An M-Learning Android Application

http://dx.doi.org/10.3991/ijes.v2i1.3677

R.S. Minto, CEFET-RJ/DEPIN, Rio de Janeiro, Brazil
V.F.M. dos Santos, CEFET-RJ/DEPIN, Rio de Janeiro, Brazil
F. Paschoal Junior, CEFET-RJ/DEPIN, Rio de Janeiro, Brazil

Abstract—This paper presents the m-Learning application for Android that allows the management of the didactical events of a teacher, including the management of the classes, works, contents, notes, communications and the access of your students. The students can access the information that the teacher publishes in this application.

Index Terms—Android, M-learning, mobile devices, virtual learning environment.

Aplicativo de M-Learning para Android

R.S. Minto, V. F. M. dos Santos and F. Paschoal Junior
CEFET-RJ/DEPIN, Rio de Janeiro, Brasil

***Resumo*—Este trabalho apresenta o aplicativo de M-Learning para Android, o qual permite o gerenciamento de eventos didáticos de um professor, incluindo o gerenciamento das aulas, trabalhos, conteúdos, notas, comunicados e o acesso de seus estudantes. Os estudantes podem acessar as informações que o professor disponibiliza nesta aplicação.**

***Palavras-chave*—Android, M-learning, mobile devices, virtual learning environment.**

I. INTRODUÇÃO

O uso das tecnologias permite um mundo cada vez mais globalizado e conectado, onde pessoas executam suas tarefas cotidianas.

A evolução da Internet, que sem dúvida segue cada dia mais veloz e acessível a todas as classes sociais, incluindo classes de baixa renda através de programas sociais e do barateamento de serviços, foi fundamental à massificação criada em consequência ao crescimento econômico e inclusão de consumidores, onde uma quantidade expressiva de dispositivos portáteis é encontrada em lojas por todo o país. Estes dispositivos são adquiridos com planos de dados e Internet com grande facilidade, os quais permitem o acesso sem fio à Internet.

Ainda assim, a educação, um dos setores mais criticados do país, evolui de modo discreto e absorve influências diretas deste avanço, com o uso de uma metodologia que vem aprimorando há alguns anos em diversas instituições de ensino: a educação à distância (EAD). Por meio desta, cursos e disciplinas são aplicados de modo não presencial.

Caldeira [1] destaca a não necessidade de um lugar concreto para acontecer a realização de um curso, mas afirma que, para isso acontecer, é preciso disciplina do aluno. Também destaca a importância do texto na aprendizagem, pois com ele é possível a comunicação entre pessoas e a troca de informações.

A. Motivação

A motivação deste trabalho foi desenvolver um aplicativo construído a partir do relacionamento de técnicas e conceitos associados com a Internet, o ambiente virtual de aprendizagem, a educação à distância e o desenvolvimento de softwares para dispositivos móveis.

O m-learning (*mobile learning*), ramificação da EAD, aparece como foco principal para desenvolver uma aplicação em plataforma Android, sendo um aplicativo que permite realizar funções que auxiliem na aprendizagem em ambientes virtuais especificamente para usuários de dispositivos móveis.

Segundo Serrão et al. [2], o m-learning pode ser definido como "qualquer tipo de aprendizagem que ocorre quando o estudante não está em um local fixo, predeterminado, ou quando este tira proveito das oportunidades oferecidas por tecnologias móveis".

Diante da exigência de aprendizagem e atualização de informações torna-se necessário o acesso aos dados e informações independentemente de localidade ou tempo, como forma de estar sempre atualizado sobre qualquer assunto a todo o momento.

Meirelles e Tarouco [3] mostram uma visão interessante sobre o m-learning. Dizem que no mundo tão globalizado de hoje e com a necessidade de cursos para capacitação dos profissionais, o ensino móvel é uma área que se mostra interessante pela disponibilidade de se ter o material em suas mãos a todo o momento.

Além disto, a venda gigantesca de dispositivos móveis, sejam eles celulares smartphones ou tablets, cresce expressivamente alavancando o mercado de desenvolvimento de softwares para estes fins. A cada dia, surgem centenas de novos apps – nome atribuído aos aplicativos desenvolvidos para plataforma móvel – despertados pelo interesse em um mercado em crescimento e de grande potencial.

Todavia, a porcentagem de aplicativos voltados ao meio educacional é relativamente pequena se comparada às demais categorias, comprometendo a expansão da EAD em curto prazo e, consequentemente, a aprendizagem virtual. Por isto, faz se necessário a criação e desenvolvimento de mais apps que viabilizem o m-learning.

O objetivo deste trabalho foi produzir um aplicativo no qual seja possível um criar disciplina, disponibilizar comunicados e avaliar os alunos inscritos em determinada disciplina. O aplicativo aceita a inscrição de alunos nas disciplinas solicitadas e emite mensagens de alerta através de e-mails. Com isso, espera-se utilizar o conceito de ubi-

quidade, melhorando a comunicação entre aluno e professor.

Franciscato e Medina [4] descrevem que os benefícios do mobile learning podem ser dados ao professor e aos alunos. Com isso, no aplicativo desenvolvido por este trabalho os alunos terão o material disponível a qualquer momento, podendo ser acessado onde quiserem. E os professores terão uma interação facilitada com os alunos para a publicação de informações.

II. CONCEITOS

Esta seção destaca os principais conceitos que complementaram e motivaram o desenvolvimento deste aplicativo.

A. Educação à Distância - EAD

A EAD, como forma de ensino que revoluciona o modo de interação entre professores e alunos, está submetida ao conceito contemporâneo de tempo e espaço nas salas de aula. Ela surgiu da necessidade de preparo profissional e cultural de pessoas que não podem, por algum motivo, frequentar salas de aulas diariamente. Após evoluir, não somente através de novas tecnologias, mas também do aprimoramento de tecnologias existentes, tornou possível a ampliação dos inúmeros cursos disponibilizados através da Internet.

De acordo com Silva et al. [5], a educação à distância é "uma modalidade de ensino e aprendizagem que vem crescendo há alguns anos. De acordo com pesquisa realizada pela Associação Brasileira de Ensino à Distância (ABED) e pelo Ministério da Educação (MEC), a demanda em cursos de especialização à distância aumentou 60% de 2008 a 2010". E, pensando que em dois anos ocorreu o aumento de 60%, é possível imaginar o quanto mais cresceu de 2010 a 2013. Portanto, o uso do ensino à distância através dos dispositivos móveis pode ser benéfico ao seu público alvo.

Assim sendo, Rodrigues [6] diz que com o grande crescimento e disponibilidade de dispositivos móveis, aliada à utilização no ensino, originou o surgimento do m-learning.

O m-learning permite aos alunos e aos professores uma maior interação, de forma que a informação e/ou o conhecimento saiam dos ambientes físicos das instituições e conquistem outros espaços, em diferentes momentos.

B. Dispositivos Móveis

O crescimento na utilização de dispositivos móveis juntamente com a vasta quantidade de cursos à distância que encontramos, originou a necessidade de se obter informação em qualquer local e momento.

Conforme Fernandes et. al [7]: "O uso de dispositivos móveis amplia as possibilidades de ensino sem limites geográficos e temporais".

Destaca-se ainda o aumento significativo do uso de dispositivos móveis para várias finalidades, mesmo que antigos dispositivos já fossem fabricados com alguma finalidade específica, restringindo sua utilização, como exemplo temos os primeiros celulares que só ligavam.

C. Android – Plataforma do passado, presente e futuro

A plataforma que mais cresce sem dúvida é a do Android que, por ser um sistema operacional utilizado em aparelhos por diversas empresas, agiliza uma padronização natural a qual facilita o desenvolvimento de aplicações para grupos de dispositivos móveis.

Mostrando ser cada vez mais promissora, mesmo tendo ainda muito a evoluir, pode-se dizer que atualmente essa plataforma é uma realidade, observando o fato de que a maioria dos smartphones disponíveis no mercado utiliza este sistema operacional.

III. MODELAGEM

Esta seção demonstra os principais diagramas da modelagem criada para o desenvolvimento deste aplicativo.

A. Diagrama de Caso de Uso

A Figura 1 demonstra o Diagrama de Caso de Uso do aplicativo desenvolvido neste trabalho, o qual indica as funcionalidades da aplicação e os atores que executam essas funcionalidades na aplicação.

B. Diagrama de Classes

Como prova de conceito à aplicação foi desenvolvido o Diagrama de Classes do aplicativo, demonstrado na Fig. 2.

O modelo de classes tem como sua principal característica representar as partes estruturais do sistema, sendo que cada classe representa uma entidade que, por sua vez, possuem seus respectivos atributos.

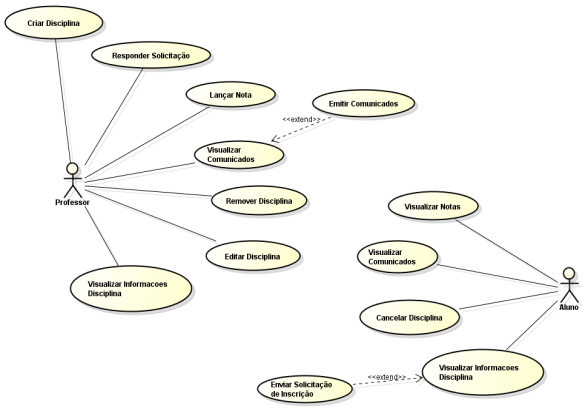

Figura 1. Diagrama de Caso de Uso

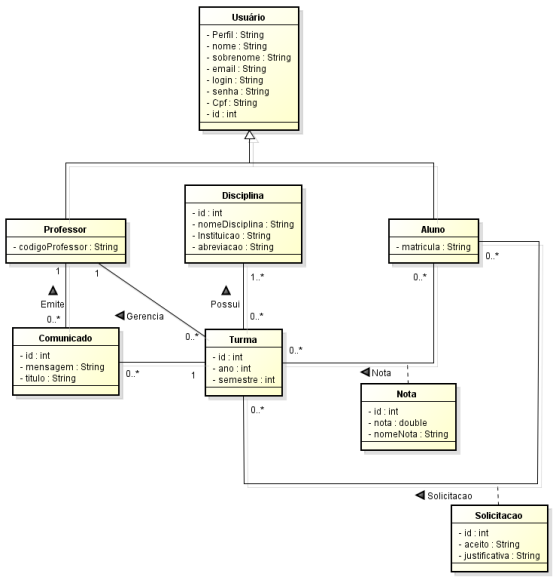

Figura 2. Diagrama de classes

C. *Hierarquia das Telas*

A Figura 3 representa o caminho entre as telas disponíveis do sistema para quem tiver o perfil de professor, após ser feito o login para navegação.

Cada tela da aplicação é uma *Activity* da aplicação. O perfil de usuário de professor concentra a maior parte das funcionalidades da aplicação e, consequentemente, a hierarquia de telas de seu perfil possui uma quantidade maior de telas em relação à hierarquia de telas do perfil de usuário de aluno.

A Figura 4 representa o caminho entre as telas disponíveis do sistema para quem tiver o perfil de aluno, após ser feito o login, para navegação no aplicativo.

Podemos notar que o número de telas disponibilizadas para o aluno é bem menor do que para o professor.

IV. IMPLEMENTAÇÃO

O processo de desenvolvimento da aplicação foi iniciado pela instalação do JDK (*Java Development Kit*), o qual é um kit de desenvolvimento que possui as ferramentas necessárias para desenvolver em linguagem Java.

Após isto, foi instalado o Android SDK (*Software Development Kit*), o qual é um kit de desenvolvimento de aplicações para serem executadas em plataforma Android.

Em sequência, foi instalado o Eclipse, o qual é uma IDE (*Integrated Development Environment*) desenvolvida em Java e que foi escolhida por sua interface simples e funcional.

Posteriormente, foi instalado sob o Eclipse um *plugin* para o desenvolvimento em Android, chamado de ADT (*Android Development Tools*).

Foi utilizado para o armazenamento dos dados o banco de dados SQLite, o qual é um banco nativo da plataforma, é formado por uma pequena biblioteca desenvolvida em linguagem C e que tem acesso às operações SQL. O SQLite é uma opção gratuita e altamente funcional.

Finalmente, para efetuar os testes do funcionamento do aplicativo foi utilizada a opção disponibilizada pelo Eclipse que possibilita criar máquinas virtuais para realizar validações de testes.

A. *Arquitetura da Aplicação Android*

Para uma melhor compreensão da arquitetura desta aplicação, é preciso conhecer algumas informações sobre aplicações de plataforma Android. A arquitetura de aplicações em Android é dividida em camadas como demostrado na Figura 5.

O Android é executado sobre um Kernel Linux, que foi desenvolvido utilizando o sistema operacional Linux. É nessa camada que encontramos os programas de gerenciamento de memória, de configurações de segurança e também os muitos drivers de hardware [8].

A máquina virtual é a *Dalvik Virtual Machine*, uma tecnologia de software livre que é utilizada pelo Android para rodar cada aplicação com o seu próprio processo. Um ponto positivo dessa camada é de ter processos independentes onde nenhuma aplicação é dependente uma da outra, ou seja, se a execução de uma aplicação for interrompida ela não afetará as outras aplicações que estiverem sendo executadas [8].

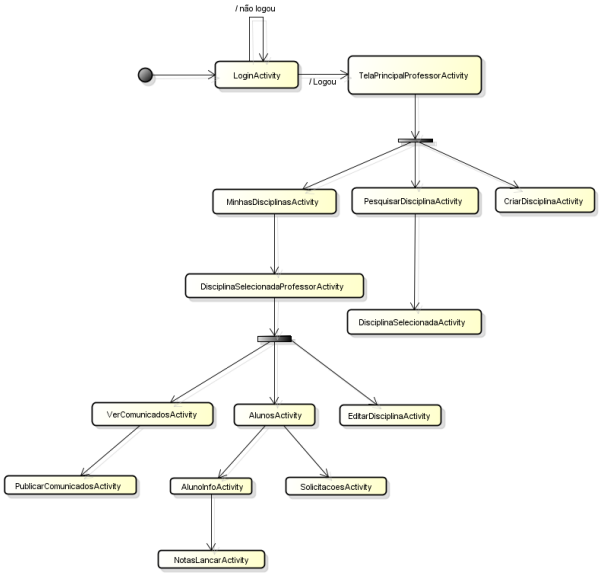

Figura 3. Hierarquia das telas do perfil de professor

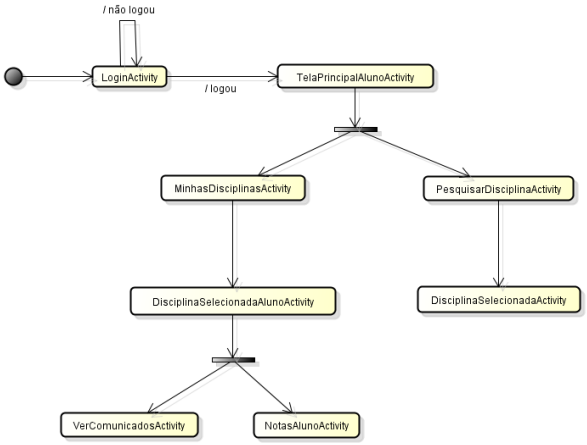

Figura 4. Hierarquia das telas do perfil de aluno

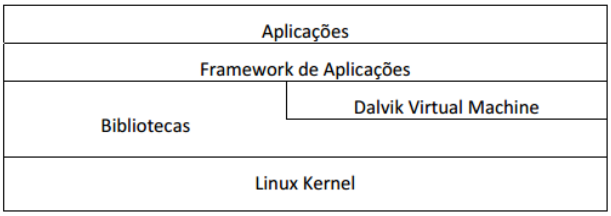

Figura 5. Arquitetura de aplicações Android [8]

Na camada Bibliotecas temos vários comandos que dizem ao dispositivo como ele deve interagir com os diferentes tipos de dados que ele lida [8].

Na camada Framework de Aplicações temos os programas que gerenciam as funções básicas do telefone [8].

Na camada Aplicações é onde se encontram as funções básicas que são gerenciadas pelo framework da aplicação, é aonde a interação do usuário com o dispositivo móvel é realizada através dos aplicativos instalados [8].

B. *Arquitetura da Aplicação*

A arquitetura utilizada pela aplicação desenvolvida é composta por três camadas, como demonstrado na Figura 6.

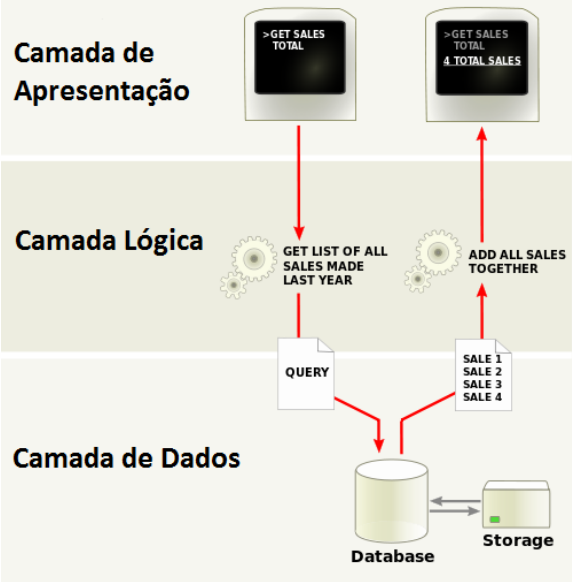

Figura 6. Camadas de arquitetura do ambiente de desenvolvimento [9]

A Camada de Apresentação permite a interação do usuário com a tela e é onde os usuários fazem as requisições de dados.

A Camada Lógica contém o servidor de aplicações e aqui é realizada a requisição a um banco de dados para retornar os pedidos do usuário.

A Camada de Dados contém o banco de dados, onde as consultas são feitas e as informações são guardadas.

A estrutura de uma aplicação Android contém o arquivo AndroidManifest.xml, onde são encontradas as informações importantes de configuração do aplicativo, como a declaração de classes, eventos e permissões. A Figura 7 demonstra um trecho do desenvolvimento feito neste arquivo.

O AndroidManifest.xml é o arquivo mais importante do sistema, constituindo a base da aplicação Android. Se as informações não forem declaradas ou forem declaradas incorretamente neste arquivo, ocorrerá um erro ao tentar executar a aplicação.

C. *Arquitetura Model,View e Controller - MVC*

A arquitetura de projeto utilizada no desenvolvimento da aplicação foi a do padrão MVC (*Model*, *View* e *Controller*) do Android. A Figura 8 demonstra esse padrão.

Este padrão, além de fácil implementação, permite dividir as funcionalidades do software nessas três camadas, as camadas *Model*, *View* e *Controller*.

A camada *Model* foi usada para administrar as informações de uma forma mais detalhada, sendo requisitada, sempre que possível, para realizar consultas, cálculos e as regras de negócio. Tem acesso às informações provenientes do banco de dados.

A camada *View* foi utilizada para realizar a comunicação com o usuário, sendo responsável por tudo que o usuário final visualiza. Essa camada não deve ter a lógica de negócio, pois essas tarefas devem ficar para a camada *Model*.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.com.cefetrj.activities"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <uses-permission
        android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/moodle"
        android:theme="@style/AppTheme" >
        <activity
            android:name="br.com.cefetrj.activities.MainActivity"
            android:label="@string/moodle" >
        </activity>
         <activity
            android:name="br.com.cefetrj.activities.LoginActivity"
            android:label="@string/moodle" >
        </activity>
        <activity
            android:name="br.com.cefetrj.activities.ApresentacaoActivity"
            android:label="@string/moodle" >
             <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="br.com.cefetrj.activities.CadastroActivity"
            android:label="@string/moodle" >
        </activity>
        <activity
            android:name="br.com.cefetrj.activities.TelaPrincipalAlunoActivity"
            android:label="@string/moodle" >
```

Figura 7. Estrutura do arquivo AndroidManifest.xml

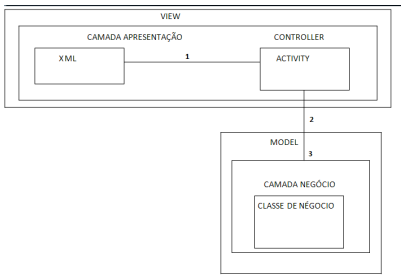

Figura 8. Representação da arquitetura MVC no Android [10]

A camada *Controller* foi utilizada para controlar todo o fluxo de informação do sistema, pegando a informação da camada *Model* e passando para a camada *View*, para ser visualizada pelo usuário.

Durante a execução da aplicação o usuário faz a sua interação com o XML, na forma de requisição de um dado pela aplicação. A *Activity* (tela da aplicação), que faz referência a esse XML, é chamada e a interação com o *Controller* é realizada. O *Controller* realiza a comunicação com o *Model* e faz a consulta que foi pedida. O *Model* recebe os dados, faz a validação, e, se estiver tudo correto, retorna o que foi solicitado pelo usuário.

D. *Aplicação*

São apresentados nessa seção os passos que foram seguidos para a implementação das telas do sistema. Após a criação do projeto "android", as classes professor, aluno, comunicado, disciplina, nota, solicitações, turma e usuário também foram criadas dentro do pacote "br.com.cefetrj.entidade". A Figura 9 apresenta o código de um documento XML que representa uma tela do sistema, a tela de alunos, a qual exibe a lista de alunos de uma determinada disciplina.

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ListView android:layout_width="fill_parent"
      android:layout_height="300dp"
      android:id="@+activity_alunos/listaDeALunos">
    </ListView>

    <RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

        <Button
            android:id="@+activity_alunos/btnVerSolicitacoes"
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="50dp"
            android:layout_centerHorizontal="true"
            android:text="@string/btnVerSolicitacoes"
            style="@style/letrasBotoes" />

    </RelativeLayout>

</LinearLayout>
```

Figura 9. Estrutura de uma tela Android em XML

A montagem das telas também é declarada dentro do arquivo XML que representa a tela da aplicação Android. O *LinearLayout* é um layout de tela que representa uma forma linear de distribuir os elementos (botões, texto, etc.) dentro da tela. O *RelativeLayout* é um layout de tela onde os elementos podem ser colocados em qualquer parte da tela, sem uma distribuição linear.

A *ListView* representa uma lista de alunos que vai ser preenchida se a disciplina contiver alunos. O *Button* é colocado dentro do *RelativeLayout* pois o seu alinhamento é colocado ao centro, o que pode também ser feito pelo *LinearLayout*, mas no *RelativeLayout* essa tarefa é facilitada.

O próximo passo foi criar uma classe no pacote "br.com.cefetrj.activities" para referenciar a tela que fora analisada. A Figura 10 demonstra um trecho desta classe.

Essa *Activity* deve ser declarada no arquivo AndroidManifest.xml e logo depois deve ser colocado o "extends Activity" para essa classe herdar a classe "android.app.Activity", a tornando uma atividade no Android.

Após estender a classe, é obrigatório escrever o método "onCreate", aonde será selecionado o arquivo XML que essa classe faz referência. Em "setContentView(R.layout.activity_alunos);" é feita essa associação, aonde activity_alunos é o nome do XML.

A Figura 11 demonstra como o *Model* é chamado a partir de uma *Activity*, onde ainda é utilizada a *AlunosActivity* com o método "PreencherListaAlunos()".

Na primeira linha deste método foi referenciada a "listaDeAlunos" na ListView "lv", que é uma *ListView* do documento XML".

Posteriormente, foi criado um *ArrayAdapter* com um *ArrayList* de String ao "*listAdapter*". Estas explicações não estão relacionadas com a chamada do *Model*, mas sim com o preenchimento da lista de alunos. No objeto "turmaModel = new(getApplicationContext());", foram atribuídos ao *Model* todas as configurações e recursos desta *Activity*, para serem usados posteriormente.

No trecho "turmaModel.getIdsAlunoTurma (perfilDados.getIdDisciplinaSelecionada());" é chamado um método da classe TurmaModel para obter os identificadores únicos dos alunos de uma determinada disciplina, passando como parâmetro o identificador da disciplina.

Como o identificador da disciplina estará sempre preenchido, pois para estar nessa tela o usuário deve selecionar uma disciplina anteriormente, logo não tem a validação dentro do método. Em "turmaPersistencia = TurmaPersistencia.getInstance(context);" as configurações e recursos da *Activity* que chamou este método serão passadas para a classe TurmaPersistencia, que é responsável por fazer a comunicação com o banco de dados. No "*return*", após ser realizada a consulta no banco, o método retorna um *array* de inteiros para a *Activity* que chamou esse método.

```
public void PreencherListaAlunos() throws Exception{

    lv = (ListView) findViewById(R.activity_alunos.listaDeALunos);

    listAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, listaAlunos);
    turmaModel = new TurmaModel(getApplicationContext());
    alunoModel = new AlunoModel(getApplicationContext());
    usuModel = new UsuarioModel(getApplicationContext());

    ArrayList<Integer> listaIdsAluno = turmaModel.getIdsAlunosTurma(perfilDados.getIdDisciplinaSelecionada());
    final ArrayList<String> ListaCpfAlunos = new ArrayList<String>();
```

Figura 10. Estrutura de um método que chama uma classe modelo

```
package br.com.cefetrj.activities;

import java.util.ArrayList;

public class AlunosActivity extends Activity {

    Button btnVerSolicitacoes;
    private ListView lv;
    private ArrayAdapter<String> listAdapter ;
    ArrayList<String> listaAlunos = new ArrayList<String>();
    private Intent intent;
    DadosLogado perfilDados = new DadosLogado();
    private AlunoModel alunoModel;
    private TurmaModel turmaModel;
    private UsuarioModel usuModel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alunos);
```

Figura 11. Estrutura de uma *Activity*

```
package br.com.cefetrj.model;

import java.util.ArrayList;

public class TurmaModel {

    private Context context;
    private TurmaPersistencia turmaPersistencia;

    public TurmaModel(Context context) {

        this.context = context;
    }

    public ArrayList<Integer> getIdsAlunosTurma(int idDisciplina)
            throws Exception{

        turmaPersistencia = TurmaPersistencia.getInstance(context);
        return turmaPersistencia.getIdsAlunosTurma(idDisciplina);
    }
```

Figura 12. Estrutura de uma classe modelo demostra o método da classe TurmaModel.

A Figura 13 demonstra um trecho da classe TurmaPersistencia, a qual faz a comunicação com o banco de dados.

No método getInstance, é verificado se a instância enviada do *Model* para essa classe está nula. Se não estiver, uma conexão com o banco de dados é aberta. Se a instância estiver nula, é criada uma conexão com o banco de dados.

No método "getIdsAlunosTurma", um *array* de inteiros é declarado para ser preenchido e retornado para a classe *Model*. Um inteiro chamado de "id" é declarado logo após o *ArrayList*, a consulta é realizada chamando a função "rawQuery", que pertence a biblioteca denominada "SQLiteDataBase", e passa a consulta SQL dentro deste método.

```
package br.com.cefetrj.persistencia;

import java.util.ArrayList;

public class TurmaPersistencia {

    private static SQLiteDatabase BancoDados;
    Cursor cursor;

    private Context context;
    private static TurmaPersistencia instance;

    public static TurmaPersistencia getInstance(Context context) {
        if (instance == null) {
            synchronized (TurmaPersistencia.class) {

                if (instance == null) {

                    instance = new TurmaPersistencia(context);

                    BancoDados = context.openOrCreateDatabase(
                            DatabaseCreate.TUTORIAL_DB, Context.MODE_PRIVATE,
                            null);
                }
            }
        }
        return instance;
    }

    private TurmaPersistencia(Context context) {

        this.context = context;

    }

    public ArrayList<Integer> getIdsAlunosTurma(int idDisciplina){
        ArrayList<Integer> listaAlunos = new ArrayList<Integer>();
        int id = 0;
        try {
            cursor = BancoDados.rawQuery("SELECT t.idAluno FROM turma t where t.idDisciplina = "+ idDisciplina +"", null);

            if  (cursor.moveToFirst()) {
                do {
                    id = cursor.getInt(cursor.getColumnIndex("idAluno"));
                    listaAlunos.add(id);
                }while (cursor.moveToNext());
            }

        } catch (Exception erro) {

        }
        return listaAlunos;
    }
```

Figura 13. Estrutura de uma classe de persistência

O resultado desta consulta é atribuído a um cursor, que é posteriormente utilizado para recuperar o valor de uma coluna. Dentro do *while*, os identificadores que são encontrados vão sendo adicionados à lista de inteiros e, após preencher tudo, a lista é retornada para o *Model*, que retorna a lista para a *Activity*, continuando o código para preencher a lista de alunos.

Um método que foi muito utilizado no decorrer do desenvolvimento desta aplicação Android foi o de envio de e-mail nativo do Android, ele chama o aplicativo de e-mail do Android, já preenchendo todos os campos devidamente, deixando para o usuário apenas a opção de enviar ou não o e-mail. Este método é chamado após o professor lançar uma nota para um determinado aluno. As informações do aluno são gravadas na variável e o e-mail do aluno é preenchido com o body, que é o corpo do e-mail, e também o assunto.

Se o usuário não estiver logado com sua conta no aplicativo de e-mail do Android, ele não envia o e-mail e informa isso ao usuário.

E. Telas da Aplicação

A seguir são apresentadas algumas das principais telas da aplicação. A Figura14 demonstra a tela inicial do usuário com o perfil de aluno.

Após realizar o login, se o perfil do usuário for de aluno, esta tela será apresentada a ele. Nela, ele poderá clicar em “Minhas Disciplinas” ou “Pesquisar”.

A Figura 15 demonstra a tela inicial do usuário com o perfil de professor.

Após realizar o login, se o perfil do usuário for de professor, esta tela será apresentada a ele. Nela, ele poderá clicar em “Minhas Disciplinas”, “Pesquisar” ou “Criar Disciplina”.

A Figura 16 demonstra a tela de uma disciplina do usuário com o perfil de aluno.

Após selecionar a disciplina, o sistema conduz o aluno para esta tela, na qual ele poderá visualizar os comunicados desta disciplina, ver suas notas e cancelar sua inscrição.

A Figura 17 demonstra a tela de uma disciplina do usuário com o perfil de professor.

Após selecionar a disciplina, o sistema conduz o professor para esta tela, na qual ele poderá realizar algumas funções.

Figura 14. Tela inicial do aluno

Figura 15. Tela inicial do professor

Figura 16. Tela de disciplina selecionada do aluno

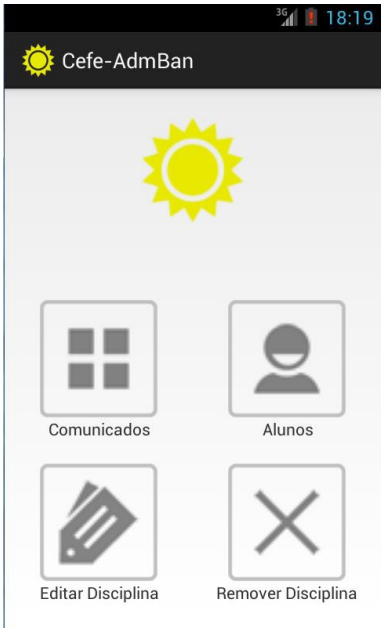

Figura 17. Tela de disciplina selecionada do professor

V. TRABALHOS RELACIONADOS

Em Oliveira e Medina [11] os autores realizam uma análise totalmente teórica sobre o tema proposto em nosso aplicativo. Eles apresentam as limitações concretas para a criação de aplicativos de aprendizagem. Na realidade apresentada, havia o questionamento sobre as futuras condições para se produzir um software capaz, e hoje, é sabido que é bem possível.

Já em Tarouco et al. [12] foi feita uma análise da infraestrutura do ensino à distância através de e-Learning, onde foi destacada a importância da Internet como principal mecanismo para a evolução do m-learning, provendo ubiquidade no processo de aprendizagem e facilitando no desenvolvimento de estratégias para educação por meio de objetos de aprendizagem.

É destacado em Quinta e Lucena [13] o uso da tecnologia de dispositivos móveis e TV digital para suplemento das tecnologias de ensino existentes através de suporte a mídias e capacidade de processamento, padronização e construção do material instrucional. Este último é feito através de Odin, uma ferramenta de conversão automática de arquivos existentes em bases de dados em objetos de aprendizagem para diferentes dispositivos. O software oferece acesso a arquivos personalizados de modo transparente tanto para o dispositivo de estudantes quanto para o dispositivo do docente.

No trabalho de Quinta e Lucena [14] os autores enfatizam que, através da popularização de tecnologias móveis e a expansão da Internet, foi viabilizado o acesso ao conteúdo educacional de maneira ubíqua e a qualquer instante, desde que o conteúdo possa estar empregado em todos os dispositivos. Os autores também apresentam os problemas associados a este uso. O software Odin aparece como solução na adaptação automática de áudio, vídeo, imagens e texto para o conteúdo a diferentes dispositivos. O software permite adicionar extensão aos servidores de aplicações para ensino á distância.

Foi realizada a análise e avaliação da aprendizagem EAD em Ribeiro et al. [15], através do AVAM MLE-Moodle utilizado no Curso de Capacitação Linguagem de Programação HTML da Universidade Federal de Santa Maria. A pesquisa mostra a satisfação dos alunos com o ambiente considerado incentivador e de grande utilidade. Nota-se maior abertura de atividades que possam ser realizadas devido a utilização dos dispositivos móveis, cumprindo assim o objetivo da usabilidade dentro do contexto do EAD. A partir disso, sente-se a necessidade de um recurso de armazenamento de vídeo que complemente atividades educacionais e auxilie professores e alunos disponibilizando o material, mantendo a característica principal da mobilidade de escolher o horário e local de estudo.

Foi realizada uma análise em Mühlbeier et al. [16] e os autores comentam sobre a evolução da tecnologia e, através da plataforma Android, analisam o aplicativo Toon-Doo, que é utilizado para a criação de histórias em quadrinhos. Foram realizados testes com usuários e, após os testes, foram realizadas perguntas de satisfação para a melhoria do aplicativo.

Em Orlandi e Isonati [17] é descrito o estudo e desenvolvimento de um sistema de autoria e distribuição de conteúdo educacional interativo para dispositivos móveis. A ferramenta permite que o aluno responda uma lista de exercícios de múltipla escolha e o sistema pode avaliar automaticamente as respostas gerando dados que podem ser usados para acompanhar o desenvolvimento dos usuários. O principal objetivo da ferramenta é permitir a verificação de dificuldades na aprendizagem de forma rápida e fácil com o auxílio do sistema computacional.

VI. CONCLUSÃO

Conclui-se que a aplicação, mesmo com menor grau de complexidade, pode ter uma boa contribuição principalmente na área educacional atuando na síntese e absorção do conhecimento necessário, ainda como importante mecanismo da EAD, podendo transformar-se também em ferramenta complementar do ensino nas salas de aula.

O aplicativo deve contribuir na interação aluno-professor de todo o modo, seja presencial, não presencial, síncrono ou assíncrono. Sua utilização em dispositivos móveis fornece ferramentas para aumentar de maneira significativa a aprendizagem nas disciplinas solicitadas. Além disso, professores podem desfrutar de maior eficiência na comunicação decorrente deste canal adicional com seus alunos. Desta forma, a aplicação agrega aos usuários a facilidade de se manter conectado à sala de aula virtual sem restrições de horário e local – característica do conceito de ubiquidade potencializado pela evolução dos aparelhos e tecnologias de acesso a Internet.

REFERÊNCIAS

[1] A. Caldeira, “Avaliação da aprendizagem em meios digitais: novos contextos”, Avaliação da aprendizagem em meios digitais: novos contextos, p. 12, abr. 2012. Disponível em:<https://www2.ufmg.br/ead/content/download/9706/70559/file/CALDEIRA.pdf>. Acesso em: 15 dez. 2012.

[2] T. Serrão, L. Braz, S. Pinto, e G. Clunie, “Construção Automática de Redes Sociais Online no Ambiente Moodle”, UNISINOS - Universidade do Vale do Rio dos Sinos, Rio Grande do Sul, Brasil, 2011. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/1647/1412>. Acesso em: 09 fev. 2014.

[3] L. Meirelles e L. Tarouco, “Framework para Aprendizagem com Mobilidade”, UFRGS - Universidade Federal do Rio Grande do Sul, Rio Grande do Sul, Brasil, 2005. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/446/432>. Acesso em: 09 fev. 2014.

[4] F. Franciscato e R. Medina, "M-Learning e Android: um novo paradigma?", UFSM - Universidade Federal de Santa Maria, Rio Grande do Sul, Brasil, 2008. Disponível em: <http://seer.ufrgs.br/renote/article/view/14671/8580>. Acesso em: 09 fev. 2014.

[5] L. Silva, F. Neto, e L. Júnior, "MobiLE: Um ambiente Multiagente de Aprendizagem Móvel para Apoiar a Recomendação Sensível ao Contexto de Objetos de Aprendizagem", Universidade Federal Rural do Semi-Árido, Mossoró, Rio Grande do Norte, 2011. Disponível em: <http://ceie-sbc.educacao.ws/pub/index.php/sbie/article/view/1593/1358>. Acesso em: 09 fev. 2014.

[6] J. Rodrigues, "Uso de m-learning no Ensino Superior", Dissertação, Universidade de Aveiro, Aveiro, Portugal, 2007. Disponível em: <http://ria.ua.pt/bitstream/10773/1533/1/2008001205.pdf>. Acesso em: 09 fev. 2014.

[7] K. Fernandes, G. Trindade, B. Souza, A. Gomes, e M. Lucena, "Question Mobile: Ampliando Estratégias de Avaliação da Aprendizagem por Meio de Dispositivos Móveis", UFRN - Universidade Federal do Rio Grande do Norte, Natal, RN - Brasil, 2012. Disponível em: <http://ceie-sbc.tempsite.ws/pub/index.php/wcbie/article/view/1940/1700>. Acesso em: 09 fev. 2014.

[8] Figura 5 Traduzida de: < http://www.redrails.com.br/2011 /04/18/google-android-como-um-ambiente-de-desenvolvimento-de-aplicacoes-para-sistemas-de-decodificacao-de-dtv-tv-digital/>Acesso em: 09 fev. 2014.

[9] Figura 6 Disponível em: <http://en.wikipedia.org/wiki/ File:Overview_of_a_three-tier_application_vectorVersion.svg> Acesso em: 09 fev. 2014.

[10] Figura 8 Disponível em: <http://mobilidade.fm/tag/mvc-para-android/> Acesso em: 09 fev. 2014.

[11] L. Oliveira e R. Medina, "Desenvolvimento de objetos de aprendizagem para dispositivos móveis: uma nova abordagem que contribui para a educação.", UFSM - Universidade Federal de Santa Maria, Rio Grande do Sul, Brasil. Disponível em: <http://lumenagencia.com.br/dcr/arquivos/desenvolvimento_de_objetos_de_aprendizagem_para_dispositivos_m%F3veis.pdf>. Acesso em: 09 fev. 2014.

[12] L. Tarouco, M.-C. Fabre, M. Konrath, e A. Grando, "Objetos de Aprendizagem para M-Learning", Sucesu. Disponível em: <http://objectosaprendizagem.no.sapo.pt/pdf/objetosdeaprendizagem_sucesu.pdf>. Acesso em: 09 fev. 2014.

[13] M. Quinta e F. Lucena, "Odin - Viabilizando e-learning em múltiplos dispositivos", Universidade Federal de Goiás, Goiânia, GO, 2007. Disponível em: <http://www.br-ie.org/WIE2010/pdf/sp01_07.pdf>. Acesso em: 09 fev. 2014.

[14] M. Quinta e F. Lucena, "Problemas e soluções em u-learning e a adaptacão de conteúdo de objetos de aprendizagem para diferentes dispositivos", Universidade Federal de Goiás, Goiânia, GO, 2007. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/1501/1266>. Acesso em: 09 fev. 2014.

[15] P. Ribeiro, F. Franciscato, P. Mozzaquatro, e R. Medina, "Validação de um Ambiente de Aprendizagem Móvel em Curso a Distância", UFSM - Universidade Federal de Santa Maria, Rio Grande do Sul, Brasil. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/1153/1056>. Acesso em: 09 fev. 2014.

[16] A. Mühlbeier, P. Mozzaquatro, R. Medina, R. Moreira, e R. Antoniazzi, "MOBILE HQ: O USO DE SOFTWARES EDUCATIVOS NA MODALIDADE M-LEARNING", UFSM - Universidade Federal de Santa Maria, Rio Grande do Sul, Brasil, 2012. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/1742/1503>. Acesso em: 09 fev. 2014.

[17] B. Orlandi e S. Isonati, "Uma Ferramenta para Distribuição de Conteúdo Educacional Interativo em Dispositivos Móveis", USP - Universidade de São Paulo, São Paulo - Brasil, 2012. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/1792/1553>. Acesso em: 09 fev. 2014.

AUTORES

Minto, R.S. graduado no Curso Superior de Tecnologia em Sistemas para Internet,CEFET-RJ, Rio de Janeiro (e-mail: ruanminto@gmail.com).

dos Santos, V.F.M. graduado no Curso Superior de Tecnologia em Sistemas para Internet,CEFET-RJ, Rio de Janeiro (e-mail: vitorfla1@hotmail.com).

Paschoal Junior, F. professor Mestre no CEFET/RJ, Rio de Janeiro (e-mail: fabiopjr@yahoo.com.br).

Submitted, February, 10, 2014. Published as resubmitted by the authors on April, 05, 2014.