



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS GUSTAVO RESQUE DOS SANTOS

**Descoberta, Navegação e Exploração de Pontos de Interesse
com Dispositivos Móveis através de Múltiplas Visões**

Belém
2015

Carlos Gustavo Resque dos Santos

**Descoberta, Navegação e Exploração de Pontos de Interesse
com Dispositivos Móveis através de Múltiplas Visões**

Dissertação de Mestrado apresentada para obtenção do grau de Mestre em Ciência da Computação. Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará.

Orientador Prof. Dr. Bianchi Serique Meiguins

Belém
2015

Carlos Gustavo Resque dos Santos

**Descoberta, Navegação e Exploração de Pontos de Interesse
com Dispositivos Móveis através de Múltiplas Visões**

Dissertação de Mestrado apresentada para obtenção do grau de Mestre em Ciência da Computação. Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará.

Data da defesa: Belém, 3 de julho de 2015

Conceito: _____

Prof. Dr. Bianchi Serique Meiguins – Orientador
Programa de Pós-Graduação em Ciência da Computação – UFPA

Prof. Dr. Nelson Cruz Sampaio Neto – Membro Interno
Programa de Pós-Graduação em Ciência da Computação – UFPA

Prof^a. Dr^a. Marcelle Pereira Mota – Membro Externo
Faculdade de Computação – ICEN/UFPA

Prof. Dr. Mario Massakuni Kubo – Membro Externo
INDRA - POLITEC – Brasília, DF

Dedico aos meus pais, Ana Sofia Resque Gonçalves e Carlos Augusto dos Santos, que sempre me deram apoio incondicional.

AGRADECIMENTOS

Agradeço a meu orientador Prof. Dr Bianchi Serique Meiguins por me aceitar como orientando e por me fornecer ensinamentos e orientações que me iluminaram durante o mestrado e que levarei para minha vida.

Agradeço à equipe do Laboratório de Visualização da Informação (LABVIS) da UFPA – em especial a Nikolas Carneiro, a Brunelli Miranda e a Tiago Araújo – por todas as atividades realizadas em equipe que me ajudaram a concluir este trabalho.

Agradeço à banca avaliadora formada pelos professores Nelson Cruz Sampaio Neto, Marcelle Pereira Mota, e Mario Massakuni Kubo por contribuições dadas a este trabalho.

Agradeço a meus pais Carlos Augusto dos Santos e Ana Sofia Resque Gonçalves, a minha noiva Aline da Silva Barbosa e a meu irmão Carlos Roosevelt Resque dos Santos o apoio, o carinho e a dedicação que me impulsionam a cada dia.

Agradeço a meus amigos e colegas de mestrado Manuel Afonso, Thiago Syllas e Reginaldo Filho a dedicação dada aos trabalhos em grupo nas disciplinas do mestrado.

Agradeço a todos que pacientemente realizaram os testes presentes neste trabalho e a todos que participaram indiretamente.

“Ser é ser percebido” (George Berkeley)

RESUMO

O avanço tecnológico dos dispositivos móveis permitiu a entrada de tecnologias inovadoras no cotidiano dos usuários, através de sensores, câmeras e recursos computacionais cada vez mais robustos e miniaturizados. Essas tecnologias possibilitam ampliar a interação e percepção do usuário sobre lugares e objetos a sua volta, possibilitando, por exemplo, o uso da realidade aumentada. Entretanto, essa oportunidade apresenta novos desafios, tais como: aspectos de desenvolvimento de aplicações para esse novo contexto ubíquo dos dispositivos, heterogeneidade dos dispositivos móveis, e como o usuário interage com essas aplicações. Assim, o objetivo principal deste trabalho é apresentar uma proposta para descoberta, navegação e exploração de pontos de interesse através de múltiplas visões com dispositivos móveis. Pontos de Interesse são lugares ou objetos que o usuário pode conhecer e interagir, com informações além das disponíveis localmente. Para atingir tal objetivo, foi proposta uma arquitetura flexível para os dados, extensível para adição ou remoção de funcionalidades, e adaptável aos diferentes tamanhos de tela, sendo que a arquitetura proposta é uma adaptação o padrão arquitetural MVC. Também foi desenvolvida uma aplicação que utiliza a arquitetura proposta e possui as funcionalidades de mapa, navegador de realidade aumentada e leitor QR Code, que são visões coordenadas para descoberta, navegação e exploração de pontos de interesse, tanto para ambientes *indoor* quanto *outdoor*. Por fim, foram realizados testes de adequação da aplicação à arquitetura com três fontes de dados diferentes e dois tamanhos de telas diferentes; e testes de usabilidade para avaliar a experiência do usuário no uso da aplicação desenvolvida.

Palavras-chave: Realidade Aumentada Móvel, Arquitetura de Software, Android.

ABSTRACT

The technological advancement of mobile devices allowed the entry of innovative technologies to the daily lives of users, through sensors, cameras and computer resources increasingly robust and miniaturized. These technologies make it possible to increase the user's interaction and perception about places and objects around him, allowing, for example, the use of augmented reality. However, this opportunity presents new challenges, such as application development aspects to this new context of ubiquitous devices, heterogeneity of mobile devices, and how the user interacts with these applications. Thus, the main objective of this paper is to present a proposal for discovery, navigation and exploration of points of interest through multiple views with mobile devices. Points of Interest are places or objects that users can explore and interact with, obtaining new information unavailable locally. To achieve this goal, this work proposes an architecture which is flexible for data, extendable for adding or removing functions, and adaptable to different screen sizes. An application was developed using the proposed architecture and it has the map, augmented reality browser and QR Code reader functions representing coordinated views for discovery, navigation and exploration of points of interest, for both indoor as outdoor environments. Finally, suitability tests were performed on the application architecture using three different data sources and two screens of different sizes; and usability tests were conducted to evaluate the user experience in the use of the application developed.

Key-words: Mobile Augmented Reality, Software Architecture, Android.

LISTA DE ILUSTRAÇÕES

Figura 1. Gráfico da utilização de dispositivos móveis por local de uso. Adaptada e traduzida de comScore (COMSCORE; IMS, 2015).	16
Figura 2. Gráfico da opinião dos usuários sobre as vantagens da utilização de aplicações RAM em relação aos métodos utilizados anteriormente. Legenda: 1-Pior que antes, 2-Não há diferença, 3-Um pouco melhor que antes, 4-Melhor do que antes, 5-Muito melhor que antes, 6-Difícil dizer. Adaptado de Olsson e Salo (2011). ¹ Navegadores RA são aplicações que utilizam predominantemente GPS e sensores e ² Reconhecedores de Padrões – RA são aplicações que utilizam reconhecimento de imagem ou marcadores para inserir o conteúdo virtual. Mais informações na subseção 2.1.	18
Figura 3. Gráfico dos resultados da pesquisa de Liu e Colaboradores que mostram a média de tempo da execução de uma tarefa, por sua dificuldade, utilizando: Realidade Aumentada (AR), Realidade Aumentada + <i>feedback</i> em tempo real (AR+F), Figura e Texto como forma de auxílio ao usuário.	20
Figura 4. Etapas deste trabalho e os métodos utilizados em cada etapa.	23
Figura 5. Contínuo Realidade-Virtualidade adaptado de Milgram e Colaboradores (1994).	27
Figura 6. Cenário típico de um ambiente de Realidade Aumentada.	28
Figura 7. Tipos de <i>display</i> pela sua posição. Adaptado de Bimber e Raskar (2005).	29
Figura 8. Gráfico da quantidade de artigos encontrados nos principais indexadores sobre Arquitetura, Android e Realidade Aumentada, por ano de publicação.	35
Figura 9. Gráfico da quantidade de artigos selecionados após a pesquisa nos principais indexadores, por tipo de aplicação proposta nos artigos.	36
Figura 10. Diagrama de classes da estrutura básica do padrão <i>Observer</i>	37
Figura 11. Diagrama de classes da estrutura básica do padrão <i>Facade</i>	38
Figura 12. Diagrama de classes da estrutura básica do padrão <i>Proxy</i>	38
Figura 13. Estrutura do padrão de arquitetura MVC.	39

Figura 14. Estrutura do padrão de arquitetura MVP.....	40
Figura 15. Ilustração do funcionamento do padrão <i>Fragments</i> na API Android. Imagem adaptada de (GOOGLE INC., 2015).....	41
Figura 16. Esquema da modificação proposta no padrão MVC para adequação ao desenvolvimento de aplicações RAM na plataforma Android.	44
Figura 17. Arquitetura proposta com o fluxo de dados e interações do usuário.....	47
Figura 18. Diagrama de Casos de Uso do projeto da aplicação ARguide.....	48
Figura 19. Diagrama de classes do projeto da aplicação ARguide.....	51
Figura 20. Conteúdo do arquivo <i>func_setup.xml</i> para exemplificar a modificação de funcionalidades no ARguide.....	54
Figura 21. Esquema do funcionamento da configuração de funcionalidades através do arquivo <i>func_setup.xml</i>	55
Figura 22. Trecho do diagrama de classes da aplicação ARguide exemplificando a adição de uma nova classe como procedimento para adicionar uma nova funcionalidade.	56
Figura 23. Tela Inicial da aplicação ARguide com áreas destacadas.	57
Figura 24. Protótipo da aplicação mostrando a Visão Navegador RA em um <i>tablet</i> (a) e em um <i>smartphone</i> (b).....	60
Figura 25. Esquema do funcionamento do radar na funcionalidade Navegador RA.	62
Figura 26. Protótipo da aplicação mostrando a Visão Navegador RA em um <i>tablet</i> (a) e em um <i>smartphone</i> (b).....	63
Figura 27. Aplicação externa realizando a leitura de um código QR Code, para posteriormente enviar ao ARguide.	65
Figura 28. Captura de tela da aplicação visualizando uma mídia de imagem em um <i>smartphone</i>	66
Figura 29. Cenário de uso artístico. Fotografia da utilização da aplicação em um <i>tablet</i> (a) e captura de tela em um <i>smartphone</i> (b).....	68
Figura 30. Cenário de uso educativo. Fotografia da utilização da aplicação em <i>smartphone</i> (a) e <i>tablet</i> (b).	69
Figura 31. Cenário de uso turístico. Captura de tela da aplicação em <i>tablet</i> (a) e <i>smartphone</i> (b).....	71

Figura 32. Perfil dos participantes do teste de usabilidade. (a) Sistema operacional que já utilizou/utiliza, (b) faixa etária, e (c) frequência de uso de dispositivo móvel.	84
Figura 33. Gráficos da média do tempo de realização das tarefas para <i>tablet</i> (a) e <i>smartphone</i> (b).	85
Figura 34. Taxa de acertos, erros e desistências por tarefas realizadas no <i>tablet</i> (a) e <i>smartphone</i> (b).	86
Figura 35. Gráficos sobre: a quantidade de usuários que realizaram padrões de toques por dispositivo, e a quantidade de vezes que os usuários repetiram esse padrão.	88
Figura 36. Arquitetura para o desenvolvimento de aplicações RAM com <i>cloud computing</i> (CHEN, XIANG, <i>et al.</i> , 2013).	115
Figura 37. Arquitetura da aplicação que auxilia os ciclistas a pedalem em grupos (CHEN, PENG e TSENG, 2011).	116
Figura 38. Arquitetura da aplicação que auxilia usuários a configurar e gerenciar sua rede doméstica através de recursos RAM (HAMID, ZHU, <i>et al.</i> , 2011).	116
Figura 39. Arquitetura do aplicativo RAM Animar (LINS, ARRUDA, <i>et al.</i> , 2014).	117
Figura 40. Arquitetura para aplicações RAM utilizarem a biblioteca Vuforia em conjunto com a biblioteca min3d (SOBOTA, KOREČKO e HROZEK, 2013).	117
Figura 41. Arquitetura da aplicação que ajuda os visitantes de um zoológico a encontrarem os animais e obterem mais informações sobre eles. (SRISUPHAB, SILAPACHOTE, <i>et al.</i> , 2014).	118
Figura 42. Arquitetura para aplicações RAM que utilizam sensores e a câmera do dispositivo móvel para desenhar os elementos virtuais (VASSELAI, REIS e GOMES, 2011).	118

LISTA DE TABELAS

Tabela 1. Comparação entre as funcionalidades das aplicações selecionadas.....	32
Tabela 2. Métodos de avaliação e seus fatores de usabilidade. Legenda: P-Pouco, M-Médio, Alto. Adaptado de Mahrin e Colaboradores (2009).....	73
Tabela 3. Princípios de usabilidade adaptados para aplicações RAM utilizadas na avaliação Heurística do ARguide. Tabela adaptada de Perutka (2014).	76
Tabela 4. Resultados da avaliação heurística realizada no <i>smartphone</i>	78
Tabela 5. Resultados da avaliação heurística realizada no <i>tablet</i>	79
Tabela 6. Esquema das fichas de avaliação Percorso Cognitivo utilizadas na avaliação da aplicação ARguide.	80
Tabela 7. Lista de tarefas realizadas pelos participantes da avaliação da aplicação ARguide por medição de desempenho.	83
Tabela 8. Padrões de toques extraídos dos <i>logs</i> registrados pela aplicação durante a realização das tarefas e os IDs de cada padrão de toque.	87
Tabela 9. Falas dos participantes extraídas durante a realização das tarefas quanto estavam interagindo com a função Mapa pelo <i>tablet</i>	90
Tabela 10. Falas dos participantes extraídas durante a realização das tarefas quanto estavam interagindo com a função Mapa pelo <i>smartphone</i>	92
Tabela 11. Comparação das avaliações realizadas e os problemas encontrados.	95
Tabela 12. Resumo dos problemas encontrados e sugestões de melhorias.	96

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
ARguide	<i>Augmented Reality guide</i>
GPS	<i>Global Positioning System</i>
MVC	<i>Model View Controller</i>
MVP	<i>Model View Presenter</i>
POI	<i>Points of Interest</i>
QR Code	<i>Quick Response Code</i>
RA	Realidade Aumentada
RAM	Realidade Aumentada Móvel
RM	Realidade Misturada
RV	Realidade Virtual
SDK	<i>Software Development Kit</i>
UML	<i>Unified Modeling Language</i>
VA	Virtualidade Aumentada

SUMÁRIO

1. INTRODUÇÃO	16
1.1 Justificativa	20
1.2 Objetivos.....	21
1.3 Metodologia.....	22
1.4 Organização do Trabalho	24
2. REFERENCIAL TEÓRICO	26
2.1 Realidade Aumentada	26
2.1.1 Sistemas de Realidade Aumentada.....	27
2.1.2 Realidade Aumentada Móvel	29
2.1.3 Navegação e a Realidade Aumentada Móvel	31
2.2 Aplicações Relacionadas	31
2.2.1 Awila.....	32
2.2.2 Junaio	32
2.2.3 Layar	33
2.2.4 Blippar	33
2.2.5 Wikitude	33
2.3 Seleção dos Padrões Arquiteturais e de Projeto	33
2.3.1 Seleção de Artigos com Propostas de Arquitetura	34

2.3.2	Seleção dos Padrões de Arquitetura e de Projeto Conhecidos	36
3.	PROPOSTA DE ARQUITETURA E ASPECTOS DE PROJETO	42
3.1	Arquitetura	42
3.2	Casos de Uso.....	47
3.3	Diagrama de Classes.....	50
3.4	Diretrizes de Modificações de Funcionalidades da Aplicação.....	53
3.4.1	Remover Funcionalidade de Mapa	54
3.4.2	Adicionar Funcionalidade de Interação com Objetos 3D.....	55
4.	PROTÓTIPO DA APLICAÇÃO ARGUIDE	57
4.1	Mapa	59
4.2	Navegador de Realidade Aumentada	61
4.3	Leitor de QR Code.....	64
4.4	Funcionalidades Gerais.....	65
5.	TESTE EM CENÁRIOS DE USO	67
5.1	Cenário Artístico	67
5.2	Cenário Educativo	69
5.3	Cenário Turístico.....	70
6.	AVALIAÇÃO DE USABILIDADE DO ARGUIDE.....	72
6.1	Avaliação por Métodos de Inspeção	74
6.1.1	Avaliação Heurística de Usabilidade.....	75
6.1.2	Avaliação de Usabilidade Percurso Cognitivo	80
6.2	Avaliação por Métodos de Teste	82
6.2.1	Perfil dos Participantes	84
6.2.2	Avaliação de Usabilidade por Medição de Desempenho	84
6.2.3	Registro de <i>logs</i> do Usuário.....	86

6.2.4	Avaliação Protocolo Pensa Alto	89
6.3	Comparação dos Resultados entre Avaliações.....	94
6.4	Propostas de Melhorias.....	96
	CONCLUSÃO	98
	REFERÊNCIAS	100
	APÊNDICE A – FICHA DE AVALIAÇÃO PERCURSO COGNITIVO.....	104
	APÊNDICE B – FICHA DE AVALIAÇÃO HEURÍSTICA.....	109
	APÊNDICE C – RESULTADOS DA AVALIAÇÃO PERCURSO COGNITIVO	112
	ANEXO A - ARQUITETURA DOS ARTIGOS SELECIONADOS	115

1. INTRODUÇÃO

O desenvolvimento tecnológico dos dispositivos móveis trouxe inúmeras vantagens para o desenvolvimento de aplicações inovadoras para a busca e exploração de Pontos de Interesse (POI – *Points of Interest*). Dispositivos móveis como *smartphones* e *tablets* permitem que o usuário carregue consigo uma gama de aplicações com acesso a Internet, sensores de localização e movimentação, e com recursos computacionais suficientes para reprodução de conteúdos multimídia.

Uma pesquisa realizada com 4.044 usuários de dispositivos móveis na América Latina indica que aproximadamente 80% dos participantes utilizam *smartphones* em todos os lugares perguntados na entrevista, dentre eles: restaurantes, transporte público e nas compras. A Figura 1 mostra o resultado desta pesquisa (COMSCORE; IMS, 2015).

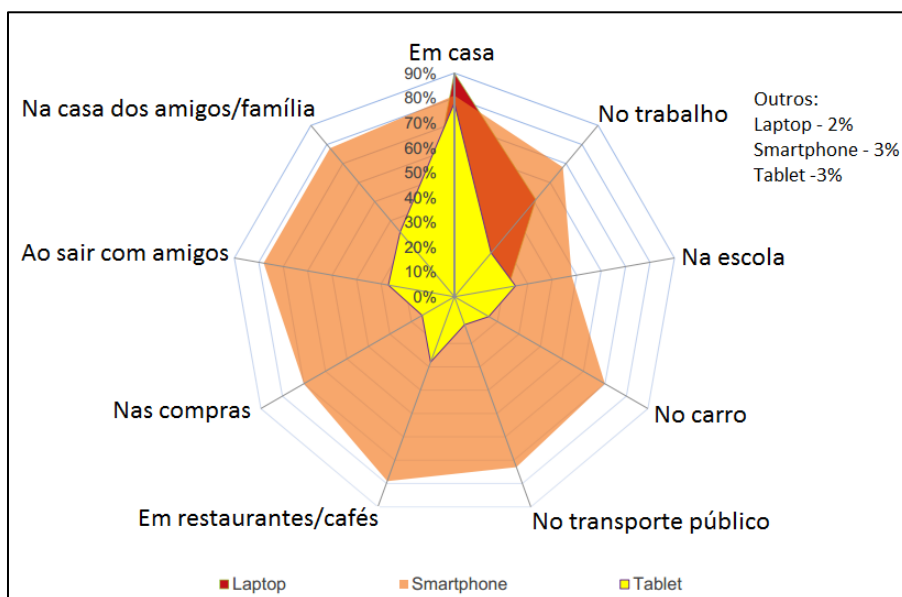


Figura 1. Gráfico da utilização de dispositivos móveis por local de uso. Adaptada e traduzida de comScore (COMSCORE; IMS, 2015).

Uma das tecnologias inovadoras que podem ser utilizadas nos dispositivos móveis é a Realidade Aumentada (RA), que tem o objetivo de melhorar a percepção humana sobre mundo real através da adição de elementos virtuais no mundo real.

Aplicações de RA em dispositivos móveis – conhecida como Realidade Aumentada Móvel (RAM) – têm se tornado cada vez mais populares, isto devido à característica inovadora da RA que chama a atenção dos usuários, ao desenvolvimento tecnológico dos dispositivos móveis que permite que aplicações mais robustas funcionem bem nestes dispositivos, e à popularidade e aceitação no mercado dos dispositivos móveis.

A RA antes era limitada aos *desktops*, pois os dispositivos móveis não tinham recursos computacionais suficientes para executar aplicações que capturam informações do contexto do usuário em tempo real e realizam várias atualizações (redesenho das informações virtuais em sua nova posição) na tela do usuário sempre que algo muda como movimentar a câmera.

Dentre as vantagens que a evolução tecnologia dos dispositivos móveis trouxe para as aplicações inovadoras, como as aplicações RAM, pode-se destacar:

- Um melhor processamento gráfico, que permite o desenho de elementos virtuais cada vez mais verossimilhantes ao real e com atualizações (redesenho deste elemento virtual) em tempo real;
- Maior memória tanto primária quanto secundária, o que permite armazenar e utilizar mídias de melhor qualidade, assim como a utilização de aplicações em paralelo sem grandes perdas de desempenho;
- Melhores tecnologias de telas que possibilitam uma melhor resolução e qualidade de cores do que é apresentado ao usuário;
- Quantidade e qualidade de sensores embutidos nos dispositivos móveis que permitem um melhor mapeamento do ambiente real do ponto de vista do usuário;
- Melhor qualidade das câmeras que possibilitam uma melhor captura do ambiente real.

Todas essas vantagens relacionadas aos dispositivos móveis e às aplicações RAM motivam estudos tanto em relação ao desenvolvimento de tais aplicações como em relação à aceitação e experiência dos usuários.

Olsson e Salo (2011) realizaram uma pesquisa com 90 participantes sobre a aceitação e experiência dos usuários em relação às aplicações RAM. Esse revelou que a maioria dos participantes utiliza essas aplicações em ambientes externos e em geral nos centros urbanos; que as razões para utilizar aplicações RAM são principalmente: a curiosidade com a RA e o desejo de conhecer algo novo; e que as melhorias que essas aplicações trouxeram, em relação a formas anteriores, foram principalmente: ver novas perspectivas em lugares ou objetos, adquirir informações importantes nos lugares e tempo certo, adquirir informações difíceis de encontrar, ter uma boa ideia das informações relacionadas com as proximidades (lugares ou objetos), como mostra o gráfico da Figura 2.

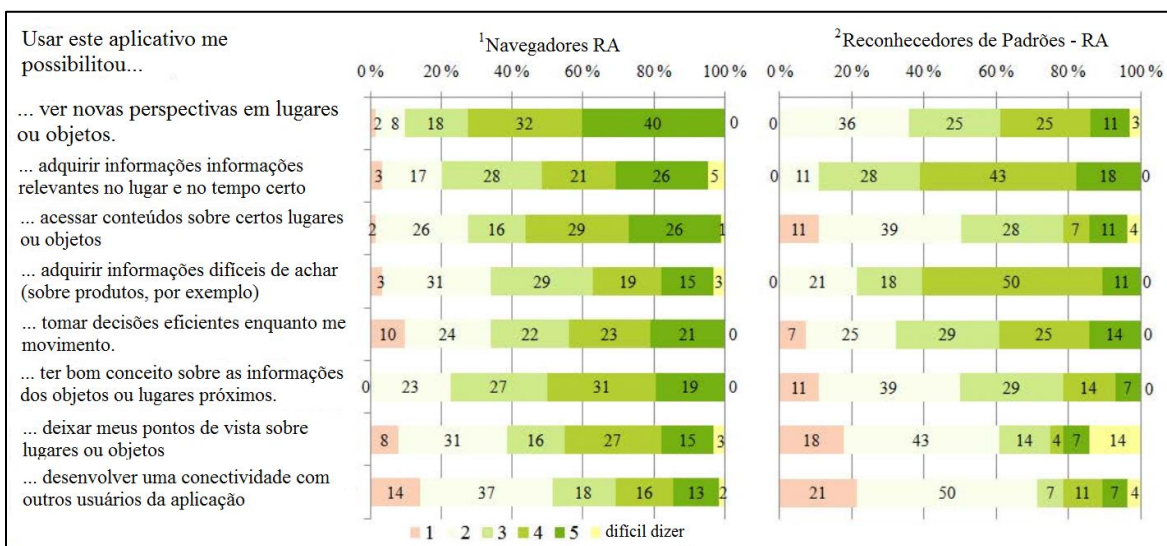


Figura 2. Gráfico da opinião dos usuários sobre as vantagens da utilização de aplicações RAM em relação aos métodos utilizados anteriormente. Legenda: 1-Pior que antes, 2-Não há diferença, 3-Um pouco melhor que antes, 4-Melhor do que antes, 5-Muito melhor que antes, 6-Difícil dizer. Adaptado de Olsson e Salo (2011).

¹Navegadores RA são aplicações que utilizam predominantemente GPS e sensores e ²Reconhecedores de Padrões - RA são aplicações que utilizam reconhecimento de imagem ou marcadores para inserir o conteúdo virtual. Mais informações na subseção 2.1.

As pesquisas de Olsson e Salo (2011) mostram que o interesse e a aceitação dos usuários sobre aplicações RAM são positivas, mas que ainda são necessários mais estudos e evoluções tanto na experiência de utilização dos usuários quanto no desenvolvimento dessas aplicações em si.

Além disso, as atuais aplicações de RAM que objetivam melhorar a busca e exploração de POIs possuem múltiplas funcionalidades (que podem ser entendidas com diferentes visões dos

mesmos POIs), porém não realizam a coordenação das múltiplas visões que o usuário possui, como mostrado na seção 2.2 (página 31).

Assim, este trabalho objetiva apresentar uma proposta para a descoberta, navegação e exploração de POIs através de múltiplas visões com a utilização de dispositivos móveis. As múltiplas visões devem ser coordenadas para fornecer ao usuário vários pontos de vista da mesma informação.

Devido à ausência de padrões no desenvolvimento de aplicações de RA e à ausência no desenvolvimento de múltiplas visões coordenadas foi necessário desenvolver e apresentar uma arquitetura e seus detalhes de projeto. A arquitetura proposta é baseada na plataforma Android, permite a extensibilidade da aplicação com baixo custo de implementação, a adaptação a uma diversidade de dispositivos com diferentes tamanhos de tela, e a flexibilidade na obtenção e utilização dos dados.

A arquitetura proposta trata-se de uma adaptação do padrão MVC (*Model View Controller*) (FOWLER, 2002) para contexto de aplicações de RA em plataforma móvel, onde a lógica de negócio sai da camada de controle e vai para camada de visão, e a camada de controle fica responsável por gerenciar as requisições entre a camada de modelo e camada visão.

Também foi utilizado o padrão de interface de usuário *Fragments*, visando a facilidade à adaptação, gerenciamento e responsividade aos diversos tamanhos de telas dos dispositivos. Além disso, foi aplicado o padrão de projeto *Proxy*, para abstração da origem dos dados (local ou remota), o padrão de projeto *Facade* para facilitar a realização de consultas e filtros nos dados e o padrão de projeto *Observer* para o gerenciamento e coordenação das múltiplas visões.

Foi desenvolvida uma aplicação baseada na arquitetura proposta, que possibilita a descoberta, a navegação, e a exploração de POIs através de três visões coordenadas, que são o mapa, o navegador de RA e o leitor de marcadores QR. A aplicação desenvolvida recebeu o nome de ARguide.

Por fim, foram realizados dois tipos teste na aplicação desenvolvida: teste de adequação à arquitetura proposta, realizado pelos próprios desenvolvedores, que utilizaram diferentes fontes de dados e tamanhos de tela; e testes de usabilidade por inspeção (especialistas) e avaliação por teste (usuários).

1.1 Justificativa

A realidade aumentada pode ajudar o usuário a realizar diversas tarefas do seu cotidiano, sejam tarefas no trabalho, em casa, ou em um passeio. Esta ajuda se dá por recursos computacionais que objetivam ampliar os sentidos do usuário em relação a uma dada tarefa. Esta tarefa pode ser, por exemplo, descobrir e visitar os pontos turísticos mais próximos, e quando um dispositivo computacional amplia o sentido de direção, visão e identificação do usuário através de marcadores virtuais a realidade aumentada ajudou o usuário a realizar sua tarefa.

A Realidade Aumentada Móvel traz este conceito para mais próximo dos usuários e da sua vida cotidiana, uma vez que os *smartphones* e *tablets* podem ser carregados e utilizados com mais facilidade em todos os lugares que o usuário se encontra.

Liu e Colaboradores (2012) avaliaram os benefícios da utilização de aplicações RAM com *feedbacks* em tempo real e em dispositivos de mão. Dezesesseis pessoas participaram do experimento realizando tarefas de configurações de um aparelho eletrônico. Os resultados mostraram que os participantes conseguiram realizar as tarefas mais difíceis mais rapidamente com a ajuda da realidade aumentada com *feedbacks* em tempo real, como mostra a Figura 3.

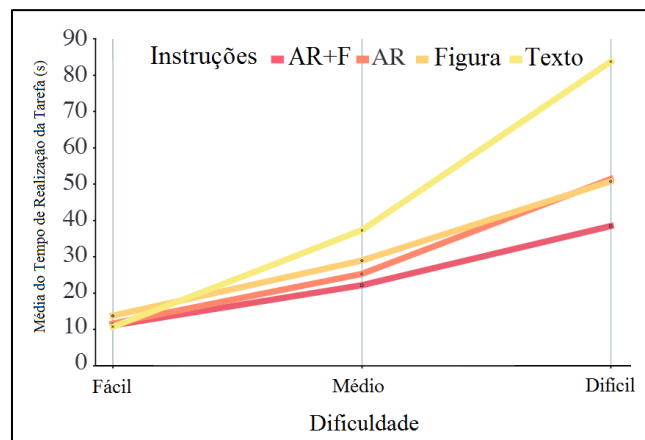


Figura 3. Gráfico dos resultados da pesquisa de Liu e Colaboradores que mostram a média de tempo da execução de uma tarefa, por sua dificuldade, utilizando: Realidade Aumentada (AR), Realidade Aumentada + *feedback* em tempo real (AR+F), Figura e Texto como forma de auxílio ao usuário.

Mesmo que os benefícios sejam claros ainda há muito que melhorar nesta área, que ao prover todos esses avanços, conseqüentemente apresenta novos desafios a serem alcançados. Arth e Schmalstieg (2011) listaram alguns desafios no desenvolvimento de aplicações RAM em

ambientes *outdoors*, como: a qualidade da câmera dos dispositivos, o consumo de bateria, a dependência da Internet, o espaço de visualização das informações, e as possíveis interações.

O contexto de aplicações RAM acrescenta ainda outros desafios, como: ausência de padrões de desenvolvimento definidos para aplicações, a maioria das aplicações apresenta pouca flexibilidade não permitindo seu uso em outros domínios, necessidade de evolução na precisão das tecnologias de localização e reconhecimento de padrões, pouco espaço para apresentação das informações virtuais, e há uma necessidade de testar a aceitação social destes novos recursos (MARTÍNEZ, SKOURNETOU, *et al.*, 2014).

A ausência de padrões de desenvolvimento de aplicações de RAM apontada por Martínez e Colaboradores (2014) foi confirmada após uma busca por propostas de arquitetura realizada nas principais bases de dados da área da computação, como descrito na Metodologia deste trabalho (seção 1.3 – primeira etapa). Porém, esta busca se limitou a plataforma Android.

Este trabalho se destina a plataforma móvel Android devido a sua popularidade e maior apoio à comunidade de desenvolvimento. A pesquisa da comScore (COMSCORE; IMS, 2015) realizada com 4044 usuários de *smartphones* ou *tablets* aponta que 78% destes são usuários da plataforma Android, considerando somente o Brasil, 82% dos participantes usam a plataforma Android.

1.2 Objetivos

O objetivo geral deste trabalho é apresentar uma proposta para a descoberta, navegação e exploração de pontos de interesse através de múltiplas visões coordenadas com a utilização de dispositivos móveis. A descoberta no sentido de encontrar algo novo, a navegação no sentido de direcionar o usuário até o ponto de interesse, e a exploração no sentido de disponibilizar dados dos sobre esses pontos de interesse.

Para propor a descoberta, navegação e exploração de POIs é necessário ter um bom projeto de desenvolvimento que permita a coordenação de múltiplas visões, ter funcionalidades que permitam estas características, e ter uma boa interface humano-computador para fornecer uma boa experiência com o usuário. Assim, os objetivos específicos do trabalho são:

- Selecionar os principais padrões de arquitetura e de projeto e adequá-los ao desenvolvimento de aplicações móveis de exploração de POIs voltadas para plataforma Android;
- Apresentar uma arquitetura e aspectos de projeto para aplicações extensíveis para funcionalidades, flexíveis para os dados utilizados e adaptáveis ao diversos tamanhos de tela dos dispositivos móveis;
- Desenvolver uma aplicação que tenha múltiplas visões e utilize a arquitetura proposta;
- Realizar testes na aplicação de adequação à arquitetura proposta:
 - Realizar testes em diferentes cenários de uso (diferentes dados);
 - Realizar testes em dispositivos com diferentes tamanhos de tela.
- Realizar testes de usabilidade na aplicação:
 - Realizar testes de inspeção com especialista em projetos de interfaces gráficas;
 - Realizar testes com usuários e questionários.

1.3 Metodologia

Esta subseção introduz os métodos utilizados neste trabalho, entretanto esses métodos serão abordados com mais detalhes no decorrer do trabalho para fins de organização. Este trabalho possui cinco objetivos específicos que são considerados como etapas para sua conclusão. Cada etapa utiliza um ou alguns métodos para sua realização. As etapas e seus respectivos métodos estão apresentados na Figura 4, que mostra um resumo dessas etapas em sequência e os métodos utilizados como tópicos a direta das etapas.

A primeira etapa corresponde ao estudo dos padrões arquiteturais existentes e o seus funcionamentos, a fim de utilizá-los ou adaptá-los para o desenvolvimento de aplicações móveis que tenham suporte à múltiplas visões, incluindo RAM, para a plataforma Android. Esta etapa possui dois métodos para sua realização, que são: a busca por propostas de arquiteturas para aplicações RAM em plataforma Android e o estudo de padrões de arquitetura e projeto apresentados pelos principais autores no tema. A identificação dos padrões é apresentada na subseção 2.3.

A segunda etapa objetiva apresentar uma arquitetura e os aspectos de projeto para o desenvolvimento de aplicações móveis que utilizem múltiplas visões coordenadas (incluindo funcionalidades de RA) para a descoberta, navegação e exploração de POIs voltado para a

plataforma Android. Para isto foram utilizados diagramas de caixas, a linguagem UML (*Unified Modeling Language*) – baseada em (FOWLER, 2003) – e a linguagem natural para explicar os detalhes presentes nos diagramas. A arquitetura e os aspectos de projeto são apresentados na seção 3.

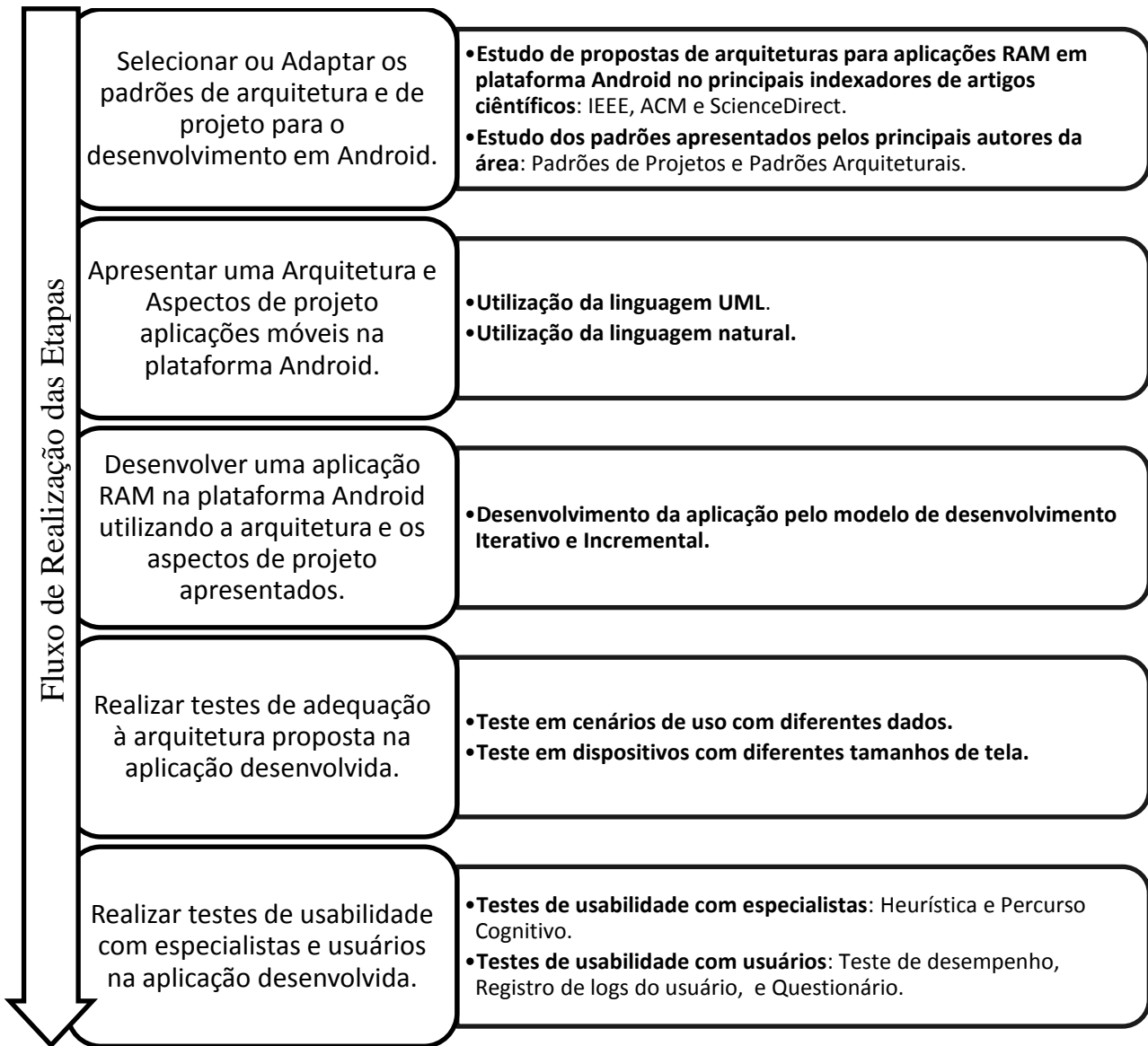


Figura 4. Etapas deste trabalho e os métodos utilizados em cada etapa.

A terceira etapa corresponde ao desenvolvimento de uma aplicação que permita ao usuário buscar, navegar e explorar POIs em múltiplas visões utilizando RA para a plataforma Android. A aplicação deve ser baseada na arquitetura apresentada na segunda etapa. O desenvolvimento da aplicação foi realizado por meio do modelo iterativo e incremental (SOMMERVILLE, 2010),

que possibilita desenvolver uma versão utilizável da aplicação com as funcionalidades básicas para seu funcionamento, e então incrementar as funcionalidades desta aplicação no decorrer do tempo. Este modelo de desenvolvimento coincide com a as características da arquitetura, já que esta foi desenvolvida para receber melhorias com pouco esforço de adequação ao código existente.

A quarta etapa refere-se ao teste de adequação da aplicação desenvolvida em relação à arquitetura proposta. Os testes de adequação objetivam identificar se a aplicação desenvolvida corresponde aos requisitos e características propostos na arquitetura.

Os testes da quarta etapa foram realizados pelos próprios desenvolvedores da aplicação a fim de avaliar a adequação à arquitetura proposta e encontrar *bugs*. Os testes utilizaram três cenários com dados diferentes em cada um deles e dois dispositivos móveis com diferentes tamanhos de tela e diferentes configurações de *hardware*. Esta etapa é apresentada na seção 5.

A quinta etapa corresponde à realização de avaliações de usabilidade na aplicação desenvolvida tanto com especialistas quanto com usuários. Os testes de usabilidade visam a identificar adequação da interface gráfica desenvolvida aos utilizadores finais da aplicação e melhorar a experiência dos usuários.

Para realizar a avaliação de usabilidade foram selecionados dois métodos de avaliação com especialistas, que foram: avaliação Heurística (NIELSEN e MOLICH, 1990) e a avaliação Percurso Cognitivo (WHARTON, RIEMAN, *et al.*, 1994). E foram selecionados três métodos de avaliação com usuários, que foram: medição de desempenho, registro de *logs* do usuário e questionário (MAHRIN, STROOPER e CARRINGTON, 2009). Estes métodos são apresentados com mais detalhes na seção 6.

1.4 Organização do Trabalho

Este trabalho está organizado em seis seções: Introdução, Realidade Aumentada, Aspectos de Projeto da aplicação RAM, Protótipo da aplicação RAM, Cenários de Uso, e Avaliação de Usabilidade do ARguide.

1 Introdução (página 16): apresenta o trabalho de maneira geral dando um panorama da área e da relevância do trabalho;

2 Referencial Teórico (página 26): realiza uma revisão sobre a área e introduz alguns conceitos importantes para o entendimento do trabalho;

3 Proposta de Arquitetura e Aspectos de Projeto (página 42): apresenta a arquitetura e os aspectos de projeto propostos;

4 Protótipo da aplicação ARguide (página 57): apresenta a aplicação desenvolvida a partir da arquitetura proposta;

5 Teste em Cenários de Uso (página 67): apresenta os resultados do teste de adequação à arquitetura em três cenários de uso;

6 Avaliação de Usabilidade do ARguide (página 72): expõe as avaliações de usabilidade que foram realizadas com especialistas e com usuários.

2. REFERENCIAL TEÓRICO

Esta seção descreve o contexto em que o trabalho está situado e o embasamento para a realização das etapas definidas na Introdução (resumo das etapas na Figura 4 – página 23). A área da realidade aumentada é descrita sucintamente abordando os fundamentos importantes para este trabalho, as principais aplicações relacionadas com o assunto do trabalho são mostradas, e os padrões arquiteturais e de projeto utilizados ao longo do trabalho são selecionados e apresentados.

2.1 Realidade Aumentada

Realidade aumentada é uma técnica utilizada para aumentar os sentidos do ser humano, seja adicionando ou completando o mundo real com conteúdo digital usando computadores (BIMBER e RASKAR, 2005).

O conceito de Realidade Aumentada é muito relacionado ao de Realidade Virtual (RV). A RV cria um mundo artificial onde é possível experimentar e explorar interativamente um mundo virtual semelhante ao mundo real por meio do senso de visão, áudio, tato ou outras formas de *feedbacks*. A RA também realiza essa experiência interativa, mas seu foco é complementar o mundo real com elementos virtuais, ao invés de criar um ambiente inteiramente virtual (BIMBER e RASKAR, 2005).

Há ainda estudos que classificam a Realidade Aumentada como uma particularização de uma área mais abrangente, conhecida como Realidade Misturada (RM). Segundo Milgram e Colaboradores (1994) a RM é composta pela RA e pela Virtualidade Aumentada (VA) (Figura 5). Na Virtualidade Aumentada o ambiente virtual é enriquecido com objetos reais, prevalecendo nesse contexto o virtual, enquanto que na RA o ambiente real é enriquecido com objetos virtuais, prevalecendo nesse contexto o real. A Realidade Misturada é a junção dos conceitos de RA e VA, sendo mais abrangente.

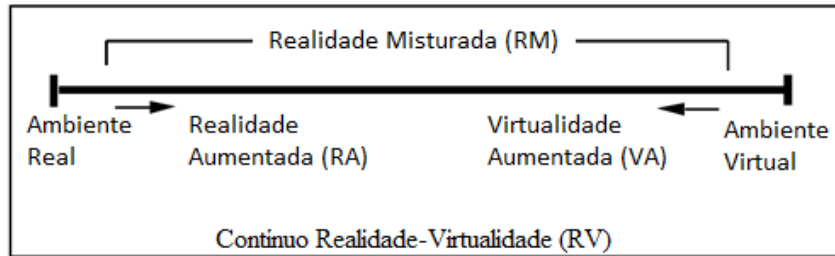


Figura 5. Contínuo Realidade-Virtualidade adaptado de Milgram e Colaboradores (1994).

Azuma e Colaboradores (2001) definem a Realidade Aumentada como um sistema que: combina elementos virtuais com o ambiente real, é interativa e em tempo real; e registra ou alinha objetos reais e virtuais em uma única cena, dando a impressão que os objetos virtuais realmente estão no ambiente real.

Existem vários sistemas de manipulação da Realidade Aumentada para plataformas desktops e dispositivos móveis. As aplicações de RA são encontradas em diferentes áreas, como: educação, jogos, física, geologia, engenharia, medicina, entretenimento, turismo, etc.

O surgimento de *tablets* e *smartphones* apoiou o desenvolvimento de aplicações de realidade aumentada para a plataforma de dispositivos móveis, visto que esses dispositivos têm como características principais seu pequeno tamanho e recursos computacionais semelhantes aos computadores *desktops*, o que facilita a mobilidade sem perder acesso à informação ou serviços. Outro fator importante dos dispositivos móveis para as aplicações de realidade aumentada é a presença de diversos sensores que auxiliam a obtenção de informações do ambiente real.

Sendo assim, as aplicações de RA para dispositivos móveis oferecem os recursos da RA sem restringir o local onde o usuário se encontra. As aplicações RAM trabalham virtualmente em qualquer lugar, adicionando uma camada palpável de informação para qualquer ambiente quando desejado.

2.1.1 Sistemas de Realidade Aumentada

A RA necessita integrar as informações virtuais dentro do ambiente real de tal forma que o usuário tenha perceba que esta informação está lhe ajudando ou ampliando seus sentidos em relação ao mundo real. Para melhor entender o funcionamento das aplicações de RA Bimber e Raskar (2005) descreveram um cenário típico para aplicações de modo geral, como mostra a Figura 6.

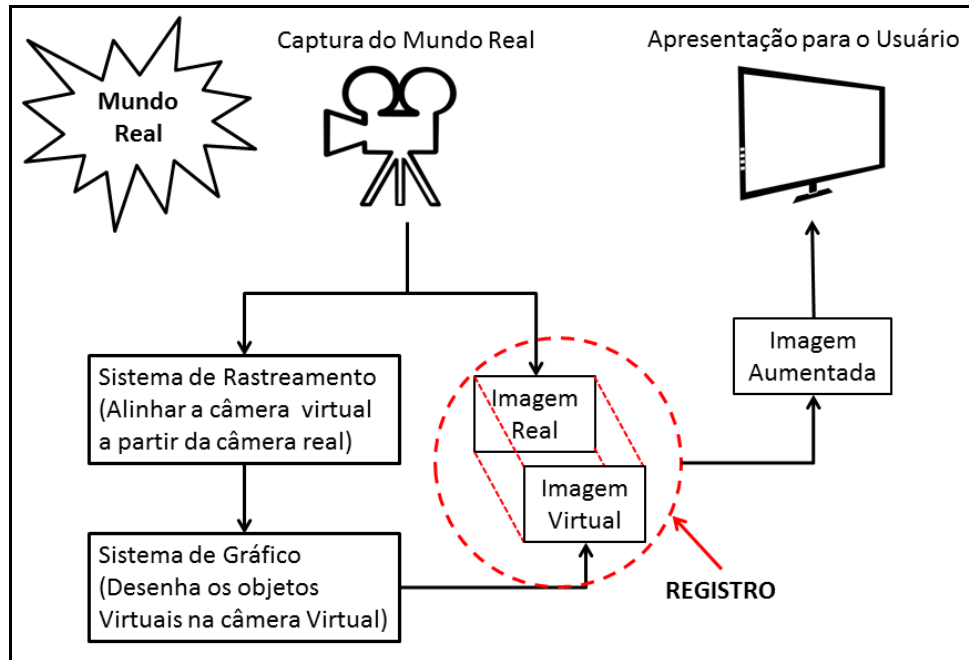


Figura 6. Cenário típico de um ambiente de Realidade Aumentada.

Os componentes presentes no cenário descrito por Bimber e Raskar (BIMBER e RASKAR, 2005) são descritos na lista a seguir resumidamente.

- **Captura do mundo real:** são os dispositivos utilizados para obter informações do mundo real, como a câmera para obter imagens e sensores para obter informações como localização e movimentação do usuário.
- **Sistema de Rastreamento:** é responsável por processar as informações brutas obtidas pela captura do mundo real de maneira que a aplicação consiga utilizar essas informações para mostrar informações virtuais ao usuário alinhadas com o mundo real. Existem dois tipos de sistemas de rastreamento:
 - **Rastreamento Óptico:** este realiza um reconhecimento de padrões nas imagens obtidas pela câmera e gera características para as informações virtuais, objetivando alinhar essas informações no mundo real. O reconhecimento de padrões pode ser na própria imagem capturada (como o reconhecimento de face) ou em marcadores (como o QR Code) presentes nestas imagens.
 - **Rastreamento por Sensores:** utiliza os dados capturados por sensores como: GPS (*Global Positioning System*), bússola, acelerômetro, etc. E calcula as

características das informações virtuais com o objetivo de alinhá-las ao mundo real.

- **Sistema gráfico:** este é responsável pelo desenho dos objetos virtuais. Este sistema é importante para melhorar a experiência do usuário em relação à qualidade gráfica das informações virtuais apresentadas, essencialmente se essas informações são tridimensionais e atualizadas em tempo real.
- **Registro:** é o ato de combinar camadas virtuais na cena real. Este deve ser feito de tal maneira que os objetos virtuais estejam bem alinhados a cena real.
- **Apresentação para o usuário:** é responsável por apresentar a imagem aumentada para o usuário. Existem vários tipos de tecnologias de apresentação do mundo aumentado, a Figura 7 mostra uma classificação dessas tecnologias de apresentação em relação a sua disposição no espaço e proximidade ao usuário.

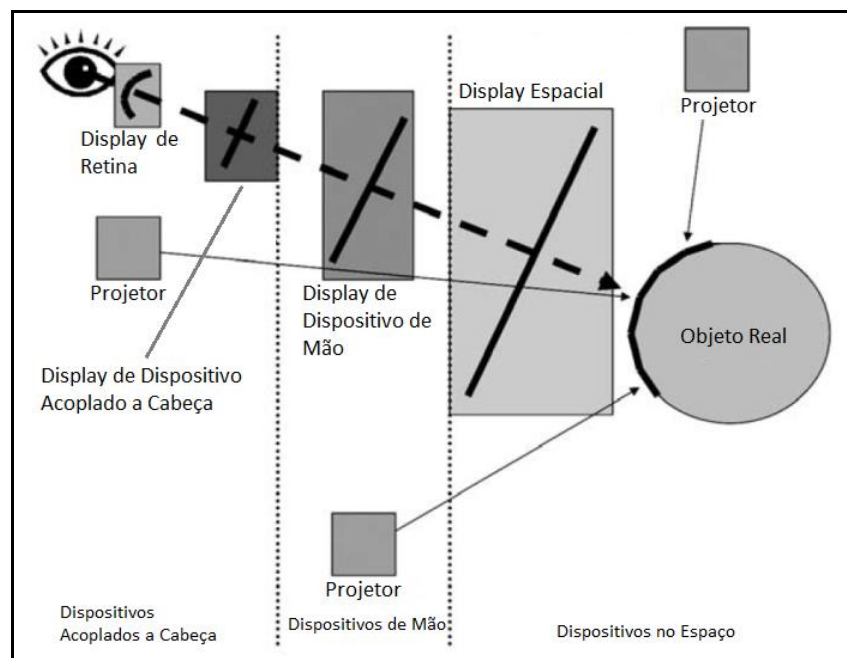


Figura 7. Tipos de *display* pela sua posição. Adaptado de Bimber e Raskar (2005).

2.1.2 Realidade Aumentada Móvel

Na RAM a cena do mundo real é capturada pela câmera do dispositivo móvel e é associada a uma camada de conteúdo virtual que é apresentado na tela do próprio dispositivo, a aplicação então apresenta ao usuário a camada virtual sobreposta à camada real como se fosse uma única camada. Como a captura da cena real e apresentação ao usuário estão no mesmo dispositivo, o

usuário tem a sensação de enxergar através de uma lente mágica, uma vez que elementos ou informações que não estavam na cena real, agora podem ser vistas através da tela do dispositivo móvel.

Além da portabilidade dos dispositivos móvel os sistemas presentes na RA estão agregados em um dispositivo, bastando apenas um smartphone ou *tablet* para usar aplicações de RA.

A lista abaixo descreve os principais desafios no desenvolvimento de aplicações RAM (MARTÍNEZ, SKOURNETOU, *et al.*, 2014) (WASSERMAN, 2010):

- Integração com sensores dos dispositivos;
- Baixa precisão das tecnologias de rastreamento;
- Restrições e diferentes características físicas dos dispositivos;
- Variabilidade na interface do usuário;
- Falta de padrões comuns que de fato sejam adotados pelos desenvolvedores de aplicativos RAM;
- Consumo de energia.

A lista abaixo descreve algumas diretrizes devem ser consideradas para o desenvolvimento de aplicações RAM, são elas (DÜNSER, GRASSET, *et al.*, 2007) (DE SÁ e CHURCHILL, 2013):

- Levar em consideração o perfil do usuário alvo, o uso outdoor, o uso de uma ou duas mãos, e o tempo de uso da aplicação;
- Seguir os bons princípios de usabilidade para RA e aplicações móveis: cena aumentada limpa, ícones e textos grandes e em camadas, interação com objetos 3D, e considerar o mundo real;
- Levar em consideração as restrições dos dispositivos: reflexão da tela e a falta de precisão no rastreamento;
- Percepção e a cognição do usuário devem ser estimuladas;
- Apresentação das informações virtuais: considerar a quantidade de informação, representação da informação, local das informações, utilização de múltiplas visões.

2.1.3 Navegação e a Realidade Aumentada Móvel

A evolução dos dispositivos móveis os tornou ferramentas poderosas para experiências de RA. Sistemas podem ser usados para mostrar representações virtuais de pontos de interesse sobrepondo o mundo real (SCHMALSTIEG e WAGNER, 2007). Aplicações desse tipo se tornaram muito populares e se mostraram eficientes ferramentas de navegação (DÜNSER, BILLINGHURST, *et al.*, 2012). Diferente dos mapas, onde se tem uma visão superior de uma região, aplicações RA adicionam informações virtuais do mundo real em uma visão antropocêntrica do lugar.

A navegação é uma função associada às tarefas de encontrar caminhos e locomoção. Enquanto acontecem movimentos no dispositivo, o usuário por meio da ferramenta consegue adquirir conhecimento espacial e estrutural, assim construindo um mapa mental dos seus arredores. Isso torna a movimentação do usuário mais intuitiva, já que as pessoas são acostumadas a usar objetos ao seu redor como referencial para se localizar.

A aplicação ARguide possui uma ferramenta de navegação de RA indicando pontos de interesse de acordo com a movimentação e direção apontada pelo usuário com o dispositivo. Ferramentas desse gênero têm grande destaque no turismo, onde pesquisas indicaram que 52% das pessoas utilizam esse tipo de ferramenta para descobrir o caminho para ir a lugar, 43% para chegar a um lugar que gostaria de ir e 23% para escolher um lugar específico para ir (TEEVAN, KARLSON, *et al.*, 2011).

2.2 Aplicações Relacionadas

Nesta subseção foram selecionadas algumas das principais aplicações relacionadas à proposta deste trabalho. Estas aplicações serviram de base para o desenvolvimento da mesma, sendo assim, possuem algumas funcionalidades em comum com a aplicação desenvolvida.

Apesar da aplicação ARguide não possuir todas as funcionalidades apresentadas, esta possui funcionalidades coordenadas, característica ausente em todas as aplicações pesquisadas. Adicionalmente, a arquitetura base da aplicação ARguide permite a adição das funcionalidades ausentes a um baixo custo de integração.

Algumas funções das aplicações apresentadas são comuns, e podem ser comparadas para verificar o quê estas aplicações oferecem para os usuários. Estas funções também foram usada

para definir uma prioridade para o desenvolvimento das funções do ARguide. A Tabela 1 mostra essa comparação entre as aplicações e as suas funcionalidades.

Tabela 1. Comparação entre as funcionalidades das aplicações selecionadas

	Navegador de RA	Mapa	Leitor de QR Code	Reconhecimento de imagem	Objetos 3D
ARguide	X	X	X		
Awila	X	X			X
Junaio	X		X	X	X
Layar	X	X	X	X	
Blippar				X	X
Wikitude	X	X		X	X

Abaixo é feita uma descrição resumida das aplicações e ferramentas selecionadas.

2.2.1 Awila

O Awila (2014) é um navegador RA que gera visualizações de POIs com informações extraídas de bases de dados com APIs (*Application Programming Interface*) públicas, como o ArcGISOnline, OpenStreetMaps e o GoogleMaps. A escolha das bases define o conteúdo que será visualizado no POI, e são baixados e carregados em tempo de execução da aplicação. O uso do mapa pode ser feito com uma projeção na visão da câmera, ou definindo um lugar fixo em que a visualização dele será configurada. A escolha de um POI só é possível enquanto o mapa está projetado na visão da câmera, e quando selecionado as informações são mostradas em uma pequena tela que fica a frente do mapa.

2.2.2 Junaio

O Junaio (METAIO, 2015) é uma plataforma RA que carrega informações de bases de dados criadas pelo próprio usuário, definido como canal. Esses canais são um conjunto de informações referentes aos POIs, e são carregadas conforme escolha do usuário. O conteúdo de cada POI pode ser visualizado de forma diferenciada pelo seu tipo. Quando selecionado, cada POI tem um conjunto de informações e um menu para alguma interação. Algumas interações são

diferenciadas, como ligar para o local, definir rota no mapa ou acessar o website do POI, mas essas ações exigem a inicialização de aplicativos externos ao Junaio.

2.2.3 Layar

O Layar (LAYAR, 2015) é uma aplicação que possibilita que os usuários busquem por POIs através do reconhecimento de imagem, Navegador RA, Mapa, e leitura de QR Code. Neste o usuário pode escolher uma das formas para busca de POIs e ao selecionar um POI ele pode interagir com o conteúdo cadastrado através de páginas. Os divulgadores de conteúdo devem cadastrar suas páginas através de uma plataforma Web que será automaticamente publicada na aplicação. O custo básico para publicar uma página na aplicação é de três euros por trinta dias.

2.2.4 Blippar

Blippar (BLIPPAR, 2015) é um aplicativo RAM que usa as funcionalidades de reconhecimento imagem e navegação RA para sobrepor informações virtuais no mundo real capturado pela câmera do dispositivo. Neste os usuários podem interagir com qualquer coisa que um *smartphone* ou *tablet* pode fazer incluindo jogos para celular, vídeos, músicas, e galerias de imagens. Em junho 2014, a Blippar comprou a empresa Layar.

2.2.5 Wikitude

Wikitude (WIKITUDE INC., 2015) é um aplicativo RAM inicialmente desenvolvido para prover aos usuários um navegador de RA com base em sensores e GPS, os usuários podiam procurar e explorar POIs através do navegador chamado de Wikitude *World Browser*. Em 2012, a empresa que desenvolve o Wikitude reestruturou o propósito de divulgação do aplicativo, tornando a ferramenta de desenvolvimento a principal forma de comercialização. Agora os provedores de conteúdo podem adquirir o Wikitude SDK (*Software Development Kit*) para disponibilizar seu conteúdo personalizado com as funcionalidades de reconhecimento de imagem, navegador RA, renderização de objetos 3D e vídeos sobrepostos ao conteúdo. A licença de uso do Wikitude SDK é paga e tem o valor por volta de 1980 euros por ano de utilização.

2.3 Seleção dos Padrões Arquiteturais e de Projeto

Para desenvolver uma aplicação móvel que permita ao usuário a descoberta, navegação e exploração de POIs é pertinente construir uma arquitetura que possibilite a adição e remoção de

funcionalidades e promova uma comunicação unificada entre essas funcionalidades, que podem ser entendidas como múltiplas visões de uma mesma informação.

Assim, se faz necessário identificar as propostas de arquiteturas existentes, verificar a adequação dessas arquiteturas ao propósito da aplicação, e caso não exista uma arquitetura adequada à aplicação pretendida, propor uma arquitetura que tenha suporte à múltiplas visões, incluindo RAM, para a plataforma Android.

A busca por propostas de arquiteturas foi realizada por dois métodos, que são: a busca por propostas de arquiteturas para aplicações RAM em plataforma Android e o estudo de padrões de arquitetura e projeto apresentados pelos principais autores no tema.

2.3.1 Seleção de Artigos com Propostas de Arquitetura

A busca por propostas de arquiteturas em artigos científicos foi realizada nos indexadores de artigos da área da ciência da computação: IEEEExplore, ACM Digital Library e ScienceDirect. Para realizar as pesquisas de artigos foram utilizadas as palavras: “Android”, “*Augmented Reality*”, “*Architecture*”, “*Coordinated Views*”, “*Coordinated Functions*”, e “*Design*”.

A fim de melhorar as pesquisas e obter resultados mais relevantes, foram utilizados conectores booleanos AND e OR entre as palavras formando as seguintes buscas: **1**- “Android” AND “*Augmented Reality*” AND (“*Architecture*” OR “*Design*”), **2** - “Android” AND “*Coordinated*” AND (“*Views*” OR “*Functions*”). Considerando como possíveis sinônimos ou palavras empregadas na mesma situação as palavras: *Design* e *Architecture*; e *Views* e *Functions*. Como de critério de busca, as palavras foram pesquisadas no resumo e título, através dos indexadores on-line de artigos.

A busca **2** não obteve nenhum resultado em todos os indexadores. Para a busca **1** foi verificado que em todos os indexadores utilizados, o artigo mais antigo é do ano de 2009, porém este artigo possui o termo “Android” para outro contexto e não o de sistema operacional para dispositivos móveis. Os outros artigos encontrados estão datados entre 2011 e 2014, como mostra a Figura 8.

Utilizando os parâmetros determinados foram obtidos 2 resultados na ACM Digital Library, 16 resultados na IEEEExplore e 5 resultados na ScienceDirect, num total de 23 artigos. Dos resultados obtidos foi feita a leitura e análise dos títulos e resumos, retirando os artigos com

temas não relacionados, estudos repetidos, estudos que não utilizam plataforma Android para dispositivos de mão, e estudos que não apresentam nenhuma descrição da arquitetura utilizada. Resultando em 1 artigos na ACM, 6 artigos na IEEE, e nenhum artigo na ScienceDirect, num total de 7 artigos.

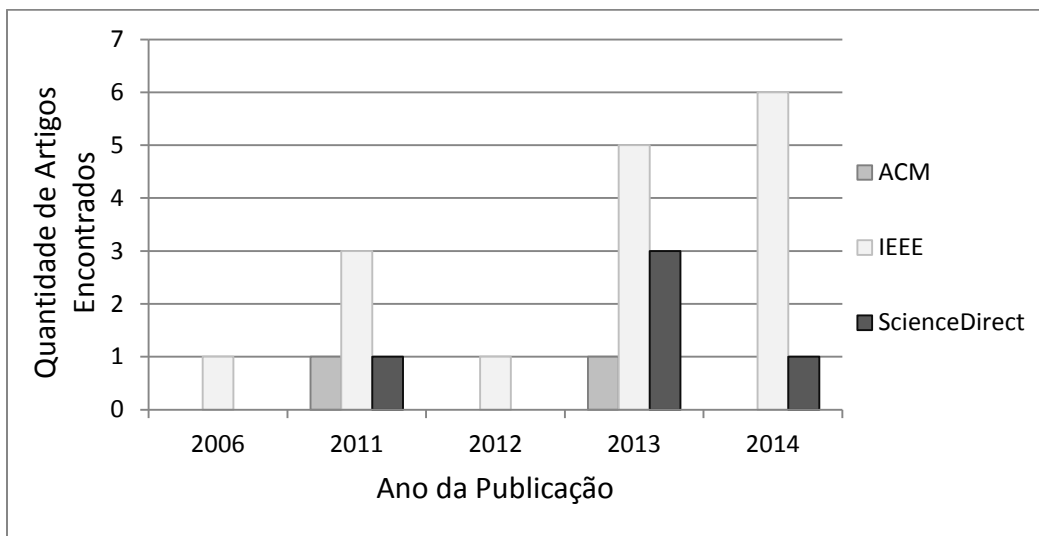


Figura 8. Gráfico da quantidade de artigos encontrados nos principais indexadores sobre Arquitetura, Android e Realidade Aumentada, por ano de publicação.

Após a leitura dos artigos foi constatado que os mesmos propõem uma aplicação ou serviço RAM e descrevem a arquitetura utilizada, não havendo uma proposta de arquitetura em si. Para classificar os resultados os artigos foram separados pelo sistema de rastreamento (subseção 2.1.1) utilizado na aplicação proposta no artigo. Resultando em três classificações: RA Padrões, RA Navegação e RA Padrões + Navegação, como mostra a Figura 9. Sendo que as aplicações RA Padrões utilizam o sistema de rastreamento óptico, as aplicações RA Navegação utilizam o sistema de rastreamento por sensores, e as aplicações RA Padrões + Navegação utilizam uma abordagem mista.

Os artigos de RA Padrões são propostas de aplicações que utilizam variados tipos de arquiteturas para o Reconhecimento de padrões, como QR Code ou imagens (LINS, ARRUDA, *et al.*, 2014), (SOBOTA, KOREČKO e HROZEK, 2013), (HAMID, ZHU, *et al.*, 2011), e (CHEN, XIANG, *et al.*, 2013). Assim, como os artigos de RA Navegação são propostas de aplicações que utilizam variados tipos de arquiteturas para navegação com sensores, como o GPS e acelerômetro (CHEN, PENG e TSENG, 2011), (VASSELAI, REIS e GOMES, 2011). O artigo de RA Padrões + Navegação representa uma abordagem mista (SRISUPHAB, SILAPACHOTE,

et al., 2014). Entretanto, nenhum dos aplicativos apresentou uma proposta de arquitetura apontando suas características e implicações de projeto de software.

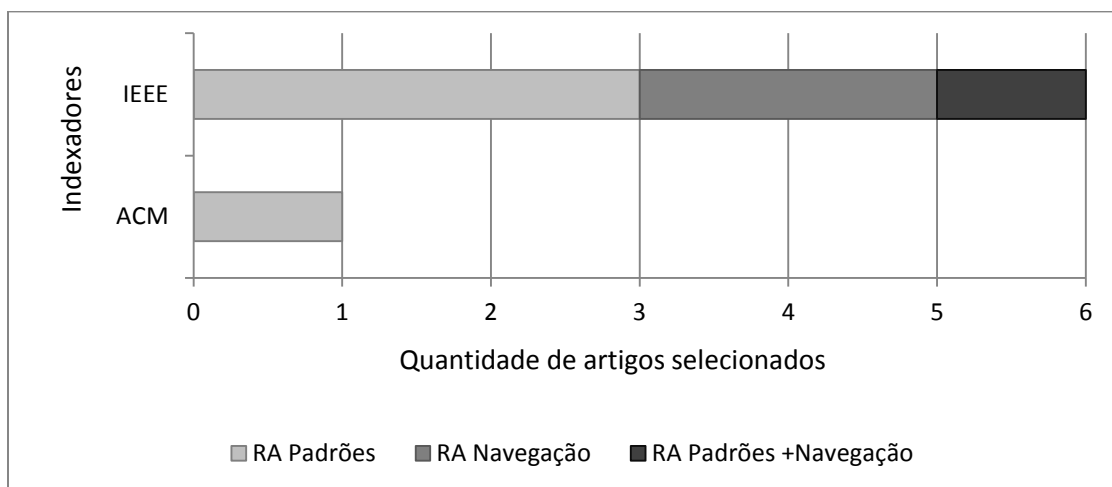


Figura 9. Gráfico da quantidade de artigos selecionados após a pesquisa nos principais indexadores, por tipo de aplicação proposta nos artigos.

O Anexo A mostra as arquiteturas presentes nos trabalhos selecionados. Nessas arquiteturas é percebida a ausência de um padrão, a ausência de tratamento para múltiplas visões coordenadas, e a ausência de uma estrutura que permita a extensibilidade de funcionalidades. Mas, vale ressaltar que o objetivo desses artigos não era propor uma arquitetura genérica, e sim explicar como o código das aplicações apresentadas estava estruturado.

2.3.2 Seleção dos Padrões de Arquitetura e de Projeto Conhecidos

Foram escolhidos três autores nas áreas de padrões de projeto, de arquitetura e de construção de interfaces gráficas, que são: Gamma e Colaboradores (1994), Martin Fowler (2006) (2002), e Nudelman (2013).

Os padrões de projeto são soluções conceituais reutilizáveis para um problema que ocorrem em um determinado contexto no projeto de software. Um padrão de projeto não representa a solução final para o problema, mas uma descrição ou modelo de como resolver tal problema.

Dentre os padrões apresentados por Gamma e Colaboradores (1994) foram selecionados três padrões de projeto para solucionar problemas com: a obtenção de dados de múltiplos destinos, a obtenção de dados simplificada via código fonte, e notificação automática de mudanças de estado, que são os padrões *Proxy*, *Facade* e *Observer* respectivamente.

O padrão de projeto *Observer* (Figura 10) possui uma interface com um método *notify()* para receber as notificações de forma unificada. Devido a essa interface eles podem ser armazenados em uma estrutura de dados única e que ao iterar por essa estrutura é possível notificar todos os observadores, como é mostrado no método *notifyObservers()*.

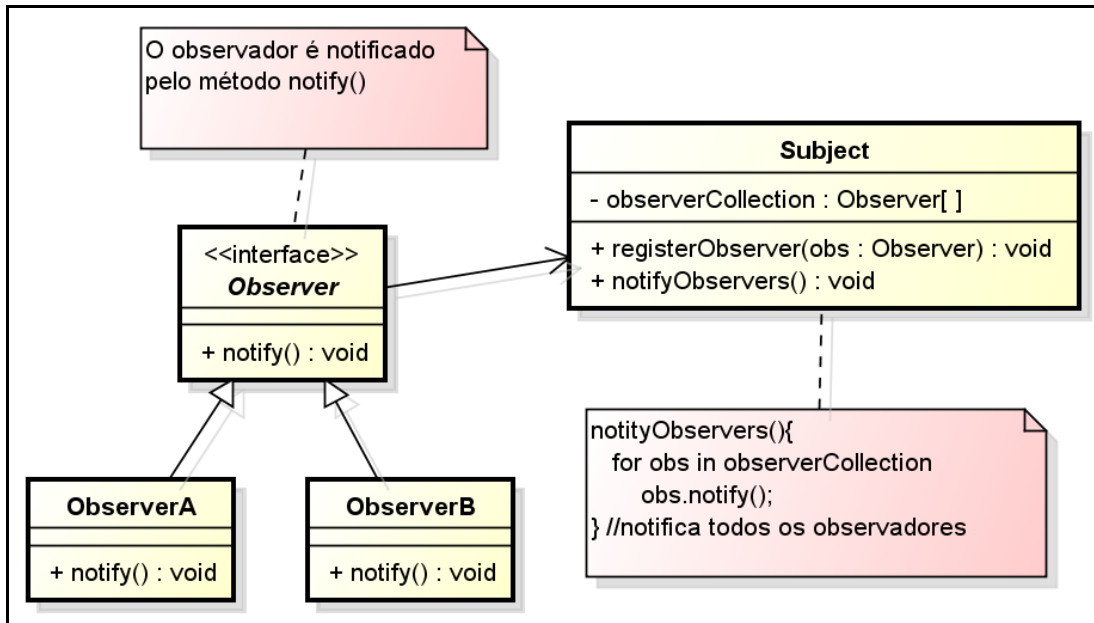


Figura 10. Diagrama de classes da estrutura básica do padrão *Observer*.

O padrão de projeto *Facade* (Figura 11) tem uma classe que disponibiliza métodos estáticos como o *doSomething()* que já realizam um determinado conjunto de tarefas para os clientes dessa classe. Esta classe tem o papel de melhorar o código fonte dos clientes deixando parte da complexidade na classe *Facade*. Além disso, um mesmo método pode ser reutilizado por mais de um cliente, ou em mais de uma situação.

O padrão de projeto *Proxy* (Figura 12) possui uma interface *Subject* que é inicialmente instanciada com a classe *Proxy*, quando um cliente chama algum método através da interface, por exemplo, o método *doSomething()*, a implementação *Proxy* realiza o seu processamento inicial, que pode ser baixar algum conteúdo da Internet, e a partir desse momento em diante todas as chamadas ao método *doSomething()* serão delegadas para a classe que realiza a atividade pretendida, por exemplo mostrar o conteúdo, que agora já foi baixado.

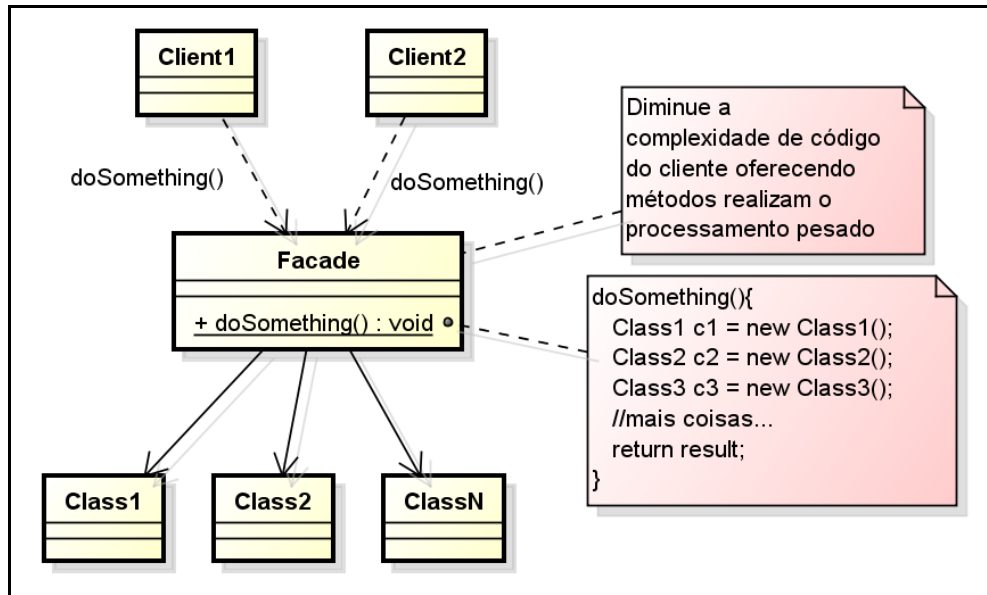


Figura 11. Diagrama de classes da estrutura básica do padrão *Facade*.

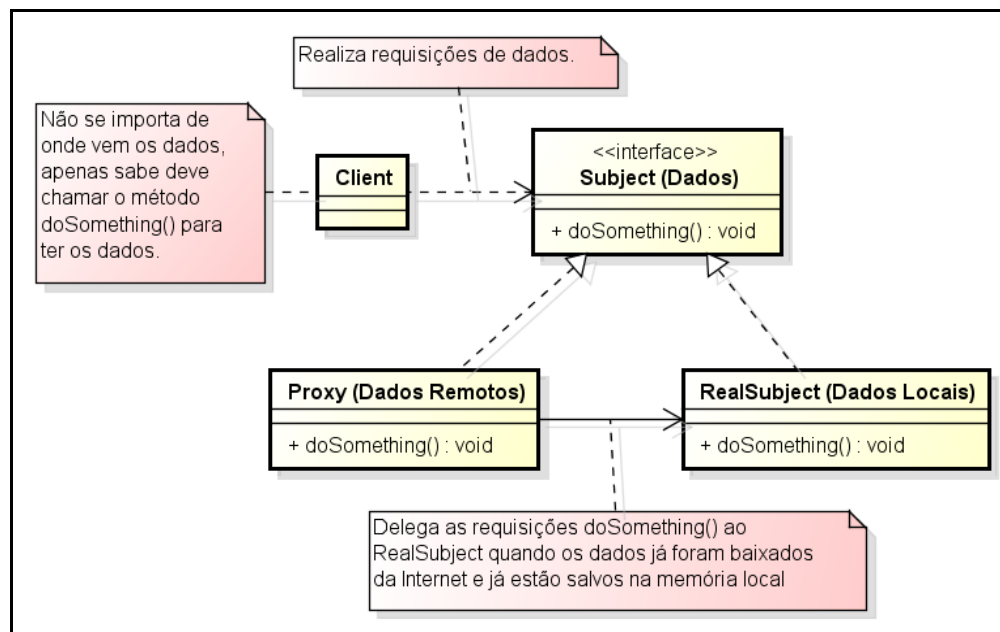


Figura 12. Diagrama de classes da estrutura básica do padrão *Proxy*.

Dentre os padrões apresentados por Fowler (2006) (2002) foram selecionados os padrões MVC e MVP (*Model View Presenter*), entretanto esses padrões arquiteturais tinham problemas de adequação ao estilo de programação da API Android. Assim, foram propostas algumas modificações nestes padrões arquiteturais, como discutido na subseção 3.1.

O padrão de arquitetura MVC (Figura 13) é um modelo teórico que prevê a separação dos módulos de um software em três camadas chamadas Visão, Controlador e Modelo. Sendo que cada uma dessas camadas devem conter códigos com três propósitos diferentes.

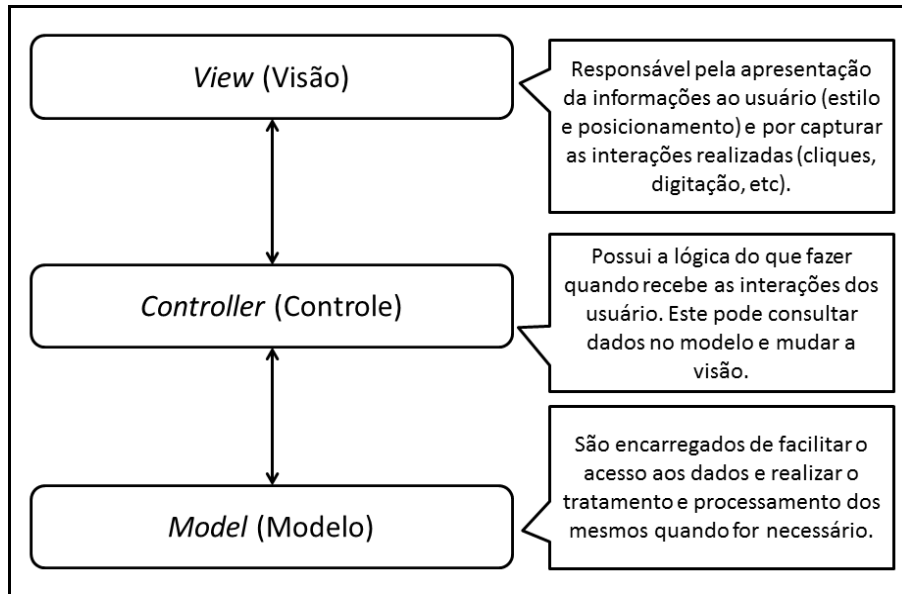


Figura 13. Estrutura do padrão de arquitetura MVC.

A visão devem conter apenas códigos referentes à apresentação de conteúdo ao usuário e referentes ao tratamento das interações com o usuário. O controlador deve conter a lógica do programa em si, ou seja, o que fazer a cada interação do usuário, quais os processamentos necessários, quais as próximas telas que devem ser mostradas a partir daí. Por fim, o modelo deve conter uma representação dos dados em forma de objetos, deixando os dados mais acessíveis em nível de programação orientada a objetos do que de modelo-relacional. O modelo pode conter também as regras de negócio dos dados.

O padrão de arquitetura MVP (POTEL, 1996) é uma proposta de adaptação do modelo MVC para a construção de interfaces gráficas. Esta prevê as visões são componentes gráficos como botões, campos de texto, rótulos, imagens, etc. E cada um desses componentes possui sua própria lógica de interação e programação.

No padrão MVP o Apresentador (*Presenter*) é responsável por mediar a comunicação entre o Modelo (*Model*) e a Visão (*View*), o Modelo, trata dos dados e a lógica de acesso aos dados, e a visão são os componentes gráficos.

Uma das formas conhecidas do MVP é o MVP *Passive View* (FOWLER, 2006) que canaliza todas as interações e os retornos do Modelo para as Visões pelo Apresentador, mas mantendo parte da autonomia presente nas Visões. Entretanto, no padrão MVP é criada uma instância de Apresentador para cada Visão deixando o projeto menos expansível, uma vez que para adicionar uma funcionalidade nova é necessário criar uma nova Visão e seu respectivo Apresentador, como mostra a Figura 14.

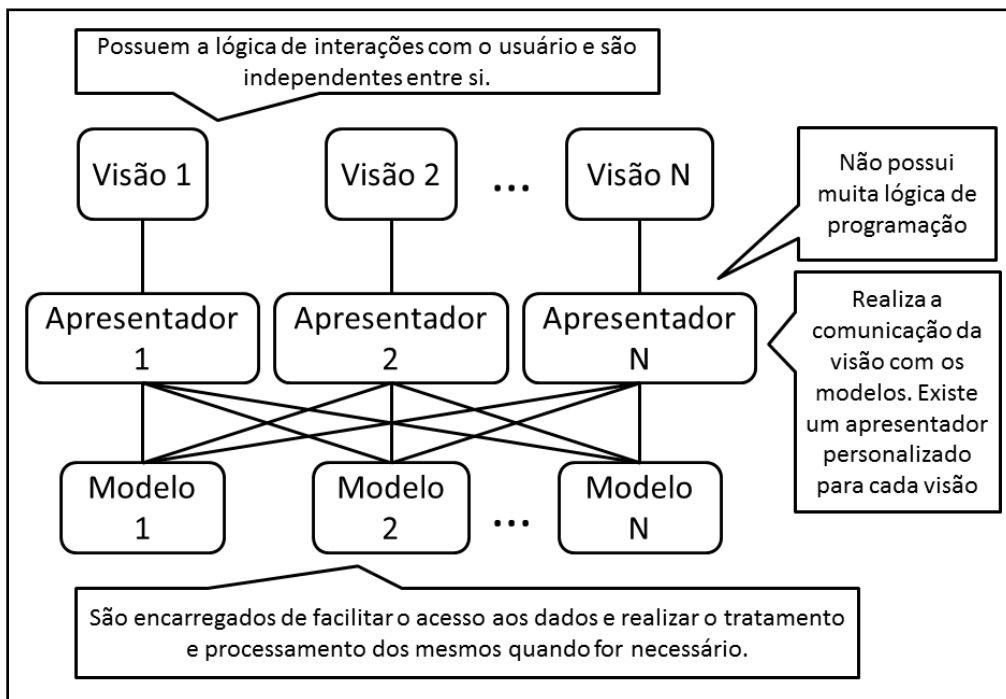


Figura 14. Estrutura do padrão de arquitetura MVP.

A camada Apresentador é semelhante ao Controlador, mas possui menos lógica de programação dando uma maior independência as visões. Entretanto, com cada componente gráfica precisa de dados personalizados ao seu propósito existe um apresentador para cada visão, que provê essa estrutura personalizada. Cada apresentador fornece apenas o que é necessário para cada visão.

Padrões de construção de interfaces gráficas são soluções modelo para os problemas encontrados com o posicionamento dos componentes gráficos na tela do usuário e possíveis interações. O padrão de construção de interfaces escolhido é o Fragmento, que já possui uma implementação na API de desenvolvimento do sistema Android.

Na implementação presente na API do Android cada Fragmento comporta-se dentro de seu próprio escopo, sendo capaz de tratar suas próprias interações, realizar processos independentes e gerenciar seus próprios componentes gráficos. Essa característica atribui mais flexibilidade e extensibilidade para a aplicação. Além disso, o padrão Fragmentos permite o design de layouts flexíveis e responsivos em aplicações móveis, o que significa que a mesma aplicação pode se adaptar a diferentes tamanhos de tela de dispositivos, evitando retrabalho de codificação para os diferentes dispositivos. A Figura 15 mostra como o padrão Fragmentos funciona em aplicações Android quando o tamanho da tela do dispositivo muda.

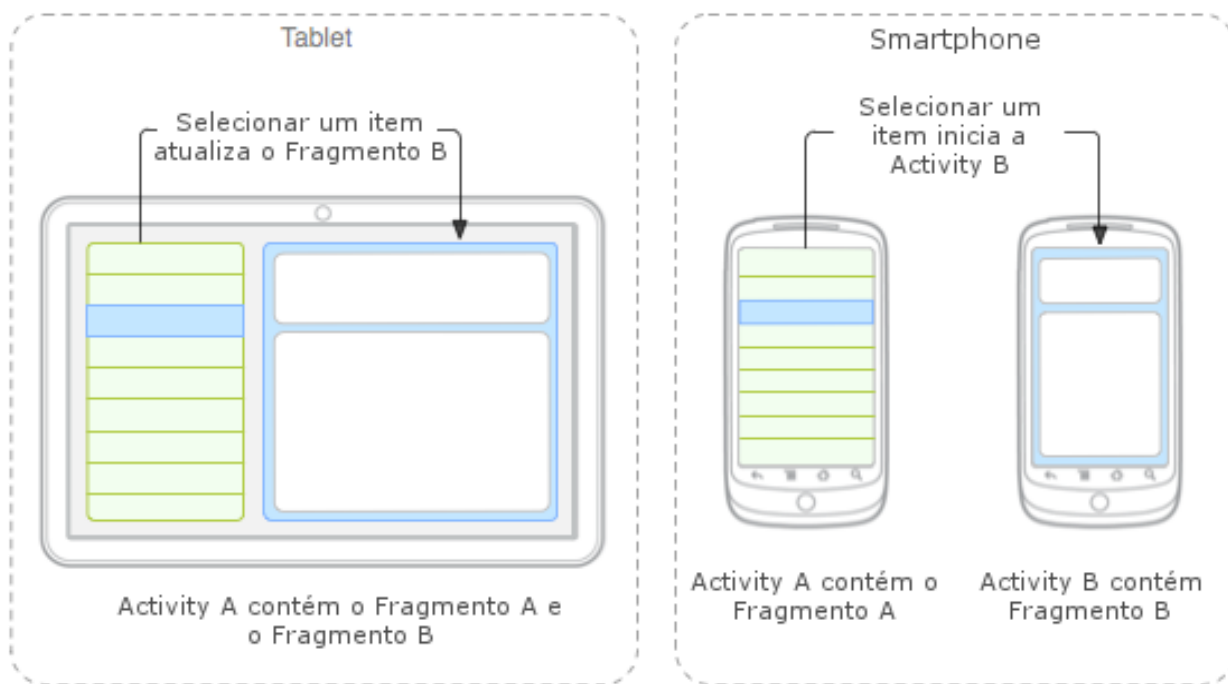


Figura 15. Ilustração do funcionamento do padrão *Fragments* na API Android. Imagem adaptada de (GOOGLE INC., 2015).

3. PROPOSTA DE ARQUITETURA E ASPECTOS DE PROJETO

Esta seção apresenta a arquitetura e os aspectos de projeto para o desenvolvimento de uma aplicação RAM. A apresentação da arquitetura e dos aspectos de projeto é realizada concomitantemente com a apresentação do ARguide, a aplicação que foi desenvolvida com base nesta arquitetura.

ARguide é uma aplicação de RAM que pode receber vários tipos de conteúdos georeferenciados e fornecer aos usuários este conteúdo de forma aumentada. Este aplicativo foi projetado para utilizar os recursos disponíveis em dispositivos móveis com sistema operacional Android, como: sensores, câmera, bússola, GPS, acesso a Internet, e outros, tornando o conteúdo da aplicação acessível e explorável por meio da RA.

A ideia desta aplicação é guiar o usuário que deseja conhecer mais sobre um determinado conteúdo, estimulando a visita de lugares e pontos de interesse, com a utilização de recursos como mapa, navegador de realidade aumentada, leitores de QR Code, e arquivos multimídia.

3.1 Arquitetura

Para oferecer os recursos de uma aplicação RAM independente dos dados que utilizados é necessário garantir uma boa modularidade de código entre os componentes que realizam as trocas de dados na aplicação, desde o nível arquitetural.

Um padrão arquitetural que ressalte a divisão de atribuições entre camadas, valorizando a modularização, a facilidade de implementação e a troca de módulos se faz necessário neste caso. Um padrão utilizado em aplicações *Web* que resolve este tipo de problema é o padrão MVC (FOWLER, 2002).

Apesar do padrão MVC se aplicar bem às aplicações *Web* foram identificados alguns problemas ao aplicá-lo diretamente em uma aplicação Android, uma vez que aplicações Android

devem seguir um ciclo de vida pré-estabelecido pela sua API de desenvolvimento e alterações na interface gráfica do usuário devem ser realizadas de uma maneira controlada, para evitar problemas de segurança.

Aplicações Android possuem como *entry point* (o primeiro trecho de código a ser executado) uma subclasse da classe *Activity*. E apesar da classe *Activity* não definir exatamente o que o usuário vê na tela do dispositivo móvel (papel este que cabe a classe *View* e suas subclasses), ela é responsável pela apresentação e manutenção do ciclo de vida de uma janela na interface gráfica da aplicação, e como uma *Activity* não pode realizar a manutenção de outras telas da aplicação esta classe não pode ter o papel de controlador na estrutura do MVC.

Dessa forma, se fez necessário reestruturar o padrão MVC com algumas modificações que fornecem uma melhor adequação ao estilo de programação de aplicações Android com a linguagem de programação Java, provocando uma modificação no papel de cada componente do padrão MVC.

A modificação proposta considera reduzir a quantidade de lógica aplicada no controlador e passar essa lógica para a camada visão, dando assim mais autonomia para as visões e uma menor dependência do controlador. Esta abordagem permite que as subclasses de *Activity* tenham mais autonomia e lógica de programação. A Figura 16 ilustra como essa ideia foi aplicada no padrão MVC.

No padrão MVC o Controlador possui lógica para as ações realizadas pelo usuário na Visão, ou seja, cada interação do usuário na interface gráfica dispara uma chamada para o Controlador que sabe o que realizar para cada tipo de interação do usuário.

Com a modificação proposta, o Controlador deixar de ter essa lógica passando a mesma para as Visões, que por sua vez apenas disparam chamadas para o Coordenador quando necessitarem realizar alguma operação de modificação de dados. Ou seja, quando o usuário interage com a interface gráfica da aplicação, a visão pode se ajustar ou realizar alguma operação sem disparar uma requisição para o coordenador, desde que essas operações não levem em consideração alterações nos dados da aplicação.

Devido à alteração proposta no comportamento da camada Controlador decidiu-se alterar o nome da camada para Coordenador, uma vez que este componente não possui mais lógica de

programação ficando com o papel de encontrar quais modelos devem ser alterados e de manter todas as visões coordenadas quando uma alteração for realizada.

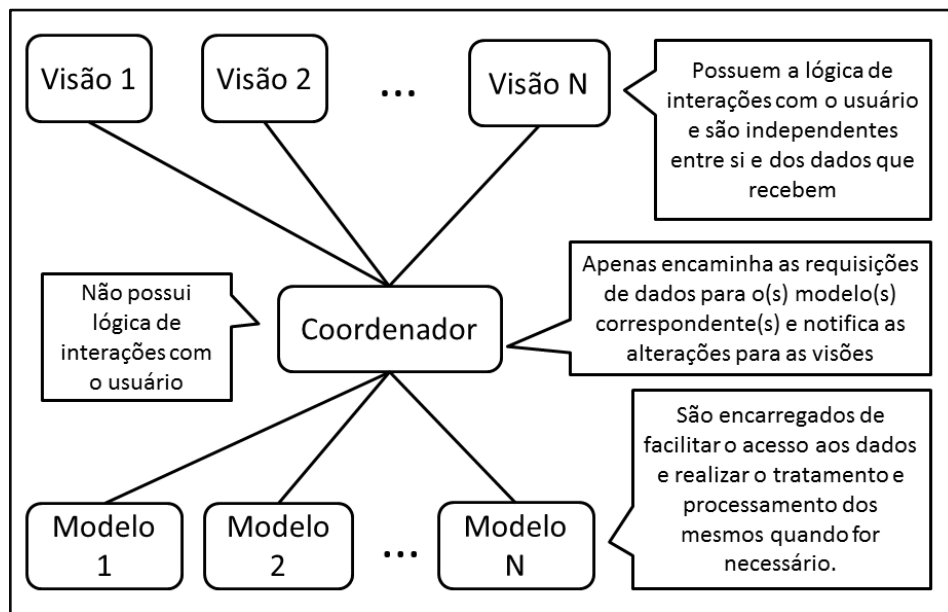


Figura 16. Esquema da modificação proposta no padrão MVC para adequação ao desenvolvimento de aplicações RAM na plataforma Android.

Uma característica importante associada ao Coordenador é a separação entre os componentes Visão e Modelo permitindo que alterações na interface gráfica do usuário não provoquem alteração na estruturação ou organização dos dados e vice e versa o que permite adicionar e remover novas Visões sem a necessidade de modificações no Modelo.

A arquitetura proposta é apresentada de maneira geral na Figura 17 (página 47), que mostra o fluxo de interações do usuário e respectivas respostas do sistema passando entre módulos e submódulos verticalmente. Os módulos apresentados estão descritos a seguir:

- **Visão:** responsável por manipular os eventos da interface do usuário, apresentar os dados vindos da camada Modelo para os usuários e encaminhar as interações do usuário para a camada Coordenador, apenas quando for necessário realizar alterações em dados. Esta camada possui mais lógica do que é previsto no MVC, essa característica permite mais autonomia para as classes *Activity* e *Fragments* da API do Android que intrinsecamente são estruturadas para receber tal lógica, como gerenciar o ciclo de vida de uma janela. Esta camada pode ter várias visões intercambiáveis, cada uma com sua própria lógica de

programação, ou seja, podem ser removidos e adicionados novas visões sem gerar modificações diretas nas camadas abaixo.

- **Coordenador:** é responsável pela comunicação entre as camadas Modelo e Visão, permitindo um intercâmbio de dados fracamente dependente entre as duas partes. Essa característica foi herdada do padrão MVC e permite que alterações realizadas na Visão não acarretem alterações no modelo e o mesmo vale para alterações no Modelo. A diferença entre o Controlador do MVC e o Coordenador é que o Coordenador possui menos lógica de programação, ou seja, ele não controla quando as Visões devem ou não aparecer ou para onde direcionar as interações do usuário. Neste caso o Coordenador apenas recebe uma solicitação de dados da Visão, repassa para a camada Modelo correspondente e atualiza todas as outras Visões de alguma alteração que tenha ocorrido, por exemplo, se o usuário realizou um filtro nos dados que estão sendo exibidos.
- **Modelo:** é responsável pela definição, estruturação e armazenamento dos dados a serem apresentados na Visão. Esta camada permanece igual à camada Modelo do MVC, uma vez que a camada Coordenador adiciona uma abstração às chamadas para os dados, esta camada pode se apresentar de várias formas. No ARguide decidiu-se criar subcamadas na arquitetura do Modelo, para facilitar a construção de *queries* (Consultas a dados) complexas e facilitar a recuperação de dados que estejam em nuvem. O Modelo é composto pelas seguintes subcamadas:
 - **Facilitador (Facade):** auxilia a realização de consultas e modificações nos dados oferecendo chamadas únicas para cada tipo de situação, isso tende a simplificar o código fonte das camadas que realizam as consultas, uma vez que, ao realizar consultas os dados podem não vir diretamente no formato desejado é necessário realizar processamentos e filtros nestes dados o que pode gerar várias linhas de código a mais em outras camadas, desfavorecendo a organização do código fonte e comprometendo a arquitetura do projeto. No caso do ARguide quem faz as requisições de dados é o Coordenador. Quando o Coordenador passa a realizar tratamento de dados é configurado um desvio em sua função, assim o Facilitador realiza este processamento e passa os dados prontos para serem usados em cada situação. A implementação do Facilitador segue o padrão de projetos *Facade* descrito por Gamma e Colaboradores (1994);

- **Proxy Modelo:** é responsável por abstrair as requisições de dados sobre qual a origem desses dados, no caso do ARguide duas origens são possíveis: na memória local do dispositivo do usuário ou na nuvem em um servidor de dados. Para os clientes desta camada não importa de onde vieram os dados, em outras palavras ele não sabem de onde veio os dados somente recebem os mesmos. Este padrão se encarrega de delegar a função de recuperação de dados para um de seus submódulos de acordo com a presença dos dados na memória local do dispositivo do usuário. Esta camada segue o padrão de projetos *Proxy* descrito por Gamma e Colaboradores (1994).
 - **Modelo Local:** é encarregada de recuperar os dados quando os mesmos estão armazenados no dispositivo do usuário, esta camada acessa a memória local do dispositivo do usuário para recuperar os dados que foram requisitados e passa esses dados para a camada Coordenador;
 - **POIs Proxy:** é encarregada de recuperar os dados quando os mesmos não estão armazenados no dispositivo do usuário, esta camada recupera os dados utilizando a Internet do dispositivo do usuário e armazena estes dados na memória do dispositivo do usuário. Assim que os dados estão armazenados esta camada deixa de responder as requisições para este dado e delega esta função para a camada Modelo Local correspondente. Caso o usuário não tenha os dados armazenados localmente e não tenha nenhum acesso a Internet este módulo entra em modo de espera até que uma conexão com a Internet seja estabelecida.

Neste projeto foi utilizado o padrão de interfaces gráficas de usuário Fragmentos (*Fragments*) (NUDELMAN, 2013) como uma possível implementação da camada Visão. O padrão de interface de usuário Fragmentos (GOOGLE INC., 2015) vem sendo bastante adotado e incentivado pela comunidade de desenvolvimento para sistemas Android, principalmente pela possibilidade de criar interfaces mais ricas, interativas e ubíquas. Este padrão foi utilizado para melhor organizar as funcionalidades disponíveis para o usuário, assim como, permitir uma melhor coordenação entre as visões.

No ARguide cada fragmento é responsável por tratar as interações que o usuário realiza sobre ele e decidir se deve ou não enviar uma chamada para a camada Coordenador. Isso significa

dizer que, por exemplo, um fragmento de navegador de realidade aumentada sabe como tratar as interações do usuário como toque e movimentação do dispositivo (no caso do navegador RA movimentar o dispositivo pode alterar a direção e a distância em relação aos POIs tendo que atualiza esta informação para o usuário), e só realiza chamadas para o Coordenador quando um POI é alterado ou selecionado.

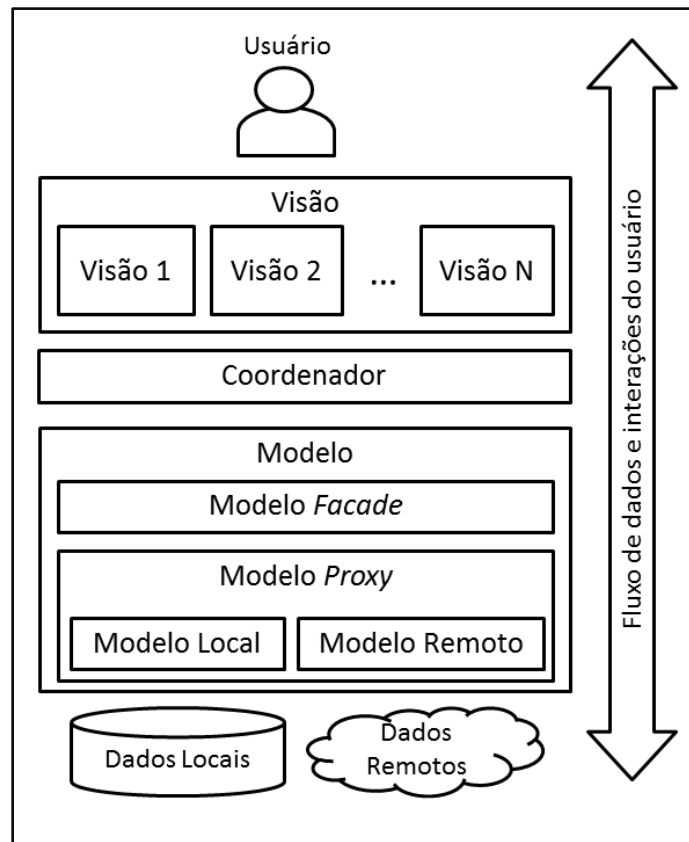


Figura 17. Arquitetura proposta com o fluxo de dados e interações do usuário.

3.2 Casos de Uso

Esta subseção apresenta os casos de uso definidos para a aplicação ARguide de acordo com as regras descritas por Fowler (FOWLER, 2003). O diagrama de casos de uso do ARguide foi desenvolvido antes da concepção da aplicação propriamente dita para auxiliar o desenvolvimento e elaboração dos requisitos do sistema, e por isso sofreu algumas alterações no decorrer do desenvolvimento, mas as principais funcionalidades foram mantidas na aplicação.

A Figura 18 apresenta o diagrama de casos de uso utilizado para a concepção dos requisitos funcionais da aplicação ARguide na linguagem UML (FOWLER, 2003).

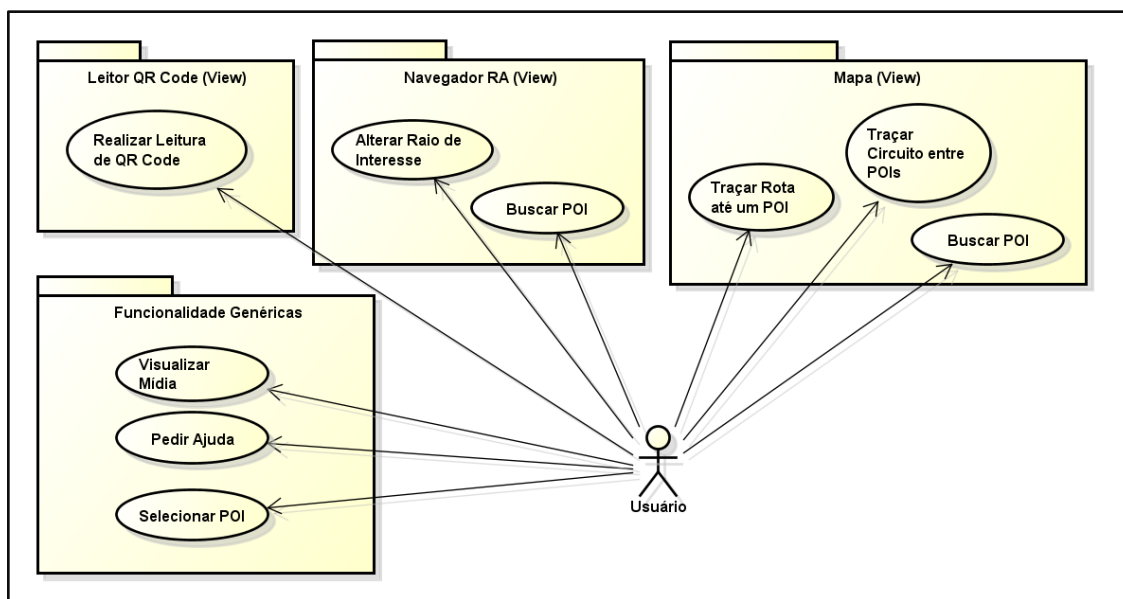


Figura 18. Diagrama de Casos de Uso do projeto da aplicação ARguide.

O diagrama de casos de uso mostra o ator Usuário interagindo com os casos de uso através de quatro módulos. O módulo inferior à esquerda nomeado “Funcionalidades Genéricas” corresponde às ações que o usuário pode realizar independente da Visão que o mesmo está interagindo. Os outros três módulos são as Visões presentes no ARguide no qual os usuários realizam interações que serão tratadas individualmente por cada Visão.

No módulo de Funcionalidades Genéricas o usuário pode: Selecionar Mídia, Interagir com Mídia, e Acessar a Ajuda. Ao selecionar um POI em uma visão o usuário terá acesso a uma lista de mídias associadas a esse POI, assim ele pode realizar a ação de selecionar uma dessas mídias. Ao selecionar uma mídia o usuário poderá interagir com essa mídia, reproduzindo um vídeo, ampliando uma imagem, lendo um texto, e assim por diante.

Caso o usuário tenha dúvidas na utilização da aplicação ele pode acessar um botão de Ajuda que descreve as funcionalidades da aplicação e como utilizá-las de maneira correta.

Através da Visão Leitor QR Code o usuário pode utilizar a câmera do seu dispositivo para realizar a leitura de códigos QR Code e extrair informações de um determinado POI ou interagir com uma determinada mídia. O QR Code que a aplicação ARguide reconhece possui um código específico que leva a um determinado POI ou mídia que já esteja cadastrado no sistema (na memória local do dispositivo ou em nuvem), ou seja, os códigos QR Code lidos não inserem

novo conteúdos para o usuário, apenas ajudam o usuário a selecionar um POI ou mídia relacionada que esteja marcado com um QR Code.

Através da Visão Navegador RA o usuário pode Buscar POI explorando o ambiente ao seu redor de forma tridimensional através da câmera do seu dispositivo e os marcadores virtuais que indicam a posição dos POI no ambiente. Esta Visão dá ao usuário uma melhor noção de direcionamento uma vez que os marcadores viabilizam que o usuário veja os POIs no ambiente real mesmo que esses POIs estejam oclusos.

Outra funcionalidade da Visão Navegador RA é Alterar o Raio de Interesse, no qual o usuário pode aumentar ou diminuir a distância máxima (entre ele e os POIs) que ele deseja para que os POIs sejam visualizados na tela. Quando o usuário ajusta essa distância máxima, todo POI que esteja além desta distância em relação ao usuário não será visualizado na tela. Esta característica é um filtro que ajuda o usuário a se concentrar em POIs que estejam mais próximos, uma vez que podem existir muitos POIs e estes podem poluir a tela do usuário.

Através da Visão Mapa o usuário pode Buscar por um POI no mapa de sua cidade/bairro, tendo como referência as ruas, praças entre outro, permite ao usuário localizar os POIs com base em sua posição atual em relação a essas referências. O usuário dispõe de funcionalidades como *zoom* e *pan* para buscar mantendo o foco na região desejada.

A segunda funcionalidade disponível na Visão Mapa é a possibilidade de traçar uma rota até um POI específico. Esta funcionalidade destaca um trajeto que parte do usuário até o POI escolhido passando pelas ruas disponíveis na cidade/bairro do usuário. O usuário pode sempre acompanhar se está realizando a trajetória correta até o POI.

A terceira funcionalidade disponível na Visão Mapa é a possibilidade de traçar um circuito entre os POIs. Esta funcionalidade destaca um circuito que parte do usuário e passa por todos os POIs disponíveis na aplicação passando pelas ruas da cidade/bairro do usuário. O usuário pode sempre acompanhar se está realizando o circuito corretamente e caso o usuário não realize uma visitação na ordem especificada a aplicação realiza um reordenamento de acordo com a preferência do usuário.

3.3 Diagrama de Classes

Esta subseção apresenta as principais características de projeto de software da aplicação ARguide, dando ênfase para a utilização dos padrões de projeto utilizados na aplicação, para o funcionamento e comunicação das principais classes envolvidas no código da aplicação, para a lógica de funcionamento da aplicação, e para a relação entre organização das principais classes do projeto de software e a arquitetura.

Visando uma melhor organização dos módulos da aplicação e uma melhor leitura em relação à sua arquitetura, as classes do projeto foram divididas em três pacotes nomeados *View*, *Coordinator* e *Model* que significam Visão, Coordenador e Modelo respectivamente.

Os pacotes correspondem às camadas do modelo MVC estabelecido na arquitetura e as classes de cada pacote possuem apenas as funcionalidades adequadas às camadas da arquitetura. Assim, o pacote *View* possui apenas classes responsáveis pelas interações com o usuário e a lógica dessas interações, o pacote *Coordinator* possui uma classe responsável por intermediar a comunicação entre as classes dos pacotes *View* e *Model* e manter as Visões coordenadas, já o pacote *Model* possui apenas classes responsáveis pelo gerenciamento e processamento de dados.

A Figura 19 mostra o diagrama de classes da aplicação dando foco nos principais métodos e composições, e heranças presentes no código da aplicação. O diagrama de classes é fundamental para o entendimento do funcionamento geral da aplicação em nível de programação, entretanto o diagrama de classes apresentado tem um propósito explicativo e por isso foram removidos alguns detalhes menos importantes a fim de melhorar a visibilidade dos módulos.

O pacote a esquerda denominado *View* contém as classes responsáveis por toda a interação com o usuário e a lógica do funcionamento dessas interações. Essas interações são tanto entradas do usuário quanto saídas para o usuário, ou seja, qualquer tipo de entrada do usuário ou saída para o usuário é considerado interação. Por exemplo, quando o usuário movimenta o dispositivo realizando rotações tridimensionais, considera-se que ele está interagindo entrando com parâmetros, que neste caso é a posição/direção do dispositivo no espaço. Isto vale tanto para textos quanto para outros tipos de interfaces multimodais.

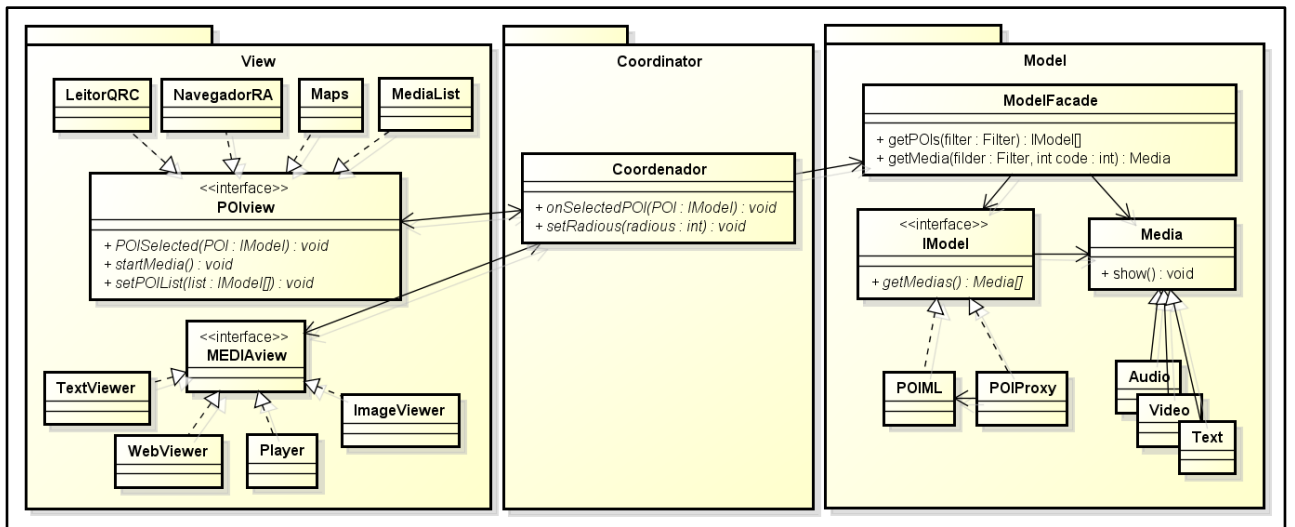


Figura 19. Diagrama de classes do projeto da aplicação ARguide.

O diagrama de classes destaca duas interfaces no pacote *View*: *POIview* e *MediaView*. A interface *POIview* representa uma maneira padrão de comunicação com as classes que fornecem as interações com o usuário da aplicação e tenham manipulação direta com os POIs. Ou seja, ela possui os métodos que podem ser chamados de maneira única tanto para informar que um POI foi selecionado, quanto para atualizar o filtro de POIs. Esta característica permite que a classe *Coordenador* trate cada implementação da interface *POIview* de maneira unificada.

Cada implementação da classe *POIview* tem sua própria lógica para tratar as interações do usuário, assim como seu próprio espaço na tela da aplicação, ou seja, cada uma delas trata as interações dos usuários de forma independente, o que possibilita que elas sejam removidas ou adicionadas na aplicação sem prejudicar o funcionamento da outra. Isto implica que, por exemplo, ao remover o Mapa da aplicação, todo o resto continua funcionando da mesma forma, e se for adicionado um módulo de reconhecimento que implementa a classe *POIview* também todos os outros módulos continuam inalterados.

A outra interface presente no pacote *View* é a *MEDIAview*, esta interface tem a mesma finalidade da interface *POIview*, entretanto não possui métodos de comunicação com a classe *Coordenador* sobre alterações em POIs. As classes que implementam esta interface não mostram uma lista de POIs para o usuário e sim outras informações, que podem estar relacionadas com um POI selecionado ou não. Este é o caso da tela que exibe um texto para o usuário sobre um determinado POI, neste caso qualquer interação com o texto não altera o estado dos POIs. Então

as implementações de *MEDIAview* recebem qual o POI selecionado atualmente da classe *Coordenador*, mas não disparam mensagens para o mesmo.

O pacote do meio denominado *Coordinator* contém uma única classe chamada *Coordenador*. Esta classe é responsável por abstrair a comunicação entre as classes do pacote *View* e a classes do pacote *Model*, em outras palavras favorecer uma separação entre os dados e as interações com o usuário. Esta separação permite que as modificações que ocorram nas classes do pacote *Model* acarretem apenas em poucas modificações em uma única classe.

A classe *Coordenador* tem uma função primordial para a coordenação das implementações de *POIview*, uma vez que ela é encarregada de notificar as alterações que ocorrem em uma classe para todas as outras implementações de *POIview*. Assim as implementações da classe *POIview* não se comunicam diretamente o que reforça a ideia de que essas classes não dependem uma da outra.

As implementações de *POIview* e a classe *Coordenador* se comunicam de acordo com o padrão de projetos *Observer* descrito por Gamma e Colaboradores (1994). Neste caso, a classe *Coordenador* registra os observadores (as implementações de *POIview*), entretanto o estado de modificação não vem dele mesmo e sim dos próprios observadores, por isso é necessário uma dupla composição entre as implementações de *POIview* e o *Coordenador*. Assim quando uma mudança ocorre no estado de um ou mais POIs, a implementação de *POIview* que recebeu essa mudança do usuário notifica o *Coordenador* e o *Coordenador* notifica as demais implementações de *POIview*, promovendo a coordenação.

O pacote *Model* contém as classes responsáveis por toda manipulação, gerenciamento e processamento dos dados da aplicação. Essas classes favorecem uma melhor comunicação entre a aplicação e os dados, abstraem a origem desses dados e facilitam que a aplicação receba outros dados sem a necessidade de recodificação.

A classe *ModelFacade* facilita as operações de recuperação, inserção, filtro e modificação de dados para a classe *Coordenador*, assim cada solicitação que a classe *Coordenador* realiza já vem pré-processada e pronta para utilização. Esta classe foi implementada seguindo o padrão de projetos *Facade* (GAMMA, HELM, *et al.*, 1994) sendo assim oferece métodos públicos em nível de classe que recebem alguns parâmetros e realizam toda a montagem e processamento dos dados para que eles sejam utilizados.

A interface *IModel* oferece um único método para recuperação dos dados da aplicação não importando onde esses dados estão armazenados. Deste modo, a classe cliente de qualquer implementação de *IModel* não precisa saber onde estão os dados, somente necessita chamar este método. Existem duas implementações de *IModel*: *POIML* e *POIProxy*. A classe *POIML* realiza operações de dados na memória local do dispositivo do usuário, quando esses dados estão disponíveis. Sempre que os dados não estão disponíveis a classe que assume a responsabilidade é a *POIProxy*, esta por sua vez, realiza operações com um servidor na nuvem e recuperando as informações solicitadas, armazena essas informações na memória local do dispositivo do usuário e delega as funções sobre este dado para a classe *POIML*, quando outra solicitação ocorrer para o mesmo dado o trabalho será realizado pela classe *POIML*.

O conjunto das duas classes *POIProxy* e *POIML* e sua interface *IModel* foram implementadas seguindo o padrão de projetos *Proxy* (GAMMA, HELM, *et al.*, 1994), entretanto esse padrão pode ser facilmente expandido para trabalhar com mais níveis de delegação, por exemplo, pode ser adicionada uma classe *POICache* que mantém em cache alguns dados mais utilizados pelo usuário.

A classe *Media* representa as mídias de modo geral e disponibiliza um método único, o *show()*, para reproduzir/exibir a mídia adequadamente. Toda classe que herda de mídia implementa esse método e sabe como reproduzir/exibir a mídia de acordo com seu tipo, sendo assim, podemos ter vários tipos de mídia como, vídeo, texto, áudio, objetos 3D, e outros sem alterar a maneira como é solicitada a sua reprodução ou exibição.

3.4 Diretrizes de Modificações de Funcionalidades da Aplicação

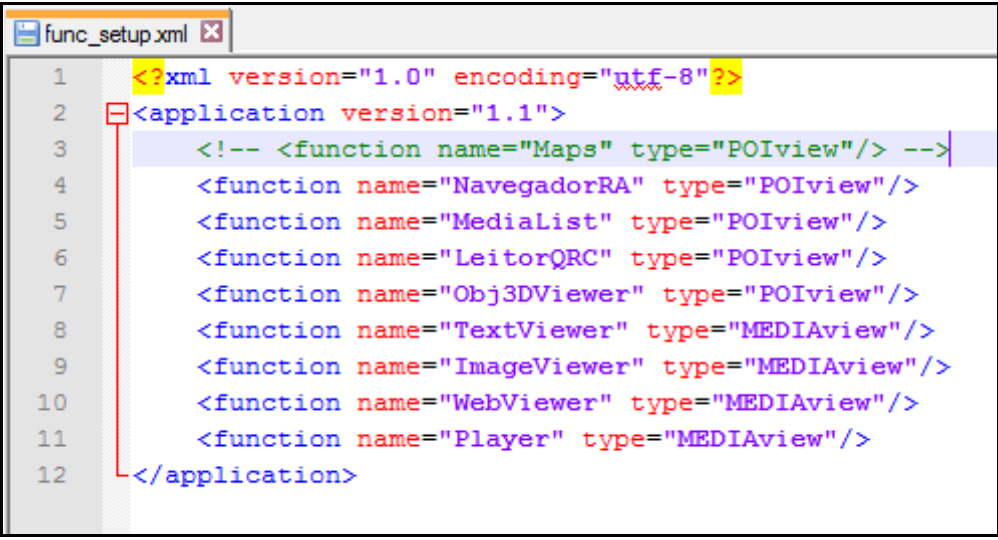
Esta subseção apresenta as diretrizes para realização de modificações e melhorias no código da aplicação ARguide apresentando as principais classes que devem ser alteradas para realizar tais modificações.

Para mostrar como devem ser feitas as alterações foram selecionados dois exemplos fictícios de alterações nas funcionalidades da aplicação, sendo o primeiro a remoção da funcionalidade de Mapa e o segundo a adição da funcionalidade de interação com objetos 3D. Abaixo são apresentados esses dois exemplos.

3.4.1 Remover Funcionalidade de Mapa

Para remover uma funcionalidade do ARguide não é necessária nenhuma alteração em códigos fonte da aplicação, basta realizar uma alteração em um arquivo XML de configuração chamado *func_setup.xml* situado na pasta *assets* do projeto da aplicação. Entretanto, este arquivo de configuração altera quais as classes que serão utilizadas na aplicação em tempo de execução, o que implica que a classe responsável pela funcionalidade mapa ainda vai existir, só não vai ser mostrada para o usuário ao executar a aplicação.

A Figura 20 mostra um exemplo de arquivo de configuração que pode ser utilizado pela aplicação para declarar quais as funcionalidades devem ser consideradas ao executar a aplicação e assim, mostrar essas funcionalidades para o usuário. Nesta imagem, a linha 3 que declara a utilização da funcionalidade de Mapa está comentada, o que significa que a aplicação deve desconsiderar essa funcionalidade, não mostrando a aba de mapa na aplicação.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <application version="1.1">
3 <!-- <function name="Maps" type="POIview"/> -->
4 <function name="NavegadorRA" type="POIview"/>
5 <function name="MediaList" type="POIview"/>
6 <function name="LeitorQRC" type="POIview"/>
7 <function name="Obj3DViewer" type="POIview"/>
8 <function name="TextViewer" type="MEDIAview"/>
9 <function name="ImageViewer" type="MEDIAview"/>
10 <function name="WebView" type="MEDIAview"/>
11 <function name="Player" type="MEDIAview"/>
12 </application>
```

Figura 20. Conteúdo do arquivo *func_setup.xml* para exemplificar a modificação de funcionalidades no ARguide.

O funcionamento por trás deste arquivo de configuração está na classe *Activity* que contém as Visões da aplicação. Esta *Activity* realiza a leitura do arquivo de configuração e instancia as Visões que estão declaradas no arquivo, mantendo essas instâncias em uma lista. A Figura 21 mostra um diagrama deste funcionamento, onde a classe *Activity* lê o arquivo XML que possui a lista das Visões que devem ser apresentadas ao usuário, instancia estas classes e as mantém em uma lista. As visões estão dentro da caixa *POIview* representando a abstração das funcionalidades em uma interface.

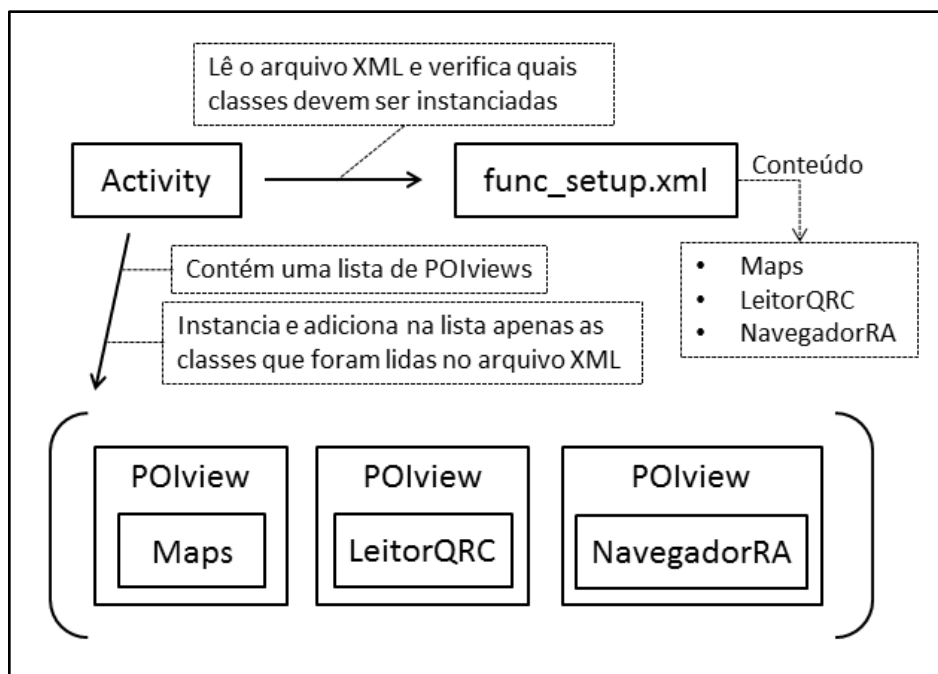


Figura 21. Esquema do funcionamento da configuração de funcionalidades através do arquivo *func_setup.xml*.

3.4.2 Adicionar Funcionalidade de Interação com Objetos 3D

Para adicionar uma nova funcionalidade no ARguide é necessário criar uma classe que implemente a interface *POIview* e declarar esta classe como funcionalidade no arquivo *func_setup.xml*. A classe criada deve conter toda a lógica da funcionalidade desejada e pode utilizar os dados que recebe da classe *Coordenador* por meio dos métodos declarados na interface *POIview*.

Para declarar a funcionalidade no arquivo *func_setup.xml* basta adicionar uma linha com o elemento *function* e escrever o nome da classe e seu tipo. A Figura 20 mostra, na linha 7 do conteúdo do arquivo, a declaração da classe *Obj3DViewer*.

A Figura 22 mostra a classe *Obj3DViewer* implementando a interface *POIview* (como exemplo de criação de uma nova classe). Da mesma maneira que as outras Visões, toda a lógica desta funcionalidade deve ser encapsulada na própria classe. Neste caso, esta classe pode desenhar um objeto 3D na cena real sem se preocupar com o que está sendo processado em outras classes. Existem dois métodos que devem ser implementados pela nova classe que são: *POISelected()* e *setPOIList()*, que são encarregados de realizar a lógica de seleção de um POI (recebido por parâmetro) na Visão e de atualizar a lista de POIs a serem visualizados.

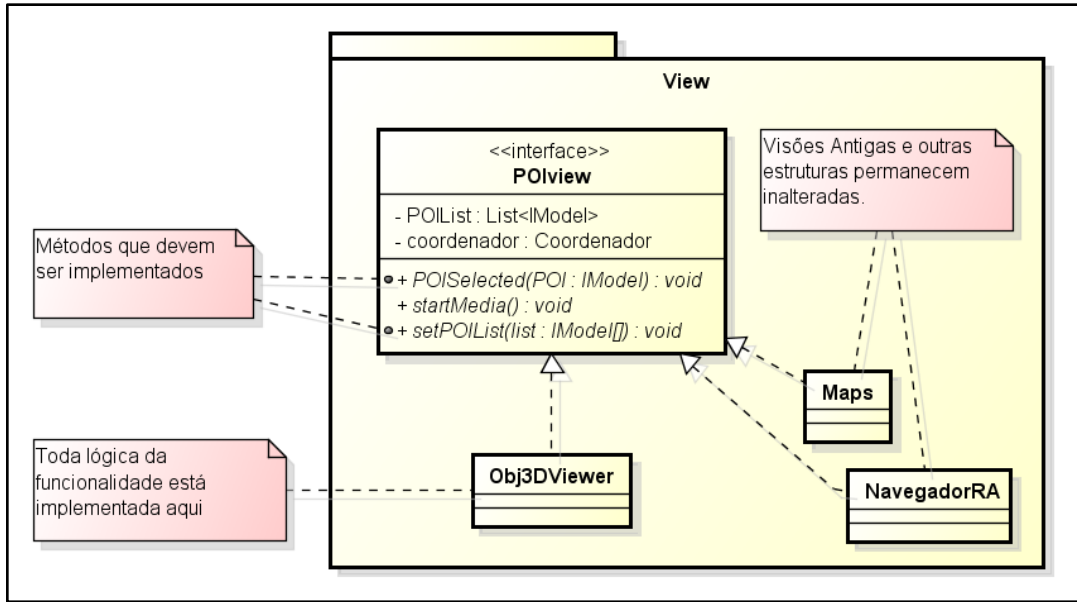


Figura 22. Trecho do diagrama de classes da aplicação ARguide exemplificando a adição de uma nova classe como procedimento para adicionar uma nova funcionalidade.

A priori toda nova funcionalidade será adicionada como uma nova aba na aplicação sem a necessidade de codificar outras classes, entretanto se for necessário adicionar uma funcionalidade em outro componente de interface gráfica, como um botão, será necessário codificar este comportamento na classe *Activity*.

4. PROTÓTIPO DA APLICAÇÃO ARGUIDE

Com a finalidade de testar e validar a arquitetura proposta foi desenvolvido um protótipo funcional do ARguide. Este nome foi dado à aplicação por se tratar de uma aplicação de Realidade Aumentada que foi projetada para utilizar diversas tecnologias disponíveis em dispositivos móveis, tais como sensores, câmera, bússola, GPS, vídeo, reconhecimento de imagem, acesso a Internet, entre outros, proporcionando a descoberta de Pontos de Interesse (POI) em navegador de RA, ou através de rotas/circuitos em mapas 2D. Essa descoberta serve como um guia virtual para o usuário. A interface inicial do ARguide é mostrada na Figura 23.

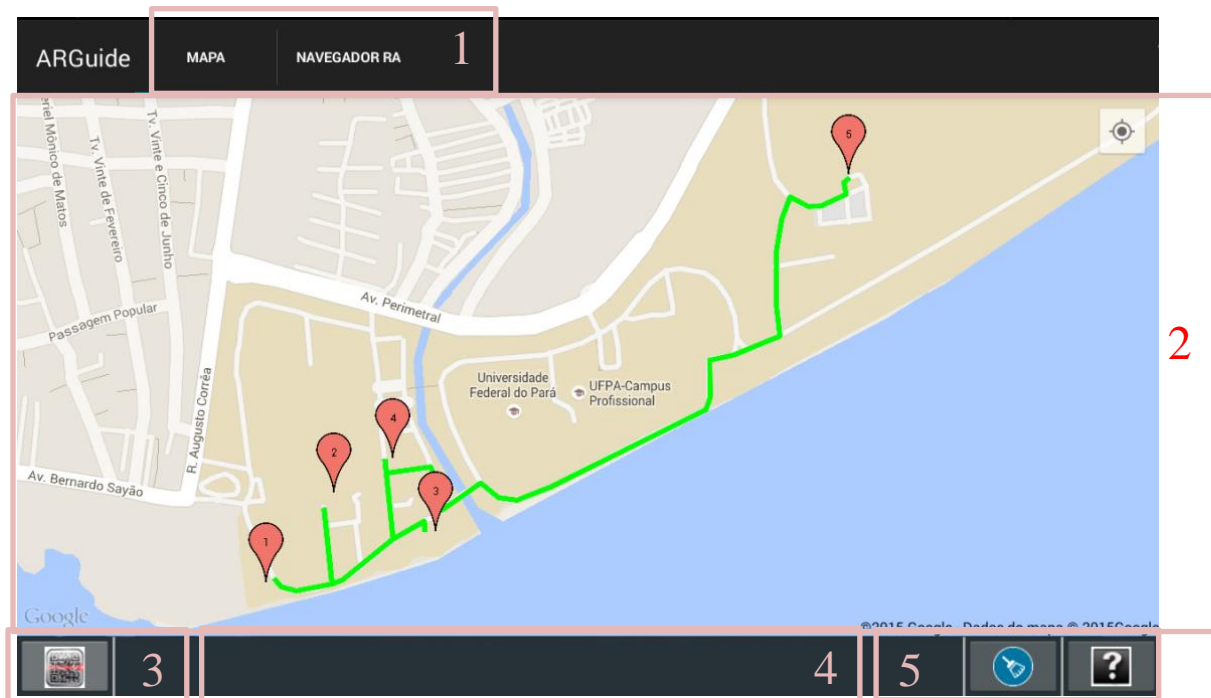


Figura 23. Tela Inicial da aplicação ARguide com áreas destacadas.

A região 1 destaca as abas da aplicação, na qual o usuário pode selecionar qual tipo de visão/interação ele vai utilizar para obter informações sobre os POIs. As interações disponíveis nas abas são o Mapa e o Navegador RA. Além do clique nas abas, o usuário pode alternar entre

elas com interações de *swipe* (passar o dedo na tela, comumente utilizada em aplicações de leitura para “virar uma página”).

A região 2 é ocupada por qualquer Visão selecionada pelo usuário. Atualmente a aplicação possui três visões, então quando o usuário alternar entre essas Visões a região 2 será ocupada pela Visão selecionada.

A região 3 é ocupada pelo botão de *scan* (ícone com laser sobre o QR Code). Este inicia a função de capturar a imagem da câmera e tentar reconhecer um marcador específico, o que é feito através de um recurso do sistema operacional para delegar esta função a outra aplicação e obter apenas sua resposta.

A região 4 é a barra de mídias. Ela exibe todas as mídias disponíveis para um POI selecionado em qualquer Visão da aplicação. A partir da barra de mídias é possível mostrar o conteúdo de uma mídia, reproduzindo um vídeo, exibindo uma imagem, reproduzindo um som, abrindo uma página web no navegador, etc.

Por fim, a região 5 apresenta dois botões que podem ser acessados a qualquer momento. O primeiro botão “limpar” (ícone da vassoura), desativa o POI selecionado, se houver, removendo qualquer mídia presente na barra de mídias e informando todas as abas que a função de “limpar” deve ser executada. O último botão, “ajuda” (ícone do ponto de interrogação) mostra uma tela sobreposta à interface da aplicação que descreve as funcionalidade da mesma e como tais funcionalidades podem ser acessadas.

O ARguide possui resumidamente três tipos de interações, que são: interações no Mapa 2D com acesso às ruas e pontos de referência da cidade em que a aplicação está rodando, Navegador RA que mostra marcadores virtuais nas direções reais dos POIs, e QR Code que realiza a leitura de códigos QR e mostra e seleciona mídias ou POIs que estão próximos ao usuário.

Com a arquitetura proposta torna-se menos custoso adicionar ou retirar uma funcionalidade da aplicação, assim como, trocar os dados da aplicação de acordo com o contexto em que se deseja utilizá-la, tornando viável a mesma aplicação ser útil para múltiplos propósitos. A troca de dados pode ser vista nos teste de cenários de uso, apresentado na seção 5.

Como as funcionalidades estão associadas à camada Visão, basta remover ou adicionar uma nova Visão com a sua própria lógica de interações para ter uma nova funcionalidade. Neste

protótipo da aplicação são necessárias algumas habilidades de programação (como implementar interfaces e codificar as lógicas de interação com o usuário) para remover e adicionar funcionalidades novas na aplicação, entretanto esta tarefa foi simplificada com a arquitetura proposta.

As subseções abaixo descrevem cada uma dessas funcionalidades e o que o usuário pode realizar por meio dessas funcionalidades.

4.1 Mapa

A aba Mapa contém um mapa desenvolvido com a tecnologia Google Maps, utilizando a versão 2 da API (GOOGLE INC., 2015), que permite aos usuários a realização de interações de *zoom*, *pan* centralização na posição do usuário, entre outras que viabilizam a visualização, busca e seleção de POIs em um mapa 2D.

A principal vantagem de visualizar os POIs em um mapa é que eles posicionados no mapa da cidade em o usuário se situa, dando ao usuário referências reais para que o mesmo tenha a noção da localização desses POIs. No mapa os POIs são colocados como marcadores na sua posição geográfica e permite que o usuário toque nesses marcadores para selecionar um POI de interesse. Ao selecionar um POI na aba mapa força a seleção do mesmo POI nas outras Visões da aplicação.

A Figura 24 mostra duas capturas de tela da aplicação ARguide com a aba Mapa ativa na tela de um *tablet* (a) e na tela de uma *smartphone* (b). Ao centro da aplicação é mostrado o mapa da cidade em que o usuário se encontra e os marcadores numerados indicando a localização dos POIs. Ao iniciar a aba mapa a aplicação sugere uma rota de visitação dos POIs destacando as ruas a serem utilizadas de verde.

A seguir estão descritas as possíveis funcionalidades que o usuário pode realizar a partir da aba mapa resumida em forma de lista.

- **Zoom e Pan:** são operações de ampliar e mover o mapa na tela do dispositivo, estas operações permitem que o usuário controle o nível de detalhes em relação à visão geral do que eles pretendem visualizar no mapa. Para ampliar e diminuir o mapa o usuário deve fazer o movimento de pinça na tela do seu dispositivo. Estas interações já estão

configuradas por padrão na API do Google Maps. Ao ampliar o mapa são apresentados cada vez mais detalhes sobre uma região, o que é conhecido como *zoom* semântico.

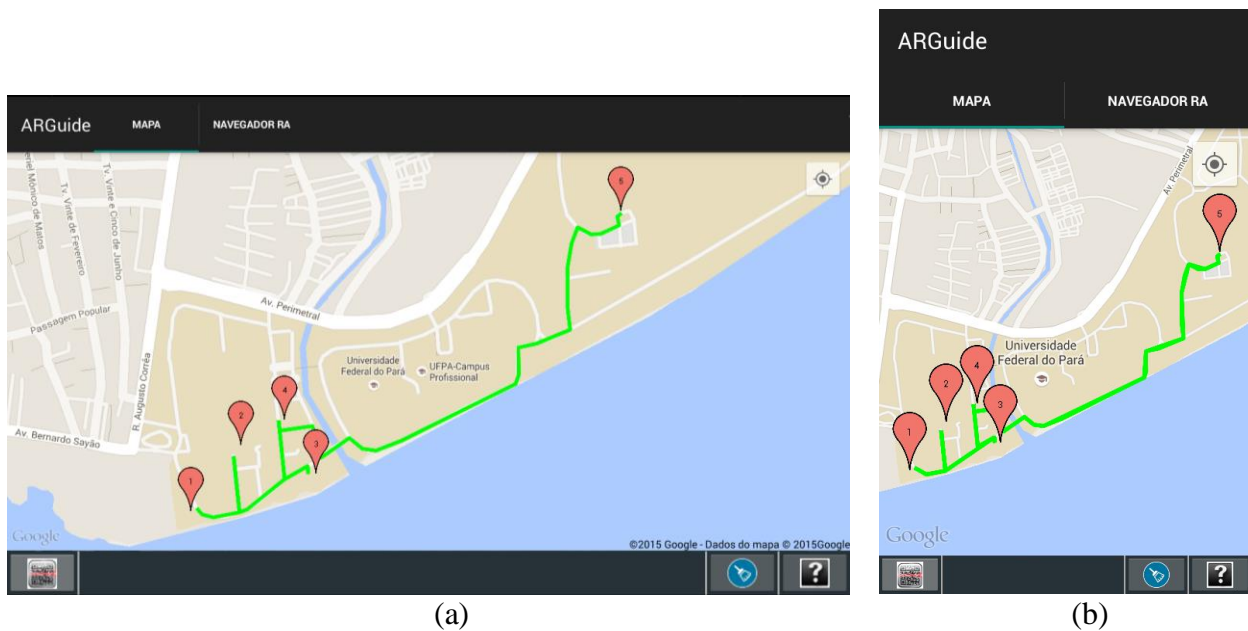


Figura 24. Protótipo da aplicação mostrando a Visão Navegador RA em um *tablet* (a) e em um *smartphone* (b)

- **Traçar Rota:** esta operação permite que o usuário visualize uma rota que parte da sua posição atual para o POI que o usuário seleciona. A rota é traçada até o POI toda vez que o usuário seleciona um POI, ou seja, toca no marcador do mapa ou seleciona em outra Visão da aplicação. A rota é mostrada ao usuário destacando as ruas que o usuário deve percorrer para chegar até o POI selecionado com uma cor diferente da cor das ruas no mapa.
- **Selecionar POI:** esta operação permite que o usuário selecione um POI para que ele possa realizar outras operações em um POI que o usuário deseje. Uma vez que os pontos de interesse são posicionados no mapa em forma de marcadores clicáveis, o usuário pode clicar/tocar em um desses marcadores para selecionar o POI em questão. Quando um POI é selecionado é apresentado ao usuário um conjunto de conteúdos multimídia sobre esse POI ao usuário, além disso, quando esta seleção ocorre todas as outras visões da aplicação são notificadas pelo coordenador e também realização a operação de seleção deste POI.
- **Traçar circuito de visitação:** esta operação permite que o usuário trace um caminho único de visitação para todos os POIs que estão no raio de interesse do usuário. Neste o

usuário pode acompanhar sua trajetória em relação ao caminho percorrido e os pontos que já foram visitados. Um circuito é sempre sugerido para o usuário automaticamente, mas o usuário pode trocar esse circuito ou pode selecionar um circuito pré-cadastrado na aplicação.

4.2 Navegador de Realidade Aumentada

A aba com a Visão Navegador RA apresenta ao usuário o *stream* de sua câmera (vídeo), representando o mundo real, acrescido de marcadores virtuais que são posicionados na direção dos POIs em relação ao usuário, marcando a direção real do POI quando o usuário aponta sua câmera para este. Com essa marcação o usuário tem sua percepção aumentada em relação a esse POI, pois sabe quais POIs possuem conteúdo na aplicação e qual a direção exata de cada ponto de interesse.

A posição de desenho de cada POI é obtida a partir do GPS e sensores de movimento presentes no dispositivo móvel do usuário. A Visão Navegador RA calcula a posição do POI em relação ao ângulo tridimensional em que o dispositivo está e desenha sobre a imagem do mundo real, obtida pela câmera do dispositivo, pequenos ícones representando cada ponto de interesse nas proximidades.

Esta é uma abordagem semelhante à *Magic Lens*, descrita por Cawood e Fiala (2008), onde a posição de cada POI é dada pela informação de geolocalização do mesmo. Dessa forma, os pontos de interesse ficam “flutuando” na tela do dispositivo e o usuário pode navegar entre eles se locomovendo ou mudando a direção para onde sua câmera está apontada, semelhante ao encontrado em outros navegadores de RA mais conhecidos, como o Wikitude (WIKITUDE INC., 2015).

Para ajudar o usuário a encontrar ou notar a existência de POIs que não estejam visíveis na direção que o usuário está apontando a câmera do seu dispositivo, a visão Navegador de RA possui um pequeno radar no canto superior esquerdo, que mostra todos os POIs no raio de alcance definido pelo usuário como pequenos pontos coloridos. Dessa forma, mesmo que o POI não esteja na direção apontada pela câmera o usuário tem a percepção que o POI existe, assim como, tem noção de que movimento fazer para que este POI torne-se visível. Esse radar representa um ângulo de 45 graus com centro na direção para onde a câmera do usuário está apontando. Os pontos dentro desse ângulo aparecem visíveis na tela como marcadores flutuantes

e como pontos no radar, os pontos dentro do raio de interesse e fora do ângulo de visão são apenas mostrados no radar, como mostrado na Figura 25.

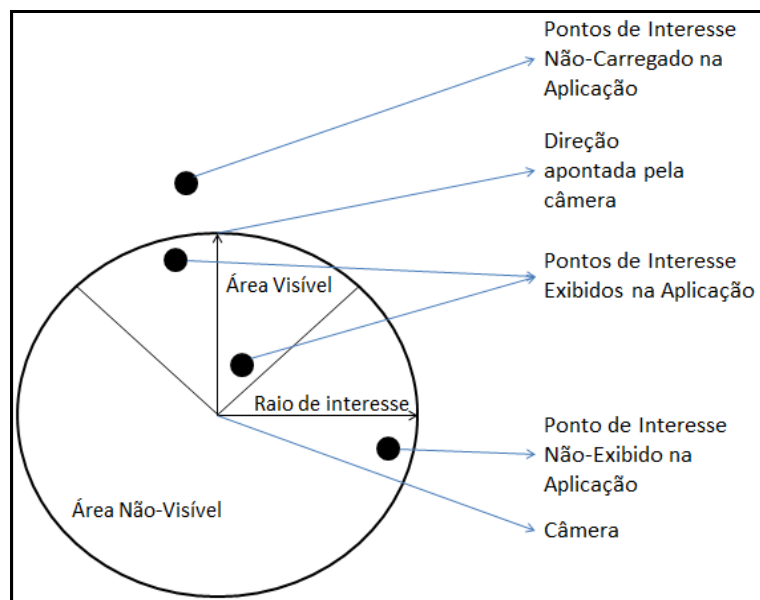


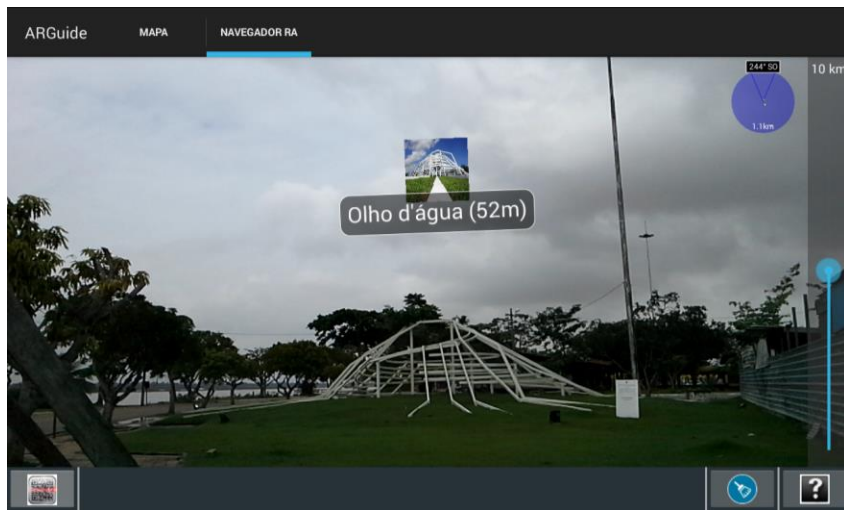
Figura 25. Esquema do funcionamento do radar na funcionalidade Navegador RA.

O mini radar é ajustado por um componente do tipo *slider* que redefine a área em que os POIs devem aparecer na tela de acordo com a distância entre os POIs e a posição atual do usuário. Esta abrangência é chamada de raio de interesse e pode variar de 4 metros até 10 quilômetros de raio.

A Figura 26 mostra duas capturas de tela da aplicação ARguide com a aba Navegador RA ativa na tela de um *tablet* (a) e na tela de um *smartphone* (b). Ao centro da aplicação é mostrado ao usuário o *stream* de vídeo da câmera do dispositivo sobreposta pelos marcadores virtuais que indicam a direção do POI em relação ao usuário. À direita o usuário pode controlar o raio de interesse por meio de um *slider*.

A seguir estão descritas as possíveis funcionalidades que o usuário pode realizar a partir da Visão de Navegador de RA:

- **Visualizar Posição/Direção do POI:** movimentação que o usuário faz com o seu dispositivo móvel e recebe como *feedback* o *stream* de vídeo de sua câmera com o marcadores virtuais posicionados na direção real dos POIs. Esta permite ao usuário encontrar POIs baseado na localização atual do usuário, obtida a partir do GPS do dispositivo móvel, ao direcionar sua câmera para os POIs disponíveis na aplicação.



(a)



(b)

Figura 26. Protótipo da aplicação mostrando a Visão Navegador RA em um *tablet* (a) e em um *smartphone* (b)

- **Selecionar POI:** clique/toque nos ícones virtuais que flutuam na tela realizando assim uma seleção do POI. Ao selecionar um POI o usuário pode ter acesso aos conteúdos multimídia relacionados a este POI e em todas as outras Visões este POI estará selecionado também. Para ser capaz de selecionar, ou seja, clicar em um marcador virtual, o usuário deve estar apontando a câmera do seu dispositivo na direção deste POI, fazendo com que o marcador virtual seja desenhado na tela.
- **Ajustar Raio Alcance:** realizada por meio de um *slider* que o usuário pode tocar e arrastar aumentando ou diminuindo o raio proporcionalmente. Esta permite que o usuário configure um raio de interesse para a exibição dos ícones virtuais na tela, assim os POIs que estiverem além da distância máxima configurada não será desenhado na tela. Este raio é medido em quilômetros e tem como centro a posição atual do usuário. Esta operação pode ser utilizada para que o usuário possa se concentrar na busca por elementos mais próximos a ele e mitiga o problema de desenhos muitos elementos na mesma porção de tela.
- **Visualizar Radar:** visualizar e notar os POIs no radar que se movimentam de acordo com o movimento de rotação da câmera em relação ao eixo vertical. O radar mostra um ponto para cada POI localizado dentro do raio de interesse configurado. Isto permite que POIs que não estejam sendo visualizados, por não estarem na direção da câmera do usuário, possam ser notados. No radar é marcado um ângulo de visualização dos POIs,

assim todos os pontos que estiverem dentro deste ângulo estão sendo desenhados na tela, e os pontos que não estão contidos no ângulo não estão desenhados na tela e conseqüentemente não estão na direção da câmera do usuário. Dessa forma, o usuário consegue induzir qual movimento de rotação deve realizar para fazer com que os POIs estejam visíveis na tela e ao mesmo tempo descubrem a posição real do POI.

4.3 Leitor de QR Code

O leitor de QR Code busca e executa um serviço de leitura de códigos QR Code disponível no dispositivo do usuário. Este serviço realiza a leitura dos códigos extraído as informações contidas em tais marcadores. Ao receber as informações extraídas do código QR Code a aplicação verifica se este código está pré-cadastrado na aplicação (essa busca é feita tanto na memória local quanto na nuvem) e assim pode disponibilizar o acesso à lista de informações do POI ou mídia referente ao código extraído.

O leitor de QR Code permite ao usuário interagir com POIs que estão muito próximos, tal que o erro do GPS não consiga distinguir um do outro. Esse geralmente é o caso de ambiente Indoor, onde os POIs estão dentro de um edifício que pode ter andares. Um exemplo desta é um museu que deseja disponibilizar informações sobre seus quadros para os visitantes, assim tanto quadros muito próximo como os que estão em andares diferentes não serão facilmente distinguidos pelo GPS, logo tanto a visão mapa quanto o Navegador RA podem errar a posição destes quadros e confundir o usuário. Ao extrair um código de um QR Code próximo ao quadro desejado evita esse tipo de problema.

A Figura 27 mostra a captura de tela de uma aplicação externa realizando a leitura de código QR Code. Esta aplicação externa é chamada a partir do botão de leitura de QR Code localizado na barra inferior do ARguide (Figura 23 – área 3). A leitura de código é realizada através de uma aplicação externa ao ARguide que já deve estar instalada no dispositivo do usuário. A captura de tela mostra esta aplicação realizando a leitura que ao finalizar é enviada para a aplicação ARguide que decide o que fazer com o código extraído.

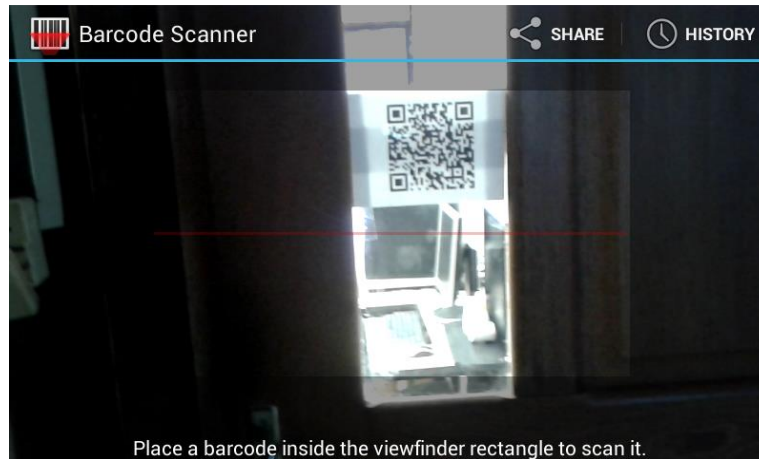


Figura 27. Aplicação externa realizando a leitura de um código QR Code, para posteriormente enviar ao ARguide.

4.4 Funcionalidades Gerais

As funcionalidades gerais da aplicação estão sempre disponíveis para o usuário independente de qual visão da aplicação esteja sendo utilizada. Estas funcionalidades tem o objetivo de auxiliar o usuário em atividades de propósito geral e em momentos de dúvida por parte do usuário. As funcionalidades gerais estão descritas abaixo:

- **Selecionar item Multimídia:** esta operação pode ser realizada pelo usuário quando o mesmo clica nos botões de mídias de um POI selecionado, como mostra a Figura 28.
- **Interagir com Conteúdo Multimídia:** esta operação está sempre disponível para o usuário quando ele seleciona um POI em qualquer uma das abas e então seleciona uma mídia que deseje visualizar. Esta permite que o usuário interaja com dados multimídia relacionados com o POI selecionado, por exemplo, vídeos, áudios, textos, imagens, e website. Estes dados multimídia ajudam os usuários a conhecerem mais sobre o POI durante, antes ou depois da visitação.
- **Limpar Seleção:** esta operação é realizada através de um botão que ao ser clicado/tocado pelo usuário limpa a seleção previamente realizada pelo usuário. Esta opção não é obrigatória para realizar outra seleção, mas remove da tela elementos que podem não interessar mais para o usuário, como o balão de detalhes no mapa e as mídias listadas na barra de mídias.

- **Ajuda:** esta operação é realizada através de um botão que ao ser clicado/tocado pelo usuário oferece dicas e guias de utilização da aplicação. A ajuda oferece uma descrição da utilização da aplicação e suas funcionalidades de forma geral.

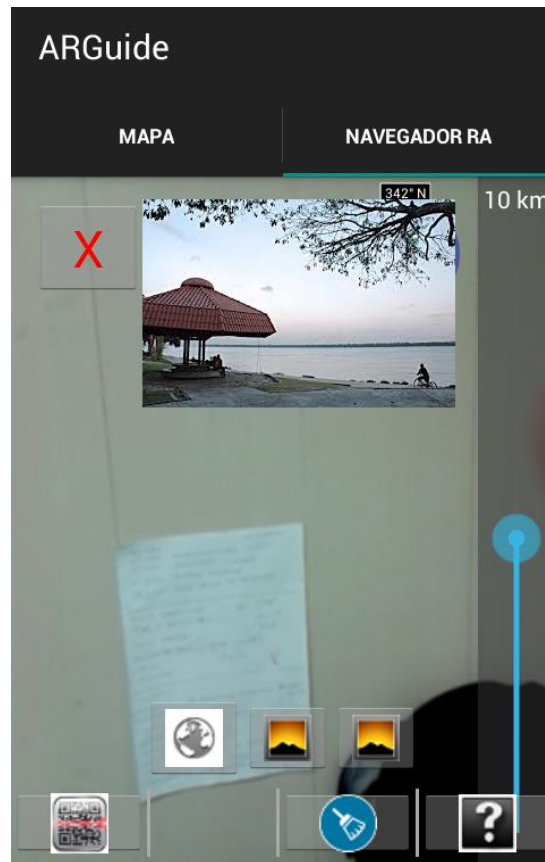


Figura 28. Captura de tela da aplicação visualizando uma mídia de imagem em um *smartphone*.

5. TESTE EM CENÁRIOS DE USO

Os cenários de uso desta aplicação foram configurados para testar a utilização de diferentes tipos de dados, assim como, diferentes tamanhos de telas com a finalidade de testar as características da arquitetura proposta.

Foram realizados três testes para os dados e dois testes para os tamanhos de tela, sendo que esses foram separados em cenários de uso. Foram configurados três cenários, são eles: cenário turístico, artístico, e educativo.

Cada um desses cenários usou seus respectivos dados, sendo, turísticos, artísticos e educativos, representado os testes de dados e dois dispositivos com o sistema operacional Android, um *tablet* e um *smartphone*, representando os testes de tamanhos de tela. Em todos os cenários foram utilizadas informações públicas.

Com a utilização de dados que variam de acordo com o cenário apresentado, o objetivo esperado é comprovar que a aplicação funciona independente do conjunto de dados carregado, ou seja, esta não é restrita à apenas um tipo de cenário ou configuração.

O *tablet* utilizado nos testes é um Samsung Galaxy Tab 3 com as seguintes configurações: processador Dual-Core 1,5GHz, 1,5GB de memória RAM, e tela de 8 polegadas. E o *smartphone* utilizado nos testes foi um Samsung Young Duos TV com as seguintes configurações: processador de 1 GHz, 750 MB de memória RAM e tela de 3,2 polegadas.

Os cenários de uso da aplicação estão descritos nas subseções a seguir.

5.1 Cenário Artístico

O cenário artístico possui dados de obras artísticas da Universidade Federal do Pará. Com este cenário um usuário pode explorar e visitar as obras artísticas espalhadas pela universidade conhecendo suas localizações e mais sobre sua história, significados e autores.

Os possíveis *stakeholders*¹ envolvidos neste cenário são: como fornecedor dos dados, organizações interessadas em divulgar as obras e artistas de uma determinada região, dando acesso aos significados, descrições e contexto histórico das obras; e como usuário, entusiastas, alunos de artes e pessoas interessadas em conhecer e visitar as obras artísticas em um determinado contexto.

A Figura 29 (a) mostra um exemplo típico da utilização da aplicação em um cenário artístico. Neste o usuário está apontando seu *tablet* para a obra artística Olho d'Água e está visualizando um marcador virtual na direção da localização exata da obra. Ao clicar neste marcador, o usuário pode ter acesso às mídias desta obra, como: textos explicativos, contexto histórico, resumo do autor, *website* da obra, etc.

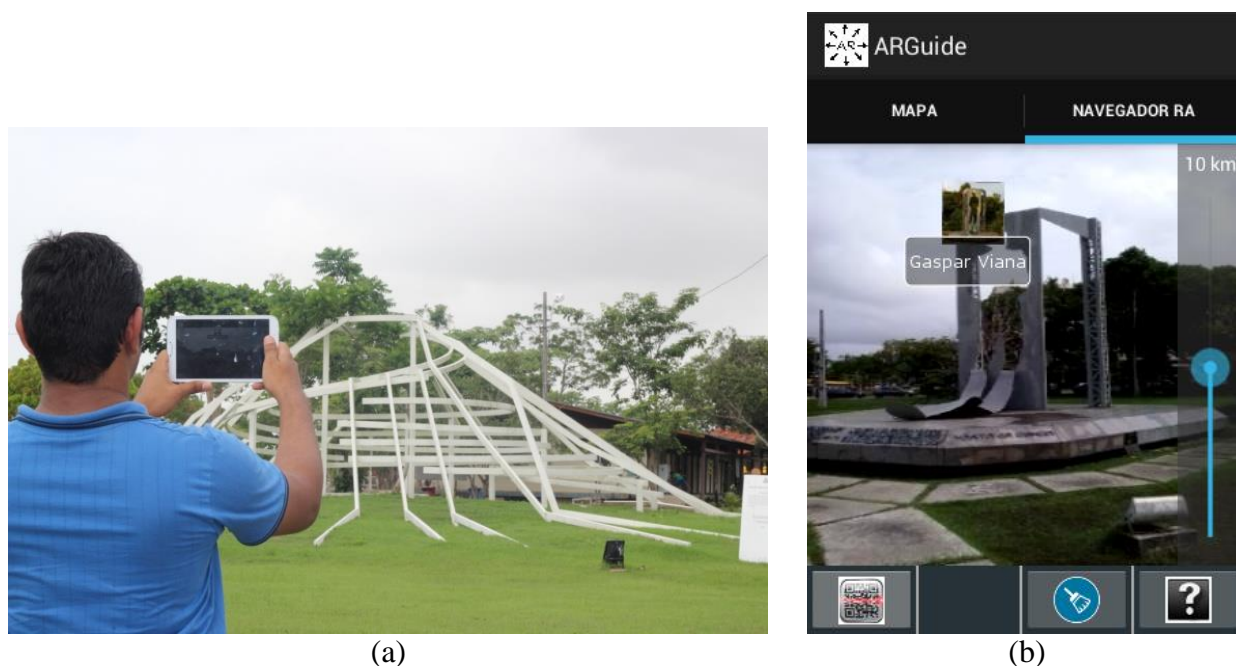


Figura 29. Cenário de uso artístico. Fotografia da utilização da aplicação em um *tablet* (a) e captura de tela em um *smartphone* (b).

A Figura 29 (b) mostra a captura de tela da aplicação no cenário artístico, desta vez em um *smartphone*. Com a captura de tela é possível ver com mais detalhes o marcador virtual na direção da obra artística. Neste é possível observar a utilização do navegador RA em um dispositivo com um tamanho de tela bem menor em relação ao *tablet* (pode ser visto com mais detalhes na Figura 26 – página 60).

¹ *Stakeholders* são as pessoas interessadas ou envolvidas com a aplicação.

Na Figura 29 (b) é percebido que o radar não aparece no canto direito dos *smartphones*. Este e outros *bugs* foram corrigidos para a avaliação com usuários (seção 6).

5.2 Cenário Educativo

O cenário educativo possui dados de informativos sobre plantas e animais, e obras do Museu Emílio Goeldi, situado em Belém do Pará, Brasil. Com este cenário imagina-se uma visita educativa ao museu com acesso aos conteúdos multimídias de animais, plantas, e obras estimulando o aprendizado e o interesse por esse tipo de conteúdo.

Os possíveis *stakeholders* envolvidos neste cenário são: como fornecedor dos dados, organizações educacionais com interesse em disponibilizar uma ferramenta de auxílio ao aprendizado e ao interesse pelo conteúdo ensinado; e como usuários, alunos e professores que desejam aprender e ensinar de maneira prática através da visita aumentada de bosques, museus e zoológicos.

A Figura 30 (a) mostra uma foto da utilização da aplicação em um cenário educacional com a utilização do *smartphone* e usando o leitor de QR Code para saber mais informações sobre a Arara-Azul no museu Emílio Goeldi. A ideia desta foto é mostrar como a aplicação pode ser utilizada em excursões educativas em zoológicos, museus, planetários, bosques, entre outros.



(a)



(b)

Figura 30. Cenário de uso educativo. Fotografia da utilização da aplicação em *smartphone* (a) e *tablet* (b).

A Figura 30 (b) apresenta a visualização de conteúdo (neste caso texto, mas poderiam ser fotos, vídeos, áudios e websites) logo após a seleção de um POI, que neste caso ocorreu através da leitura de um código QR Code, mas esta seleção poderia ocorrer pelo toque em um ícone virtual no Navegador RA ou toque em um marcador no mapa.

5.3 Cenário Turístico

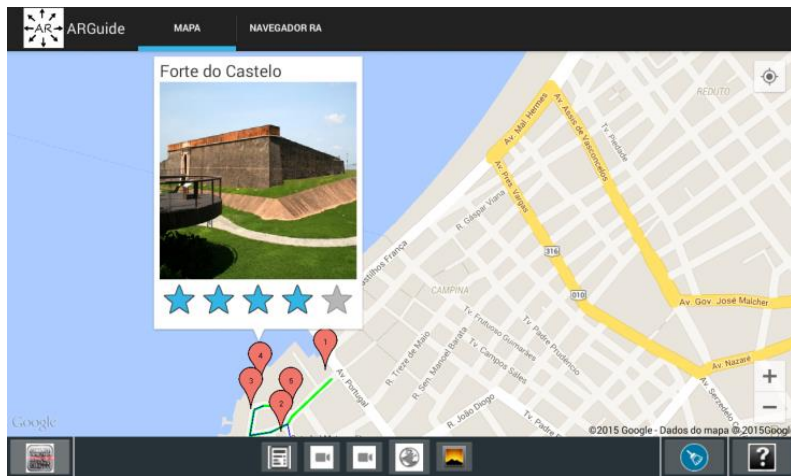
O cenário turístico possui dados relacionados com os pontos turísticos da cidade de Belém do Pará, Brasil. Com este cenário um usuário pode conhecer e visitar os pontos turísticos da cidade com acesso a roteiros turísticos aumentados.

Os possíveis *stakeholders* envolvidos neste cenário são: como fornecedor dos dados, organizações interessadas em promover o turismo da cidade de forma mais interativa e tecnológica, dando um acesso fácil, rápido e aumentado para os pontos turísticos de uma cidade; e como usuário, turistas interessados em visitar a cidade e ao mesmo tempo conhecer mais sobre a história e obras arquitetônicas.

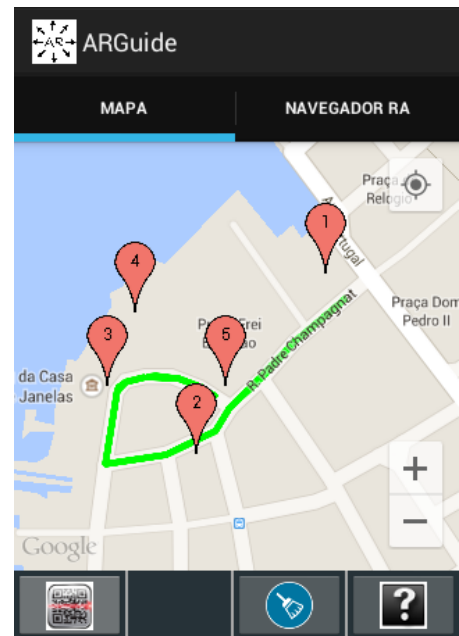
A Figura 31 (a) e (b) mostram a utilização da aplicação em um cenário turístico com o recurso de mapas 2D em *tablet* e *smartphone*, respectivamente.

Na Figura 31 (a) o usuário selecionou um POI (neste caso uma atração turística) tocando em um marcador no mapa, e lhe foi apresentada uma imagem do local, uma nota de 0 a 5 (estrelas azuis na imagem), e o nome do local. Também foi apresentada uma rota que pode ser visualizada pela linha verde e é ordenada através de numerações nos marcadores. A Figura 31 (b) mostra a mesma rota, porém em um dispositivo com tela menor.

Embora cada um dos cenários tenha mostrado funcionalidades distintas é possível utilizar todas as funcionalidades em cada um desses cenários, assim como em outros cenários, dependendo apenas do conteúdo que será utilizado. Além disso, é possível adicionar e remover funcionalidades para melhor adequar a aplicação às necessidades do cenário de utilização. Por exemplo, em uma visita a um museu com muitos andares, pode ser útil incluir uma funcionalidade de reconhecimento de imagem, enquanto que a utilização do Navegador RA será prejudicada devido a ausência de rastreamento dos andares e a perda a precisão do GPS, dando a impressão que todos os POIs estão no mesmo andar.



(a)



(b)

Figura 31. Cenário de uso turístico. Captura de tela da aplicação em *tablet* (a) e *smartphone* (b).

6. AVALIAÇÃO DE USABILIDADE DO ARGUIDE

A avaliação de usabilidade do ARguide foi feita a partir de métodos de avaliação divididos em duas categorias: métodos de inspeção e métodos de teste de acordo com a taxonomia proposta por Mahrin e Colaboradores. (2009).

Os métodos de inspeção são relatos de um ou mais especialistas sobre uma versão final ou protótipo (versão beta) da aplicação. Os métodos de inspeção demandam menos tempo para serem realizados (se comparado com as avaliações com usuários) e podem descobrir vários tipos de problemas na interface da aplicação.

Os métodos de teste ou métodos observacionais envolvem usuários testando a versão final ou protótipo (versão beta) da aplicação. Estes testes geralmente consomem mais tempo para preparação, realização e análise dos resultados. Entretanto, este tipo de teste é mais próximo dos usuários reais e podem avaliar se os usuários conseguem utilizar a interface da aplicação de forma satisfatória.

A Tabela 2 foi proposta por Mahrin e Colaboradores (2009) e mostra os métodos de avaliação de usabilidade medidos por treze fatores para cada um desses métodos. Este resultado foi utilizado como um dos critérios de seleção dos métodos utilizados para a avaliação de usabilidade do ARguide.

Martínez e Bandyopadhyay (2014) realizaram uma análise de quatro métodos de avaliação de usabilidade em uma aplicação de RAM e concluíram que a utilização de mais de um tipo método de avaliação de usabilidade geram melhores resultados para a avaliação de forma geral, já que os resultados obtidos por cada tipo de avaliação se complementam gerando uma avaliação mais abrangente e completa. Assim, decidiu-se utilizar mais de um tipo de avaliação de usabilidade para o ARguide.

Como métodos de inspeção foram escolhidos o percurso cognitivo (WHARTON, RIEMAN, *et al.*, 1994) e a avaliação heurística (NIELSEN e MOLICH, 1990), pois estes métodos tiveram boas avaliações de acordo com a Tabela 2, foram testados em aplicações RAM e geraram bons resultados no trabalho de Martínez e Bandyopadhyay (2014).

Tabela 2. Métodos de avaliação e seus fatores de usabilidade. Legenda: P - Pouco, M - Médio, A - Alto. Adaptado de Mahrin e Colaboradores (2009)

Categoria	Método de Avaliação	Fatores de Usabilidade												
		Compreensível	Fácil de Aprender	Bem Estruturado	Bem Apresentável	Bem Suportado	Possui Ferramenta	Completo	Conciso	Não Ambíguo	Atualizado	Operável	Navegável	Personalizável
Inspeção de Usabilidade	Heurística	P	P	M	M	M	M	P	P	P	M	P	M	P
	Percurso Cognitivo	M	A	A	A	A	M	P	M	P	P	P	M	P
	Percurso Pluralístico	P	P	M	M	M	P	P	P	P	P	P	P	P
	Inspeções de Características	P	P	P	P	P	A	P	P	P	P	P	A	P
	<i>Checklist</i> de Diretrizes	P	P	A	A	A	P	P	P	P	P	P	M	P
	Inspeção por Perspectivas	P	P	A	A	A	A	M	M	P	M	P	M	P
Teste de Usabilidade	Co-Descoberta	A	A	P	P	M	M	P	M	A	P	A	A	P
	Protocolo Pergunta-Resposta	A	A	M	M	M	M	M	M	A	P	A	A	M
	Protocolo Pensa Alto	A	M	M	M	M	M	P	M	M	P	A	A	P
	Medição de Desempenho	M	M	P	M	M	P	P	P	P	P	M	A	P
	Registro de logs do Usuário	P	P	P	P	P	P	P	P	P	P	M	P	P
	Observação de Campo	P	P	P	P	P	P	P	P	P	P	M	P	M
Relato de Usuário	Questionário	M	M	M	M	M	M	M	M	M	M	M	M	M
	Entrevista	M	M	M	M	M	M	M	M	M	M	M	M	M

Como métodos de teste foram escolhidos a medição de desempenho, o registro de *logs* do usuário e o protocolo pensa alto, pois o protocolo pensa alto foi bem avaliados e os demais podem ser realizados em paralelo com os usuários, uma vez que os testes com os usuários preferencialmente não devem tomar muito tempo dos entrevistados.

As subseções abaixo explicam cada um dos métodos escolhidos, como eles foram aplicados para a avaliação de usabilidade da aplicação ARguide e o resultados gerados com essas avaliações.

6.1 Avaliação por Métodos de Inspeção

Os dois métodos escolhidos para a avaliação de usabilidade do ARguide possuem duas visões diferentes, sendo a avaliação heurística mais geral e o percurso cognitivo mais específico. Assim, essas duas avaliações tendem a se complementar.

Na avaliação heurística o especialista avalia as telas da aplicação sem uma ordem específica e registra todas as características que ele considera problemática ou que necessite de ajuste. Para julgar a interface gráfica o avaliador segue princípios de usabilidade (definidos antes da avaliação) e sempre que achar um ponto fraco ele deve apontar quais desses princípios foram violados. Após gerar uma série de pontos fracos na interface da aplicação podem ser sugeridas algumas medidas para resolver ou mitigar esses problemas.

Na avaliação percurso cognitivo o especialista avalia a interface da aplicação seguindo um roteiro (uma sequência definida de passos) para realizar uma ou mais tarefas dentro da aplicação. A cada passo deste roteiro o avaliador julga quais as dificuldades que o usuário terá ao realizar esta atividade. Após julgar todos os passos, o resultado gerado indicará exatamente em que momento o usuário possivelmente terá dificuldades e qual parte da interface da aplicação está com problema.

Abaixo está descrito como cada uma dessas avaliações foram realizadas no ARguide, destacando como elas foram conduzidas e quais os parâmetros de cada uma, por exemplo, número de avaliadores, quais as condições de teste, entre outros.

Todos os dados utilizados para os testes estão relacionados com os serviços disponíveis na Universidade Federal do Pará. Foram utilizados dois dispositivos diferentes para as avaliações sendo um *tablet* e um *smartphone*. O *tablet* é um Samsung Galaxy Tab III com o processador Dual-Core de 1,5GHz de frequência, 1,5GB de memória RAM e 8 polegadas de tamanho de tela. E o *smartphone* é um Samsung Galaxy SII Lite com Dual-Core de 1 GHz de frequência, 768 MB de memória RAM e 4 polegadas de tamanho de tela.

Os especialistas que participaram das avaliações por método de inspeção são alunos do último e penúltimo semestre do curso de Ciência da Computação e Sistemas de Informação, que já desenvolveram ou participaram do desenvolvimento de um ou mais aplicativos para o sistema operacional Android, envolvendo a construção de interfaces gráficas.

Antes de iniciar as avaliações foi apresentado aos avaliadores um vídeo que explica de forma geral a aplicação e o seu propósito, dando ênfase tanto nas características da aplicação como nas suas funcionalidades, para que os avaliadores tenham um conhecimento prévio da aplicação auxiliando a sua avaliação.

Após as explicações de como o avaliador deve proceder na avaliação, é iniciada a fase de avaliação em si, na qual os avaliadores possuem tempo livre. As fichas de avaliação utilizadas nos testes podem ser vistas no Apêndice A e no Apêndice B, sendo avaliação heurística e percurso cognitivo respectivamente.

6.1.1 Avaliação Heurística de Usabilidade

A avaliação heurística de usabilidade do ARguide foi realizada por duas duplas de especialistas que não conheciam a interface gráfica da aplicação, sendo uma dupla analisando a aplicação no *tablet* e a outra no *smartphone*. Possibilitando comparar os resultados para cada tamanho de tela.

Logo após a apresentação do vídeo sobre a aplicação são mostradas as heurísticas que os avaliadores devem utilizar como princípios a serem seguidos na aplicação e são dadas instruções (explicando verbalmente) de como realizar uma avaliação heurística.

As heurísticas utilizadas na inspeção são uma adaptação das heurísticas de Nielsen (NIELSEN, 1994) para serem utilizadas em aplicações de RAM de navegação e exploração de POIs proposta por Perutka (2014).

A Tabela 3 mostra as dez heurísticas que foram utilizadas na inspeção, sendo que a primeira coluna é uma descrição das heurísticas originais de Nielsen e a segunda coluna é uma descrição das heurísticas adaptadas.

A Tabela 4 e a Tabela 5 mostram os resultados obtidos na avaliação heurística da aplicação sendo executada em *smartphone* e *tablet*, respectivamente. Os problemas de travamento e *bugs* foram desconsiderados na análise.

Na avaliação com *tablet* foram encontrados 14 problemas na aplicação, enquanto que na avaliação com *smartphone* foram encontrados 9 problemas. A heurística mais violada no *tablet* foi a 1, e no *smartphone* foi a 6.

Tabela 3. Princípios de usabilidade adaptados para aplicações RAM utilizadas na avaliação Heurística do ARguide. Tabela adaptada de Perutka (2014).

Nº	Heurística Original	Versão Adaptada
1	Visibilidade dos estados da aplicação	
	O sistema deve sempre manter o usuário informado sobre o que está acontecendo, através de <i>feedbacks</i> adequados em um tempo razoável.	O sistema deve sempre manter o usuário informado sobre o que está acontecendo. Informações sobre a base de dados utilizada e categoria dos pontos de interesse devem sempre estar visíveis, assim como o estado do rastreamento e orientação. Todas as informações devem ser apresentadas através de um <i>feedback</i> apropriado em um tempo razoável.
2	Equilíbrio entre o sistema e o mundo real	
	O sistema deve falar a língua dos usuários, com palavras, frases, e conceitos familiares aos usuários, ao invés de termos técnicos do sistema. Siga as convenções do mundo real, fazendo as informações aparecerem em uma ordem natural e lógica.	A aplicação deve mostrar a orientação e posição corretas do usuário, do mesmo modo que a posição e distância para pontos de interesses visíveis. As unidades de distância devem respeitar as unidades que são utilizadas no local que o usuário está.
3	Controle do usuário e liberdade	
	Usuários escolhem frequentemente funções equivocadas do sistema, e é necessário uma “saída de emergência” indicada de modo claro para sair do estado indesejado sem precisar passar por diálogos extensos. Deve suportar refazer e desfazer.	Permitir ao usuário trocar livremente entre banco de dados e categorias de pontos de interesse. Permitir também a escolha entre navegação com mapas e com vídeo sobreposto. Retornar ao mesmo estado depois de trocar aplicações. Considere retorna ao estado padrão depois de uma longa interrupção.
4	Consistência e Padrões	
	Usuários não devem se preocupar se palavras, situações ou ações diferentes significam a mesma coisa. Siga os padrões da plataforma.	Os controles da interface do usuário devem respeitar as convenções usadas em dispositivos móveis. Todos os símbolos gráficos devem ser claros e intuitivos.
5	Prevenção de erros	
	Em primeiro lugar, melhor que uma boa mensagem de erro é um design cuidadoso que previne os erros aconteçam. Elimine condições que levem ao erro ou verifique essas condições e apresente aos usuários com uma opção de confirmação antes que possam confirmar a ação.	A aplicação deve desenvolvida de modo que previna erros de acontecer. Se não for possível, a aplicação deve ser capaz de se recuperar.

6	Reconhecimento ao invés de lembrança	
	Minimize o uso da memória dos usuários fazendo objetos, ações e opções visíveis. Não force o usuário a lembrar-se da informação de uma parte do diálogo para a outra. Instruções para uso do sistema devem ser visíveis ou facilmente recuperáveis quando apropriado.	Mostrar para o usuário "onde" ele está na aplicação e fazer a navegação através da aplicação simples. As funções mais importantes (como busca e troca entre navegação com mapas e vídeo sobreposto) devem ser sempre acessíveis.
7	Flexibilidade eficiência de uso	
	Aceleradores – invisíveis para o usuário iniciante – devem oferecer aumento da velocidade de interação para o usuário especialista, de modo que o sistema possa atender tanto o usuário experiente e o inexperiente.	Faça as funções/banco de dados favoritos do usuário facilmente acessíveis. Permitir ao usuário configurar atalhos para essas configurações.
8	Estética e <i>design</i> minimalista	
	Diálogos não podem conter informação que é irrelevante ou raramente necessária. Toda unidade de informação extra compete com unidades de informação relevantes e diminui sua visibilidade relativa.	As telas não podem conter informações irrelevantes ou desnecessárias.
9	Ajude o usuário a reconhecer, diagnosticar e corrigir erros	
	Mensagens de erro devem ser expressas em linguagem plana (sem códigos), indicando precisamente o problema e construtivamente sugerir uma solução.	Mensagens de erro devem ser expressas em linguagem plana (sem códigos), indicando precisamente o problema e construtivamente sugerir uma solução.
10	Ajuda e documentação	
	Mesmo que seja melhor que o sistema possa ser usado sem nenhuma documentação, pode se tornar necessária pra prover ajuda e documentação. Qualquer informação deve ser fácil de procurar, focada na tarefa do usuário, listando passos concretos que devem ser feitos e não deve ser muito grande.	Prover alguma ajuda/documentação. Ajuda rápida e as documentações detalhadas devem estar disponíveis.

Dentre os problemas citados na avaliação com o *smartphone* pode-se destacar a ausência de rótulos textuais nos marcadores do mapa, a ausência de legendas para as rotas, a visualização de

mídia sobreposta ao conteúdo, e a dificuldade de operar o filtro de distância através do *slider* lateral. Estes problemas também foram percebidos nos testes com usuários.

Tabela 4. Resultados da avaliação heurística realizada no *smartphone*.

Nº da Heurística	Problema	Possível Solução
6	Os marcadores possuem números ao invés de rótulos, na aba mapa.	Utilizar rótulos para os marcadores.
8,6	Ausência do botão de busca ou campo de busca na aba mapa.	Inserir um botão de busca na aplicação.
8	Na aba mapa, a imagem com <i>rating</i> do marcador está grande demais.	Adicionar a funcionalidade de redimensionar textos e imagens (mídias em geral).
1,6	Não há indicação sobre o significado da rota verde no mapa.	Inserir uma descrição de ajuda ou legenda.
10	Na aba Navegador RA não há indicação do movimento que o usuário deve realizar para visualizar POIs.	Inserir essa indicação na Ajuda.
10	A ajuda disponibilizada não oferece suporte necessário aos usuários.	Adicionar uma ajuda mais apropriada aos usuários.
8	Na aba Mapa, a visualização de mídias ocupa muito espaço na tela do <i>smartphone</i> .	Redimensionar as mídias.
10,1,6	Quando o QR Code é reconhecido não é mostrado um diálogo flutuante.	Inserir diálogo flutuante indicando o que foi selecionado.
6,10,1	O usuário não consegue identificar o filtro aplicado no <i>slider</i> do Navegador RA.	Corrigir a escala utilizada no <i>slider</i> .

Dentre os problemas citados tanto na avaliação com *smartphone* quanto na avaliação com *tablet* destaca-se: a notificação ao usuário sobre traçar uma rota (que é feita automaticamente ao selecionar um POI), a utilidade do botão Limpar que pode não ser percebida, o desajuste no algoritmo de posicionamento dos POIs no Navegador RA, e a ausência de notificação adequada ao reconhecer um QR Code.

Dentre os problemas citados na avaliação com *tablet* destaca-se: a falta de visibilidade para o ícone de fechar mídia, o tamanho de alguns componentes ficaram desproporcionais, a dificuldade

de saber a utilidade de cada ícone na primeira utilização, e o tamanho da informação retornada pela leitura do QR Code.

Tabela 5. Resultados da avaliação heurística realizada no tablet.

Nº da Heurística	Problema	Possível Solução
1	Deve ser informado que a linha azul é o trajeto para o local selecionado.	Legenda de cores utilizadas nos trajetos.
1	Utilizar cores mais contrastantes para diferenciar trajetos escolhidos dos trajetos fixos.	Selecionar cores com maior contraste.
1	Independente do POI escolhido, a linha do circuito sempre atualiza e por último é exibido o trajeto até o POI.	Atualizar somente o trajeto até o POI (linha azul).
4	O ícone de fechar a mídia não está bem visível.	Diminuir a transparência deste ícone e posicioná-lo a direita da mídia.
10	Na primeira utilização fica difícil saber o que são os ícones na barra de ferramentas.	Disponer um <i>tour</i> ao usuário na primeira utilização.
8,4	A exibição de notas sobre o local escolhido é grande e não combina com a interface.	Diminuir o tamanho da janela e a quantidade de informação.
4	O ícone de limpar seleção de POI não tem um ícone claro e seu efeito não é tão perceptível.	Melhorar a transição de estados quando o botão é clicado, para que a mudança seja perceptível.
1	A cada interação os trajetos no mapa são atualizados.	Atualizar os trajetos somente quando necessário ou quando selecionado outro ponto de interesse
4	A barra de mídias fica azul a cada toque nos POIs.	Não mudar a cor da barra de mídias.
2	As posições dos POIs no Navegador RA não estão precisas.	Melhorar a precisão, fixando os ícones, evitando que os marcadores apareçam no teto.
1	Os marcadores do Navegador RA não estão estáveis, ficando difícil de selecioná-los.	Deixá-los mais estáveis
3	Ao marcar um POI no Navegador RA o balão de detalhes no Mapa não é atualizado.	Mostrar detalhes no Mapa para os marcadores clicados no Navegador RA
1	Quando a câmera identifica o QR Code ela congela e não dá nenhum retorno para o usuário.	Informar que o QR Code foi identificado
1	A informação retornada com o QR Code (texto reconhecido) não ficou muito visível, ficou pequeno.	Deixar mais claro o retorno de informações como um pop-up, por exemplo.

6.1.2 Avaliação de Usabilidade Percurso Cognitivo

A avaliação de usabilidade percurso cognitivo do ARguide foi realizada por duas duplas de especialistas que não conheciam a interface gráfica da aplicação. Estas duas duplas são diferentes das duplas que participaram da avaliação heurística. A primeira dupla analisou a aplicação através do *tablet* e a segunda dupla analisou a aplicação através do *smartphone*. O que permite avaliar os passos que apresentam problemas para cada tipo de dispositivo (tamanhos de tela diferentes).

Logo após a apresentação do vídeo sobre a aplicação foi ensinado aos avaliadores como executar uma avaliação de percurso cognitivo de forma verbal e foram apresentadas as fichas que contém as tarefas, com seus respectivos passos, a serem avaliadas e a ficha de avaliação.

As fichas das tarefas possuem as seguintes informações: resumo da tarefa, descrição da tarefa, os passos necessários para realizar esta tarefa no aplicativo, e o uma descrição do provável usuário da aplicação. As três primeiras informações guiam a avaliação de cada tarefa e a descrição do provável usuário informa ao avaliador sob qual ponto de vista ele deve realizar a avaliação. Ou seja, o avaliador deve se pôr no lugar do provável usuário e imaginar se ele consegue realizar uma determinada tarefa.

As fichas de avaliação são tabelas cujas colunas são os passos de uma tarefa e as linhas são perguntas a serem respondidas por cada passo de uma tarefa. Essas perguntas devem ser respondidas pelos avaliadores da seguinte forma: 1 para totalmente improvável, 2 para improvável, 3 para talvez, 4 provável, e 5 para completamente provável. A Tabela 6 mostra a estrutura básica da ficha de avaliação para uma tarefa.

Tabela 6. Esquema das fichas de avaliação Percurso Cognitivo utilizadas na avaliação da aplicação ARguide.

Tarefa X				
Questões	Passo 1	Passo2	...	Passo N
O usuário vai tentar realizar ação correta?				
O usuário vai notar que a ação correta está disponível?				
O usuário vai associar a ação correta com o efeito desejado?				
Caso a ação correta seja realizada, o usuário consegue perceber o progresso em direção a realização da tarefa?				

Os resultados gerados pela avaliação percurso cognitivo foram muito extensos, por isso estes podem ser visualizadas no Apêndice C em forma de tabelas. Os resultados gerados pela avaliação percurso cognitivo tiveram uma visão positiva da aplicação gerando poucos problemas. Assim, as notas que menores ou igual a 3 (escala de 1 a 5) foram consideradas como problemas e foram destacadas nas tabelas do Apêndice C.

Em ambos dispositivos os avaliadores verificaram que é difícil para o usuário notar que a rotação do dispositivo é necessária para encontrar a direção dos POIs. As tarefas de buscar marcadores no mapa, traçar uma rota e realizar ajuste de raio de interesse tiveram uma boa avaliação, o que entra em conflito com os resultados mostrados nos testes com usuários, onde os usuários enfrentaram dificuldades ao realizar estas tarefas.

A tarefa 6 que está relacionada com a leitura de QR Code não apresentou nenhum problema na avaliação percurso cognitivo avaliado no *smartphone*, entretanto esta tarefa apresentou alguns problema na avaliação de teste com o protocolo pensa alto e a avaliação heurística apontou problemas quanto a notificação do usuário após a realização da leitura do QR Code.

Para as tarefas 1 e 2 que estão relacionadas com a busca de POIs e seleção de mídias foi notada uma dificuldade em relação a buscar POIs pelo Navegador RA, entretanto na tarefa 4 (que está relacionada busca e direção de um POI pelo Navegador RA) da avaliação com *tablet* não foi observado este problema, o que pode ser considerado uma inconsistência nos dados.

Os resultados obtidos na avaliação de percurso cognitivo foram resumidos e estão apresentados abaixo:

- O usuário terá dificuldade em localizar os marcadores virtuais pelo Navegador RA;
- O usuário terá dificuldade em localizar os marcadores no mapa;
- O usuário provavelmente não vai notar as abas da aplicação no *smartphone*;
- O usuário não vai notar a ação do *slider* de alcance no *tablet*;
- O usuário não vai saber o que fazer após a leitura do QR Code no *tablet*;
- O usuário terá dificuldades para notar que os marcadores virtuais no Navegador RA são tocáveis no *smartphone*.

6.2 Avaliação por Métodos de Teste

Os três métodos escolhidos para a avaliação com usuários são complementares e podem ser realizados como uma única avaliação com o usuário. Pois, ao mesmo tempo em que o usuário realiza as tarefas estas podem ser gravadas em vídeo com áudio e as interações do usuário podem ser registradas pelo sistema.

Na avaliação de medição de desempenho o usuário realiza um conjunto de tarefas em um determinado cenário enquanto o avaliador coleta informações sobre a utilização da aplicação pelo usuário, por exemplo, o tempo para realizar uma tarefa e a taxa de erros do usuário. O avaliador não pode ajudar o usuário a realizar uma tarefa e nem deve dar suas sugestões durante a realização das tarefas. As tarefas realizadas podem ser gravadas tanto em áudio como em vídeo para uma análise posterior mais minuciosa.

Na avaliação de registro de *logs* do usuário são dadas tarefas para os usuários realizarem em um determinado cenário, e durante a realização dessas tarefas, o sistema registra as interações do usuário com o sistema em *logs* que são analisados posteriormente. Esta avaliação é útil para determinar as características do usuário em relação ao sistema, por exemplo, quais as rotas (interações que direcionam os usuários para novas interfaces e assim por diante) mais acessadas, quantas vezes o usuário clicou em determinado botão, quantas vezes o usuário realizou determinada ação.

Na avaliação protocolo pensa alto, os usuários podem verbalizar seus pensamentos, opiniões, sentimentos e qualquer outra observação enquanto utiliza a aplicação. A utilização da aplicação pode ser pré-definida em forma de tarefas. As falas dos usuários são registradas e analisadas posteriormente, indicando pontos fortes e fracos da aplicação.

As avaliações de teste foram realizadas com vinte participantes que já utilizaram pelo menos um dispositivo móvel, como *smartphone* ou *tablet*, mas não tiveram contato com aplicações RA. Sendo que dez participantes realizaram tarefas no *smartphone* e os outros dez realizaram as mesmas tarefas no *tablet*.

Antes de iniciar a avaliação foi apresentado aos participantes um vídeo que explica de forma geral a aplicação e o seu propósito, dando ênfase tanto nas características da aplicação e na sua utilização, para que os participantes tenham um conhecimento prévio da aplicação, auxiliando a

realização das tarefas. Este vídeo tem propósito semelhante aos vídeos de divulgação de aplicativos em mercados eletrônicos.

Após a apresentação do vídeo são apresentadas as tarefas que o usuário deve realizar. Estas tarefas estão escritas em forma lista e são dadas em um papel para o participante acompanhar o andamento das tarefas realizadas em relação as que ainda faltam realizar.

A Tabela 7 lista as tarefas que devem ser realizadas pelos participantes do teste. A primeira coluna indica o número da tarefa, a segunda coluna apresenta o texto da descrição da tarefa exatamente como foi apresentada para o usuário, a terceira coluna apresenta o contexto desta tarefa, e a quarta coluna indica número mínimo de ações (cliques em botões, troca de aba, e qualquer outra interação) necessárias para realizar a tarefa.

Tabela 7. Lista de tarefas realizadas pelos participantes da avaliação da aplicação ARguide por medição de desempenho.

Nº	Descrição	Contexto	Interações Mínimas
1	Selecione a Biblioteca Central.	Mapa, Navegador RA.	1
2	Visualize uma mídia do Centro de Convenções.	Mapa, Navegador RA, Mídias.	2
3	Trace uma rota da sua posição até o Hospital Betina Ferro.	Mapa.	1
4	Encontre a direção da Reitoria pelo Navegador RA.	Navegador RA.	2
5	Diminua o raio de alcance para visualizar somente os pontos mais próximos.	Navegador RA, Filtro de POIs.	2
6	Realize a leitura de um QR Code, visualize as mídias e diga do que se trata.	Leitor QR Code, Mídias.	2

Após a apresentação das tarefas o participante inicia a realização das tarefas com tempo livre para finalizar cada tarefa. A realização das tarefas foi gravada em forma de vídeo (com foco na tela do dispositivo de teste e das mãos dos participantes) e áudio (foco nas falas dos participantes) e foram realizadas em uma sala, não havendo visitação ou caminhada para os POIs.

6.2.1 Perfil dos Participantes

Os participantes do teste de usabilidade são todos usuários frequentes de dispositivos móveis, sendo que a maioria utiliza ou já utilizou o sistema operacional Android, têm faixa etária majoritariamente entre 18 e 35 anos, e utilizam seu dispositivo móvel com muita frequência.

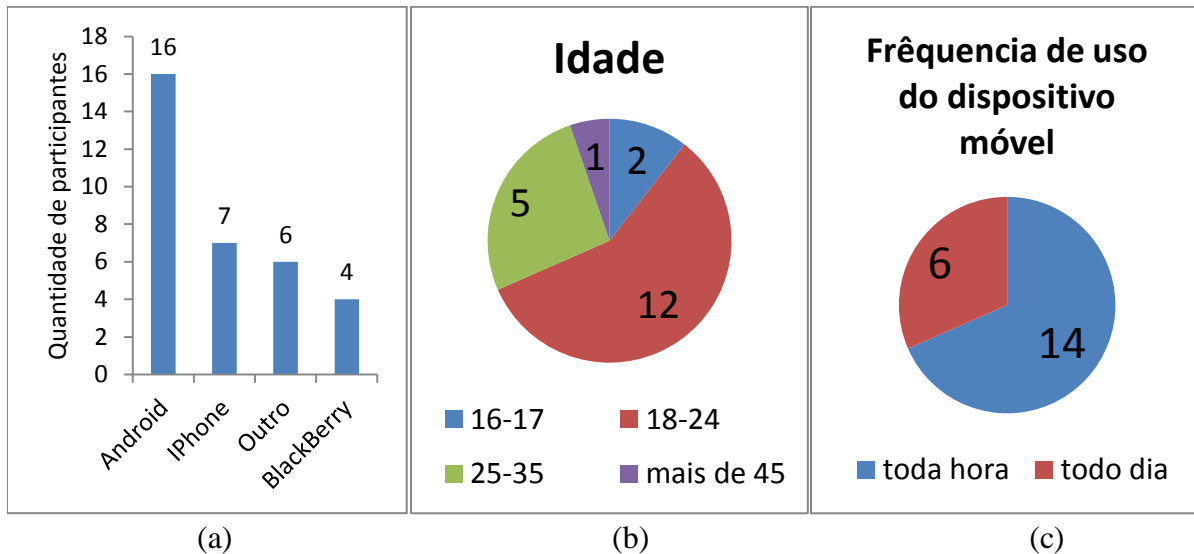


Figura 32. Perfil dos participantes do teste de usabilidade. (a) Sistema operacional que já utilizou/utiliza, (b) faixa etária, e (c) frequência de uso de dispositivo móvel.

O perfil dos participantes do teste de usabilidade é semelhante ao perfil dos participantes da pesquisa realizada pela comScore (2015) sobre a utilização de dispositivos móveis na América Latina, que aponta uma maior utilização do sistema operacional Android sobre os outros sistemas.

6.2.2 Avaliação de Usabilidade por Medição de Desempenho

Os gráficos da Figura 33 mostram a média de tempo que os participantes levaram para realizar cada tarefa pelo *tablet* (a) e pelo *smartphone* (b). Segundo essas médias de tempo os usuário levaram menos tempo para realizar as tarefas no *smartphone*, com exceção da Tarefa 3 que pede para o participante traçar uma rota de sua posição até um POI.

Outro ponto importante presente na Figura 33, tanto para o *tablet* quanto para o *smartphone*, é o tempo levado para realizar a Tarefa 6, que corresponde a leitura de um QR Code e em seguida dizer do que se trata o QR Code lido. Nesta tarefa o sistema não alertou os participantes de forma que eles percebessem que algo estava disponível depois da leitura.

O aplicativo pisca o fundo do componente onde as médias ficam (Figura 23 – região 4, página 57) três vezes na cor azul, isto na tentativa de mostrar para o usuário que as médias disponíveis para ele mudaram quando este realiza uma seleção em um POI. Entretanto, foi observado que esta medida não é suficiente quando se troca de uma visão para outra como é o caso da leitura de QR Code. Uma hipótese para este comportamento é que ao trocar de visão, a segunda visão chama mais atenção do usuário que a barra de médias (mesmo piscando), fazendo com que o usuário volte a interagir com a outra visão ao invés de interagir com as médias.

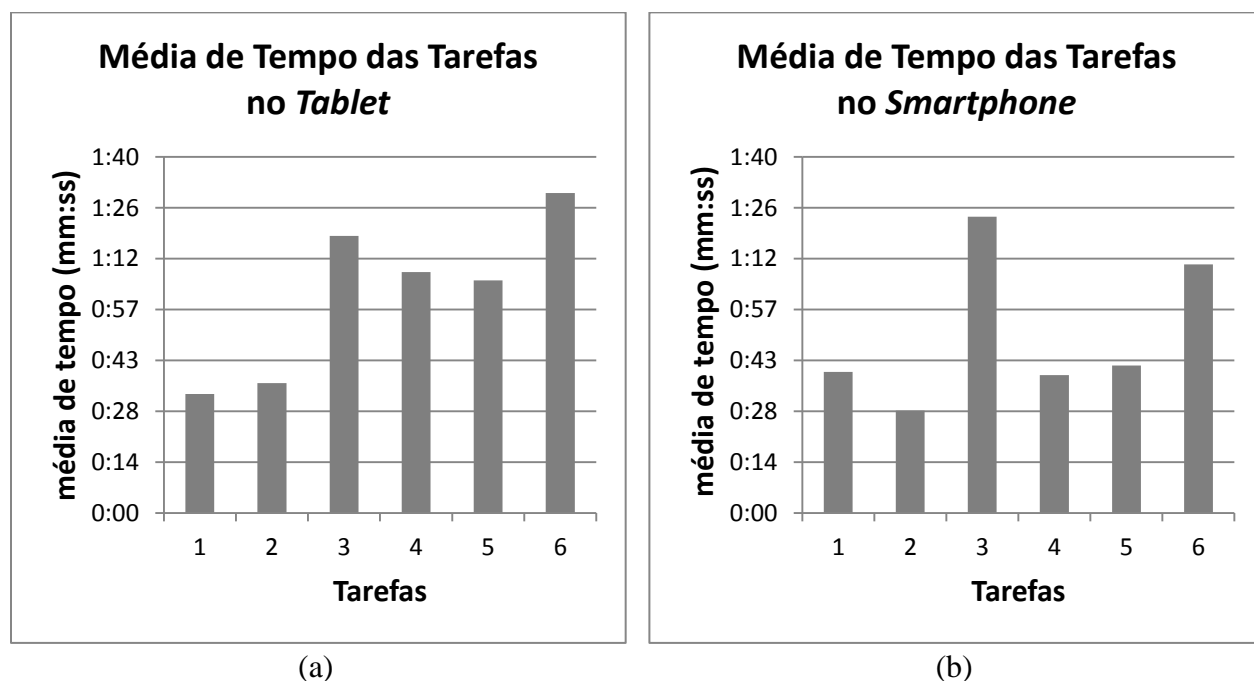


Figura 33. Gráficos da média do tempo de realização das tarefas para *tablet* (a) e *smartphone* (b).

A Figura 34 mostra o gráfico da taxa de acertos, erros e desistências dos participantes por tarefa, utilizando o *tablet* (a) e o *smartphone* (b). Estes gráficos mostram que a taxa de desistência e erro foi um pouco maior no *smartphone*, com destaque para as tarefas 3 e 5, que estão associadas as funções rota e configuração de raio de alcance.

A função de rota na aplicação é acionada automaticamente quando um usuário seleciona um POI e visualiza o mapa, esta configuração foi feita pensando em diminuir o número de interações que o usuário precisa realizar para concluir a tarefa. Porém, através deste resultado foi percebido que esta função está tão automática (apenas a seleção realiza a função) que o usuário não percebe que realizou a tarefa.

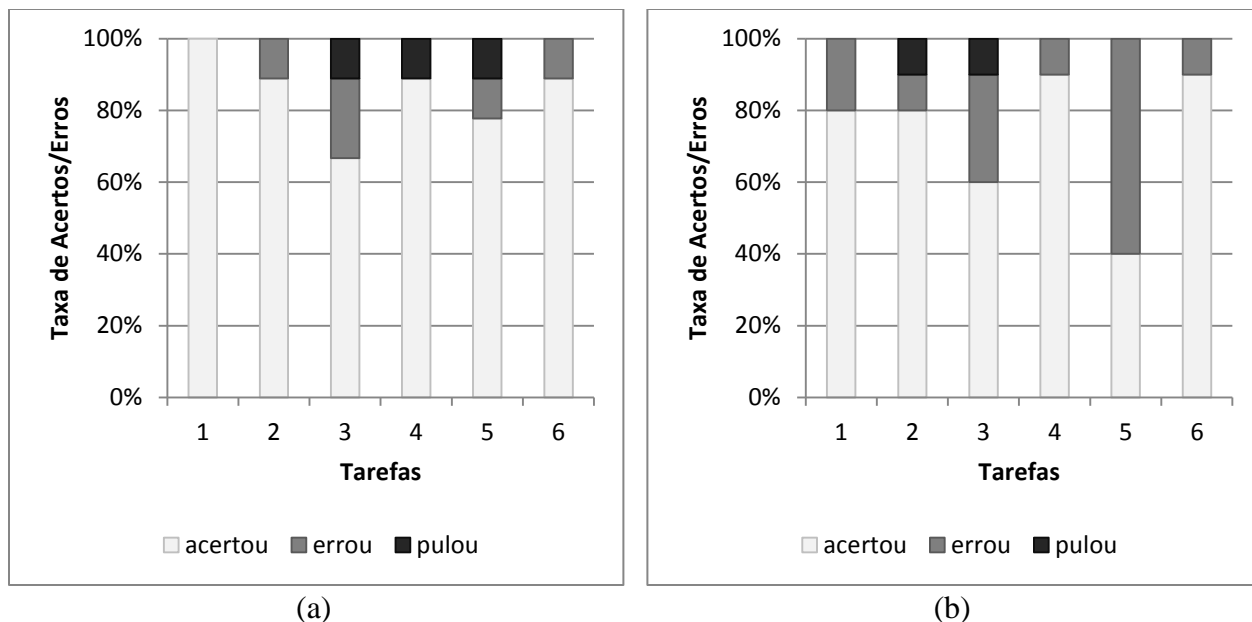


Figura 34. Taxas de acertos, erros e desistências por tarefas realizadas no *tablet* (a) e *smartphone* (b).

A função de ajustar o raio de alcance é configurada em um componente do tipo *slider* que fica posicionado ao lado direito da visão Navegador RA (Figura 26, página 63), neste o usuário deve arrastar um marcador (círculo azul) para cima ou para baixo, a fim de aumentar ou diminuir o raio de alcance. O *feedback* visual desta interação é mostrada no radar, que fica ao lado esquerdo superior do *slider*, e nos marcadores que começam a aparecer ou desaparecer da tela, quando a câmera está apontada adequadamente.

Uma hipótese para a alta taxa de erros ao configurar o raio de alcance no *smartphone* é o tamanho do radar que na tela do *smartphone* fica muito pequeno não dando um *feedback* apropriado para o usuário.

6.2.3 Registro de *logs* do Usuário

Os padrões de toques na aplicação foram extraídos dos *logs* que o sistema fazia dos usuários durante a realização das tarefas. Os dados brutos foram selecionados a partir da análise dos vídeos gravados durante a realização das tarefas. O tempo em cada registro de *log* foi cruzado com o tempo gravado no vídeo, e assim foi possível extrair dos dados brutos apenas os toques e interações representativas para a avaliação.

A partir da extração dos dados representativos foram classificados 24 padrões de toques, que estão listados na Tabela 8 com seus respectivos IDs para referências futuras no texto. Esses

padrões de toques são apenas toques considerados como errados, ou seja, os toques que os usuários fizeram de forma a corretamente concluir as tarefas foram desconsiderados.

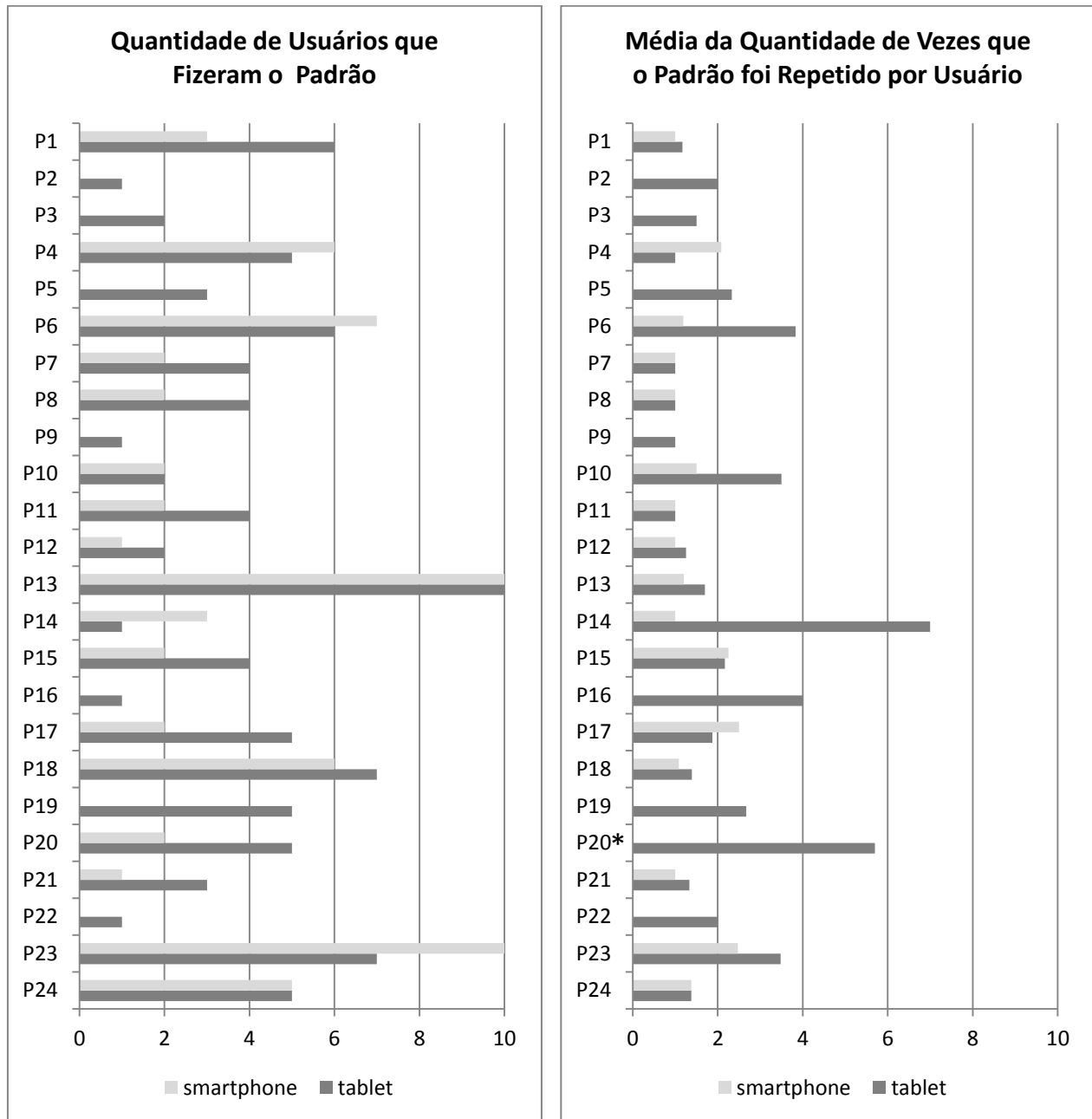
Tabela 8. Padrões de toques extraídos dos logs registrados pela aplicação durante a realização das tarefas e os IDs de cada padrão de toque.

ID do Padrão	Descrição do Padrão de Toque
P1	Clique involuntário no botão de voltar.
P2	Clique involuntário no botão QR Code.
P3	Clique na barra de status do Android.
P4	Clique na foto do balão detalhes do marcador no mapa.
P5	Clique na logo ARguide.
P6	Clique na mídia de imagem.
P7	Clique no botão “sua posição” no mapa.
P8	Clique no botão de ajuda.
P9	Clique no botão de ajuda para fechar ajuda.
P10	Clique no botão de mídia de POI selecionado anteriormente.
P11	Clique no botão do Google Maps.
P12	Clique no botão limpar.
P13	Clique no botão QR Code mais de uma vez ou no momento errado.
P14	Clique no marcador da posição do usuário no mapa.
P15	Clique no radar.
P16	Deslizar mídia de imagem.
P17	Diminuir de mais o raio de alcance.
P18	Fechou mídia Web antes de carregar.
P19	Movimentar a câmera incorretamente para descobrir POIs.
P20	Pan no mapa sem <i>zoom out</i> .
P21	Pinçar a câmera.
P22	Pinçar mídia de texto.
P23	Tocar no marcador errado.
P24	Trocar de aba e voltar.

Com essa classificação foi possível contabilizar a quantidade de usuários que realizaram cada um desses padrões e a quantidade de vezes que os usuários repetiram esse padrão. A Figura 35 mostra os gráficos desses dados, sendo (a) é média de usuários e (b) a média de repetições. Vale ressaltar, que a média de repetições considerou apenas os usuários que fizeram o padrão, por exemplo, se dois usuários fizeram um padrão X, então se tira a média de vezes que esses dois usuários realizaram o padrão X.

A Figura 35 (b) oculta um valor discrepante que é o P20 para uma melhor visualização dos demais padrões. O padrão P20 foi observado quando o usuário realiza *pan* no mapa sem saber a

direção em que ele deve ir e sem distanciar o mapa (*zoom out*). A discrepância deste padrão mostra que quando os usuários ficam perdidos (geralmente quando não aparece nenhum marcador no mapa) tendem a ficar movimentando o mapa (*pan*) mesmo sem saber para onde, a fim de encontrar algo.



P20* é um ponto discrepante: 27,33

(a)

(b)

Figura 35. Gráficos sobre: a quantidade de usuários que realizaram padrões de toques por dispositivo, e a quantidade de vezes que os usuários repetiram esse padrão.

Uma hipótese para este comportamento é que os usuários simplesmente esquecem ou não notaram que a funcionalidade de encolher o mapa (*zoom out*) pode ajuda-los a encontrar os POIs. Outra hipótese é que o usuário considera que buscar com o mapa ampliado é melhor por mostrar mais detalhes, como nomes de ruas, praças, etc.

O padrão P13 e P23 foram realizados por todos os usuários, sendo que o P23 não foi realizado por três usuários no *tablet*. Após uma análise nos vídeos e *logs* foi verificado que esses três usuários realizaram as tarefas pelo Navegador RA ao invés do Mapa. Após essas análises, o mais provável é que P23 foi realizado por todos os usuários por não ter rótulos textuais nos marcadores do mapa e que o P13 foi realizado por todos os usuários porque após a leitura do QR Code o sistema não chama atenção suficiente para as mídias disponíveis, nem notifica o usuário apropriadamente sobre qual POI foi selecionado pelo QR Code.

Os padrões P4, P6 e P24 foram realizados por aproximadamente metade dos usuários e de forma semelhante em ambos dispositivos (*tablet* e *smartphone*). Os padrões P4 e P6 estão relacionados com o clique dos usuários em imagens a fim de ampliá-las e o P24 geralmente ocorre quando o usuário está com dificuldades para realizar uma tarefa.

Os padrões P1, P19 e P21 foram ligeiramente mais realizados por usuários que utilizaram o *tablet*. Foi verificado que esta diferença no P1 é pelo fato do *tablet* ser maior e mais pesado que o *smartphone*, deixando a tarefa de segurar, apontar, e tocar na tela (segurando o dispositivo com uma mão) mais difícil, sendo mais suscetível a toques involuntários. O padrão P19 se deu em poucos casos por um problema de ajuste no algoritmo que posiciona os marcadores na tela. E no padrão P21 foi observado que alguns usuários desejavam ver os marcadores com mais detalhes, tal como no mapa, tentando ampliar a câmera no sendo de aumentar os marcadores ou aplicar o filtro de alcance.

6.2.4 Avaliação Protocolo Pensa Alto

Durante a realização das tarefas os usuários estão concentrados nas atividades que devem fazer, neste momento podem expressar seus sentimentos naturalmente com algumas frases. Como as tarefas foram gravadas em vídeo e áudio foi possível extrair essas frases que demonstram os sentimentos dos usuários em relação a alguns componentes, situações ou comportamentos da aplicação.

Foram extraídas todas as falas significativas dos vídeos revendo as gravações e anotando sempre que preciso. As falas foram classificadas por dispositivo e por função que estava sendo utilizada quando a frase foi falada. Para cada fala foi identificado o problema na interface gráfica para o qual a fala se aplica.

A Tabela 9 possui as falas dos participantes que utilizaram o *tablet* para realizar as tarefas e a Tabela 10 possui as falas de participantes que utilizaram *smartphone*. A primeira coluna de cada dessas tabelas possui o problema relacionado com a fala do usuário e na segunda coluna está a fala em si.

As falas relacionadas com o balão de detalhes do mapa (Figura 31 - a) estão relacionadas com o fato de que o balão cobre parte do mapa e sobrepõem os marcadores. Como não há um botão de fechar os usuários ficam nervosos. Para fechar esse balão o usuário deve clicar em qualquer outro lugar no mapa, mas foi identificado que essa interação não é intuitiva para alguns usuários.

Foram identificadas 12 falas sobre os marcadores no mapa quando utilizado o *tablet* e no *smartphone* foram identificadas 5 falas. Essas falas geralmente estão relacionadas com a ausência de um rótulo de texto nos marcadores. Os marcadores estão com um rótulo numérico que indica uma sequência de visitação sugerida. Entretanto, foi identificado que os usuários não notaram essa sequência e não queriam essa informação, sendo prioridade um rótulo textual.

A funcionalidade de traçar rota pelo aplicativo foi a mais problemática na visão mapa. As falas geralmente estão se referindo a dúvidas de onde e como fazer uma rota. Para fazer uma rota na aplicação basta selecionar um marcador e o sistema traça uma rota entre a posição do usuário e o marcador clicado. Mas, desta forma os usuários não perceberam que a rota já estava traçada, e quando percebiam falavam frases como: “Já fez a rota?!”.

Tabela 9. Falas dos participantes extraídas durante a realização das tarefas quanto estavam interagindo com a função Mapa pelo *tablet*.

Utilizando <i>tablet</i>	
Problema (qtd.)	Falas no Mapa
Balão Detalhes Mapa (2)	Como é que eu fecho isso aqui?! (balão mapa)
	Acho que essa imagem deveria sair da frente. (balão mapa)

Marcadores Mapa (12)	Pois é, como é que eu vou saber o que é esse ponto vermelho?
	Agora eu vou ter que ficar procurando um por um? (marcadores)
	Não aparece o nome! (No marcador do mapa)
	Onde está o centro de convenções?
	Talvez se tivesse com um nome... (marcadores)
	Achei! Na sorte.
	Não é para aparecer o nome? (No marcador do Mapa)
	Eu acho que é aqui, não dá pra ver se é, mas acho que é aqui.
	Se o CEPS é aqui, então a Biblioteca é aqui.
	Não estou achando nada aqui.
	Ai meu Deus... Cadê o hospital?
	Eu não estava conseguindo clicar no marcador. (marcadores sobrepostos)
Rota até o POI (4)	Para traçar rota...? Aqui?
	Agora como é que eu vou fazer isso? (Rota até o POI)
	Como é que eu faço para traçar uma rota... ?
	Já fez a rota?!
Rota até o POI e circuito entre POIs (2)	A linha azul está muito fraca, confunde com o verde.
	Como é que traça uma rota? Já está traçada?!
Visualizador de Mídias (1)	Está faltando expandir a imagem (ao tocar na mídia de imagem)
Problema (qtd.)	Falhas no Navegador RA (tablet)
Desajuste do algoritmo de posicionamento (2)	Engraçado está aparecendo aqui (radar) e não está aparecendo na tela.
	No teto?!
Marcadores (6)	Como se fosse uma bússola? Para eu ir, indo, indo, indo até chegar lá?
	Hum, tá aqui...
	Hum, entendi...
	Esse aqui? Reitor? (marcador sobreposto)
	Cadê a reitoria daqui?
	Como é que eu vou achar a Reitoria aqui?!
Radar pequeno (1)	Não tinha prestado atenção que dava pra ver a distância aqui (Radar).
Raio de alcance (4)	Aqui eu aumentando e diminuindo não está ajudando muito. (slider alcance)
	Não estou entendendo. (ajustar raio de alcance)
	Ah, não sei. (ajustar raio de alcance)
	É... Não consegui...

Raio de alcance e Radar	Creio que seja aqui o raio de alcance... 500, 300, 96. Só tem um ponto!
Problema (qtd.)	Falas no QR Code (tablet)
Ausência de <i>feedback</i> adequado (4)	Tá... Não me deu nenhuma informação, que era pra ter me dado...
	Era para falar: "vá para algum lugar", mas não ele só leu, e o quê que eu tenho que fazer?
	Ele tem que dizer para onde eu devo ir depois de ler o QR Code.
	Onde é que eu aperto?
Mídia permaneceu aberta (1)	O que essa imagem tem a ver aqui?

Os problemas encontrados na Visão Navegador RA estão relacionadas principalmente com o ajuste do raio de alcance, e com o posicionamento dos marcadores na tela. Foi percebido que os usuários não entendem como funciona o raio de alcance e não conseguem assimilar que ao movimentar o *slider* é ajustado esse alcance. Uma hipótese é que o *slider* dá feedbacks somente no radar que está muito pequeno, e essa movimentação, para cima e para baixo, não deixa o processo intuitivo.

O posicionamento dos marcadores atrapalha os usuários quando existe mais de um marcador na mesma direção e o algoritmo tenta espalhá-los, para que o usuário consiga visualizar todos. Entretanto, esse espalhamento nem sempre ocorre de maneira satisfatória e acaba posicionando marcadores muito acima do horizonte ou muito abaixo.

Tabela 10. Falas dos participantes extraídas durante a realização das tarefas quanto estavam interagindo com a função Mapa pelo *smartphone*.

Utilizando <i>smartphone</i>	
Problema (qtd.)	Falas no Mapa (<i>smartphone</i>)
Balão Detalhes Mapa (2)	Eu só vejo o centro de convenções
	Não tem como fechar essa janela? (Balão detalhes no Mapa)
Busca no Mapa (4)	Acho que eu já estou perdido...
	Almirante Barroso? Cadê a universidade daqui?
	Não tem nenhum ponto... Mas, também eu estava indo para o lado errado...
	Onde é a biblioteca?
Mídia de Imagem (2)	Só que a mídia ficou na minha cara... (mídia de imagem)
	Eu quero clicar no que está atrás da janela (Detalhes Mapa) e não consigo.

Rota até o POI (10)	Ok, traçar a rota eu não sei fazer...
	Acho que faltou alguma coisa que me indicasse onde fazer a rota...
	Já está traçado...?! (a rota)
	Mas como que eu vou traçar uma rota até o hospital Betina Ferro?
	É essa linha azul aqui?
	Mas, traçar rota pelo aplicativo?
	Mas eu tenho que achar as direções? (Para traçar rota)
	Tá difícil! (traçar rota)
	Se eu quiser começar uma rota, eu posso indicar que eu estou começando aqui?
	Ah! Esta aqui, traçou a rota... Como eu fiz isso?
Marcadores Mapa (5)	Esses pontos deviam ter algum nome ou sigla para agente reconhecer...
	Cadê o centro de Convenções que eu tinha achado ele antes? Agora não estou achando mais...
	Mas eu não sei onde é... Cadê o Betina?
	Qual é o centro de convenções aqui?
	Agora eu achei a Biblioteca... Ainda agora quando eu estava procurando
Saiu da aplicação (1)	Ih!... Agora ele sumiu... (tocou no botão de mídia Web)
Problema (qtd*)	Falas no Navegador RA (<i>smartphone</i>)
Busca no Mapa	Achei o Centro de Convenções que eu estava procurando...
Desajuste do algoritmo de posicionamento (3)	Está aqui a Reitoria. Está no céu, mas está aqui.
	A reitoria fica do lado do Centro de Convenções, mas não está aparecendo aqui...
	Não está bem centralizado, mas aparece...
Marcadores sem profundidade, apenas distância.	Mas ai a reitoria fica atrás do centro? Como é que funciona?
Não ver marcadores	Como é que eu vou saber onde está a Reitoria aqui... Ah! Para Cá!
Raio de alcance (2)	Como é que diminui o raio de alcance
	Não sei fazer isso... (diminuir o raio de alcance)
Problema (qtd*)	Falas no QR Code (<i>smartphone</i>)
Balão Detalhes Mapa permaneceu aberto (1)	Eu acho que o QR Code é sobre o Hospital (Olhando para os detalhes no mapa)
Mídia permaneceu aberta (1)	É sobre a Biblioteca Central... Ou não... É sobre o Restaurante Universitário.

As falas relacionados com o leitor de QR Code são principalmente por não fornecer um *feedback* adequado aos usuários quando estes realizam a leitura do QR Code. O sistema pisca a

barra de mídia 3 vezes, mas como comentado anteriormente, isto não chama atenção suficiente dos usuários.

6.3 Comparação dos Resultados entre Avaliações

As avaliações de usabilidade da aplicação ARguide foram realizadas na mesma versão da aplicação, que foi uma melhoria da versão utilizada nos testes de cenário de uso, por isso é possível listar os problemas de usabilidade encontrados e realizar uma comparação entre as avaliações.

Cada avaliação gerou um conjunto de problemas que podem se repetir de avaliação para avaliação, assim a Tabela 11 contém os problemas gerados pelas avaliações de usabilidade e em quais avaliações foram observados cada problema. A primeira coluna desta tabela indica o componente de interface gráfica que apresenta o problema, a segunda descreve o problema em si, e as demais colunas indicam, para cada avaliação realizada, se o problema foi encontrado a partir da avaliação.

A partir dessa comparação é possível verificar quais problemas foram identificados por meio das avaliações com especialistas e confirmadas pela avaliação com usuário, quais problemas foram identificados unicamente por um método de avaliação e quais os problemas foram observados em todos os métodos.

O método que mais identificou problemas foi o protocolo pensa alto (22 problemas) seguido da avaliação heurística (17 problemas) e o terceiro foi o registro de *logs* do usuário (13 problemas). Embora a avaliação de registro de logs tenha gerado um grande número de problemas em relação às demais, esta avaliação depende dos resultados obtidos a partir da análise dos vídeos, principalmente para realizar a filtragem das interações corretas dos usuários.

Os problemas “Não saber o que fazer após a leitura” e “A ação do *slider* não é compreensível” foram observados em todas as avaliações, assim são problemas que possivelmente devem ter maior prioridade para ações de melhorias, uma vez que estes problemas foram observados tanto pela observação de usuários quanto pelos especialistas que analisaram a interface gráfica.

Tabela 11. Comparação das avaliações realizadas e os problemas encontrados.

Componente	Problema	Avaliações				
		Heurística	Percurso Cognitivo	Med. Desempenho	Registro de logs	Protocolo Pensa Alto
Ajuda	Não saber o que fazer na primeira utilização.	X			X	X
	Ajuda não dá o suporte necessário.	X				
Ausência de busca textual	Não há um campo para realizar busca por texto.	X				
Balão Detalhes Mapa	O balão fica sobrepostos nos POIs o que dificulta a visualização					X
	Não ter uma forma fácil de fechar o balão					X
	Ocupar muito espaço da visualização do mapa					X
	Não há interações com a imagem do balão				X	
	Balão não atualiza ao selecionar POI em outra visão	X				X
Barra de Mídias	Barra de mídias troca de cor ao selecionar POI	X				
Botão Mídia Web	Não saber o que é e sair da aplicação sem querer				X	X
Leitor QR Code	Não saber o que fazer após a leitura	X	X	X	X	X
	Feedback da leitura não é adequado	X				X
Marcadores no Mapa	Dificuldades em localizar os marcadores		X	X	X	X
	Não possuem rótulos de texto	X			X	X
	A operação de zoom não é utilizada corretamente			X	X	X
Marcadores no Navegador RA	Dificuldade em localizar os marcadores	X	X		X	X
	Problemas no algoritmo de posicionamento	X				X
	Não perceber a interação de toque nos marcadores		X			
	Marcadores se mexem muito dificultando o toque	X				
Radar	Informação de distância não está bem visível					X
Rating de POIs	O rating está grande de mais	X				
Rotas no mapa	Não há informações sobre o significado das linhas	X				X
	As rotas não estão visíveis o suficiente	X		X		X
	Não perceber que a rota já foi traçada			X		X
	Não saber como traçar a rota					X
Slider de alcance	A ação do slider não é compreensível	X	X	X	X	X
Visualizador de Mídias	Mídia não fecha ao selecionar POIs pelo QR Code				X	X
	O botão de fechar mídia não está bem visível	X			X	X
	Ausência de ampliação da imagem ao tocar nela				X	X
Abas das Visões	Não vai notar as abas		X			
	Trocar de aba e voltar				X	
Botão Limpar	Não está compreensível	X				

Foram observados seis problemas apenas com a avaliação heurística, cinco problemas apenas com o protocolo pensa alto, e dois problemas apenas com o percurso cognitivo. A avaliação percurso cognitivo não gerou muitos resultados, pois as tarefas eram compostas por poucos passos e aplicação não possui muitas interações com menus e conjuntos de botões. Assim, foi observado que a avaliação percurso cognitivo pode ser mais bem aplicada em softwares com mais componentes clicáveis e com muitas telas e interações.

6.4 Propostas de Melhorias

A partir dos resultados obtidos nos testes de usabilidade foi feito um resumo dos problemas identificados e foram sugeridas melhorias para cada problema. Essas sugestões de melhorias serão analisadas pela equipe de desenvolvimento, que dará início ao segundo ciclo de desenvolvimento iterativo.

A Tabela 12 mostra um resumo dos problemas encontrados e as suas respectivas propostas de melhorias.

Tabela 12. Resumo dos problemas encontrados e sugestões de melhorias.

Problema	Proposta de Melhoria
Leitura de QR Code não dá <i>feedback</i> apropriado.	Abrir mídias em uma janela separada. Sendo que, celular ocupando a tela inteira e em tablet ocupando meia tela. Esta janela deve conter o nome do POI selecionado em destaque, e as possíveis interações, como traçar rotas, ver conteúdo, e ver página Web deste POI.
Balão de detalhes no mapa cobre informações importantes.	
Rota e circuito não são notados pelos usuários	
A mídia Web não notifica o usuário que vai sair da aplicação.	
Imagens não são ampliáveis	Colocar todas as imagens na janela de conteúdos do POI e permitir que as imagens fiquem em tela cheia.
Usuários perdidos no mapa	Colocar marcadores nos cantos do mapa quando o usuário não estiver com marcadores na tela. Assim o usuário saberá para onde deve movimentar a câmera quando não houver nenhum marcador.
Marcadores sem rótulos	Adicionar texto nos marcadores do mapa ao invés de números. Os textos devem ser mostrar de acordo com o espaço disponível. Os números com sugestões de visitas devem ser mostradas apenas quando o usuário requisitar.

Mídias ficam sobrepostas às visões	As mídias devem aparecer na janela de conteúdos. Caso seja necessário mostrar uma mídia sobreposta a cena real, deve ser implementada uma nova visão que dê essa possibilidade a partir de um marcador ou reconhecimento de imagem.
Raio de alcance não está compreensível	Remover o <i>slider</i> lateral e permitir esse ajuste pela interação de pinçar, como se estivesse ampliando uma imagem. Podem ser colocadas linhas de distâncias sobrepostas à câmera de forma que toda vez que o usuário pinçar a câmera ele terá o <i>feedback</i> visual pela movimentação dessas linhas.
Marcadores no Navegador RA não dão informação visual de distância	
Algoritmo de posicionamento do Navegador RA confunde os usuários e dificulta a localização.	Manter os marcadores sempre na linha do horizonte, ou seja, alinhado com os olhos do usuário. Quando houver muitos POIs na mesma direção, as linhas de distância devem separar alguns deles, e se ainda sim tiverem muitos POIs agrupados, eles devem ser concatenados em um marcador mostrando o número de marcadores concatenados.
Radar não está bem visível e não é muito perceptível	O radar pode ser trocado pela visão mapa em miniatura. Este mapa deve ser trocado (entre Navegador RA e mapa) de forma que o usuário perceba que o radar é a mesma visão que o mapa.
Ajuda textual não dá o suporte necessário	Desenvolver um <i>tour</i> para a primeira utilização da ferramenta, tendo um exemplo e ensinando passo a passo para o usuário como realizar uma tarefa na ferramenta.
Toques involuntários em voltar fecham a aplicação	Antes de fechar a aplicação, perguntar se o usuário deseja realmente sair da aplicação. A confirmação pode se dá por um segundo toque no botão de voltar.

CONCLUSÃO

Este trabalho apresentou uma proposta para a descoberta, navegação e exploração de POIs através de múltiplas visões com a utilização de dispositivos móveis. Para isso, foi proposta uma arquitetura para aplicações RAM extensível, flexível, e adaptável, sugerindo uma adaptação no padrão MVC e considerando a utilização de padrões de projetos, como *Proxy Remoto*, *Facade* e *Observer*, assim como a utilização do padrão de interfaces gráficas *Fragments*.

A aplicação ARguide foi desenvolvida como implementação da proposta de arquitetura e foram realizados teste de adequação à esta arquitetura. Os testes mostraram que a aplicação funciona bem mesmo com a troca dos dados, assim como para tamanhos de tela diferentes, tendo então atendido a arquitetura e mostrando que a mesma realiza bem o seu propósito.

Para avaliar a descoberta, navegação e exploração de POIs pela interface gráfica da aplicação foram realizados testes de usabilidade com especialistas e usuários, que revelaram alguns problemas de *design* de interface e problemas técnicos como bugs. Essas problemas comprometem a boa descoberta, navegação e exploração de POIs, assim foram sugeridas melhorias para essa implementação.

Foram realizadas cinco avaliações de usabilidade na aplicação desenvolvida, estas mostraram vários problemas de interface gráfica da aplicação que devem ser corrigidos para as próximas versões. Os resultados das avaliações foram apresentados e foi feito uma comparação entre os resultados gerados por cada uma dessas avaliações. Assim, foi possível identificar quais os problemas foram notados em todas as avaliações e quais avaliações obtiveram maior quantidade de problemas identificados.

Após essa comparação entre os resultados gerados pelas avaliações foi identificado que para aplicações no estilo da aplicação ARguide as avaliações de protocolo pensa alto e avaliação heurística se destacaram, enquanto que a avaliação percurso cognitivo não obteve resultados

satisfatórios, sendo possivelmente indicada para softwares que possuem tarefas com mais interações telas e botões, o que geralmente é o caso de aplicações para o ambiente *desktop*.

Dentre os pontos fortes da proposta, pode-se destacar: a possibilidade de adicionar novas funcionalidades na aplicação devido à lógica da aplicação ter passado para as visões, basta adicionar uma nova visão que saiba se comunicar com a camada Coordenadora para ter uma nova funcionalidade; a possibilidade de adicionar vários tipos de dados, com o único requisito dos dados serem georeferenciados (para o funcionamento do Mapa e Navegador RA), permitindo que a aplicação seja genérica servindo a muitos propósitos; e a adaptação a diferentes tamanhos de tela que é uma característica muito comum em dispositivos móveis com o sistema operacional Android.

Dentre os pontos fracos da aplicação, pode-se destacar: a necessidade do fornecedor dos dados ter que lidar com a configuração do projeto (códigos e diretórios da aplicação) para gerar sua aplicação personalizada; e a ausência de conteúdo 3D que é importante para uma experiência de realidade aumentada, embora adicionar uma nova funcionalidade tenha se tornado mais fácil.

A partir deste trabalho foram desenvolvidas três publicações em anais de congresso, que foram as seguintes: Considerações de Projeto e Implementação de uma Aplicação Android de Realidade Aumentada Móvel (SANTOS, CARNEIRO, *et al.*, 2014), Uma Aplicação de Realidade Aumentada Móvel para Ambientes Indoor e Outdoor (SANTOS, CARNEIRO, *et al.*, 2014), e Aplicações Android de Realidade Aumentada em Arquitetura Extensível, Flexível e Adaptável (ARAUJO, CARNEIRO, *et al.*, 2015).

Os trabalhos futuros são aplicar melhorias propostas e verificar qual a eficiência da utilização de múltiplas visões com uma interface gráfica bem ajustada, adicionar novas visões à aplicação, melhorar visualização dessas múltiplas visões, permitindo que os usuários vejam mais de uma visão simultaneamente, adicionar comandos de voz na aplicação, favorecendo a utilização do Navegador RA majoritariamente com as duas mãos, entre outras melhorias.

Após as melhorias da ferramenta serão feitas mais avaliações tanto de usabilidade, para estabelecer um quadro comparativo, como avaliações de desempenho com o objeto de medir a qualidade das modificações aplicadas.

REFERÊNCIAS

ARAÚJO, T.; CARNEIRO, N.; MIRANDA, B.; SANTOS, C. G.; MEIGUINS, B. S. **Aplicações Android de Realidade Aumentada em Arquitetura Extensível, Flexível e Adaptável**. XI Simpósio Brasileiro de Sistemas de Informação. Goiás: Instituto de Informática da Universidade Federal de Goiás. 2015. p. 63-70.

ARTH, C.; SCHMALSTIEG, D. **Challenges of Large-Scale Augmented Reality on Smartphones**. ISMAR 2011 Workshop: Enabling Large-Scale Outdoor Mixed Reality and Augmented Reality. Basel: [s.n.]. 2011.

AWILA. **AWILA Augmented Technologies**, 2014. Disponível em: <<http://www.aug-tech.co.uk/augmented-reality/>>. Acesso em: junho 2015.

AZUMA, R.; BAILLOT, Y.; BEHRINGER, R.; FEINER, S.; JULIER, S.; MACINTYRE, B. Recent Advances in Augmented Reality. **Computer Graphics and Applications, IEEE**, Los Alamitos, CA, v. 21, n. 6, p. 34-47, Novembro 2001.

BIMBER, O.; RASKAR, R. **Spatial Augmented Reality: Merging Real and Virtual Worlds**. Danvers, MA: A K Peters/CRC Press by Taylor & Francis Group, 2005.

BLIPPAR. **Blippar**, 2015. Disponível em: <<https://blippar.com/en/>>. Acesso em: Junho 2015.

CAWOOD, S.; FIALA, M. **Augmented Reality: A Practical Guide**. 1^a. ed. Dallas, Texas: Pragmatic Bookshelf, 2008.

CHEN, L.-W.; PENG, Y.-H.; TSENG, Y.-C. **An Augmented Reality Based Group Communication System for Bikers Using Smart Phones**. IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). Seattle, WA: IEEE. 2011. p. 325-327.

CHEN, Y.; XIANG, L.; ZHANG, J.; LIU, L. **Research about Mobile AR System based on Cloud Computing**. 22nd Wireless and Optical Communication Conference (WOCC). Chongqing: IEEE. 2013. p. 355 - 359.

COMSCORE; IMS. **IMS Mobile in LatAm Study**. comScore Inc. e IMS Corporate. Acessado em: <http://www.comscore.com/Insights/Presentations-and-Whitepapers/2015/comScore-IMS-Mobile-in-LatAm-Research-Study>. [S.l.]. 2015.

DE SÁ, M.; CHURCHILL, E. F. Mobile Augmented Reality: A Design Perspective. In: HUANG, W.; ALEM, L.; LIVINGSTON, M. A. **Human Factors in Augmented Reality Environments**. New York, NY: Springer, 2013. Cap. 6, p. 139-164.

DÜNSER, A.; BILLINGHURST, M.; WEN, J.; LEHTINEN, V.; NURMINEN, A. Exploring the use of handheld AR for outdoor navigation. **Computers & Graphics**, Elmsford, NY, v. 36, n. 8, p. 1084-1095, Dezembro 2012.

DÜNSER, A.; GRASSET, R.; SEICHTER, H.; BILLINGHURST, M. **Applying HCI Principles in AR Systems Design**. Proceedings of the IEEE Virtual Reality Conference: Mixed Reality User Interfaces: Specification, Authoring, Adaptation Workshop (MRUI'07). Charlotte, NC: IEEE. 2007. p. 37-42.

FOWLER, M. **Patterns of Enterprise Application Architecture**. 1ª. ed. [S.l.]: Addison-Wesley Professional, 2002.

FOWLER, M. **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. 3ª. ed. Boston: Addison-Wesley Professional, 2003.

FOWLER, M. GUI Architectures. **Martin Fowler**, 2006. Disponível em: <<http://martinfowler.com/eaDev/uiArchs.html>>. Acesso em: 1 Junho 2015.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Indianapolis: Addison-Wesley, 1994.

GOOGLE INC. Android Developers. **Fragments API Guide**, 2015. Disponível em: <<http://developer.android.com/guide/components/fragments.html>>. Acesso em: 1 junho 2015.

GOOGLE INC. Google Developers. **Google Maps Android API**, 2015. Disponível em: <<https://developers.google.com/maps/documentation/android/?hl=pt-br>>. Acesso em: 1 Junho 2015.

HAMID, A. K. A.; ZHU, W.; KAWAHARA, Y.; TOHRU, A. **Design and Implementation of an AR-assisted Tool for Basic Home Network Management**. IEEE/IPSJ International Symposium on Applications and the Internet. Munich, Bavaria: IEEE. 2011. p. 208-213.

LAYAR. Layar, 2015. Disponível em: <<https://www.layar.com/>>. Acesso em: Junho 2015.

LINS, C.; ARRUDA, E.; NETO, E.; ROBERTO, R.; TEIXEIRA, J. M. **Animar: Augmenting the Reality of Storyboards and Animations**. XVI Symposium on Virtual and Augmented Reality. Salvador, BA: [s.n.]. 2014. p. 106-109.

LIU, C.; HUOT, S.; DIEHL, J.; MACKAY, W. E.; BEAUDOUIN-LAFON, M. **Evaluating the Benefits of Real-time Feedback in Mobile Augmented Reality with Hand-held Devices**. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Austin: ACM. 2012. p. 2973-2976.

MAHRIN, M. N.; STROOPER, P.; CARRINGTON, D. **Selecting Usability Evaluation Methods for Software Process Descriptions**. Proceedings of the 16th Asia-Pacific Software Engineering Conference. Penang: [s.n.]. 2009. p. 523-529.

MARTÍNEZ, H.; BANDYOPADHYAY, P. **Analysis of Four Usability Evaluation Methods Applied to Augmented Reality Applications**. Tampere University of Technology e University of Helsinki. Finlândia. 2014.

MARTÍNEZ, H.; SKOURNETOU, D.; HYPPÖLÄ, J.; LAUKKANEN, S.; HEIKKILÄ, A. Drivers and Bottlenecks in the Adoption of Augmented Reality Applications. **Journal on Multimedia Theory and Applications**, 2014. 20-26.

METAIO. Junho, 2015. Disponível em: <<http://www.juniao.com/>>. Acesso em: Junho 2015.

MILGRAM, P.; TAKEMURA, H.; UTSUMI, A.; KISHINO, F. **Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum**. Telemanipulator and Telepresence Technologies. Boston: [s.n.]. 1994. p. 282-292.

NIELSEN, J. Heuristic Evaluation. In: NIELSEN, J.; MACK, R. L. **Usability Inspection Methods**. New York, NY: Katherine Schowalter, 1994. Cap. 2.

NIELSEN, J.; MOLICH, R. **Heuristic Evaluation of User Interfaces**. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA: ACM. 1990. p. 249-256.

NUDELMAN, G. **Android Design Patterns: Interaction Design Solutions for Developers**. Indianapolis: Wiley & Sons, 2013.

OLSSON, T.; SALO, M. **Online User Survey on Current Mobile Augmented Reality Applications**. Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality. Washington: IEEE Computer Society. 2011. p. 75-84.

PERUTKA, O. Usability testing and heuristic evaluation of the Wikitude navigational application. In: SCHAEFFER, S. E. **Usability Evaluation for Augmented Reality**. Helsinki, Finland: Department of Computer Science, University of Helsinki, 2014. Cap. 3, p. 51-57.

POTEL, M. **MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java**. Taligent Inc. [S.l.]. 1996.

SANTOS, C. G. R.; CARNEIRO, N.; MIRANDA, B.; MEIGUINS, B. S. **Uma Aplicação de Realidade Aumentada Móvel para Ambientes Indoor e Outdoor**. Livro dos Anais do XV Workshop de Realidade Virtual e Aumentada. Marília: Caroline Kraus Luvizotto. 2014. p. 120-126.

SANTOS, C. G.; CARNEIRO, N.; LIMA, B.; MEIGUINS, A.; MEIGUINS, B. S. **Considerações de Projeto e Implementação de uma Aplicação Android de realidade aumentada móvel**. Conferência Íbero-Americana de Computação Aplicada. Porto: [s.n.]. 2014. p. 119-126.

SCHMALSTIEG, D.; WAGNER, D. **Experiences with Handheld Augmented Reality**. Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07). Washington, DC: IEEE Computer Society. 2007. p. 1-13.

SOBOTA, B.; KOREČKO, Š.; HROZEK, F. **Mobile mixed reality**. 11th IEEE International Conference on Emerging eLearning Technologies and Applications. Stary Smokovec: IEEE. 2013. p. 355-358.

SOMMERVILLE, I. **Software Engineering**. 9^a. ed. Boston: Pearson Education, Inc. , 2010.

SRISUPHAB, A.; SILAPACHOTE, P.; SIRILERTWORAKUL, N.; UTARA, Y. **Integrated ZooEduGuide with Multimedia and AR, From the largest living classrooms to wildlife conservation awareness**. TENCON 2014 - 2014 IEEE Region 10 Conference. Bangkok: IEEE. 2014. p. 1-4.

TEEVAN, J.; KARLSON, A.; AMINI, S.; BRUSH, A. J. B.; KRUMM, J. **Understanding the Importance of Location, Time, and People in Mobile Local Search Behavior**. Proceedings of the 13th international conference on human computer interaction with mobile devices and services (MobileHCI '11). New York, NY: ACM. 2011. p. 77–80.

VASSELAI, G. T.; REIS, D. S.; GOMES, P. C. R. **A Case Study of Augmented Reality for Mobile Platforms**. XIII Symposium on Virtual Reality. Uberlandia: IEEE. 2011. p. 225-231.

WASSERMAN, A. I. **Software Engineering Issues for Mobile Application Development**. Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research. Nova York: ACM. 2010. p. 397-400.

WHARTON, C.; RIEMAN, J.; LEWIS, C.; POLSON, P. The Cognitive Walkthrough Method: A Practitioner's Guide. In: NIELSEN, J.; MACK, R. L. **Usability Inspection Methods**. New York, NY, USA: John Wiley & Sons, Inc., 1994. p. 105-140.

WIKITUDE INC. wiktitude See more. **Wiktitude SDK**, 2015. Disponivel em: <<http://www.wiktitude.com/products/wiktitude-sdk/>>. Acesso em: 1 Junho 2015.

APÊNDICE A – Ficha de Avaliação Percurso Cognitivo

Ficha de Avaliação Percurso Cognitivo

Dados do(s) Avaliador(es)

Formação: _____, Curso: _____.

Experiência com desenvolvimento Android ou de aplicações para dispositivos móveis:

Dispositivo utilizado na avaliação: tablet, smartphone.

Apresentação

A avaliação de usabilidade Percurso Cognitivo tem o objetivo de identificar problemas de usabilidade de forma estruturada seguindo uma linha de interações definidas. Esta é realizada por especialistas na construção de interfaces gráficas que devem ser se por no lugar do usuário e avaliar se o usuário é capaz de seguir a linha de raciocínio corretamente.

Cada possível passo que o usuário pode realizar na aplicação é avaliado através de 4 questões, que são respondidas em uma escala definida para cada possível passo do usuário a fim de realizar uma determinada tarefa. Ao final são sugeridas correções para os possíveis problemas encontrados.

Questões a serem respondidas

As questões abaixo devem ser respondidas para cada tarefa de acordo com as tabelas de avaliação.

Q1: O usuário vai tentar realizar ação correta? **Q2:** O usuário vai notar que a ação correta está disponível? **Q3:** O usuário vai associar a ação correta com o efeito desejado? **Q4:** Caso a ação correta seja realizada, o usuário consegue perceber o progresso em direção à realização da tarefa?

Escala das Respostas

1 - Totalmente Improvável; **2** - Improvável, **3** - Talvez, **4** - Provável, **5** - Completamente Provável.

Perfil do Usuário

Usuário que possuem experiência na utilização de smartphones, mas sem experiência com a utilização de aplicações de Realidade Aumentada. Os usuários utilizam smartphones no seu dia-a-dia, com experiência na utilização de aplicações como: WhatsApp, Facebook, e Gmail.

Tarefa 1. Selecione a Biblioteca Central

Passos para realização da tarefa

1. Se a aba Mapa está ativa:
 - a. Buscar pela Biblioteca Central entre os marcadores;
 - b. Tocar no marcador da Biblioteca Central.
2. Senão, se a aba Navegador RA está ativa:
 - a. Rotacionar dispositivo até visualizar o marcador da Biblioteca Central;
 - b. Tocar no marcador da Biblioteca Central.

Ficha de Avaliação

Tarefa 1. Selecione a Biblioteca Central.				
Questões	Passo 1.a	Passo 1.b	Passo 2.a	Passo 2.b
Q1				
Q2				
Q3				
Q4				

Tarefa 2. Visualize uma mídia do Centro de Convenções

Passos para realização da tarefa

1. Se a aba Mapa está ativa:
 - a. Buscar pelo Centro de Convenções entre os marcadores;
 - b. Tocar no marcador do Centro de Convenções;
 - c. Tocar no botão de uma mídia.
2. Senão, se a aba Navegador RA está ativa:
 - a. Rotacionar dispositivo até visualizar o marcador do Centro de Convenções;
 - b. Tocar no marcador do Centro de Convenções;
 - c. Tocar no botão de uma mídia.

Ficha de Avaliação

Tarefa 2. Visualize uma mídia do Centro de Convenções.						
Questões	Passo 1.a	Passo 1.b	Passo 1.c	Passo 2.a	Passo 2.b	Passo 2.c
Q1						
Q2						
Q3						
Q4						

Tarefa 3. Trace uma rota até o Hospital Betina Ferro

Passos para realização da tarefa

1. Se a aba Mapa não está ativa:
 - a. Selecionar aba Mapa.
2. Buscar pelo Hospital Betina Ferro entre os marcadores;
3. Tocar no marcador do Hospital Betina Ferro.

Ficha de Avaliação

Tarefa 3. Trace uma rota até o Hospital Betina Ferro			
Questões	Passo 1.a	Passo 2	Passo 3
Q1			
Q2			
Q3			
Q4			

Tarefa 4. Encontre a direção da Reitoria pelo Navegador RA

Passos para realização da tarefa

1. Se a aba Navegador RA não está ativa:
 - a. Selecionar aba Navegador RA.
2. Rotacionar dispositivo até visualizar o marcador da Reitoria.

Ficha de Avaliação

Tarefa 4. Encontre a direção da Reitoria pelo Navegador RA		
Questões	Passo 1.a	Passo 2
Q1		
Q2		
Q3		
Q4		

Tarefa 5. Diminua o raio de alcance para visualizar somente os pontos mais próximos

Passos para realização da tarefa

1. Se a aba Navegador RA não está ativa:
 - a. Selecionar aba Navegador RA.
2. Deslize o slider lateral para baixo até certo ponto.

Ficha de Avaliação

Tarefa 5. Diminua o raio de alcance para visualizar somente os pontos mais próximos		
Questões	Passo 1.a	Passo 2
Q1		
Q2		
Q3		
Q4		

Tarefa 6. Realize a leitura de um QR Code, visualize as mídias e fale do que se trata

Passos para realização da tarefa

1. Toque no botão de leitura de QR Code;
2. Aponte a câmera para um código QR Code até que a leitura seja realizada;
3. Toque em uma mídia.

Ficha de Avaliação

Tarefa 6. Realize a leitura de um QR Code, visualize as mídias e diga do que se trata			
Questões	Passo 1	Passo 2	Passo 3
Q1			
Q2			
Q3			
Q4			

Sugestões de Melhorias

A large, empty rectangular box with a thin black border, intended for providing suggestions or feedback. The box is currently blank.

APÊNDICE B – Ficha de Avaliação Heurística

Ficha de Avaliação Heurística

Dados do (s) Avaliador (es)

Formação: _____, Curso: _____.

Experiência com desenvolvimento Android ou de aplicações para dispositivos móveis:

Dispositivo utilizado na avaliação: tablet, smartphone.

Apresentação

A avaliação heurística tem por objetivo avaliar a usabilidade da aplicação, verificando se as interfaces da aplicação estão coerentes com as heurísticas definidas, e propor possíveis soluções para as interfaces que apresentarem problemas.

O avaliador deve ter em mente as diretrizes conceituais apresentadas e procurar na aplicação problemas identificados na interface gráfica que violem essas diretrizes de acordo com a ficha de avaliação. As diretrizes são apresentadas na próxima página.

Perfil de Usuário

Os usuários desta aplicação possuem experiência na utilização de smartphones, mas sem experiência com a utilização de aplicações de Realidade Aumenta. Os usuários utilizam smartphones no seu dia-a-dia, com experiência na utilização de aplicações como: WhatsApp, Facebook, e Gmail.

Heurísticas

Nº	Heurísticas
1	Visibilidade dos estados da aplicação
	O sistema deve sempre manter o usuário informado sobre o que está acontecendo. Informações sobre a base de dados utilizada e categoria dos pontos de interesse devem sempre estar visíveis, assim como o estado do rastreamento e orientação. Todas as informações devem ser apresentadas através de um <i>feedback</i> apropriado em um tempo razoável.
2	Equilíbrio entre o sistema e o mundo real
	A aplicação deve mostrar a orientação e posição corretas do usuário, do mesmo modo que a posição e distância para pontos de interesses visíveis. As unidades de distância devem respeitar as unidades que são utilizadas no local que o usuário está.
3	Controle do usuário e liberdade
	Permitir que o usuário troque livremente entre banco de dados e categorias de pontos de interesse. Permitir também a escolha entre navegação com mapas e com vídeo sobreposto. Retornar ao mesmo estado depois de trocar aplicações. Considere retorna ao estado padrão depois de uma longa interrupção.
4	Consistência e Padrões
	Os controles da interface do usuário devem respeitar as convenções usadas em dispositivos móveis. Todos os símbolos gráficos devem ser claros e intuitivos.
5	Prevenção de erros
	A aplicação deve desenvolvida de modo que previna erros de acontecer. Se não for possível, a aplicação deve ser capaz de se recuperar.
6	Reconhecimento ao invés de lembrança
	Mostrar para o usuário "onde" ele está na aplicação e fazer a navegação através da aplicação simples. As funções mais importantes (como busca e troca entre navegação com mapas e vídeo sobreposto) devem ser sempre acessíveis.
7	Flexibilidade eficiência de uso
	Faça as funções/banco de dados favoritos do usuário facilmente acessíveis. Permitir que o usuário configure atalhos para essas configurações.
8	Estética e <i>design</i> minimalista
	As telas não podem conter informações irrelevantes ou desnecessárias.
9	Ajude o usuário a reconhecer, diagnosticar e corrigir erros
	Mensagens de erro devem ser expressas em linguagem plana (sem códigos), indicando precisamente o problema e construtivamente sugerir uma solução.
10	Ajuda e documentação
	Prover alguma ajuda/documentação. Ajuda rápida e documentação detalhada devem estar disponíveis.

Avaliação

#	Nº da(s) Heurística(s)	Problema	Possível Solução
1			
2			
3			
4			
5			
6			

APÊNDICE C – Resultados da Avaliação Percurso Cognitivo

Os resultados da avaliação Percurso Cognitivo estão dispostos nas tabelas abaixo, primeiro para *tablet* e depois para *smartphone*.

Tablet

Tarefa 1.

Questões	Passo 1.a	Passo 1.b	Passo 2.a	Passo 2.b
Q1	5	5	3	4
Q2	5	4	3	4
Q3	3	4	4	5
Q4	4	5	4	4

Tarefa 2.

Questões	Passo 1.a	Passo 1.b	Passo 1.c	Passo 2.a	Passo 2.b	Passo 2.c
Q1	5	5	5	3	4	5
Q2	5	4	5	3	4	5
Q3	3	4	5	4	4	5
Q4	4	5	5	5	5	5

Tarefa 3.

Questões	Passo 1.a	Passo 2	Passo 3
Q1	5	5	3
Q2	5	5	4
Q3	5	5	5
Q4	5	5	5

Tarefa 4.

Questões	Passo 1.a	Passo 2
Q1	5	4
Q2	5	4
Q3	5	4
Q4	5	5

Tarefa 5.

Questões	Passo 1.a	Passo 2
Q1	5	4
Q2	5	4
Q3	5	3
Q4	5	3

Tarefa 6.

Questões	Passo 1	Passo 2	Passo 3
Q1	5	5	3
Q2	5	5	3
Q3	5	3	5
Q4	5	2	5

Smartphone

Tarefa 1.

Questões	Passo 1.a	Passo 1.b	Passo 2.a	Passo 2.b
Q1	4	4	4	2
Q2	4	4	3	2
Q3	5	5	5	4
Q4	5	5	2	5

Tarefa 2.

Questões	Passo 1.a	Passo 1.b	Passo 1.c	Passo 2.a	Passo 2.b	Passo 2.c
Q1	4	4	5	4	2	4
Q2	4	4	5	3	2	5
Q3	5	5	5	5	4	5
Q4	5	5	5	2	5	5

Tarefa 3.

Questões	Passo 1.a	Passo 2	Passo 3
Q1	3	4	4
Q2	5	4	4
Q3	5	5	5
Q4	5	5	5

Tarefa 4.

Questões	Passo 1.a	Passo 2
Q1	4	4
Q2	5	3
Q3	5	5
Q4	5	5

Tarefa 5.

Questões	Passo 1.a	Passo 2
Q1	2	4
Q2	5	5
Q3	5	4
Q4	5	4

Tarefa 6.

Questões	Passo 1	Passo 2	Passo 3
Q1	4	4	4
Q2	5	5	4
Q3	5	5	5
Q4	5	5	5

ANEXO A - Arquitetura dos Artigos Seleccionados

Neste anexo estão dispostas as arquiteturas apresentadas nos trabalhos seleccionados em bases científicas. A seleção foi descrita na subseção 2.3.1. Abaixo estão as imagens que foram extraídas destes artigos.

Chen e Colaboradores (2013) apresentaram uma arquitetura para aplicações RAM com base na tecnologia cloud para identificar e rastrear objetos reais e sobrepor com elementos virtuais (Figura 36). Apesar de a arquitetura apresentada ter definido uma comunicação com a Internet para buscar identificar o mundo real, não uma proposta sobre múltiplas visões, expansão de funcionalidades e sobre o intercâmbio entre dados locais e dados remotos.

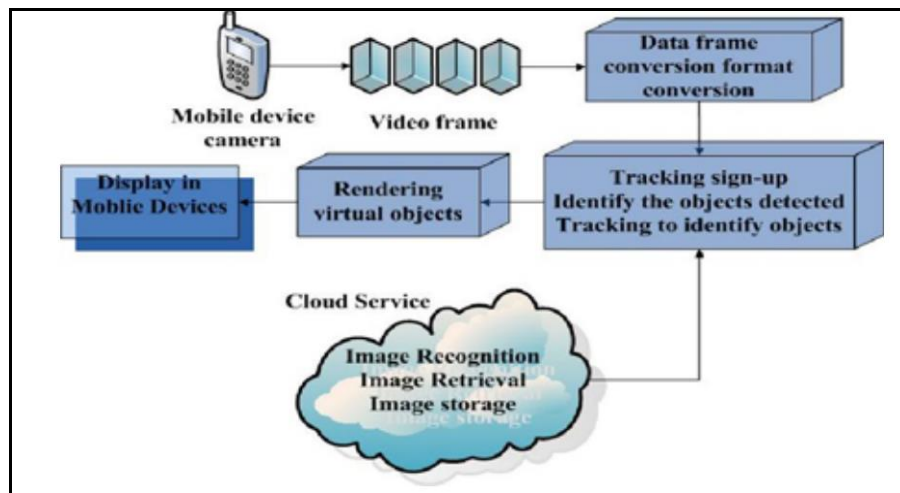


Figura 36. Arquitetura para o desenvolvimento de aplicações RAM com *cloud computing* (CHEN, XIANG, *et al.*, 2013).

Chen, Peng e Tseng (2011) apresentaram uma proposta de aplicação para o gerenciamento de corridas de bicicleta em grupo, utilizando a RAM. A arquitetura apresentada (Figura 37) aborda mais aspectos de comunicação em rede do que aspetos de implementação em si.

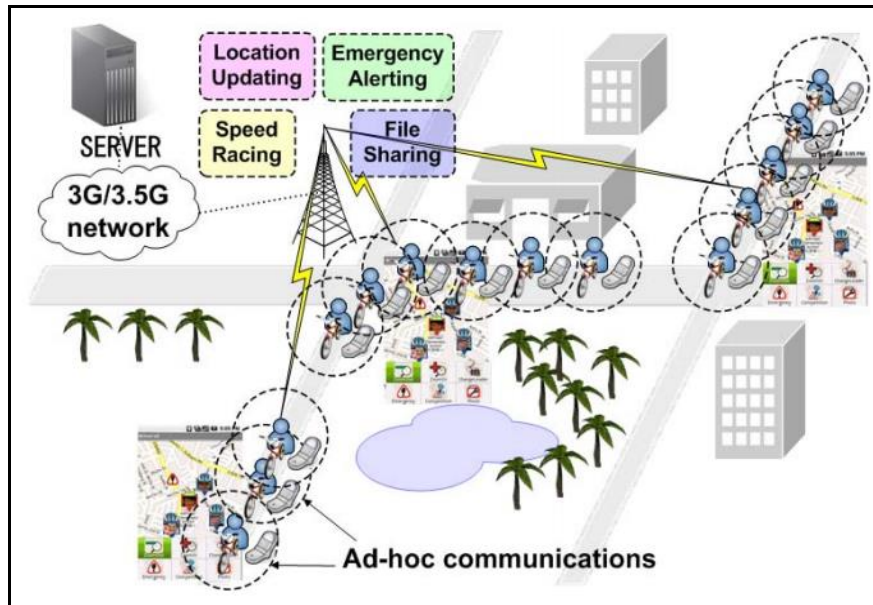


Figura 37. Arquitetura da aplicação que auxilia os ciclistas a pedalam em grupos (CHEN, PENG e TSENG, 2011).

Hamid e Colaboradores (2011) apresentaram uma aplicação que auxilia a configuração e o gerenciamento de redes domésticas com a utilização da RAM. A arquitetura apresentada (Figura 38) tem foco no propósito da aplicação e não prevê a adição de novas visões aos usuários, sendo específica para utilização com marcadores.

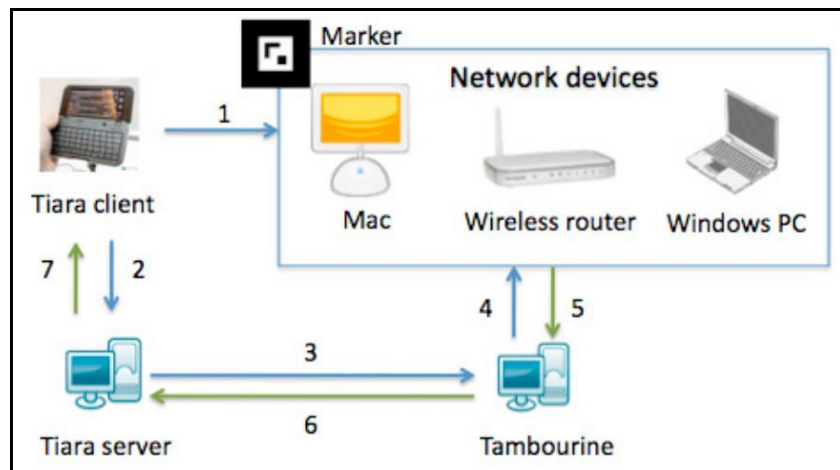


Figura 38. Arquitetura da aplicação que auxilia usuários a configurar e gerenciar sua rede doméstica através de recursos RAM (HAMID, ZHU, *et al.*, 2011).

Lins e Colaboradores (2014) apresentaram uma aplicação chamada Animar que ajuda os usuários a desenvolver desenhos a mão. Esta aplicação sobrepõe os traços do desenho no papel

fazendo com que os usuários se baseiem nessas informações virtuais para desenhar. A arquitetura da aplicação (Figura 39) apresenta alguns módulos de código, mas está focada no propósito da aplicação, sendo assim, não prevê a adição de novas visões, de expansão da aplicação ou de flexibilidade dos dados.

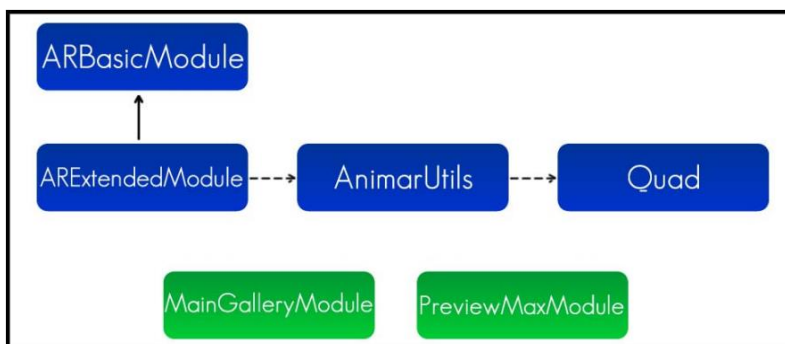


Figura 39. Arquitetura do aplicativo RAM Animar (LINS, ARRUDA, et al., 2014).

Sobota e Colaboradores (2013) apresentaram uma proposta de implementação para aplicações RAM que utilizam a biblioteca Vuforia em conjunto com a biblioteca min3d (Figura 40). Segundo os autores a biblioteca min3d tem melhor desempenho para desenha elementos 3d do que a biblioteca Vuforia. Apesar de a arquitetura ser genérica, esta prevê apenas uma funcionalidade, não possui características de expansão e coordenação de múltiplas visões, e não aborda como os dados são obtidos.

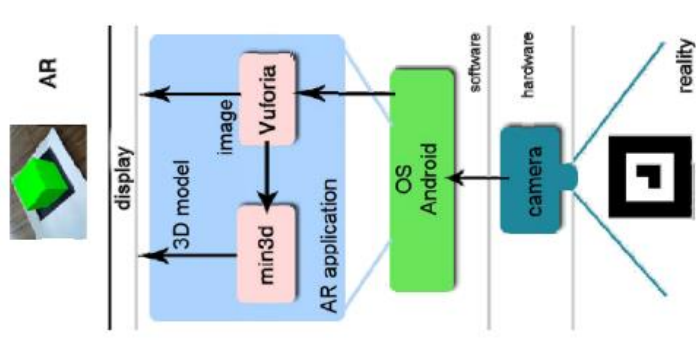


Figura 40. Arquitetura para aplicações RAM utilizarem a biblioteca Vuforia em conjunto com a biblioteca min3d (SOBOTA, KOREČKO e HROZEK, 2013).

Srisuphab e Colaboradores (2014) apresentam uma aplicação para a descoberta e navegação de POIs em um zoológico utilizando a tecnologia RAM para atrair a atenção dos visitantes (Figura 41). Vasselai e Colaboradores apresentam uma arquitetura para aplicações RAM que

utilizam os sensores dos dispositivos móveis (Figura 42). As duas arquiteturas são semelhantes e possuem um módulo central chamado *Engine*, que é responsável por gerenciar todas as visões, sensores, dados interface gráfica, etc. Apesar dessa abordagem de envolver muitos recursos, ela não tem uma boa modularização de código e não prevê uma coordenação de múltiplas visões.

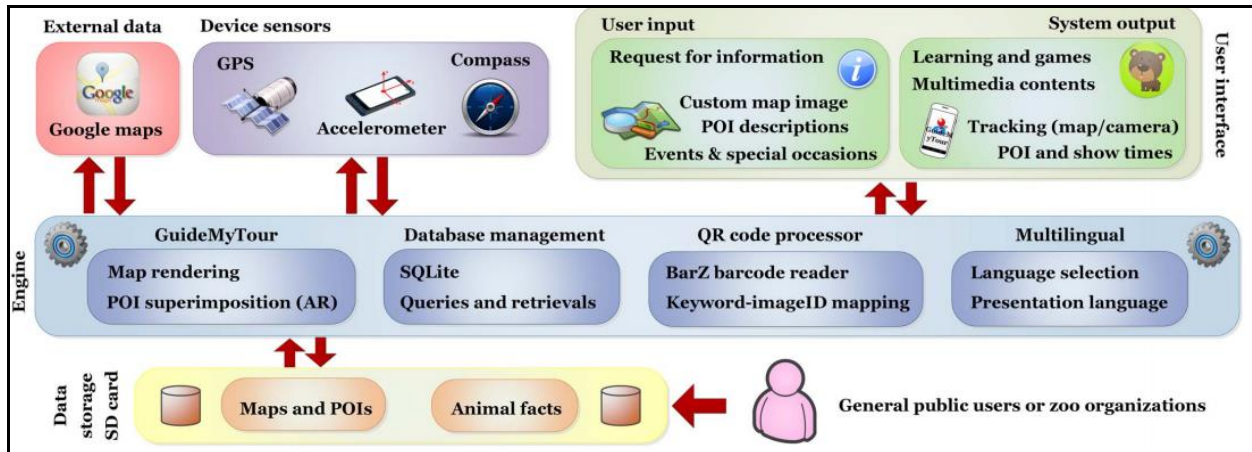


Figura 41. Arquitetura da aplicação que ajuda os visitantes de um zoológico a encontrarem os animais e obterem mais informações sobre eles. (SRISUPHAB, SILAPACHOTE, *et al.*, 2014).

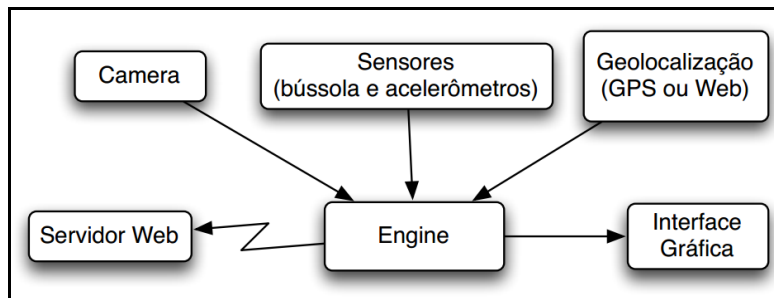


Figura 42. Arquitetura para aplicações RAM que utilizam sensores e a câmera do dispositivo móvel para desenhar os elementos virtuais (VASSELAI, REIS e GOMES, 2011).