

extMobilisTTS: Uma Arquitetura de Aplicação Móvel para Suporte a Fóruns usando Text-to-Speech em Ambientes Virtuais de Aprendizagem

Wellington W. F. Sarmiento¹, Gabriel A. L. Paillard¹, Wedson de S. Lima¹,
Katyne F. Rabelo¹, Andrei B. B. Torres¹, Humberto O. O. Gomes¹,
Paulo M. B. Costa¹, Mauro C. Pequeno¹

¹Instituto Universidade Virtual – Universidade Federal do Ceará (UFC)
CEP 60440-554 – Fortaleza – CE – Brazil

{extmobilistts} @virtual.ufc.br

Abstract. *This paper presents an architecture for communication between mobile devices and a Virtual Learning Environment (VLE) called extMobilisTTS, and a client application (Solar Mobilis) which uses the hardware features of the device and the usage of external services, such as a text-to-Speech (TTS) module, favoring the interaction between teachers and students in discussion forums of the VLE.*

Resumo. *Este trabalho apresenta uma arquitetura de comunicação entre dispositivos móveis e um Ambiente Virtual de Aprendizagem (AVA), chamada ext-MobilisTTS, bem como uma aplicação cliente (Solar Mobilis) que se utiliza de recursos do hardware do dispositivo e de serviços externos à aplicação, como um módulo Text-to-Speech (TTS) para conversão de texto para voz, favorecendo a interação entre professores e alunos em fóruns de discussões de um AVA.*

1. Introdução

As atuais Redes Telemáticas modificaram o ritmo de vida das pessoas e levaram à necessidade cada vez maior de acesso à informação mesmo estando fora de suas instituições de trabalho ou estudo, ou ainda, em trânsito. Hoje, existem diversos tipos de dispositivos móveis que possibilitam a comunicação e a execução de aplicações computacionais. Assim como aconteceu com outras tecnologias, como é o caso dos computadores, os dispositivos móveis também estão influenciando a maneira como se faz educação, desta forma disponibilizando novas ferramentas que possibilitam o processo de aprendizagem, tanto no ensino presencial quanto à distância.

A utilização de dispositivos móveis para o processo de aprendizagem é chamada *Mobile Learning* (m-Learning) [Trifonova 2003][Lee et al. 2005]. É neste cenário de *m-Learning* que se insere a proposta de integração de um AVA, em nosso estudo de caso, o Solar versão 2.0, e de aplicações móveis de suporte à Educação a Distância. A arquitetura proposta neste trabalho é uma extensão do Mobilis, apresentado em [Sarmiento 2007]. A arquitetura Mobilis descreve como um conjunto de aplicações móveis pode acessar um AVA, estendendo suas funcionalidades tradicionais e acrescentando funções adaptadas à realidade dos dispositivos móveis, mantendo o foco do uso de acordo com as potencialidades destes dispositivos. O que se propõe é um uso específico dos recursos essenciais de

um dispositivo móvel, tais como microfone, conexão com internet, reprodução de vídeo, áudio e *touchscreen* para acessar informações e interagir com um AVA.

A extensão do Mobilis explora o uso de tecnologias atuais como as plataformas móveis Android e iOS. Desta forma, alunos e professores terão formas alternativas de interagir com fóruns de AVA, utilizando o recurso de áudio, por exemplo. Como primeira aplicação a compor esta arquitetura, foi criado o Solar Mobilis, para acesso a fóruns através de áudio. Uma primeira versão desta aplicação foi apresentada no artigo [Paillard et al. 2012] e sua evolução continuou até o presente trabalho.

A presente pesquisa apresenta a arquitetura estendida do Mobilis - Extended Mobilis TTS ou extMobilisTTS -, que utiliza como base os recursos de reprodução de textos via *Text-to-Speech* (TTS - texto para fala) e de gravação de voz, bem como um estudo de caso feito através da aplicação Solar Mobilis de comunicação com o AVA Solar 2.0. A estrutura deste trabalho é composta pela seção 2, que apresentará os trabalhos relacionados ao tema abordado; seção 3, que expõe de forma mais detalhada o ambiente extMobilisTTS; na seção 4 será apresentado o estudo de caso com o SOLAR versão 2.0 e, por fim, as conclusões do estágio atual desta pesquisa.

2. Trabalhos Relacionados

Com a chegada dos *smartphones*, surgiu a necessidade de adequar o conteúdo dos AVA às novas formas de comunicação móvel. Antes, tal comunicação era feita unicamente pelo navegador do celular. Hoje, é possível que seja realizada por meio de aplicações específicas que possibilitam o aproveitamento total dos recursos do dispositivo.

Existem trabalhos que começam a explorar as funcionalidades dos dispositivos móveis. Em [Al Tunaiji and Zemerly 2010] é apresentada a proposta de uma aplicação chamada *M-learning and University Student Organizer* (MUSO) que permite ao estudante verificar atualizações referentes a sua vida acadêmica como notas, faltas, data de provas e notícias da universidade. Já em [Blackboard 2011] é descrita uma aplicação voltada para *smartphones* contendo notícias, blogs, notas, visualização de vídeo e áudio, *roster* e discussão. O sistema [Mobl21 2011] possui mais funcionalidades voltadas para *m-Learning* e contém ferramentas de avaliação como *Quizzes*, *Flashcards* e guias de estudo tanto *online* quanto *offline*, abrangendo, além dos *smartphones*, os *tablets*. Tais aplicações normalmente fazem uso de meios textuais para a comunicação entre os AVA e as aplicações móveis. Neste caso, recursos multimídia dos *smartphones* são relegados a segundo plano. É neste contexto que a extMobilisTTS se insere.

3. Descrição da Arquitetura

3.1. Visão Geral do Sistema

Esta seção mostra a arquitetura geral do extMobilisTTS, bem como a aplicação criada a partir desta, o Solar Mobilis. O código fonte da aplicação cliente e servidor, bem como o executável da aplicação cliente e a documentação, encontram-se no servidor <http://www.github.com/wwagner33/extmobilis>.

3.1.1. O que o Sistema faz

O Mobilis, especificado por [Sarmiento 2007], define uma arquitetura de integração entre AVA e dispositivos móveis através do uso de *web services*, utilizando para a construção destas tecnologias como *Simple Object Access Protocol* (SOAP), .NET e/ou JAVA.

Para uma maior facilidade de integração entre as aplicações utilizadas e uma maior qualidade na troca de dados entre o dispositivo móvel e o *web service*, propomos utilizar uma nova tecnologia para a criação deste último. Assim, está-se propondo uma arquitetura que estende o Mobilis, chamada extMobilisTTS.

O extMobilisTTS é uma arquitetura Cliente-Servidor na qual a aplicação servidora é um AVA que fornece acesso a seu conteúdo por meio de uma API (*Application Programming Interface*) de acesso.

A aplicação cliente se utiliza dos recursos nativos das plataformas em que foram criadas para aprimorar a forma na qual um usuário pode interagir com o AVA (Fóruns, Avisos, Mensagens etc.). Está previsto, também, nessa arquitetura, o uso de API externas que fornecem serviços de terceiros, que não se comunicam diretamente com o AVA e são acessadas pelo lado do cliente, como o acesso a um módulo TTS, por exemplo.

3.2. Descrição do Cliente

Foi desenvolvido um sistema cliente chamado Solar Mobilis para acessar os fóruns do AVA Solar 2.0, a fim de validar o funcionamento da arquitetura. Para tanto, foram realizados testes para verificar a viabilidade de uso em diversos tipos de dispositivos e conexões. Estes testes são detalhados na seção 4.2.

As principais características da aplicação são: recuperar as novas mensagens dos fóruns para leitura; responder com texto, com a possibilidade de anexar arquivos de áudio gravado pela própria aplicação; e ler as mensagens recebidas para o usuário a partir de um módulo TTS (*text-to-speech*).

Os dados (postagens de usuários e informações de cursos) são recebidos no formato *JavaScript Object Notation* (JSON) e decodificados para serem armazenados no dispositivo móvel usando SQLite.

Para o envio de dados, estes são convertidos para objetos JSON e enviados para o servidor Solar 2.0 através de chamadas a API.

O módulo TTS faz uso da API *Speak* do Microsoft Bing que recebe como parâmetro um *array* de caracteres e retorna um arquivo de áudio que contém o texto falado. Esse arquivo é salvo temporariamente no dispositivo durante a sua reprodução. As chamadas são feitas na forma de blocos onde uma postagem é dividida em várias chamadas para que o usuário não espere muito tempo até que o áudio inicie.

3.3. Descrição da Comunicação com o Servidor

O AVA Solar 2.0, construído com o *framework Ruby On Rails* e utilizando as rotas baseadas no REST (definido em [Fielding 2000]), disponibiliza acesso a seus fóruns de discussão através de uma API (documentação disponível em <https://github.com/wwagner33/solar/wiki/API-Forum>). Isso facilita a comunicação com

aplicações externas e diminui a quantidade de código escrito, tornando-o mais legível e facilitando sua manutenção.

Após uma etapa obrigatória de *login*, utilizando os mesmos dados de usuário e senha do AVA, o aplicativo recebe um *token – hash* aleatório para ser utilizado nas próximas requisições. Esse *hash* identifica o usuário no sistema e é renovado a cada nova etapa de *login*. Requisições ao AVA, pela aplicação móvel, não são aceitas sem este *hash*.

Na troca de informações entre a aplicação móvel e o Solar 2.0, são utilizados objetos JSON no formato `{chave:valor}`.

Essa forma de comunicação permite uma otimização na quantidade de caracteres trocados entre a aplicação móvel e o servidor, pois assim não temos um *overhead* de cabeçalhos ou caracteres desnecessários, como as *tags* em uma aplicação que utiliza *eXtensible Markup Language* (XML) e o tempo de desserialização é menor [Hameseder et al. 2011].

Utilizando o *Ruby On Rails* com o que é disponibilizado por esta plataforma de arquitetura REST e considerando os métodos HTTP, a aplicação servidora é capaz de mapear a requisição e redirecionar para a classe e método adequados. Desta forma, podemos manter a aplicação com código reduzido e mais simples, tornando fácil a manutenção e criação de novas funcionalidades, o que agiliza o processo de desenvolvimento.

3.4. Interface e Usabilidade

O modelo de *design* do aplicativo móvel do sistema Solar Mobilis foi focado nas *guidelines* do Android 4.0, plataforma da Google [Google 2011]. As *guidelines* são guias atualizadas periodicamente com regras de bom uso de desenvolvimento e *design* de interfaces para a plataforma, de modo a padronizar alguns itens e tornar os aplicativos fáceis de se utilizar e ao mesmo tempo esteticamente agradáveis.

A interface do Solar Mobilis tem como objetivo principal, do ponto de vista pedagógico, tornar o uso da aplicação fácil e objetiva para os alunos do AVA Solar 2.0. É importante que eles consigam acessar uma versão móvel e simplificada do fórum que não provoque muitas distrações, visto que o foco é o conteúdo e a aprendizagem.

Partindo destas necessidades, dividimos a interface em 3 áreas distintas: barra de ação principal: exibe o título da tela atual e botões de ações relevantes (ex.: Responder a Todos); área de título: uma região fixa que identifica o título do fórum e o período de postagem; conteúdo principal (postagens): exibe os posts dos usuários, identificados por avatares personalizáveis, *nickname* e data de envio. O conteúdo da postagem é limitado a exibir as 5 primeiras linhas.

O objetivo da tela é facilitar a visualização e participação do usuário no fórum. Na barra de ação principal, o único botão disponível diretamente é o de responder ao fórum, que será a função mais utilizada pelos usuários e que é o foco da atividade.

Ao tocar sobre uma postagem, uma barra de ação contextual é exibida no lugar da barra de ação principal (ver Figura 1), e nela são exibidas funções específicas à postagem selecionada: expandir (caso ela tenha sido truncada, por passar do limite de 5 linhas), reproduzir via TTS, marcar como favorita, responder e ver mais informação.

A ação de gravação de uma postagem foi desenvolvida de modo a ficar claro para



Figura 1. Tela padrão de postagens (esquerda) e com barra contextual (direita)

o usuário que ele pode enviar uma mensagem textual e/ou anexar a gravação desta mensagem. Por isso, o ícone do áudio está claramente inserido na mensagem, podendo o usuário, ao clicar nesse ícone, editar ou excluir a gravação.

4. Estudo de Caso

4.1. Solar Mobilis

O aplicativo cliente foi desenvolvido utilizando um conjunto de ferramentas e bibliotecas que serão descritas a seguir.

Como ambiente de desenvolvimento foi utilizado o Eclipse IDE versão 3.7. Para que o IDE esteja apto para desenvolvimento de aplicativos para Android, é necessário instalar o *Android Development Tools* (ADT), *plugin* para Eclipse IDE fornecido pelo Google, que habilita o Eclipse para gerenciar projetos que utilizam o Android SDK e gere os arquivos binários para o sistema Android [Google 2012a].

O *Android Software Development Kit* (SDK) fornece as bibliotecas de API e as ferramentas de desenvolvimento necessárias para construir, testar e depurar aplicativos para Android [Google 2012b].

A *Microsoft Translator API* fornece uma série de serviços que podem ser acessados por diversos meios [Microsoft 2012]. No caso do Solar Mobilis, utilizamos o serviço de TTS para realizar a leitura de cada uma das postagens de um fórum para o usuário, de forma que o mesmo possa executar outras tarefas enquanto toma conhecimento do que foi discutido.

Foram utilizadas bibliotecas adicionais para facilitar certos aspectos da implementação: *JSON.simple* foi utilizada na versão 1.1.1 [Fang 2012]; *Lightweight Object Relational Mapping* (ORM) *Java Package* (OrmLite) na versão 4.4.1 [Watson 2012]; Apache Commons IO 2.1 [Apache 2012b]; HttpMime 4.1.3 [Apache 2012a]; ActionBarSherlock 4.1.0 [Wharton 2012].

4.2. Testes e Análise de Resultados

Testes preliminares foram realizados com o objetivo de verificar o desempenho do aplicativo em situações próximas as de uso diário. Tais testes foram realizados pela própria equipe de desenvolvimento e alguns colaboradores externos, a fim de que se pudesse ter

Duração total da leitura	903s
Tempo de espera total	46,047s
Tempo de Espera Médio por postagens	1,535s

Tabela 1. Teste de leitura de 30 postagens usando TTS e rede Wi-Fi (10 mbps)

Duração total da leitura	965s
Tempo de espera total	99,844s
Tempo de Espera Médio por postagens	3,328s

Tabela 2. Teste alternando uso de TTS em rede HSPA, GPRS e EDGE

uma noção sobre a conectividade do sistema, a estabilidade da informação frente a um ambiente real de uso de *smartphones* e o comportamento da interface com o usuário.

4.2.1. Teste de Conexão

Para aferir a conexão entre a aplicação Solar Mobilis e o servidor de AVA, foram realizados testes de comunicação usando diferentes tecnologias de celular e redes móveis locais.

Os testes foram realizados em um fórum contendo 30 postagens, com uma média de 300 caracteres. Esta seria a carga a ser recebida pelo usuário do AVA através do Solar Mobilis. Características dos testes: os testes foram realizados em redes Wi-Fi, HSPA, GPRS e EDGE e foram utilizados três parâmetros. A Duração Total de Leitura é o tempo que o aplicativo Solar Mobilis leva para realizar sua comunicação com o servidor de TTS, transformar todas as mensagens (postagem) em áudio e vocalizá-las para o usuário. O Tempo de Espera Total é o tempo que o sistema levou para iniciar todas as 30 postagens para o usuário (*delay* total de leitura). Por fim, a Tempo de Espera Médio por postagens é o *delay* que uma postagem leva para que seja iniciada sua vocalização.

O que se pode notar nas tabelas 1 e 2 é que o *delay* de início da leitura da mensagem é muito menor (54%) em redes Wi-Fi (10 mbps) do que com as tecnologias de comunicação por pacotes em celulares. Este resultado era esperado devido a diferença da taxa de transferência das tecnologias usadas. No entanto, ao receber o arquivo de áudio e começar a reproduzi-lo para o usuário, o tempo total de leitura é próximo (6,4%), pois depende somente do codificador de áudio do próprio celular e variações do processamento.

4.2.2. Teste em Ambiente Móvel

Teste de acesso ao fórum para leitura e ditado (gravação) feito em um automóvel com mudança de células de telefonia (percorrendo uma distância de aproximadamente 5 quilômetros). Foram realizadas 10 postagens em um fórum e leitura de outras 10. Este teste levou em consideração a satisfação do usuário e sua percepção do texto ditado.

Tivemos como resultado:

- 3G: A leitura foi feita de maneira fluida, sem apresentar qualquer tipo de interrupção, mesmo transitando entre as células.

- 2,5G: Verificou-se um pequeno atraso entre a leitura do cabeçalho da postagem, que informa o nome do aluno e o horário de envio, e o conteúdo. O mesmo atraso foi percebido entre o final da leitura de uma postagem e a seguinte.

Novamente, pode-se perceber que a modificação na tecnologia de transmissão de pacotes usadas no 3G (HSPA) e no 2,5G (EDGE) influencia a usabilidade da aplicação.

4.2.3. Adequação da Interface em Diferentes Resoluções

A plataforma Android foi criada para ser utilizada em uma ampla gama de dispositivos e formatos. Portanto, os aplicativos devem ser flexíveis e escalonáveis de maneira a se ajustar em tais dispositivos e manter um *layout* de fácil uso e agradável ao interagente.

O objetivo deste teste foi verificar como o aplicativo Solar Mobilis se comporta em 3 resoluções de tela com diferentes densidades, representadas em termo de DPI (*dots per inch* - pontos por polegada), especificadas pelo Google como LDPI, MDPI e HDPI, nos seguintes dispositivos: Sony Xperia X10 Mini: 240x320 *pixels*; LG Optimus One P500: 320x480 *pixels*; Motorola Droid 2: 480x854 *pixels*.

Após utilizar o aplicativo nos três dispositivos, verificou-se que ao seguir as recomendações de espaçamento de elementos, *grid* (disposição dos elementos em áreas proporcionais da tela) e tamanhos de fonte do *guideline* [Google 2011] provido pelo Google, todos os elementos mantiveram uma boa legibilidade e que os elementos da interface foram de fácil interação, independentemente da resolução da tela.

5. Conclusões

O objetivo principal da arquitetura extMobilisTTS é a utilização dos recursos dos celulares e *smartphones* em todo seu potencial para auxiliar o acesso a ferramentas de comunicação de AVA. Esta arquitetura foi desenvolvida na forma da aplicação Solar Mobilis que, em um primeiro momento, contará somente com acesso a fóruns, porém, pretende-se incluir outros mecanismos de comunicação que sejam favorecidos pelo uso dos dispositivos móveis (vídeo, gravação de imagens etc.). Assim, com os testes realizados pode-se verificar que a aplicação Solar Mobilis se comporta de forma satisfatória em seus primeiros testes relacionados a conectividade, satisfação do usuário e adequação da interface gráfica. Um ponto importante é que se utilizou a técnica de TTS baseada em servidor remoto, o que diminuiu o gasto de processamento dos equipamentos móveis utilizados, embora tenha refletido no aumento do uso da banda de comunicação.

Como trabalhos futuros, pretende-se portar a aplicação para Apple iOS e Microsoft Windows Phone, abrangendo assim os principais sistemas móveis em uso. Também almeja-se expandir a aplicação permitindo o acesso às outras funcionalidades de comunicação do AVA Solar 2.0.

6. Demonstração

A aplicação será apresentada no evento da SBRC 2013 através do uso do Solar Mobilis em tempo real e demonstradas suas funcionalidades quanto ao envio de mensagens gravadas e vocalização de textos postados em fórum do AVA. A tecnologia de comunicação usada será o HSPA ou Rede Wi-Fi disponível no evento. Para melhor apreciação da platéia, o processo será transmitido através de câmera de vídeo para um projetor multimídia.

7. Agradecimentos

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico.

Referências

- Al Tunaiji, A. and Zemerly, M. (2010). Muso: An m-learning amp; university student organizer platform. In *Ubi-media Computing (U-Media), 2010 3rd IEEE International Conference on*, pages 155 –160.
- Apache (2012a). Apache commons HttpMime. <http://hc.apache.org/httpcomponents-client-ga/httpmime/>.
- Apache (2012b). Apache commons IO. <http://commons.apache.org/io/>.
- Blackboard (2011). Blackboard móbile - aplicação para android. <http://www.blackboard.com/platforms/mobile/overview.aspx>.
- Fang, Y. (2012). JSON simple - a simple java toolkit for JSON. <http://code.google.com/p/json-simple/>.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California.
- Google (2011). Android design guidelines. <http://developer.android.com/design/index.html>.
- Google (2012a). ADT plugin - Android Development Tools. <http://developer.android.com/tools/sdk/eclipse-adt.html>.
- Google (2012b). Android SDK. <http://developer.android.com/sdk/index.html>.
- Hameseder, K., Fowler, S., and Peterson, A. (2011). Performance analysis of ubiquitous web systems for smartphones. In *Performance Evaluation of Computer Telecommunication Systems (SPECTS), 2011 International Symposium on*, pages 84 –89.
- Lee, V., Schneider, H., and Schell, R. (2005). *Aplicações Móveis – Arquitetura, Projeto e Desenvolvimento*. Editora Pearson Education do Brasil e Makron Books.
- Microsoft (2012). Microsoft Translator API. <http://api.microsofttranslator.com>.
- Mobl21 (2011). Mobl21. <http://www.mobl21.com/>.
- Paillard, G. A. L., Costa, P. M. B., Rabelo, K. F., Sarmiento, W. W. F., and Harriman, C. L. S. (2012). Extended mobilis: a integration of learning management system with mobile application to m-learning environment. *6th Euro American Conference on Telematics and Information Systems, 2012, Valência. EATIS '12 Proceedings of the 6th Euro American Conference on Telematics and Information Systems*.
- Sarmiento, W. W. F. (2007). Integração de um ambiente virtual de aprendizagem com aplicações móveis de suporte a educação à distância. Master's thesis, Programa de Pós-graduação do Departamento de Teleinformática da Universidade Federal do Ceará.
- Trifonova, A. (2003). Mobile learning - review of the literature. *Technical Report DIT-03-009*. <http://eprints.biblio.unitn.it/archive/00000359/01/009.pdf>.
- Watson, G. (2012). ORM Lite - lightweight object relational mapping (ORM) java package. <http://ormlite.com>.
- Wharton, J. (2012). ActionBarSherlock. <http://actionbarsherlock.com/>.