



CENTRO UNIVERSITÁRIO UNIVATES
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

APLICATIVO ANDROID PARA O AMBIENTE UNIVATES VIRTUAL

Marcel Dall'Oglio

Lajeado, julho de 2013

Marcel Dall'Oglio

APLICATIVO ANDROID PARA O AMBIENTE UNIVATES VIRTUAL

Monografia apresentada ao Centro de Ciências Exatas e Tecnológicas do Centro Universitário UNIVATES, como parte da exigência para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Prof. Ms. Alexandre Stürmer Wolf

Lajeado, julho de 2013

Marcel Dall'Oglio

APLICATIVO ANDROID PARA O AMBIENTE UNIVATES VIRTUAL

A Banca examinadora abaixo aprova a Monografia apresentada ao Centro de Ciências Exatas e Tecnológicas, do Centro Universitário UNIVATES, como parte da exigência para a obtenção do grau de Bacharel em Engenharia da Computação:

Prof. Ms. Alexandre S. Wolf – orientador
Centro Universitário UNIVATES

Prof. Maria Claudete Schorr Wildner
Centro Universitário UNIVATES

Prof. Dr. Pedro Antônio M. de Campos Velho
Centro Universitário UNIVATES

Lajeado, 2 de julho de 2013

RESUMO

Com a expansão da internet e das transformações dos modelos de ensino, surgiram softwares para apoiar a aprendizagem de forma virtual. Desse contexto surgiu o Moodle, um sistema de gestão de aprendizagem de forma virtual e com código fonte aberto, possibilitando assim a sua adaptação e utilização em diversas Universidades e Instituições de ensino. Atualmente é o software utilizado na Univates como base para o seu ambiente de aprendizagem virtual intitulado de Univates Virtual. Este projeto propõe o desenvolvimento de um aplicativo para a plataforma Android, a fim de facilitar o acesso aos seus recursos disponibilizados no ambiente virtual (Univates Virtual) por parte da comunidade acadêmica da Univates, através dos dispositivos móveis. A partir do desenvolvimento realizado, foi possível observar que este projeto pode colaborar de forma expressiva no caso específico do ambiente Univates Virtual, mostrando que a aplicação funciona de forma eficaz naquilo em que foi projetada, dessa forma possibilitando a comunidade acadêmica manter-se atualizada sobre os recursos disponibilizados através do ambiente de aprendizado virtual.

Palavras-chave: Moodle. Android. Dispositivos Móveis. Web Service.

ABSTRACT

With the expansion of the Internet and the transformation of teaching models, software emerged to support learning in virtual form. This context emerged Moodle, a management system in a virtual learning and open source, thus enabling its adaptation and use in various universities and educational institutions. Currently the software is used in Univates as the basis for your virtual learning environment called Univates Virtual. This project proposes the development of an application for the Android platform in order to facilitate access to resources available in the virtual environment (Univates Virtual) by the Univates academic community, via mobile devices. From development conducted, it was observed that this project can contribute significantly in the specific case of Univates Virtual environment, showing that the application works effectively in what was projected, thereby allowing the academic community to keep updated about the resources available through the virtual learning environment.

Keywords: Moodle. Android. Mobile Devices. Web Service.

LISTA DE FIGURAS

Figura 1 - Arquitetura da aplicação	16
Figura 2 – Arquitetura da plataforma Android.....	31
Figura 3 - Representação do fluxo de utilização da aplicação.	38
Figura 4 - Diagrama da base de dados	39
Figura 5 - Tela de acesso da aplicação.....	42
Figura 6 – Tela de autenticação do usuário	43
Figura 7 - Tela de cursos.....	44
Figura 8 - Tela de tarefas do curso	45
Figura 9 - Tela de tarefa com descrição completa	46
Figura 10 - Tela de recursos do curso.....	47
Figura 11 - Tela de correio eletrônico do curso	48
Figura 12 - Tela de e-mail	49
Figura 13 - Tela de cursos com opções visíveis.....	50
Figura 14 – Tela de atualização dos dados.....	51
Figura 15 - Tela de configurações.....	52
Figura 16 - Tela para sair do aplicativo com opção de apagar dados	53
Figura 17 – Notificações da aplicação.....	55

LISTA DE CÓDIGOS

Listagem 1 – Função que inicia o serviço em background	54
Listagem 2 – Permissões adicionadas no arquivo AndroidManifest.xml	56

LISTA DE TABELAS

Tabela 1 – Distribuição das versões da plataforma Android	40
---	----

LISTA DE ABREVIATURAS E SIGLAS

ADT -	Android Developer Tool
API -	Application Programming Interface
DIME -	Direct Internet Message Encapsulation
FTP -	File Transfer Protocol
GNU GPL -	GNU General Public License
HTML -	Hyper Text Markup Language
HTTP -	Hyper Text Transfer Protocol
HTTPS -	Hyper Text Transfer Protocol Secure
HTTPR -	Hyper Text Transfer Protocol Reliable
J2ME -	Java Plataform Micro Edition
JMS -	Java Message Service
JVM -	Java Virtual Machine
LDAP -	Lightweight Directory Access Protocol
LMS -	Learning Management System
MLE -	M-Learning Extension
MOMO -	Mobile Moodle
MOODLE -	Modular Object Oriented Dynamic Learning Environment
MVC -	Model View Control
MVD -	Máquina Virtual Dalvik
OHA -	Open Handset Alliance
PHP -	PHP Hypertext Preprocessor
RPC -	Remote Procedure Call
SDK -	Software Development Kit

SGDB -	Sistema de Gerenciamento de Banco de Dados
SMTP -	Simple Mail Transfer Protocol
SOAP -	Single Object Access Protocol
SSL -	Secure Sockets Layer
TLS -	Transport Layer Security
TIC -	Tecnologia da Informação e Comunicação
UDDI -	Universal Description, Discovery and Integration
URL -	Uniform Resource Locator
XML -	eXtensible Markup Language
WSDL -	Web Service Definition Language
WUST -	WSDL, UDDI e SOAP Technologies

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos	14
1.2 Estrutura do trabalho	16
 2 TRABALHOS RELACIONADOS.....	18
2.1 Moodle Mobile.....	18
2.2 Moodbile.....	19
2.3 Outros projetos de LMS para dispositivos móveis	21
 3 TECNOLOGIAS ENVOLVIDAS.....	24
3.1 Moodle.....	24
3.2 Apache.....	26
3.3 Web Service	27
3.3.1 XML	28
3.3.2 WSDL	28
3.3.3 UDDI	29
3.3.4 SOAP	29
3.4 Android.....	30
3.4.1 Arquitetura.....	31
3.4.2 Kernel	31
3.4.3 Bibliotecas.....	32
3.4.4 Android Runtime	32
3.4.5 Application Framework	33
 4 APLICAÇÃO ANDROID	35
4.1 Análise de requisitos.....	35
4.1.1 Requisitos funcionais.....	35
4.1.2 Requisitos não funcionais.....	36
4.2 Desenvolvimento	36
4.2.1 Visão geral.....	37
4.2.2 Ferramentas.....	40
4.2.3 Funcionalidades.....	41

4.2.4 Serviço em background	53
4.2.5 Permissões	55
5 WEB SERVICE	57
5.1 Visão geral	57
5.2 Ferramentas	58
5.3 Funcionalidades	58
5.3.1 Autenticação.....	59
5.3.2 Consulta de usuário.....	59
5.3.3 Consulta de cursos.....	60
5.3.4 Consultas de tarefas.....	60
5.3.5 Consulta de recursos	60
5.3.6 Consulta de e-mails	61
6 CONCLUSÃO	62
6.1 Trabalhos futuros	63
REFERÊNCIAS.....	64

1 INTRODUÇÃO

A educação tem evoluído nos últimos anos com a utilização da tecnologia de informação. Com o advento da Internet, uma grande quantidade de material educacional que até então não era possível ter acesso, agora pode ser acessada rapidamente. Desta forma, é possível superar as barreiras geográficas ou de tempo sem sacrificar as vantagens do sistema de formação tradicional, permitindo complementar ou mesmo satisfazer plenamente o aprendizado dos alunos.

Uma das ferramentas mais utilizadas nos sistemas de *e-learning* (aprendizagem eletrônica) é o Sistema de Gestão da Aprendizagem, que em inglês é conhecido como *Learning Management System* (LMS). Um software instalado em um servidor web para administrar e oferecer atividades de formação não presenciais, permitindo administrar o sistema, os usuários e os seus cursos.

Dentro do LMS encontra-se o Moodle (Modular Object Oriented Dynamic Learning Environment), criado em 1999 por Martin Dougiamas. O Moodle é um sistema de gerenciamento de cursos ou ambiente virtual de aprendizagem de distribuição gratuita que ajuda os educadores a criar comunidades de aprendizagem on-line, atualmente conta com mais de 80 mil *websites* registrados e mais de 69 milhões de usuários, sendo o Brasil o terceiro país com mais sites registrados, como pode ser visualizado em <http://moodle.org/stats>.

É facilmente perceptível o crescimento constante que tem tido e embora a evidência sugira que a contagem começa a estabilizar o número de sites registrados, ainda é esperado um futuro promissor, especialmente ao nos

adaptarmos ao novo paradigma da educação *m-learning* (aprendizagem móvel). Segundo Junior, Coutinho e Alexandre (2006), num futuro bem próximo esta tecnologia do *m-learning* será fortemente empregada dada a sua facilidade de acesso a toda hora e em qualquer espaço geográfico.

O fato de ser um software livre, de código aberto, podendo ser instalado gratuitamente, aliado ao por menor de permitir que o utilizador altere e adapte o ambiente, de acordo com as suas próprias necessidades, fez com que a plataforma Moodle rapidamente estivesse presente no dia a dia de diversas instituições, sendo amplamente utilizada em vários países e apresentando taxas de crescimento exponencial de utilização (SANTOS, 2012).

O Moodle em especial, é o ambiente de aprendizagem virtual utilizado na Univates, conhecido como Univates Virtual, e o ambiente que será utilizado para este projeto. Outra tecnologia a ser utilizada neste projeto, que também colabora com a evolução do uso da tecnologia de informação na educação, são os dispositivos móveis como os *smartphones* e os *tablets*.

O Android é uma plataforma de código aberto para *smartphones* e *tablets*, baseada no Sistema Operacional Linux. A maior parte do seu código foi lançada pelo Google em 2007, sob licença livre. É a plataforma na qual será projetada a aplicação, devido ao fato de ser *open source* (código livre) e estar bem popularizada entre os dispositivos móveis (GOOGLE DEVELOPERS, 2013).

Conforme Pereira e Da Silva (2009), a plataforma Android foi construída com a intenção de permitir aos desenvolvedores criarem aplicações móveis que possam tirar total proveito do que um aparelho portátil possa oferecer.

1.1 Objetivos

O objetivo principal do projeto é desenvolver um software (aplicativo) nativo para sistemas móveis Android, tecnologia que permitirá aos usuários do ambiente de aprendizagem Univates virtual acessar o conteúdo e as atividades a partir de dispositivos móveis.

Além da adaptação do conteúdo para ficar de acordo com o tipo de tela dos dispositivos móveis, o usuário do aplicativo não necessita ter uma conexão permanente com o servidor e também não exige o uso de um navegador no seu dispositivo móvel.

Para uma aplicação desta natureza, é necessário o desenvolvimento de um serviço para disponibilizar a capacidade de fornecer sua funcionalidade por meio da web, um conjunto de normas e protocolos para transferência de informações entre aplicações. Dessa forma, o projeto final, é desenvolver um cliente que utilize o *Web Service* (serviço web) para acessar determinados conteúdos e atividades do Univates Virtual, permitindo aos usuários a capacidade de desfrutar destas funcionalidades através de dispositivos móveis.

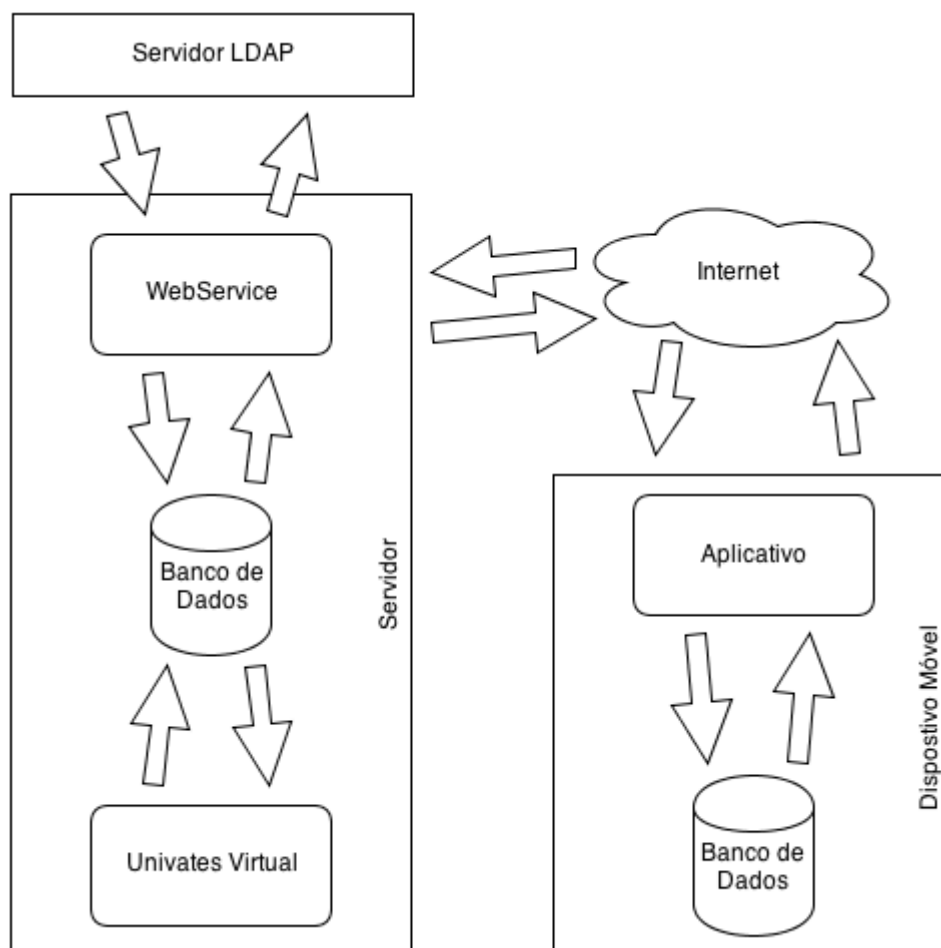
Assim sendo, o objetivo principal do projeto é fornecer aos usuários do ambiente Univates Virtual a capacidade de conexão ao sistema a partir de dispositivos móveis, e tornar a consulta de informações mais adequadas quando necessário, como quando em movimento ou sem possuir um computador a disposição, além de fornecer uma interface que é adaptada para as telas reduzidas dos dispositivos móveis.

Devido à mobilidade, surge o problema de conectividade, ou seja, sem períodos de conexão com a Internet. Assim, outro importante objetivo é permitir o usuário continuar utilizando a aplicação mesmo quando se perde a conexão com a Internet, utilizando os dados salvos na aplicação anteriormente, e assim que a conexão com a Internet estiver novamente disponível, verificar por novas atualizações das informações.

Na Figura 1 é apresentada a arquitetura da aplicação. É possível observar que o *Web Service* funcionará com o mesmo banco de dados utilizado pelo Univates Virtual e irá fornecer os dados à aplicação do dispositivo móvel através da Internet após realizar a autenticação do usuário no servidor LDAP (Lightweight Directory Access Protocol). O servidor LDAP tem a função de verificar as credenciais do cliente e verificar as informações solicitadas que estão armazenadas neste servidor. Ao receber os dados do *Web Service*, a aplicação utilizará o banco de dados do

dispositivo móvel para armazenar as informações, possibilitando assim um melhor desempenho em acessar as informações, além de permitir as consultas em momentos sem conectividade.

Figura 1 - Arquitetura da aplicação



Fonte: Elaborado pelo Autor.

1.2 Estrutura do trabalho

O presente trabalho está estruturado da seguinte forma: Capítulo 1 é feita uma introdução sobre os ambientes virtuais de aprendizagem, dispositivos móveis e os principais objetivos da aplicação. No Capítulo 2 são apresentados os trabalhos relacionados ao tema e as principais diferenças entre a aplicação a ser desenvolvida e os projetos já existentes. No Capítulo 3 são descritos as tecnologias que foram utilizadas durante o desenvolvimento da aplicação. No Capítulo 4 são descritos as informações relacionadas ao desenvolvimento da aplicação, como a definição dos

requisitos, da linguagem de programação, da implementação do banco de dados e das funcionalidades. No capítulo 5 são descritos as informações relacionadas ao desenvolvimento do *Web Service*, como a linguagem de programação, ferramentas utilizadas e a implementação das suas funcionalidades. Finalmente no Capítulo 6 é apresentada uma conclusão sobre o trabalho e os trabalhos futuros a serem desenvolvidos.

2 TRABALHOS RELACIONADOS

Atualmente já existem diversos trabalhos relacionados à utilização do Moodle através de dispositivos móveis. A seguir serão apresentados os projetos Moodle Mobile, Moodbile e outros projetos de dispositivos móveis para sistemas de gestão de aprendizagem, destacando as principais diferenças destas aplicações em relação a este projeto.

2.1 Moodle Mobile

O Moodle Mobile é um aplicativo oficial do Moodle para as plataformas Android e iOS. Ele é baseado em um cliente HTML5, que é conectado a um servidor Moodle versão 2.4 ou superior, desde que possuam acesso habilitado ao *Web Service* para dispositivos móveis.

Entre suas principais características estão:

- Design ágil para o telefone e *tablets*;
- Carregar uma imagem em sua área de arquivo privado;
- Gravar um arquivo de áudio e enviá-lo para a área de arquivos privados;
- Enviar uma mensagem privada a um participante do curso;

- Dar uma nota pessoal sobre um participante do curso;
- Adicionar um participante do curso para o sua lista de contatos do telefone;
- Realizar uma chamada para um participante do curso ao tocar o seu número de telefone;
- Localizar o endereço de um participante no mapa do Google;
- Fazer o download e visualizar alguns recursos;
- Acesso rápido aos conteúdos dos cursos.

O Moodle Mobile também está substituindo o antigo aplicativo para iOS oficial do Moodle, o MyMoodle, e está disponível para download na loja de aplicativos do Google e da Apple. Existem muitas diferenças das características do aplicativo Moodle Mobile em relação a este projeto, e a necessidade de utilização da versão 2.4 ou maior do ambiente de aprendizagem Moodle, inviabiliza a sua utilização no ambiente Univates Virtual, pois o Moodle utilizado utiliza uma versão inferior que não é compatível com o mesmo.

2.2 Moodbile

O Moodbile é um projeto de software *open source*, cujo objetivo principal é permitir que aplicações móveis de aprendizagem conseguissem trabalhar com o LMS Moodle.

Este projeto foi iniciado pelo grupo de pesquisa SUSHITOS da Universidade Politécnica da Catalunha (UPC), cujos membros estão envolvidos com a comunidade Moodle.org desde 2004, participando do desenvolvimento do Moodle 2.0, *Wiki* e o subsistema de *Web Services*. Entre os componentes que fazem parte do projeto Moodbile, encontramos três partes distintas:

- a) Especificação dos *Web Services* do Moodle: O projeto Moodbile é responsável pelo desenvolvimento e manutenção da API (Application Programming Interface) de *Web Services* do Moodle 2.0 adiante para permitir que aplicativos móveis possam acessar seus serviços. Por isso, tem sido necessário especificar e desenvolver uma longa lista de serviços que permitem aos clientes ter acesso a um grande número de funcionalidades do sistema Moodle;
- b) Servidor Moodbile: *Plugin* que implementa as especificações do Moodbile e foi testado para trabalhar com o Moodle 2.0 e 2.1, entre outros;
- c) Aplicativo Cliente Moodbile: Fazem parte do projeto três aplicativos clientes para o Moodbile. Um aplicativo para o sistema operacional móvel da Apple (iOS), um aplicativo para a plataforma Android e também um aplicativo em HTML (HyperText Markup Language) 5, fazendo uso dos *Web Services* fornecidos pelo Moodbile. Os três aplicativos possuem características e funcionalidades semelhantes, mas respeitando os padrões de design e usabilidade, por isso os designs dos aplicativos divergem ligeiramente.

O aplicativo para plataforma Android do projeto Moodbile é semelhante em diversas características a este projeto, mas no caso específico do ambiente Univates Virtual, que é baseado no sistema do Moodle versão 1.9.2, não é possível a sua utilização, por isso a necessidade do desenvolvimento de um *Web Service* e uma aplicação Android específica para o ambiente Univates Virtual. A seguir são apresentadas as características do aplicativo para o sistema operacional Android do projeto Moodbile:

- Modo *off-line*: A aplicação é capaz de funcionar sem necessidade de haver uma conexão permanente com a Internet;
- Cursos: Listar os cursos ativos do usuário conectado;

- Fóruns: Listar os fóruns dos cursos ativos, permitir o usuário criar novos tópicos e responder mensagens;
- Tarefas: Listar as tarefas dos cursos ativos, com as notas e os comentários dos professores;
- Usuários: Listar os participantes dos cursos ativos e possibilidade de visualizar o perfil de cada participante.

As características da aplicação do projeto Moodbile possuem certas semelhanças com as características deste projeto, menos na questão do fórum que não está prevista nessa aplicação, além de não contemplar outras funcionalidades que serão apresentadas posteriormente no Capítulo 4, página 35.

2.3 Outros projetos de LMS para dispositivos móveis

Neste ponto, vamos discutir outros tipos de aplicações móveis que são responsáveis por trazer o mundo do *e-learning* para os seus utilizadores. Entre estas aplicações podemos encontrar dois grupos:

- a) As aplicações específicas para dispositivos móveis: Tais aplicações são caracterizadas usando os recursos do hardware do dispositivo para o qual são desenvolvidas, permitindo que os usuários interajam com recursos do telefone, como a câmera ou seus arquivos. Estes tipos de aplicações, por serem nativas, tem um acesso mais rápido e mais leve do que o obtido quando se navega por páginas da web. No entanto, a principal desvantagem destes é o de ter um maior custo de desenvolvimento quando se pretende desenvolver várias versões do aplicativo para diferentes dispositivos móveis;
- b) Sites otimizados para dispositivos móveis: Hoje, a grande maioria dos dispositivos móveis é equipada com acesso à Internet. Neste caso, é encontrado o conteúdo de sites de *e-learning*, buscando trazer o seu conteúdo a qualquer tipo de dispositivo móvel, independentemente do

modelo em si, graças à conectividade oferecida por este tipo de dispositivo. Com isso, encontra-se um menor custo de desenvolvimento, pois permite o acesso a vários tipos de dispositivos, evitando o desenvolvimento de aplicativos diferentes. No entanto, este modelo possui inconveniências, uma vez que o fato de que este acesso é através de *Web Services*, torna o acesso mais lento e, provavelmente, o aumento do consumo de dados.

Depois de analisar esses dois casos, é possível determinar quais aspectos como usabilidade (para adequar o conteúdo da aplicação para a tela do dispositivo) e da persistência de dados (para evitar gastos excessivos de dados nos acessos à Internet, armazenando os dados na memória do dispositivo), são pontos importantes a ter em mente no desenvolvimento de um aplicativo de *e-learning* para dispositivos móveis, que na maioria das vezes busca as suas informações através da web.

Dentro desses grupos de aplicações podemos encontrar alguns projetos, tais como:

- Hinkelman, que em 2005 apresentou um módulo para o Moodle 1.6 que permitia a realização de testes a partir de dispositivos móveis japoneses, a partir de uma versão simplificada do Moodle desenvolvida especificamente para dispositivos móveis;
- A universidade Open University, que há vários anos vem trabalhando na integração do Moodle com dispositivos móveis, depois de vários estudos apresentados, os levou a apresentar em 2009 uma aplicação móvel de *e-learning* para dispositivos móveis que fazem uso do Moodle;
- MOMO (Mobile Moodle) e MLE (M-Learning Extension) são dois outros projetos que permitem o acesso às funcionalidades do Moodle 1.9 através de dispositivos móveis. Neste caso o projeto MLE acabou absorvendo o projeto MOMO, é um aplicativo J2ME (Java Platform, Micro Edition) que juntamente com uma versão web permite o acesso a

cursos do Moodle, suas aulas, tarefas, recursos, fóruns e *wikis*, entre outros;

- Moviltest oferece uma aplicação J2ME sobre o Moodle 1.9 que permite downloads de testes, executar no celular e enviá-los de volta para o servidor Moodle;
- Mpage é outro projeto em cima do Moodle 1.9 que consiste em um aplicativo de dispositivo iPhone que permite o acesso a cursos, recursos, bate-papos, fóruns e *quiz*;
- MyMoodle que é o antigo aplicativo oficial do Moodle para iPhone, projetado para a versão 2.1 ou superior do Moodle, permite acessar lista de participantes, enviar e baixar arquivos, trabalhar *off-line*, acessar a versão web do Moodle, entre outras funcionalidades, atualmente foi substituído pelo projeto Moodle Mobile.

3 TECNOLOGIAS ENVOLVIDAS

Neste capítulo são apresentadas e descritas as tecnologias envolvidas no desenvolvimento do projeto. Entre as tecnologias apresentadas estão: O Moodle que é o ambiente virtual de aprendizagem, o qual foi utilizado para o desenvolvimento do ambiente Univates Virtual. O servidor web Apache. O *Web Service*, que é o serviço web que irá realizar a comunicação de dados entre o Univates Virtual e a aplicação do dispositivo móvel, ainda será apresentada. A plataforma de dispositivos móveis Android.

3.1 Moodle

O *e-learning* ou aprendizagem eletrônica é o caminho de acesso ao conhecimento através de Tecnologia da Informação e Comunicação (TIC), que abrange todas as formas de mídia eletrônica para aprender e ensinar. Com este sistema é possível ter uma educação à distância totalmente virtualizada através do uso da Internet e de ferramentas como *e-mail*, *sites*, fóruns, etc.

O Moodle se encontra dentro deste grupo, e ao longo dos anos, encontramos cada vez mais presente nas escolas e faculdades ou instituições de ensino que fazem uso das suas ferramentas, como forma de melhorar a qualidade de ensino.

O Moodle é uma ferramenta didática largamente utilizada e que pode contribuir muito para o desenvolvimento acadêmico dos seus usuários. Esta ferramenta proporciona também, a facilidade do acesso à informação por ser um

software livre e pela grande interação que proporciona entre seus usuários (ALENCAR et al., 2012).

O Moodle, portanto é um sistema de gerenciamento de cursos ou ambiente virtual de aprendizagem de distribuição gratuita, que ajuda os educadores a criar comunidades de aprendizagem *on-line*, conhecidos como LMS (Learning Management System). O código é escrito em PHP (PHP Hypertext Preprocessor) licenciado sob a GNU GPL (GNU General Public License) (MOODLE, 2013). Com isso, determina-se que o Moodle é protegido, mas qualquer pessoa tem permissão para copiar, usar ou modificar seu código, desde que a pessoa concorde em compartilhar seu código fonte com os outros e não modificar ou eliminar a licença original e os direitos autorais, e aplicar a mesma licença derivada a partir deste trabalho, dessa forma torna-se um serviço flexível e personalizável.

Para a instalação do Moodle, é necessária uma plataforma que suporte PHP e a disponibilidade de um banco de dados, suportando mais de um gerenciador de banco de dados. Além disso, é possível adicionar novos módulos e pacotes de idiomas.

As principais características do Moodle são:

- Administração do site: Utilização de temas, a linguagem, e que lhe permite adicionar módulos;
- Administração de usuários: Gerenciamento de acesso, controle de *e-mail*, etc;
- Gestão do curso: O professor tem o controle sobre as atividades.

Os principais módulos que compõem o Moodle são:

- Tarefas: Alunos enviam as tarefas que são avaliadas pelo professor;
- Fórum: Os alunos e professores podem criar tópicos e escrever mensagens sobre algum assunto;

- Questionário: Os alunos devem responder as questões do professor.
- Recursos: É possível fazer *upload* e download de todos os tipos de conteúdo digital;
- *Wiki*: Os estudantes podem colaborar no mesmo documento;
- Diário: Para informação privada entre professor e aluno.

O Univates Virtual é um ambiente de aprendizagem desenvolvido em software livre baseado no Moodle versão 1.9.2. Atualmente está sendo utilizado por diversos professores para apoiar as disciplinas presenciais e para a viabilização de algumas disciplinas oferecidas na modalidade de ensino à distância. A solicitação e o gerenciamento de um espaço neste ambiente são realizados pelo professor da disciplina. Onde o aluno tem acesso apenas aos ambientes das disciplinas nas quais está regularmente matriculado.

3.2 Apache

O Apache, ou Apache HTTP Server, é um servidor web *open-source* desenvolvido e mantido por uma comunidade aberta de desenvolvedores sob a proteção da Apache Software Foundation. O aplicativo está disponível para uma grande variedade de sistemas operacionais incluindo Unix, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows, entre outros (APACHE , 2012).

Por ser um sistema de código aberto, foi muito difundido mundialmente devido às possibilidades de alteração e de melhorias em seu funcionamento, realizáveis por programadores iniciantes. Uma grande vantagem do servidor Web Apache consiste na existência de diferentes versões do programa, as quais permitem que o servidor funcione em diferentes sistemas operacionais, de forma transparente ao usuário (ABREU et al., 2012).

O Apache suporta uma variedade de recursos, muitos implementados como módulos compilados que ampliam a funcionalidade do seu núcleo. Algumas das linguagens de computação suportadas pelo Apache são o Perl, Python e o PHP.

Para garantir segurança nas transações HTTP, o servidor dispõe de um módulo no qual adiciona a capacidade do servidor em atender requisições utilizando o protocolo HTTPS (HyperText Transfer Protocol Secure). Este protocolo utiliza uma camada SSL (Secure Sockets Layer) e TLS (Transport Layer Security) para criptografar todos os dados transferidos entre o cliente e o servidor, provendo maior grau de segurança, confidencialidade e confiabilidade dos dados. A camada SSL é compatível com certificados X.509, padrão utilizado nos certificados digitais fornecidos e assinados por grandes entidades certificadoras no mundo.

3.3 Web Service

O *Web Service* é um serviço que visa permitir a comunicação entre duas aplicações através de uma interface bem definida (CERAMI, 2002). É um serviço disponível na Internet para fornecer informações através de um XML (eXtensible Markup Language) padrão, independente da linguagem de programação ou sistema operacional utilizado.

A independência de linguagem e plataforma dos *Web Services* tornou-os excepcionalmente úteis na integração de sistemas. O único requisito para um sistema se comunicar a um *Web Service* é interpretar uma mensagem XML (ZANETTI; DE CAMARGO, 2012).

Para tornar possível a integração automática de aplicações com o *Web Service* é desejável que o serviço utilize padrões comuns da Internet, seja auto descritivo e possa ser facilmente encontrado por mecanismos de buscas.

Para que os serviços possam ser facilmente encontrados, há um diretório comum onde são armazenadas informações sobre os serviços disponíveis. Este diretório é chamado de UDDI (Universal Description, Discovery and Integration) (CERAMI, 2002).

As descrições de serviços disponíveis no UDDI são documentos no padrão WSDL (Web Service Definition Language). Independente de a descrição estar disponível no UDDI, este possui, além de outras informações, a URL que aponta para o local onde está armazenada a descrição completa do serviço.

Nas seções seguintes detalham-se as tecnologias empregadas, conhecidas como WUST (WSDL, UDDI e SOAP (Single Object Access Protocol) Technologies). Estas tecnologias unidas a XML compõem a estrutura básica dos *Web Services*.

3.3.1 XML

A *Extensible Markup Language* (XML) é o padrão adotado para o transporte de dados (ABINADER; LINS, 2006). Simples de ser compreendida, fácil de ser transportada e compatível com diversas plataformas, a XML é a responsável por grande parte da interoperabilidade entre aplicações de ambientes heterogêneos permitida pelos *Web Services*. A XML serve de base para as outras três tecnologias fundamentais dos *Web Services*: SOAP, UDDI e WSDL.

3.3.2 WSDL

A *Web Services Descriptor Language* (WSDL) é a linguagem utilizada para criar uma autodescrição pública do serviço (ABINADER; LINS, 2006). Entre as informações que podem constar neste documento estão: o padrão utilizado para troca de mensagem (SOAP, JMS (Java Message Service), entre outros), protocolo de transporte utilizado (HTTP (Hyper Text Transfer Protocol), HTTPR (Hyper Text Transfer Protocol Reliable), DIME (Direct Internet Message Encapsulation), entre outros) e o endereço lógico (URL) do serviço.

A WSDL é baseada na XML e o formato das descrições é definido e validado utilizando XML Schema (XSD). A possibilidade de criar descrições dos serviços e, com isso, facilitar a integração automatizada de aplicações, consiste em uma das principais vantagens dos *Web Services* sobre os outros modelos distribuídos.

3.3.3 UDDI

O *Universal Description Discovery and Integration* (UDDI) é o diretório onde as informações sobre os *Web Services* podem ser armazenadas (CERAMI, 2002). O diretório contém mecanismos que permitem a descoberta (procura) de serviços. Uma das principais vantagens da utilização do UDDI é a garantia de que todos os *Web Services* disponíveis que atendam aos requisitos exigidos pelo solicitante tenham sido considerados durante o processo de escolha.

3.3.4 SOAP

O *Single Object Access Protocol* (SOAP) é o protocolo adotado pela maioria dos *Web Services* para troca de mensagens (ABINADER; LINS, 2006). Baseado em XML, o SOAP é projetado para que possa ser utilizado sobre qualquer protocolo, incluindo SMTP (Simple Mail Transfer Protocol), HTTP e FTP (File Transfer Protocol).

O funcionamento do SOAP é similar ao do RPC (Remote Procedure Call) em que a mensagem enviada pelo cliente deve informar qual método deverá ser invocado e os valores de seus parâmetros, se estes existirem. O serviço devolverá uma mensagem contendo o valor retornado pelo método. Esta abordagem deve ser cuidadosamente projetada para causar o menor acoplamento possível entre as aplicações, pois o servidor irá esperar parâmetros de tipos de dados específicos e um formato de mensagem rigorosamente definido. Se a linguagem ou plataforma do solicitante não possuir o tipo de dado esperado ou for incapaz de gerar a mensagem no formato requerido, a comunicação estará comprometida.

Uma alternativa ao modelo de comunicação baseado em RPC é o modelo fundamentado em troca de mensagens. Neste modelo, o servidor espera um documento inteiro para processar e não uma mensagem contendo uma sequência rigorosa de parâmetros. Se o modelo for implementado de maneira síncrona, o solicitante irá esperar algum retorno do serviço, seja somente uma confirmação do recebimento do documento ou algum valor ou documento.

Porém, caso a implementação for feita de maneira assíncrona, o cliente não aguardará uma resposta, cabendo ao serviço decidir enviar ou não algo para o cliente. Em ambos os modelos de comunicação, o conteúdo da mensagem estará dentro de um envelope SOAP.

Segundo estudos de Silva (apud RIBEIRO; BRAGA; SOUSA, 2011), em suma, é um protocolo para acessar *Web Services*. A arquitetura tem sido desenhada para ser independente de qualquer modelo particular de programa e de outras implementações específicas. Seus dois maiores objetivos são a simplicidade e a extensibilidade.

3.4 Android

O Android é uma plataforma *open source* para *smartphones* e *tablets*, baseada no sistema operacional Linux, utilizando a linguagem de programação Java. Foi criada e é mantida pelo Open Handset Alliance (OHA), um consórcio formado por mais de 40 empresas de tecnologia e comunicação, liderada pelo Google. A maior parte do seu código foi lançada pelo Google em 2007, sob uma licença livre, especificamente Apache 2.0.

Os principais objetivos dessa plataforma é ser moderna e flexível, a fim de possibilitar o desenvolvimento de novos aplicativos de forma rápida e moderna, além de possibilitar a personalização das aplicações e componentes do seu sistema operacional (LECHETA, 2009).

Por possuir o código de fonte aberto poderá estar em constante evolução, podendo ser alterado para receber novas tecnologias, conforme estas forem surgindo e as comunidades de desenvolvimento trabalhem em conjunto para construir aplicações móveis inovadoras (PEREIRA; DA SILVA, 2009).

A seguir são apresentadas as principais partes da arquitetura da plataforma Android.

3.4.1 Arquitetura

A arquitetura do Android é dividida da seguinte forma: Linux Kernel, Bibliotecas, Android Runtime, Application Framework e Aplicativos, assim como apresentado na Figura 2. Conhecer a arquitetura, bibliotecas e os principais componentes da plataforma, é muito importante para saber identificar a os limites e dependências no desenvolvimento de uma aplicação.

Figura 2 – Arquitetura da plataforma Android



Fonte: Pereira e Da Silva (2009).

3.4.2 Kernel

Os dispositivos móveis, assim como os computadores e outros dispositivos eletrônicos, possuem arquiteturas distintas entre si. Para que um sistema consiga operar com diferentes tipos de hardware, faz-se necessários os sistemas operacionais. Segundo Tanenbaum (2003), um Sistema Operacional deve ser capaz de gerenciar o processador, memória e outros dispositivos de entrada e saída, além

de fornecer, aos programas de usuário, uma interface simplificada com o hardware. Nos casos dos dispositivos móveis, a tendência de criação de sistemas operacionais, é crescente justamente pela necessidade do gerenciamento dos recursos que os *smartphones* disponibilizam, como processamento e memória. Além disso, é necessário que a criação de software não seja apenas para um único modelo de dispositivo, mas para um determinado sistema operacional, ficando este responsável por gerenciar as particularidades de cada dispositivo.

O Android utiliza o *kernel* do Linux 2.6, que é responsável pelos serviços de segurança, gerenciamento de memória, processos, rede e *drivers*, este último componente é muito importante, pois garante que o desenvolvedor não precisará se preocupar em como acessar ou gerenciar dispositivos específicos do equipamento em questão, produzindo assim uma abstração entre o hardware e o software.

3.4.3 Bibliotecas

Durante o desenvolvimento de programas, existem rotinas básicas que muitas vezes são comuns em muitos softwares, como abrir arquivos, realizar cálculos aritméticos, que não são o foco principal do desenvolvedor. Para se evitar a reescrita dos códigos de funções e rotinas que são repetidas ou possuem altas complexidades, existem as bibliotecas. As bibliotecas possuem códigos e dados que auxiliam na execução de serviços e permite a separação de partes do programa (modularização). O Android possui um conjunto de bibliotecas, disponíveis para a criação de seus aplicativos como a System C Library, Media Libraries, Surface Manager, Lib WebCore, SGL, 3D libraries, FreeType e SQLite. Tais bibliotecas permitem a manipulação de vídeos, imagens, sons, animações, banco de dados, etc.

3.4.4 Android Runtime

Uma aplicação Android é interpretada de maneira análoga ao byte-code Java, com a diferença que o Android gera códigos Dalvik executáveis (.dex), e não os *byte-code* (.class) do Java, tais códigos são interpretados pela Máquina Virtual

Dalvik (MVD). Criada por Dan Bornstein, a MVD é uma alteração da Máquina Virtual Java (JVM), a qual é otimizada para os objetivos que o Android visa suprir (BORNSTEIN, 2008). Após a compilação, todos os arquivos .dex e outros recursos utilizados pela aplicação (exemplo: imagens, sons, etc.), são compactados em um arquivo do tipo .apk (Android Package File), sendo este arquivo, a aplicação finalizada e pronta para ser distribuída e instalada em qualquer dispositivo com Android, (LECHETA, 2009). A MVD também é incluída no SDK (Software Development Kit) 4 do Android, onde transformam os códigos das classes Java (.class) em códigos Dalvik executáveis (.dex), que posteriormente serão executados pelo emulador Android (BORNSTEIN, 2008).

3.4.5 Application Framework

Além das bibliotecas, os desenvolvedores têm a sua disposição, diversos outros componentes que o Android disponibiliza como, o provedor de conteúdo, gerenciador de janela, telefone, recursos, atividades e muitos outros. A possibilidade de acesso e modificação dos componentes do Android é um diferencial, em relação a outras plataformas para dispositivos móveis. Tais componentes permitem que as aplicações criadas possam interagir com todo o sistema do dispositivo móvel, podendo o desenvolvedor, alterar qualquer componente que faça parte do sistema para que esse seja adaptado as suas necessidades ou as do aplicativo que esta sendo desenvolvido.

Os componentes permitem a interoperabilidade entre os vários subsistemas do dispositivo móvel. Por exemplo, um aplicativo pode acessar a agenda de contatos e realizar uma chamada, capturar uma imagem pela câmera, descobrir qual a localização do celular, etc. No endereço do site Android Open Source Project, (<http://source.android.com/>), pode-se obter toda a documentação e ter acesso aos códigos da plataforma Android (GOOGLE ANDROID, 2012). O framework de aplicativo fornece componentes que auxiliam no desenvolvimento dos programas, permitindo a criação de listas, grades, caixas de texto, botões, etc. Alguns componentes se destacam (GOOGLE DEVELOPERS, 2013), tais como:

- O provedor de conteúdo, que controla o acesso aos dados realizados pelos aplicativos;
- O gerenciador de notificações, que habilita os aplicativos a exibirem informações e avisos na barra de status do aparelho, assim como a reagirem às notificações recebidas;
- O gerenciador de atividades, que controla o ciclo de vida das aplicações e permite sua execução em segundo plano.

4 APLICAÇÃO ANDROID

A proposta deste trabalho foi desenvolver um aplicativo na plataforma Android para o ambiente Univates Virtual, onde mais a frente será detalhada os seus requisitos funcionais, não funcionais, além de suas etapas de desenvolvimento, visão geral do aplicativo, ferramentas utilizadas no desenvolvimento e as suas funcionalidades.

4.1 Análise de requisitos

Através da análise das necessidades do aplicativo, foram definidos requisitos que podem ser funcionais ou não funcionais. Sendo os requisitos funcionais aqueles que definem as características, capacidades e segurança e os requisitos não funcionais que afetam outras questões, tais como facilidade de uso, desempenho, confiabilidade ou interface.

4.1.1 Requisitos funcionais

A seguir são listados os requisitos funcionais da aplicação:

- Listar os cursos ativos do usuário no presente semestre;
- Listar os *e-mails* recebidos pelo usuário relacionados ao curso;
- Listar recursos que foram adicionados ao curso;
- Listar tarefas e suas informações relacionadas ao curso;

- Autenticação de usuário e senha;
- Sincronização das informações de forma manual e automática;
- Notificação de novas tarefas, correio eletrônico e recursos;
- Notificação de fim de prazo para entrega de tarefas.

4.1.2 Requisitos não funcionais

Além das funcionalidades já citadas, são necessários à aplicação outros aspectos para atender as necessidades dos usuários. Persistência dos dados da aplicação é um importante requisito não funcional, permitindo que as informações sejam acessadas quando a aplicação passar por um período sem conectividade com a Internet. Outra funcionalidade importante é o tratamento de eventos para telas de múltiplos toques e mudança de orientação do dispositivo móvel. Como existem diferentes tipos de tamanho de tela em dispositivos Android, o layout da aplicação deve ser adequado dinamicamente ao formato da tela para que tenha uma melhor visualização. A interface deve ser intuitiva, a fim de facilitar a utilização da aplicação pelo usuário.

4.2 Desenvolvimento

Como já mencionado anteriormente, já existem projetos desenvolvidos para o Moodle como o caso do projeto Moodbile, o aplicativo Moodle Mobile, entre outros, possuem funcionalidade e design semelhante a este projeto, mas que não possuem utilização específica para o ambiente Univates Virtual. Por isso a necessidade do desenvolvimento de uma aplicação voltado para este projeto.

Partindo deste princípio, a preocupação nesta parte do projeto foi desenvolver um aplicativo para a plataforma Android eficiente, de fácil utilização, possuir confiabilidade e um bom desempenho, já que o mesmo deve funcionar em dispositivos móveis com capacidade de hardware limitada.

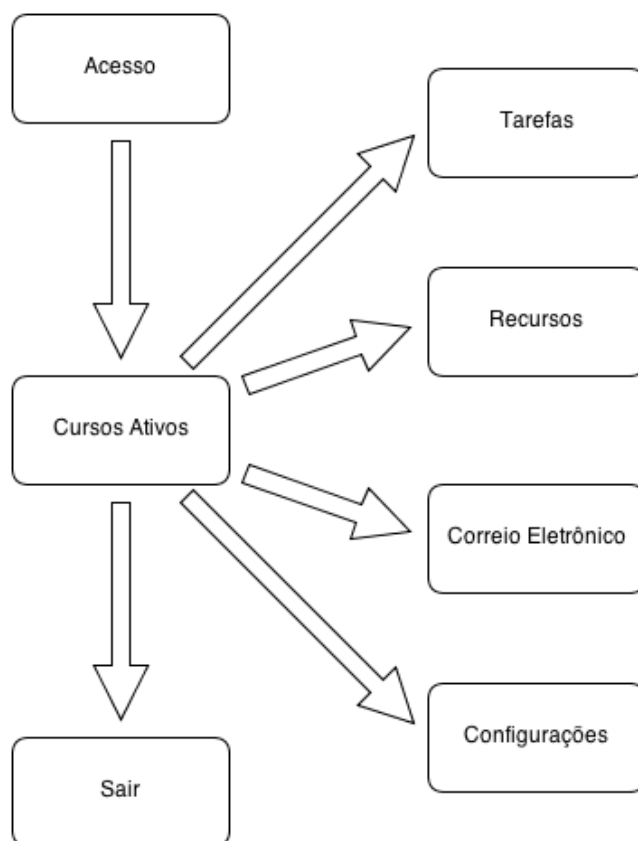
A seguir será apresentada uma visão geral do aplicativo, as ferramentas utilizadas no seu desenvolvimento, as suas funcionalidades, o serviço em *background* e suas permissões.

4.2.1 Visão geral

Um panorama do funcionamento do sistema será explicado baseando-se no diagrama da Figura 3, que representa o fluxo de ações previstas pelo projeto da aplicação.

No diagrama estão listados os recursos da aplicação e a representação como elas podem ser acessadas. Inicialmente, para poder usufruir os demais recursos, o usuário necessita realizar o acesso, utilizando seu código de aluno e sua senha do ambiente Univates Virtual. Após o acesso, aplicação levará diretamente para a tela de cursos ativos do usuário, no caso, aqueles cursos que o usuário faça parte e encontram-se ativos no ambiente Univates Virtual. A partir deste recurso, o usuário tanto pode acessar as tarefas, recursos, correio eletrônico e configurações, ou optar por sair da aplicação.

Figura 3 - Representação do fluxo de utilização da aplicação.

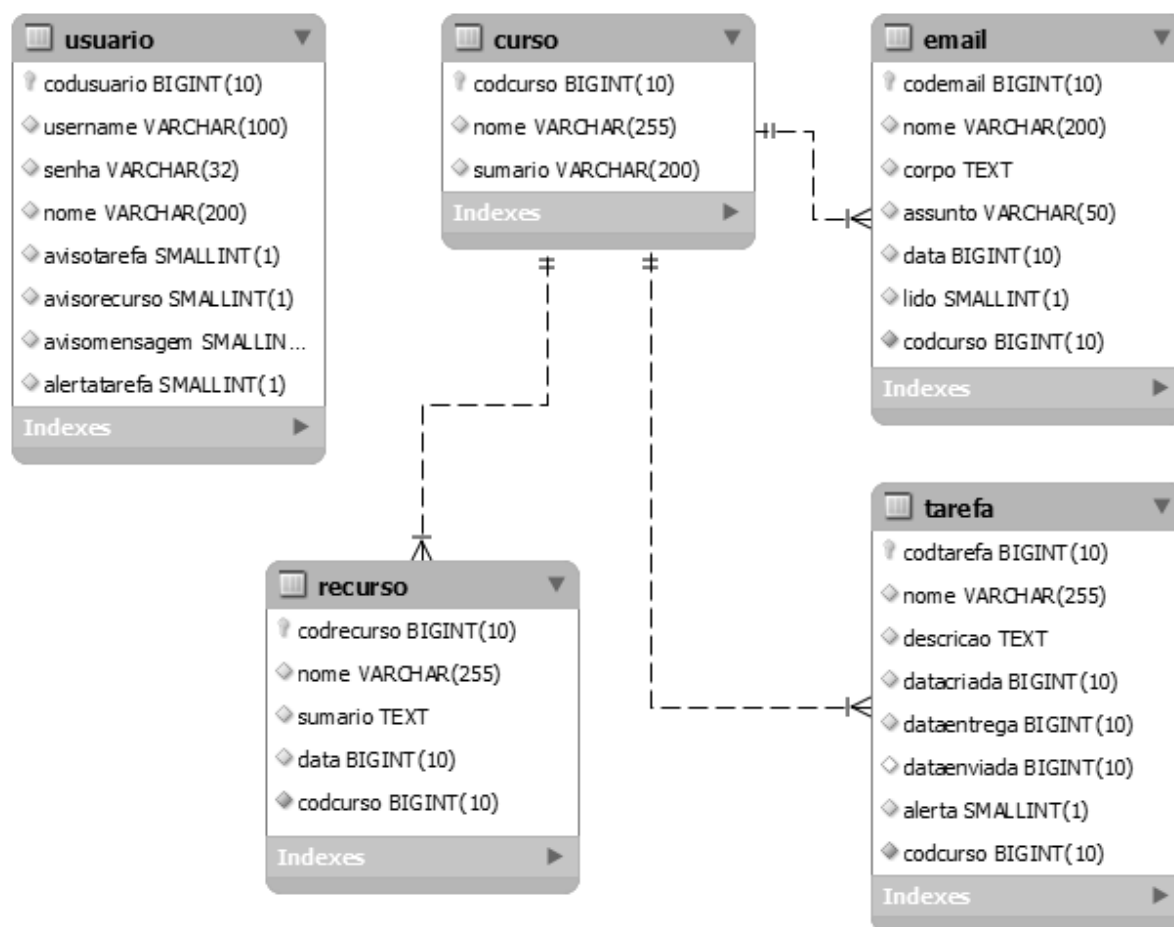


Fonte: Elaborado pelo Autor.

Estas ações não representam o exatamente o número de telas da aplicação, pois ao acessar a tela de tarefas, é possível tocar sobre uma tarefa existente e uma nova tela exibirá mais informações sobre a mesma, assim como no correio eletrônico.

Conforme o diagrama apresentado na Figura 4, uma base de dados foi desenvolvida para salvar os dados da aplicação, de forma que a aplicação não necessite estar conectada a Internet para continuar o seu funcionamento, além de guardar alguns dados importantes para o funcionamento mais adequado da aplicação, salvando informações de configurações do usuário. A base de dados foi desenvolvida baseada na necessidade da aplicação em utilizar os dados armazenados, que foram buscados através do *Web Service* e os campos foram modelados para haver uma compatibilidade com o modelo de banco de dados utilizado pelo Moodle versão 1.9.2.

Figura 4 - Diagrama da base de dados



Fonte: Elaborado pelo Autor.

A tabela *usuario* foi modelada para armazenar as informações relativas ao usuário, e também tem a função de armazenar as configurações de alertas e avisos. Essa tabela não possui nenhum relacionamento com as outras tabelas da base de dados, mas ela armazena dados importantes que são utilizados na comunicação da aplicação com o *Web Service*.

A tabela *curso* foi modelada para armazenar as informações dos cursos nos quais o aluno estiver participando.

As tabelas *tarefa*, *recurso* e *email* foram modelados para armazenar as informações das tarefas, dos recursos e do correio eletrônico de cada curso, por isso elas possuem relação com a tabela *curso*.

4.2.2 Ferramentas

Como a aplicação foi desenvolvida para a plataforma Android, utilizou-se no seu desenvolvimento a ferramenta Eclipse com o *plugin* Android Developer Tool (ADT) ou ferramenta de desenvolvimento Android, que além de auxiliar no desenvolvimento da aplicação no padrão *Model View Control* (MVC), também possui um editor de layout gráfico e outras ferramentas que auxiliam o processo de desenvolvimento.

Foi utilizada no desenvolvimento da aplicação a versão 2.3.3, API 10, do SDK do Android, denominada de Gingerbread, pois como pode ser observado na Tabela 1, é a versão com maior distribuição atualmente entre os dispositivos móveis com a plataforma Android (GOOGLE DEVELOPERS, 2013). Os dados da tabela foram coletados em um período de 14 dias até 1 de maio de 2013. Todas as versões com menos de 0,1% não são apresentadas. Os aplicativos que utilizam essa versão possuem compatibilidade com as versões mais novas, então este aplicativo poderá ser utilizado pelas versões mais novas do Android.

Tabela 1 – Distribuição das versões da plataforma Android

Version	Codename	API	Distribution
1.6	Donut	4	0,1%
2.1	Eclair	7	1,7%
2.2	Froyo	8	3,7%
2.3 - 2.3.2	Gingerbread	9	0,1%
2.3.3 - 2.3.7		10	38,4%
3.2	Honeycomb	13	0,1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	27,5%
4.1.x	Jelly Bean	16	26,1%
4.2.x		17	2,3%

Fonte: Google Developers (2013).

Foi utilizado também a biblioteca kSOAP 2 Android versão 3.0.0 para o desenvolvimento do tratamento da aplicação pelos dados obtidos através do *Web Service*. O kSOAP 2 Android é uma biblioteca desenvolvida em linguagem Java, construída a partir da biblioteca kSOAP 2, com o objetivo de ser um cliente *Web Service* SOAP, sendo eficiente e não exigindo muitos recursos da plataforma Android.

Para auxiliar no desenvolvimento e modelagem da base de dados da aplicação, foi utilizado a ferramenta SQLite Expert Personal 3 e para o gerenciamento da base de dados, foi utilizada a biblioteca SQLite nativa da plataforma Android.

4.2.3 Funcionalidades

A princípio as funcionalidades desta aplicação foram projetadas procurando realizar de forma simples e eficiente, as tarefas que os usuários normalmente realizam ao acessar o ambiente Univates Virtual através do navegador de Internet em um computador desktop ou notebook. A seguir serão apresentadas as telas da aplicação e comentadas as suas funcionalidades.

4.2.3.1 Acesso

Ao ser acessado, o aplicativo apresenta uma tela inicial, conforme a Figura 5, onde o usuário deve realizar o acesso na aplicação validando seus dados no ambiente Univates Virtual. Essa tela possui apenas dois campos de texto, no qual o usuário deve preencher com o seu código de aluno da Univates e sua senha de acesso. Também é apresentado um botão para acessar as demais funcionalidades, após a autenticação ser concluída.

Figura 5 - Tela de acesso da aplicação

A imagem mostra a interface de login de um aplicativo móvel. No topo, há uma barra de status com ícones de sinal, bateria e o horário 17h23. Abaixo, o título 'UNIVATES VIRTUAL' aparece em uma barra cinza. O fundo da tela é azul. No centro, o texto 'UNIVATES VIRTUAL' é exibido em branco. Abaixo dele, o rótulo 'Código de Aluno' precede um campo de entrada contendo o número '509119'. Logo abaixo, o rótulo 'Senha' precede um campo de entrada com pontos e uma barra de cursor; este campo está destacado por uma borda laranja. Na base, um botão cinza com o texto 'Acessar' está disponível.

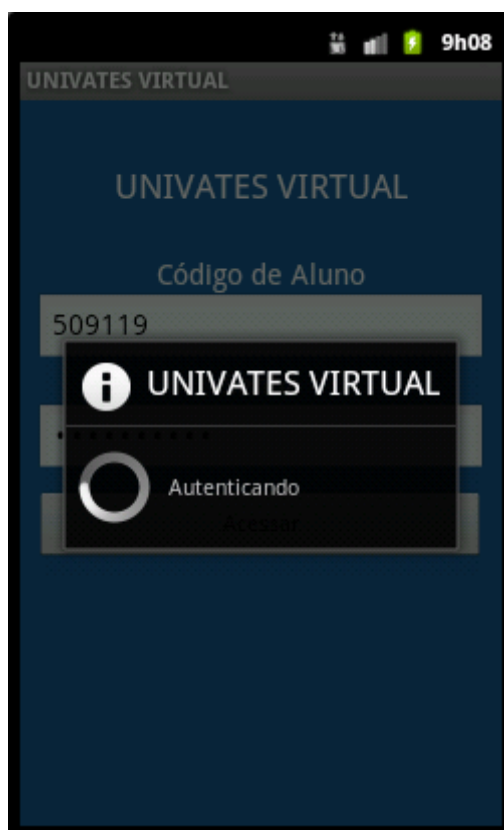
Fonte: Elaborado pelo Autor.

4.2.3.2 Autenticação do usuário

Para realizar a autenticação do usuário no ambiente do Univates Virtual, é necessário o dispositivo móvel estar conectado a Internet, permitindo assim a troca de dados com *Web Service*. Caso a conexão existir, e o usuário for autenticado, a aplicação busca os dados no ambiente Univates Virtual e salva as informações na base de dados da aplicação, permitindo assim, que estas informações possam estar disponíveis ao usuário mesmo em períodos sem conexão do dispositivo móvel com a Internet.

Conforme a Figura 6, enquanto a aplicação realiza a troca de informações com o *Web Service* para realizar a autenticação do usuário, a tela esmaece e exibe uma mensagem para informar o usuário sobre a autenticação.

Figura 6 – Tela de autenticação do usuário

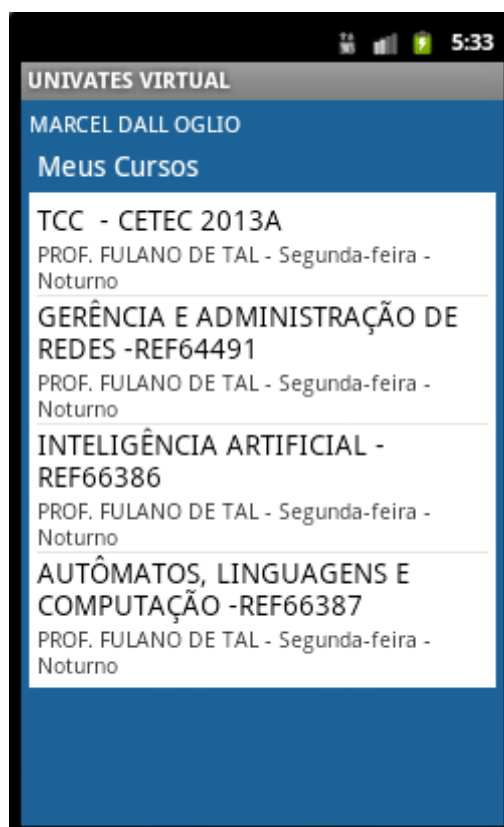


Fonte: Elaborado pelo Autor.

4.2.3.3 Cursos

Após a etapa de autenticação, uma nova tela é chamada e exibida pela aplicação, conforme apresentado na Figura 7, a tela de cursos exibe uma lista com os cursos ativos do usuário no ambiente Univates Virtual. Cada item da lista apresenta o nome do curso em destaque, o nome do professor, o dia da semana e o turno em que o curso ocorre.

Figura 7 - Tela de cursos



Fonte: Elaborado pelo Autor.

4.2.3.4 Tarefas

Ao selecionar um item da lista de cursos pressionando sobre o mesmo, a aplicação exibirá uma nova tela, conforme a Figura 8, na qual apresenta três opções, tarefas, recursos e correio eletrônico. Por padrão a tela mostra primeiramente a lista de tarefas relacionadas ao curso selecionado, mas ao deslizar o dedo sobre a tela do dispositivo, a próxima opção é exibida, ou simplesmente pressionando a opção desejada. No caso da tela de tarefas, é exibido o nome da tarefa a data limite de entrega da tarefa no ambiente Univates Virtual, a data em que a tarefa foi enviada, caso já tenha sido enviada, ou tarefa não enviada, caso ainda não tenha sido enviada.

O aplicativo foi desenvolvido, para exibir somente as informações mais importantes e que auxiliem o usuário a organizar-se em relação aos seus cursos em atividade. Devido ao tamanho reduzido da maioria dos dispositivos móveis, exibir todas as informações do ambiente Univates Virtual na aplicação poderia acarretar

em uma interface poluída e com menos eficiência, além de aumentar o consumo de dados através do *Web Service*.

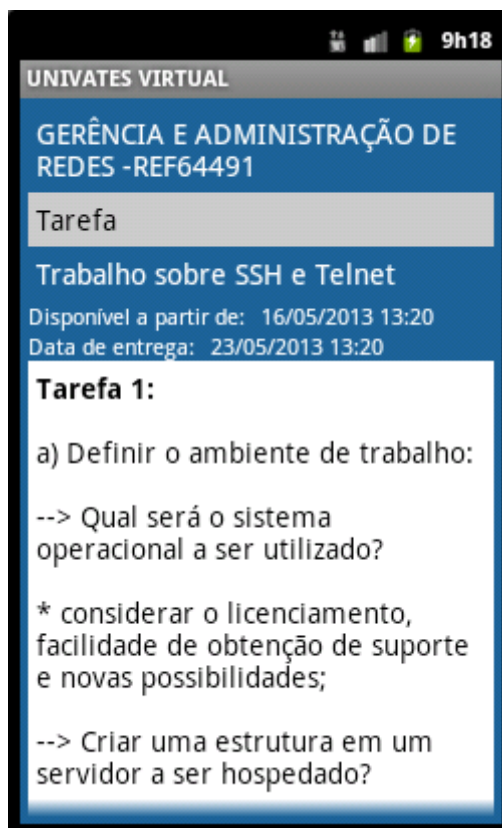
Figura 8 - Tela de tarefas do curso



Fonte: Elaborado pelo Autor.

Ao pressionar em um dos itens da lista, é exibida uma nova tela com a descrição completa da tarefa solicitada pelo professor do curso, conforme a Figura 9. Nesta tela aparecem, o nome do curso em que a tarefa foi disponibilizada, o título da tarefa, a data em que foi disponibilizada, a data de entrega da tarefa definida pelo professor e a descrição completa da tarefa.

Figura 9 - Tela de tarefa com descrição completa



Fonte: Elaborado pelo Autor.

4.2.3.5 Recursos

A tela de recursos do curso, conforme a Figura 10 é muito semelhante à tela de tarefas do curso. Os itens são exibidos em uma lista. Cada item possui o nome e o sumário do recurso, conforme disponibilizado pelo professor. A diferença desta tela fica por conta de que ao pressionar o item não é exibida uma nova tela, já que todas as informações estão nesta tela.

Figura 10 - Tela de recursos do curso



Fonte: Elaborado pelo Autor.

4.2.3.6 Correio eletrônico

A tela de correio eletrônico, conforme a Figura 11, também funciona de forma muito semelhante à tela de tarefas, exibindo uma lista de itens, que correspondem aos e-mails recebidos pelo usuário no determinado curso. Cada item apresenta o remetente, o assunto e a data em que o *e-mail* foi enviado. Ao pressionar o *e-mail*, é exibida uma nova tela com o texto do corpo do *e-mail*.

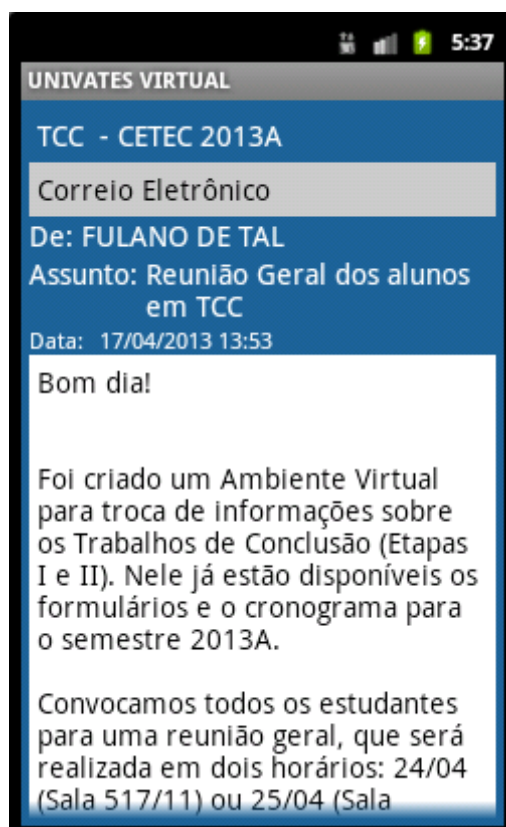
Figura 11 - Tela de correio eletrônico do curso



Fonte: Elaborado pelo Autor.

Na Figura 12, a tela de *e-mail* exibe o curso do qual pertence o *e-mail*, ao remetente, o assunto, a data em que foi enviado, e o corpo do *e-mail* de forma completa. Essa tela possui a funcionalidade de apenas visualizar o *e-mail* recebido pelo usuário, não sendo possível respondê-lo, apaga-lo ou realizar outras funções normalmente permitidas através de um navegador web, estas funcionalidades não foram inicialmente implementadas por exigirem diversas adaptações no *Web Service*, e não estarem nos requisitos da aplicação, sendo assim, ficam como trabalhos futuros a serem realizados conforme apresentadas posteriormente no Capítulo 6, página 63.

Figura 12 - Tela de e-mail

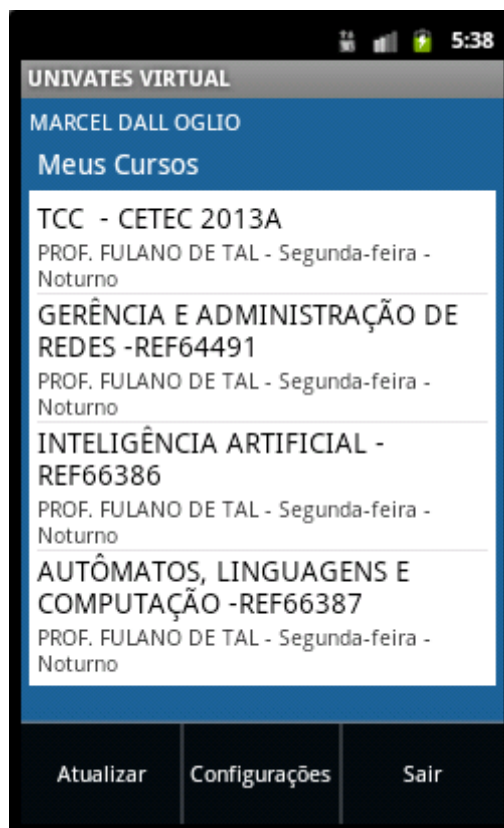


Fonte: Elaborado pelo Autor.

4.2.3.7 Opções

Ao pressionar a tecla ou opção Menu do dispositivo, na tela de cursos, a aplicação passa a exibir um pequeno menu de opções na parte inferior da tela do dispositivo, conforme apresentado na Figura 13. A opção Atualizar, ao ser pressionado executa uma verificação em busca de atualizações no ambiente virtual, trocando informações com o *Web Service*. A opção Configurações exibirá uma tela com as opções de configuração da aplicação, que será apresentada de forma detalhada posteriormente. A opção Sair possui a funcionalidade de sair da aplicação e apagar os dados do usuário, caso seja optado, e será apresentada de forma detalhada posteriormente na página 52.

Figura 13 - Tela de cursos com opções visíveis.

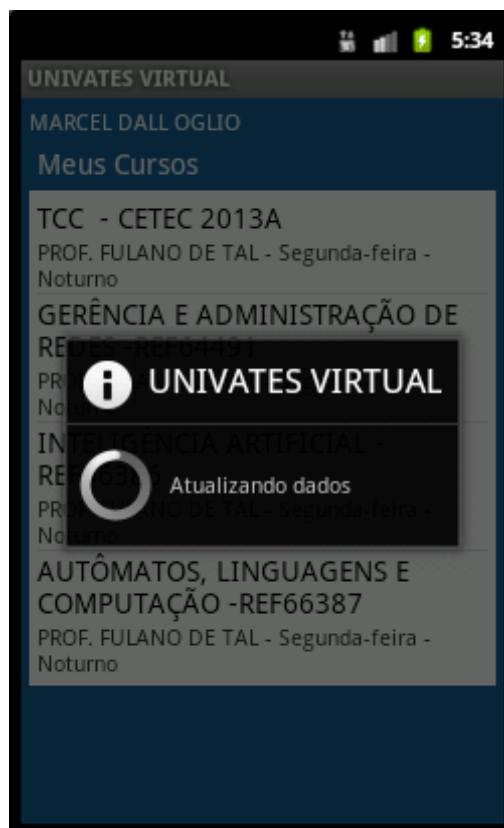


Fonte: Elaborado pelo Autor.

4.2.3.8 Atualização dos dados

Quando a opção Atualizar for pressionada a aplicação busca os dados do ambiente virtual através do *Web Service*, por este motivo, assim como na autenticação do usuário na tela de acesso, a tela esmaece e exibe uma mensagem ao usuário, para informa-lo que a atualização dos dados está sendo executada, conforme apresentado na Figura 14. Ao exibir a mensagem, as outras funcionalidades ficam bloqueadas, para evitar que outros processos entrem em execução. Ao terminar a atualização, a mensagem é ocultada e usuário volta a ter permissão para acessar outras funcionalidades do aplicativo.

Figura 14 – Tela de atualização dos dados



Fonte: Elaborado pelo Autor.

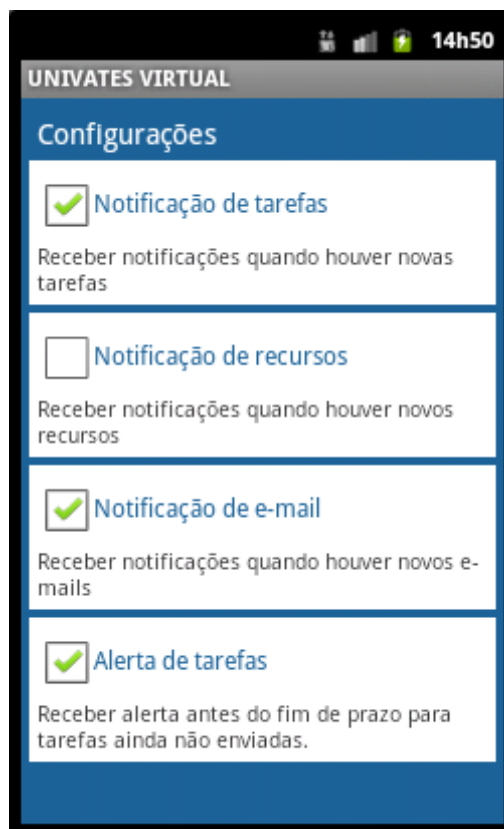
4.2.3.9 Configurações

Ao pressionar a opção Configurações, uma nova tela é exibida, conforme apresentado na Figura 15, onde estão presentes as opções e a descrição das funcionalidades que usuário pode optar por estarem ativadas ou desativadas no aplicativo, conforme o *checkbox* ao lado de cada item estiver marcado. Existem quatro opções ao usuário para receber notificações, quando uma nova tarefa for disponibilizada pelo professor, quando um novo recurso for disponibilizado pelo professor, quando um novo *e-mail* for recebido pelo usuário e quando uma tarefa esta próxima do prazo de entrega e ainda não tenha sido enviada pelo aluno, e todas essas opções começam ativadas por padrão no aplicativo.

Ao pressionar o *checkbox* de uma das opções, no mesmo instante a informação é salva pela aplicação e seu estado de ativado passa para desativado, ou vice-versa, sendo exibida uma pequena mensagem ao usuário na parte inferior da tela. Essas notificações e o alerta de tarefa funcionam em *background* (não

visível para o usuário) no dispositivo, e essas funcionalidades serão apresentadas de maneira detalhada posteriormente na página 53.

Figura 15 - Tela de configurações

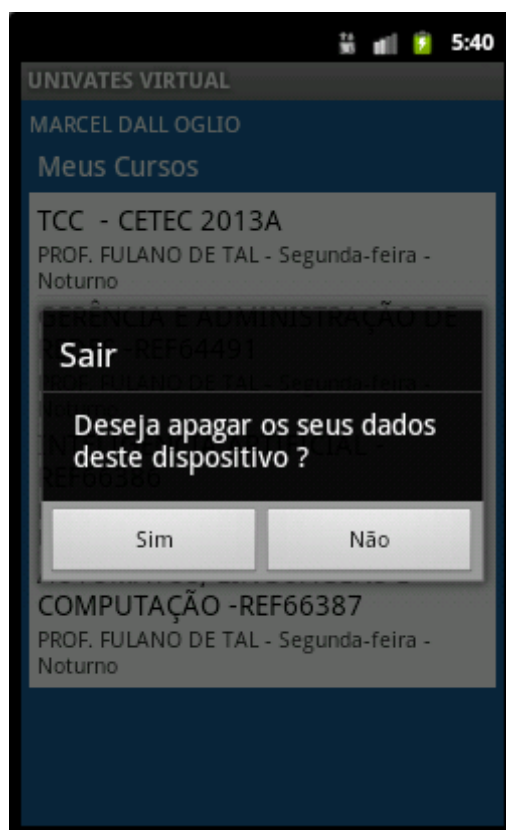


Fonte: Elaborado pelo Autor.

4.2.3.10 Sair da aplicação

Caso o usuário pressionar a opção Sair uma mensagem será exibida na tela, conforme apresentado na Figura 16, questionando ao usuário a possibilidade de apagar os seus dados do dispositivo.

Figura 16 - Tela para sair do aplicativo com opção de apagar dados



Fonte: Elaborado pelo Autor.

Caso o usuário optar pela opção Sim, os dados serão apagados do dispositivo, a aplicação, as verificações de atualizações e os alertas em background serão encerrados. Caso o usuário opte pela opção Não, a aplicação é finalizada, os dados do usuário continuam salvos no dispositivo. As verificações de atualizações e alertas em background continuam funcionando normalmente, e a próxima vez que a aplicação for executada pelo usuário, não será necessário fornecer o código de aluno e a senha.

4.2.4 Serviço em background

Na plataforma Android, um Serviço é um componente de aplicação que representa tanto o desejo de uma aplicação executar uma operação de longa duração enquanto não há interação com o usuário ou fornecer funcionalidades para outras aplicações usarem (GOOGLE DEVELOPERS, 2013).

Neste projeto foi criado um serviço com intuito de executar uma operação de longa duração, que é buscar por novas atualização no ambiente virtual através do *Web Service* em um determinado intervalo de tempo. O serviço em *background* é uma parte da aplicação que não fica visível ao usuário do dispositivo móvel, nesta aplicação ele inicia após a autenticação do usuário. Enquanto o dispositivo móvel estiver ligado, ele funciona por trás das outras aplicações e tem como principais funcionalidades, atualizar os dados do aplicativo e emitir notificações, assim como também é utilizado por outros aplicativos do Gmail, Facebook, Instagram entre outros.

O serviço desta aplicação é iniciado pela função *startService*, conforme apresentado na Listagem 1, onde é criado um novo objeto *Timer*, que fica programado para criar um novo objeto através da função *scheduleAtFixedRate*, o *TimerTask*, que representa uma tarefa que vai executar a função *verificarAtualizacoes*, em um determinado intervalo de tempo, sendo este tempo definido na variável local *intervalo*.

Listagem 1 – Função que inicia o serviço em background

```
public void startService() {

    long intervalo = 30 * 60000;

    new Timer().scheduleAtFixedRate(new TimerTask() {

        public void run() {

            verificarAtualizacoes();

        }

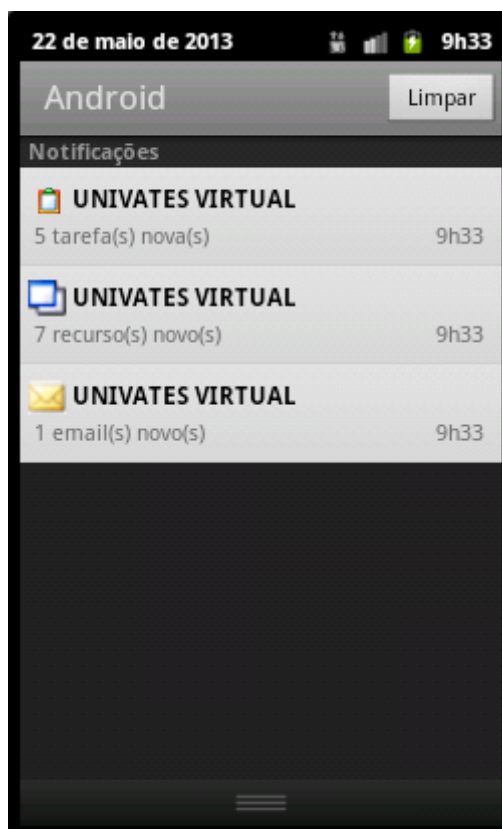
    }, intervalo, intervalo);

}
```

Fonte: Elaborado pelo autor.

Ao verificar por atualizações, o serviço primeiramente verifica pela opção do usuário optar por receber notificações, que podem ser alteradas nas configurações da aplicação e a disponibilidade de conexão com a Internet pelo dispositivo móvel, no caso de positivo nessas situações, a aplicação busca os dados através do *Web Service*, e no caso de haver atualizações de tarefas, recursos ou *e-mails* é gerado pelo serviço notificações de acordo com os dados, conforme representado na Figura 17. Também pode ser gerada uma notificação no caso de uma tarefa estiver próxima do prazo de entrega e ainda não tenha sido enviada pelo aluno, neste caso a informação é buscada na base de dados da aplicação, não sendo necessário o dispositivo estar conectado a Internet.

Figura 17 – Notificações da aplicação



Fonte: Elaborado pelo autor.

4.2.5 Permissões

O Android é um sistema operacional com privilégios separados, em que cada aplicativo é executado com um sistema de identidade distinta (GOOGLE

DEVELOPERS, 2013). E recursos adicionais de aplicações são fornecidos através de um mecanismo de permissão.

Nesta aplicação foram utilizados diversos recursos adicionais fornecidos pela plataforma Android e como consequência foi necessário a inclusão de permissões no arquivo `AndroidManifest.xml` da aplicação para que esses recursos funcionassem. Assim quando o usuário for instalar o aplicativo em seu dispositivo móvel, ele deve concordar que o aplicativo tenha permissão para acessar estes recursos. Nesta aplicação foram adicionadas permissões para o uso de recursos como a possibilidade de, verificar a conectividade com a Internet, verificar a utilização da conexão *Wifi*, acessar a Internet através do dispositivo para realizar as requisições ao *Web Service* para buscar e atualizar os dados do ambiente Univates Virtual, criar vibração no dispositivo quando criadas notificações e verificar a inicialização completa do dispositivo para inicializar os serviços em *background*. Conforme apresentado na Listagem 2, os códigos para permissão de uso dos recursos pela aplicação.

Listagem 2 – Permissões adicionadas no arquivo `AndroidManifest.xml`

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

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

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

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

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

Fonte: Elaborado pelo autor.

5 WEB SERVICE

Para a utilização dos dados através do aplicativo para Android, é necessário buscar as informações na base de dados do Univates Virtual utilizando um *Web Service*, ou seja, um serviço através da Internet. O *Web Service* tem como objetivo de realizar a comunicação entre o aplicativo e o Univates Virtual, fornecendo as informações que forem solicitadas. A seguir será apresentada uma visão geral do *Web Service*, as ferramentas utilizadas no seu desenvolvimento e as suas funcionalidades.

5.1 Visão geral

Como a versão do Moodle utilizado no Univates Virtual não possui componentes que permitam a comunicação entre diferentes plataformas, foi necessário o desenvolvimento de um *Web Service*. O *Web Service* foi projetado para ser utilizado especificamente para o aplicativo desenvolvido neste projeto, o qual não o impede ser utilizado para outras funcionalidades, desde que utilizado o mesmo protocolo de comunicação. Ele foi projetado especificamente para esta aplicação a fim de torna-lo simples e eficiente.

Para o usuário, o *Web Service* é transparente, tendo a finalidade de transmitir os dados requisitados pela aplicação de forma como que o usuário estivesse no próprio ambiente do Univates Virtual. Suas funcionalidades, de autenticação e buscar dados do usuário, consultar cursos, tarefas, recursos e *e-mails*, serão apresentadas de forma detalhada a partir da página 58.

5.2 Ferramentas

Para apoiar no desenvolvimento do *Web Service* foi utilizada a versão 1.9.2 do LMS Moodle, afim de entender seu funcionamento de forma detalhada, permitindo um correto desenvolvimento das funcionalidades do *Web Service*. A linguagem de programação adotada para o desenvolvimento do *Web Service* foi o PHP versão 5.4.4, pelo fato de o PHP ser a mesma linguagem de programação utilizada no Moodle. Permitindo assim, o *Web Service* ser implementado no mesmo servidor web que o Moodle.

O servidor web utilizado para permitir o funcionamento do Moodle foi o software Apache HTTP Server versão 2.4.4. O sistema de gerenciamento de banco de dados (SGDB) utilizado para armazenar o dados do Moodle foi o software PostgreSQL versão 9.0.13. Para apoiar no desenvolvimento das consultas aos dados armazenados pelo Moodle foi utilizado a ferramenta phpPgAdmin versão 5.1.

O PHP possui um módulo nativo para criação de *Web Services*, denominado *SOAP Extension*, mas para auxiliar no desenvolvimento deste *Web Service* foi utilizada uma biblioteca também desenvolvida em PHP, o NuSOAP versão 0.9.5, que serve como alternativa ao *SOAP Extension*. Essa biblioteca foi escolhida principalmente pela sua simplicidade e facilidade de uso, oferecendo suporte a criação automática do WSDL. O NuSOAP é composto por um grupo de classes PHP que permite o consumo de *Web Services* baseados em SOAP de maneira simples pelos desenvolvedores, além de não requerer nenhuma extensão especial do PHP (NUSOAP, 2012).

5.3 Funcionalidades

Para este projeto, as funcionalidades do *Web Service* foram definidas de acordo com o requisitos elaborados no desenvolvimento do aplicativo para a plataforma Android. Para não haver um consumo de dados elevado pela aplicação e pelo *Web Service* ao consultar os dados, foi definido que, como por exemplo, na consulta aos dados do usuário, fossem retornado somente as informações utilizadas

na aplicação e não todas as informações que são gravadas na base de dados pelo Moodle para cada usuário.

Por questão de segurança, foi criado na base de dados um usuário especial, com permissões específicas de apenas realizar consultas, impossibilitando assim que dados sejam alterados ou apagados da base de dados através do *Web Service*, apenas podendo serem consultados.

A seguir serão apresentadas detalhadamente cada funcionalidade do *Web Service* e os dados que ele retorna ao ser requisitado.

5.3.1 Autenticação

A autenticação do usuário da aplicação é uma funcionalidade interna do *Web Service*. Sempre que é realizada uma requisição ao *Web Service*, a autenticação é chamada recebendo o nome de usuário e a sua senha como parâmetros. Se o usuário for autenticado com sucesso é retornado um valor booleano verdadeiro, se não, é retornado um valor booleano falso.

Para realizar a autenticação do usuário é utilizada uma conexão ao servidor LDAP e depois uma chamada para verificar a autenticidade do usuário no servidor.

5.3.2 Consulta de usuário

Essa funcionalidade do *Web Service* consiste em receber um pacote SOAP ao ser requisitado, como o nome de usuário e a senha do Moodle, chama a função de autenticação para validar o usuário, e em caso de sucesso na autenticação, realiza a consulta a seus dados no banco de dados como, o seu código de usuário do Moodle, nome e sobrenome. No caso de o usuário não ser autenticado, por nome de usuário inválido, por senha inválida, ou a consulta retornar um resultado inválido ou algum outro motivo, o *Web Service* retornar um pacote SOAP com valor nulo para a aplicação, e no caso de a autenticação ser validada será retornado um pacote com um objeto com os dados do usuário.

5.3.3 Consulta de cursos

Na consulta de cursos, ao ser requisitado o *Web Service* recebe um pacote SOAP com o nome de usuário do Moodle e a senha, e assim como na consulta de usuário chama a função de autenticação, e após, consulta os cursos em que o usuário está devidamente matriculado no período atual. Caso a consulta ocorrer corretamente, o *Web Service* retorna para a aplicação um pacote SOAP com uma lista de objetos com os dados dos cursos, como o nome do curso, o código do curso e o sumário do curso. No caso de a consulta não ter um resultado válido ou o usuário não ser autenticado, é retornado um pacote com valor nulo para aplicação.

5.3.4 Consultas de tarefas

Assim como na consulta de cursos, a consulta de tarefas, ao ser requisitado o *Web Service* recebe um pacote SOAP com o nome de usuário do Moodle e a senha, consulta as tarefas de todos os cursos em que o usuário está devidamente matriculado no período atual. Caso a consulta ocorrer corretamente o *Web Service* retorna a aplicação, um pacote SOAP com uma lista de objetos com os dados das tarefas, como o código da tarefa, o código do curso, o nome da tarefa, a descrição da tarefa, a data em que foi disponibilizada pelo professor, a data limite de entrega e a data em que foi enviada para avaliação. No caso de a consulta não ter um resultado válido ou o usuário não ser autenticado, é retornado um pacote com valor nulo para aplicação.

5.3.5 Consulta de recursos

Na consulta de recursos, ao ser requisitado o *Web Service* recebe um pacote SOAP com o nome de usuário do Moodle e consulta os recursos de todos os cursos em que o usuário está devidamente matriculado no período atual. Caso a consulta ocorrer corretamente o *Web Service* retorna a aplicação, um pacote SOAP com uma lista de objetos com os dados dos recursos, como o código do recurso, o código do curso, o nome do recurso, o sumário do recurso e a data em que foi disponibilizado

pelo professor. No caso de a consulta não ter um resultado válido ou o usuário não ser autenticado, é retornado um pacote com valor nulo para aplicação.

5.3.6 Consulta de e-mails

Na consulta de *e-mails*, ao ser requisitado o *Web Service* recebe um pacote SOAP com o nome de usuário do Moodle e consulta os *e-mails* dos cursos em que o usuário está devidamente matriculado no período atual. Caso a consulta ocorrer corretamente o *Web Service* retorna a aplicação, um pacote SOAP com uma lista de objetos com os dados dos *e-mails*, como o código do *e-mail*, o código do curso, o assunto do *e-mail*, o nome do remetente do *e-mail*, o corpo do *e-mail* e a data em que foi enviado. No caso de a consulta não ter um resultado válido ou o usuário não ser autenticado, é retornado um pacote com valor nulo para aplicação.

6 CONCLUSÃO

Através do estudo realizado é possível observar que com a crescente utilização de ambientes virtuais de aprendizagem como o Moodle por Instituições de ensino, a facilidade crescente dos alunos em obter acesso a Internet e a crescente utilização de dispositivos móveis, grande parte com a plataforma Android, fica eminente a demanda de softwares capazes de fazer uma integração entre o ambiente de aprendizagem e o dispositivo móvel, para fornecer ao estudante a possibilidade de desenvolver o seu conhecimento de qualquer localização com acesso a Internet.

A partir do desenvolvimento realizado foi possível observar que este projeto pode colaborar de forma expressiva no caso específico do ambiente Univates Virtual, mostrando que a aplicação funciona de forma eficaz naquilo em que foi projetada, dessa forma possibilitando a comunidade acadêmica manter-se atualizada sobre os recursos disponibilizados através do ambiente de aprendizado virtual.

A integração entre diferentes plataformas de sistemas sempre são obstáculos tecnológicos, mas pode ser observado por este projeto que, com a utilização de tecnologias existentes, esses obstáculos podem ser superados, obtendo assim a garantia que a integração tenha sucesso nos seus objetivos.

O *Web Service* desenvolvido neste projeto mostrou-se eficiente na comunicação com a aplicação desenvolvida, mesmo sendo plataformas e linguagens de programação distintas, o que permitiu que o mesmo possa ser reutilizado em diversas outras aplicações, mesmo que em diferentes plataformas e

em diferentes linguagens de programação, evitando assim que o mesmo torne-se restrito e pouco aproveitado.

Devido ao potencial de utilização que os dispositivos móveis vêm apresentando atualmente, cada vez mais se mostra necessário prestar atenção a essa nova tendência, a fim de aproveitar melhor a capacidade do ensino virtual e a exploração dessas tecnologias, muito devido também às mudanças das novas gerações de estudantes, que cada vez mais dispõem destes artefatos e já acostumados a acessar informações de forma rápida, através dos dispositivos móveis.

6.1 Trabalhos futuros

Existem diversas melhorias que ainda podem ser desenvolvidas para tornar a aplicação e o *Web Service* ainda mais completos, a fim de satisfazer mais necessidades na utilização do ambiente Univates Virtual pelos estudantes, como por exemplo:

- Acesso ao perfil dos participantes de cada curso;
- Envio de mensagens de bate-papo entre os participantes;
- Enviar, responder e apagar *e-mails* do curso;
- Enviar ou reenviar arquivos de tarefas;
- Realizar downloads de recursos disponibilizados pelos professores;
- Acessar cursos antigos em que o usuário participou;
- Acessar o portfólio e realizar download e *upload* de arquivos;
- Realizar compartilhamento de arquivos do portfólio com outros participantes.

REFERÊNCIAS

ABINADER, Jorge Abilio; LINS, Rafael Dueire. **Web services em Java**. Brasport, 2006.

ABREU, Thiago WM et al. **Parametric identification techniques applied to dynamic modeling of an Apache webserver**. Sba: Controle & Automação Sociedade Brasileira de Automática, v. 23, n. 1, p. 38-48, 2012. Disponível em: <<http://www.scielo.br/pdf/ca/v23n1/a04v23n1.pdf>>. Acesso em: 01 de jun. 2013.

ALENCAR, Andréia de Souza et al. **O Moodle como Ferramenta Didática**. In: Anais do Congresso Nacional Universidade, EAD e Software Livre. 2012.

APACHE, Software Foundation. **The Apache Foundation Software**, 2012. Disponível em: <<http://www.apache.org>>. Acesso em: 01 de jun. 2013.

BORNSTEIN, Dan. **Dalvik vm internals**. In: Google I/O Developer Conference. 2008. p. 17-30. Disponível em: <http://fiona.dmcs.pl/podyplomowe_smtm/smob3/Presentation-Of-Dalvik-VM-Internals.pdf>. Acesso em: 11 de set. 2012.

CERAMI, Ethan. **Web services essentials**: distributed applications with XML-RPC, SOAP, UDDI & WSDL. O'Reilly Media, Incorporated, 2002.

GOOGLE ANDROID. **Android Open Source Project**, 2012. Disponível em: <<http://source.android.com>>. Acesso em: 24 de set. 2012.

GOOGLE DEVELOPERS. **Android Developers**, 2013. Disponível em: <<http://developer.android.com>>. Acesso em: 21 de maio 2013.

JUNIOR, João Batista Bottentuit; COUTINHO, Clara; ALEXANDRE, Dulclerci Sternadt. **M-Learning e Webquests**: as novas tecnologias como recurso pedagógico. Anais do Simpósio Brasileiro de Informática na Educação. 2006. p. 70-72.

LECHETA, Ricardo R. **Google Android - Aprenda A Criar Aplicações**: Para Dispositivos Móveis Com O Android SDK. Novatec Editora, 2009.

MARCELO, Antônio. **Apache**. Brasport, 2005.

MOODLE. **About Moodle**, 2012. Disponível em: <<https://moodle.org/about/>>. Acesso em: 15 de ago. 2012.

NUSOAP. **SOAP Toolkit for PHP**, 2012. Disponível em: <<http://nusoap.sourceforge.net>>. Acesso em: 18 de set. 2012.

PEREIRA, Lucio Camilo Oliva; DA SILVA, Michel Lourenço. **Android para desenvolvedores**. Brasport, 2009.

SANTOS, J. R. **A Moodle nas práticas pedagógicas de uma escola básica: realidade ou ficção na inserção das TIC em sala de aula**. Educação, Formação & Tecnologias, 2012. Disponível em: <<http://eft.educom.pt/index.php/eft/article/view/280/159>>. Acesso em: 21 de maio 2013.

SILVA apud RIBEIRO, Ticianne de Gois; BRAGA, Antônio Rafael; PIMENTEL, Veronica Lima. **Um Ambiente Virtual de Ensino e Aprendizagem Baseado em Web Semântica e Web Services**. Centro Federal de Educação Tecnológica do Ceará–CEFET-Ce, 2007.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. Editora Pearson, 2 ed. São Paulo, 2003.

ZANETTI, Alberto René; DE CAMARGO, Valter Vieira. **Web Service no ambiente escolar: um estudo de caso**. Revista TIS, v. 1, n. 1, 2012. Disponível em: <<http://www.revistatis.dc.ufscar.br/index.php/revista/article/view/13/17>>. Acesso em: 25 de mar. 2013.