por isso, requerem que um *browser* esteja instalado [25]. Estes tipos de *apps* não são capazes de acessar todos os recursos do aparelho, ocasionando limitações aos desenvolvedores

[30]. Aplicativos **híbridos** se comportam como um aplicativo nativo, fornecem acesso aos recursos do aparelho e podem ser disponibilizados no repositório de aplicativos fornecidos pela plataforma. A construção destes sistemas é baseada em HTML 5, como uma aplicação web comum, e um *framework* é responsável por criar uma interface de comunicação com os recursos do aparelho [20].

Devido ao suporte multiplataforma, à facilidade e à rapidez de construção e à tentativa de diminuir os efeitos da fragmentação [44], o desenvolvimento de aplicações híbridas vem crescendo e diversos frameworks surgiram para auxiliar os desenvolvedores. Não apenas frameworks para facilitar a comunicação entre a aplicação e o sistema operacional, mas também frameworks para construção de páginas responsivas, que se adequam à variedade de telas existentes, com componentes que se assemelham aos de uma aplicação nativa, como botões, listas, split-views, grids e menus [6].

Em menor número, surgiram também frameworks específicos para testar as aplicações. Com a aplicação funcionando no próprio aparelho, pode-se realizar testes direcionados para uma plataforma específica, verificando a comunicação com os recursos nativos assim como também em um navegador ou emulador para validar a construção das páginas HTML e a interatividade fornecida pelo *Javascript*. Também são realizados testes exploratórios nos quais a aplicação é “bombardeada” por um grande número de requisições e estima-se um limite desejado para que a aplicação continue funcionando sem problemas. Exemplos deste tipo de framework são o *AppDoctor* [23], o *MonkeyRunner* [17] e o *MobiGuitar*[4].

2.2 Teste de Software

A prática de teste de software faz parte de um processo maior, de validação e verificação (V&V), na qual é avaliado se o produto de software construído está de acordo com os seus requisitos funcionais e não-funcionais (**verificação**) e se entrega as funcionalidades esperadas pelos clientes (**validação**) [42].

A fim de verificar e validar um *software*, são necessárias a identificação de erros, defeitos e falhas nos projetos. Koscianski e Soares [28] definem **defeito** como sendo parte do produto de software e refere-se a ele como sendo algo que está implementado no código de maneira incorreta. A **falha** seria o resultado errado, provocado por um defeito e o **erro**, por sua vez

é também reconhecido por Jorgensen [26] como um *engano (mistake)*, uma ação humana de inserir um trecho defeituoso no código-fonte do *software*.

Sendo assim, o teste de *software* consiste de uma análise dinâmica do produto e é uma atividade relevante para a identificação e eliminação de erros que persistem [29]. Fundamentalmente, a atividade de teste é o processo de executar um programa com o objetivo de encontrar defeitos [33].

2.2.1 Tipos e Níveis de Teste de Software

Para realizar as atividades de V&V são utilizados diferentes testes que permeiam todas as etapas de desenvolvimento do produto de software e podem ser divididos de acordo com a sua natureza ou nível de arquitetura do *software*.

Os tipos de testes existentes podem ser divididos em testes baseados em especificação ou implementação do *software* [29]:

- **Testes baseados em especificação** são testes conhecidos como *caixa-preta* ou funcionais, nos quais o código-fonte do *software* não é diretamente analisado. O *software* é executado e seu comportamento é examinado sem a necessidade de inspeções no código-fonte.
- **Testes baseados em implementação**, também conhecidos como testes caixa-branca ou estruturais, se baseiam no conhecimento da estrutura interna do programa, e são realizadas inspeções no código-fonte do *software* a fim de exercitar partes do código que possam manifestar defeitos.

Com relação aos níveis de teste, a literatura acadêmica define diversas categorias [29; 33; 42]. Entre elas, as mais comumente utilizadas são:

1. **Testes de Unidade:** Compõem testes que avaliam pequenas unidades do sistema, como retornos de funções, valores de variáveis e interações entre as classes do sistema;
2. **Testes de Integração:** Avalia a interação entre os subsistemas do *software*;
3. **Testes de Sistema:** São testes que avaliam o software a partir de sua arquitetura. Uma versão completa do sistema é criada e avaliada de acordo com sua especificação;

4. **Testes de Aceitação:** É a última fase de testes antes da implantação do sistema. Composta de testes realizados para garantir que o *software* produzido está de acordo com os requisitos especificados e as expectativas do cliente.

Os conceitos de tipos e níveis de testes de software serão utilizados no decorrer desta dissertação para compreender os questionamentos realizados aos participantes durante os *surveys* exploratório e confirmatório e entrevistas.

2.2.2 Testes em Aplicativos Móveis

Gao *et al.* [16] definem *mobile app testing* como sendo o conjunto de atividades de teste para aplicações web e nativas em dispositivos móveis utilizando métodos bem definidos e ferramentas para garantir a qualidade em funcionalidades, comportamentos, performance e qualidade do serviços, assim como mobilidade, usabilidade, interoperacionalidade, conectividade, segurança e privacidade.

Mas as aplicações móveis são tão diferentes para que se pense em estratégias específicas de teste? Kirubakaran e Kharthikeyani [27] e Muccini *et al.* [32], realizam esta discussão em seus trabalhos, afirmando características próprias dos *apps* e como estas características impactam a atividade de testes. A Tabela 2.1 resume os principais conceitos apresentados pelos autores.

Características distintivas dos apps	Implicações nos testes
Conectividade	Testes Funcionais, Performance, Segurança, Teste de Confiabilidade em diferentes redes
Conveniência (Qualidade de Design)	Testes de Interface (GUI)
Diversidade de Dispositivos e SO's	Matriz de Testes baseada na Diversidade de Cobertura
Touch Screens	Testes de Usabilidade e Performance
Novas Linguagens de Programação	Testes caixa preta e caixa branca, análise de Bytecode
Limitações de Recursos	Testes Funcionais e Monitoramento de Performance
Sensibilidade ao Contexto	Testes Funcionais Dependentes do Contexto

Tabela 2.1: Tabela de Características dos *apps* e impacto nos testes

As características apresentadas na tabela possuem aspectos próprios dos *apps* comparados às aplicações web ou desktop. Por exemplo, a conectividade dos dispositivos acontece paralelamente em diferentes tecnologias como *Wifi*, *Bluetooth*, *3G* ou *4G*, Infravermelho, etc. E, por isso, demandam testes que aumentem a segurança, a disponibilidade e a confi-

abilidade da aplicação em mais numerosos cenários. Um outro exemplo é a diversidade de dispositivos, Sistemas Operacionais e versões das bibliotecas de desenvolvimento dos aplicativos, é uma característica difundida em estudos da área e conhecida como *fragmentação interna* ou *externa* [44; 37];

2.3 Pesquisa Qualitativa e Análise Temática

Estudos qualitativos são recentes na história da computação quando comparados aos métodos existentes nas ciências humanas [40] e seus dados, em forma de palavras ou imagens, carregam por definição, conforme afirma Parra *et al.* [35], mais informações do que os dados quantitativos.

Estudos utilizando técnicas de análise qualitativa como a Teoria Fundamentada [18] e Análise Temática [7] já são utilizadas pela comunidade científica em Ciência da Computação [25; 31; 41; 3; 10].

Não há consenso entre os autores sobre um único modo estruturado de conduzir a análise. De acordo com Braun e Clarke [7], a Análise Temática é um método de análise qualitativa para identificar, analisar e descrever padrões existentes nos dados no qual os seguintes passos estão presentes:

Coleta dos Dados, na qual são utilizados métodos qualitativos como entrevistas, questionários, *brainstorming*, sessões de *think-aloud* [40], ou repositórios de dados públicos ou privados;

Codificação, que consiste na criação de códigos que identifiquem trechos menores dos dados;

Criação de Temas que agrupam códigos cujos significados sejam aproximados. Esta etapa pode se repetir para agrupar os próprios temas em temas maiores, caso a similaridade entre eles seja evidenciada;

Elaboração de um relatório contendo a informação identificada.

No caso de estudos em que a coleta dos dados é realizada por meio de gravações de áudio ou vídeo, é necessária a etapa de transcrição dos dados coletados para a criação dos

códigos. E, durante a etapa de análise, podem ser utilizados gráficos ou mapas para facilitar a visualização dos temas.

2.4 Considerações Finais

Os aplicativos móveis são softwares desenvolvidos para atuar em ambientes que possuem limitações de recursos e uma infinidade de cenários diretamente proporcionais à sensibilidade ao contexto que lhes é permitido o acesso. Sendo assim, estas aplicações reúnem características próprias que afetam diretamente a atividade de testes e a garantia da qualidade do software produzido. Como a atividade de testes é planejada e executada por pessoas, a análise unicamente quantitativa dos dados pode não ser capaz de retratar a prática dos desenvolvedores e testadores, e por isso, o uso de técnicas de análise qualitativas como a Análise Temática são adequadas e podem trazer importantes contribuições.

Capítulo 3

Trabalhos Relacionados

Apesar de ser um assunto relativamente recente, a literatura acadêmica já possui diversos estudos que abordam o desenvolvimento e testes em aplicações móveis [36; 32; 27]. O objetivo deste capítulo é apresentar alguns destes trabalhos, assim como descrever e comparar o reflexo das suas influências nesta pesquisa.

Picco *et al* [36] elaboraram uma revisão de um trabalho anterior [38], publicado no ano 2000, comparando as afirmações daquela época sobre computação móvel e os desafios da realidade atual. Nesta revisão, os autores constataram pouco interesse da comunidade acadêmica pelo tema. Em uma busca nos principais periódicos de engenharia de software (TSE, TOSEM, ICSE, ASE, ESEC/FSE), Picco *et al.* constataram que o prefixo "mobil" estava presente em apenas 5 títulos de trabalhos publicados entre os anos de 2010 e 2012. Ao realizar novas buscas, desta vez acrescentando os termos "*Android*" e "*phone*", apenas no ano de 2012 foram encontrados 9 trabalhos. As tabelas 3.1 e 3.2 ilustram os resultados das buscas.

Atualmente, realizando uma busca simples na biblioteca da IEEE¹, apenas o termo *Android* retorna mais de 5.500 resultados, e na biblioteca da ACM², mais de 2.400 trabalhos. Estas informações demonstram o considerável aumento do interesse da comunidade acadêmica pelo tema.

Ainda de acordo com Picco *et al*, as práticas de teste em aplicações móveis não pareciam ter mudado consideravelmente nos últimos dez anos desde a publicação de seu trabalho, e que técnicas capazes de verificar cenários que se adequem à variedade de situações em

¹Disponível em: <http://ieeexplore.ieee.org/Xplore>

²Disponível em: <http://dl.acm.org/>

Tabela 3.1: Número de *papers* cujo título continha a *string* "mobil"

	Todos	"mobil"	%
Antes de 1995	2664	1	0,04%
1995-1997	537	4	0,74%
1998-2000	766	19	2,48%
2001-2003	853	21	2,46%
2004-2006	984	9	0,91%
2007-2009	794	8	1,01%
2010-2012	1173	5	0,43%

Tabela 3.2: Distribuição 2010-2012

	2010	2011	2012
<i>mobil</i> *	1	0	4
<i>Android</i>	0	0	4
<i>phone</i>	0	0	1

que um sistema móvel pode se encontrar precisam ser aprimoradas. Apesar das afirmações importantes, não foi realizado um estudo empírico para verificar a eficácia de ferramentas ou técnicas de teste.

Gao *et al* [17] também apontam a necessidade de adaptação da atividade de testes ao contexto *mobile*. Segundo ele, uma das principais características que afetam a prática dos testes é que a aplicação móvel deve manter as características de funcionamento não apenas a qualquer tempo, mas também em todos os espaços físicos, levando em consideração que os sensores presentes no dispositivo são capazes de acessar informações sobre o ambiente, como localização e temperatura.

O trabalho de Gao *et al* se preocupa em detalhar as características únicas das aplicações móveis e realizar um tutorial do processo de testes a ser utilizado em diferentes tipos de *apps* e diferentes equipes de software. Ele divide as abordagens de testes da seguinte forma:

Testes baseados em emuladores: Que possuem baixo custo, mas têm limitações e podem apresentar dados não confiáveis.

Testes baseados em dispositivos: São testes que demandam gastos com a compra de dispositivos e, mesmo assim, não são capazes de alcançar todos os cenários possíveis.

Testes em nuvem (Cloud Testing): Utilizam serviços terceirizados para execução dos testes em diferentes dispositivos.

Testes baseados em multidão: Sugere contratar *freelancers* ou buscar comunidades de usuários finais para executar os testes.

As contribuições de Gao *et al.* também incluem uma discussão acerca dos desafios para os desenvolvedores e alerta as necessidades atuais que servem como oportunidades para possíveis pesquisas de desenvolvimento na área.

Neste mesmo sentido Henry Muccini *et al.* [32] realizaram um estudo do estado da arte acerca dos desafios existentes no teste de aplicações móveis. O trabalho responde as questões: (1) *As aplicações móveis são tão diferentes para que se pense em novas estratégias específicas de teste?*, (2) *Quais são os desafios para testes de aplicações móveis?* e (3) *Qual o papel que a automação toma nesse contexto?*. Suas conclusões apontam a necessidade de abordagens específicas e a necessidade de automação, principalmente devido à constante mudança de requisitos dos aplicativos móveis.

Os desafios citados no trabalho coincidem com um trabalho semelhante, de Kirubakaran e Kharthikeyani [27]. Contudo, estes últimos acrescentaram soluções aos desafios encontrados por meio da automação dos testes, que diminuem a necessidade de retrabalho do desenvolvedor a cada novo ciclo de desenvolvimento e impede o viés da identificação de falhas, uma vez que cada teste é escrito uma única vez e executado quantas vezes forem necessárias.

Apesar de trazer contribuições à discussão da prática de testes em *apps*, os estudos supracitados apenas fazem uma leitura do estado da arte e não realizam estudos empíricos para fundamentar suas descobertas, e, apesar de sugerir técnicas, processos e ferramentas, não demonstram preocupação com o que está sendo efetivamente utilizado pela comunidade de desenvolvedores, que podem, inclusive, não pesquisar a solução para os problemas seus cotidianos na literatura científica. Estes aspectos abrem oportunidades para estudos como o realizado pelo presente trabalho.

Em contrapartida, há estudos como o de Joorabchi *et al.* [25], que realizaram um trabalho cujo objetivo era compreender as principais práticas e desafios do desenvolvimento de software *mobile* nativo e entre várias plataformas. Os autores realizaram entrevistas com 12 desenvolvedores *seniors* de 9 companhias diferentes e implementaram um *survey* que alcançou 188 respondentes da comunidade de desenvolvedores. A análise dos dados coletados seguiu a metodologia qualitativa de *Grounded Theory* [18]. De acordo com os autores, esta metodologia é adequada quando o objetivo do pesquisador é aprender como pessoas

gerenciam situações problemáticas e como pessoas compreendem e lidam com as situações.

Entre seus principais resultados estão:

Avançando em direção à fragmentação e não à unificação: A fragmentação é um dos desafios encontrados por desenvolvedores e testadores *mobile*.

Suporte a testes : segundo o estudo, há muito pouco sendo feito pelos testes na prática. O suporte à automação dos testes ainda é muito limitado.

Testes manuais prevalecem: 64% dos respondentes afirmaram que testam seus aplicativos manualmente. Até mesmo empresas maiores ainda têm resistência à automação.

Desenvolvedores são testadores: 80% dos participantes informaram que os desenvolvedores testam as próprias aplicações e 28% deixam esta etapa também para os clientes (*beta testers*).

Outros resultados foram obtidos, como a preferência dos desenvolvedores na construção de aplicações móveis nativas em detrimento das híbridas, devido à sua usabilidade e performance e também foi detectada a dificuldade em desenvolver *apps* para dispositivos com resoluções e recursos de hardware distintos, mesmo que utilizando o mesmo sistema operacional.

Apesar do foco na prática real dos desenvolvedores de software *mobile*, o estudo não se preocupou especificamente nos testes e ferramentas utilizadas, preferindo uma abordagem mais geral do tema e não há detalhamento da utilização de Teoria Fundamentada (*Grounded Theory*).

A estruturação do presente trabalho contou ainda com a influência de outros diversos estudos em *mobile* e *mobile testing*. Estes estudos colaboraram com a compreensão de *Grounded Theory* [41] e Análise Temática [31; 7], e também a compreensão do estado da arte. Serão citados resumidamente a seguir:

J. Gao *et al.* [16] e Villanes *et al.* [43] apresentam o conceito de *Testing as a Service* ou TaaS. O uso de serviços na nuvem que realizem a verificação dos aplicativos de acordo com testes pré-definidos relacionados aos recursos do aparelho que a aplicação necessita para rodar. Os desenvolvedores, depois de inserir as características de seus aplicativos (como acesso à câmera, gps e outros recursos), podem enviá-los a um serviço de teste na nuvem

que roda os testes em diferentes dispositivos. Seus resultados e experiências apontam para o sucesso deste tipo de teste e trabalhos futuros para o desenvolvimento de frameworks baseados neste tipo de abordagem.

Amal *et al.* [4] apresentam o *framework* (MobiGUITAR) para testes baseados em modelos no contexto de aplicações móveis. Realiza uma comparação com outros dois frameworks (*Monkey* e *Dynodroid*) e aponta para o diferencial que é a capacidade de o usuário customizar as entradas para teste, enquanto que o *Monkey* e o *Dynodroid* geram entradas aleatórias.

Ridene *et al* [37] apontam problemas existentes no teste de aplicações móveis, incluindo a fragmentação. Sua contribuição é a criação de uma DSML (Domain-specific Model Language) para realizar os testes.

Angulo e Ferre [6] diferenciam tipos de abordagens multiplataforma em diferentes aplicativos (Facebook, Whatsapp e Twitter). Procura verificar se os aplicativos respeitam o guia de estilo das diferentes plataformas. O foco é UX (User Experience), o trabalho realiza avaliações por questionários para verificar qual das abordagens apresenta melhor "look and feel" e performance.

Xanthopoulos e Xinogalos [44] realizam uma comparação das abordagens multiplataforma e nativos, apontam o *Phonegap* e o *Titanium* como alternativas. Eles citam que a Apple pode não aceitar algumas aplicações híbridas, e cria uma tabela de comparação entre as diferentes abordagens. Além disso, define os termos UX, APP, FRAGMENTATION e PLATAFORM. Para validação, cria uma mesma aplicação usando as diferentes abordagens e conclui: (1) *"Look and Feel" das aplicações nativas é superior*, (2) *Relação custo-benefício nas aplicações híbridas é melhor, devido a não replicação do código para cada plataforma*.

Apesar de focar em análise de segurança no acesso à web por aplicações móveis, o trabalho de Hale e Hanson [20] apresenta dados concretos sobre quantidade de dispositivos e acesso à internet, além de uma tabela completa que diferencia os aplicativos híbridos dos nativos.

Choudhary *et al.* [9] apontam diferentes ferramentas utilizadas para testes em aplicações móveis e seus tipos de abordagens. Realiza um experimento com 60 aplicações reais e procura identificar aspectos de cobertura, eficácia, compatibilidade e facilidade de uso. Seus resultados provêm uma boa visualização do estado da arte na área.

Charkaoui *et al.* [8] e Heitkotter *et al.* [22] se preocupam em apresentar um qua-

dro acerca dos principais frameworks utilizados no desenvolvimento de aplicações híbridas (cross-platform), com destaque para o *Titanium* e o *Phonegap*.

Hu *et al.* [23] apresenta uma ferramenta (AppDoctor), criada para realizar testes funcionais exploratórios em aplicações mobile. O AppDoctor suporta diferentes tipos de entrada para a aplicação, como duplos cliques e cliques longos, rotação, etc. Foi realizado um estudo empírico e encontrados 72 bugs por meio da ferramenta. As possibilidades apontam para o amadurecimento de testes em aplicações móveis, com assertivas direcionadas para este contexto. Entretanto, o trabalho é focado em aplicações nativas em Android, deixando de lado as híbridas e baseadas em web.

Escoffier e Lalanda [15] apresentam um Framework (h-ubu) que representa modelos da aplicação em javascript. Testes baseados em modelos que podem ser facilmente cambiáveis, facilitando a evolução e manutenção das aplicações. Foca nas aplicações híbridas, utilizando javascript e HTML5, mas não realiza experimentos para validar o seu uso.

Haller [21] utiliza testes centrados no usuário, e um serviço na nuvem para gerar casos de teste e os automatiza com o uso de um framework (Swisscom IT). Os testes detectaram falhas nos aplicativos, principalmente durante sua inicialização, que poderiam ser evitados com simples testes de regressão.

Collins *et al.* [12] apresentam um quadro contendo os tipos de teste utilizados em experiências com a indústria. Apesar de não conduzir um experimento em si, os autores falam sobre experiências valiosas no processo de construção de aplicações móveis para a indústria.

O trabalho de Singer *et al.* [41] é uma pesquisa qualitativa que estuda o uso do *Twitter* por desenvolvedores de software. O *design* do seu estudo incorpora elementos de Teoria Fundamentada e inspira a etapa de coleta de dados deste trabalho.

Melo *et al* [31] descreve em seu estudo as relações existentes entre a academia e indústria quanto à documentação e verificação de conformidade arquitetural em projetos de desenvolvimento de software. Sua pesquisa é qualitativa e utiliza uma abordagem proposta por Braun e Clarke [7] conhecida como *Análise Temática*. Esta mesma abordagem foi utilizada neste estudo para guiar a análise das entrevistas e elaboração do roteiro do *survey* confirmatório.

Capítulo 4

Descrição do Método

Este capítulo descreve os passos metodológicos planejados e executados neste estudo. As seções a seguir detalharão o público alvo, as questões de pesquisa e o tipo de análise utilizada em cada etapa.

4.1 Planejamento da Pesquisa

O planejamento da pesquisa foi iniciado juntamente com o levantamento do estado da arte sobre o tema de desenvolvimento e testes em aplicações móveis. Os autores estudados, conforme descreve o Capítulo 3, demonstraram que há interesse da academia no tema e que ainda restam esclarecimentos sobre a real prática dos desenvolvedores quanto à verificação e validação dos *apps*

Além dos estudos bibliográficos, foram pesquisados repositórios *opensource*, como o *F-Droid*¹, que continham aplicações móveis de código aberto, a fim de identificar os testes automáticos que estavam presentes nos projetos.

Foi observado que aplicações móveis disponíveis como *K-9 Mail*², *ChatSecureAndroid*³ e *Writeily-pro*⁴, entre outras, possuem poucos ou nenhum arquivo de testes automatizados ou guias para o uso de testes manuais. A Tabela 4.1 aponta alguns dos softwares visitados que possuíam testes automáticos. Como esta etapa era apenas para reconhecimento, não

¹Disponível em: <https://f-droid.org/>

²Disponível em: <https://f-droid.org/packages/com.fsck.k9/>

³Disponível em: <https://github.com/guardianproject/ChatSecureAndroid>

⁴Disponível em: <https://github.com/plafue/writeily-pro>

foram armazenados dados sobre os demais *apps* que não possuíam testes automáticos em seus repositórios.

Tabela 4.1: *Apps Open-source* Visitados

App	Descrição	Testes Automáticos	Ferramentas	SLOC
ChatSecure	Mensageiro	Unitários, Stress	<i>JUnit, Monkey</i>	61714
andStatus	Rede Social	Unitários, GUI	<i>JUnit, ActivityMonitor</i>	53047
Flym	Notícias	Unitários	<i>JUnit</i>	12687
ownCloud	News Reader Notícias	Unitários, GUI	<i>JUnit, Roboelectric, GradleTestRunner</i>	23496
Writeily-Pro	Notas	Unitários	<i>JUnit</i>	5946
K9mail	Email	Unitários, GUI	<i>JUnit, Roboelectric</i>	104405
TurtlePlayer	Multimídia	Unitários	<i>JUnit</i>	13111

Baseados nos levantamentos bibliográficos e na quantidade baixa de testes automáticos encontrados nos repositórios visitados, foram elaboradas as perguntas de pesquisa.

As primeiras duas perguntas: *Como os desenvolvedores lidam com a automação dos testes de aplicações móveis?* e *Quais os tipos de testes automáticos mais utilizados pela comunidade?* buscam investigar se ainda há a preferência por testes manuais apontada na literatura [25] e se a diversidade das ferramentas atuais de teste [17] estão cumprindo o seu papel na prática dos desenvolvedores.

O foco em automação está presente nas duas primeiras perguntas de pesquisa devido à intenção inicial dos pesquisadores em verificar o motivo da falta de *scripts* de testes automatizados nos repositórios visitados. Estas perguntas foram mantidas por comporem um reflexo da prática dos desenvolvedores e testadores que o estudo pretende retratar.

A terceira pergunta de pesquisa: *Qual o grau de satisfação dos participantes, em geral, com as atuais práticas de teste?* busca identificar se o processo de desenvolvimento de aplicações móveis está incluindo o planejamento e execução dos testes, de maneira que os participantes dos projetos permaneçam confiantes da qualidade do software desenvolvido.

A confiança e satisfação dos participantes nas práticas de teste realizadas podem expor o nível de eficácia das ferramentas de teste disponíveis e atualmente utilizadas, o conhecimento das equipes acerca das técnicas de teste e como isto afeta os *apps* desenvolvidos.

São aspectos que representam a maturidade das equipes e desenvolvedores e devem estar presentes em um estudo que pretende descrever o *status* da prática de testes em aplicações móveis.

A quarta pergunta: *Como é a relação entre a literatura e a prática de testes?* busca verificar se as contribuições da academia, por meio de artigos e desenvolvimento de ferramentas, são conhecidas e consultadas pelos participantes no seu cotidiano.

Compreende-se que a literatura científica, em qualquer área do conhecimento, não tem como principal objetivo criar conceitos práticos para serem aplicados diretamente na indústria, entretanto, existem diversas ferramentas elaboradas no meio acadêmico que possuem aplicações e contribuições significativas para a prática de testes. Exemplos de tais ferramentas são o AppDoctor [23], o MTControl [13], o Karma [24], entre outras também citadas no capítulo 3 desta dissertação.

A pergunta: *Como os desenvolvedores lidam com os desafios encontrados?* pretende verificar se os desafios encontrados na literatura [32; 27; 25] são os mesmos encontrados pelos desenvolvedores na prática e qual os procedimentos utilizados para vencer os obstáculos.

Esta pergunta é utilizada para identificar não apenas quais são os desafios, como por exemplo ferramentas ineficazes, fragmentação dos ecossistemas ou o uso de testes manuais ou automatizados. Além disto, o estudo pretende compreender como estes obstáculos são superados, quais os principais meios utilizados para que o funcionamento do *app* não seja comprometido em diferentes sistemas operacionais ou que novos desafios que não estão na literatura podem ser destacados.

A partir das perguntas de pesquisa descritas, foram planejadas as etapas seguintes do estudo, os roteiros que nortearam os *surveys* e as entrevistas.

4.1.1 Estudo Piloto

A partir das perguntas de pesquisa e das pesquisas bibliográficas foram construídos roteiros para os *surveys* exploratório e confirmatório e entrevistas.

A fim de verificar a existência de erros e inconsistências nos questionários e diminuir o tempo necessário para as respostas de cada participante, foi feito um estudo piloto com voluntários integrantes de projetos de desenvolvimento de um laboratório da Universidade Federal de Campina Grande.

O piloto consistia em disponibilizar aos voluntários uma versão do questionário baseada nos roteiros elaborados previamente pelos pesquisadores. As respostas dos participantes não fariam parte da análise dos resultados do estudo, mas forneceram feedback sobre o tempo médio de respostas e sugestões de melhorias fornecidas pelos voluntários.

Era desejado que o tempo de resposta de um questionário não fosse superior a 10 minutos, para que os participantes reais do estudo não fossem desencorajados a responder. Sendo assim, quando o estudo obteve um tempo de resposta igual ou maior que o limite estipulado, o roteiro era reduzido para se adequar ao tempo esperado e as modificações apropriadas eram implantadas e uma nova sessão do piloto era realizada.

As versões finais dos roteiros estão presentes nos apêndices desta dissertação, juntamente com o termo de consentimento das entrevistas.

4.1.2 *Snowball Sampling*

O método de preparação e distribuição de um questionário possui custos relativamente baixos para o pesquisador e, potencializado pela *Internet*, é capaz de coletar uma grande quantidade de respostas.

O *survey* funciona muito bem quando pretende coletar respostas de indivíduos que não estão relacionados, por exemplo, uma pesquisa de satisfação sobre um novo tipo de produto pode alcançar diversos tipos de usuários, com diferentes níveis de renda e escolaridade, de grupos completamente distintos. Entretanto, para grupos específicos a estrutura social, como ambientes de trabalho compartilhados e interesses em comum, não podem ser desconsiderada e tende a ampliar o alcance da pesquisa [11].

Coleman [11] e Goodman [19] apresentaram o conceito de *Snowball Sampling*, que consiste em recrutar participantes para o estudo partindo de um grupo inicial de participantes denominados *sementes*. Estes participantes iniciais, tal como é o caso do presente estudo, geralmente fazem parte da lista de contatos dos pesquisadores ou são identificados em empresas ou nas redes sociais. As sementes indicarão novos participantes que, por sua vez, também podem indicar novos respondentes sucessivamente. As Figuras 4.1, 4.2 e 4.3 ilustram esta técnica.

De acordo com Dudovskiy [14], a técnica de *SnowBall Sampling* pode ser utilizada de formas diferentes, entre elas:

Snowball Sampling Linear é caracterizada pela limitação de cada participante de identificar apenas um candidato para participar da pesquisa, o processo é então repetido até que a amostragem esteja completa.

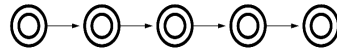


Figura 4.1: Snowball Sampling Linear

Snowball Sampling Exponencial e Não-Discriminante é o método que alcança a maior quantidade de respondentes, pois cada semente pode indicar vários respondentes, aumentando a base de dados exponencialmente. O pesquisador não elege critérios para seleção dos indivíduos.

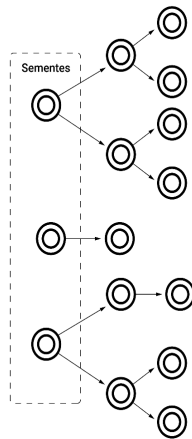


Figura 4.2: Snowball Sampling Exponencial Não-Discriminante

Snowball Sampling Exponencial Discriminante tem o funcionamento similar ao Exponencial Não-Discriminante, entretanto, apenas um dos candidatos indicados pelas sementes deve ser selecionado por meio de critérios definidos previamente pelo pesquisador.

Este método é indicado para ser utilizado em populações desconhecidas ou raras, onde os indivíduos não foram reconhecidos previamente ou não são facilmente encontrados [34].

No caso do presente estudo, a população de desenvolvedores de *apps* não é rara nem desconhecida, contudo esta observação é válida para os indivíduos que atuam como testadores

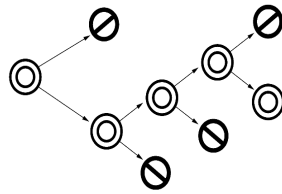


Figura 4.3: Snowball Sampling Exponencial Discriminante

nas equipes. Como veremos nas seções de resultados e discussão desta dissertação, os papéis de testador e de desenvolvedor muitas vezes não são desempenhados pelo mesmo profissional, quando existem equipes dedicadas para testes por exemplo. Outro aspecto importante para a utilização de Snowball sampling em nossa pesquisa é a nítida ampliação da taxa de respostas para os surveys.

Na primeira etapa, do *survey* exploratório, Utilizamos uma versão do *Snowball Sampling* Exponencial Não-Discriminante para alcançar a maior quantidade possível de respondentes da comunidade, já nas etapas de entrevistas e *survey* confirmatório os participantes indicados pelas sementes foram filtrados, de modo a mesclar as melhores características das técnicas de *Snowball Sampling* Discriminante e Não-Discriminante.

4.2 Execução do Estudo

A metodologia utilizada para coleta dos dados do estudo seguiu uma sequência de passos descritos e também utilizados por Melo *et al* [31] e Singer *et al* [41]. Sendo a elaboração de um *survey* exploratório, uma sequência de **entrevistas** e um *survey* confirmatório.

O *survey* exploratório teve como objetivo identificar o perfil de estudantes, profissionais e pesquisadores de técnicas de desenvolvimento e testes em aplicações móveis assim como as práticas de testes neste cenário. O questionário foi enviado diretamente para universidades, empresas e institutos de tecnologia.

Nesta etapa, era permitido ao participante indicar outros profissionais para participarem do estudo, baseado no método *Snowball Sampling* [19]. Os participantes indicados eram convidados pessoalmente pelo pesquisador para colaborar com o estudo. Ao todo foram obtidas 44 respostas e 28 participantes se voluntariaram para a etapa de entrevistas.

A etapa de **entrevistas** pretendia investigar as informações coletadas durante o *survey*

exploratório, permitindo ao participante detalhar a sua experiência com o processo de desenvolvimento, documentação e execução dos testes, as ferramentas utilizadas e os principais desafios encontrados no contexto móvel. Cada um dos participantes que se voluntariaram foram contatados individualmente para realizar a entrevista. 12 participantes foram entrevistados.

As entrevistas realizadas foram transcritas e a análise dos dados se baseou no processo de Análise Temática [7], com a criação de códigos e o agrupamento em temas para correlacionar as respostas dos participantes, identificar padrões e levantar afirmações para a etapa do *survey* confirmatório.

O ***survey* confirmatório** foi elaborado a partir de um roteiro que sumariza as conclusões obtidas nas etapas anteriores por meio de afirmações a serem julgadas pelos participantes. O objetivo foi verificar o grau de concordância dos participantes com as afirmações e realizar a comparação com os resultados do *survey* exploratório e das entrevistas.

Para esta última etapa, o questionário foi disponibilizado *online* e enviado para profissionais de desenvolvimento *mobile*, evitando o reenvio para participantes anteriores do estudo que possuíam menor tempo de experiência. Ao todo 54 respostas foram coletadas, sendo apenas 5 respondentes conhecidos participantes das etapas anteriores.

4.3 Considerações Finais

A metodologia empregada na coleta de dados do estudo foi construída a partir de estudos bibliográficos, elaboração de roteiros, questionários e entrevistas, e se baseou no método qualitativo de Análise Temática para avaliar seus resultados e torná-los um retrato mais fiel da realidade dos desenvolvedores *mobile*. Devido a não repetir os envios na etapa do *survey* confirmatório, cerca de 90 desenvolvedores foram alcançados. Os demais resultados de cada etapa, as discussões e conclusões do estudo serão apresentados nos próximos capítulos.

Capítulo 5

Resultados

Este capítulo apresenta os principais resultados de cada etapa da pesquisa. Serão caracterizados a quantidade de respostas, o perfil dos participantes, o processo de transcrição e codificação das entrevistas e o nível de concordância dos participantes com as afirmações levantadas.

5.1 Survey Exploratório

O *survey* elaborado continha 19 perguntas, organizadas para coletar dados como a empresa ou universidade a que o participante estaria vinculado, a sua experiência com desenvolvimento e testes em outras tecnologias e ambientes, práticas comuns de testes e automação. O participante também poderia se candidatar para a etapa de entrevistas e preencher um dos campos com o seu contato (campos opcionais).

O envio do *survey* se deu inicialmente através da lista de contatos dos autores, e, em um segundo momento, a partir da indicação de outros profissionais pelos próprios participantes, caracterizando o método *Snowball Sampling* [19].

5.1.1 Dados profissionais dos participantes

Ao todo, foram coletadas 44 respostas. Foram identificados 17 profissionais trabalhando exclusivamente em empresas e 27 trabalhando em pesquisas relacionadas ao desenvolvimento *mobile* ou em projetos de parceria entre empresas e universidades (Institutos de Tecnologia).

A média de experiência com desenvolvimento de software informada pelos participantes foi de 7 anos, sendo que menos de 21% possuíam menos de 4 anos de experiência. A experiência com testes de software também foi questionada e foi observada a média de 2,3 anos.

Ao serem questionados sobre as funções que já desempenharam durante os projetos que participaram, em uma questão de múltipla escolha, 68% dos participantes informaram que trabalharam com desenvolvimento, assim como também 68% dos participantes informaram que ocuparam o papel de testador. No campo "Outros" os participantes também informaram os cargos de *Líder de desenvolvimento*, *Administrador de Servidores e Suporte*, *Líder de Teste* e *Acessor de Projeto em CMS*.

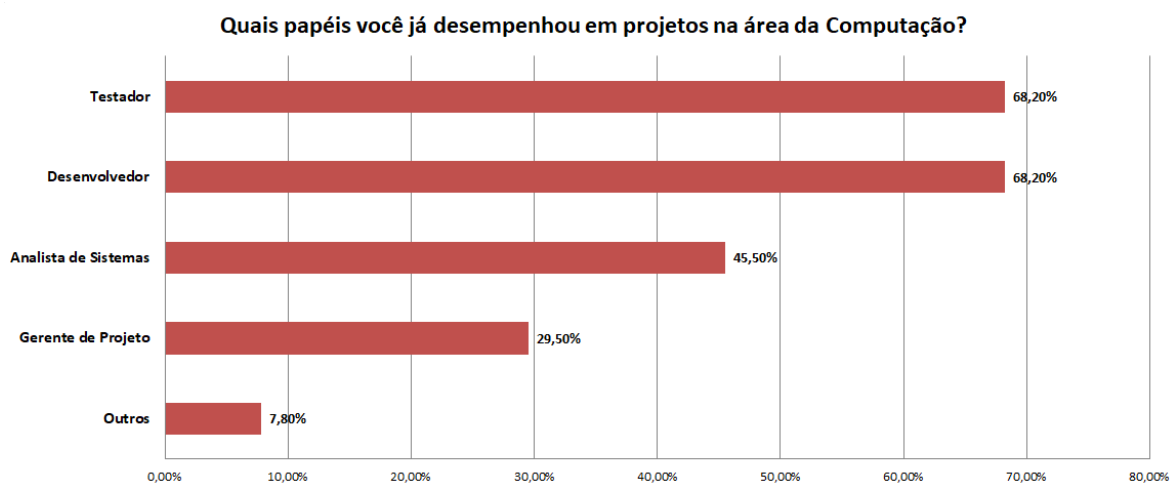


Figura 5.1: Cargos desempenhados pelos participantes

Era desejável também conhecer quais as plataformas móveis para as quais o participante desenvolveu. A plataforma *Android* foi a mais indicada, com 77% do total de respostas. O desenvolvimento híbrido, utilizando tecnologias web, como *Javascript*, *HTML* e *CSS* foi o segundo mais indicado, com 39% das respostas. A plataforma *iOS*, da Apple foi citada por apenas 5% dos respondentes. Os participantes que não haviam desenvolvido para nenhuma plataforma móvel foram filtrados para não fazer parte da etapa de entrevistas. A Figura 5.2 ilustra as respostas obtidas.

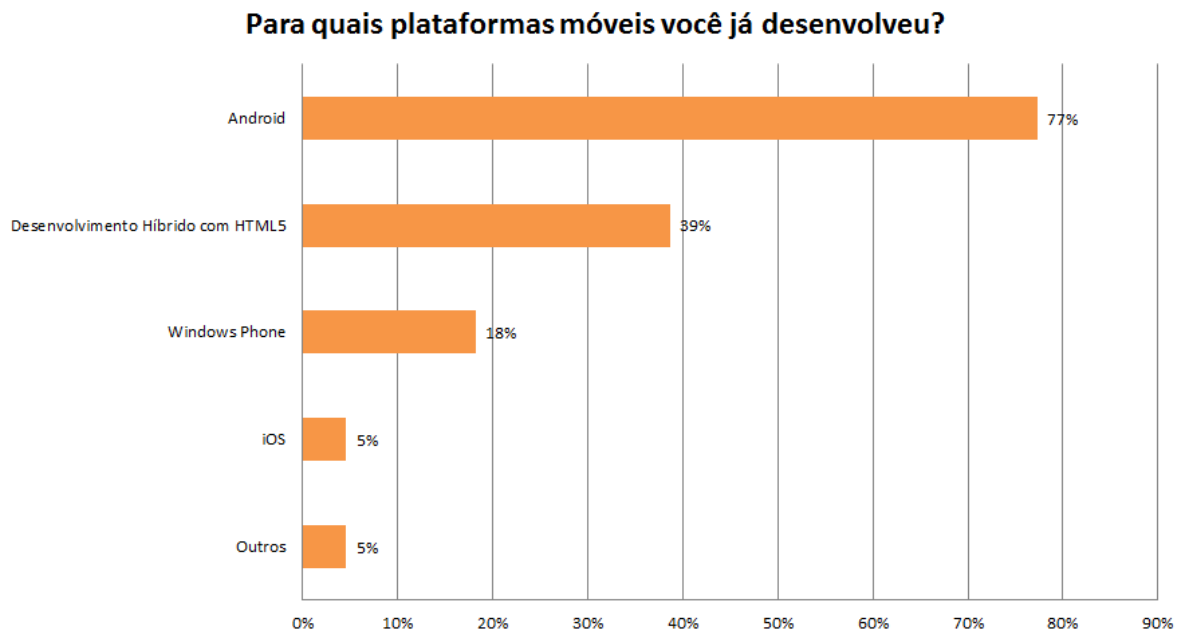


Figura 5.2: Experiência dos participantes em diferentes plataformas móveis

5.1.2 Papéis desempenhados pelos participantes

A divisão dos papéis de desenvolvedor e testador nas equipes pode ser um fator que influencia o nível de automação dos testes. Os testadores teriam tempo disponível para criação de *scripts*, planejamento das etapas de teste e seleção adequada das ferramentas a serem utilizadas no processo. Ademais, a identificação da experiência do participante também seria um fator a ser examinado durante a etapa de entrevistas e traria respostas às duas primeiras questões de pesquisa do estudo.

Ao investigar a semelhança entre os resultados das respostas dos papéis de testador e desenvolvedor, foi observado que 46% dos participantes que informaram terem ocupado o cargo de desenvolvedor também ocuparam o cargo de testador. Esta observação pode evidenciar que a divisão entre estes papéis na prática não é clara. Esta afirmação é reforçada pelo posicionamento de um dos participantes em um campo de detalhamento sobre as equipes de teste.

Existem processos distintos. Mas a relação entre desenvolvedor e tester é muito próxima. [Participante 16]

Ainda seguindo a linha de divisão entre os papéis de testador e desenvolvedor, um dos questionamentos da pesquisa é se é comum entre os times de desenvolvimento a existência

de uma equipe específica para testes. 45% dos participantes responderam que a instituição a que eles estavam vinculados mantinha equipes separadas para esta atividade enquanto 55% informaram que não havia uma equipe separada.

Na sua instituição, existe equipe separada para testes?

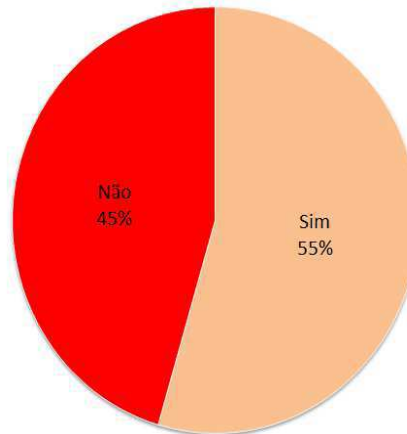


Figura 5.3: Equipes Separadas para testes

Aos participantes também foi permitido detalhar o fluxo de desenvolvimento e comunicação entre as equipes. Mais de 16% dos participantes informaram que as equipes trabalham juntas e, por vezes, as equipes se revezam nas atividades, dependendo do projeto e da etapa do desenvolvimento, um ou mais integrantes são alocados para realizar os testes necessários. Um dos participantes explicou:

Quando um projeto novo é fechado, um testador é alocado nesse projeto. A partir daí o testador trabalha junto com os desenvolvedores nos Sprints. [Participante 25]

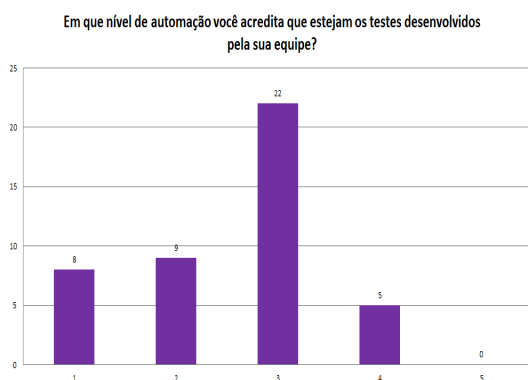
Da mesma forma, um outro participante também detalhou que, apesar de existir uma equipe separada, ela está presente durante todo o desenvolvimento do software.

Sim. A equipe de testes interage diretamente com o time de desenvolvimento para compreender desde a concepção as necessidades de validações. [Participante 27]

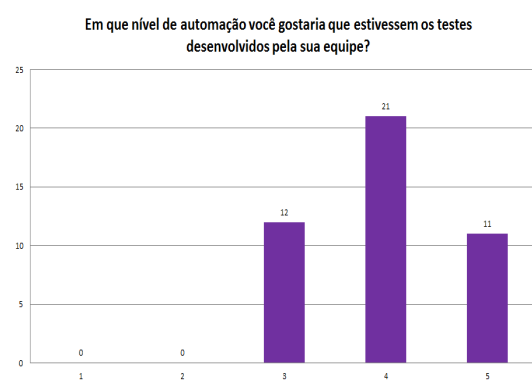
5.1.3 Automação de Testes

A relação dos desenvolvedores com a automação de testes está presente nas duas primeiras questões de pesquisa do estudo e foi abordada desde o *survey* exploratório, no qual os participantes responderam a duas perguntas sobre o nível de automação em suas equipes.

A primeira pergunta apresenta uma escala de 1 a 5, sendo o nível 1 testes totalmente manuais e o nível 5 testes totalmente automatizados. Foi descrito aos participantes que o nível 3 representaria uma abordagem híbrida em que testes automatizados seriam utilizados juntamente à técnicas de validação manual.



(a) Níveis de Automação nas equipes



(b) Níveis desejados pelos participantes

A segunda pergunta indica qual o nível que o participante desejaria que estivesse a automação de testes em suas equipes. Seguindo a mesma escala da pergunta anterior. Os resultados estão resumidos nas Figuras 5.4a e 5.4b.

As respostas indicam que os níveis de automação existentes nas equipes estão, em sua maioria, seguindo um processo híbrido, em que testes manuais e automatizados são utilizados paralelamente. Entretanto, os participantes indicaram que gostariam de um nível de automação 4 ou 5, sem fazer uso de técnicas de validação exclusivamente manuais.

A diferença entre o nível de automação presente nas equipes e o nível desejado pelos participantes indica uma insatisfação com as práticas de teste existentes nas equipes no tocante à automação. Este aspecto é importante para o objetivo do estudo em sua terceira questão de pesquisa, que pretende identificar o grau de satisfação dos desenvolvedores com as atuais práticas de teste.

Perguntamos aos participantes quais os tipos de teste automáticos que eles consideravam mais eficazes para detecção de falhas nos aplicativos. Foi permitido escolher mais de um

tipo de teste e, caso não utilizassem testes automáticos, os participantes poderiam escolher a opção "Apenas testes manuais". cerca de 70% dos respondentes afirma que os testes unitários e de interface são mais eficazes e mais de 22% escolheram apenas testes manuais. A opção "Outros" contou com 29% das respostas, nas quais foram citadas técnicas de análise de cobertura, testes de estresse, testes de aceitação e testes de compatibilidade. A pergunta não teve como objetivo separar os tipos e níveis de teste.

5.1.4 Ferramentas de Testes

As ferramentas automáticas de testes utilizadas pelos desenvolvedores são outro aspecto da automação no contexto móvel. O uso de uma ou outra ferramenta indica quais os tipos e níveis de teste que estão sendo mais frequentemente automatizados, assim como o uso de poucas ferramentas ou apenas abordagens de teste manual indicam insatisfação com as ferramentas existentes ou mesmo a falta de conhecimento acerca dos recursos disponíveis. Retratar estes aspectos é parte do objetivo deste estudo.

As respostas da seção anterior, sobre a automação dos testes, se mostraram coerentes com os tipos de ferramentas utilizadas pelos respondentes. Uma das questões do *survey* listava uma sequência de ferramentas conhecidas pelos autores e solicitava ao participante a marcação das ferramentas utilizadas por eles. As ferramentas *Selenium* e *Monkey*, que são utilizadas para testes de interface gráfica e performance, respectivamente, foram as mais citadas, seguidas da ferramenta *Robotium*, também de automação de testes de interface. No campo "Outros" foram citadas em destaque as ferramentas *Appium* e *Espresso*.

A ferramenta *AppDoctor*, por sua vez, não foi citada pelos participantes. Divulgada com resultados no trabalho de Hu *et al* [23], mas não conhecida entre os participantes.

5.1.5 Literatura

De acordo com a quarta questão de pesquisa, que tem como objetivo a compreensão da relação entre a literatura e a prática das equipes, uma sessão do *survey* foi direcionada ao tipo de literatura utilizada pelos desenvolvedores como base para criar cenários de testes no contexto móvel.

Os participantes responderam, em sua maioria (75%), que utilizam a literatura comercial,

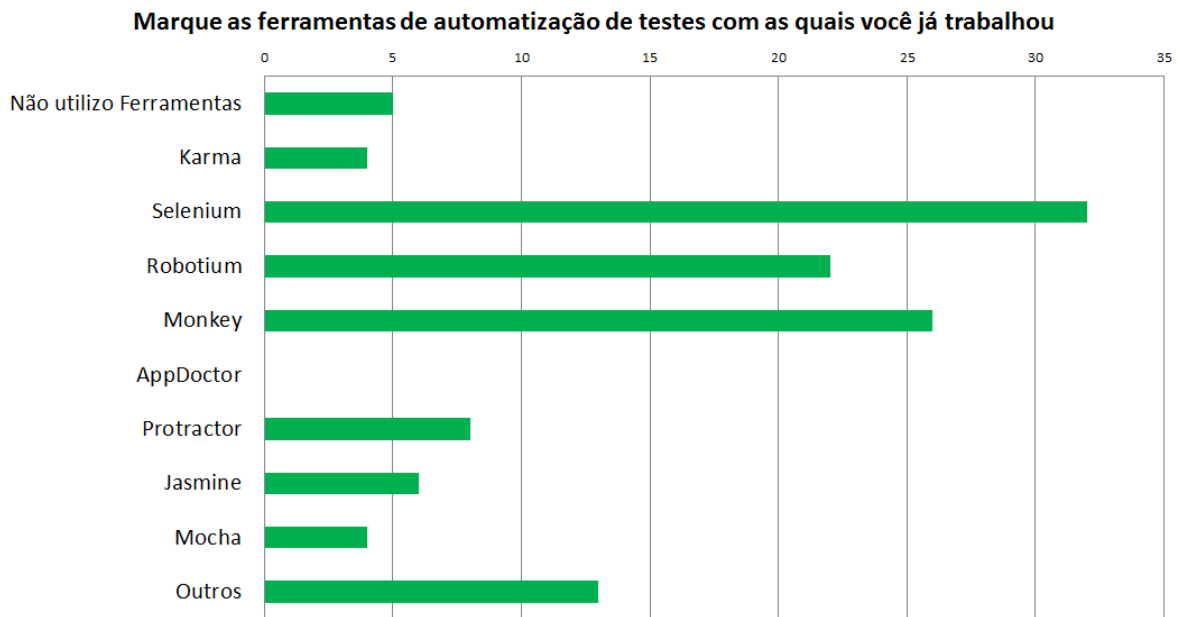


Figura 5.5: Ferramentas utilizadas pelos respondentes

como sites, blogs e revistas. A literatura científica ficou em segundo lugar, com 50% e a opção "Guias e Manuais da própria empresa" obteve 34% das respostas.

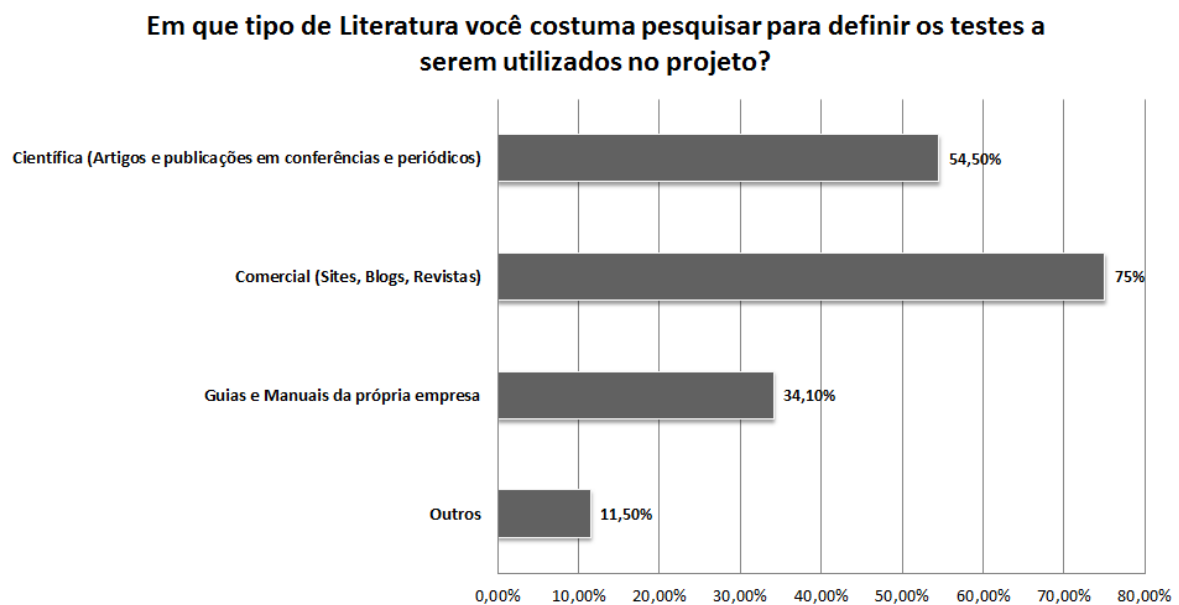


Figura 5.6: Tipo de literatura pesquisada pelos participantes

Entendendo que a maioria dos participantes esteja vinculado a projetos em institutos de Universidades, e ,apesar de não estarem alocados em empresas privadas, a maior parte dos participantes (como verificado nas etapas posteriores de entrevistas e *survey* confirmatório)

faz parte de projetos conveniados a estas empresas e desempenham papéis semelhantes de desenvolvimento, análise, gerência de projetos e testes de software. Acreditamos que a taxa de respostas de 50% para a literatura científica foi influenciada diretamente pelo ambiente acadêmico no qual estão imersos os participantes. Este aspecto foi melhor filtrado nas próximas etapas e será apresentado nas próximas seções deste capítulo.

A última parte do *survey* perguntava ao participante se ele desejava participar da etapa de entrevistas da pesquisa. 28 respondentes (64%) se candidataram e informaram um meio de contato.

5.2 Entrevistas

O início das entrevistas se deu logo após a análise do *survey* exploratório. Os principais questionamentos coletados foram organizados em um roteiro, disponível no Apêndice B desta dissertação. As entrevistas foram guiadas pelo roteiro e seguiram um padrão semi-estruturado, no qual as perguntas e a sua sequência podiam ser modificadas de acordo com a interação com o entrevistado. Este modelo de entrevista é geralmente utilizado combinado com métodos de análise qualitativos, como a Teoria Fundamentada e a Análise Temática [40].

O objetivo das entrevistas foi detalhar questionamentos levantados durante a análise do *survey* exploratório, como a experiência dos participantes, o perfil profissional e o processo detalhado de testes utilizado pelas equipes, desde a documentação até a execução, incluindo o tipo de literatura utilizada para encontrar cenários de testes adequados para utilizar nas aplicações.

As principais seções das entrevistas eram a **identificação** do participante, na qual, ao iniciar a gravação, o pesquisador indicava o número respectivo do participante no questionário da etapa anterior. Esta seção servia para guiar possíveis perguntas que surgiriam de acordo com a interação com o entrevistado. A segunda seção era uma caracterização da **experiência do participante**, de modo a esclarecer qual a interação que ele teve com o desenvolvimento de aplicações em geral e as diferenças percebidas com relação ao contexto móvel. A terceira etapa era o detalhamento do processo de **testes de aplicativos móveis**, desde o planejamento até a execução em sua equipe, incluindo os principais desafios encontrados e as ferramentas

utilizadas. Um resumo dos objetivos e perguntas feitas aos participantes está presente na Tabela 5.1.

Seção	Descrição	Exemplos de Perguntas
Identificação	Respectivo número do participante no survey exploratório, para facilitar identificação	Não há
Experiência	Principais experiências do participante. Identificar tecnologias, cargos, formação.	<ul style="list-style-type: none"> - Fale um pouco sobre a sua experiência na computação e o desenvolvimento de aplicações móveis. - Para quais plataformas desenvolveu? - Quais as principais diferenças que você encontrou ao realizar testes no contexto móvel em relação às aplicações desktop ou web?
Testes em Apps	Conversa geral sobre testes em aplicações móveis, desde o planejamento até a execução.	<ul style="list-style-type: none"> - As práticas de teste em sua equipe incluem/incluam um plano de testes? - Você costuma automatizar os testes? - Existe alguma ferramenta ou técnica de teste que você utilizou, mas não ficou satisfeito com os resultados? - Na sua experiência, quais os desafios atuais para garantir a qualidade do software no contexto móvel?

Tabela 5.1: Seções das Entrevistas e Exemplos de Perguntas Presentes no Roteiro

Todos os participantes que se candidataram na etapa do *survey* exploratório foram contatados individualmente, de acordo com o meio de contato informado no questionário. Dos 28 candidatos voluntários, 12 responderam o contato e realizaram a entrevista. Uma taxa de 43% de respostas.

Os participantes responderam a um email contendo um termo de pesquisa que assegurava o sigilo de dados que pudessem identificá-los, mas permitia a divulgação dos resultados. O email continha a descrição da pesquisa, a identidade e o contato dos pesquisadores responsáveis. Uma cópia está disponível no Apêndice C.

A duração das entrevistas teve, em média, 30 minutos. Todas as conversas foram gravadas e transcritas, para que fosse possível codificar as respostas obtidas.

As perguntas levaram em consideração a experiência do participante com desenvolvimento de software, em geral, e sua experiência com o desenvolvimento e testes de aplicações móveis.

5.2.1 Codificação

O processo de codificação, conforme explicam Braun e Clarke [7] implica em uma leitura e familiarização do pesquisador com os dados escritos e a criação de códigos para incluir os trechos lidos. Conforme ilustra a Figura 5.7.

Ao todo, foram criados 29 códigos. A próxima etapa foi a busca por *temas*, um conjunto de códigos que se referem a trechos similares. Cada código pode ser entendido no formato:

{

- P: É comum esse papel de testador? Equipes separadas só para testes?
- E: Na empresa na qual trabalho hoje, o INDT (antigo Instituto Nokia de Tecnologia - INdT), temos equipe dedicadas a testes dependendo do projeto pode ter 2 ou 3 testers alocados para isso e um tester, no máximo fica alocado em 2 projetos ao mesmo tempo, quando necessário e sim, isso é comum em empresas de desenvolvimento de software hoje em dia (e necessário para se garantir a qualidade do que é produzido)
- P: No caso o desenvolvedor fica responsável pelos testes unitários e depois a responsabilidade é dos testers?

Participante: Desenvolvedor

[A7] Comentário: Trabalho Empresa

[A8] Comentário: Experiência: Empresa INDT (Instituto Nokia de Tecnologia)

[A9] Comentário: Equipe de testes Separada, "para garantir qualidade"

Figura 5.7: Leitura dos textos e início da codificação

```

“nome”,
“tema”,
“descrição”,
“trecho”: [{"entrevista”, “texto”, “resumo”}, ...]
}

```

Após o agrupamento por temas, a codificação foi organizada e mantida em uma planilha, juntamente com os trechos referentes.

Tema	Código	Entrevista	Trecho	Observações
Experiência	Tecnologias	10	desde 2004 com desenvolvimento mobile .,Minhas primeiras plataformas mobile foram JavaME (Nokia Series 40/S60, Samsung, Motorola, Sony Ericsson e Siemens) depois fui para o Symbian (JavaME e C++) Android e Windows Phone. e recentemente participei de alguns projetos voltados para desktop (Windows/Linux/Mac), usando Qt (C++) e, no momento, estou trabalhando em um projeto voltado para Desktop (Windows only) com Windows Forms/WPF	JavaME, Symbian, C++, Android, Windows Phone, Windows, Linux, Mac, Qt
Experiência	Cargo	10	Bom, trabalho com desenvolvimento, desenvolvimento desde 99-2000 inicialmente com desenvolvimento web e desde 2004 com desenvolvimento mobile .	Atua como Desenvolvedor
Experiência	Formação	10	Mestrado concluído	Informação conhecida pelo pesquisador, encontrada no currículo do entrevistado

Figura 5.8: Exemplo da organização de um tema

As seguintes subseções deste capítulo se baseiam nos temas observados nesta etapa, e são a experiência dos participantes, a relação com testes de software no contexto mobile, a

relação dos entrevistados com a literatura de testes, documentação e desafios encontrados pelos desenvolvedores no contexto móvel.

5.2.2 Experiência dos participantes

A experiência dos participantes da etapa de entrevistas foi analisada de acordo com as informações de formação, cargo, tipo de instituição a que estava vinculado, conhecimento em diferentes tecnologias e plataformas e tempo de experiência.

Todos os participantes da etapa de entrevistas eram profissionais com ensino superior completo, pós-graduação, mestrado ou doutorado na área de Computação. 5 participantes trabalhavam em empresas do setor privado, 6 participantes trabalhavam como profissionais em projetos conveniados a empresas e apenas 1 (um) participante trabalhou apenas na universidade, como pesquisador na área de desenvolvimento de software *mobile*.

A maior parte dos entrevistados trabalhou com diversos tipos de sistemas, como sistemas *Web*, baseados em *HTML5*, *CSS* e *Javascript*, por exemplo, e tecnologias mais voltadas para *Desktop* como e *C++*. Todos trabalharam com desenvolvimento ou testes de *apps*.

O tempo de experiência em computação variou de 2 anos a 20 anos, enquanto que a experiência em *mobile* variou de 0,15 anos (2 meses) a 5 anos. Nota-se que, mesmo participantes com maior tempo de experiência geral possuem menos tempo de experiência com desenvolvimento *mobile*, provavelmente devido à sua recente emergência. Estas evidências estão presentes também nos *surveys* exploratório e confirmatório, onde apenas 7% dos respondentes afirmaram possuir mais de 4 anos de experiência.

Na Tabela 5.2, que resume as informações descritas, a identificação dos participantes se refere ao respectivo índice de resposta ao *survey* exploratório, a fim de facilitar a busca de informações complementares que não foram repetidas durante as entrevistas.

5.2.3 Testes em *Apps*

As práticas de testes e sua automação foram aspectos observados durante as entrevistas. Os códigos analisados para este tema foram o planejamento dos testes, se existe ou não e como é construído, os tipos e níveis de testes utilizados, se há preferência por testes automáticos ou manuais e a experiência do participante com as ferramentas de teste disponíveis na

Participante	Formação	Instituição	Tecnologias	Tempo Geral	Tempo Mobile
02	Mestrado	Empresa/Universidade	Delphi, HTML, CSS, Pearl, PHP, ASP.NET, Android Nativo, Windows Phone, Phonegap	>2 anos	1,5 anos
10	Mestrado	Empresa	JavaME, Symbian, C++, Android, Windows Phone, Windows, Linux, Mac, Qt	17 anos	12 anos
14	Mestrado	Universidade	Android, Android Studio	10 anos	0,15 anos
15	Graduação Completa	Empresa/Universidade	Android, "Web"	18 anos	1 ano
16	Graduação Completa	Empresa/Universidade	Android, "Desktop", Qualidade de Software	3 anos	0,25 anos
17	Doutorado	Empresa/Universidade	Sistemas Web (HTML, CSS, Javascript), Sistemas embarcados, Clipper, DOS, Android, Android Studio, iOS	20 anos	4 anos
18	Graduação Completa	Empresa	Web, Android, Windows Phone	2,5 anos	0,5 anos
27	Pós-graduação	Empresa/Universidade	Android, Web	6 anos	3 anos
28	Graduação Completa	Empresa	Android, iOS, Windows Phone.	5 anos	5 anos
34	Graduação Completa	Empresa	Cordova (HTML5, Javascript), Android, iOS	10 anos	1 ano
39	Graduação Completa	Empresa	Test Manager, Visual Studio, Windows, Android, JUnit	2 anos	2 anos
42	Graduação Completa	Empresa/Universidade	Android	2,5 anos	2,5 anos

Tabela 5.2: Experiência dos Participantes

comunidade.

Cinco participantes informaram que não havia um plano de testes específico que guiasse a execução dos testes durante as etapas do projeto. Roteiros baseados em casos de uso são utilizados ou uma simples validação manual da aplicação. Os entrevistados que informaram que suas equipes possuem documentação de planejamento dos testes afirmaram que ela era elaborada no início do projeto e sofria modificações de acordo com o andamento do projeto.

A existência de equipes de testes também pode variar de acordo com a instituição e com a quantidade de funcionários disponíveis. O participante 17, que possui maior experiência em geral (20 anos) e experiência em mobile acima da média dos outros participantes (4 anos) explicou mais detalhadamente esta interação entre o planejamento e as equipes:

...Já trabalhei em sistemas onde eles [a empresa] tinham uma equipe para desenvolver e um time de validação de produto, um time de teste. Eu já trabalhei em sistemas onde era uma equipe pequena visando gerar um produto para lançar no mercado. Nesse cenário normalmente o foco era mais no desenvolvimento e os testes eram deixados para o final...depende mais do time, qual a equipe que você tem em mãos. [Participante 17]

Apenas 3 dos participantes entrevistados informaram não haver distinção entre os papéis de desenvolvedor e testador. Sendo que, dos 5 participantes que eram funcionários de empresas, apenas 1 informou que não havia equipes separadas para testes. A presença de planejamento e equipes de teste em empresas aponta para o amadurecimento da atividade de

teste de software no contexto móvel quando comparado aos resultados do estudo de Joorabchi *et al* [25].

5.2.4 Relações com a literatura

Um dos temas levantados durante a entrevistas foi a relação dos desenvolvedores e testadores com a literatura científica. Observando o crescimento de trabalhos relacionados aos testes em *apps*, as ferramentas e técnicas desenvolvidas pela academia estariam sendo consultadas pelas equipes como suporte para suas atividades?

Sobre a literatura utilizada pelo participante para fundamentar suas pesquisas em sobre os testes adequados, era pedido que ele detalhasse sua resposta já coletada anteriormente, durante o *survey* exploratório. Dos entrevistados, 83% utilizam a literatura comercial, como revistas, sites e blogs; 58% utilizam a literatura científica e apenas 25% utilizam guias e manuais da própria empresa. 50% dos participantes, apesar de trabalhar em projetos para empresas, estão alocados em ambientes geridos por universidades. Devido à semelhança entre o percentual que utiliza literatura científica e os participantes vinculados à projetos em universidades, é possível entender que o ambiente acadêmico influencia a utilização desta literatura. Dos cinco participantes que trabalham exclusivamente em empresas, dois não utilizavam a literatura acadêmica.

Artigos (científicos) são ótimos para lhe dar embasamento teórico sobre um determinado assunto . Livros e artigos em revistas (comercial), te auxiliam com problemas que você pode encontrar no dia a dia. [Participante 10]

A literatura científica eu acho que serve para outro contexto, é para o antes, talvez para o planejamento. [Participante 02]

A utilização mais evidenciada é a comunidade de desenvolvedores e fóruns como o *StackOverflow*¹, citado por todos os participantes. Outras comunidades também foram citadas, como o *Agile Testers*², *Experts Exchange*³ e os fóruns específicos das plataformas e a comunidade open-source.

¹Disponível em: <https://stackoverflow.com/>

²Disponível em: <http://agiletesters.com.br/>

³Disponível em: <https://www.experts-exchange.com/>

*Karma*⁴ é uma ferramenta para automação de testes em *Javascript*, utilizada em testes de aplicações móveis híbridas, que são baseadas em tecnologia *web* como *HTML5*, *CSS3* e *Javascript* com algum *framework* como o *Phonegap* para criar a interface entre a camada *web* e o hardware do dispositivo, citado pelos participantes do *survey* exploratório e pelos entrevistados 02 e 15, este *framework* foi idealizado a partir de um trabalho de mestrado [24].

5.2.5 Documentação

Outro tema analisado durante as entrevistas foi o detalhamento das equipes acerca da elaboração de requisitos e a utilização desta documentação para guiar a validação e verificação do software construído. Além disso, as diferenças encontradas ao documentar um aplicativo móvel em comparação a um sistema *web* ou *desktop*.

Em alguns casos, a documentação também é utilizada para lidar com a fragmentação, um dos participantes comentou:

Tem alguns requisitos quando ele [o cliente] pede pra gente: "versão 4.4 para cima". Então a gente tem que dar suporte a partir da 4.4. Isso é obrigatório.
[Participante 27]

O planejamento dos testes, com a descrição dos cenários e o uso ou não de instituições de referência para criar os casos de testes também foram analisados como parte da documentação.

5.2.6 Desafios Encontrados

Este tema surgiu a partir da declaração dos entrevistados sobre os obstáculos encontrados no desenvolvimento e testes das aplicações móveis. As declarações dos participantes englobam principalmente a variedade de versões de software e hardware dos dispositivos, a automação dos testes e a cultura de teste nas equipes.

O problema mais frequentemente citado foi a fragmentação. Apesar de diferentes abordagens para facilitar a reescrita de código para diferentes sistemas operacionais como o desenvolvimento híbrido, utilizando *HTML5* e um *framework* como o *Phonegap*, ainda há desafios

⁴Disponível em: <https://karma-runner.github.io/1.0/index.html>

para os desenvolvedores no sentido de preparar um software capaz de se adaptar às diferentes características dos dispositivos, como tamanhos de tela e interações com o usuário e com o contexto.

Abaixo seguem as citações que fazem parte dos códigos que compõem o tema, transcritos das entrevistas:

Fragmentação dos ecossistemas, isso dificulta muito manter a qualidade, pois você não tem como testar em todos os aparelhos alvo e sempre vai ter um caso que você não testou. [Participante 10]

Eu acho que a qualidade de software no dispositivo móvel é muito desafiadora e complexa, por vários motivos. Dentre eles a variedade de modelos de dispositivos. Por mais que exista uma versão de android, mas existem variações dessas versões, então este é um ponto que eu acho complexo. [Participante 16]

Para aplicativo móvel você começa a “explodir” diferentes modelos de device, de marcas, então você tem resolução, tamanho de tela, capacidade de memória, versão de API, e principalmente no caso do Android . Então assim, o primeiro ponto que chama a atenção na hora é a parte de fragmentação, que causa um esforço gigantesco. [Participante 17]

Para lidar com este desafio, algumas abordagens são utilizadas:

Aquisição de dispositivos físicos: São adquiridos aparelhos que representem uma amostra próxima do dispositivo alvo (resolução principalmente, memória, etc.). O principal problema desta abordagem é o custo elevado para aquisição dos equipamentos.

Uso de Emuladores: Neste caso são utilizados softwares que simulam os ambientes gerados. Esta alternativa tem baixo custo, mas, de acordo com um dos participantes, esta prática pode não ser suficiente para capturar falhas em dispositivos reais. Apesar de ser a única opinião entre os entrevistados (os outros não relataram o uso de emuladores), este participante era o mais experiente em testes *mobile* (5 anos).

O que é muito importante e depende muito do custo do projeto: não usar emulador! Usar dispositivos físicos. Há uma grande influência de

layout e de regra de negócio também. Já peguei diversas falhas que não aconteciam em emuladores, mas aconteciam em dispositivos ou aconteciam nos emuladores, mas nos dispositivos não aconteciam. [Partipante 28]

Serviços de Teste (*Cloud Testing*): Seriam utilizados serviços *online* terceirizados para enviar as aplicações, que seriam testadas ou emuladas em diferentes dispositivos. Este tipo de serviço não parece ser bem difundido. Nas entrevistas, os participantes, em sua maioria, utilizavam dispositivos físicos próprios ou da empresa em alternativa. O *survey* confirmatório contou com uma afirmação específica sobre este tópico, para os desenvolvedores avaliarem.

Documentar as limitações do *app*: Dois participantes informaram o uso da própria documentação para informar os tipos de aparelhos e versões do sistema operacional em que a aplicação foi devidamente testada. Em casos em que sejam conhecidos os dispositivos em que o aplicativo será utilizado, esta prática pode ser uma alternativa, entretanto, ao disponibilizar o aplicativo em uma loja da plataforma, não há este controle.

Além disso, foi observada a necessidade de um amadurecimento dos desenvolvedores e testadores sobre o seu papel na construção de um software de qualidade. As práticas de testes utilizadas nas equipes devem partir de uma análise, planejamento dos testes e execução dos testes adequados para cada situação. Um dos participantes ressaltou como principal desafio em sua equipe:

Cultura. Porque ferramenta e tecnologia hoje já existe. Minha dificuldade hoje é mostrar pra o testador que ele não se limita mais a uma fase do desenvolvimento de software. Ele tem que estar presente desde a análise até a implantação...E também mostrar para o meu desenvolvedor que ele não pode ser só um desenvolvedor de software. Ele tem que ser um desenvolvedor de software com qualidade, ele tem que entender de testes, ele tem que automatizar junto comigo. Hoje é um problema mais cultural do que puramente tecnológico. [Participante 28]

Esta cultura foi citada por alguns entrevistados como o entrevistado 10, que afirma que

costuma criar os testes unitários e automatizar o processo antes mesmo de iniciar a implementação.

Eu adquiri o costume de antes de iniciar o desenvolvimento (codificação de uma funcionalidade), preparo logo os Unit Test, dessa forma agilizo meu trabalho e quando pronto, submeto o patch para ser testado pela equipe de testes a equipe de testes, geralmente, aguarda fecharmos uma versão para poder efetuar os testes.

Há também o desafio de lidar com um ambiente cuja prioridade é a comunicação. Logo, um software que não seja essencial, deve ser interrompido no evento de uma chamada telefônica ou chegada de uma mensagem de texto. Ao considerar nas características de portabilidade, o software também deve funcionar corretamente não apenas a qualquer momento, mas em qualquer lugar e utilização de sensores exponencia a quantidade de cenários que devem ser testados. Durante as entrevistas notou-se que os testes de sensibilidade ao contexto raramente são utilizados. Esta afirmação e outras estarão presentes no *survey* confirmatório para avaliação dos desenvolvedores.

5.2.7 Considerações Finais

Os resultados da análise das entrevistas apontam para divergências no perfil dos participantes. O perfil "Desenvolvedor" conhece poucas ferramentas de teste automático, preferindo testes funcionais manuais e se adapta ao contexto móvel utilizando recursos de outras áreas de sua experiência. O perfil "Testador" procura documentar o processo, criar versões seguras do sistema e conhece diferentes tipos e níveis de testes, além de utilizar ferramentas de automação. O perfil que une as duas características na amostragem desta pesquisa são participantes experientes na área de desenvolvimento e testes (com mais de 5 anos de experiência), que mesclam abordagens de validações manuais e automáticas, conhecem o processo e atendem a prazos mantendo a qualidade do software. Este perfil é minoria e corresponde a 16% da amostragem.

5.3 Survey Confirmatório

A terceira etapa da pesquisa foi um *survey* confirmatório, que resumizou as afirmações encontradas durante as etapas anteriores. Esta etapa obteve 54 respostas, sendo que houve reenvio apenas para 4 participantes da etapa anterior, que trabalhavam em empresas e possuíam maior experiência. Esta seção apresentará o perfil dos participantes, as afirmações incluídas no questionário e as avaliações feitas pelos respondentes.

5.3.1 Perfil dos Participantes

Devido às dúvidas que surgiram com a especificação da experiência dos participantes no *survey* exploratório, o novo *survey* incluiu o detalhamento do tempo de experiência e os cargos ocupados pelos participantes. É possível identificar estudantes que cursam mestrado ou doutorado, se são profissionais ou estudantes bolsistas e o tempo dedicado junto a equipes de desenvolvimento *mobile*.

Conforme ilustra o gráfico 5.9, a maior parte dos respondentes (76%) possui graduação completa, sendo que uma parte destes participantes está vinculada ou já concluiu alguma pós-graduação.

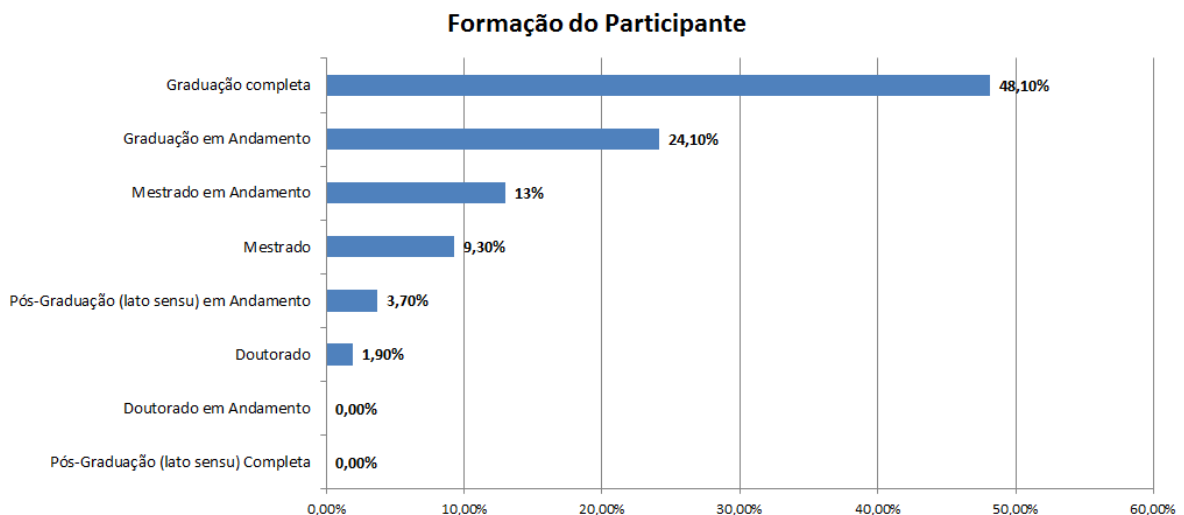


Figura 5.9: Formação dos participantes - *Survey Confirmatório*

Uma outra pergunta do questionário era em que tipo de instituição os participantes exerciam seu trabalho. A maioria dos participantes (mais de 40%) trabalha em empresas pri-

vadas, enquanto que outros 29% se declaram profissionais bolsistas que trabalham em convênios entre empresas e universidades. Outros participantes se declaram "desenvolvedores autônomos" (*freelancers*), gerentes de projeto e também estudantes bolsistas.

Apesar de 24% dos participantes informarem, na questão anterior, que ainda estão com a graduação em andamento, 11% dos participantes em geral declaram que são estudantes bolsistas participando de projetos de convênios entre universidades e empresas. Estas informações aumentam a confiança na experiência dos participantes, pois mesmo antes de sua formação estão trabalhando em projetos de desenvolvimento em contato com empresas.

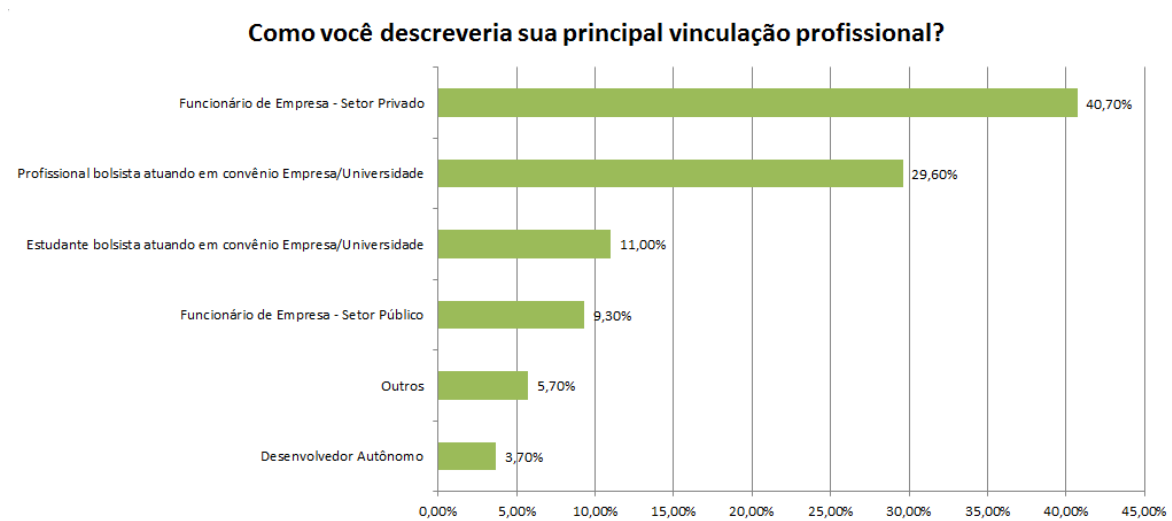


Figura 5.10: Vinculação profissional dos participantes - *Survey Confirmatório*

Além do tipo de instituição a que o respondente está vinculado, um outro fator é o papel que ele ocupa na empresa. Assim como no *survey* exploratório, a grande maioria (81%) se declara principalmente como "desenvolvedor". Nesta pergunta não foi possível selecionar mais de uma alternativa, como pôde ser feito na primeira etapa do estudo. Sendo assim, é possível perceber que o papel principal de testador ainda é um desenvolvedor que se divide nas tarefas de desenvolvimento e qualidade de software.

Sobre o tempo de experiência com desenvolvimento de aplicativos móveis, a maioria dos participantes possui entre 6 meses e 2 anos de experiência. Esta proporção é similar à encontrada nas etapas anteriores.

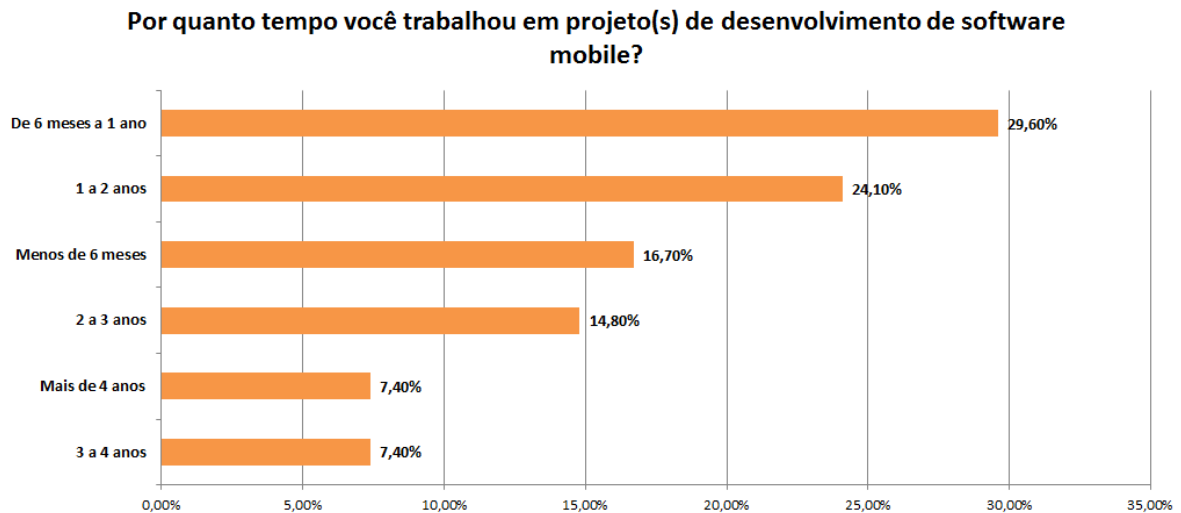


Figura 5.11: Tempo de Experiência dos Participantes - *Survey Confirmatório*

5.3.2 Frases e Avaliações

Partindo da experiência das entrevistas e dos dados do primeiro *survey*, foram elaboradas afirmações que deviam ser avaliadas pelos respondentes do *survey* confirmatório.

Estas frases foram apresentadas aos participantes e foi dada uma escala *Likert*, com valores de 1 a 5, onde 1 significava "Discordo Totalmente", 5 significava "Concordo Totalmente" e 3 significava "Neutro" ou "Não Concordo nem Discordo". A Figura 5.12 apresenta uma visão geral de todas as frases e avaliações dos respondentes. As subseções em seguida farão uma descrição detalhada destas frases, agrupando-as por temas.

Os grupos a que pertencem estas frases correspondem aos temas elaborados na etapa de entrevistas. Estes grupos e os resultados das avaliações também serão detalhados nas próximas subseções.

Satisfação

A fim de responder a uma das perguntas de pesquisa do estudo, o *survey* contou com a pergunta: *Qual o seu grau de satisfação com as práticas de teste desenvolvidas pela sua equipe?*. Observando o gráfico presente na Figura z, não é possível concluir que os participantes estavam satisfeitos ou insatisfeitos com as práticas.

No caso de dados em escala *Likert*, dados ordinais e não-paramétricos, foi escolhido o

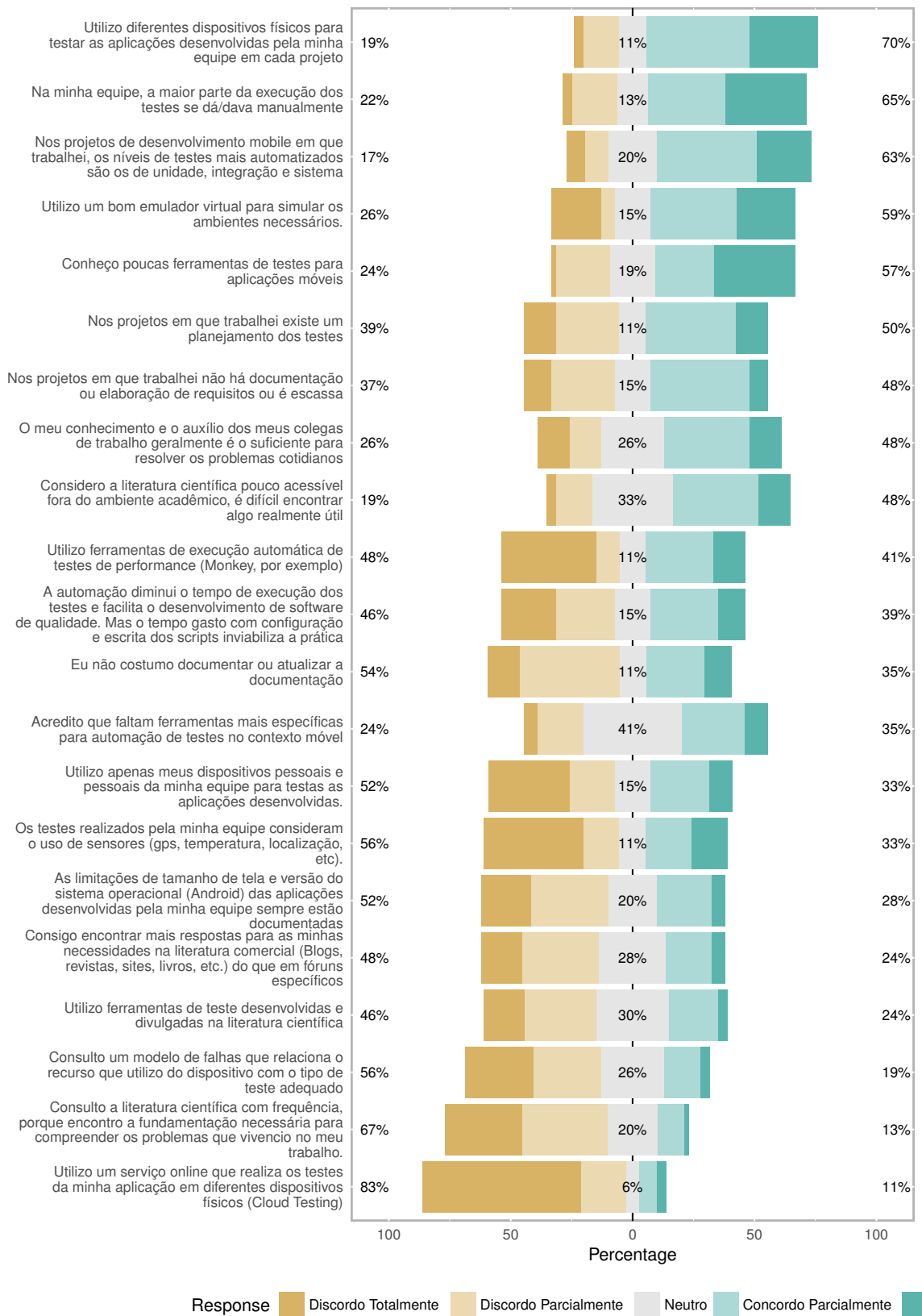


Figura 5.12: Respostas dos Participantes à Pesquisa (Likert)

teste de Wilcoxon⁵ para tirar conclusões sobre a população.

Para esta pergunta foi utilizado o teste bi-caudal, para saber se há diferença estatisticamente significativa entre as respostas e a mediana. As hipóteses nula e alternativa foram testadas.

- *H0*: Não há diferença estatisticamente significativa entre a mediana amostral e a mediana populacional.
- *H1*: Há diferença estatisticamente significativa entre as medianas.

Neste caso, o valor de *p-value* foi de 0,67, e não é possível rejeitar a hipótese nula. Apesar da maioria dos participantes informar não estar satisfeito com as práticas de teste de software nas equipes.

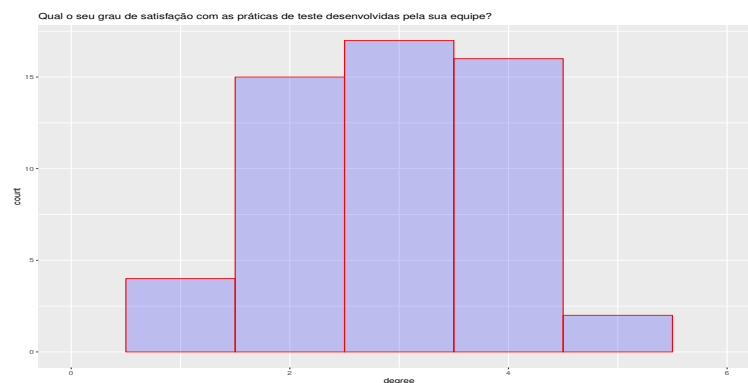


Figura 5.13: Respostas dos Participantes sobre a satisfação com a prática de testes

Código Fonte 5.1: testes em R

```
require(readr)

#read answers file
satisfacao <- read_csv("satisfaction.csv")

#generate graphics
boxplot(satisfacao$Q1)
hist(satisfacao$Q1)

#normality analysis
```

⁵Disponível em: <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/wilcox.test.html>

```
shapiro.test(satisfacao$Q1)
#RESULTS: W = 0.90278, p-value = 0.0003537

#wilcoxon test
wilcox.test(satisfacao$Q1, alternative = c("two.sided"), paired = FALSE,
            mu=3, conf.level = 0.95)
#RESULTS: V = 325, p-value = 0.6719
            #alternative hypothesis: true location is not equal to 3
```

Automação de Testes

Os níveis e tipos de testes de software aplicados no contexto móvel, assim como a sua automação, foram temas levantados durante as entrevistas, de onde surgiram as afirmações resumidas na Figura 5.14.

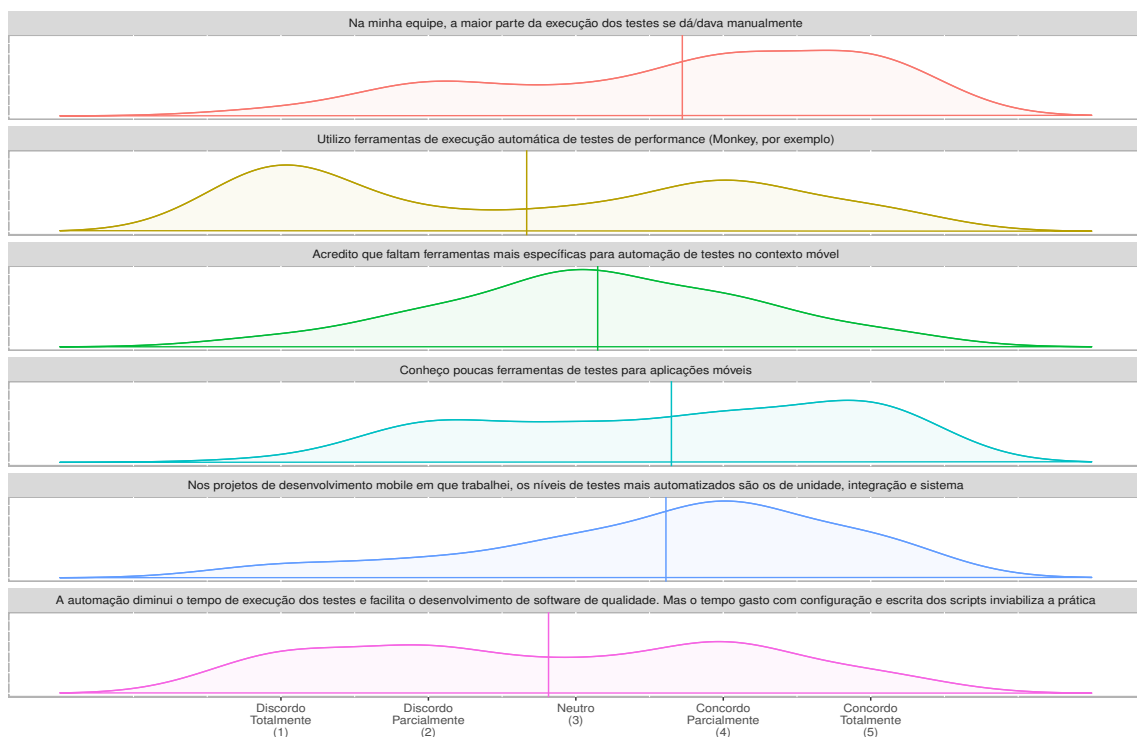


Figura 5.14: Respostas dos Participantes sobre automação dos testes

A pergunta *Na sua experiência, que tipos de testes são mais utilizados para garantir a qualidade das aplicações móveis?* obteve resultados similares aos encontrados no survey exploratório. A maior parte dos participantes (81,5%) afirmaram que testes funcionais são

os mais utilizados, seguidos por testes de performance (59%) e Testes Estruturais 46%. Uma quantia significativa dos participantes (mais de 11%) afirmou não ter conhecimento.

Com um número alto de respondentes afirmando a utilização de testes de performance, seria esperado que houvesse reflexo na utilização de ferramentas para testes automáticos de performance. Entretanto, mais de 48% dos respondentes discordaram da afirmação *Utilizo ferramentas de execução automática de testes de performance (Monkey, por exemplo)*. Como houve concordância de 40% dos participantes, e apenas 11% se posicionaram neutros, foi realizado um teste estatístico para determinar a qual a tendência de discordância ou concordância na amostra.

Código Fonte 5.2: Teste de Wilcoxon Aplicado à afirmação AF3

```
require(readr)

#read answers file
automation_test <- read_csv("automation.csv")

#normality analysis
shapiro.test(automation$AF3)
#RESULTS: W = 0.81794, p-value = 1.119e-06

#wilcoxon test
wilcox.test(automation$AF3, alternative = c("greater"), paired = FALSE,
            mu=3, conf.level = 0.95)
#RESULTS:
      V = 399, p-value = 0.9778
      alternative hypothesis: true location is greater than 3
```

Sendo assim, existe a preferência dos desenvolvedores por testes funcionais e testes de performance, mas não é possível afirmar que a maioria dos desenvolvedores utiliza ferramentas automáticas de teste de performance.

Neste sentido, o *survey* confirmatório também explorou a utilização de testes manuais entre os desenvolvedores com a afirmação *"Na minha equipe, a maior parte da execução dos testes se dá/dava manualmente"*. O grau de concordância dos participantes foi de 65%, enquanto que apenas 22% discordaram parcialmente ou completamente. Confirmando que há a preferência por testes funcionais e testes de performance, mas a execução dos testes é,

preferencialmente, manual.

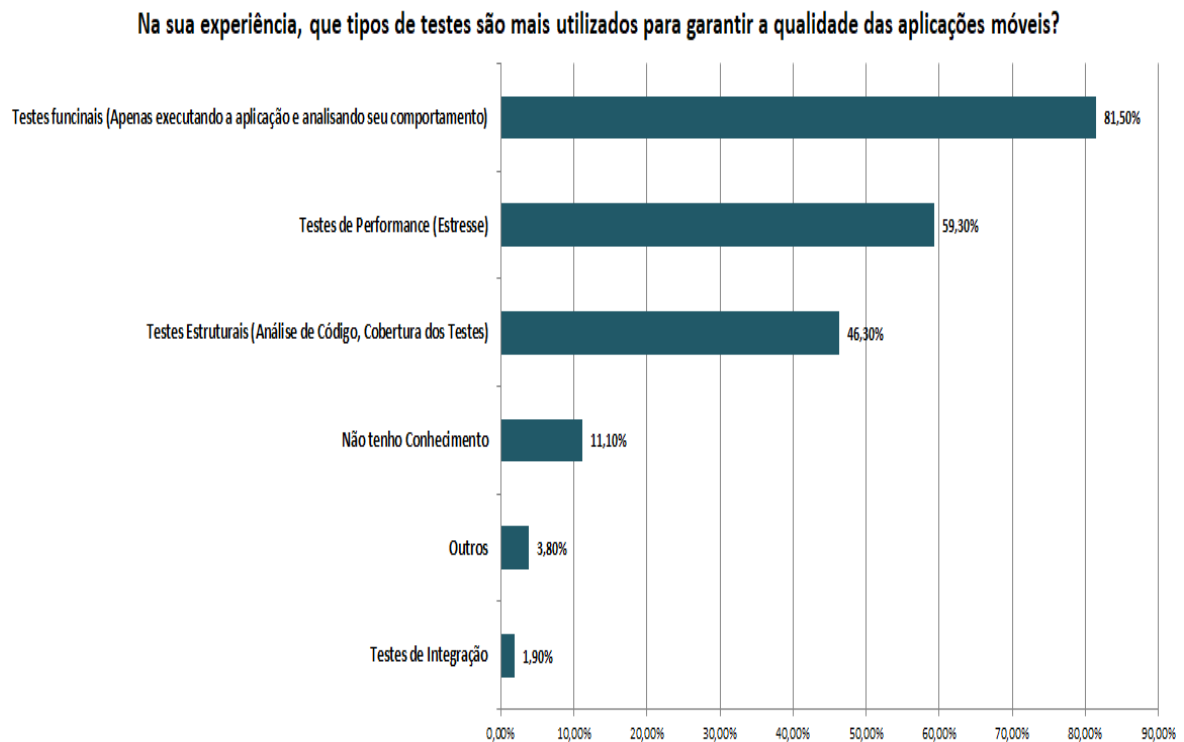


Figura 5.15: Respostas dos Participantes sobre os tipos de testes utilizados no contexto móvel

Um dos motivos para não utilizar as ferramentas de automação disponíveis pode ser o desconhecimento dos desenvolvedores sobre as ferramentas existentes. Mais de 40% dos participantes não se posicionaram na afirmação *Acredito que faltam ferramentas mais específicas para automação de testes no contexto móvel* e a afirmação *Conheço poucas ferramentas de testes para aplicações móveis* obteve mais de 57% de concordância.

Sendo assim, o uso de testes automáticos parece estar restrito aos testes funcionais e de performance, utilizando poucas ferramentas de automação, como o *Monkey*. A preferência dos participantes por testes manuais é evidente e pode estar ligada ao próprio desconhecimento de diferentes técnicas e ferramentas de teste.

Literatura

A relação dos participantes com a literatura foi analisada principalmente pela influência da literatura científica ou comercial no trabalho dos desenvolvedores e testadores. As 5 afirmações com este tema foram avaliadas pelos respondentes e estão resumidas no gráfico 5.16.



Figura 5.16: Respostas dos Participantes sobre a literatura consultada

Durante as entrevistas, foi notada a dificuldade de alguns participantes em encontrar as referências necessárias para aplicar em seus trabalhos. Para exemplificar esta dificuldade, o participante 14, estudante de doutorado na área de testes em aplicações móveis afirmou que não há informações suficientes sobre ferramentas e técnicas de teste na literatura científica.

Algumas ferramentas, poucas, encontro todas as informações, agora outras ferramentas já tem pouca informação sobre como utilizar, e técnicas é que não tem mesmo.

...Eu acredito que as aplicações móveis vieram a ser tão populares recentemente. Acho que agora a comunidade está começando a "tomar" uma atenção maior para as aplicações móveis. Como é uma "coisa" recente, talvez seja por isso que não tenha tanta informação ainda.

Com o objetivo de verificar estas observações sobre a utilização da literatura científica, foi apresentada a afirmação *Consulto a literatura científica com frequência, porque encontro a fundamentação necessária para compreender os problemas que vivencio no meu trabalho.* A avaliação dos participantes condisse com os depoimentos coletados nas entrevistas, mais

da metade (66%) dos respondentes discordaram da afirmação, enquanto que apenas 12% concordaram.

Outra afirmação avaliada pelos participantes, *Considero a literatura científica pouco acessível fora do ambiente acadêmico, é difícil encontrar algo realmente útil*, teve concordância de mais de 48%, e discordância de apenas 18%.

Durante a revisão bibliográfica deste estudo foram encontradas ferramentas de suporte à atividade de testes desenvolvidas pela comunidade acadêmica, como o *AppDoctor* [23], *MobiGuitar* [5], *DynoDroid Amal2014* e *MTControl* [13]. Entretanto, apesar de existência destas ferramentas, a sua "utilidade" foi questionada na etapa de entrevistas e no *survey* exploratório nenhuma delas foi citada. No *survey* confirmatório, a afirmação: *Utilizo ferramentas de teste desenvolvidas e divulgadas na literatura científica* obteve 46% de discordância e apenas 24% de concordância.

Em uma das perguntas da entrevista *Existe alguma ferramenta ou técnica de testes que você utilizou, mas não ficou satisfeito(a) com os seus resultados?*, um dos participantes chegou a citar o *MobiGuitar*.

O que a gente usou e teve dificuldade foi o *MobiGuitar*, é uma ferramenta de pesquisa que ela visa a criação automática de testes, mas ela tinha muita limitação. Imagino que a limitação dela é por conta de ser um protótipo acadêmico.

[Participante 17]

Estes dados apontam o distanciamento da literatura científica e a prática de testes das equipes. Diferentemente do *survey* exploratório, os respondentes foram confrontados com afirmações ao invés de apenas citar o tipo de literatura utilizada, o que nos permite uma análise mais detalhada dos motivos.

A amostra de participantes desta etapa também possui mais indivíduos que integram o quadro de funcionários de empresas privadas e não estão alocados em projetos de universidades, o que pode ter refletido na diferença nos resultados.

Ao mesmo tempo em que a literatura científica parece estar distante da prática dos desenvolvedores, a literatura comercial parece estar mais próxima. Nas entrevistas, um dos participantes comentou:

Quando você faz uma busca nas principais ferramentas de busca na *web*,

você geralmente não encontra este tipo de literatura [científica], você encontra mais referências em blogs, em sites especializados. [Participante 34]

A maioria dos participantes citou o uso da comunidade de desenvolvedores, em fóruns como o *StackOverFlow* para resolver os problemas cotidianos do profissional. Estes resultados inspiraram a afirmação *Consigo encontrar mais respostas para as minhas necessidades na literatura comercial (Blogs, revistas, sites, livros, etc.) do que em fóruns específicos*. A avaliação dos participantes do *survey* confirmatório resultou em uma discordância de 48%, contra 24% de concordância. Enquanto que a afirmação *O meu conhecimento e o auxílio dos meus colegas de trabalho geralmente é o suficiente para resolver os problemas cotidianos* obteve 48% de concordância entre os participantes.

Como os valores de diferença entre concordância e discordância dos participantes foram muito altos (sempre próximo do dobro de respondentes), não foi necessário utilizar testes estatísticos. Observa-se a preferência dos desenvolvedores em utilizar fóruns e a comunidade *open-source* em detrimento da literatura científica ou comercial.

Sendo assim, os dados apontam que devido, principalmente, à sua atividade recente, a literatura científica ainda não alcançou o espaço entre as equipes de desenvolvimento de aplicações móveis. As ferramentas desenvolvidas e divulgadas no meio acadêmico não são conhecidas pelos participantes. A literatura comercial, como sites, blogs e revistas são utilizadas pelos desenvolvedores, mas o material disponibilizado na comunidade e entre os colegas de trabalho ainda é a maior fonte de referências para as equipes.

Documentação

A documentação nas equipes de software envolve desde elementos de elaboração de requisitos até o planejamento e execução dos testes e a apresentação das limitações presentes para garantir a qualidade das aplicações construídas.

Para analisar estas características, foram elaboradas 5 afirmações, baseadas nas etapas anteriores de entrevistas e *survey* exploratório. O resumo das respostas coletadas está presente na Figura 5.17.

Nas entrevistas, houve divergências sobre a existência da documentação e planejamento dos testes em projetos de aplicações móveis. Em geral, os participantes informaram que a

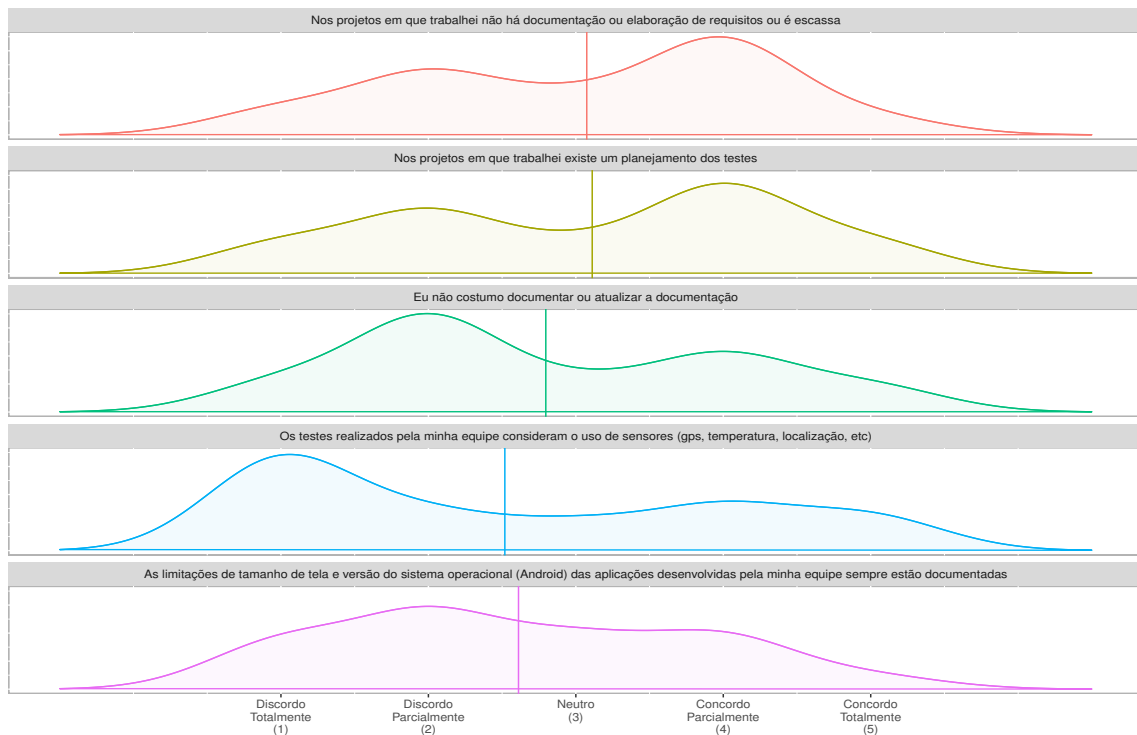


Figura 5.17: Respostas dos Participantes sobre a documentação e cenários de teste

documentação, quando existe, é semelhante ao de uma aplicação *web* ou *desktop* e que o planejamento dos testes se dá a partir dela.

A afirmação *Nos projetos em que trabalhei não há documentação ou elaboração de requisitos ou é escassa* obteve mais de 48% de concordância e 37% de discordância. Utilizando a mesma metodologia de testes estatísticos apresentados nas seções anteriores, com o teste de Wilcoxon, não foi possível afirmar a ausência de documentação nas equipes.

Ainda de acordo com este resultado, a afirmação *Eu não costumo documentar ou atualizar a documentação* teve discordância de 53% entre os respondentes e apenas 35% de concordância.

De acordo com 50% dos participantes o planejamento dos testes também está presente no processo das equipes. Nas entrevistas, os participantes informaram a existência dos testes, mas, em times que contavam com equipes separadas especificamente para testes, o planejamento e execução de testes ocorriam separadamente e os desenvolvedores geralmente desconheciam os detalhes do processo.

Embora o planejamento e a documentação dos *apps* pareça ser uma prática comum entre

as equipes de desenvolvimento, a característica de sensibilidade ao contexto dos dispositivos não é considerada nas atividades de teste. A afirmação *Os testes realizados pela minha equipe consideram o uso de sensores (gps, temperatura, localização, etc)* obteve discordância de mais de 55% dos participantes.

Na etapa de entrevistas, foi observado que uma das alternativas para lidar com a fragmentação era utilizar a documentação para apresentar as limitações dos *apps* e evitar problemas futuros de utilização do *software*. Contudo, a afirmação *As limitações de tamanho de tela e versão do sistema operacional (Android) das aplicações desenvolvidas pela minha equipe sempre estão documentadas* obteve 55% de discordância entre os participantes e obteve suporte dos testes estatísticos.

Fragmentação

A fragmentação interna e externa foi o principal desafio apresentado pelos participantes das entrevistas. Este desafio tornou-se um tema e foram elaboradas 5 afirmações que foram avaliadas pelos participantes nesta etapa. O resumo das respostas está presente na Figura 5.18.

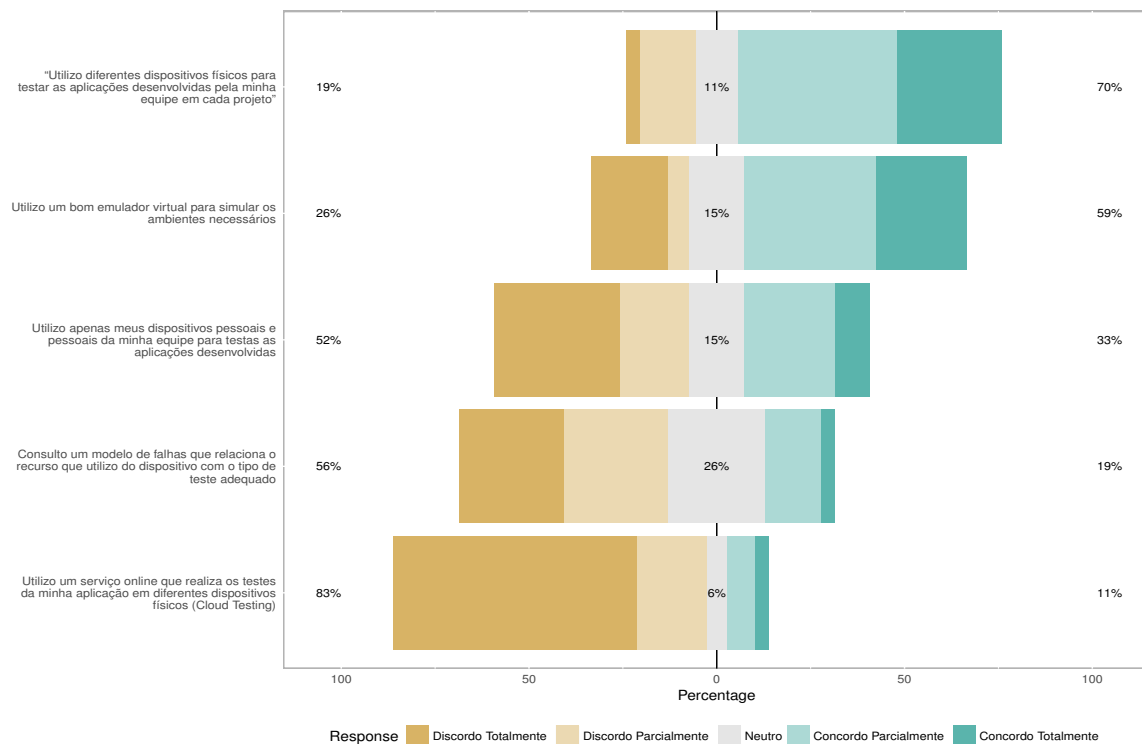


Figura 5.18: Respostas dos Participantes sobre a fragmentação

O participante das entrevistas que possuía mais experiência geral informou um problema comum. Segundo ele, não existe um modelo de falhas bem conhecido para a elaboração de cenários de testes no contexto móvel. Apesar da existência de uma instituição de referência, a *App Quality Alliance* [13] e de trabalhos como o de Nascimento *et al.* [13], os desenvolvedores parecem não utilizar um modelo que facilite a elaboração dos cenários. A afirmação *Consulto um modelo de falhas que relaciona o recurso que utilizo do dispositivo com o tipo de teste adequado* obteve 55% de discordância e 25% dos respondentes não quiseram se posicionar. Uma modelagem das falhas seria um dos primeiros passos para lidar com a fragmentação.

A alternativa mais utilizada atualmente é testar o *app* em diferentes dispositivos físicos. A afirmação *Utilizo diferentes dispositivos físicos para testar as aplicações desenvolvidas pela minha equipe em cada projeto* alcançou 70% de concordância entre os respondentes do questionário. 2% discorda totalmente e 14% discordam apenas parcialmente. Um dos participantes detalhou sua confiança nesta alternativa:

"Testar a aplicação em diferentes dispositivos é fundamental para o perfeito funcionamento em escala de milhares de usuários." [Participante 16]

Em outra afirmação, *Utilizo apenas meus dispositivos pessoais e pessoais da minha equipe para testar as aplicações desenvolvidas*, pôde-se observar que os dispositivos físicos utilizados não são, em sua maioria, os dispositivos pessoais do próprio testador ou equipe, restando então a aquisição dos equipamentos pela própria empresa. 51% dos participantes discordaram desta afirmação enquanto que apenas 33% concordaram parcialmente ou totalmente.

O uso de emuladores foi notado durante as entrevistas, e criticado por um dos participantes, que informou que eles seriam menos confiáveis do que os dispositivos físicos. Esta alternativa também foi analisada nesta etapa com a afirmação *Utilizo um bom emulador virtual para simular os ambientes necessários*. Esta afirmação teve 59% de concordância. 35% dos respondentes informaram que concordavam parcialmente com esta afirmação, o que pode significar que utilizam um emulador, mas não o consideram adequado. Apenas 25% dos participantes discordaram desta afirmação.

A utilização de serviços *online* terceirizados de testes de aplicações móveis (Cloud Tes-

ting) foi a alternativa menos utilizada entre os participantes. 64% dos participantes discordaram totalmente da afirmação *Utilizo um serviço online que realiza os testes da minha aplicação em diferentes dispositivos físicos (Cloud Testing)*. Outros 18% discordaram parcialmente, totalizando cerca de 82% de discordância, enquanto que apenas 11% concordaram parcialmente ou totalmente desta afirmação.

5.4 Ameaças à validade

As afirmações presentes em nosso estudo estão limitadas às características da amostra de profissionais coletada. Parte dos profissionais que responderam aos *surveys* e foram entrevistados não são funcionários exclusivos da indústria, mas atuam em convênios, alocados em instalações das universidades. Além disso, são residentes apenas no Brasil, devido, inclusive às limitações de idioma. Também não há medida de quantos profissionais existem efetivamente na área de qualidade de *software mobile* e, portanto, não é possível estabelecer uma quantidade considerada estatisticamente significativa para generalizar todas as conclusões. Estas limitações caracterizam ameaças à validade externa do estudo

Na tentativa de amenizar este efeito, os *surveys* foram enviados para todo o Brasil, obtendo a maioria das respostas nas regiões Norte e Nordeste. O mesmo foi feito com as entrevistas, que contaram com dois participantes residentes na região Norte.

Quanto à distribuição do *survey* exploratório, utilizamos inicialmente a rede de contato dos pesquisadores para propagar o estudo, e os participantes envolvidos, o que pode caracterizar uma ameaça à validade interna. Além disso, na etapa do *survey* confirmatório, foram convocados 5 participantes que já haviam sido contatados nas etapas de *survey* exploratório e entrevistas.

Contudo, as afirmações do *survey* confirmatório não foram retiradas diretamente do discurso e das respostas de cada candidato individualmente, a participação de diferentes respondentes forneceu os indícios necessários para a construção de uma afirmação. Os 5 participantes que foram contatados para responder na etapa do *survey* confirmatório possuíam experiência mais alta e sua contribuição nas respostas seria valiosa.

Ademais, A taxa de respostas inicial foi baixa, sendo a maioria das respostas dos participantes indicados através do método *Snowball Sampling*. Para amenizar o efeito de re-

lacionamento entre os primeiros participantes e os pesquisadores, o *survey* confirmatório foi enviado para fóruns e comunidades de desenvolvimento conhecidas apenas na etapa de entrevistas por meio dos participantes.

A ameaça de constructo, no entanto, está presente no estudo, visto que está limitada à capacidade dos pesquisadores de elaborarem roteiros que capturem com maior objetividade as respostas necessárias às conclusões da pesquisa e na etapa de análise temática das entrevistas apenas um pesquisador foi responsável pela codificação das transcrições.

Esta última ameaça foi amenizada pela consultoria dos colegas e orientadores que contribuíram com revisões importantes no processo de construção dos roteiros e questionários.

5.5 Considerações Finais

Este capítulo apresentou os principais resultados da execução do estudo, de acordo com a metodologia descrita no capítulo 4. O *survey* exploratório teve o objetivo de visualizar as primeiras impressões sobre o tema, verificar o perfil e coletar dados sobre ferramentas e satisfação dos desenvolvedores de software *mobile* quanto à atividade de testes em suas equipes e também recrutar voluntários para a etapa de entrevistas.

A etapa de entrevistas, por sua vez, foi realizada utilizando o padrão semi-estruturado e teve dois objetivos principais: detalhar as informações coletadas durante o *survey* exploratório e, a partir daí, propor temas para serem avaliados durante o *survey* confirmatório.

O *survey* confirmatório contou com os temas alçados durante as entrevistas para planejar afirmações que foram submetidas à avaliação dos participantes. A análise dos resultados permitiu responder as questões de pesquisa do estudo e analisar as ameaças à validade.

De acordo com as respostas obtidas, é indicada a preferência dos desenvolvedores por testes manuais e, quando há testes automatizados nas equipes, são utilizadas ferramentas de testes em nível funcional, como o *Monkey*. As equipes específicas para testes existem em parte considerável das instituições, mas a tendência é que os próprios desenvolvedores realizem os testes, sendo o papel de testador alternado durante o processo de desenvolvimento.

Capítulo 6

Discussão

Após a coleta e análise dos resultados, Este capítulo tem o objetivo de resumir os principais resultados e afirmações do trabalho. O formato escolhido para apresentação das conclusões é a interpretação dos dados de acordo com as perguntas de pesquisa que nortearam o estudo.

6.1 *RQ1: Como os desenvolvedores lidam com a automação dos testes em aplicações móveis?*

Em todas as etapas do estudo, já a partir do *survey* exploratório, foi percebida a preferência dos desenvolvedores por testes manuais em detrimento dos testes automatizados.

Conforme apontam os dados do primeiro *survey*, a maioria dos participantes afirmou que gostaria que houvesse mais automação em suas equipes. Desta forma, acreditamos que os benefícios de automatizar o processo de testes são conhecidos entre os desenvolvedores.

Posteriormente, durante as entrevistas, participantes declararam que o prazo curto para entregas e as constantes modificações nos requisitos são a principal causa da falha na elaboração de scripts automatizados.

Ao ser questionado sobre o que seria feito caso retomasse um dos projetos de desenvolvimento móvel em que atuava como testador, um dos participantes comentou:

Continuaria com o teste no meu cenário de desenvolvimento, mediante que eu não tinha muito tempo para explorar e investigar métodos de automação para dispositivos móveis. [Participante 16]

Outro participante informou que por não participar de projetos "críticos" sua equipe optou por não implantar testes automatizados.

...Realmente, móvel não tem automatização. Já tentei iniciar alguma coisa com Karma, mas devido ao orçamento, tempo, inclusive criticidade dos projetos que trabalhei. Não eram projetos críticos. Então terminávamos não fazendo.
[Participante 02]

Como os prazos curtos dificultariam a automação dos testes, era esperado que os desenvolvedores tivessem problemas na elaboração dos *scripts* e, portanto, esta etapa subtrairia tempo precioso do processo de desenvolvimento.

No entanto, No *survey* confirmatório não foi possível dar suporte estatístico à hipótese de que a demora na escrita dos *scripts* de teste automático inviabilizaria a prática. Embora a maioria dos participantes tenha concordado com esta afirmação e um dos participantes afirmar que não havia tempo suficiente para realizar os testes.

Um dos participantes do *survey* confirmatório acrescentou que a configuração e escrita dos *scripts* seria uma etapa inevitável, que não inviabilizaria a prática no cotidiano. Logo, acreditamos que o problema da automação está além destes fatores.

O principal fator que influencia a falta de testes automáticos pode ser a falta de conhecimento dos desenvolvedores acerca das ferramentas e técnicas de testes disponíveis. No *survey* confirmatório mais de 57% dos respondentes afirmou conhecer poucas ferramentas de testes no contexto *mobile* e mais de 40% não soube se posicionar se faltam mais ferramentas específicas para testes.

Conforme observado anteriormente, na revisão bibliográfica deste trabalho e descrito no capítulo 4 desta dissertação, em um repositório de aplicações móveis *Open Source*, o *F-Droid*, onde encontramos aplicações como *K-9 Mail*, *ChatSecureAndroid* e *Writeily-pro*, entre outras, que possuem poucos ou nenhum arquivo de testes automatizados.

Esta observação é confirmada em uma das seções do *survey* confirmatório em que os participantes informaram que, em suas equipes, eram realizados poucos testes ou que a cobrança por testes era sempre muito baixa para projetos *mobile*.

O uso de testes manuais dificulta o teste de regressão e aumenta o custo do projeto ao tornar necessário que o testador repita várias vezes os mesmos métodos, utilizando para isso

tempo que poderia ser direcionado para outras necessidades do processo de V&V. Além disto, não há registro da atividade do testador, não é possível identificar qual a cobertura exata da atividade de testes e, caso o próprio desenvolvedor seja o responsável por testar as funcionalidades, o impacto das alterações implementadas em versões anteriores do código pode não ser detectada.

6.2 RQ2: *Quais os tipos de testes automáticos mais utilizados pela comunidade?*

Os dados coletados durante os *surveys* exploratório, confirmatório e as entrevistas entram em acordo quanto aos tipos de teste utilizados.

Quando há automação, a preferência dos desenvolvedores é por testes funcionais, utilizando ferramentas como o *Monkey* e testes de interface, utilizando ferramentas como o *Robotium* e o *Selenium*. Os testes estruturais foram indicados no *survey* confirmatório, mas ferramentas de automação para este tipo de teste não foram sugeridas.

Além disso, há falta de planejamento e execução de testes em que são considerados cenários que utilizam sensibilidade ao contexto. Em sistemas que são executados em um ambiente cujos sensores são capazes de monitorar o ambiente a que estão expostos e onde há diversas características de conectividade e preocupações com autonomia podem surgir problemas relacionados à segurança, privacidade e recursos limitados que não devem ser desconsiderados.

Ainda assim, no *survey* confirmatório, a afirmação *Os testes realizados pela minha equipe consideram o uso de sensores (gps, temperatura, localização, etc)* obteve mais de 70% de discordância pelos participantes. Entretanto, um dos participantes das entrevistas informou que, mesmo que as aplicações móveis desenvolvidas não acessassem diretamente os recursos disponibilizados pelo dispositivo, havia um roteiro de testes que precisava ser executado para garantir que não haveria interferências no desempenho dos *apps* desenvolvidos por sua empresa.

6.3 RQ3: Qual o grau de satisfação dos participantes com as atuais práticas de teste?

A satisfação dos participantes com as práticas de testes de *software* em suas equipes foi outro aspecto analisado em todas as etapas do estudo.

Entretanto, a satisfação ou não dos desenvolvedores não parece estar conectada a apenas um fator. No *survey* confirmatório, ao serem diretamente questionados acerca do seu grau de satisfação, não foi possível obter suporte estatístico para afirmar que os respondentes não estão satisfeitos com a prática de testes utilizada em suas equipes.

Mas, em um campo específico do questionário permitido ao participante expor o motivo de sua satisfação ou insatisfação, um respondente, apesar de ter respondido estar satisfeito anteriormente, relatou:

"A equipe na qual eu trabalhei utilizava bastantes testes manuais, e principalmente voltados a interface das aplicações. Alguns tipos de testes não eram realizados, tanto por falta de experiência de parte da equipe, como também por falta de tempo e de pessoal(e.g. testes A-B, testes de regressão, testes de integração etc)"[Participante 07]

A automação dos testes, já discutida anteriormente, é um dos fatores que influencia a satisfação dos desenvolvedores. No *survey* exploratório, os participantes informaram, em sua maioria, que gostariam de um grau maior de automação dos testes. Um dos participantes desta etapa fez um comentário semelhante:

Testes manuais dificultam a identificação de regressões no código: tarefas já testadas anteriormente precisam ser verificadas manualmente e as várias repetições fazem as pessoas relevarem pontos que podem ser importantes. Uma suíte de testes automatizados resolveria esse problema. [Participante 02]

Outros dois fatores observados foram existência ou não de uma equipe específica para testar as aplicações e a falta de conhecimento dos desenvolvedores acerca dos testes, já discutida anteriormente.

Ao serem questionados sobre a confiança na qualidade do *software* produzido quando existe uma equipe de testes separada, mais de 87% os respondentes do *survey* confirmatório concordaram, e um dos participantes comentou:

Já trabalhei com a própria equipe realizando os testes e atualmente temos um time separado, a diferença é perceptível pois como a equipe externa não participa no desenvolvimento ela acaba notando mais coisas que os desenvolvedores não notam por estar lidando com o aplicativo todos os dias. [Participante 36]

Sobre a falta de conhecimento dos desenvolvedores, durante as entrevistas foi observado que não há conhecimento disseminado na comunidade sobre métricas e técnicas de teste adequadas ao contexto *mobile*, e, portanto, não há parâmetro específico para guiar o interesse do profissional sobre as práticas da equipe.

Um dos participantes citou a falta de uma 'cultura de testes' como sendo um dos desafios encontrados para garantir a qualidade do *software* produzido. Esta cultura de testes pode estar ligada à elaboração de modelos de falhas, referências para os desenvolvedores e a aproximação da literatura científica dos problemas reais enfrentados pelos desenvolvedores.

6.4 RQ4: Como é a relação entre a literatura e a prática de testes?

O *survey* exploratório elucidou a distância existente entre a literatura científica e a prática de testes no contexto de dispositivos móveis.

De maneira geral, durante as entrevistas, esta distância foi detalhada. Alguns dos participantes citaram que a literatura científica era interessante para ilustrar conceitos mais aprofundados da engenharia de *software*, mas para a prática do dia a dia não seria indicada.

Citando alguns dos participantes:

A [literatura] científica geralmente é mais formal, mais dentro do ambiente acadêmico e é sempre uma descoberta nova, para a base de alguma coisa. [Participante 02]

...Já a parte de artigos científicos a gente tem dificuldade de encontrar. São artigos que tem uma qualidade muito boa, mas a gente não tem acesso ao conteúdo por não encontrar. [Participante 16]

Livros e artigos em revistas [literatura comercial] te auxiliam com problemas que você pode encontrar no dia a dia. [Participante 16]

...Eu consigo resolver hoje mais problemas do mercado no mercado do que na literatura acadêmica. [Participante 28]

As ferramentas presentes na automação dos testes desenvolvidas no meio acadêmico parecem também ser desconhecidas dos participantes. No *survey* confirmatório, a seguinte afirmação foi avaliada: *Utilizo ferramentas de teste desenvolvidas e divulgadas na literatura científica*, em uma escala de um (1) a cinco (5), mais de 57% dos respondentes discorda um pouco ou totalmente (2 ou 1) desta afirmação e apenas 14% concorda um pouco (4) com ela.

Cerca de 43% dos participantes concorda um pouco ou totalmente (4 ou 5) da afirmação *Considero a literatura científica pouco acessível fora do ambiente acadêmico, é difícil encontrar algo realmente útil* e apenas 14% discorda um pouco (2) dela.

Além da preferência pela literatura comercial em detrimento da literatura científica, foram destacados o uso de fóruns como o *Stack OverFlow*¹ e fóruns específicos da comunidade de testes em redes sociais. Segundo os participantes, vêm da comunidade de desenvolvimento e testes as maiores contribuições.

6.5 RQ5: Como os desenvolvedores lidam com os desafios encontrados?

O principal desafio encontrado pelos participantes da pesquisa é a fragmentação [44]. A variedade de dispositivos, tamanhos de telas e variação entre as versões do próprio sistema operacional.

As ferramentas indicadas pelos participantes para lidar com este desafio são restrições na documentação do software, os testes em diferentes dispositivos, localmente ou em um

¹Disponível em: <https://stackoverflow.com>

serviço terceirizado, conhecido como *Cloud Testing* ou *Testing as a Service* [43] ou o desenvolvimento de aplicações híbridas [8].

A documentação do software pode limitar as versões testadas do *app* em diferentes dispositivos, indicando a faixa de utilização adequada entre versão do sistema operacional e o hardware necessário.

No *survey* confirmatório, mais de 71% dos respondentes afirma que existe a documentação e elaboração de requisitos em suas equipes, e que costumam atualizá-la. Contudo, apenas 42% afirmam que as limitações das aplicações (como tamanho de tela e versão do S.O.) estão sempre documentadas.

Os serviços de *Cloud Testing* realizam testes funcionais no aplicativo, de acordo com as referências da equipe de desenvolvimento. O serviço conta com diferentes dispositivos físicos e cobra de acordo com a quantidade de testes necessários.

Ambas as abordagens diminuem o custo de compra de aparelhos com diferentes configurações de hardware e possibilitam lidar com a fragmentação.

Outro desafio encontrado é a falta de referências para determinados tipos de falhas em aplicações móveis. Quando questionados sobre as referências utilizadas para gerar cenários de testes, participantes das entrevistas citaram que não tem referências para isso, e que utilizam o conhecimento de colegas e da sua própria experiência ou os testes solicitados pelas lojas de aplicativos para lidar com as falhas.

Uma das ferramentas citadas para lidar com este desafio foi o MTControl [13], uma ferramenta de apoio à geração de casos de teste para aplicações móveis. Baseado em diretrizes da AQuA (App Quality Alliance), a ferramenta solicita informações dos recursos do aparelho utilizados pela aplicação, como acesso à Internet, câmera ou GPS, e cria uma lista de casos de testes a serem considerados pelos desenvolvedores. Embora o MTControl facilite o trabalho de validação, não há ainda a criação dos testes apontados como necessários pela ferramenta ou experimentos que demonstrem sua eficácia e satisfação dos desenvolvedores com os casos de teste sugeridos.

Capítulo 7

Conclusões

Este capítulo apresenta um breve resumo das principais conclusões deste trabalho. Serão apresentados os objetivos as contribuições e sugestões de trabalhos futuros que podem ter início a partir do presente estudo.

O objetivo geral deste estudo é prover uma visualização do quadro das práticas de testes de *software* aplicadas no cotidiano das equipes de desenvolvimento de aplicações móveis (*apps*) no Brasil, utilizando métodos de análise qualitativa.

Inicialmente foi elaborado um *survey* exploratório, enviado para desenvolvedores e testadores presentes nas listas de contato dos pesquisadores. A seguir, o recrutamento de novos participantes seguiu a abordagem *Snowball Sampling*, que se caracterizou pela indicação de novos respondentes por meio de participantes anteriores.

O principal objetivo deste *survey* foi coletar os primeiros dados e reunir voluntários para uma seção de entrevistas semiestruturadas. Estas entrevistas foram gravadas, transcritas e codificadas, de acordo com o método qualitativo de Análise Temática, para, em seguida, serem avaliadas por profissionais de desenvolvimento de *apps* em um *survey* confirmatório.

Este último *survey* foi elaborado a partir das informações coletadas nas duas etapas anteriores e trazia uma série de afirmativas que deveriam ser avaliadas pelos participantes em uma escala *Likert*. Por fim, os dados coletados foram analisados para responder as perguntas de pesquisa do estudo.

Os resultados desta pesquisa demonstram que o conhecimento dos desenvolvedores sobre os testes adequados para serem utilizados no contexto *mobile* ainda é incipiente e necessita não apenas de tecnologia aplicada ao processo, mas de uma cultura de testes que enfatize as

características próprias dos dispositivos e vincule soluções adequadas ao processo de desenvolvimento, incluindo a automação dos testes.

A preocupação deste estudo em analisar a relação dos desenvolvedores com a literatura também evidenciou a preferência da utilização de fóruns e a literatura comercial como alternativa à literatura científica assim como também os principais motivos de insatisfação dos desenvolvedores com a qualidade dos *softwares* construídos.

7.1 Contribuições

A literatura possui trabalhos como o de Joorabchi *et al.* [25], Gao *et al.* [17], entre outros, que realizam estudos sobre a natureza do desenvolvimento de aplicações móveis e os testes de *software* utilizados para melhorar a qualidade dos *apps* construídos. No entanto, o presente trabalho enfatiza a prática de testes de *software* do cotidiano das equipes. Este tipo de trabalho não foi encontrado na revisão bibliográfica desde o princípio deste estudo.

O aspecto de estudo qualitativo também difere da maioria dos trabalhos e traz contribuições para que pesquisadores e estudantes utilizem técnicas similares que são cada vez mais populares na Ciência da Computação. Conforme afirma Parra *et al.* [35], os dados qualitativos são capazes de descrever cenários com maior riqueza de informações e resultar em uma análise mais profunda da natureza dos problemas encontrados.

O quadro de desenvolvimento de aplicações móveis e sua influência nas relações das empresas e consumidores é expressivo e necessita da atenção da academia. Este trabalho produz um fundamental teórico para que pesquisadores vislumbrem possibilidades de pesquisa e que sejam desenvolvidos trabalhos que elaborem soluções baseadas na prática dos desenvolvedores, ampliando as relações entre academia e indústria.

7.2 Trabalhos Futuros

Com os resultados deste trabalho, é possível a visualização da atividade de práticas de testes de software aplicados ao contexto móvel, os tipos de testes utilizados, as ferramentas de automação e alguns dos desafios encontrados pelos desenvolvedores.

A partir deste quadro, o planejamento de alguns trabalhos futuros, como os citados a

seguir, pode ser facilitado.

Manual de práticas de teste: Nosso estudo foi capaz de verificar os tipos de testes e algumas das ferramentas preferidas pelos desenvolvedores e testadores. Entretanto, não foi possível elaborar um manual de práticas, focado no processo de desenvolvimento de software *mobile*. Tal contribuição pode auxiliar a construção de uma cultura de testes incorporada ao processo.

Modelo de Falhas: Um dos problemas citados por participantes das entrevistas é que não há modelos para tipos de falhas comuns nos *apps*. Modelos que relacionem os recursos utilizados com o tipo de teste associado. Este trabalho pode ser uma boa contribuição e pode estar associado ao manual citado anteriormente

Ferramentas de Testes: Como apresentado durante o capítulo dos resultados, muitos participantes desconhecem ferramentas de teste automáticos, principalmente as que são desenvolvidas e divulgadas na literatura científica. A criação ou melhoria de algumas das ferramentas de testes automáticos para atender às reais necessidades dos desenvolvedores podem ser elaboradas a partir dos resultados deste estudo.

A elaboração de manuais, modelos de referência e criação ou melhorias nas ferramentas de automação de testes existentes podem ser um fator de grande importância para a aproximação da indústria e a literatura científica.

Bibliografia

- [1] ITU releases latest global technology development figures, 2013.
- [2] Stack Overflow Developer Survey 2017, 2017.
- [3] Steve Adolph, Wendy Hall, and Philippe Kruchten. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4):487–513, 8 2011.
- [4] Domenico Amal, Anna Rita Fasolino, Por Rio Tramontana, Bryan Dzung Ta, and Atif M Memon. MobiGUITAR Automated Model-Based Testing of Mobile Apps. *Software, IEEE*, 32(5):53 – 59, 2014.
- [5] Domenico Amalfitano, Anna Rita Fasolino, Porfirio Tramontana, Bryan Dzung Ta, and Atif M Memon. MobiGUITAR: Automated Model-Based Testing of Mobile Apps. *IEEE Software*, 32(5):53–59, 9 2015.
- [6] Esteban Angulo and Xavier Ferre. A Case Study on Cross-Platform Development Frameworks for Mobile Applications and UX. *Proceedings of the XV International Conference on Human Computer Interaction*, 2014.
- [7] V Braun and V Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.
- [8] Salma Charkaoui, Zakaria Adraoui, and El Habib Benlahmar. Cross-platform mobile development approaches. *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, pages 188–191, 2014.
- [9] Shauvik Roy Choudhary, Alessandra Gorla, and Alessandro Orso. Automated Test Input Generation for Android: Are We There Yet? *CoRR*, 2015.

-
- [10] Gerry Coleman and Rory O 'connor. Investigating software process in practice: A grounded theory perspective. 2007.
- [11] James Coleman. Relational Analysis: The Study of Social Organizations with Survey Methods. *Human Organization*, 17(4):28–36, 12 1958.
- [12] Eliane Collins, Arilo Dias-Neto, and Vicente F. de Lucena Jr. Strategies for Agile Software Testing Automation: An Industrial Experience. In *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, pages 440–445. IEEE, 7 2012.
- [13] Juliana P Do Nascimento, Jonathas S Dos Santos, and Arilo C Dias-Neto. MTControl: Ferramenta de Apoio à Gestão de Testes de Aplicativos Móveis Baseada nas Diretrizes do AQuA. pages 200–69077, 2015.
- [14] John Dudovskiy. *The Ultimate Guide to Writing a Dissertation in Business Studies: A step by-step assistance*. 2016.
- [15] Clément Escoffier and Philippe Lalanda. Managing the Heterogeneity and Dynamism in Hybrid Mobile Applications. 2015.
- [16] Jerry Gao, Xiaoying Bai, Wei-Tek Tsai, and Tadahiro Uehara. Mobile Application Testing: A Tutorial. *Computer*, 47:46–55, 2014.
- [17] Jerry Gao, Wei-Tek Tsai, Ray Paul, Xiaoying Bai, and Tadahiro Uehara. Mobile Testing-as-a-Service (MTaaS) – Infrastructures, Issues, Solutions and Needs. *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, pages 158–167, 2014.
- [18] B.G. Glaser and A.L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, 1967.
- [19] Leo A. Goodman. Snowball Sampling. *The Annals of Mathematical Statistics*, 1961.
- [20] Matthew L. Hale and Seth Hanson. A Testbed and Process for Analyzing Attack Vectors and Vulnerabilities in Hybrid Mobile Apps Connected to Restful Web Services. *2015 IEEE World Congress on Services*, pages 181–188, 2015.

-
- [21] Klaus Haller. Mobile Testing. *ACM SIGSOFT Software Engineering Notes*, 38(6):1–8, 2013.
- [22] Henning Heitkötter, Sebastian Hanschke, and Tim a Majchrzak. Comparing Cross-Platform Development Approaches for Mobile Applications. *Web Information Systems and Technologies*, 140:120–138, 2013.
- [23] Gang Hu, Xinhao Yuan, Yang Tang, and Junfeng Yang. Efficiently, effectively detecting mobile app bugs with AppDoctor. *Proceedings of the Ninth European Conference on Computer Systems - EuroSys '14*, pages 1–15, 2014.
- [24] Vojtěch Jína. JavaScript Test Runner. 2013.
- [25] Mona Erfani Joorabchi, Ali Mesbah, and Philippe Kruchten. Real Challenges in Mobile App Development. *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 15–24, 2013.
- [26] Paul. Jorgensen. *Software testing : a craftsman's approach*. CRC Press, 2002.
- [27] B. Kirubakaran and V. Karthikeyani. Mobile application testing — Challenges and solution approach through automation. *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pages 79–84, 2013.
- [28] André Koscianski, Michel Dos, Santos Soares, and Segunda Edição. Qualidade de Software Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software.
- [29] José Carlos Maldonado, Ellen Francine, Barbosa Auri, Marcelo Rizzo Vincenzi, Márcio Eduardo Delamaro, and Mario Jino. Introdução ao Teste de Software *.
- [30] Emiliano Masi, Giovanni Cantone, Manuel Mastrofini, Giuseppe Calavaro, and Paolo Subiacco. Mobile Apps Development: A Framework for Technology Decision Making. pages 64–79. *Proceedings of International Conference on Mobile Computing, Applications, and Services*, 2012.

- [31] Izabela Melo, Gustavo Santos, Dalton Dario Serey, and Marco Tulio Valente. Perceptions of 395 Developers on Software Architecture's Documentation and Conformance. *SBCARS*, 2016.
- [32] Henry Muccini, Antonio di Francesco, and Patrizio Esposito. Software testing of mobile applications: Challenges and future research directions. *7th International Workshop on Automation of Software Test (AST 2012)*, pages 29–35, 2012.
- [33] Glenford J Myers. *The art of software testing*. John Wiley & Sons, Inc., 2nd edition, 2004.
- [34] João Osvaldo Dewes Orientadora and Professora Dra Luciana Neves Nunes. Amostragem em Bola de Neve e Respondent-Driven Sampling: uma descrição dos métodos.
- [35] Amalia Parra, Victor Basili, and Steven Burke. The Package-Based Development Process in the Flight Dynamics Division. 1998.
- [36] Gian Pietro Picco, Christine Julien, Amy L. Murphy, Mirco Musolesi, and Gruia-Catalin Roman. Software engineering for mobility: reflecting on the past, peering into the future. *Proceedings of the on Future of Software Engineering - FOSE 2014*, pages 13–28, 2014.
- [37] Youssef Ridene, Franck Barbier, Nicolas Belloir, and Nadine Couture. A DSML for Mobile Phone Applications Testing. *Proceedings of the 10th . . .*, pages 1–6, 2010.
- [38] Gruia-Catalin Roman, Gian Pietro Picco, and Amy L Murphy. Software Engineering for Mobility: A Roadmap. 2000.
- [39] M Satyanarayanan. Fundamental Challenges in Mobile Computing.
- [40] Forrest. Shull, Janice. Singer, and Dag I. K. Sjøberg. *Guide to advanced empirical software engineering*. Springer, 2008.
- [41] Leif Singer, Fernando Figueira Filho, and Margaret-Anne Storey. Software Engineering at the Speed of Light: How Developers Stay Current Using Twitter. 2014.
- [42] Ian Sommerville. *Software engineering*. Pearson, 2011.

-
- [43] Isabel Karina Villanes, Erick Alexandre Bezerra Costa, and Arilo Claudio Dias-Neto. Automated Mobile Testing as a Service (AM-TaaS). In *2015 IEEE World Congress on Services*, pages 79–86, 2015.
- [44] Spyros Xanthopoulos and Stelios Xinogalos. A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications. *Proceedings of the 6th Balkan Conference in Informatics*, pages 213–220, 2013.

Apêndice A

Roteiro do *Survey* Exploratório

IDENTIFICAÇÃO DO PARTICIPANTE

1. Qual é o seu nome? [Opcional]
2. Identifique a sua instituição de trabalho
 - (a) Universidade (Qual?)
 - (b) Empresa (Qual?)
 - (c) Desenvolvedor Independente
 - (d) Outra (Citar)
3. Qual a melhor forma de contactá-lo? (Deixe aqui seu endereço de email ou contato de rede social, etc) [Opcional]

CARACTERÍSTICAS PROFISSIONAIS DO PARTICIPANTE

1. Há quanto tempo você trabalha na área da Computação?
2. Você tem alguma experiência em testes de software para dispositivos móveis?
 - (a) Sim
 - i. Fale um pouco sobre a sua experiência, quanto tempo trabalhou, onde trabalhou, quais plataformas tem mais familiaridade e se você acredita que os resultados do seu trabalho melhoraram a qualidade dos produtos.

-
- (b) Não
3. Quais papéis você já desempenhou em projetos na área da computação?
- (a) Desenvolvedor
 - (b) Testador
 - (c) Analista de Sistemas
 - (d) Outra (Citar)

TESTE DE APPS

1. Para quais plataformas móveis você já desenvolveu?
- (a) Android
 - (b) iOS
 - (c) Windows Phone
 - (d) Desenvolvimento Híbrido com HTML5
 - (e) Outro (citar)
2. Na sua perspectiva, como você descreve a importância do teste de software aplicado aos apps? (Sugestões)
3. Você costuma automatizar os testes?
- (a) Sim
 - i. Que tecnologias você utiliza para automatização? Cite algumas ferramentas.
 - (b) Não
 - i. Você conhece alguma ferramenta para automatização?
 - ii. Qual o motivo de não automatizar?
4. Que tipo de teste automático você considera mais eficaz para detecção de falhas nos apps?
- (a) Apenas Testes Manuais

- (b) Testes Unitários
 - (c) Testes de performance
 - (d) Testes de Interface
 - (e) Testes de usabilidade
 - (f) Outro (Citar)
5. Em que tipo de literatura você costuma pesquisar para definir os testes a serem utilizados no projeto?
- (a) Científica
 - (b) Comercial (Sites, blogs, revistas)
 - (c) Política da empresa (Manuais, guias, etc)
 - (d) Nenhuma das opções acima (Citar)
6. No caso da sua instituição, existe equipe separada para testes?
- (a) Sim
 - (b) Não
7. Em que nível de automação você acredita que estejam os testes desenvolvidos pela sua equipe? Sendo 1 (um) apenas testes manuais e 5 (cinco) apenas testes automáticos.
- (a) 1
 - (b) 2
 - (c) 3
 - (d) 4
 - (e) 5
8. Em que nível de automação você gostaria que estivessem os testes desenvolvidos pela sua equipe? Sendo 0 (zero) apenas testes manuais e 5 (cinco) apenas testes automáticos.
- (a) 1

- (b) 2
- (c) 3
- (d) 4
- (e) 5

CONTATO E AGRADECIMENTO

1. Suas respostas são muito importantes para nós. Agradecemos sua disponibilidade. Nossa pesquisa é qualitativa e segue uma metodologia conhecida como “Análise Temática”, na qual utilizamos dados coletados por meio de surveys e entrevistas para obter nossos resultados. Precisamos de uma grande quantidade de participantes para que nossos resultados sejam validados.
2. Você se disponibiliza para uma entrevista?
 - (a) Sim
 - i. Deixe-nos um meio de contato
 - (b) Não
3. Você indica alguém para responder a este questionário?
 - (a) Sim
 - i. Deixe-nos um meio de contato para esta pessoa.
 - (b) Não

Apêndice B

Roteiro das Entrevistas

IDENTIFICAÇÃO DO PARTICIPANTE

1. Número do respondente para identificação da linha no banco de dados

EXPERIÊNCIA DO PARTICIPANTE

1. Fale um pouco sobre a sua experiência na computação e o desenvolvimento de aplicações móveis.
2. Para quais plataformas desenvolveu? Incluindo aplicações “Não- mobile”.
3. Qual o tempo de experiência para cada plataforma?
4. Outra (Citar)
5. Em quais plataformas atuou como testador?
6. Quais as principais diferenças que você encontrou ao realizar testes no contexto móvel em relação às aplicações desktop ou web? (Pergunta para quem já desenvolveu para Desktop).
7. Existem diferenças com relação à documentação e elaboração de requisitos?

TESTE DE APPS

1. As práticas de teste em sua equipe incluem/incluam um plano de testes?

-
- (a) Sim
 - i. Descreva a construção do plano. É/Era elaborado antes ou depois da etapa de desenvolvimento?
 - ii. É/Era comum a necessidade de mudança nos planos de teste?
 - iii. Que tipos de testes estão/estavam presentes no plano?
 - (b) Não
 - i. Como se dá/dava o planejamento e execução de testes em seus projetos?
2. Que níveis de teste você considera que são mais abordados no contexto móvel? Existem níveis de teste que não são considerados neste contexto? Considerando:
- (a) Testes de Unidade
 - (b) Testes de Integração
 - (c) Testes de Sistema
 - (d) Testes de Aceitação
 - (e) Testes de Usabilidade
3. Na perspectiva de testador, como garantir o correto funcionamento de aplicações que funcionam em diversos tipos de dispositivos e variações de sistemas operacionais?
4. Os aparelhos móveis possuem características próprias de conectividade, segurança, portabilidade, sensibilidade ao contexto e autonomia. Cenários que envolvem estas características são frequentemente abordados pelos testadores?
- (a) Sim
 - i. Quais são as referências utilizadas para gerar os cenários?
 - ii. Que abordagens de teste são utilizadas para cada cenário?
 - iii. Existem ferramentas específicas para estes casos?
 - (b) Não
 - i. Você considera quais os motivos? Faltam referências, ferramentas, conhecimento mais aprofundado do ambiente?

-
5. Você considera que alguma das plataformas existentes (Móveis) possui maior suporte das ferramentas de teste?
6. Você costuma automatizar os testes?
- (a) Sim
 - i. Existem tipos de teste mais fáceis de automatizar que outros?
 - ii. Que tipos de teste não são possíveis de automatizar?
 - iii. Você conhece alguma ferramenta para automação?
 - (b) Não
 - i. Qual o motivo de não automatizar?
7. Existe alguma ferramenta ou técnica de teste que você recomendaria fazer parte deste estudo?
8. Existe alguma ferramenta ou técnica de teste que você utilizou, mas não ficou satisfeito com os resultados?
- (a) Sim
 - i. Quais foram estas tecnologias ou técnicas?
 - ii. Qual(is) o(s) motivo(s) da insatisfação?
 - (b) Não
 - i. Continue
9. Você considera que as ferramentas de teste atuais cumprem bem o seu papel ou faltam ferramentas mais específicas?
10. Descreva desde o princípio o processo de testes que é utilizado para garantir a qualidade dos softwares desenvolvido pela sua equipe.
11. Sobre o tipo de literatura que você afirmou consultar no questionário (Ler). Você costuma encontrar as referências necessárias para aplicar em seu trabalho?
- (a) Sim

- i. Na sua opinião, porque existe essa dificuldade?
 - ii. O que é feito quando a referência não é encontrada?
 - (b) Não
 - i. Continue

- 12. Você costuma visitar fóruns específicos para desenvolvimento ou testes em aplicações móveis?
 - (a) Sim
 - i. Poderia citar alguns desses fóruns?
 - (b) Não
 - i. Continue

- 13. Na sua experiência, quais os desafios atuais para garantir a qualidade do software no contexto móvel?

Apêndice C

Termo de Consentimento da Entrevista

INFORMAÇÕES SOBRE A PESQUISA:

Título do Projeto: Estudo da Prática de Testes no Desenvolvimento de Aplicações Móveis

Orientadores Responsáveis: Prof. Dr. Wilkerson Andrade e Prof. Dr. Dalton Serey

Pesquisador Responsável: Achilles Luciano

Telefones para contato: (83) 99655-3632

E-mail para contato: achiles@copin.ufcg.edu.br

OBJETIVOS DA PESQUISA:

Esta pesquisa pretende analisar as práticas de testes em aplicações móveis Android na comunidade de desenvolvedores e disponibilizar os resultados juntamente com a elaboração de um documento de referência para a comunidade.

Sua participação envolve uma entrevista, que será gravada se assim você permitir, e que tem a duração aproximada de 20 minutos.

A participação nesse estudo é voluntária e se você decidir não participar ou quiser desistir de continuar em qualquer momento, tem absoluta liberdade de fazê-lo.

Na publicação dos resultados desta pesquisa, sua identidade será mantida no mais rigoroso sigilo. Serão omitidas todas as informações que permitam identificá-lo(a).

TERMO DE CONSENTIMENTO:

Se você concordar voluntariamente em participar do estudo acima descrito, como sujeito. Torna-se necessário a resposta a este e-mail copiando o trecho a seguir:

“Declaro ter sido devidamente informado e esclarecido pelo pesquisador Achilles Luciano sobre o objetivo da pesquisa, os procedimentos nela envolvidos, assim como a garantia do sigilo de informações pessoais e profissionais envolvidos na minha participação. Me foi fornecida a oportunidade de fazer perguntas e recebi telefone e endereço de e-mail para entrar em contato, caso tenha dúvidas. Foi-me garantido que não sou obrigado a participar.”

Apêndice D

Roteiro do *Survey* Confirmatório

FORMAÇÃO DO PARTICIPANTE

1. Qual sua formação acadêmica?
 - (a) Graduação em Andamento
 - (b) Graduação
 - (c) Pós-graduação (lato sensu) em Andamento
 - (d) Pós-graduação (lato sensu)
 - (e) Mestrado em Andamento
 - (f) Mestrado
 - (g) Doutorado em Andamento
 - (h) Doutorado

DADOS PROFISSIONAIS

1. Como você descreve sua atual vinculação profissional?
 - (a) Funcionário de empresa – setor público
 - (b) Funcionário de empresa – setor privado
 - (c) Profissional bolsista atuando em Convênio Empresa – Universidade
 - (d) Estudante bolsista atuando em Convênio Empresa – Universidade

- (e) Desenvolvedor Autônomo
 - (f) Outro
2. Por quanto tempo você trabalhou em projeto(s) de desenvolvimento de software mobile?
- (a) Menos de 6 meses
 - (b) Seis meses a 1 ano
 - (c) 1 a 2 anos
 - (d) 2 a 3 anos
 - (e) 3 a 4 anos
 - (f) Mais de 4 anos
3. Durante este(s) projeto(s), você atuou principalmente como:
- (a) Desenvolvedor de Software
 - (b) Analista de Sistemas
 - (c) Testador
 - (d) Gerente de Projeto
 - (e) Outro
4. Qual a área que considera ser a sua especialidade?
- (a) Desenvolvimento de Software – Web
 - (b) Desenvolvimento de Software – Mobile
 - (c) Desenvolvimento de Software – Desktop
 - (d) Qualidade de Software
 - (e) Gerência de Projetos
 - (f) Manutenção de servidores
 - (g) Pesquisa Científica
 - (h) Outra

-
5. A instituição em que esteve vinculado(a) durante o(s) projeto(s) de desenvolvimento mobile mantém/mantinha equipes separadas de desenvolvimento e testes?
- (a) Sim
 - (b) Não
- 6.
7. Considerando a quantidade de desenvolvedores contratados, esteve vinculado(a) durante o(s) projeto(s) de desenvolvimento mobile pode ser considerada:
- (a) Grande (Mais de 80 funcionários)
 - (b) Média (Mais de 40 funcionários)
 - (c) Pequena (Até 40 funcionários)
8. Sobre a afirmação: “Quando há uma equipe separada especificamente para testes, a confiança que tenho na qualidade do software em que estou trabalhando aumenta”. Qual seu grau de concordância, de 0 a 5, sendo 0 – “Discordo totalmente” e 5 – “Concordo totalmente”.
9. Se desejar, explique ou comente a resposta anterior. [opcional]

TIPOS DE TESTE UTILIZADOS

10. Na sua experiência, que tipos de testes são mais utilizados para garantir a qualidade das aplicações móveis?
- (a) Testes de Performance (Estresse)
 - (b) Testes funcionais (apenas executando a aplicação e analisando seu comportamento)
 - (c) Testes Estruturais (Análise do código, cobertura dos testes)
 - (d) Não tenho conhecimento
 - (e) Outro
11. Sobre as frases abaixo, informe o seu grau de concordância, de 1 a 5, sendo 1 – “Discordo totalmente” e 5 – “Concordo totalmente”:

-
- (a) “Na minha equipe, a maior parte da execução dos testes se dá/dava manualmente”
 - (b) “Utilizo ferramentas de execução automática de testes de performance (Monkey, por exemplo)”
 - (c) “Acredito que faltam ferramentas mais específicas para automação de testes no contexto móvel”
 - (d) “Conheço poucas ferramentas de testes para aplicações móveis”
 - (e) “Nos projetos de desenvolvimento mobile em que trabalhei, os níveis de testes mais automatizados são os de unidade, integração e sistema”
 - (f) Outro
12. Se desejar, comente alguma das frases da questão anterior. [opcional]
13. Sobre a afirmação: “A automação diminui o tempo de execução dos testes e facilita o desenvolvimento de software de qualidade. Mas o tempo gasto com configuração e escrita dos scripts inviabiliza a prática”. Qual seu grau de concordância, de 0 a 5, sendo 0 – “Discordo totalmente” e 5 – “Concordo totalmente”.
14. Qual o seu grau de satisfação com as práticas de teste desenvolvidas pela sua equipe? (0 a 5) Sendo 0 – Totalmente Insatisfeito e 5 – Totalmente satisfeito.

LITERATURA

1. Sobre as frases abaixo, informe o seu grau de concordância, de 0 a 5, sendo 0 – “Discordo totalmente” e 5 – “Concordo totalmente”:
- (a) “Costumo pesquisar na literatura científica com menor frequência porque consigo resolver a maioria dos problemas que encontro pesquisando em fóruns na comunidade de desenvolvedores” “Consigo encontrar mais respostas para as minhas necessidades na literatura comercial (Blogs, revistas, sites, livros, etc.) do que em fóruns específicos”
 - (b) “Considero a literatura científica pouco acessível fora do ambiente acadêmico, é difícil encontrar algo realmente útil.” “Utilizo ferramentas de teste desenvolvidas e divulgadas na literatura científica”

- (c) “O meu conhecimento e o auxílio dos meus colegas de trabalho geralmente é o suficiente para resolver os problemas cotidianos”
- (d) “Consulto a literatura científica com frequência, porque encontro a fundamentação necessária para compreender os problemas que vivencio no meu trabalho.”

DOCUMENTAÇÃO

1. Sobre as frases abaixo, informe o seu grau de concordância, de 0 a 5, sendo 0 – “Discordo totalmente” e 5 – “Concordo totalmente”:
 - (a) “Nos projetos em que trabalhei não há documentação ou elaboração de requisitos ou é escassa”
 - (b) “Nos projetos em que trabalhei existe um planejamento dos testes”
 - (c) “Eu não costumo documentar ou atualizar a documentação.”
 - (d) “Os testes realizados pela minha equipe consideram o uso de sensores (gps, temperatura, localização, etc).”
 - (e) “As limitações de tamanho de tela e versão do sistema operacional (Android) das aplicações desenvolvidas pela minha equipe sempre estão documentadas”
 - (f) “consulto um modelo de falhas que relaciona o recurso que utilizo do dispositivo com o tipo de teste adequado”

FRAGMENTAÇÃO

1. Sobre as frases abaixo, informe o seu grau de concordância, de 0 a 5, sendo 0 – “Discordo totalmente” e 5 – “Concordo totalmente”:
 - (a) “Utilizo diferentes dispositivos físicos para testar as aplicações desenvolvidas pela minha equipe em cada projeto”
 - (b) “Utilizo apenas meus dispositivos pessoais e pessoais da minha equipe para testar as aplicações desenvolvidas.”
 - (c) “Utilizo um bom emulador virtual para simular os ambientes necessários.”
 - (d) “Utilizo um serviço online que realiza os testes da minha aplicação em diferentes dispositivos físicos (Cloud Testing)”