Aplicações Android de Realidade Aumentada em Arquitetura Extensível, Flexível e Adaptável

Alternative Title: Android Augmented Reality Applications in Extensible, Flexible, and Adaptable Architecture

Tiago Araújo
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Pará
tiagodavi70@gmail.com

Nikolas Carneiro
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Pará
nikolas.carneiro@gmail.com

Brunelli Miranda
Faculdade de Computação
Universidade Federal do Pará
brunelli.miranda@gmail.com

Carlos Santos
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Pará
gustavo.cbcc@gmail.com

Bianchi Meiguins
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Pará
bianchi@ufpa.br

RESUMO

A proposta deste trabalho é apresentar uma arquitetura para aplicações Android de Realidade Aumentada Móvel (RAM) que seja extensível, flexível e adaptável. Extensível por permitir a incorporação de novas funcionalidades na aplicação; flexível por permitir a troca de conteúdo/dados gerando aplicações personalizadas; e adaptável aos diversos tamanhos de telas que os dispositivos móveis apresentam. Para isso, foi considerada uma arquitetura baseada no padrão MVC (Model, View, Controller) adaptado para contexto da plataforma móvel Android, nesta adaptação a lógica de negócio sai da camada de controle e vai para camada de visão, sendo assim, a camada de controle fica responsável apenas por gerenciar as requisições entre as camadas modelo e visão, garantindo uma modularização dessa aplicação. Também foi utilizado o padrão de interface de usuário Fragments, visando uma melhor adaptação aos diversos tamanhos de telas dos dispositivos móveis. Além disso, foi aplicado o padrão de projeto Proxy Remoto, para abstração da origem dos dados (local ou remota), e o padrão de projeto Facade para facilitar a realização de consultas e filtros nos dados. Por fim, são apresentados cenários de uso com três tipos diferentes de dados e dispositivos com tamanhos de telas diferentes para validação da arquitetura proposta.

Palavras-Chave

Realidade Aumentada, Aplicações Móveis, Arquitetura.

ABSTRACT

This work aims to present an architecture for Android Mobile Augmented Reality (MAR) applications which has to be extensible, flexible and adaptable. Extensible to allow the addition of new functionalities in the application; flexible to allow the change of content/data generating custom applications; and adaptable to the many mobile devices screen sizes. For this reason, we considered an architecture based on the MVC pattern adapted to the context of Android mobile platform, in this adaptation the business logic get out the controller layer and reaches the view layer, in so doing, the controller layer becomes responsible only for managing the request between the model and view layers, ensuring the application modularization. Also, the Fragments user interface pattern was utilized, aiming to a better adaptation to the many mobile devices screen sizes. Furthermore, we applied the Remote Proxy design pattern, for abstraction of data source (local or remote), and the Facade design pattern to facilitate the use of queries and filters in data. Finally, we present usage scenarios with different data and screen size devices to validate the proposed architecture.

SBSI 2015, May 26–29, 2015, Goiânia, Goiás, Brazil.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures – *Domain-specific architectures, Patterns.*

General Terms

Management, Design.

Keywords

Augmented Reality, Mobile Applications, Architecture.

1. INTRODUÇÃO

A evolução tecnológica dos smartphones e tablets, principalmente no que diz respeito a qualidade de renderização e câmera, uso do GPS e sensores, como bússola e acelerômetro, viabilizou a utilização de aplicações de Realidade Aumentada nestes dispositivos. As áreas de aplicações da Realidade Aumentada Móvel (RAM) são as mais variadas possíveis, como as áreas de jogos, viagens, compras on-line, ciências, saúde, redes sociais, entre outras [11].

Com todos esses recursos tecnológicos disponíveis o desafio passa a ser o desenvolvimento de aplicações que utilizem esses recursos de forma integrada, seja em aplicações nativas ou híbridas (aplicações web para dispositivos móveis). Outras questões relevantes consideram as restrições e características diferentes de hardware e software dos vários dispositivos, questões de segurança, variabilidades na interface gráfica do usuário, o consumo de energia do dispositivo, etc [12].

Apesar do crescimento acelerado desta área e de inúmeras aplicações disponíveis para uso, não há padrões de desenvolvimento definidos para aplicações RAM, a maioria das aplicações apresenta pouca flexibilidade de dados não permitindo seu uso em outros domínios, há necessidade de evolução na precisão das tecnologias de localização e reconhecimento de padrões, e pouco espaço para apresentação das informações virtuais [8].

Este trabalho propõe um modelo arquitetural para aplicações de Realidade Aumentada Móvel, voltado para plataforma Android, que permite a extensibilidade da aplicação com baixo custo de implementação, adaptação a uma diversidade de dispositivos com tamanhos diferentes de tela, e flexibilidade na obtenção e utilização dos dados.

Neste sentido, foi utilizada uma adaptação do padrão MVC 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 Remoto, para abstração da origem dos dados (local ou remota), e o padrão de projeto Facade para facilitar a realização de consultas e filtros nos dados. Também foram seguidas as boas práticas e principais diretrizes para o desenvolvimento de aplicações móveis e aplicações de RA.

Por fim, serão apresentados cenários de uso com dados e dispositivos com tamanho de telas diferentes para validação da arquitetura proposta. O primeiro cenário considera um roteiro artístico das obras da Universidade Federal do Pará, o segundo considera um roteiro educacional no Museu Emílio Goeldi e o terceiro considera um roteiro turístico com atrações turísticas da cidade de Belém-Pará-Brasil. As aplicações móveis têm como principais funcionalidades: guiar os usuários em ambiente aumentado por GPS, guiar o usuário em mapas 2D, e apresentar conteúdos aumentados através de reconhecimento de padrões QR Code. O ambiente aumentado é apresentado através da tela do dispositivo combinando a cena real com conteúdos virtuais (imagem, texto, vídeo, objetos 2D/3D).

2. REALIDADE AUMENTADA MÓVEL

A Realidade Aumentada Móvel (RAM) é a aplicação do conceito de RA em plataformas móveis, que significa misturar conteúdos virtuais e cenas do ambiente real na tela do dispositivo móvel [6].

Na RAM a cena do mundo real é capturada pela câmera do dispositivo móvel e associada a uma camada de conteúdo para apresentação na tela do dispositivo, a aplicação cria outra camada com conteúdos virtuais, e então sobrepõe uma camada à outra para apresentação ao usuário.

Normalmente, as aplicações de RAM apresentam duas abordagens para misturar conteúdo virtual nas cenas reais: marcadores ou localização. A abordagem de marcadores utiliza um software de reconhecimento de um padrão em particular (como QR Code) para selecionar o conteúdo, e a abordagem por localização faz uso do GPS do dispositivo e da orientação da câmera para definir o que deve ser apresentado ao usuário.

Como principais características de aplicações RAM, têm-se [7]:

- Apresentação do ambiente aumentado na tela dos dispositivos móveis;
- Tecnologia de rastreamento: GPS, e reconhecimento de padrões em imagens;
- Sistema gráfico é responsável pela renderização dos objetos virtuais em cenas aumentadas;
- Sistema de mistura de mundos: faz a mistura da cena real com objetos virtuais.

Os desafios no desenvolvimento de aplicações RAM são [8, 12]:

- 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

Algumas diretrizes devem ser consideradas para o desenvolvimento de aplicações RAM, são elas [2, 3]:

- 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;
- Avaliação: adaptar diretrizes de avaliação de usabilidade para aplicações móveis e RA, avaliar um domínio por vez, e utilizar usuários reais.

3. ARGUIDE

3.1 Aspectos Conceituais

ARguide é uma aplicação de realidade aumentada móvel que pode receber vários tipos de conteúdos georeferenciados e fornecer aos usuários este conteúdo de forma aumentada. Este aplicativo projetado para utilizar os recursos disponíveis em dispositivos móveis com sistema operacional Android, como: sensores, câmera, bússola, GPS, acesso a Internet, etc e tornar o conteúdo

da aplicação acessível e explorável por meio da realidade aumentada.

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

A aplicação foi arquitetada de modo a oferecer os recursos de uma aplicação RAM para o contexto personalizado do fornecedor de conteúdo da aplicação, sem a necessidade de programação para a plataforma Android, entretato nesta versão da aplicação é necessário algum conhecimento em configurações de aplicações Android. Ou seja, nesta versão da aplicação o usuário insere os dados (contexto personalizado) em um projeto genérico do ARguide e gera o arquivo executável.

A Figura 1 apresenta o fluxo de construção de uma aplicação ARguide com contexto personalizável. O primeiro passo trata-se dos dados georeferenciados fornecidos por uma pessoa ou organização, neste caso pode ser um hospital, um governo, uma entidade, uma empresa etc, que deseja fornecer um conteúdo aumentado para seus usuários. No segundo passo o conteúdo é armazenado em diretórios específicos dentro de um projeto genérico do ARguide e é gerado um executável que pode ser instalado em dispositivos móveis com o sistema operacional Android. No terceiro passo o fornecedor do conteúdo terá uma aplicação de realidade aumentada móvel com os seus dados fornecidos na primeira etapa dispondo dos recursos de realidade aumentada móvel para visitação, exploração e descoberta desse conteúdo.

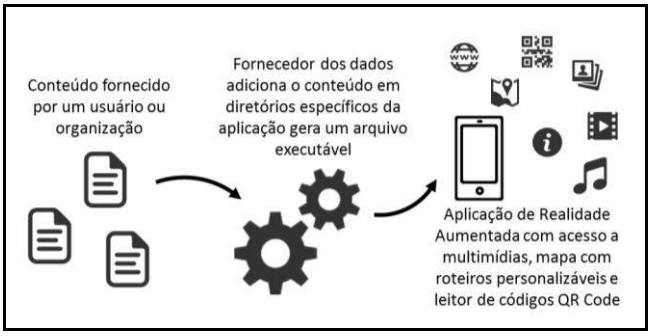

Figura 1. Ideia do funcionamento da aplicação para gerar aplicações personalizadas.

A aplicação gerada, independente do conteúdo adicionado, possui três partes, são elas: aba mapa, onde podem ser visualizados POIs (Pontos de Interesse do inglês *Points of Interest*) e rotas em um mapa 2D; aba navegador de RA, que permite visualizar pontos de interesse com recursos de GPS, bússola e acelerômetro; e um leitor QR Code, que permite recuperar conteúdos associados a QR Codes.

3.2 Arquitetura

O ARguide foi projetado para oferecer os recursos de uma aplicação RAM independente dos dados que forem fornecidos. Desse modo, para garantir que o ARguide ofereça tais recursos é necessário garantir uma boa modularidade de código entre os componentes que realizam as trocas de dados na aplicação, desde o nível arquitetural.

Na arquitetura do ARguide decidiu-se utilizar uma abordagem semelhante ao padrão MVC (*Model*, *View*, *Controller*, que pode ser traduzido como Modelo, Visão e Controlador), porém com algumas modificações que fornecem uma melhor adequação ao estilo de programação de aplicações Android com a linguagem Java.

Esta decisão arquitetural provocou uma ligeira, porém efetiva modificação no papel de cada componente do padrão MVC. Esta modificação considera reduzir a quantidade de lógica aplicada no controlador e passar essa lógica para a visão, dando assim mais autonomia para as visões e uma menor dependência do controlador. A Figura 2 ilustra como essa ideia foi aplicada no padrão MVC.

Devido a essa alteração no comportamento do componente Controlador decidiu-se alterar o nome seu nome para Roteador (Router), uma vez que este componente não possui mais lógica fazendo apenas o papel de encontrar quais modelos devem ser alterados, quando isso for requisitado. Outro papel importante associado ao Roteador é a separação entre os componentes Visão e Modelo permitindo que alterações em um desses componentes não reflitam no outro.

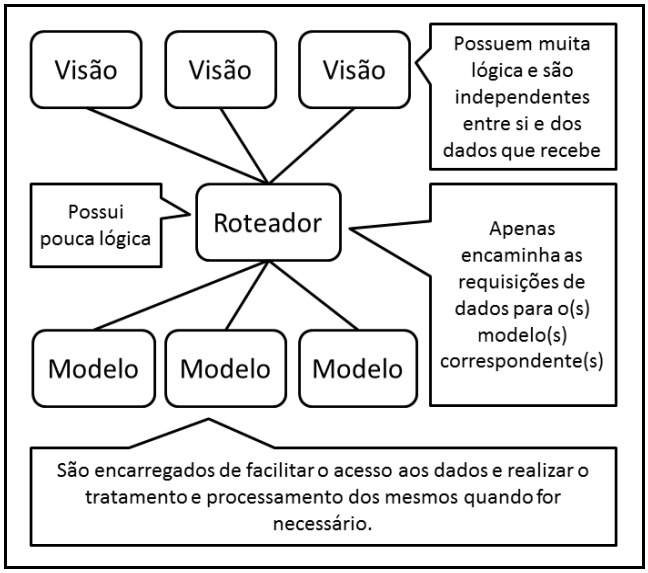

Figura 2. Modificação proposta no padrão MVC.

Aplicações Android possuem como *entry point*[1] uma classe Activity, que não só define o que está sendo visto na tela (por meio das classes View[2] e de layouts em XML), mas também é responsável pela manutenção do contexto da aplicação e pelo ciclo de vida de uma janela na aplicação. Sendo assim, muita responsabilidade e lógica da aplicação estão diretamente associadas com a classe Activity, logo essa modificação realizada no padrão MVC adequa melhor esta arquitetura ao estilo de programação de aplicações Android com Java.

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

[1] *Entry Point*: primeiro arquivo a ser executado em uma aplicação.

[2] *View*: Classe da API Android responsável pelo desenho elementos gráficos na tela do usuário.

usuário. Com a modificação proposta o Controlador deixar e ter essa lógica e passa apenas a direcionar chamadas para os respectivos Modelos quando for solicitado. Dessa maneira as Visões passam a ter mais lógica e apenas vão disparar chamadas para o Roteador/Controlador quando necessitarem realizar alguma operação de modificação de dados. Essa medida permite adicionar e remover novas Visões sem a necessidade de modificações no Roteador ou no Modelo.

Uma das inspirações para a modificação proposta foi o padrão MVP que também é uma variante do padrão MVC, mas é aplicado para construções de interfaces gráficas com o usuário. O padrão MVP define o *entry point* na Visão, que considera a mesma como uma estrutura de Widgets[3] que devem ser responsivos às interações do usuário, o Apresentador (*Presenter*) como responsável por mediar a comunicação entre o Modelo (*Model*) e a Visão (*View*), e o Modelo, trata dos dados e a lógica de acesso aos dados [10]. Uma das formas conhecidas do MVP é o MVP *Passive View* [4] que canaliza todas as interações e os retornos do Modelo para as Visões pelo Apresentador, mas mantendo parte da autonomia presente nas Visões. Entretanto neste padrão é criada uma instância de Apresentador para cada Visão deixando o projeto menos expansível, uma vez que para adicionar uma funcionaldade nova é necessário criar uma nova Visão e seu respectivo Apresentador.

Além dessa arquitetura neste projeto foi utilizado o padrão de interfaces gráficas de usuário Fragmentos (*Fragments*) [9], possibilitando 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.

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, que pode sofrer alterações, como remoção do mapa ou adição de uma nova forma de interagir, sem alterar o funcionamento dos outros meios de interação. 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 arquitetura proposta é apresentada de maneira geral na Figura 3, 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**: camada responsável por manipular os eventos da interface do usuário, apresentar os dados vindos da camada Modelo e encaminhar as interações do usuário para a camada Roteador, apenas quando for necessário realizar alterações em dados. Possui três sub-módulos intercambiáveis:
  - **Mapa**: responsável por apresentar um mapa 2D, pontos de interesses, e rotas;
  - **Navegador de RA**: responsável por apresentar conteúdo aumentado no dispositivo, tendo como referência o GPS e a orientação da câmera do dispositivo (capturada através da bússola e acelerômetro);
  - **Leitor de QR Code**: responsável pela leitura de um marcador tipo QR Code, e apresentar conteúdo virtual aumentado do marcador na tela do dispositivo associado ao QR Code especifico.
- **Roteador**: responsável pela comunicação entre o modelo e a visão, permitindo um intercâmbio fracamente dependente entre as duas partes. Essa característica permite que alterações realizadas na visão não acarretem alterações no modelo e o mesmo vale para alterações no modelo.
- **Modelo**: responsável pela definição dos dados a serem apresentados na Visão. É composto por:
  - **Facilitador (Facade)**: auxilia a realização de consultas e modificações mais facilmente, este é o padrão de projetos Facade descrito por Gamma e colaboradores [5];
  - **Proxy Modelo**: Abstrai as requisições aos dados. Para seus clientes não importa de onde vieram os dados, este padrão se encarrega de delegar a função de recuperação de dados para um de seus sub-módulos, este módulo trata-se do padrão de projetos Proxy Remoto [5].
    - **Modelo Local**: quando os dados requisitados estão armazenados no dispositivo do usuário este módulo recupera os dados e passa para a camada Roteador;
    - **POIs Proxy**: quando os dados não estão armazenados no dispositivo do usuário este módulo recupera os dados através da rede, armazena os dados na memória local do dispositivo.

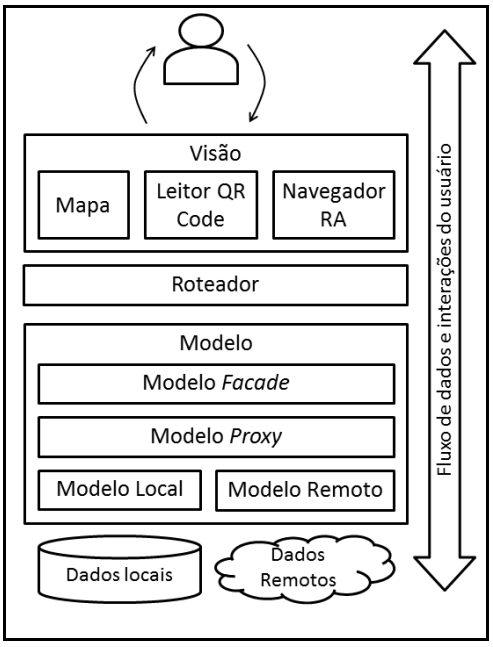

Figura 3. Arquitetura proposta e o fluxo de dados e interações do usuário

[3] *Widgets*: Componentes de interface gráfica do usuário. Exemplo: Botões, check-boxs, caixas de texto, etc.

3.3 Funcionalidades

O ARguide possui resumidamente três tipos de interações, que são: interações com Mapa, Navegador RA (Realidade Aumentada) e QR Code. As interações com Mapa e Navegador RA foram divididas em abas na aplicação e as interações com leitor de QR Code são acionadas através de um botão.

Com a arquitetura proposta torna-se menos custoso adicionar ou retirar uma funcionalidade da aplicação. Como os dados da aplicação podem variar de acordo com o fornecedor, pode ser que o mesmo deseje adicionar novas funcionalidades e não queira uma funcionalidade já existente. Como as funcionalidade estão associadas a camada de Visão, basta remover ou adicionar uma nova Visão para ter esse efeito. Embora, nesta versão seja necessário algumas habilidades de programação para remover e adicionar funcionalidade esta tarefa foi simplificada com a arquitetura proposta.

Abaixo é descrita cada uma dessas funcionalidades e o que o usuário pode realizar por meio dessas funcionalidades:

3.3.1 Aba Mapa

A aba Mapa contém um mapa desenvolvido com a tecnologia Google Maps, que permite aos usuários a realização de interações em dispositivos móveis que viabilizam a visualização, busca e seleção de POIs em um mapa 2D. A partir dessa aba o usuário pode realizar as seguintes operações:

- **Zoom e Pan**: são operações de ampliar e mover o mapa na tela do dispositivo, estas operações são importantes tanto para dar uma visão geral dos POIs, assim como, visualizar detalhes de um ou alguns POIs.
- **Selecionar Rota**: uma vez que dados inseridos na aplicação são georeferenciados, o aplicativo é capaz de traçar uma rota de visitação POIs de acordo com a posição atual do usuário.
- **Selecionar POI**: com os pontos de interesse marcados no mapa o usuário pode clicar/tocar em um POI realizando uma seleção. Quando um POI é selecionado é apresentado ao usuário um conjunto de conteúdos multimídia sobre esse POI ao usuário.
- **Acompanhar sua localização em relação a Rota**: trata-se do usuário poder acompanhar sua trajetória em relação a rota que foi sugerida pela aplicação. Como a localização atual do usuário é mostrada no mapa ele pode saber o que já foi visitado e o que ainda falta visitar.

3.3.2 Aba Navegador RA

A aba Navegador RA apresenta ao usuário o stream de sua câmera (vídeo), representando o mundo real, acrescido de marcadores virtuais que são posicionados em frente aos POIs, marcando a posição 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. Esta é uma abordagem semelhante à *Magic Lens*, descrita por Cawood [1], onde a posição de cada POI é dada pela informação de geolocalização do mesmo.

A partir dessa aba o usuário pode realizar as seguintes operações:

- **Visualizar Posição/Direção do POI**: permite ao usuário encontrar POIs baseado na sua localização atual, obtida a partir do GPS do dispositivo móvel, ao direcionar sua câmera para os POIs disponíveis na aplicação.
- **Selecionar POI**: permite que o usuário da aplicação clique/toque no ícone virtual posicionado na direção do POI selecionando o mesmo. Ao selecionar um POI o usuário tem acesso aos seus conteúdos multimídia.
- **Ajustar Raio Alcance**: permite que o usuário configure um raio de interesse para a exibição dos ícones virtuais. Este raio é medido em quilômetros e tem como centro a posição atual do usuário. Todos os POIs que estiverem a uma distância maior que esse raio não seram visualizados pelo usuário. Esta operação pode ser utilizada para que o usuário possa se concentrar na busca por elementos mais próximos a ele.
- **Visualizar Radar**: a aplicação disponibiliza um pequeno radar no canto superior direito da tela. Esse 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.

3.3.3 Aba Leitor de QR Code

A aba de 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 extraindo as informações contidas em tais marcadores. Ao receber as informações extraídas pelo serviço a aplicação pode disponibilizar o acesso à lista de informações do ponto de interesse em questão (se o POI for identificado) ou mostrar uma mídia diretamente.

3.3.4 Funcionalidades Gerais

As funcionalidades gerais da aplicação estão sempre disponíveis para o usuário independente de qual aba esteja ativa. Estas funcionalidades objetivam auxiliar o usuário em atividades gerais e em momentos de dúvida. As funcionalidades gerais estão descritas abaixo:

- **Visualizar Conteúdo Multimídia**: Esta operação está sempre disponível para o usuário quando ele seleciona um POI em qualquer uma das abas, ela permite que o usuário visualize arquivos 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 ao mesmo.
- **Limpar Seleção**: esta opção limpa a seleção realizada pelo usuário em um momento anterior. Esta opção não é obrigatória para realizar outra seleção, mas remove da tela elementos que podem não interessar mais para o usuário.
- **Ajuda**: auxilia o usuário dando dicas de utilização da aplicação. A ajuda oferece uma descrição completa da utilização da aplicação e suas funcionalidades.

3.4 Geração de Aplicações Personalizadas

Com ARguide é possível gerar aplicações de realidade aumentada com dados personalizados a partir de um projeto (códigos fonte da aplicação organizados em um conjunto de pastas) genérico, ou seja sem os dados. Neste caso o fornecedor dos dados deve copiar seus dados em um formato específico em uma pasta desse projeto. Um requisito para os dados fornecidos é que eles devem ser georeferenciados.

Tendo copiado os dados na pasta destinada a esse propósito, o projeto está pronto para ser compilado gerando um arquivo executável, neste caso um arquivo com extensão APK, que são arquivos instaladores de aplicações em dispositivos Android.

Após gerar o APK o fornecedor pode distribuir sua aplicação para seus usuários da forma que lhe for conveniente, uma das formas populares de distribuição de aplicações Android é através do mercado eletrônico Google Play.

4. CENÁRIOS DE USO

Os cenários de uso apresentados neste trabalho foram configurados para testar a utilização de diferentes tipos de dados, assim como, diferentes tamanhos de telas com a finalidade de validar 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 feitos 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.

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 Yound Duos TV com as seguintes configurações: processador de 1GHz, 750MB de memória RAM e tela de 3,2 polegadas.

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

4.1 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*[4] 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 4 mo turísticas intestra 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[5] 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, web site da obra, etc.

A Figura 5 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.

Figura 4. Foto da utilização da aplicação em um cenário artístico com a utilização de *Tablet* no modo Navedor RA.

Figura 5. Captura de tela da utilização da aplicação em um cenário artístico com a utilização do *Smartphone* no modo navegador RA.

[4] *Stakeholders*: são as partes interessadas na aplicação, ou seja, pessoas ou organizações que tenham interesse ou sejam afetados pela aplicação.

[5] A escultura "Olho d'água", do artista plástico Acácio Sobral: http://www.portal.ufpa.br/imprensa/noticia.php?cod=3051

4.2 Cenário Educativo

O cenário educativo possui dados de informativos sobre plantas e animais, e obras do Museu Emílio Goeldi[6], situado em Belém-Pará-Brasil. Com este cenário imagina-se uma visitação 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 auxilio ao aprendizado e ao interesse pelo conteúdo ensinado. E como usuário: alunos e professores que desejam aprender e ensinar de maneira prática através da visitação aumentada de bosques, museus e zoológicos.

A Figura 6 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.

Figura 6. Foto da utilização da aplicação em cenário educacional com a utilização do *Smartphone* no modo Leitor de QR Code.

A Figura 7 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.

Figura 7. Foto da utilização da aplicação em cenário educacional com a utilização de *Tablet* para visualização do conteúdo de texto relacionado com a Arara-Azul.

4.3 Cenário Turístico

O cenário turístico possui dados relacionados com os pontos turísticos da cidade de Belém-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 desta.

As Figuras 8 e 9 mostram a utilização da aplicação em um cenário turístico com o recurso de mapas 2D em *Tablet* e *Smartphone* respectivamente.

Na Figura 8 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 9 mostra a mesma rota, porém entretando em um dispositivo com tela menor.

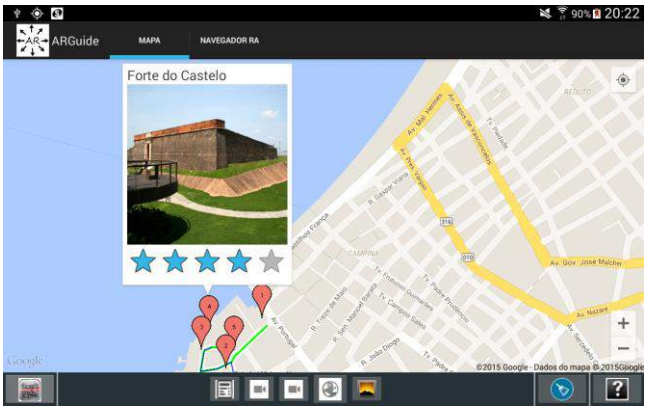

Figura 8. Captura de tela da aplicação em cenário turístico com a utilização de *Tablet* no modo mapa.

[6] Museu Paraense Emílio Goeldi: http://www.museu-goeldi.br/portal/

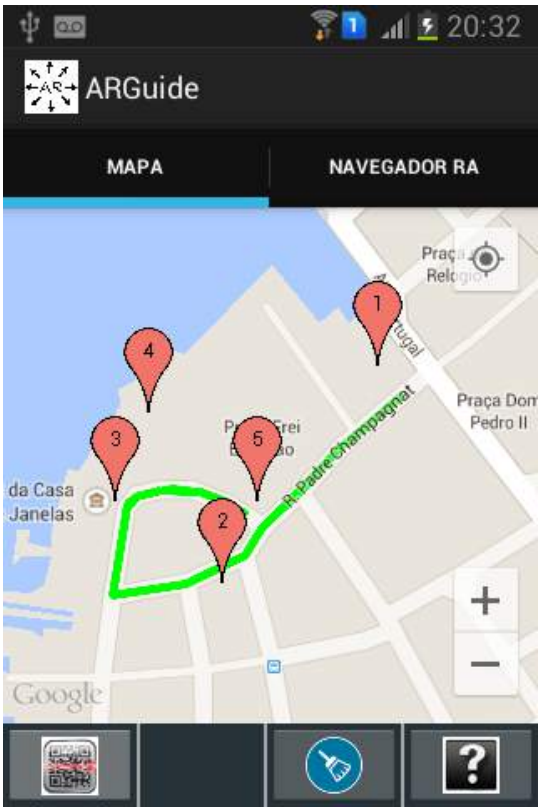

Figura 9. Captura de Tela da aplicação em cenário turístico com a utilização de *Smartphone* no modo mapa.

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, dependo 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 visitação 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.

5. CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma proposta de arquitetura para aplicações Android de Realidade Aumentada Móvel (RAM) extensível, flexível, e adaptável, sugerindo uma modificação no padrão MVC e considerando a utilização de padrões de projetos, como Proxy Remoto e Facade, assim como, a utilização do padrão de interfaces gráficas Fragments.

Dentre os pontos fortes da arquitetura proposta, pode-se destacar: a possibilidade de adicionar novas funcionalidades na aplicação devido a lógica da aplicação ter passado para as visões, basta adicionar uma nova visão que saiba se comunicar com o Router para ter uma nova funcionalidade; a possibilidade de adicionar vários tipo de dados, com o único requisito dos dados serem georefenciados, permitindo que a aplicação seja genérica servindo a muitos propósitos; e a adaptação à 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; a falta de teste de usabilidade para garantir uma boa experiência com o usuário; e a ausência de conteúdo 3D que é importante para uma experiência de realidade aumentada.

Os trabalhos futuros são aplicar melhorias nestes pontos fracos identificados com as seguintes medidas: desenvolver um gerador automático de aplicações, onde o fornecedor do conteúdo georeferenciado possa adicionar e organizar seus dados através de uma interface gráfica; realizar testes de usuabilidade como checklists e experimentos com usuários reais; e adicionar mais funcionalidades como conteúdos 3D, reconhecimento de imagem e acesso a redes sociais como Facebook e Twitter.

6. REFERÊNCIAS

[1] Cawood, S., Fiala, M. 2008. *Augmented Reality: A Pratical Guide*. Pragmatic Bookshelf, (Dallas, Texas, USA, Jan, 2008).

[2] de Sá, M., Churchill, E. 2012. Mobile Augmented Reality: *A Design Perspective. Human Factors in Augmented Reality Environments*. Springer, (New York, NY, USA), 139-164.

[3] Dünser, A. *et al*. 2007. Applying HCI principles to AR systems design. *Mixed Reality User Interfaces: Specification, Authoring, Adaptation (MRUI'07) Workshop Proceedings*. (Charlotte, NC, USA), 37-42.

[4] Fowler, M. 2002. *Patterns of Enterprise Application Architecture*. Addison-Wesley Longman Publishing Co. Inc, (Boston, MA, USA, Nov, 2002).

[5] Gamma, E. *et al*. 1994. *Design Patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc, (Boston, MA, USA, Nov, 1994).

[6] Grubert, J., Grasset, R. 2013. *Augmented Reality for Android Application Development*. Packt Publishing Ltd., (Birmingham, UK, Nov, 2013).

[7] Joorabchi, M. et al, 2013. Real Challenges in Mobile App Development. *International Symposium on Empirical Software Engineering and Measurement. ACM / IEEE*, (Baltimore, Maryland, USA, Out 10-11, 2013), 15-24.

[8] Martínez, H. *et al*. 2014. Drivers and Bottlenecks in the Adoption of Augmented Reality Applications. *Journal of Multimedia Theory and Application*. Vol. 2, No. 1, 27-44.

[9] Nudelman, G. 2013. *Android Design Patterns: Interaction Design Solutions for Developers*. John Wiley & Sons Inc. (Indianapolis, Indiana, Mar, 2013).

[10] Potel, M. 1996. *MVP: Model View Presenter The Taligent Programming Model for C++ and Java*. Taligent Inc.

[11] van Krevelen D. W. F., Poelman R. A Survey of Augmented Reality Technologies, Applications and Limitations. The International Journal of Virtual Reality. Vol. 9, No. 2. (June 2010), 1-20.

[12] Wasserman, Anthony I. 2010. Software Engineering Issues for Mobile Application Development. *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research* (Santa Fe, New Mexico, USA), 397-400.