

DIEGO NOGUEIRA TEIXEIRA

**DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA ACESSO AO AMBIENTE
VIRTUAL DE APRENDIZAGEM MOODLE (MOBIMOODLE).**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2008

DIEGO NOGUEIRA TEIXEIRA

**DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA ACESSO AO AMBIENTE
VIRTUAL DE APRENDIZAGEM MOODLE (MOBIMOODLE).**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de concentração:

Engenharia de Software

Orientador:

Prof. Dr. Ahmed Ali Abdalla Esmín

LAVRAS
MINAS GERAIS – BRASIL
2008

**Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca
Central da UFLA**

Teixeira, D. N.

Desenvolvimento de um Aplicativo Móvel Para Acesso ao Ambiente Virtual de Aprendizagem Moodle (MobiMoodle) / Diego Nogueira Teixeira – Minas Gerais, 2008, 39p., il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Dispositivos Móveis. 2. Aplicativos Móveis. 3. Ambiente Moodle. I. TEIXEIRA, D. N..
II. Universidade Federal de Lavras. III. Título.

DIEGO NOGUEIRA TEIXEIRA

**DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA ACESSO AO AMBIENTE
VIRTUAL DE APRENDIZAGEM MOODLE (MOBIMOODLE).**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 20 de Junho de 2008.

Prof^a. Dra. Marluce Rodrigues Pereira

Prof. Dr. André Luiz Zambalde

Prof. Dr. Ahmed Ali Abdalla Esmim
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL
2008

DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA ACESSO AO AMBIENTE VIRTUAL DE APRENDIZAGEM MOODLE (MOBIMOODLE).

RESUMO

Neste trabalho foi realizado o desenvolvimento de um aplicativo para acesso ao ambiente virtual de aprendizagem Moodle, o MobiMoodle. Para tanto foi utilizada uma arquitetura cliente/servidor, sendo utilizada assim as tecnologias J2ME e PHP para o desenvolvimento do cliente e do servidor respectivamente. O MobiMoodle, possibilitou o acesso ao ambiente Moodle e a utilização dos recursos do mesmo.

Palavras-chave: sistemas móveis, Moodle, dispositivos móveis.

MOBILE APPLICATION DEVELOPMENT FOR MOODLE VIRTUAL LEARNING ENVIRONMENT ACCESS (MOBIMOODLE).

ABSTRACT

This work presents the development of an application for access to the virtual learning environment Moodle, the MobiMoodle. The application was developed on client/server architecture and it has made use of the technologies J2ME and PHP for developing respectively the client and the server. The MobiMoodle made possible the access to the Moodle environment and it resources use.

Key-Words: mobile systems, Moodle, mobile devices

SUMÁRIO

SUMÁRIO	V
LISTA DE FIGURAS	VII
1. INTRODUÇÃO	1
1.1. ESTRUTURA DO TRABALHO	2
2. REFERENCIAL TEÓRICO	3
2.1. AMBIENTES VIRTUAIS DE APRENDIZAGEM	3
2.1.1. O Ambiente Moodle	4
2.1.1.2. Características Gerais	5
2.1.1.3. Tipos de Usuário	6
2.1.1.4. Arquitetura do Ambiente.....	6
2.2. A LINGUAGEM PHP.....	7
2.3. MOBILIDADE	8
2.4. APRENDIZADO MÓVEL (<i>MOBILE LEARNING</i>)	9
2.5. DISPOSITIVOS MÓVEIS.....	10
2.6. DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS.....	12
2.7. TECNOLOGIAS PARA DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS	13
2.7.1. A Tecnologia J2ME	14
2.7.1.1. Característica Gerais.....	14
2.7.1.2. A API kXML.....	18
3. METODOLOGIA	19
3.1. TIPO DE PESQUISA	19
3.2. PROCEDIMENTOS	19
3.2.1. Levantamento e Análise de Requisitos	19
3.2.2. Modelagem do Sistema.....	20
3.2.3. Desenvolvimento do Sistema.....	21
3.2.3.1. Servidor.....	21
3.2.3.2. Cliente	23
4. RESULTADO E DISCUSSÃO	24
4.1. CARACTERÍSTICAS DO SISTEMA.....	24
4.2. VISÃO GERAL E DECISÕES DE DESENVOLVIMENTO	24
4.3. MOBIMOODLE: DEFINIÇÃO DO SERVIDOR.....	27
4.4. MOBIMOODLE: LOGIN.....	27
4.5. MOBIMOODLE: MEUS CURSOS E MENU PRINCIPAL.....	29

4.6. MOBI MOODLE: PARTICIPANTES DO CURSO	30
4.7. MOBI MOODLE: TAREFAS DO CURSO	32
4.8. MOBI MOODLE: TABELA DE NOTAS DO ALUNO	33
4.9. CONSIDERAÇÕES FINAIS	34
5. CONCLUSÃO	35
6. SUGESTÃO PARA TRABALHOS FUTUROS.....	36
REFERENCIAL BIBLIOGRÁFICO	37

LISTA DE FIGURAS

FIGURA 2.1 – TELA PRICIPAL DA DISCIPLINA DE PRODUÇÃO DE SOFTWARE I, DO DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO DA UFLA, NO MOODLE.	5
FIGURA 2.2 - ALGUNS MODELOS DE DISPOSITIVOS MÓVEIS COM SUPORTE JAVA.	12
FIGURA 2.3 - CAMADAS DE ARQUITETURA J2ME (FONTE: (MUCHOW, 2004)).	15
FIGURA 2.4 - ARQUITETURA DO MID (FONTE: (MUCHOW, 2004)).....	17
FIGURA 2.5 – HIERARQUIA DA CLASSE <i>DISPLAYABLE</i> (FONTE: MUCHOW(2004)).	18
FIGURA 3.1 - DIAGRAMA DE CASOS DE USO DO SISTEMA MOBI MOODLE.	21
FIGURA 3.2 – FORMAS DE ACESSO AO MOODLE.	22
FIGURA 4.1 - MOBI MOODLE: DIAGRAMA DE FLUXO - MIDP-2.0.....	26
FIGURA 4.2 - MOBI MOODLE: TELA DE DEFINIÇÃO DA URL DO SERVIDOR VISTA DE UM EMULADOR DE TELEFONE CELULAR.....	27
FIGURA 4.3 - MOBI MOODLE: TELA DE <i>LOGIN</i> , VISTA DE UM EMULADOR DE TELEFONE CELULAR.....	28
FIGURA 4.4 - MOBI MOODLE: TELAS PARA DIGITAÇÃO DO NOME DE USUÁRIO E SENHA, VISTAS DE UM EMULADOR DE TELEFONE CELULAR.	28
FIGURA 4.5 - MOBI MOODLE: TELAS DE ERRO DE <i>NOME DE USUÁRIO</i> E OU <i>SENHA</i> INCORRETOS E NÃO FORNECIDOS, RESPECTIVAMENTE, VISTAS DE UM EMULADOR DE TELEFONE CELULAR.	29
FIGURA 4.6 - MOBI MOODLE: TELA “MEUS CURSOS”, QUE LISTA OS CURSOS DO USUÁRIO, VISTA DE UM EMULADOR DE TELEFONE CELULAR.	30
FIGURA 4.7. MOBI MOODLE: TELA PRINCIPAL DO CURSO, VISTA DE UM EMULADOR DE TELEFONE CELULAR. ..	30
FIGURA 4.8 - MOBI MOODLE: LISTA DE PARTICIPANTES DO CURSO, VISTA DE UM EMULADOR DE TELEFONE CELULAR.....	31
FIGURA 4.9 - MOBI MOODLE: TELA DE VISUALIZAÇÃO DOS DADOS DETALHADOS DOS PARTICIPANTES, VISTA DE UM EMULADOR DE TELEFONE CELULAR.	31
FIGURA 4.10. MOBI MOODLE: TELA COM A LISTA DE TAREFAS DO CURSO, VISTA DE UM EMULADOR DE TELEFONE CELULAR.....	32
FIGURA 4.11 - MOBI MOODLE: TELA DE EXIBIÇÃO DAS ESPECIFICAÇÕES DETALHADAS DA TAREFA, VISTA DE UM EMULADOR DE TELEFONE CELULAR.	33
FIGURA 4.12. MOBI MOODLE: TELA DE EXIBIÇÃO DE NOTAS, VISTA DE UM EMULADOR DE TELEFONE CELULAR.	33

1. INTRODUÇÃO

A Educação a Distância (EaD) tem sido uma alternativa para solucionar as exigências sociais e pedagógicas, contando com o apoio dos avanços das novas tecnologias da informação e da comunicação. A EaD passou a ocupar uma posição estratégica para satisfazer as amplas e diversificadas necessidades de qualificação das pessoas adultas procurando atender às demandas crescentes por formação e atualização de conhecimentos e práticas profissionais.

Com isso, tem havido um grande crescimento na educação a distância pela internet, com o apoio de ambientes virtuais de aprendizagem (AVA), o que tem facilitado em muito a EaD, tentando reduzir ao máximo as distâncias entre docentes e discentes, trazendo um contato e aprendizado mais constantes. Entre os mais utilizados, o Ambiente de Aprendizagem Dinâmico Modular Orientado a Objeto (Moodle) tem se destacado, sendo utilizado atualmente no mundo todo por universidades, escolas de ensino médio e fundamental, empresas e diversos outros tipos de instituições. De acordo com as estatísticas do site oficial do Moodle, em novembro de 2007, somavam-se 336 mil usuários registrados em todo o mundo, um crescimento de 14,3% em relação a novembro de 2006 (MOODLE, 2007).

Por outro lado, a sociedade atual passa por uma urgência cada vez maior de conhecimentos. O homem sente a necessidade de poder realizar ações com a possibilidade de se locomover, tendo um melhor aproveitamento do seu tempo, em momentos desperdiçados até então, como nos meios de transporte e até mesmo em filas de repartições públicas e estabelecimentos bancários.

A mobilidade do usuário requer novos modelos de aplicações que aproveitem o poder computacional espalhado em toda a rede, não residente em um dispositivo que tem a capacidade esporádica de comunicação. Esta é a essência da computação pervasiva (*pervasive computing*), onde as aplicações do usuário estão sempre disponíveis, em qualquer lugar onde este se encontre, mantendo o acesso à rede e ao seu ambiente computacional. Deste modo os aparelhos móveis vêm se tornando uma ferramenta indispensável ao homem, principalmente os telefones celulares segundo dados disponibilizados pela Agência Nacional de Telecomunicações (ANATEL), que mostra que ao final de outubro de 2007 o Brasil apresentava 114.69 milhões de aparelhos celulares habilitados, aproximadamente 60 aparelhos para cada 100 habitantes, um crescimento de

18,68% em relação a outubro de 2006 (ANATEL, 2007). Estes dados mostram que grande parte da população já possui aparelhos celulares e que já usa estes, além de efetuar ligações, também para realizar suas atividades do dia-a-dia ou mesmo de lazer. Têm-se assim, a grande necessidade de se ter em celulares, *palms* e *pdas*, *softwares* antes projetados apenas para computadores.

Neste contexto, percebeu-se a possibilidade de se unir o Moodle, como ambiente virtual de aprendizagem e, dispositivos móveis, permitindo assim um aprendizado constante em horários do dia desperdiçados até então, como também, permitindo as pessoas que não possuem computador ou acesso a internet em suas residências, a possibilidade de realizar cursos, acessando o Moodle via aparelhos celulares, que são mais acessíveis a população.

O presente trabalho tem como objetivo a implementação do MobiMoodle, um aplicativo para telefones celular, que permitirá o acesso a recursos do ambiente virtual de aprendizagem Moodle. O aplicativo será desenvolvido utilizando as tecnologias PHP e J2ME.

1.1. Estrutura do Trabalho

O presente trabalho está estruturado da seguinte forma:

O Capítulo 2 apresenta um levantamento dos pontos chave para o entendimento dos principais conceitos abordados no trabalho através de uma revisão da literatura. Neste capítulo, explana-se sobre Ambientes Virtuais de Aprendizagem, apresentando características do ambiente virtual de aprendizagem Moodle, que são cruciais para o entendimento do trabalho. Em seguida são apresentadas características do PHP, que será uma das tecnologias utilizadas no neste trabalho, e também conceitos de mobilidade e do aprendizado móvel. Além disso, são citadas algumas características de dispositivos móveis e seus aplicativos. Após um entendimento dos dispositivos móveis abordamos sobre o desenvolvimento de aplicações para dispositivos móveis, passando então às tecnologias para desenvolvimento de aplicações móveis, abordando mais profundamente a tecnologia J2ME e suas características.

O Capítulo 3 apresenta a Metodologia utilizada no desenvolvimento do trabalho.

O Capítulo 4 apresenta os Resultados e Discussão do trabalho, apresentando assim o aplicativo desenvolvido, suas características e as decisões de desenvolvimento.

O Capítulo 5 apresenta a Conclusão do trabalho.

2. REFERENCIAL TEÓRICO

2.1. Ambientes Virtuais de Aprendizagem

A Educação a Distância, EaD, é um método não presencial em educação que pressupõe uma forma de ensino/aprendizagem, onde professores e alunos não estão em contato físico nem, necessariamente, interagindo ao mesmo tempo (KRAEMER e MARTINS, 2004).

Segundo Palácio (2002) citado por Kraemer e Martins (2004), as estratégias disponíveis que viabilizam a EaD são inúmeras, tais como: Correspondência, telefone, televisão, rádio, dentre outras. Todas estas podem ser consideradas não presenciais e são utilizadas isoladamente ou integradas entre si. No entanto, o que possibilitou a legalização deste meio de ensino foram os recursos de acesso e interação à informação que temos a nossa disposição no contexto atual. Isso pode ser atribuído à Internet que é um veículo de comunicação de massa que proporciona interação entre aluno e professor em tempo real, integrando todos os recursos que já existiam numa só ferramenta, os ambientes virtuais de aprendizagem.

Ambientes Virtuais de Aprendizagem são sistemas que utilizam um protocolo de rede para realizar a comunicação e as atividades de aprendizagem de forma que os alunos adquiram o mesmo conhecimento que seriam obtidos no ensino presencial (NETO & BRASILEIRO, 2002 *apud* RIBEIRO, 2007).

Os ambientes virtuais de aprendizagem (AVA), também conhecidos como *Virtual Learning Environment* (VLE), *Learning Management System* (LMS) ou *Course Management System* (CMS), são softwares desenvolvidos sobre uma metodologia pedagógica para auxiliar a promoção de ensino e aprendizagem virtual ou semipresencial (WIKIPEDIA, 2008).

Trata-se, portanto, de uma abstração de uma sala de aula tradicional em um meio virtual através de uma rede de computadores como a Internet. Esta abstração, no entanto, nem sempre se dá em todos os níveis possíveis, sendo restrita às limitações das mídias digitais, do hipertexto e das linguagens de programação voltadas ao desenvolvimento de sistemas Web. Além disso, estas tecnologias costumam levar em consideração as peculiaridades que o meio digital oferece, as quais são exploradas para oferecer ambientes mais atrativos aos usuários, ou seja, ambientes inclusivos (RIBEIRO, 2007).

Alguns exemplos de ambientes virtuais de aprendizagem disponíveis no mercado

são o *Modular Object-Oriented Dynamic Learning Environment* (Moodle), o Sistema Online de Aprendizagem (SOLAR), o TelEduc, o Agentes Micromundos e Análise do Desenvolvimento no Uso de Instrumentos (Amadeus), dentre outros, sendo foco deste trabalho o Moodle, melhor descrito na sessão seguinte.

2.1.1. O Ambiente Moodle

A palavra Moodle é um acrônimo para *Modular Object-Oriented Dynamic Learning Environment* (Ambiente de Aprendizagem Dinâmico Modular Orientado a Objeto), o que significa que o Moodle possui uma arquitetura modular facilitando a sua customização por programadores e adaptação do mesmo de acordo com a necessidade do usuário, também é dinâmico e flexível, facilitando para o professor construir o seu próprio modelo pedagógico (MOODLE, 2008).

É um pacote de software livre para gestão da aprendizagem e de trabalho colaborativo, que permite a criação de cursos online com páginas de conteúdo das disciplinas, grupos de trabalho e comunidades de aprendizagem, ou seja, um ambiente virtual de aprendizagem (MOODLE, 2008).

Segundo o site oficial, ele foi desenvolvido pelo australiano Martin Dougiamas que iniciou seu desenvolvimento nos anos 1990s e, atualmente, continua a ser desenvolvido por seus colaboradores das comunidades de software livre em várias partes do mundo (MOODLE, 2008).

De acordo com as estatísticas do site oficial do moodle, existem atualmente 35829 sites registrados com o sistema, com mais de 350 mil usuários (MOODLE, 2007).

O Moodle é fornecido gratuitamente como software *Open Source* (sob a GNU Public License). Basicamente, isto significa que o Moodle é protegido por direito autoral, mas oferece outras permissões. Todos estão autorizados a copiar, modificar e usar Moodle desde que concorde com: '*fornecer o código-fonte para outros; não modificar ou remover a licença original e os direitos autorais*', e '*aplicar esta mesma licença para qualquer trabalho derivativo*' (MOODLE, 2008).

Observa-se na figura 2.1 a imagem da tela principal de uma sala virtual do Moodle, da disciplina de Produção de Software I, do departamento de Ciência de Computação, da UFLA.

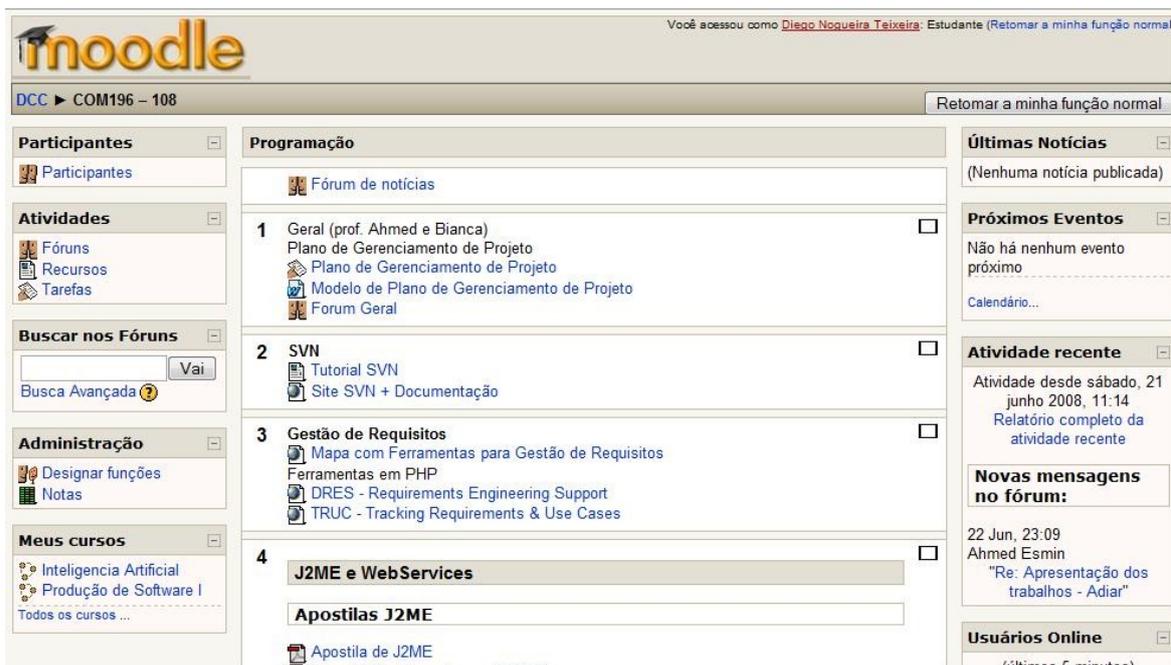


Figura 2.1 – Tela principal da disciplina de Produção de Software I, do Departamento de Ciência da Computação da UFLA, no Moodle.

2.1.1.2. Características Gerais

O Moodle funciona em diversos sistemas operacionais (SO), desde que disponham de servidores *Web* com suporte à linguagem PHP e a um SGBD (preferencialmente MySQL ou PostgreSQL). É um sistema desenvolvido na linguagem PHP sob um modelo de desenvolvimento onde comunidades de *software* livre contribuem excessivamente com o seu aprimoramento, garantindo uma vasta gama de módulos e de idiomas disponíveis para o *software* (RIBEIRO, 2007).

De acordo com Ribeiro (2007), o Moodle é caracterizado por sua divisão em blocos e módulos, assim como é feito na maioria dos portais para *Web*, também conhecidos como *Content Management System*, porém, é enquadrado na categoria *Learning Management System*, devido à sua aplicabilidade.

Através desta arquitetura, novos módulos podem ser desenvolvidos de forma independente, disponibilizados e utilizados de acordo com necessidades próprias de cada usuário. Os módulos do Moodle podem ser descritos como subsistemas responsáveis por alguma atividade virtual ou pelo controle administrativo. Os blocos são estruturas programáveis que servem para agrupar *links* para módulos e/ou apresentar informações aos usuários (RIBEIRO, 2007).

2.1.1.3. Tipos de Usuário

Sobre os tipos de usuários, segundo Ribeiro (2007), o ambiente divide-se basicamente entre:

- Visitantes: usuários sem vínculo com o sistema e que possui de pouca a nenhuma visibilidade do ambiente. É uma categoria utilizada para que usuários possam se ambientar ou conhecer um pouco mais de um ambiente e os cursos que este oferece antes de se cadastrar efetivamente;
- Alunos: usuários com permissões restritas a realização das atividades, interações entre os outros usuários e configurações pessoais;
- Tutores: usuários com permissões restritas a criação de atividades dentro de um curso ou sala virtual e, em geral, possui os mesmos privilégios dos Alunos;
- Autores de cursos: são usuários com permissão para adicionar novos cursos;
- Administradores: usuários com permissões para realizar qualquer tipo de operações e acessar qualquer arquivo do *site*.

A hierarquia entre Administradores, Tutores, Alunos e Visitantes possui um caráter complementar. Na maioria dos casos, os alunos têm as permissões dos visitantes, os tutores têm as permissões dos alunos e os administradores têm as permissões dos tutores (RIBEIRO, 2007).

2.1.1.4. Arquitetura do Ambiente

Segundo Ribeiro (2007), o Ambiente Moodle pode ser dividido em duas grandes partes:

- De caráter geral, ou estrutural: que está relacionada com a estrutura básica para suportar os recursos que o ambiente oferece. Estão inclusos nesta categoria as bibliotecas de funções de uso geral, as ferramentas para o controle de cursos, de usuários e visões, de autenticação (*log-in*) no sistema, de eventos e de geração de *backup*, enfim, a estrutura comum às implantações do ambiente em um computador servidor.
- De caráter específico: que está relacionada com os módulos e os blocos auxiliares. Estes podem ser ativados ou desativados de acordo com as necessidades específicas de utilização do ambiente. Alguns módulos de maior utilização são

inclusos no pacote de instalação do Moodle. Outros, por serem mais específicos, são disponibilizados separadamente.

Quanto a sua estrutura interna, seu código-fonte é desenvolvido em camadas e é predominantemente usado o paradigma procedural nas camadas mais altas (nos módulos, por exemplo), havendo a utilização do paradigma orientado a objetos em alguns níveis mais baixos e específicos do sistema.

2.2. A Linguagem PHP

Segundo Melo e Nascimento (2007), PHP é a abreviação de *Hypertext Preprocessor* ou em tradução livre “Pré-processador de Hipertexto”.

De acordo com Dall'Oglio (2007), a linguagem de programação PHP foi criada no outono de 1994 por Rasmus Lerdorf, sendo inicialmente formada por um conjunto de *scripts* voltados à criação de páginas dinâmicas que Rasmus utilizava para monitorar o acesso ao seu currículo na internet. A sigla inicialmente significava “*Personal Home Pages*” que, apenas em 1998, em sua terceira versão, desenvolvida com a colaboração de Andi Gutmans e Zeev Suraski, teve seu significado alterado para “PHP: *Hypertext Preprocessor*” a fim de retratar a nova realidade de uma linguagem com propósitos mais amplos.

O PHP é uma linguagem de *script* de código aberto que tem como objetivo primário a geração de conteúdo dinâmico para páginas da internet, o que quer dizer que ao invés de criar um programa para gerar e imprimir *HyperText Markup Language* (HTML), você pode escrever HTML com o código PHP embutido para gerar conteúdo dinâmico (MELO e NASCIMENTO, 2007).

Serrão *et. al* (2001) descreve que PHP é uma poderosa ferramenta que surgiu da necessidade de os programadores *web* utilizarem recursos capazes de permitir a interatividade com os utilizadores e gerarem conteúdo dinâmico.

Melo e Nascimento (2007) ressalta como uma das vantagens do PHP, ser executado no lado do servidor. Seu código-fonte não é exibido ao internauta, que apenas terá acesso ao seu HTML resultante.

Várias outras vantagens além desta acima são citadas por Melo e Nascimento (2007), como:

- Custo: o *PHP* não tem custo de licença justamente por ser um *software* livre. Seu modelo de licenciamento é o da *GPL – General Public License* (Licença Pública Geral), largamente utilizado em *softwares* livres.
- Manipulação de arquivos como texto plano, do tipo PDF, documentos DOC, entre outros, incluindo sua criação, exclusão, alteração etc.
- Geração de imagens dinâmicas para validação de formulários.
- Criptografia de dados.
- Definição de *cookies* e sessões.
- Definição de interfaces para *webservices*.
- Manipulação de arquivos XML.
- Suporte a vários bancos com acesso nativo como: DB2, Firebird, Informix,
- Interbase, MySQL, Oracle, PostgreSQL, SQL Server etc.
- Suporte a vários padrões e interfaces como: COM, POP3, IMAP etc.
- Independência de plataforma: roda em Windows, Linux, Unix, Mac etc.
- Suporte a Orientação a Objetos consistente na versão 5.
- Curva de aprendizado reduzida para quem está aprendendo a linguagem.
- Sintaxe semelhante com a de linguagens consagradas como C e Java.
- Integração com vários servidores *web* como Apache, IIS, Xitami, entre outros.
- Servidores de hospedagem tanto gratuitos como pagos em grande quantidade no Brasil e no mundo.
- Documentação oficial em português.
- Possibilidade do desenvolvimento também para *desktop*, usando a mesma linguagem.

Além de ter todas as vantagens citadas acima, o PHP foi adotado no presente trabalho, principalmente por já ser a plataforma utilizada no desenvolvimento do Moodle. Isso possibilita uma implantação e integração mais prática e fácil do sistema MobiMoodle, em um servidor que já possui o Moodle instalado.

2.3. Mobilidade

Considerando que este trabalho tem por objetivo o desenvolvimento de um aplicativo móvel, o MobiMoodle, e que este é voltado para usuários que realmente

necessitarão de acessá-lo dos mais diversos locais, é importante também definir os conceitos de mobilidade que são empregados no presente trabalho.

Segundo Lago (2007), as seguintes definições podem ser empregadas em mobilidade:

- Cobucci (2007) citado por Lago (2007) afirma que de forma mais abrangente, mobilidade é a propriedade do que é móvel ou do que obedece às leis do movimento, inconstância e dinamismo;

- Segundo Netto (2006) citado por Lago (2007), mobilidade é o termo utilizado para identificar dispositivos que podem ser operados à distância ou sem fio. Dispositivos que podem variar desde um simples *bip* até os mais modernos *Pockets*;

- Já MeioBit (2006) citado por Lago (2007), afirma que mobilidade é a facilidade de acessar recursos de informática (entrada de dados, *web*, e-mail, consulta a dados) fora do ambiente principal de trabalho, podendo ou não estar em movimento. Também, deve ser considerado o uso do equipamento sem precisar de uma fonte de energia por pelo menos quatro horas de uso e poder usá-lo em pé;

- De acordo com MobileLife (2005) citado por Lago (2007), mobilidade pode ser definida como a capacidade de poder se deslocar ou ser deslocado facilmente. No contexto da computação móvel, mobilidade se refere ao uso pelas pessoas de dispositivos móveis portáteis funcionalmente poderosos que ofereçam a capacidade de realizar facilmente um conjunto de funções de aplicação, sendo também capazes de conectar-se, obter dados e fornecê-los a outros usuários, aplicações e sistemas.

2.4. Aprendizado Móvel (*Mobile Learning*)

Dadas as atuais necessidades de locomoção dos profissionais e a crescente necessidade de sua capacitação, torna-se necessário criar mecanismos que possibilitem ao estudante continuar a aprender mesmo estando fora da instituição de ensino.

Aliando-se os dispositivos computacionais com a comunicação móvel celular obtém-se a computação móvel, que permite a um estudante acessar conteúdos e interagir com professores e outros estudantes a partir de praticamente qualquer lugar. Nestas condições tem-se o denominado “*mobile learning (m-learning)*”, ou seja, Aprendizado Móvel (PELISSOLI e LOYOLLA, 2004).

Segundo Nyiri (2002), o paradigma *mobile learning* ou *m-learning* surge

aproveitando-se da disponibilidade de dispositivos móveis e considerando-se as necessidades específicas de educação e treinamento. Acontece quando a interação entre os participantes se dá através de dispositivos móveis, tais como celulares, *i-pods*, *laptops*, entre outros.

O *m-learning* é caracterizado pelo aprendizado em movimento ou aquele que, mesmo parado, utiliza dispositivos móveis como ferramenta de apoio (WIKIPEDIA, 2008). Seu grande potencial encontra-se na utilização da tecnologia móvel como parte de um modelo de aprendizado integrado, caracterizado pelo uso de dispositivos de comunicação sem fio, de forma transparente e com alto grau de mobilidade (AHONEN *et. al.*, 2003).

O *m-learning* é a fusão de diversas tecnologias de processamento e comunicação de dados que permite ao grupo de estudantes e aos professores uma maior interação. Basicamente, o *m-learning* faz uso das tecnologias de redes sem fio, dos novos recursos fornecidos pela telefonia celular, da linguagem XML, da linguagem JAVA, da linguagem WAP, dos serviços de correio de voz, serviços de mensagens curtas (SMS), da capacidade de transmissão de fotos, serviços de e-mail, *multimedia message service* (MMS) e provavelmente em pouco tempo estará disponível o uso de vídeo sob demanda (PELLISSOLI e LOYOLLA, 2004).

Essa oferta de serviços de telecomunicações e de artefatos computacionais, capazes de prover mobilidade aos diferentes participantes de projetos educacionais, apresenta a oportunidade para o desenvolvimento de pesquisas no campo da computação móvel aplicadas à educação (MEIRELLES *et al.*, 2004).

Dispositivos de comunicação sem fio oferecem uma extensão natural da educação a distância via computadores, pois contribuem para a facilidade de acesso ao aprendizado, por exemplo, na obtenção de conteúdo específico para um determinado assunto, sem hora e local pré-estabelecidos. Portanto, *mobile learning*, ou aprendizagem móvel é uma das derivações da Educação a distância, ou *e-learning* (LEHNER *et. al.*, 2002).

2.5. Dispositivos Móveis

White e Hemphill (2002) definem dispositivos móveis como aqueles dispositivos computacionais que são pequenos o suficiente para que possam ser facilmente utilizados enquanto são transportados. Eles fornecem ao usuário uma porção da capacidade de

computação e informação disponíveis em sistemas de informação fixos em casa ou no local de trabalho. Em geral, a maioria permite ser sincronizados com os sistemas fixos para atualização de *softwares* e dados.

Segundo Johnson (2007), existe diversas categorias de dispositivos que podem ser considerados móveis. São elas:

- A dos *laptops* (ou *notebooks*), que são computadores portáteis, com capacidade computacional equivalente a um *desktop*.
- A dos PDAs, com menor capacidade computacional. Estes possuem telas pequenas, mas seu poder de processamento é bem alto, se comparado com um telefone celular, mas baixo, se comparado aos *notebooks*. Têm suporte a aplicativos desenvolvidos com linguagens de programação de alto nível, recursos multimídia, acesso à rede etc.
- E por fim a de telefones celulares. Antigamente, tinham telas minúsculas e quase nenhum recurso. Hoje, contudo, existem aparelhos com memória expansível, acesso à rede *Bluetooth*, suporte a Java etc.

Os dispositivos móveis mais populares atualmente são os telefones celulares. Hoje, têm-se celulares que além de sua função básica (ligações), enviam SMS e e-mail, tocam músicas no formato mp3, tiram fotos, inclui agenda, calculadora, *chat*, entre outros (BARROS, 2005).

A figura 2.2 mostra uma vasta quantidade de modelos de telefones celular e PDAs com suporte a Java.



Figura 2.2 - Alguns modelos de dispositivos móveis com suporte Java.

2.6. Desenvolvimento de Aplicações Móveis

De acordo com Forman e Zahorjan (1994) citado por Santos (2007), a popularidade dos dispositivos móveis produz uma demanda de desenvolvimento de aplicações específicas, porém, os dispositivos móveis têm características diferentes daquelas encontradas em computadores pessoais. Estes dispositivos têm uma CPU com menor poder de processamento, memórias de menor capacidade e fonte de energia de baixa duração, em geral baterias ou pilhas, o que faz com que o uso destes dispositivos seja restrito. A tela tem tamanho bastante reduzido e mecanismos de entrada de dados diferentes, que restringem o tamanho e a interatividade da interface com o usuário. Além disto, a conexão de rede tem uma largura de banda reduzida, latência e taxa de erros maiores quando comparados com redes cabeadas. Este conjunto de características faz com que o desenvolvimento de aplicações para dispositivos móveis não seja feito da mesma forma e nem com as mesmas ferramentas utilizadas em aplicações para computadores pessoais.

Segundo Johnson (2007), as limitações impostas para desenvolver aplicativos para dispositivos de poucos recursos computacionais requerem alguns cuidados especiais, como:

- Ter em vista o tamanho reduzido das telas ao projetar a aplicação, o que demanda um bom planejamento de como dispor os elementos nas telas e dividi-los em diversas telas;

- Projetar a aplicação para que seja bem eficiente em termos de recursos, usando somente o necessário, pois deixar o processador estressado pode resultar no travamento da interface;

De acordo com Lee *et al* (2005) a arquitetura para aplicações móveis utiliza camadas e filas. A divisão em camadas descreve a divisão do código da aplicação em uma única máquina.

A camada de código que interage mais perto do usuário é comumente chamada de camada de apresentação. A segunda camada é freqüentemente denominada camada de negócios, pois em geral ele trata da lógica de negócios. A terceira camada é freqüentemente denominada camada de acesso a dados. Em geral, ela trata da comunicação com o banco de dados ou com a origem de dados (SANTOS, 2007).

As filas descrevem a divisão de trabalho de código da aplicação em diversas máquinas. A segmentação em filas geralmente envolve a colocação de módulos de código em máquinas diferentes num ambiente de servidores distribuídos. Se o código da aplicação já estiver em camadas, o processo de segmentação em filas ficará muito mais simples. As filas descrevem arquiteturas de servidor, e em geral não contamos dispositivos de cliente como uma fila. Embora isso seja possível, não é uma convenção comum (SANTOS, 2007).

As aplicações móveis devem ser projetadas como qualquer outra aplicação corporativa complexa. Deve-se desenvolver uma arquitetura do projeto, um conjunto de casos de uso, fluxogramas, modelos de projeto, modelos de dados e todos outros artefatos geralmente associados com o projeto técnico. É extremamente necessário escolher tecnologias e linguagens antes de os requisitos e o projeto terem sido completados (SANTOS, 2007).

2.7. Tecnologias Para Desenvolvimento de Aplicações Móveis

Existem várias linguagens para desenvolvimento de aplicações para sistemas móveis, tais como, *Java 2 Micro Edition (J2ME)*, *Binary Runtime Environment for Wireless (BREW)*, *Wireless Application Protocol (WAP)*, e *i-mode*.

Será apresentada em mais detalhes a linguagem J2ME, a qual foi escolhida para o desenvolvimento deste trabalho.

2.7.1. A Tecnologia J2ME

Segundo Lima *et. al.* (2007), citado por Lago (2007), a plataforma *Java* vem se desenvolvendo continuamente, oferecendo suporte a uma crescente variedade de tipos de aplicativos. Como consequência, houve um grande aumento do seu tamanho, o que passou a conflitar com o objetivo do *write once, run anywhere* (WORA), uma vez que nem todo dispositivo oferecia suporte a processos que demandavam uma grande quantidade de memória. Como uma primeira tentativa de contornar o problema, a Sun introduziu, em 1997, uma variedade de plataformas *Java*, cada uma voltada a um segmento do mercado, sendo que a principal diferença entre elas era o subconjunto da *Application Programming Interface* (API) do *Java* a que ofereciam suporte.

Conforme a tecnologia foi se desenvolvendo, sentiu-se a necessidade de se unificar as plataformas orientadas a dispositivo e em 1999 surgiu o *Java 2 Micro Edition*, que é uma arquitetura que permite o uso da tecnologia *Java* em dispositivos com diversas características e funcionalidades (LAGO, 2007).

De acordo com Muchow (2004), a plataforma J2ME é a edição da linguagem *Java* projetada para dispositivos com memória, vídeo e poder de processamento limitados, destinado diretamente aos dispositivos consumidores com poder limitado, como celulares e *paggers*. Muitos destes dispositivos não possuem opção de *download* e instalação de *software* além daqueles configurados na sua fabricação.

Antes do surgimento da tecnologia J2ME as aplicações tinham que ser escritas na linguagem nativa de cada dispositivo usando bibliotecas proprietárias, o que as tornavam incompatíveis com dispositivos diferentes (SANTOS, 2007).

2.7.1.1. Característica Gerais

Segundo a Sun (2007), a arquitetura J2ME é modular e escalável. Esta modularidade e escalabilidade é definida pela tecnologia J2ME em um modelo de 3 camadas embutidas sobre o sistema operacional do dispositivo: a camada de perfil (*Profile*), a camada de configuração (*Configuration*) e a camada representada pela máquina virtual *Java* (*Java Virtual Machine*), juntamente com o sistema operacional presente no dispositivo (*Host Operation System*). A aplicação encontra-se logo acima da camada de perfil. Estas três camadas podem ser observadas na figura 2.3.

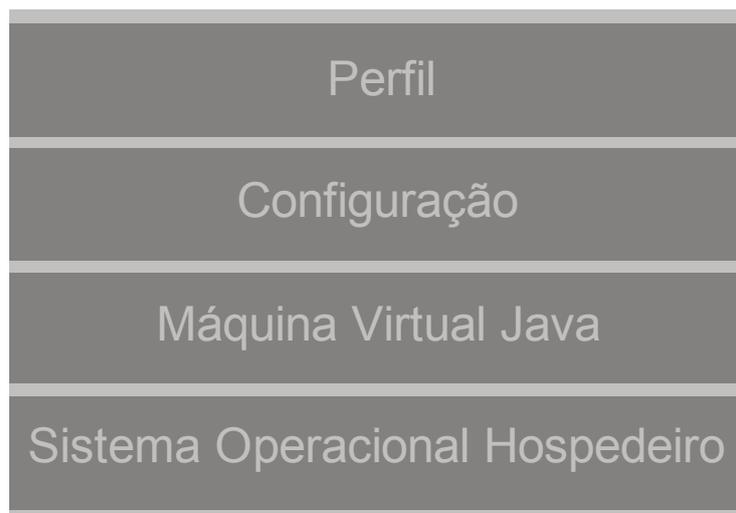


Figura 2.3 - Camadas de arquitetura J2ME (Fonte: (MUCHOW, 2004)).

A máquina virtual está posicionada acima do sistema operacional do dispositivo, pois é a máquina virtual a responsável por definir quais as limitações de execução das aplicações nos dispositivos. A camada de configuração define o nível de funcionalidades e serviços oferecidos pela máquina virtual. Por fim, o perfil é um conjunto de bibliotecas específico para uma determinada configuração, mas uma configuração pode suportar vários perfis (SILVA, 2006).

Segundo Pinheiro (2003) citado por Silva (2006), um perfil funciona como o complemento da configuração, em que esta define o mínimo de funcionalidades e o perfil atende detalhes específicos do dispositivo. Ou seja, a configuração descreve de forma geral uma família de dispositivos, enquanto o perfil fica mais específico para um tipo particular de dispositivo em uma dada família (Kuhnen, 2003). Os perfis existentes para a CLDC são diferentes dos existentes para CDC. Os perfis podem ser classificados como:

- FP: o *Foundation Profile* complementa a configuração do CDC, sendo o nível mais baixo da mesma, fornecendo base para dispositivos em rede sem interface com o usuário.
- PP: o *Personal Profile* é o perfil CDC utilizado em dispositivos que necessitam de um suporte completo para interface com o usuário. O PP substitui a antiga tecnologia *Personal Java*, que é uma versão reduzida da J2SE, destinada a rodar em dispositivos de pouca memória e baixo poder de processamento.
- PBP: o *Personal Basis Profile* é uma divisão do PP que fornece um ambiente para dispositivos conectados, que suportem um nível básico de apresentação

gráfica.

- MIDP: o *Mobile Information Device Profile* ou Perfil de Dispositivo de Informação Móvel é um perfil direcionado para dispositivos móveis ligado ao CLDC, fornecendo um ambiente completo para o desenvolvimento de aplicações para dispositivos móveis. Ele define funcionalidades como interface do usuário, suporte à rede, métodos de entrada dados, manipulação da tela do dispositivo (*display*), armazenamento persistente, entre outros.

A camada de configuração é classificada de acordo com as capacidades de cada dispositivo. Segundo Muchow (2004), as configurações estão divididas em: *Connected Device Configuration* (CDC), que é um conjunto de APIs para dispositivos “fixos”, como um computador ligado à televisão; *Connected Limited Device Configuration* (CLDC), que é um conjunto de APIs destinada a dispositivos com poder de processamento, vídeo e memória limitados, os quais geralmente são móveis.

Ainda segundo Muchow (2004), uma configuração está intimamente ligada a uma máquina virtual Java (VM), a qual poderá assumir duas formas:

- Para sistemas compatíveis com a CDC, ela será a máquina virtual “tradicional”, JVM, tendo o mesmo conjunto de recursos do J2SE (Java 2 Standard Edition).
- Para sistemas compatíveis com a CLDC, ela será a máquina virtual KVM, ou uma máquina virtual que atenda às especificações exigidas pela CLDC e que leva em consideração os recursos limitados disponíveis nos dispositivos que se encaixam nessa configuração.

Segundo Depiné (2002), existem algumas limitações na configuração CLDC em relação à edição J2SE. As principais são:

- Não há suporte para números de ponto flutuante, portanto os tipos *float* e *double* não podem ser usados. Isso ocorre principalmente por causa do *hardware* que não suporta, pois operações com ponto flutuante requerem um maior processamento;
- Sem suporte a finalização (*object.finalize()*);
- Suporte limitado a erros. Existem apenas três classes de erros: *java.lang.Error*, *java.lang.OutOfMemoryError*, e *java.lang.VirtualMachineError*. Erros de *Runtime* são manipulados de maneira dependente de implementação, que

pode terminar a aplicação ou efetuar um *reset* no dispositivo. Exceções de aplicação são utilizadas normalmente, como no J2SE.

- Não há suporte a *Java Native Interface* (JNI), por questões de segurança e performance.

A figura 2.4 mostra a arquitetura do dispositivo móvel de informação (MID). Onde temos a parte do *hardware* (*Mobile Information Device*), logo acima, rodando sobre o *hardware*, temos o sistema operacional (*Native Operating System*). Do lado direito, acima, temos as aplicações nativas do dispositivo (*Native Applications*). Um exemplo destes aplicativos seria os programas de configuração que vêm instalados nos dispositivos. O CLDC é instalado no sistema operacional e é a base para o MIDP. Existem as classes específicas providas pelo fabricante do equipamento (*OEM-Specific Classes*) e as aplicações específicas do fabricante. Estas podem acessar APIs do MIDP e/ou as classes específicas do fabricante.

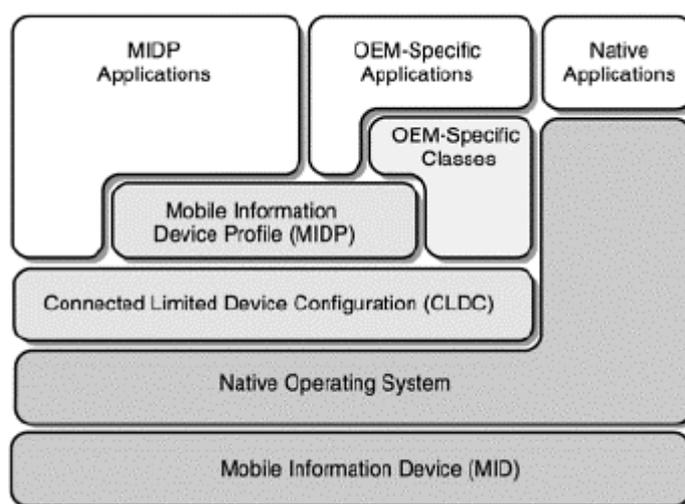


Figura 2.4 - Arquitetura do MID (Fonte: (MUCHOW, 2004))

O MIDP introduziu uma hierarquia de classes específica para interface com o usuário, o pacote *javax.microedition.lcdui*. Assim como o AWT, a aparência de cada componente gráfico depende do dispositivo. A Figura 2.5 descreve a hierarquia das classes desse pacote. Todas são filhas da classe *Displayable*, que representa um objeto que pode ser colocado na tela.

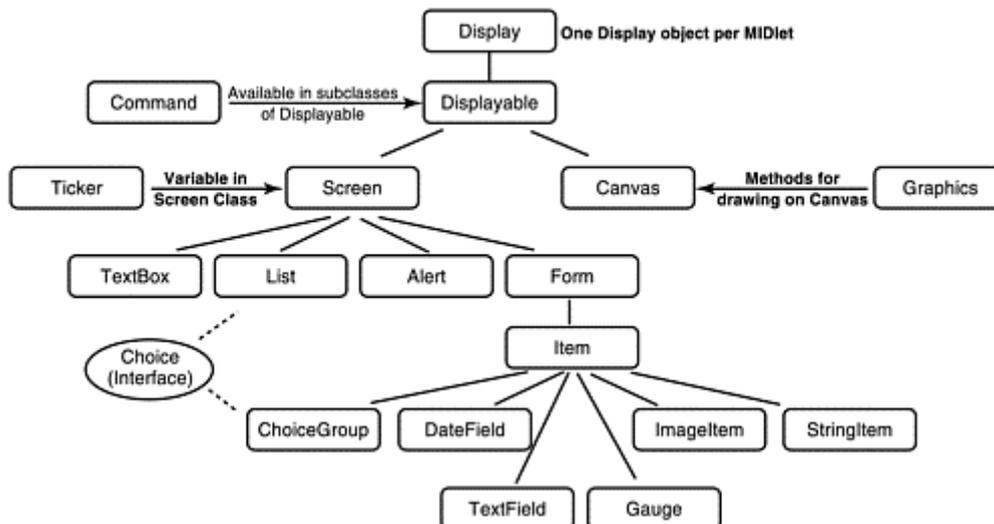


Figura 2.5 – Hierarquia da Classe *Displayable* (Fonte: MUCHOW(2004)).

Além das configurações e perfis o J2ME possui pacotes de APIs opcionais podendo ser ampliada pela combinação de vários pacotes. As APIs são bibliotecas de programação específicas a uma determinada tecnologia e que aumentam as capacidades do ambiente Java, caso estejam implementadas no dispositivo (SUN, 2008).

Exemplos de APIs opcionais são o WMA (*Wireless Messaging API*), que adiciona controle de mídia aos programas J2ME nos dispositivos que o permitem, e o MMAPI (*Mobile Media API*), que permite manipular funcionalidades multimídia dos dispositivos. Estes pacotes são chamados opcionais porque, mesmo que não estejam implementados em uma aplicação, o mesmo continua a ser uma aplicação J2ME.

2.7.1.2. A API kXML

O kXML é um parser de XML estável e otimizado, para ser utilizado em pequenos dispositivos com recursos limitados de memória e processamento, e desta forma sendo ideal para ser utilizado com J2ME (LECHETA, 2006).

Segundo Lecheta (2006), o kXML utiliza uma técnica para realizar o parser de um XML, conhecida como “pull parser”, onde uma pequena quantidade de dados é analisada de cada vez. A aplicação faz o parser do XML solicitando sempre a próxima parte que deve ser processada, isto geralmente é feito em um loop até o fim do XML. Desta forma, a aplicação pode exibir as informações do XML a medida que o parser é efetuado.

3. METODOLOGIA

3.1. Tipo de Pesquisa

Segundo Jung (2004) e observando o método científico tem-se que este projeto é de natureza tecnológica, com objetivos de caráter exploratório, utilizando procedimentos experimentais.

Este trabalho é de natureza aplicada, pois usa os conhecimentos básicos de desenvolvimento de aplicações móveis e do ambiente virtual de aprendizagem Moodle e sua arquitetura, para a geração de um novo produto que seria o MobiMoodle, um sistema móvel para acesso ao ambiente virtual de aprendizagem.

Os objetivos são de caráter exploratório, pois, pretende-se obter um novo produto de *software*, que pode ser caracterizado como uma inovação tecnológica.

Os procedimentos utilizados são experimentais, pois será desenvolvido um protótipo de *software*.

3.2. Procedimentos

Os procedimentos metodológicos utilizados para a realização do presente trabalho foram: pesquisa, modelagem e desenvolvimento de *softwares*. Estes são os objetivos propostos e serão apresentados a seguir.

Inicialmente houve a idealização e fundamentação dos princípios que este projeto deveria seguir. Com a definição destas bases, o primeiro procedimento metodológico efetuado foi a realização de um levantamento bibliográfico, visando definir quais das funcionalidades do ambiente de aprendizagem Moodle seriam desenvolvidas, sendo que ao mesmo tempo, foi realizado um estudo da tecnologia Java usada para desenvolvimento de aplicações móveis, J2ME, a qual foi utilizada para o desenvolvimento do aplicativo.

Após esta fase foi feita uma modelagem inicial do projeto, na qual foram criadas diretrizes básicas para o desenvolvimento do mesmo, posteriormente, foi executado o desenvolvimento do sistema e finalmente a análise dos resultados obtidos com o sistema, fazendo-se testes de sua execução em emuladores de telefones celulares.

3.2.1. Levantamento e Análise de Requisitos

Após o levantamento bibliográfico tendo realizado então um estudo mais

aprofundado da estrutura do ambiente Moodle e levando em consideração os recursos disponíveis para o desenvolvimento deste trabalho, foi definido que seriam desenvolvidos apenas funcionalidades de consulta do ambiente e, que são acessíveis a usuários com perfil de aluno. Sendo assim, restringiu-se o escopo do projeto às funcionalidades abaixo citadas:

- Listar e visualizar participantes do curso;
- Listar e visualizar tarefas do curso;
- Visualizar Notas;

No entanto, para que fossem desenvolvidas estas funcionalidades, inicialmente tiveram que ser desenvolvidas funcionalidades básicas como:

- Selecionar servidor Moodle;
- Efetuar login no ambiente e manter seção através de “cookies”;
- Listar e visualizar “Meus Cursos”.

3.2.2. Modelagem do Sistema

Com base nas informações coletadas durante a idealização inicial do projeto, filtrados pelos objetivos do trabalho e as devidas análises realizadas na fase anterior, deu-se início à modelagem do aplicativo.

Através do levantamento dos requisitos, onde foram definidas as funcionalidades a serem desenvolvidas, foi possível elaborar um diagrama de casos de uso que foi utilizado como fundamento durante todo o desenvolvimento do MobiMoodle.

A Figura 3.1 apresenta o diagrama de casos de uso do aplicativo. Nele, o ator usuário pode: selecionar o servidor Moodle com suporte ao sistema móvel, efetuar login no sistema, que após autenticar o usuário, retorna a lista dos cursos dos quais o usuário participa, visualizar o curso desejado, listar e visualizar as tarefas a serem feitas, listar e visualizar todos os participantes do curso e visualizar as suas notas no curso. Todos os casos de uso que necessitam de dados do servidor Moodle, utilizam-se do caso de uso autenticar usuário, que garante uma maior segurança.

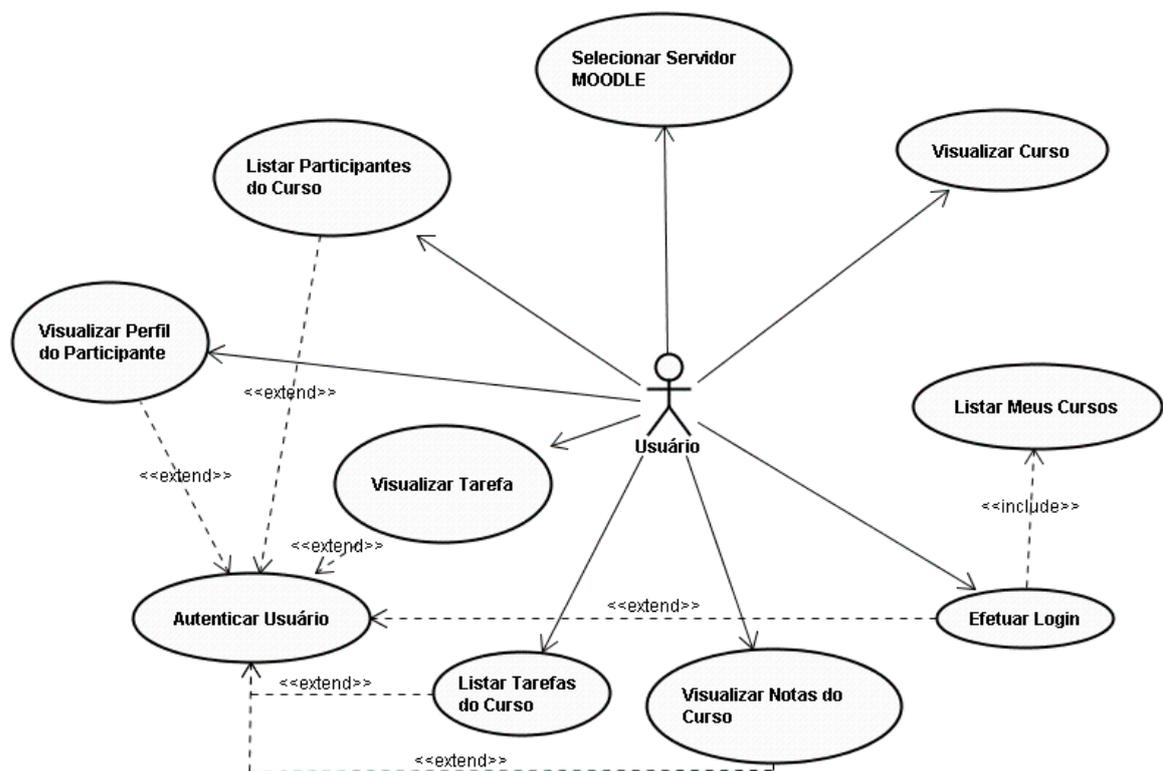


Figura 3.1 - Diagrama de Casos de Uso do Sistema MobiMoodle.

3.2.3. Desenvolvimento do Sistema

Os requisitos foram definidos levando em consideração que este é um sistema que utiliza uma arquitetura cliente/servidor, portanto, existe uma parte desenvolvida para dispositivos móveis (Cliente), na plataforma J2ME e outra parte desenvolvida para o Moodle (servidor), com a utilização da linguagem PHP.

3.2.3.1. Servidor

Para o desenvolvimento da parte do servidor, foi necessária inicialmente a instalação do ambiente Moodle, versão 1.9. Assim que foi instalado o ambiente, definiu-se a utilização da linguagem PHP no desenvolvimento do módulo de execução em servidor, pois esta linguagem já era utilizada no desenvolvimento do Moodle, portanto o servidor já possuía suporte a esta linguagem, não necessitando a instalação de outra tecnologia, facilitando a integração dos sistemas e possibilitando o reuso de código.

Desta forma, foram desenvolvidos os módulos que permitem o tratamento das informações do servidor Moodle a serem exibidas. Estes módulos retornam códigos XML que exibem apenas os dados que são necessários ao aplicativo cliente e, reutilizam toda a

validação de usuário, restrições e demais funções já existentes em PHP, restando apenas algumas alterações e complementações. O servidor passou então a ter duas formas de apresentação de dados, uma em HTML, que é a interface padrão já existente do Moodle, para usuários com acesso via browser e, outra em XML, que é requisitado pelo cliente móvel, ou seja, o mesmo usuário pode acessar e utilizar o Moodle através do ambiente web em uma estação de trabalho e quando necessitar, pode utilizar o aplicativo móvel para acessar as mesmas informações, porém de uma forma diferente no seu aparelho de telefonia celular (Figura 3.2).

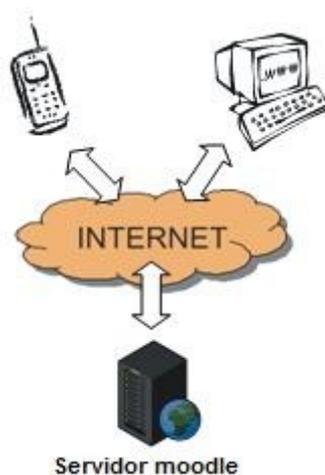


Figura 3.2 – Formas de acesso ao Moodle.

Um único módulo não se fez da utilização de XML como retorno de dados. Este módulo utiliza funções de autenticação de usuário, do Moodle, que, além de verificar o nome de usuário e a senha do mesmo, inicia uma sessão do usuário e retorna um cabeçalho com o *cookie* da sessão gerada. No corpo da página retornada, é exibido apenas “SUCCESS”, no caso de ter efetuado o *login* com sucesso, ou, no caso contrário, o devido erro.

O cookie citado acima é requisitado pelos demais módulos implementados, para que seja recuperada a sessão do usuário e assim, permitir o acesso do mesmo as informações requisitadas.

Para realizar este desenvolvimento, foram utilizadas algumas ferramentas como PHPDesigner e NetBeans. Foi de grande valia também para o desenvolvimento dos módulos do servidor, a documentação online das funções do Moodle, disponível em <http://phpdocs.moodle.org>.

3.2.3.2. Cliente

O cliente foi desenvolvido na plataforma J2ME, levando em consideração que esta plataforma é atualmente suportada por grande parte dos dispositivos móveis existentes no mercado.

Foi utilizada para a manipulação de dados de retorno do servidor, uma API para manipulação de XML, conhecida como kXML. Esta recebe os dados da conexão HTTP com o servidor e recupera os dados de acordo com instruções passadas.

A funcionalidade de *login* não utiliza esta API, pois como já foi dito, o retorno do servido, neste caso, não é em XML. Esta funcionalidade do cliente recebe o cabeçalho com o *cookie* de sessão e a resposta de sucesso ou erro. O *cookie* recebido é armazenado para ser posteriormente reenviado ao servidor em requisições futuras, permitindo assim o acesso do usuário as demais funcionalidade.

A principal ferramenta utilizada para o desenvolvimento do aplicativo cliente foi o IDE NetBeans 6.0.1, que fornece ferramentas de edição de interface e até mesmo algumas funcionalidades básicas de interfaces prontas para serem usadas, como por exemplo a tela de login, tela de espera, listas, dentre outras, o que agiliza o desenvolvimento.

Além disso, foi utilizado o emulador de telefones celulares Sun Java(TM) Wireless Toolkit 2.5.2 for CLDC, acoplado ao NetBeans IDE.

4. RESULTADO E DISCUSSÃO

Como resultado da pesquisa realizada foi desenvolvido o MobiMoodle, um aplicativo móvel para acesso a recursos do ambiente Moodle e, os devidos módulos para o Moodle que dê o suporte necessário à execução do aplicativo móvel. Nas seções a seguir serão mostradas as características do *software* produzido e descrição da implementação, onde são mostradas as decisões tomadas, as classes e como é o fluxo do sistema.

4.1. Características do Sistema

Inicialmente, para que um aparelho celular possa executar o sistema corretamente, este deve possuir suporte a configuração CLDC 1.1 e ao perfil MIDP 2.0.

Para que fosse possível a manipulação do XML retornado pelo servidor, foi necessário utilizar a API para J2ME chamada kXML. A versão atual do MIDlet tem o tamanho de 221184 *bytes*. Portanto o aparelho deve possuir este espaço livre em memória para que a aplicação seja instalada.

4.2. Visão geral e Decisões de desenvolvimento

O MobiMoodle é uma versão simplificada do Moodle desenvolvida para telefones celulares. Ele permite a realização de consultas a alguns dados disponibilizados no ambiente. A decisão de permitir apenas consultas deve-se a complexidade e a grande quantidade de dados do ambiente Moodle. Algumas funcionalidades do ambiente necessitam um maior conhecimento não só do funcionamento do ambiente, como de sua estrutura.

O Moodle é um sistema bastante grande e que possui funcionalidades como lições, que não seria viável sua utilização em celular, devido ao grande volume de dados que precisariam ser exibidos, sobrecarregando a interface. A interação do usuário com o sistema também se tornaria inviável devido ao volume de dados que este necessitaria digitar para interagir com o sistema, que, como sabemos, é de extrema dificuldade a realização de digitação em grande quantidade via dispositivo celular. Porém não é impossível essa implementação, ficando assim como idéia para a continuidade do trabalho.

O objetivo do MobiMoodle é permitir consultas simples que podem ser necessárias

eventualmente, como por exemplo a verificação de suas notas ou mesmo, se existem tarefas a serem desenvolvidas.

A sua interface é simples e de fácil navegabilidade, possuindo sempre comandos de “Entrar”, que permite a entrada no item selecionado, “Voltar”, que retorna sempre a tela anterior e “Sair”, que pode sair completamente do sistema ou retornar para a tela de *login* para que se possa entrar no sistema com outro usuário.

Como dito anteriormente, o MobiMoodle foi desenvolvido sob a arquitetura cliente/servidor e pode ser visto como uma estrutura repartida em módulos de funcionamento no cliente e no servidor. Portanto, o dispositivo móvel faz requisições ao sistema servidor, e os dados requisitados do servidor são retornados em formato XML, para facilitar a recuperação dos mesmos. No entanto a captura e a manipulação destes dados podem requerer muito processamento, podendo ocorrer alguma demora na exibição dos dados.

Sempre que é necessária a requisição de dados do servidor, é feito o uso de telas de espera. Estas telas permitem a execução de tarefas e a captura de erros de execução, direcionando para diferentes telas em caso de sucesso ou erro. Além disso, elas servem como uma tela de informação ao usuário, exibindo mensagens de aguarde e a tarefa que esta sendo executada, permitindo que este saiba o que esta acontecendo e não se assuste caso demore um pouco para obter um resultado.

No diagrama de fluxo da figura 4.1, que foi gerado pelo NetBeans, pode se observar a existência das telas de espera e das demais telas do sistema, além do fluxo de telas e navegação do sistema, que serão melhor descritas nas sub-seções seguintes, onde as descrições são feitas para cada tela do sistema.

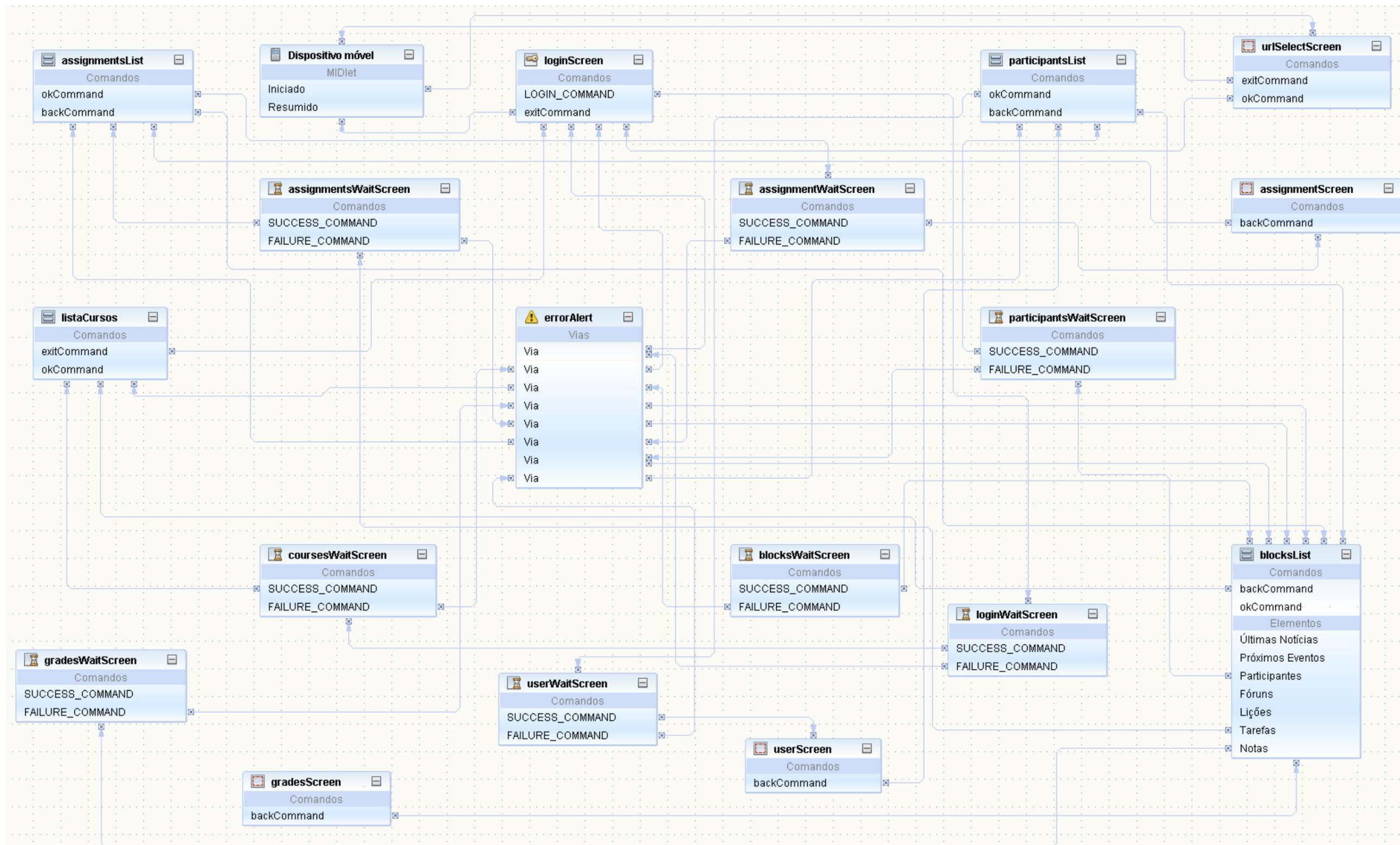


Figura 4.1 - MobiMoodle: Diagrama de Fluxo - MIDP-2.0.

4.3. MobiMoodle: Definição do Servidor

Assim que o usuário inicia o sistema, a primeira tela a ser exibida ao mesmo é a tela para definição da URL do servidor Moodle. Esta pode ser observada na figura 4.2.

Pode-se perceber que é necessário que seja fornecida a URL do servidor Moodle. Este servidor deve estar com o módulo de suporte ao MobiMoodle. Atualmente, a validade da URL só é verificada ao tentar efetuar *login* no sistema. Porém, se a URL não tiver sido fornecida, o sistema retorna uma tela com a seguinte mensagem de erro: “O endereço do servidor Moodle deve ser fornecido!”. Ao clicar em “Done”, será retornado para a tela anterior, para que seja fornecido a URL.

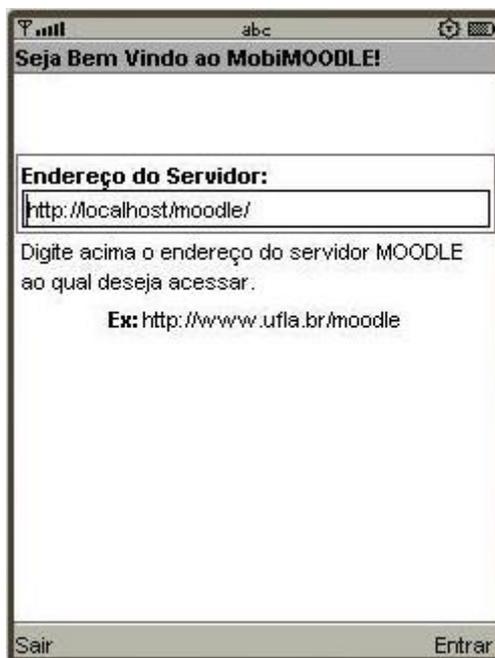


Figura 4.2 - MobiMoodle: Tela de definição da URL do servidor vista de um emulador de telefone celular.

4.4. MobiMoodle: Login

Logo após a definição da URL do servidor, é exibida para o usuário a tela de login, como pode ser visto na figura 4.3.

Nesta tela alguns dados são requeridos do usuário. Estes dados são referentes aos dados previamente cadastrados pelo usuário durante a primeira vez que ele acessou o Moodle através de um computador pessoal e efetuou o registro de um *nome de usuário* e *senha*. Nessa tela, ao selecionar os campos *nome de usuário* e *senha*, serão exibidas telas onde deverá ser informado pelo usuário do sistema o *nome de usuário* e a *senha* cadastrados, como pode ser visto na figura 4.4. Nenhum sistema de criptografia é utilizado

no cliente, ficando a cargo do servidor.

A partir deste momento ele se conecta ao servidor do sistema e, em caso de sucesso no *login*, o aplicativo armazena o *cookie* de sessão retornado pelo servidor e, posteriormente exibe uma tela contendo a lista dos cursos nos quais o usuário foi previamente cadastrado. Caso contrário uma tela de erro é mostrada e o usuário retorna à tela de *login*.

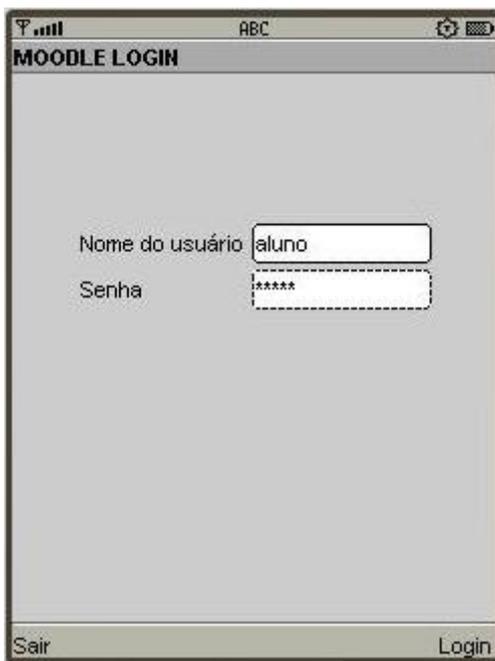


Figura 4.3 - MobiMoodle: Tela de *login*, vista de um emulador de telefone celular.

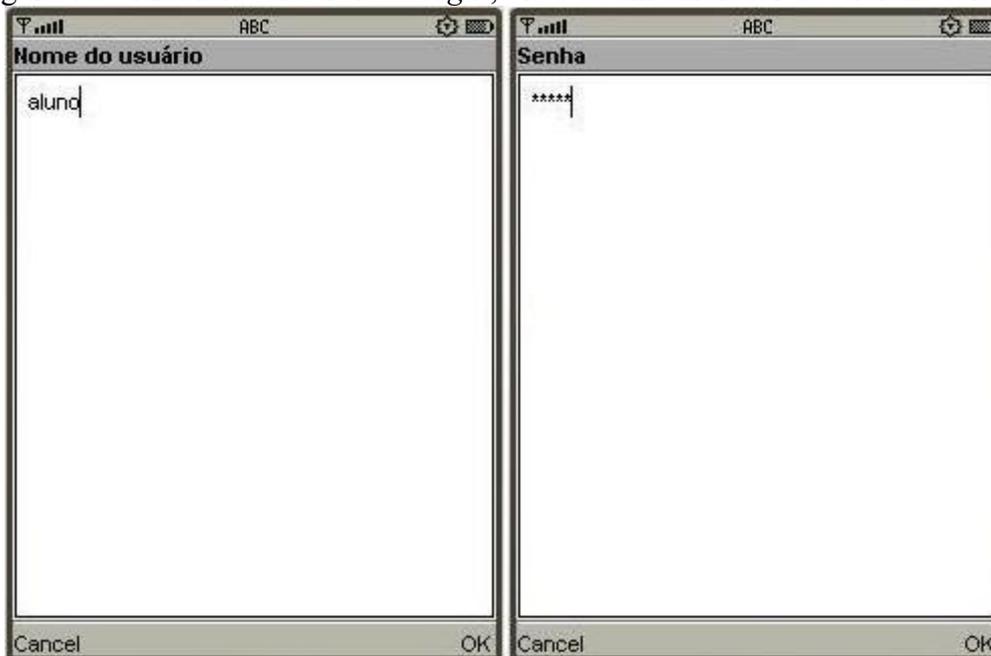


Figura 4.4 - MobiMoodle: Telas para digitação do Nome de Usuário e Senha, vistas de um emulador de telefone celular.

As telas de erro exibidas caso, o *nome de usuário* ou a *senha* estejam incorretos ou, não tenham sido fornecidos, podem ser vistas na figura 4.5.

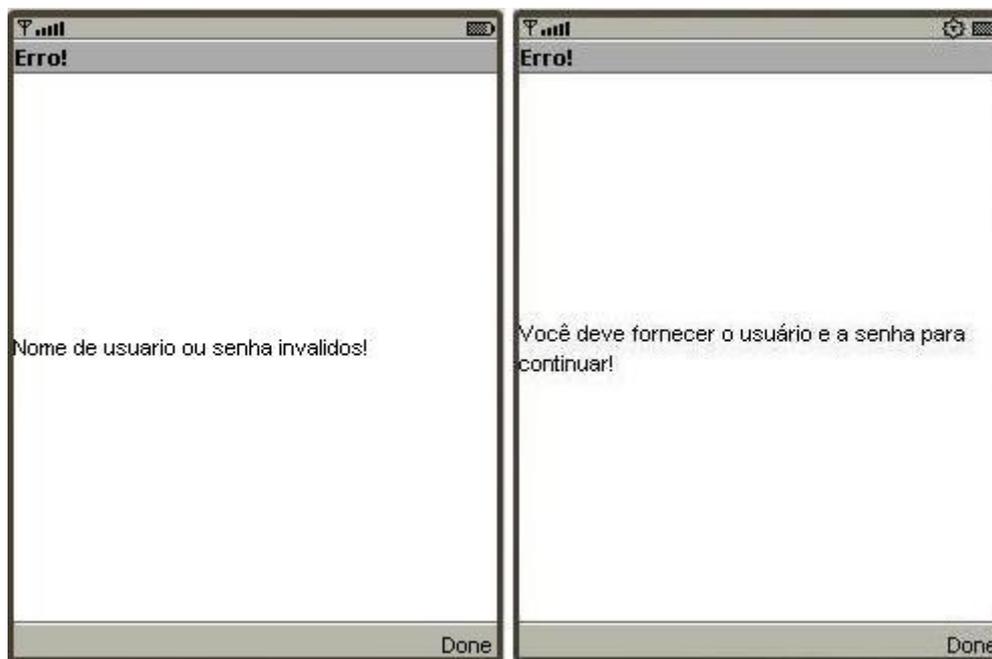


Figura 4.5 - MobiMoodle: telas de erro de *nome de usuário* e ou *senha* incorretos e não fornecidos, respectivamente, vistas de um emulador de telefone celular.

4.5. MobiMoodle: Meus Cursos e Menu Principal

Após efetuado o *login* com sucesso, será exibida para o usuário uma tela com a lista dos cursos nos quais o usuário está cadastrado como participante, como pode ser visto na figura 4.6. Se o usuário clicar em “Sair”, será exibida a tela de *login* novamente. Do contrário, ao selecionar o curso desejado e clicar em “Entrar”, será exibida para o usuário a tela principal do curso, onde são listados os recursos disponíveis para o curso, como *tarefas*, *notas*, *participantes*, dentre outras.

A figura 4.7, mostra a tela principal do curso. Nesta tela poderá ser selecionado o menu que o usuário deseja visualizar. Caso o usuário clique em “Voltar”, a lista de cursos será exibida novamente e poderá ser selecionado outro curso para ser visualizado.



Figura 4.6 - MobiMoodle: Tela “Meus Cursos”, que lista os cursos do usuário, vista de um emulador de telefone celular.



Figura 4.7. MobiMoodle: Tela principal do curso, vista de um emulador de telefone celular.

4.6. MobiMoodle: Participantes do Curso

Os participantes do curso poderão ser visualizados selecionando o menu “Participantes” na tela principal e clicando em “Entrar”. Inicialmente será exibida uma lista com o nome completo de todos os participantes cadastrados no curso, como se vê na

figura 4.8. Nesta tela o usuário poderá navegar pelos participantes e selecionar qualquer um, caso queira ver os dados detalhados disponíveis deste, o usuário deverá clicar em “Entrar”.

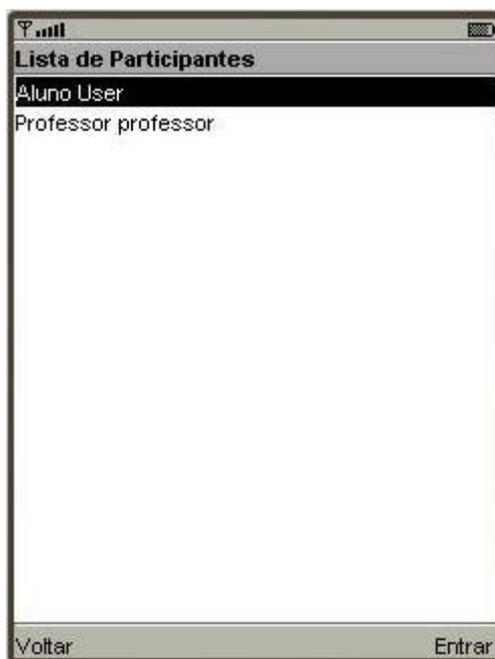


Figura 4.8 - MobiMoodle: Lista de participantes do curso, vista de um emulador de telefone celular.

Os dados detalhados, disponíveis, do usuário participante do curso poderão ser visualizados como na figura 4.9.



Figura 4.9 - MobiMoodle: Tela de visualização dos dados detalhados dos participantes, vista de um emulador de telefone celular.

4.7. MobiMoodle: Tarefas do Curso

Da mesma forma que os participante são vistos, o usuário também poderá visualizar a lista de tarefas a fazer, porém esta lista exibirá na tela, não somente o título da tarefa, mas também a data de entrega da mesma. No entanto, isso deixa a interface levemente sobrecarregada de informações, porém de maneira que ainda possibilite uma fácil visualização dos dados. Esta tela pode ser vista na figura 4.10. Nesta tela também poderá ser selecionada a tarefa que desejar visualizar detalhadamente.

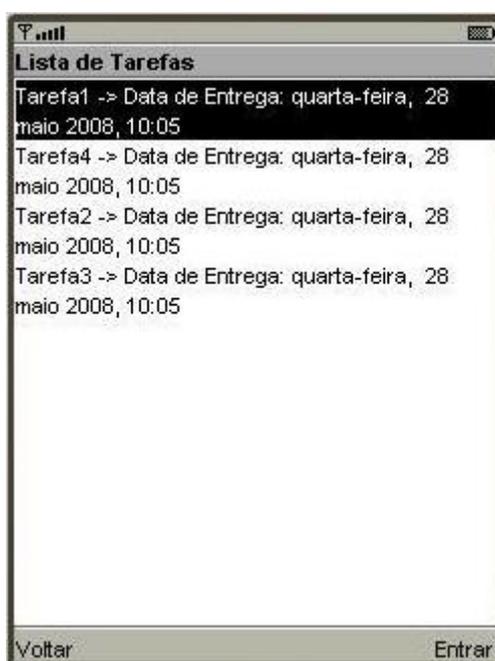


Figura 4.10. MobiMoodle: Tela com a lista de tarefas do curso, vista de um emulador de telefone celular.

Ao selecionar a tarefa e clicar em “entrar”, será exibida tela mostrada na figura 4.11. Nesta tela o usuário tem visualização das informações detalhadas da tarefa selecionada. Caso o professor já tenha dado algum retorno para a tarefa selecionada, aparecerão alguns campos extras, que são o “*Feedback*”. No ambiente *web*, alguns tipos de tarefas permitem a entrega de tarefas através da postagem de arquivos ou mesmo das resoluções digitadas diretamente na página. Estes recursos de postagem não estão disponíveis no aplicativo móvel, mas é uma idéia para desenvolvimentos futuros.

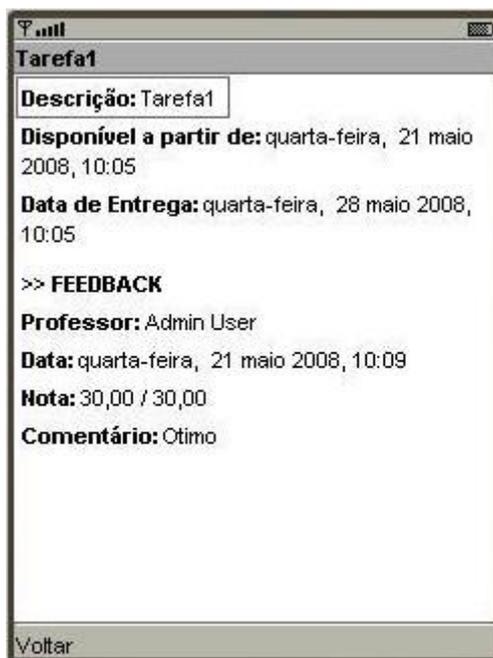


Figura 4.11 - MobiMoodle: Tela de exibição das especificações detalhadas da tarefa, vista de um emulador de telefone celular.

4.8. MobiMoodle: Tabela de Notas do Aluno

As notas do usuário para o curso selecionado também poderão ser visualizadas ao selecionar o menu “Notas” e clicar em “Entrar”. Como pode se observar na figura 4.12, será exibida uma tabela com o item avaliado, a nota em pontos e a porcentagem referente.

The screenshot shows a mobile application window titled 'Notas'. It displays a table with the following data:

Item:	Nota:	%
Tarefa1	30,00	100,00 %
Tarefa2	-	-
Tarefa3	10,00	33,33 %
Tarefa4	25,00	83,33 %
Total	72,22	72,22 %

A 'Voltar' button is located at the bottom left of the screen.

Figura 4.12. MobiMoodle: Tela de exibição de Notas, vista de um emulador de telefone celular.

Observe que ao final da tabela é exibido a nota total adquirida no curso. Os itens que não foram avaliados pelo professor até o momento, aparecerão na tabela com “-” no local onde deveria constar a nota e a porcentagem.

4.9. Considerações Finais

A utilização de dispositivos móveis tornou-se quase indispensável e, acima de tudo, acessível à maioria das pessoas. A tecnologia móvel foi capaz de atingir o cotidiano das pessoas e fazer parte da vida delas, modificando suas rotinas e formas de tomar decisões, podendo ser considerada hoje, uma revolução. Muitas pessoas não vivem sem celular, outras não abrem mão de estar com seu *PDA* conectado na Internet e ao mesmo tempo estar se deslocando pela casa. As tecnologias móveis são, atualmente, muito úteis, pois dão maior mobilidade para a execução de tarefas.

Por outro lado, o crescimento da EaD nos últimos anos permitiu que este alcançasse uma posição de respeito. Esta tem sido uma área de grandes pesquisas atualmente, onde pesquisadores buscam melhores formas de educar, sem a necessidade de o aluno estar presente em uma sala de aula, já que o cotidiano do Brasileiro tem se tornado a cada dia mais corrido. E por que não aproveitar as tecnologias móveis disponíveis para auxiliar nesta forma de aprendizagem?

A internet já tem sido utilizada há algum tempo como auxílio na EaD, que deixou de utilizar meio como cartas, para utilizar-se de emails e fóruns disponíveis em ambientes virtuais de ensino. Hoje se tem notado até mesmo que alunos deixaram de frequentar bibliotecas e passaram a consultar livros e enciclopédias em meio digital.

Ainda não se sabe se a tecnologia tem auxiliado na melhoria da qualidade de ensino, porém com certeza a educação tem atingido uma maior parcela da população.

Foi nesse intuito que surgiu a idéia do MobiMoodle, que possibilita uma mobilidade ainda maior na educação. A versão atual do aplicativo desenvolvido não atinge a interatividade desejada, mas é um início e, pode ser complementada de modo a oferecer mais funcionalidades e uma maior interatividade entre professores e alunos ou mesmo entre alunos e alunos.

O MobiMoodle permitirá ao usuário se manter informado sobre as atividades que estão sendo desenvolvidas no ambiente Moodle a qualquer momento do dia e em qualquer lugar.

5. CONCLUSÃO

Nos estudos realizados e utilizando a metodologia proposta, foi alcançado o objetivo do desenvolvimento do aplicativo, chegando à conclusão de que o aplicativo desenvolvido, o MobiMoodle, possibilitou o acesso ao ambiente Moodle e a utilização dos recursos do ambiente que foram propostos. Porém, para uma maior interatividade do aluno com o sistema, possibilitando que este seja utilizado como ferramenta móvel de aprendizagem, sob o conceito do m-learning, seria necessário que fossem desenvolvidas outras funcionalidades.

O protótipo desenvolvido neste trabalho é uma contribuição inicial à EaD, possibilitando que os estudantes estejam em contato constante com o ambiente virtual Moodle, verificando suas atividades de qualquer local onde haja uma cobertura celular. Para uma maior contribuição à EaD algumas sugestões para trabalhos futuro são descritas no capítulo 6.

6. SUGESTÃO PARA TRABALHOS FUTUROS

Mediante exposto na conclusão, como trabalho futuro, pode-se desenvolver mais funcionalidades do MobiMoodle, a ponto que ele, além da funcionalidades atuais, permita também que o usuário possa participar ativamente de Fóruns e Chats, como também permitir a realização de lições online ou postagem de arquivos de tarefa.

Atividades como a leitura de material disponibilizado também seria possível para alguns dispositivos móveis.

Outro trabalho futuro seria o aperfeiçoamento da interface, deixando esta mais amigável e, com maior navegabilidade.

Módulos opcionais do Moodle também poderiam ser adicionados ao aplicativo móvel, como currículo e outros.

REFERENCIAL BIBLIOGRÁFICO

AHONEN, M.; JOYCE, B.; LEINO, M.; TURUNEN, H. **Mobile Learning - A Different Viewpoint**, In KYNÄSLAHTI, H.; SEPPÄLÄ, P. (Ed). **Professional Mobile Learning**. Helsinki: IT Press, 2003.

ANATEL – AGENCIA NACIONAL DE TELECOMUNICAÇÃO, **Dados Relevantes do SMC / SMP, de 23/11/2007**, WWW. Disponível em <http://www.anatel.gov.br>. Acesso em 05/12/2007, 2007.

BARROS, A. R.. **Estudo sobre a utilização da tecnologia wireless no desenvolvimento de aplicações distribuídas multi-usuário**, Monografia de Graduação – DCC – UFLA, 2005.

DALL'OGGIO, P. **PHP Programando com Orientação a Objetos: *Inclui Design Patterns***. 1. ed. São Paulo, Novatec, 2007. 576 p.

DEPINÉ, F. M.; **Protótipo de software para dispositivos móveis utilizando Java ME para cálculo de regularidade em rally**. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Universidade Regional de Blumenau, 2002.

JOHNSON, T. M.. **Java para Dispositivos Móveis**, Editora Novatec, 2007.

JUNG, C. F. **Metodologia para pesquisa & desenvolvimento: aplicada a novas tecnologias, produtos e processos**. Rio de Janeiro/RJ: Axcel Books do Brasil Editora, 2004.

KRAEMER, M. E. P.. **Educação a Distância no Ensino Superior: Um Olhar para a Sustentabilidade**. In: 11º Congresso Internacional de Educação a Distância. Salvador – Bahia. 2004.

KUHNEN, A.. **Prototipo de Uma Aplicação LBS Utilizando GPS conectado em celulares para consultar dados georeferenciados**, Trabalho de Conclusão de Curso

(Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2003.

LAGO, D. G.. **SOAC: Proposta de um Sistema Online de Auxílio ao Cafeicultor com foco na Mobilidade**. Monografia de Graduação – DCC – UFLA, Minas Gerais, 2007.

LECHETA, R. R.. **kXML – Fazendo parser de um XML com J2ME**. Revista Java Magazine, publicada em 19/06/2006, Disponível em <http://www.devmedia.com.br/articles/viewcomp_forprint.asp?comp=2099>.

LEE, V.; SCHNEIDER, H.; SCHELL, R. **Aplicações Móveis: arquitetura, projeto e desenvolvimento**. São Paulo/SP: Pearson Education do Brasil, 2005.

LEHNER, F.; NÖSEKABEL, H. **The Role Of Mobile Devices In E-Learning - First Experiences With A Wireless E-Learning Environment**. In: IEEE INTERNATIONAL WORKSHOP ON WIRELESS AND MOBILE TECHNOLOGIES IN EDUCATION - WMTE, 2002.

MEIRELLES L.; TAROUCO L.; ALVES C. **Telemática Aplicada à Aprendizagem com Mobilidade**. RENOTE - Revista Novas Tecnologias na Educação, Porto Alegre, 2004.

MELO, A. A.; NASCIMENTO, M. G. F.. **PHP Profissional: Aprenda a desenvolver sistemas profissionais orientados a objetos com padrões de projeto**. 1. ed., São Paulo, Novatec, 2007. 464 p.

MOODLE. **Moodle Documentation**, WWW. Disponível em <<http://www.moodle.org>>. Acesso em 06/12/2007, 2007.

MOODLE. **Moodle Documentation**, WWW. Disponível em <<http://www.moodle.org>>. Acesso em 20/05/2008, 2008.

MUCHOW, J. W. **Core J2ME - Tecnologia & MIDP**. São Paulo/SP: Pearson Makron Books, 2004.

NYIRI, K. **Towards a philosophy of m-Learning**. In: IEEE INTERNATIONAL WORKSHOP ON WIRELESS AND MOBILE TECHNOLOGIES IN EDUCATION - WMTE, 2002.

PELISSOLI, L; LOYOLLA, W. **Aprendizado Móvel (m-learning): Dispositivos e Cenários**. In: 11º Congresso Internacional de Educação a Distância. Salvador – Bahia. 2004.

RIBEIRO, R. T., **Desenvolvimento de Módulos de Controle Acadêmico Para o Ambiente Moodle**, Monografia de Graduação - DCC - UFLA, 2007.

SANTOS, N. V. M.. **Mobile Ear Trainer: Sistema Móvel de Treinamento Auditivo Para Músicos**, Monografia de Graduação – DCC – UFLA, 2007.

SERRÃO, C.; MARQUES, J. **Programação com PHP**, 1.ed., Editora: FCA, 2001, 564 p.

SILVA, K. G. **Comparativo Entre as Plataformas J2ME e .NET Compact Framework Para Desenvolvimento de Aplicações Para Dispositivos Móveis**, Trabalho de Conclusão do curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, 2006.

SUN, **Java 2 Platform, Micro Edition**. Janeiro, 2008. WWW. Disponível em <<http://java.sun.com/javame/>>

WHITE, J.; HEMPHILL, D.. **Java 2 Micro Edition**, USA, Manning Publications; 1st edition, April 2002.

WIKIPEDIA, **Ambiente Virtual de Aprendizagem**. Maio, 2008. WWW. Disponível em <http://pt.wikipedia.org/wiki/Ambiente_virtual_de_aprendizagem>