Aspectos de Implementação de um Serviço de Voz em Aplicações de Realidade Aumentada Móvel

Implementation Aspects of a Voice Service in Mobile Augmented Reality Applications

Tiago Araújo[*], Brunelli Miranda[†], Carlos Santos[*], Nikolas Carneiro*, Marissa Carvalho*, Gleidson Sousa[†], Bianchi Serique*[†], Nelson Neto*[†]

*Programa de Pós-Graduação em Ciência da Computação - PPGCC

[†]Faculdade de Computação – FACOMP

Universidade Federal do Pará, Belém, Pará, 66075-110

***Resumo* – O uso de aplicações de realidade aumentada móvel tem aumentado nos últimos anos, permitindo a integração de vários tipos de interações que antes não eram tão difundidas. Existe atualmente uma grande quantidade de tecnologias, entretanto não há um padrão de desenvolvimento para aplicações que aceitam diversas interações. Este artigo propõe alguns aspectos de implementação com boas práticas de programação para aplicações de realidade aumentada móvel utilizando interação por voz e sistema Android. Para se chegar nesses aspectos, foi feito uma revisão técnica nas guias de desenvolvimento e API's das bibliotecas utilizadas na ferramenta. Foi condensado e apresentado boas práticas de implementação utilizando interação por voz em aplicações de realidade aumentada móvel.**

***Abstract* – The use of mobile augmented reality applications has increased in recent years, enabling the integration of several types of interactions that were not as widespread. There is currently a lot of technology, but there isn't a pattern of development for applications that accept several interactions. This paper proposes some implementation aspects with good programming practices for mobile augmented reality applications using voice interaction and Android system. To reach these aspects, a technical review was made focusing in development guides and API's used in the tool library. It was condensed and presented good practical implementation using voice interaction of mobile augmented reality applications.**

Keywords—realidade aumentada, voz, implementação, aplicação móvel, interação, Android, reconhecimento de voz

I. INTRODUÇÃO

A Realidade Aumentada Móvel (RAM) ganhou um importante destaque em múltiplas áreas de aplicação, dentre elas: engenharias, saúde, educação turismo, automação e outras. Este destaque está ligado a vários fatores, como: o desenvolvimento tecnológico dos dispositivos móveis, aumento da popularidade e comercialização de *smartphones* e *tablets*, e a inovação e facilidades da Realidade Aumentada (RA).

Apesar do desenvolvimento tecnológico e da popularidade crescente da RAM serem motivações para sua ampla utilização, são os estudos na usabilidade das ferramentas RAM que podem manter uma boa experiência de uso, consolidando assim a vida útil da utilização dessas ferramentas nas mais diversas áreas.

Martínez *et al*. [1] apresenta alguns desafios para o desenvolvimento de aplicações de RAM. Dentre os desafios apresentados, pode-se destacar a falta de padrões de desenvolvimento e o pouco espaço para a apresentação de informações.

A falta de padrões de desenvolvimento pode ser considerada tanto escassez de padrões para codificação dessas aplicações de RAM, assim como para a organização e *design* das interfaces gráficas. E o pouco espaço para apresentação das informações se dá pelo fato de que a RA precisa mostrar ao usuário o ambiente real acrescido de informações virtuais em sincronia [2], e isto, em uma tela de dispositivos móveis como um *smartphone*, que pode variar de 3 a 8 polegadas.

Inevitavelmente, o projetista terá que optar por deixar algumas informações escondidas (geralmente em menus que podem apresentar várias opções) infringindo alguns princípios de usabilidade, como a primeira heurística de Nielsen [3], em troca de uma melhor visibilidade do conteúdo principal e deixando a tela do usuário menos carregada. Porém, esconder funcionalidades em menus pode acarretar em um maior esforço cognitivo para encontrar e utilizar tais funcionalidades.

Um importante recurso que pode contribuir com a organização da interface gráfica no pouco espaço de tela é a utilização de *feedbacks* e comandos voz na qual o usuário pode interagir e receber informações utilizando a comunicação oral. Um modelo de desenvolvimento pronto para desenvolvedores pode ajudar a integração desse serviço em aplicações que antes só tinham interação por toques na tela.

Este trabalho apresenta um padrão de desenvolvimento de um serviço de voz voltado a aplicações RAM. Ele cobre a arquitetura geral desse tipo de serviço, vai refinando em um

modelo de implementação para sistemas Android e encerra com os aspectos que devem ser levados em consideração no momento de implementar esse serviço.

II. INTERAÇÃO POR VOZ

A voz é um dos meios de comunicação mais utilizados pelos seres humanos, estando entre as três formas de comunicação (voz, gestos e expressões faciais) mais utilizadas no dia-a-dia de uma pessoa [6]. A voz, quando utilizada em uma aplicação, está dentro do conceito de linguagem natural.

A interação por linguagem natural pode ser definida como a comunicação entre humano e máquina utilizando uma linguagem familiar ao humano [7]. A área de Interação Humano-Computador (IHC) atribui papel importante para esse tipo de interação, devido aos benefícios oferecidos ao usuário, entre eles, o fato das ações serem mais intuitivas, minimizando assim o esforço cognitivo e permitindo que o usuário se concentre na tarefa e não na interação em si [8]. A interação por voz também pode facilitar a busca do usuário por uma funcionalidade, ou até mesmo servir como atalho para a execução da funcionalidade em questão [9].

Apesar da utilização de comandos por voz poder variar de acordo com o tipo de aplicação, de maneira geral, o reconhecimento de voz nas aplicações segue um modelo padrão, que pode ser observado na figura 1.

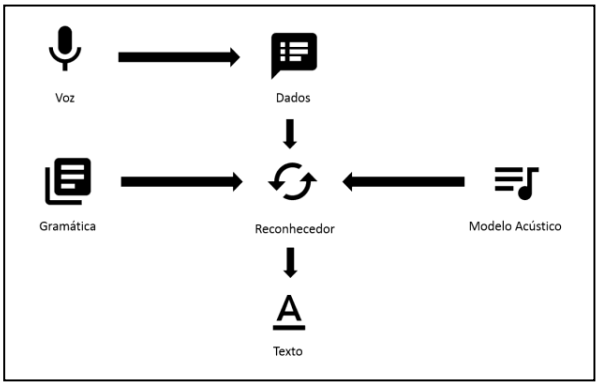

Fig 1. Funcionamento padrão de um modelo de reconhecimento de voz

No modelo apresentado, o usuário entra com a informação em forma de comando de voz, que é repassada para o reconhecedor, onde ela é analisada juntamente com a gramática e modelo acústico empregados na aplicação. O resultado desse processo é então retornado ao usuário em forma textual. Esse modelo geral exemplifica aplicações de interação por voz encontrados em vários sistemas de reconhecimento de voz [10].

III. ARQUITETURA

Um serviço de voz precisa de alguns elementos chaves para poder realizar a síntese e reconhecimento de voz. O processo visto na figura 1 mostrou o modelo geral independente da aplicação utilizada. Para aplicações de realidade aumentada móvel esse modelo se encaixa com algumas mudanças, relacionadas aos componentes de realidade aumentada.

Uma arquitetura para aplicações de RAM que utilizam serviços de voz pode ser utilizada para definir ações de implementação para os principais componentes desse tipo de aplicação. Os componentes utilizados são baseados em componentes de um dispositivo Android, que já contém alguns elementos prontos para reconhecimento e síntese de voz.

Estes componentes foram divididos em dois nós principais, relacionados aos dois principais módulos para esse modelo, serviço de voz e a Aplicação de RAM. Os componentes do serviço de voz são um reconhecedor, um sintetizador e as gramáticas. Os componentes de uma aplicação RA foram divididos em quatro, os trabalhos de [11], [12], [13], [14] e [15] apresentam componentes semelhantes quando se foca em desenvolvimento. Foi definido um padrão dentro dessas aplicações, definidos pela sua função dentro do seu escopo:

- Dados: Dados que a aplicação utiliza para gerar conteúdo para o usuário.
- Renderizador: Motor gráfico para gerar modelos de realidade aumentada na tela da aplicação.
- Reconhecimento: Conjunto de algoritmos ou ferramentas utilizados pela aplicação para gerar interação com usuário, como reconhecimento de imagens, posição georeferenciada, utilização de sensores.
- Coordenador: O componente que liga o Renderizador e o Reconhecimento para gerar o ambiente de realidade aumentada para o usuário.

A figura 2 mostra um diagrama desses componentes para a obtenção deste serviço de voz para aplicações de RAM.

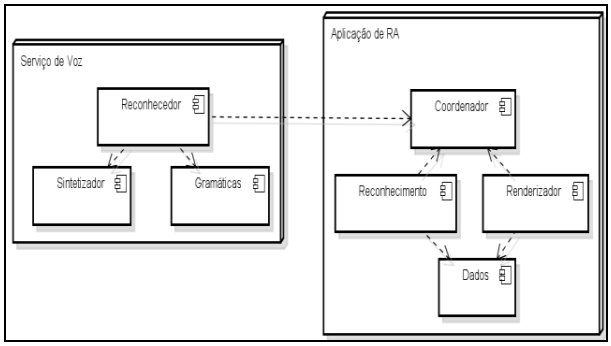

Fig 2. Diagrama de componetes de um serviço de voz em aplicações RAM

A ligação entre o reconhecedor do serviço de voz e do coordenador da aplicação de RAM é importante para manter a sincronização das interações. Um comando que tem sua ação muito atrasada ou uma ação da aplicação que seja inapropriada para um contexto podem dificultar a aceitação do serviço pelo usuário[5].

O diagrama de componentes da figura 3 mostra o conceito geral da aplicação, detalhes mais específicos podem ser vistos

na figura 2, que mostra um diagrama de classes para o serviço de voz em uma aplicação de RAM.

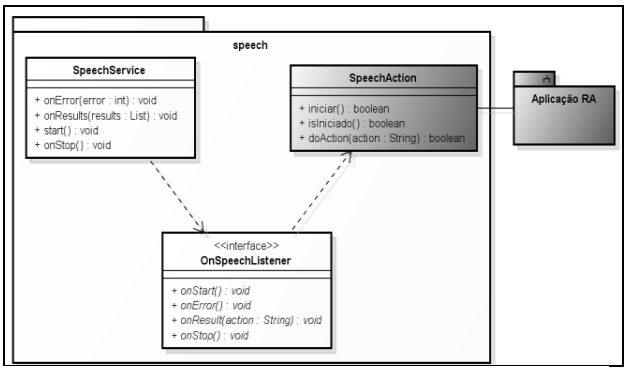

Fig 3. Diagrama de classes de um serviço de voz em aplicações RAM

A especificação de cada classe do diagrama e suas relações pode ser vista abaixo:

1) **SpeechService**: Essa classe é responsável pela comunicação realizada com o sintetizador online. Deve cercar os eventos para enviar e receber os resultados do sintetizador. Os métodos dessa classe são explicados abaixo:

- **start**: Esse método inicia o reconhecimento do serviço de voz ligando o microfone. É encerrado assim que percebe silêncio por parte da entrada do usuário.
- **onStop**: Método que é chamado quando é percebido silêncio, para que a entrada de áudio pura seja encerrada, permitindo manipulação por parte da classe. Esse é o momento de enviar os resultados para o sintetizador.
- **onResults(results : List)**: Nesse método deve-se encaminhar o conjunto de palavras para o reconhecedor poder realizar as ações necessárias fazendo comparação com a gramática.
- **onError(error : int)**: Esse método é chamado em caso de qualquer erro no processo de envio e recebimento do processo de síntese de voz. Erros que podem acontecer estão relacionados a erros de rede, áudio não ser compreendido como voz, não entendimento do que foi dito e falta de resposta do sintetizador.

2) **OnSpeechListener**: É uma interface para ligar os eventos dependetes do sintetizador que ocorrem no serviço de voz. Os métodos são chamados em *SpeechService* e suas chamadas manipuladas em *SpeechAction*.

3) **SpeechAction**: É a classe que implementa os serviços do reconhecedor. Ela recebe por meio da interface que implementa os resultados da síntese de voz e aplica as ações necessárias relacionadas a gramática. O método *doAction* recebe uma palavra da gramática utilizada pela aplicação e realiza uma ação ou cancela a mesma mediante definição já feita.

Essa arquitetura mostra um conjunto básico de classes que devem ser implementados e como devem ser relacionados com componentes de uma aplicação de RAM. Serviços já implementados no sistema Android podem ser utilizados, como o acesso ao microfone e acesso ao que foi gravado.

IV. SERVIÇO DE VOZ

A ligação entre os componentes de uma aplicação de RAM com reconhecimento de voz pode ser feita através de serviços já presentes no sistema Android, o que garante o desempenho otimizado da aplicação, já que usa chamadas nativas do próprio sistema operacional.

A figura 4, de autoria própria, mostra o fluxo do serviço de voz implementado utilizando o Google Now [16], o serviço de síntese e reconhecimento de voz padrão do Android.

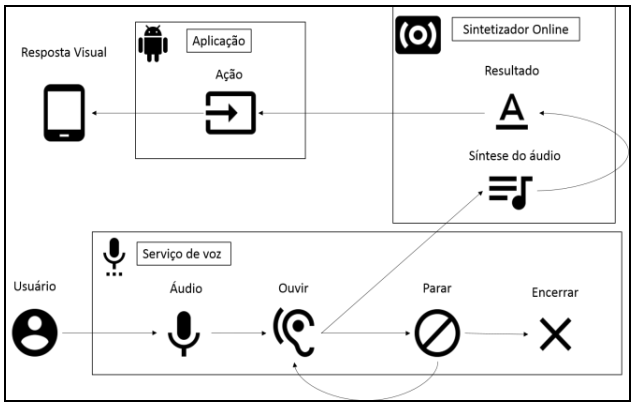

Fig 4. Fluxo de troca de mensagens na implementação do serviço

O serviço de voz implementado possui alguns estados para garantir a eficiência, acurácia e corretude do reconhecedor. Os estados e funções do reconhecedor são listados a seguir:

1) **Ouvir**: Esse estado liga o microfone do dispositivo e espera a entrada de áudio do usuário. Nesse estado o áudio puro é convertido em um formato de dados compatível com o o reconhecido pelo sintetizador. Assim que ele detecta silêncio, o reconhecedor muda de estado para o estado de Parar.

2) **Parar**: Nesse estado é necessário parar o reconhecimento da entrada do usuário, e enviar os dados para o sintetizador. Abrir uma conexão e esperar o sinal positivo de entrega são as etapas para esse envio. De acordo com a ação realizada, pode-se passar pro estado de encerrar, ou voltar para o estado de Ouvir.

3) **Encerrar**: Esse estado finaliza o serviço de reconhecimento. Faz as verificações necessárias para liberar o microfone para o sistema, e encerra qualquer conexão ainda aberta com a internet usada em função do reconhecedor.

V. ASPECTOS DE IMPLEMENTAÇÃO

Os aspectos podem ser divididos em duas partes principais: relacionadas a aplicação e relacionadas a tecnologia. Aspectos de aplicação são as medidas que devem ser tomadas para que o serviço de voz seja integrado a aplicação de RA. A figura 5 mostra o processo de aplicação

das diretrizes na aplicação de RA. As diretrizes que devem ser seguidas para implementação do serviço estão listadas abaixo:

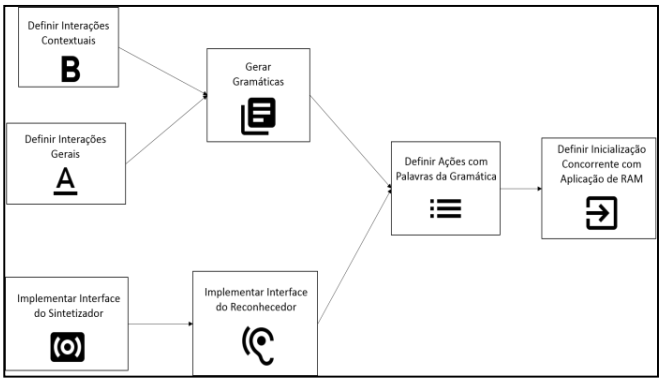

Fig 5. Aspectos gerais de implementação

1) **Definir Interações Contextuais:** Defnir comandos contextuais é necessário para que os comandos que são dependentes de certo contexto sejam executados somente nesse contexto. Exemplos de contexto podem ser a visualização de certo dado, a navegação em um mapa, a ação diante de um objeto virtual. As interações em que se deseja ter interação devem ser mapeadas em um conjunto de palavras e deve ser associada a uma condição de contexto presente na aplicação

2) **Definir Interações Gerais:** Comandos gerais são necessáriaos parar ações que são gerais em mais de um contexto da aplicação. São comandos gerais ações de menu que podem ser acessadas em qualquer momentos, comandos padrões da própria tecnologia utilizada, como voltar, página principal (home) e configurações. Essas interações devem ser mapeadas para que o contexto não seja aplicado a elas, o conjunto de palavras não pode ser concorrente com nenhuma palavra de nenhum contexto.

3) **Gerar Gramáticas:** Definidas as interações que serão possíveis em contextos e de forma geral, o conjunto dessas palavras deve ser usado para geração de uma gramática que será usada para as ações da aplicação. Essa gramática deve ser organizada de modo que o reconhecedor possa utilizar essas palavras para implementar essas ações, que depende do formato que o sintetizador e o reconhecedor podem processar.

As diretrizes que devem ser seguidas para tecnologia estão listadas abaixo:

1) **Implementar Interface do Sintetizador:** O sintetizador pode estar integrado ou não ao telefone, mas é necessário ter uma interface de acesso a ele, ao microfone, ou ambos. Essa interface deve ser criada para processamento ou direcionamento da voz do usuário. Ela deve ter métodos para retorno do resultado do que foi captado para a interface do reconhecedor.

2) **Implementar Interface do Reconhecedor:** Essa interface deve ser implementada para direcionar o resultado do sintetizador em alguma ação da aplicação. Deve ter métodos para esperar os resultado do sintetizador e métodos para cercar os estados possíveis do reconhecimento, como ouvindo, começando a ouvir, parando de ouvir e inciando síntese.

3) **Definir Ações com Palavras da Gramática:** As ações devem ser associadas no reconhecedor, com as palavras que foram definidas na gramática. Essas ações devem podem ser as gerais ou de contexto, desde que sejam as mesmas acessíveis através da interface gráfica. Mais de uma palavra pode acionar a mesma ação, desde que esteja mapeada na gramática.

4) **Definir Inicialização Concorrente com Aplicação de RAM:** O passo final para implementação desse serviço é definir como o serviço vai ser iniciado. Um botão que aciona o serviço é usado em algumas aplicações de interação por voz [17], mas pode ser definido do modo que for necessário para a aplicação. Essa implementação é altamente dependente da tecnologia, já que envolve paralelismo de execução e interação, e sistemas operacionais móveis implementam de forma diferenciada acesso a ferramentas como microfone e execução do modo de ouvir dos mesmo.

Esses aspectos são relevantes no desenvolvimento de qualquer serviço de voz em uma aplicação RAM. O exemplo de reconhecedor de voz utilizado é padrão de sistemas Android, sendo assim esse modelo pode ser replicado com fidelidade nesses sistemas.

CONSIDERAÇÕES FINAIS

Esse trabalho propõe uma série de aspectos de implementação que devem ser levados em consideração para que serviços de voz possam ser integrados com sucesso em aplicações de RAM.

Esses aspectos foram usados como base para o desenvolvimento do serviço de voz de uma aplicação de RAM, que pode ser vista na figura 6. Um breve conjunto de palavras usadas para as gramáticas está presente também, dividas em seu determinado contexto.

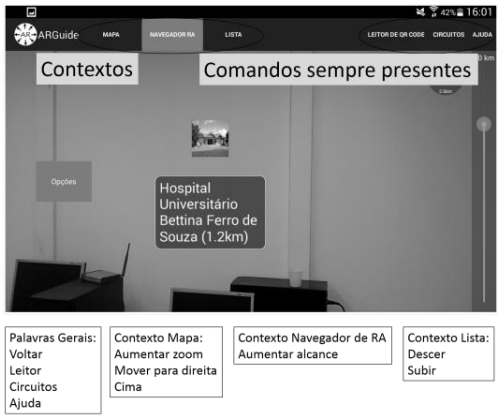

Fig 6. Exemplo de aplicação e palavras definidas nas gramáticas

Foi definido com uma base em sistema Android, então certas decisões feitas em relação a tecnologia foram feitas escolhendo o que era mais eficaz dentro desse contexto. Futuros trabalhos devem cobrir exceções de outros sistemas.

Os aspectos apresentados procuram facilitar o desenvolvimento de aplicações RAM que tem como objetivo integrar um serviço de voz. Uma arquitetura e diagramas foram apresentados para suportar esse modelo.

A aderência a esses aspectos deve facilitar que aplicações de RAM possam melhorar suas interações e que os usuários se sintam mais a vontade para escolher variadas interações em seus ambientes de RAM.

REFERÊNCIAS

[1] H. Martínez, D. Skournetou, J. Hyppola, S. Laukkanen, and A. Heikkila,"Drivers and bottlenecks in the adoption of augmented reality applications," Journal of Multimedia Theory and Application, 2014.

[2] Pulli, P.; Pyssysalo, T.; Metsavainio, J.-P.; Komulainen, O., "CyPhone-experimenting mobile real-time telepresence," in Real-Time Systems, 1998. Proceedings. 10th Euromicro Workshop on , vol., no., pp.10-17, 17-19 Junho 1998

[3] NIELSEN, J. "Heuristic Evaluation". Em: NIELSEN, J.; MACK, R. L. Usability Inspection Methods. New York, NY: Katherine Schowalter, 1994. Cap. 2

[4] Cawood, S., Fiala, M., "About Augmented Reality", Augmented Reality: A Pratical Guide. Pragmatic Bookshelf, USA 2008, cap 1.

[5] T. Olsson and M. Salo. "Online User Survey on Current Mobile Augmented Reality Applications". Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality. Washington: IEE Computer Society. 2011. pp 75-84.

[6] A. Teixeira et al. "Speech-Centric Multimodal Interaction for Easy-To-Access Online Services – A Personal Life Assistant for the Elderly". 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion – DSAI, 2013.

[7] B. Shneiderman. "Designing the User Interface: Strategies for Effective Human-Computer Interaction", 3rd Edition, pp 293-295.1998.

[8] N. Vidakis, M. Syntychakis, G. Triantafyllidis and D. Akoumianakis. "Multimodal Natural User Interaction for Multiple Applications: The Gesture – Voice Example". International Conference on Telecommunications and Multimedia – TEMU, 2012.

[9] L. Xia, K. Kai, W. Xiaochun and W. Dan. Research and Design of the "Voice-Touch-Vision" Multimodal Integrated Voice Interaction in the Mobile Phone, 2010.

[10] Dolezal, J.; Kencl, L., "A unifying architecture for easy development, deployment and management of voice-driven mobile applications," in Network and Service Management (CNSM), 2011 7th International Conference on , vol., no., pp.1-5, 24-28 Oct. 2011

[11] Tang, L.Z.W.; Kian Sin Ang; Amirul, M.; Bin Mohamed Yusoff, M.; Chee Keong Tng; Bin Mohamed Alyas, M.D.; Joo Ghee Lim; Kyaw, P.K.; Folianto, F., "Augmented reality control home (ARCH) for disabled and elderlies". Em Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on , vol., no., pp.1-2, 7-9 April 2015

[12] Vullamparthi, A.J.; Nelaturu, S.C.B.; Mallaya, D.D.; Chandrasekhar, S., "Assistive Learning for Children with Autism Using Augmented Reality". Em Technology for Education (T4E), 2013 IEEE Fifth International Conference on , vol., no., pp.43-46, 18-20 Dec. 2013

[13] Buti Al Delail, Luis Weruaga, and M. Jamal Zemerly. 2012. "CAViAR: Context Aware Visual Indoor Augmented Reality for a University Campus". Em Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03 (WI-IAT '12), Vol. 3. IEEE Computer Society, Washington, DC, USA, 286-290.

[14] Loris D'Antoni, Alan Dunn, Suman Jana, Tadayoshi Kohno, Benjamin Livshits, David Molnar, Alexander Moshchuk, Eyal Ofek, Franziska Roesner, Scott Saponas, Margus Veanes, and Helen J. Wang. 2013. Operating system support for augmented reality applications. In Proceedings of the 14th USENIX conference on Hot Topics in Operating Systems (HotOS'13).

[15] C. Santos, N. Carneiro, B. Miranda, B. Serique. "Uma Aplicação de Realidade Aumentada Móvel para Ambientes Indoor e Outdoor". XV Workshop de Realidade Virtual e Aumentada, Marília – SP, pp 120-126, November 2014.

[16] Google Inc., "Google Now", https://www.google.com/landing/now/. Dezembro 2014.

[17] Buti Al Delail, Luis Weruaga, and M. Jamal Zemerly. 2012. "CAViAR: Context Aware Visual Indoor Augmented Reality for a University Campus". Em Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03 (WI-IAT '12), Vol. 3. IEEE Computer Society, Washington, DC, USA, 286-290.