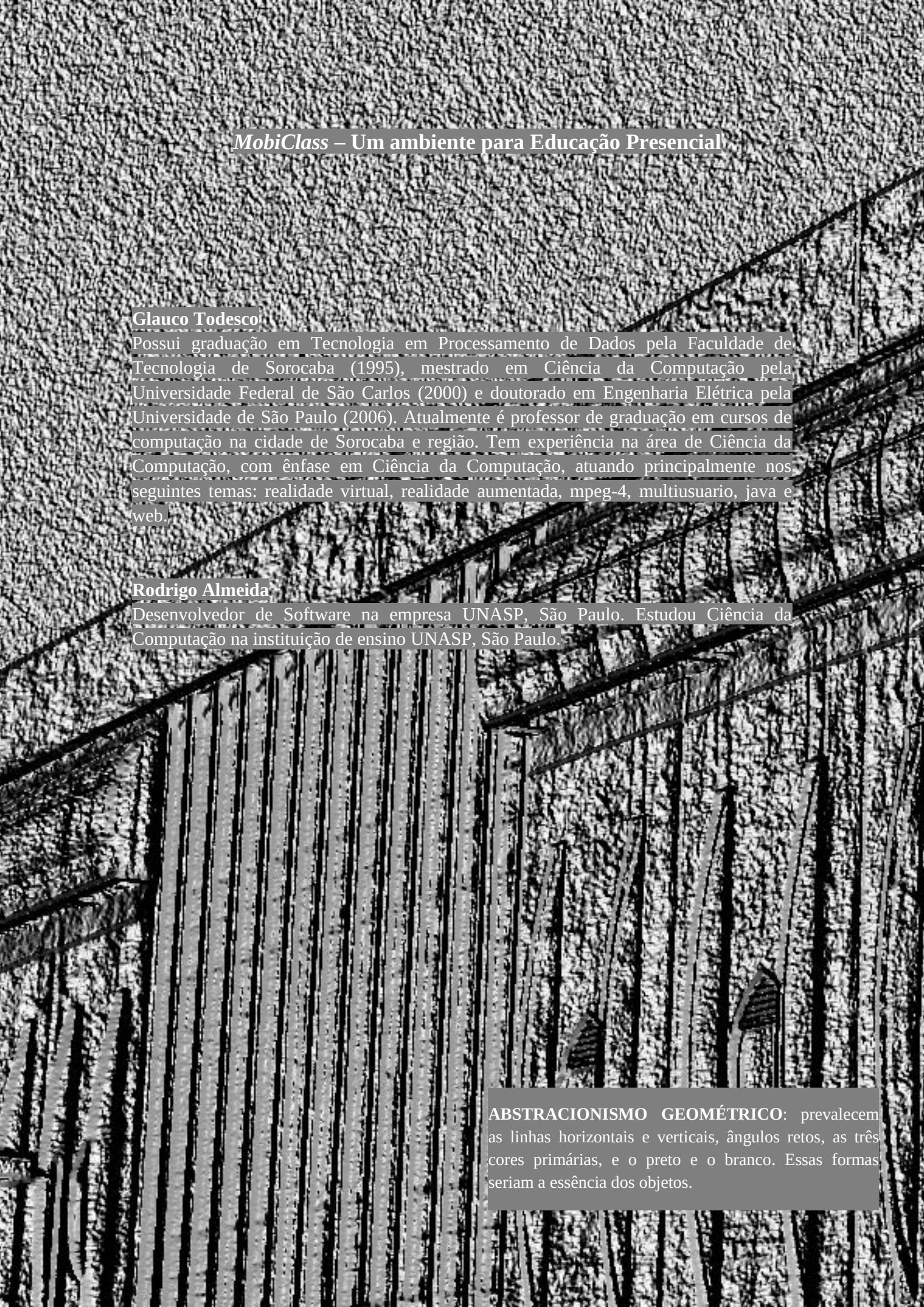

MobiClass – Um ambiente para Educação Presencial

Glauco Todesco

Possui graduação em Tecnologia em Processamento de Dados pela Faculdade de Tecnologia de Sorocaba (1995), mestrado em Ciência da Computação pela Universidade Federal de São Carlos (2000) e doutorado em Engenharia Elétrica pela Universidade de São Paulo (2006). Atualmente é professor de graduação em cursos de computação na cidade de Sorocaba e região. Tem experiência na área de Ciência da Computação, com ênfase em Ciência da Computação, atuando principalmente nos seguintes temas: realidade virtual, realidade aumentada, mpeg-4, multiusuario, java e web.

Rodrigo Almeida

Desenvolvedor de Software na empresa UNASP, São Paulo. Estudou Ciência da Computação na instituição de ensino UNASP, São Paulo.

ABSTRACIONISMO GEOMÉTRICO: prevalecem as linhas horizontais e verticais, ângulos retos, as três cores primárias, e o preto e o branco. Essas formas seriam a essência dos objetos.

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

MobiClass – Um ambiente para Educação Presencial

Glauco Todesco[1]
Rodrigo Almeida[2]

Recebido em 26. II. 2014. Aceito em 28. IV. 2014.

Resumo. Com o barateamento e a popularização dos dispositivos móveis, diversas possibilidades no desenvolvimento de aplicações voltadas especificamente para serem usadas em aulas presenciais podem ser propostas e tal perspectiva abre novos campos de pesquisas relacionadas ao tema computação e educação. Este artigo apresenta um ambiente de ensino denominado de *MobiClass*, que tem por objetivo disponibilizar conteúdos e atividades em aulas presenciais usando dispositivos móveis baseados na plataforma aberta Android. As principais características do ambiente são a de promover dentro de uma sala de aula tradicional meios de colaboração entre os alunos para resolução de exercícios e formas de ter o resultado e correção automática das atividades, permitindo que o professor tenha um retorno imediato do desempenho da classe em relação aos conteúdos apresentados em uma aula presencial. Esse artigo apresenta os principais requisitos do ambiente, a arquitetura geral do sistema, cenário de um caso de uso, as tecnologias utilizadas no desenvolvimento do protótipo e os resultados obtidos até o momento.

Palavras-chave: Aplicações Móveis; Aulas Presenciais; Objetos de Aprendizagem.

Abstract. MobiClass - An environment for Traditional Learning. With falling prices and popularization of mobile devices, the various possibilities to develop applications specifically for use in traditional classrooms may be proposals and this perspective opens new research fields on the theme computing and education. This paper presents a learning environment called MobiClass, which aims to provide contents and activities into traditional classroom using mobile devices based on the open platform Android. The main features of the environment are to promote within a traditional classroom means of collaboration among students to solve exercises and ways to get the automatic result and correction of activities, allowing the teacher to have an immediate return of the class performance in relation to the content that was presented in a traditional classroom. This article presents the main requirements of the environment, the overall system architecture, the scene of a use case, the technologies used in the development of the prototype and the results obtained.

Keywords: Mobile Application; Traditional Classroom; Learning Objects.

[1] UNASP, *Campus* São Paulo – glauco.todesco@unasp.edu.br (autor para correspondência)
[2] UNASP, *Campus* São Paulo – rodrigo.almeida@unasp.edu.br

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

1 Introdução

O uso de computadores em sala de aula ainda é tema de discussão entre educadores e pesquisadores, e na literatura é possível encontrar diversos estudos relacionados ao uso de meios digitais para a construção de objetos de aprendizagem como em Longary (2012) e Silveira (2012). Audino e Nascimento (2012) apresentam vários conceitos relacionados à definição do termo 'objetos de aprendizagem', como, por exemplo, em Spinelli (2005) que afirma que "Um objeto virtual de aprendizagem é um recurso digital reutilizável que auxilie na aprendizagem de algum conceito e, ao mesmo tempo, estimule o desenvolvimento de capacidades pessoais, como, por exemplo, imaginação e criatividade".

O problema a ser investigado nesse artigo parte de como construir objetos de aprendizagem para serem usados em sala de aula para promover formas interativas e colaborativas de apresentação e avaliação de conteúdos, tendo como premissa que professor ainda continua com a responsabilidade de criação do conteúdo, formato de apresentação e formas de avaliação. O sistema proposto é baseado na plataforma Android (PEREIRA, 2009) e assemelha-se em parte, do ponto de vista do usuário, com as tradicionais ferramentas de apresentação baseadas na metáfora de uma sequência de *slides*, porém com a inclusão de novos recursos que permitem dinamizar a condução de aulas presenciais.

2 Arquitetura do sistema

Para definir a arquitetura do sistema, os seguintes requisitos foram considerados:

- O sistema é para ser usado em sala de aula (aulas presenciais) e supõe que os alunos possuem um dispositivo móvel (um por aluno ou um por grupo de alunos);
- O sistema deve permitir a apresentação de conteúdos e a realização de atividades de forma individual ou coletiva, promovendo a colaboração entre os alunos nas atividades propostas;

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

- O professor deverá ter liberdade de escolher, organizar e definir as atividades a serem executadas em sala de aula a partir de uma ferramenta de autoria;
- O ambiente deve ser flexível para permitir a inclusão de novas funcionalidades ou novos tipos de objetos de aprendizagem;
- Cabe ao professor ou aos alunos, inserir os conteúdos das atividades a serem exploradas em sala de aula;
- O professor poderá ter acesso a qualquer momento às informações da atividade corrente em sala de aula, obtendo os dados do que um aluno ou grupo está fazendo ou ainda ter uma visão geral da turma;
- O professor poderá a qualquer momento ter acesso às todas as atividades já desenvolvidas pelos alunos e poderá usar tais informações como forma de avaliação;
- Tanto o professor como os alunos devem identificar-se para poder acessar o sistema.

Baseado nos itens acima, a arquitetura geral do sistema é apresentada. Ela é formada por três camadas básicas conforme apresentada na Figura 1.

Figura 1. Arquitetura do Sistema.

Todas as informações das atividades, professores, classes e alunos são armazenadas em um Banco de Dados. O Gerenciador do Sistema é a camada principal responsável em controlar todas as atividades e conteúdos que serão apresentados em sala de aula. Ele provê um conjunto de serviços que a Aplicação Cliente poderá utilizar como autenticação, acesso aos dados de uma classe, acesso a uma atividade, desempenho do aluno, desempenho de uma classe, controle de uma atividade, entre outros.

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

Já a Aplicação Cliente, que é executada nos dispositivos móveis, apresentará os conteúdos e atividades definidos pelo professor. Cabe ao professor definir e organizar as estruturas das suas atividades através de uma ferramenta de autoria. Para tornar flexível a inclusão de novas opções ou possibilidades na Aplicação Cliente, uma arquitetura baseada no conceito de *plug-ins* foi definida. Um *plug-in*, também conhecido como módulo de extensão, define um conceito que permite estender as funcionalidades de um sistema com facilidade. Cada *plug-in* disponibiliza uma funcionalidade específica como uma apresentação de conteúdo, questionário, um jogo, elementos 3d, entre outros. Um *plug-in* pode ser classificado como um objeto de aprendizagem reutilizável.

2.1 Cenário de uso

Para exemplificar o uso do ambiente, o seguinte cenário é apresentado. Esse cenário supõe que cada aluno possui um dispositivo móvel e que na sala de aula existe um projetor. O professor, antes da aula, deve construir a atividade através de uma ferramenta de autoria e cadastrar a atividade para uma turma de alunos de um determinado curso. Ao construir a atividade o professor define quais elementos usar e em qual ordem eles ficarão disponíveis para os alunos. Todo conteúdo apresentado deve ser inserido pelo professor ou ainda pelos alunos quando o professor assim achar necessário. Nesse cenário o professor escolheu quatro *plug-ins* para desenvolver uma nova atividade, são eles: apresentação de conteúdo, questionário, jogo da forca e jogo palavras cruzadas. A metáfora a ser usada aqui é muito próxima das que existem nas ferramentas de apresentação de conteúdos, que geralmente são baseadas em uma sequência de *slides*. Desta forma o professor organizou o seguinte roteiro da atividade a ser desenvolvida em sala de aula conforme apresentado na Figura 2:

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

1. Apresentação de conteúdo (parte 1);
2. Questionário individual com 8 perguntas (8 minutos de tempo de execução);
3. Apresentação de conteúdo (parte 2);
4. Questionário em grupo (até 4 alunos por grupo) com 12 perguntas (15 minutos de tempo de execução);
5. Jogo da forca em grupo (30 minutos de tempo de execução);
6. Jogo de palavras cruzadas individual (20 minutos de tempo de execução).

Figura 2. Roteiro de uma atividade (aula presencial).

Para o jogo da forca o professor não define nenhuma palavra, cabe aos alunos colocar as palavras e as dicas durante a aula em função do conteúdo apresentado. Já o jogo de palavras cruzadas foi definido previamente pelo professor. Ao final do processo de criação da atividade a ferramenta de autoria cria um arquivo XML (*eXtensible Markup Language* – Linguagem de Marcação Estendida) que descreve todas as relações espaciais e temporais, bem como grava todas as informações da atividade no Banco de Dados. A Figura 3 apresenta o formato do arquivo *activity.xml* gerado pela ferramenta de autoria.

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<activity id='1'>

<plugin>
    <presentation id='1'>
        <slide id='1' >  <!-- Slide data -->  </slide>
        <slide id='2' />
    </presentation>
</plugin>

<plugin>
    <quiz id='1' maxTime='8' group='false'>
        <question id='1' type='1'> <!-- Question data -->  </question>
        <question id='2' type='2' />
    </quiz>
</plugin>

<plugin>
    <presentation id='2'>
        <slide id='3' > <!-- Slide data --> </slide>
        <slide id='4' />
    </presentation>
</plugin>

<plugin>
    <game type='forca' id='8' group='true' maxTime='14'>
        <!-- game data -->
    </game>
</plugin>

<plugin>
    <game type='crossword' id='10' group='false' maxTime='10'>
        <!-- game data -->
    </game>
</plugin>

</activity>
```

Figura 3. Formato do arquivo activity.xml.

A ordem de que cada *plug-in* aparece no arquivo define também a ordem de apresentação dos mesmos. Dentro de um elemento <*plugin*> é definido o tipo da atividade: <*presentation*>,<*quiz*> e <*game*>. A partir do arquivo XML a Aplicação Cliente saberá como montar a atividade nos dispositivos dos alunos. Cada tipo de elemento XML possui um identificador único que permite que a aplicação busque por informações desse elemento na base de dados. Outra opção a ser avaliada é de passar todas as informações da atividade pelo arquivo XML. No dia da aula o professor libera a atividade para que a mesma possa ser acessada pelos alunos e carrega a aplicação, apresentando a mesma no projetor da sala de aula. Os alunos devem fazer a autenticação no ambiente para poder acessar a atividade do dia. Ao selecionar a atividade, a aplicação carrega o arquivo activity.xml e constrói a atividade conforme a estrutura apresentada na Figura 2. A primeira parte da atividade é a apresentação

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

de conteúdo que é linear, ou seja, controlada pelo professor, na relação mestre/escravo. Se o professor troca o conteúdo da apresentação, a mesma é trocada no projetor e nos dispositivos móveis dos alunos. Desta forma, o que o professor está mostrado no projetor, está sendo apresentado também nos dispositivos dos alunos. Os alunos não poderão avançar a apresentação e terão que seguir a mesma sequência definida pelo professor.

Após apresentar a primeira parte do conteúdo, um questionário é apresentado para os alunos com oito perguntas e o mesmo deve ser respondido de forma individual. Nesse momento o tempo estipulado pelo professor começa a ser contato para que os alunos respondam as perguntas. Após o termino desse tempo ou quando todos os alunos responderem as perguntas é apresentado o desempenho de forma individual para cada aluno e no projetor o resultado do desempenho da classe. Se o desempenho da classe não for satisfatório, o professor poderá rever o conteúdo desejado. O resultado de uma avaliação de forma instantânea é um bom indicativo para que o professor tenha meios de avaliar a classe de forma rápida e ainda descobrir de forma individual ou coletiva quais conteúdos devem ser revistos.

Supondo que o desempenho do questionário 1 foi satisfatório o professor continuará apresentado a segunda parte do conteúdo. Ao fim da apresentação da parte 2 do conteúdo um novo questionário é apresentado para a classe, só que agora ele deve ser feito de forma coletiva (em grupo). O próprio ambiente cria os grupos de alunos de forma dinâmica e os alunos se reúnem para resolver o segundo questionário. Diversos critérios podem ser adotados para a criação dos grupos como desempenho anterior e rotatividade de grupos para possibilitar a criação de grupos diferentes todas as vezes. Após o termino do tempo para responder o segundo questionário ou a conclusão do questionário pelos grupos, o resultado do mesmo é apresentado para os grupos e no projetor da sala é apresentado o resultado da classe. Cabe novamente ao professor rever algum conteúdo ou ainda passar para a próxima atividade que é o jogo da forca. No jogo da forca cada grupo define um conjunto de palavras e suas dicas, de modo que cada grupo jogue com as palavras criadas pelos outros grupos. Após o termino desta atividade o professor poderá fazer algum comentário ou ainda apresentar o desempenho da classe. A última atividade da aula é o jogo de palavras cruzadas que deverá ser respondido de forma individual pelos alunos.

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

Finalmente após o termino de todas as atividades o professor poderá novamente apresentar o desempenho final da classe e fazer o fechamento da aula. Todo o desempenho do aluno será guardado no Banco de Dados (atividades em grupo e atividades individuais) e o professor poderá usar esse desempenho na composição da média ou menção final de cada aluno.

3 Desenvolvimento

A seguir serão apresentadas as principais partes relacionadas ao desenvolvimento do sistema.

3.1 Banco de dados

Os principais elementos da arquitetura do banco de dados para suportar o ambiente proposto foram implementados em SQLServer. O núcleo da arquitetura permite gerenciar alunos, professores, classes e atividades, porém o banco de dados crescerá conforme a inclusão de novas funcionalidades no ambiente através da proposição de novos *plug-ins*. Na abordagem atual todas as informações serão armazenadas no banco de dados e desta forma não é possível que as aplicações dos alunos trabalhem de forma *off-line*.

3.2 Gerenciador do sistema

O gerenciador do sistema é o principal elemento da arquitetura proposta por prover um conjunto de serviços que permitem que as aplicações dos alunos acessem as informações necessárias para realizar uma atividade. A abordagem utilizada foi de construir um conjunto de serviços baseados em duas tecnologias: Serviços Web (SAMPAIO, 2006) e Serviços de Mensagens (GOOGLE, 2013). A principal diferença entre um serviço web e um serviço de mensagem é de quem é o responsável em chamar o serviço ou disparar o mesmo. Em serviços web as aplicações clientes é que são responsáveis em chamar um serviço disponível no

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

gerenciador do sistema e já no serviço de mensagens o gerenciador do sistema é quem dispara a sua execução para enviar algum tipo de informação para as aplicações móveis. Quatro grupos de serviços foram definidos, sendo os Serviços de Autenticação, Serviços Acadêmicos e Serviços de Atividades como serviços web e os Serviços de Aplicativos como serviços de mensagem.

Serviços de autenticação têm por objetivo permitir e controlar o acesso dos alunos e professores.

Os serviços acadêmicos visa oferecer facilidades para o gerenciamento de alunos, professores, classes, turmas e disciplinas. Por exemplo: ao ser autenticado no sistema a Aplicação Cliente deve obter as suas disciplinas de um aluno. Do lado do aplicativo do professor é necessário que o mesmo tenha mecanismos para obter informações sobre as suas disciplinas, atividades e alunos.

Os serviços de atividades oferecem mecanismos para cadastrar uma atividade e meios de controlá-las como visibilidade, início, fim, bloqueio, desempenho, entre outras.

Os serviços de aplicativos oferecem mecanismos genéricos de controle para as aplicações, como controle do andamento das atividades propostas, controle de como os grupos ou alunos irão trabalhar ou ainda como será organizado a participação de cada grupo ou aluno. Alguns aspectos que o professor deve definir em uma atividade acabem usando os serviços de aplicativos, por exemplo: todos interagem ao mesmo tempo ou somente um grupo ou aluno por vez interage com a aplicação cliente para responder a um questionário ou realiza uma jogada? De quem é a vez de jogar ou realizar uma interação (grupo ou aluno). Qual é o próximo grupo/aluno a jogar? Um exemplo de como aplicar tais serviços seria no cenário de uso apresentado no item 2.1, no qual um aluno ou grupo ao errar uma reposta de um questionário ou uma letra do jogo da forca, passaria a vez para o próximo aluno ou grupo.Para cada atividade corrente de uma sala de aula, o Gerenciador do Sistema cria uma sessão para controlar o andamento da atividade. Os dois principais dados guardados em uma sessão são: quais alunos estão participando da atividade e qual é o *status* de cada aluno.

Todos os serviços formam implementados em Java 1.7 usando como servidor de aplicação o TomCat 7.0 e a API (*Application Programming Interface* – Interface de Programação de Aplicativos) de persistência JPA (*Java Persistence API*) 2.0 para fazer o

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

mapeamento objeto relacional. Os serviços web foram construídos baseados na abordagem SOAP (*Simple Object Access Protocol* – Protocolo Simples de Acessos a Objetos), usando os *plug-ins* da ferramenta de desenvolvimento Eclipse denominadas de *Axis2-codegen 1.4* e *Axis2-archiver 1.4.* Os serviços de mensagens estão baseados na especificação do *Google Cloud Messaging* – GCM (ANDROID, 2013), que provê a estrutura necessária para que uma aplicação do lado do servidor comunique-se com as aplicações clientes quando for necessário.

3.3 Aplicação aluno/professor

A Aplicação Cliente foi implementada para a plataforma Android (*Software Development Kit* – Kit de Desenvolvimento de Software, API versão 18). Para consumir os serviços web foi utilizado a biblioteca *ksoap2-2.3* que realiza as chamadas SOAP (CHAPPEL, 2002), além da API *com.google.android.gcm* para realizar o recebimento dos serviços de mensagens.

O início da aplicação começa com a autenticação do aluno, escolha da disciplina e atividade. Ao escolher a atividade o arquivo XML apresentado na Figura 2 referente à atividade escolhida pelo aluno é carregado usando a biblioteca *XMLPullParse*. Com o *parse* do arquivo *activity.xml* finalizado a aplicação saberá quais *plug-ins* serão usados e cria uma estrutura de lista para armazenar a ordem de apresentação de cada *plug-in*, sendo que o primeiro *plug-in* da lista de atividades é carregado e apresentado para o aluno e professor. Para tornar o sistema mais leve a opção inicial foi de carregar um *plug-in* por vez, ou seja, somente as informações necessárias para carregar o primeiro *plug-in* são carregadas do banco de dados e somente quando o primeiro *plug-in* for finalizado os dados do segundo *plug-in* serão carregados. Tal abordagem poderá ser revista futuramente em função do desempenho e do tempo de carregamento dos *plug-ins*.

A estrutura básica de um *plug-in* é formada pela extensão da classe *mobiclass.activity.PlugIn* que possui, entre outros, os seguintes métodos básicos: *init()*, *start()*, *stop()*, *end()*, *back(), next()* e *getPerformance()*. Através desses métodos é possível controlar toda a atividade de um *plug-in* e encadear os eventos necessários para que o cenário

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

apresentado no item 2.1 seja possível de ser executado. A classe *mobiclass.activity.PlugIn* é uma extensão da classe *android.app.Activity* com a inclusão dos métodos de controle.

Outra parte importante do sistema está relacionada em como a Aplicação Cliente interage com os *plug-ins* para perceber os seus eventos. Para tratar desses aspectos a interface *mobiclass.activity.PlugInEvent* foi definida e ela deve ser implementada pela classe principal do projeto que também deve ter uma instância da classe *mobiclass.activity.PlugInListener* que é a responsável em escutar os eventos dos *plug-ins*.

Um evento importante que um *plug-in* deve informar para a Aplicação Cliente é quando ele é finalizado, como por exemplo, fim de um questionário. Quando a aplicação é notificada desse evento, a mesma verifica na lista de atividades qual é o próximo *plug-in* a ser carregado (chamada do método *init()* da classe *mobiclass.activity.PlugIn*), porém esse só poderá ser iniciado (chamada do método *start()* da classe *mobiclass.activity.PlugIn*) quando a Aplicação Cliente for notificada pelo Gerenciador do Sistema através do serviço de mensagens. Outro aspecto importante da arquitetura do sistema é que os *plug-ins* não possuem qualquer interface direta com o Gerenciador do Sistema, somente a Aplicação Cliente.

Toda vez que um evento é recebido pela Aplicação Cliente, a mesma pode fazer chamadas aos serviços disponibilizados e apresentados no item 3.2 como, por exemplo, uma chamada para salvar desempenho do aluno ou informar que um aluno finalizou uma das atividades apresentadas na Figura 2, para que o Gerenciador do Sistema atualize as informações da sessão da atividade correspondente.

3.4 *Plug-In* Quiz

O primeiro *plug-in* desenvolvido permite a criação de questionários de múltipla escolha. Após a construção do objeto *QuizPlugIn* (*mobiclass.plugin.QuizPlugIn*), a Aplicação Cliente através do arquivo *activity.xml* da Figura 3 carrega o identificador do questionário e usa o mesmo como argumento na chamada de um Serviço Web que retorna um arquivo XML que contém a estrutura do questionário: perguntas, alternativas e respostas. A Figura 4

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

apresenta um trecho de um arquivo que contém a estrutura de um questionário retornado por um serviço web para uma aplicação cliente.

```
<?xml version="1.0" encoding="UTF-8"?>
<quiz>
    <category>

                <title>Quiz de Matemática</title>
                <text>Matemática módulo 1</text>
                <question>
                        <text> Qual é a raiz de 2? </text>
                        <choice>1</choice>
                        <choice>1.634</choice>
                        <choice>1.414</choice>
                        <choice>2</choice>
                        <answer>2</answer>
                </question>
                <question>
                        <text>Quanto é 30/2?</text>
                        <choice>20</choice>
                        <choice>12</choice>
                        <choice>15</choice>
                        <answer>2</answer>
                </question>
        </category>
</quiz>
```

Figura 4. Arquivo XML de um questionário.

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

4 Resultados e conclusões

A Figura 5 apresenta as principais telas do protótipo desenvolvido.

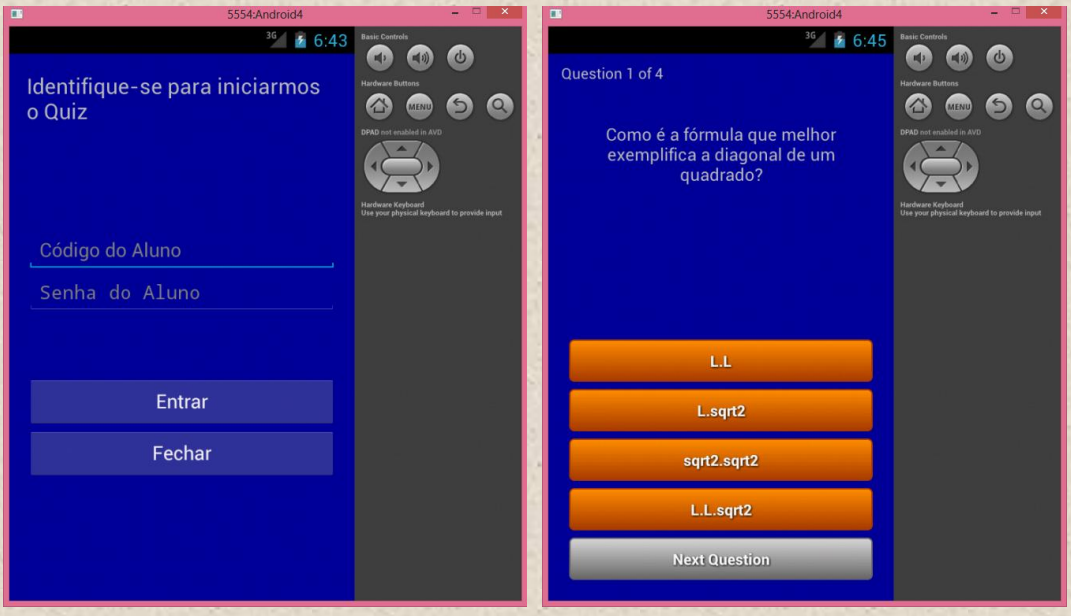

Figura 5. Telas do Protótipo com o Plug-in Quis.

A imagem da esquerda é a tela de autenticação do aluno e a imagem da direita apresenta uma das perguntas de um questionário. Os resultados obtidos até o momento motivam a continuação da pesquisa. O desenvolvimento do sistema encontra-se em fase adiantada com os principais elementos apresentados na Figura 1 especificados e prototipados. Por ser um projeto amplo e com várias possibilidades, várias propostas de trabalhos futuros são apresentadas:

- Criação de uma ferramenta de autoria para gerar as atividades;
- Criação de um sistema de gerenciamento de alunos, professores, cursos, classes e atividades;
- Interfaces com outros ambientes de ensino como o Moodle (2013);

Revista de Ciência, Tecnologia e Cultura da FATEC Itu
Itu/SP, n°. 3, p. 179 – 192, junho de 2014.

- Desenvolvimento de *plug-ins* genéricos para serem usados em qualquer disciplina;
- Desenvolvimento de ferramentas de anotação;
- Ferramentas de análise de desempenho.

5 Referências Bibliográficas

AUDINO, D. F.; NASCIMENTO, R. S. "**Objetos de Aprendizagem - Diálogos entre Conceitos e uma Nova Proposição Aplicada à Educação**". Em Revista Contemporânea de Educação, Vol. 5, No. 10, 2010. páginas 128–148.

GOOGLE**. "Google Cloud Messaging".** Disponível em: http://developer.android.com/google/gcm. 2013. Acesso em Dezembro de 2013.

MOODLE**. "A Course Management System".** Disponível em: https://moodle.org. Acesso em Dezembro de 2013.

PEREIRA, S. M. L.; OLIVEIRA, L. C. "**Android para Desenvolvedores**". 1ªEdição, São Paulo: Brasport Livros e Multimedia, 2009, 223 Páginas.

SILVEIRA, S. M.; CARNEIRO, M. L. F. "**Desconstruindo Objetos de Aprendizagem: reflexões sobre sua qualidade de uso**". XXIII Simpósio Brasileiro de Informática na Educação – SBIE 2012, Rio de Janeiro.

SPINELLI, W. "**Aprendizagem Matemática em Contextos Significativos: Objetos Virtuais de Aprendizagem e Percursos Temáticos**". Dissertação. Universidade de São Paulo. São Paulo. 2005.

SAMPAIO, C. "**SOA e WebServices em Java**". Rio de Janeiro, Brasport, 2006, 260 Páginas.