

CYNARA LIRA DE CARVALHO SOUZA

***M - Learning.*** Desenvolvendo aplicação para o aprendizado móvel.

**PETROLINA – PE**  
**2006**

CYNARA LIRA DE CARVALHO SOUZA

***M - Learning. Desenvolvendo aplicação para o aprendizado móvel.***

Monografia apresentada a Banca Examinadora da Faculdade de Ciências Aplicadas e Sociais de Petrolina – FACAPE, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Jorge Luis Cavalcanti Ramos

CYNARA LIRA DE CARVALHO SOUZA

***M - Learning*. Desenvolvendo aplicação para o aprendizado móvel.**

Monografia apresentada a Banca Examinadora da Faculdade de Ciências Aplicadas e Sociais de Petrolina – FACAPE, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Petrolina, 15 de dezembro de 2006.

BANCA EXAMINADORA

---

---

---

*Dedico este trabalho à minha família, aos meus amigos, e em especial a meu esposo Carlos Alberto e aos meus filhos Karina e Marcelo Henrique que sempre estiveram ao meu lado compartilhando momentos especiais como à realização deste trabalho.*

## **AGRADECIMENTOS**

A Deus, em primeiro lugar, por todas as coisas boas que me proporcionou e pela oportunidade de está concluindo mais uma etapa em minha vida, superando obstáculos e desafios.

À minha família pelo apoio e incentivo.

Aos meus pais que me fizeram ser a pessoa que sou com seus conselhos e lições de vida e por sempre estarem ao meu lado torcendo por mim.

Ao meu orientador, Prof. Jorge Cavalcanti por sua paciência e dedicação para que este trabalho fosse concluído com sucesso.

Ao Prof. Jocélio Passos pela sua imensa contribuição colaborando na execução deste projeto.

Sou especialmente grata ao meu amigo Tassio Virgínio pela valiosa ajuda neste projeto, sem a qual não conseguiria concluí-lo.

À minha amiga Lays Tacianny, companheira de tantos estudos, pela força, amizade e momentos de descontração.

A todas as pessoas que contribuíram direta e indiretamente para conclusão deste trabalho.

**"Se o teu sonho for maior que ti**

**Alonga tuas asas**

**Esgarça os teus medos**

**Amplia o teu mundo**

**Dimensiona o infinito**

**E parte em busca da estrela... "**

***(Trecho de Voa - Ivan Lins / Leda Selma)***

## RESUMO

A utilização de dispositivos móveis na educação criou um novo conceito, denominado *Mobile Learning* ou *m-Learning*. Seu grande potencial encontra-se na utilização de 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.; SYVÄNEN ET AL., *apud* Barros, 2005].

A aplicação de novas tecnologias na área do ensino e da aprendizagem constitui um exemplo concreto que realça algumas necessidades a (sensibilização, formação, atualização), em especial na procura de novas soluções educativas que satisfaçam alunos, professores, instituições de ensino, e a sociedade em geral. Embora inicialmente existam algumas dificuldades de adaptação, esta evolução possibilita a criação de novas soluções que visem satisfazer necessidades e perspectivas das pessoas, facilitando-lhes tarefas quotidianas, libertando-lhes tempo e recursos. Se por um lado se requer sensibilização e formação no uso de novas tecnologias, por outro, para tornar efetivo o seu interesse, exige-se uma permanente atualização na forma de como as adaptar às necessidades sociais.

Será apresentada uma aplicação desenvolvida para celular que disponibiliza provas simuladas para vestibulares explorando o processo de aprendizagem através de dispositivos portáteis, uma forma de inserir tecnologia para melhoria e inovação das atividades inerentes a estudantes em preparação para vestibulares.

**Palavra Chave:** *Mobile Learning*.

## ABSTRACT

The use of mobile devices in the education has created a new concept, denominated Mobile learning or m-learning its great potential is based in the use of mobile technology as part of a model of integrated learning characterized by the use of wireless communication devices, bearing a transparent shape and with way high level of mobility [AHONEN ET AL.; SYVÄNEN ET AL., *apud* Barros, 2005].

The application of new technologies in a teaching and learning field, constitutes a solid example that highlights some needs the (sensitization, formation, updating), especially in the search of new educational solutions that can satisfy our students, teachers, teaching institutions, and the society in general. Although initially there are some adaptations hindrances; this updating makes possible the creation of a new breakthrough that has the aim to satisfy needs and people perspectives facilitating their everyday tasks, adding and giving them time and resources.

If on one side it is requested sensitization and formation in the use of new technologies, on the other hand, and to turn effective their interest, a permanent updating is demanded in a way that can be adapted to the former social needs.

An application will be presented developed for cell phone that makes available simulated tests for college entrance exams exploring the learning process through portable devices, a form of inserting technology for improvement and innovation of the correlated activities to students in preparation for college entrance exams.



## SUMÁRIO

CAPÍTULO 1 .....	01
INTRODUÇÃO .....	01
1.1. Motivação .....	02
1.2. Objetivos.....	03
1.2.1. Objetivo Geral.....	03
1.2.2. Objetivos Específicos .....	03
1.3. Estrutura do Trabalho .....	03
CAPÍTULO 2 .....	05
DISPOSITIVOS MÓVEIS .....	05
2.1. Principais Desafios.....	06
2.2. Telefonia Móvel Celular.....	07
2.2.1 Tecnologias de Telefonia Celular .....	10
2.2.1.1.Tecnologia TDMA .....	10
2.2.1.2. Tecnologia GSM .....	10
2.2.1.3. Tecnologia CDMA .....	10
2.3. Tecnologia <i>Bluetooth</i> .....	12
2.4. Utilizações e Aplicações.....	24
2.4.1. Utilização de Dispositivos Móveis para o Ensino de Ciências.....	12
2.4.1.2. Computação móvel em museus. MuseuM – uma aplicação de m-learning com realidade virtual.....	13
2.4.1.3. Computação Móvel aplicada ao estudo da genética.....	14
2.4.1.4. Dispositivos móveis na Educação Física .....	15
CAPÍTULO 3 .....	16
<i>MOBILE LEARNING</i> .....	16
3.1. Visão Geral da Computação Móvel .....	16
3.2. <i>Mobile Learning</i> .....	18
3.2.1.Aprendizagem utilizando dispositivos móveis .....	19
3.2.2.Vantagens e Desvantagens de dispositivos móveis no ensino.....	23
CAPÍTULO 4 .....	25
TECNOLOGIAS E ARQUITETURA DE DESENVOLVIMENTO DE APLICAÇÕES PARA CELULAR .....	25
4.1. Tecnologias disponíveis .....	25
4.1.1. Tecnologia BREW .....	25
4.1.2. Tecnologia Java.....	26
4.1.3. Tecnologia Java 2 <i>Micro Edition</i> - J2ME.....	28
4.1.2.1 – Camada de Perfil .....	29
4.1.2.2. – Camada de Configuração .....	30
4.1.2.3. – Camada do Interpretador.....	31
4.1.2.4. – Funcionamento do MIDLET .....	31
4.1.2.5. Interface .....	32
4.2. Arquitetura de um sistema móvel .....	33
4.2.1.Unidades Móveis ( <i>Mobile Units</i> ).....	34
4.2.2. Unidades Móveis e Estações de Base.....	34
4.2.3. Arquitetura Física.....	34
4.2.4. Arquitetura Lógica .....	35
4.3. Banco de Dados Móveis .....	37
CAPÍTULO 5 .....	38

METODOLOGIA .....	38
5.1. Observação, cenários e protótipos.....	38
5.1.1. Observação dos usuários.....	38
5.1.2. Cenários .....	39
5.1.3. Requisitos.....	40
5.1.4. Prototipagem de baixa fidelidade .....	41
5.2. Diagramas UML.....	42
5.2.1. Diagrama de Caso de Uso.....	42
5.2.2. Diagrama de Classe .....	42
5.2.3. Diagrama de Atividades .....	43
5.3 Técnicas e Ferramentas utilizadas.....	43
CAPÍTULO 6 .....	45
DESENVOLVIMENTO DA APLICAÇÃO .....	45
6.1. Observação do Usuário.....	45
6.2. Descrição do Cenário atual .....	45
6.2.1. Cenário 1 - <i>Realização de um simulado por parte do Professor</i> .....	46
6.2.2. Cenário 2 – <i>Realização de um simulado por parte do Aluno</i> .....	47
6.3.1 Cenários Futuros .....	48
6.3.2. Cenário Futuro – <i>Realização do simulado pelo aluno através aplicação.</i> .....	48
6.4. Requisitos não funcionais .....	49
6.5. Requisitos funcionais .....	49
6.6. Protótipo de baixa fidelidade 1 .....	50
6.7. Modelagem do Protótipo2 .....	50
6.7.1. Diagrama de Caso de Uso.....	50
6.7.2. Diagrama de Classe .....	51
6.8. Modelagem do Servidor da Aplicação .....	56
6.9. Arquitetura da Aplicação.....	57
6.10. Modelagem do Banco de Dados .....	57
6.11. Operacionalidade da Implementação .....	58
CAPÍTULO 7 .....	65
CONCLUSÃO .....	65
TRABALHOS FUTUROS .....	66
REFERÊNCIAS BIBLIOGRÁFICAS.....	67
ANEXOS.....	72
Anexo I – CÓDIGO FONTE.....	72

## LISTA DE FIGURAS

Figura 1 - Exemplo de dispositivos móveis. Fonte (Schafer, 2004).....	017
Figura 2 - Exemplo de um smartphone (adaptada de Palm, 2004).....	08
Figura 3 – Modelo de comunicação do protocolo WAP.....	09
Figura 4 – Tecnologias celulares utilizadas no Brasil (em %). ....	11
Figura 5 - Redes celulares no Mundo (em %). ....	11
Figura 6 - Telas de aplicações de educação física em <i>palmtops</i> [PALMGEAR, <i>apud</i> Barros, 2005]. ....	15
Figura 7 – Interação do estudante com o <i>m-learning</i> . ....	19
Figura 8 - Usuário estudando com um PDA [SHARPLES ET AL., <i>apud</i> Barros, 2005]. .....	22
Figura 9 - Plataforma Java 2, Java 2 Micro Edition. ....	27
Figura 10 – Camadas do J2ME .....	29
Figura 11 – Aplicações Midlet .....	30
Figura 12 - Ciclo de vida de um Midlet .....	32
Figura 13 - Arquitetura de um Sistema de Computação Móvel.....	33
Figura 14 - Ambiente de computação móvel em uma arquitetura Cliente/Servidor convencional. ....	37
Figura 15 – Etapas principais da metodologia .....	38
Figura 16 - Eclipse SDK 3.2 .....	44
Figura 17 - <i>Emulador DefaultGrayPhone</i> .....	44
Figura 18 - Diagrama de Atividades do professor.....	46
Figura 19 - Diagrama de Atividades do Aluno. ....	47
Figura 20 – Principais telas de papel da aplicação. ....	50
Figura 21 - Diagrama de caso de uso da aplicação.....	50
Figura 22 - Diagrama de Classes .....	51
Figura 23 – Diagrama do Servidor da Aplicação .....	56
Figura 24 – Arquitetura da Aplicação.....	57
Figura 25 - Modelagem do Banco de Dados.....	57
Figura 26 – Diagrama de atividades do servidor (PC) da aplicação.....	59
Figura 27 – Diagrama de Atividades do cliente (celular) da aplicação .....	59
Figura 28- Tela de login.....	62
Figura 29- Tela de lista de provas.....	63
Figura 30- Tela de lista de questões.....	63
Figura 31 - Tela de Pergunta com as respectivas respostas .....	64
Figura 32- Tela de Aviso - “errado”	
Figura 33 – Tela de aviso - “certa” .....	64

## LISTA DE QUADROS

<b>Quadro 1</b> - Vantagens e Desvantagens na utilização de PDAs e telefones celulares no ensino .....	24
--------------------------------------------------------------------------------------------------------	----

## CAPÍTULO 1

### INTRODUÇÃO

O termo *Mobile Learning* ou *m-Learning* tem sido empregado para tratar da utilização de dispositivos de comunicação sem fio, de forma transparente e com alto grau de mobilidade, como parte de um modelo de aprendizado integrado. A educação pode ser vista como um processo de exploração, descoberta e construção de conhecimento [MARÇAL, SANTOS, VIDAL, ANDRADE, RIOS, 2005].

Essa oferta de serviços de telecomunicações e de artefatos computacionais, capazes de prover mobilidade aos diferentes participantes de projetos educacionais, apresenta uma oportunidade para o desenvolvimento de pesquisas no campo da computação móvel aplicadas à educação [MEIRELLES ET AL., *apud* Marçal, 2005].

Atualmente os telefones celulares estão presentes no cotidiano da maioria das pessoas, transformando-se em um importante instrumento para a universalidade do conhecimento, ao fornecer a possibilidade de acesso a conteúdos das mais diversas áreas em diferentes momentos do dia-a-dia, seja na aula ou no trabalho.

As pessoas estão ficando mais e mais acostumadas ao conceito de estarem constantes e virtualmente conectadas e interativas, o que tem um impacto em seus estilos de vida bem como em suas necessidades.

Segundo a ANATEL (Agência Nacional de Telecomunicações), em outubro de 2006 foi registrado 770.895 novas adesões à telefonia móvel celular alcançando a marca de 96.641.799 habilitações a estes serviços [ANATEL, 2006].

O surgimento de novas tecnologias para transmissão de dados via telefonia celular e de novos modelos de telefones celulares com altas taxas de transmissão capazes de transmitir dados, imagens e em alguns casos vídeo, torna interessante o uso da telefonia celular para o *m-learning* que será referenciado no capítulo que trata dos dispositivos móveis.

Inúmeros trabalhos têm sido desenvolvidos com enfoque na utilização de dispositivos móveis para educação e treinamento [RATTO ET AL.; SUKIMOTO; KYNASLAHTI ET AL., *apud* Marçal, 2005].

Nesse contexto, este trabalho analisa os dispositivos móveis e dispositivos portáteis *Personal Digital Assistant* (PDAs), os PocketPCs, TabletPCs [PDA, 2006] e a nova geração de telefones celulares como elementos de ajuda no processo de ensino/aprendizagem e propõem um modelo de sistema que permita utilizar um telefone celular como dispositivo para ajudar neste processo, fornecendo ao aluno suporte ao processo de aprendizado e serão identificados e discutidos os principais cenários para o uso destes dispositivos como elementos de acesso à informação.

Neste trabalho, será desenvolvida uma aplicação que disponibiliza provas simuladas em dispositivos móveis para auxílio a aprendizagem.

### 1.1. Motivação

As tecnologias de computação móvel encontram-se, atualmente, em franca evolução e parecem destinadas a transformar-se no novo paradigma dominante da computação [MYERS & BEIGL, *apud* Barros, 2005].

Destaca-se na evolução da computação móvel a aplicação educacional como inovadora nesta área. O termo *Mobile Learning* ou *m-learning* tem sido utilizado para tratar o uso de dispositivos móveis como parte de uma modelo de aprendizado integrado.

Na área educacional destacamos estudantes em preparação para processos seletivos para ingressarem em universidades. Nas instituições em que trabalham com essa preparação são desenvolvidas provas com base nesses processos. É fato que estes testes são elaborados exatamente para que os alunos adquiram mais segurança nos processos seletivos como também é uma forma de ter um acompanhamento no rendimento dos conhecimentos adquiridos durante o ano de preparação. O conteúdo é extenso e são diversas as disciplinas as quais devem ser estudadas e exercitadas durante o período de preparação, portanto são necessárias provas para exercitar tempo e inserção de conhecimento. Com isso surge a origem da concepção da aplicação.

Aplicações para dispositivos móveis para o processo de ensino-aprendizagem, por ser uma tecnologia inovadora, provocam questões desafiadoras como: Quais os dispositivos e cenários para *m-learning*? Como é o processo de desenvolvimento para aplicações para celular? Diante disso, pretende-se realizar um estudo sobre a *m-learning*, a inserção de dispositivos móveis na educação, como instrumento de auxílio à aprendizagem, bem

como desenvolver uma aplicação para alunos pré-vestibulandos baseado nesta tecnologia.

## **1.2. Objetivos**

### **1.2.1. Objetivo Geral**

Compreender os conceitos de *m-learning* relevantes para o desenvolvimento de aplicações voltadas para dispositivos móveis, e os próprios dispositivos móveis e cenários utilizados para esta tecnologia, o processo de desenvolvimento de aplicações utilizando esses dispositivos na educação, apresentando uma aplicação voltada para esse fim.

### **1.2.2. Objetivos Específicos**

- Realizar um estudo sobre os dispositivos móveis para *m-learning*
- Estudar o conceito *Mobile Learning* ou *m-Learning*;
- Realizar um estudo sobre o desenvolvimento de uma aplicação para celulares;
- Apresentar a metodologia utilizada para desenvolvimento de uma aplicação para celular ;
- Apresentar os resultados do processo de desenvolvimento de uma aplicação para celular.

## **1.3. Estrutura do Trabalho**

Esta monografia está dividida em sete capítulos.

O segundo capítulo trata dos dispositivos móveis como telefones inteligentes, suas utilizações e aplicações e tecnologias empregadas pela telefonia móvel celular entre outras informações relevantes ao assunto deste capítulo.

No terceiro capítulo será apresentado um estudo mais aprofundado sobre o conceito *Mobile Learning* ou *m-Learning*;

O quarto capítulo apresenta tecnologias e arquitetura de desenvolvimento de aplicação para celular: linguagens, banco de dados e outras informações necessárias à exploração deste tema;

O quinto capítulo será mostrado a metodologia da aplicação proposta baseada em *m-learning*;

O sexto capítulo demonstrará o resultado do desenvolvimento da aplicação proposta através de diagramas, protótipo, etc.

O último capítulo trata das considerações finais sobre o trabalho e sugestões para trabalhos futuros.



## CAPÍTULO 2

### DISPOSITIVOS MÓVEIS

De acordo com os atuais conceitos de computação móvel e acesso sem fio (*wireless*), denominam-se dispositivos portáteis ou móveis aos PDAs (*Palm e Pocket PC*) [PDA, 2006], celulares e *smartphones* (telefones inteligentes) em geral [LEH, 2006].

Os dispositivos móveis passaram a ser computadores que podem ser facilmente levados a qualquer lugar, criados para atenderem profissionais e pessoas em movimento que necessitam de rapidez, facilidade e segurança no acesso a informações corporativas e pessoais. Além disso, as grandes inovações trazidas pela tecnologia *wireless* fizeram com que a indústria deste setor tenha tido um crescimento explosivo nos últimos anos, tornando-se uma das mais eficientes e rápidas áreas tecnológicas do mundo, permitindo que as pessoas comuniquem-se de forma barata e fácil sem ficarem presas aos seus telefones ou computadores de mesa.

Conforme Dalfovo et al (*apud* Schaefer, 2004), a abordagem de dispositivos móveis (celulares e PDA), nos remetem a equipamentos presentes no cotidiano das pessoas e estão se tornando formas eficazes na busca de comunicação segura e de preferência *on-line*. Eles permitirão ao usuário deslocar-se junto com seu ambiente computacional e ter um acesso constante às fontes de informações. Na Figura 1 pode-se ver algumas categorias de dispositivos móveis.



Figura 1 - Exemplo de dispositivos móveis. Fonte (Schafer, 2004)

Esses dispositivos oferecem muitos recursos, entre eles está o ambiente Java (onde os aplicativos desenvolvidos na linguagem Java são executados) [JAVA, 2006], que

possibilita o desenvolvimento de inúmeras aplicações para auxiliar na coleta e análise de informações e na tomada de decisões.

## 2.1. Principais Desafios

A computação móvel está em constante ascensão, PDAs [PDA, 2006], computadores portáteis e celulares em especial, e várias promessas como *internet* de bolso, câmeras de alta resolução, videoconferência, livros eletrônicos entre outras, contudo, toda novidade traz mudanças e com os dispositivos móveis e *wireless* não foi diferente.

Segundo Alcântara [*apud* Schaefer, 2004], existem desafios e paradigmas no campo sociológico. Os usuários costumam a trocar o papel por uma tela, por mais fina que seja. O fato mais importante a ser observado é a maneira como se usa um celular ou dispositivo portátil, além das limitações da tela, ele não é usado da mesma forma de como se utiliza um PC, mesmo que consigamos atingir seu mesmo desempenho tecnológico. Um celular é usado em situações específicas e tem-se que dar importância a esses fatos, ou fica-se limitado na hora de construir os produtos.

Ainda, conforme Alcântara [*apud* Schaefer, 2004], além dos desafios no campo sociológico, há desafios no campo técnico. Hoje, há dois grandes problemas:

- a) largura de banda;
- b) duração da bateria.

O primeiro nos leva de encontro com a computação convencional, já que nela temos o mesmo obstáculo, a segurança. As informações são transmitidas pelo ar, através de ondas eletromagnéticas, transmissões *wireless*, onde a confiabilidade é questionada. No segundo, já existem grandes avanços, a capacidade de armazenamento melhorou sensivelmente e novas pesquisas estão sendo feitas para solucionar o problema.

O autor cita ainda outros desafios que precisam ser solucionados para que os dispositivos móveis revolucionem o modo como os computadores são usados hoje:

- a) portabilidade;
- b) consumo de energia;
- c) integridade dos dados;
- d) espaço de armazenamento;

e) interface reduzida: tela limitada, sem disponibilidade de janelas.

Mas, atualmente, grandes fabricantes fazem investimentos acentuados nesta área para fazer com que tenhamos mais recurso tecnológico nos dispositivos móveis, embora ainda não se faça uso de todos esses recursos. O mercado está esperando um crescimento grande no número de usuários conectados a rede sem fio. Em estudo elaborado pela Microsoft Research [ALCÂNTARA, *apud* Schaefer, 2004], o número de pessoas conectadas a redes sem fio ultrapassou a barreira de um bilhão em 2004.

Demonstra-se que o setor de dispositivos móveis aliados a recursos como *wireless*, atraíram uma atenção científica e mercadológica recente, fazendo dele uma área em franca expansão, expandindo a computação para outros domínios até então inexplorados [SCHAEFER, 2004].

Acredita-se em superação desses desafios, pois a transmissão de dados por dispositivos móveis está se tornando uma nova frente de negócios, onde diversos setores da economia buscam cada vez mais uma forma de diminuir tempo e custos e melhorar a qualidade de seus serviços. Neste contexto, os dispositivos móveis, juntamente com outras tecnologias, podem proporcionar soluções a vários setores, tais como: comercial, jornalístico, publicitário, educacional entre outros. Neste trabalho será demonstrado a implementação de um aplicativo para dispositivos móveis, visando ao setor educacional.

## **2.2. Telefonia Móvel Celular**

Muitos dos dispositivos móveis, principalmente os celulares, disponíveis no mercado apresentam recursos de transmissão de dados, contudo, o tráfego desses dados ainda é muito caro.

Com a entrada de novas operadoras no mercado nacional, surgiram muitas ofertas de aparelhos e tarifas bem mais baixas. O tão usado “celular pré-pago”, com a novidade da compra de créditos, também permitiu que pessoas de mais baixa renda pudessem controlar seu gasto e fazer uso da telefonia celular.

Atualmente o telefone celular é um dispositivo com preço acessível estando em franca expansão no território brasileiro e conseqüentemente sendo utilizado por pessoas de diversas faixas etárias, isso torna o telefone celular um elemento muito importante e que

deve ser levado em consideração nos projetos de *m-learning*, ou seja, aprendizagem móvel.

Um sistema móvel é definido como uma rede de comunicações por rádio que permite mobilidade contínua por meio de células. A comunicação sem fio, por outro lado, implica em comunicação por rádio sem necessariamente requerer passagem de uma célula para outra durante a conversação [ALENCAR, *apud* Paludo, 2003].

Muitas pessoas não sabem, exatamente, quais as funções do celular, utilizando-o apenas para fazer e receber chamadas. Mesmo os modelos mais simples de telefone possuem acesso a diversos serviços, oferecidos pelas operadoras, que podem modificar o dia-a-dia.

A Figura 2 apresenta o exemplo de um dispositivo móvel (um *smartphone*), destacando seus recursos. É importante observar, através da imagem, a convergência de diferentes recursos em um único dispositivo, o que permite, além da comunicação por voz, a transmissão de dados remota (*e-mail*) ou local (infravermelho), som, a captura de imagem (câmera) e a entrada de dados através de um teclado.



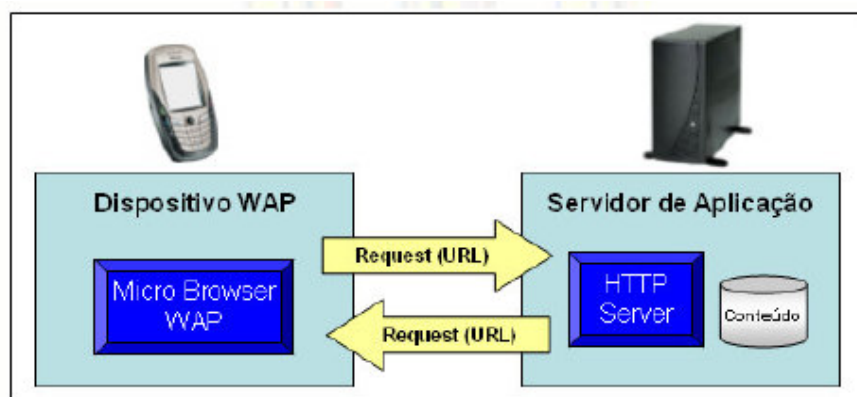
Fonte: Barros, 2005

**Figura 2 - Exemplo de um smartphone (adaptada de Palm, 2004).**

Os *smartphones* ou telefones inteligentes apresentam um propósito muito interessante, unir no mesmo aparelho o *handheld* e o celular - acoplado tanto para fazer ligações quanto para navegar na *web*. O principal ponto negativo dos *smartphones* é o preço muito alto.

O WAP (*Wireless Application Protocol*) é um protocolo de transmissão de dados na Internet para dispositivos móveis, padronizado pelo WAP Forum [WAP Forum, 2004]. O WAP foi projetado para suportar comunicações sem fio e para ser independente do dispositivo e do sistema celular utilizado.

O modelo de comunicação WAP é similar ao modelo de comunicação Web. Isso significa que ele provê vários benefícios para a comunidade desenvolvedora de aplicações, incluindo um modelo de programação familiar e a capacidade de reutilização das ferramentas atuais, como os servidores Web, ver **Erro! Fonte de referência não encontrada..** Atualmente, o protocolo WAP é muito usado para transmissão de dados na telefonia celular.



Fonte: Barros, 2005

**Figura 3 – Modelo de comunicação do protocolo WAP.**

Esse modelo de comunicação é para sites para celular. O modelo acima não se aplica a proposta deste trabalho, pois a aplicação a ser desenvolvida ficará inserida no celular e necessitará do protocolo HTTP (*HyperText Transmission Protocol* – Protocolo de Transmissão por Hipertexto) [HTTP, 2006] para a busca dos dados no servidor da aplicação. No capítulo do desenvolvimento da aplicação aprofundaremos mais este assunto.

## 2.2.1 Tecnologias de Telefonia Celular

### 2.2.1.1. Tecnologia TDMA

A sigla TDMA vem do inglês *Time Division Multiple Access*, que quer dizer "Acesso Múltiplo por Divisão de Tempo" [TDMA, 2006]. O TDMA é um sistema de celular digital que funciona dividindo um canal de frequência em até seis intervalos de tempo distintos. Cada usuário ocupa um espaço de tempo específico na transmissão, o que impede problemas de interferência [PALUDO, 2003].

### 2.2.1.2. Tecnologia GSM

A tecnologia GSM (*Global System for Mobile Communications*), ou Sistema Global para Comunicações Móveis, sistema móvel celular (padrão europeu) empregando tecnologia digital [PALUDO, 2003].

É considerada a evolução do sistema TDMA (*Time Division Multiple Access*). Oferece uma série de opções de chamadas, como espera, desvio de chamada, chamada com restrição, uma variedade de serviços de dados em que é dispensável o uso de *modems* específicos, a troca dos dados do usuário entre telefones através do *Sim-Card* (Microchips portáteis que armazenam o código e a linha do celular) e acesso mais rápido a serviços WAP (*wireless application protocol*) e *Internet*, através do sistema GPRS (*General Packet Radio Service*-tecnologia que implementa a transmissão de dados por pacote às redes GSM, chamado serviço celular de 2ª e meia Geração).

Essa tecnologia promete resolver o problema de segurança nas operações financeiras, devido padrão digital e também por possuir a criptografia de ponta a ponta nas estações rádio-base, estações que fazem a comunicação entre os celulares independente de rede. A criptografia de 128 bits é tanto para o canal de voz quanto para o de dados. [PALUDO, 2003].

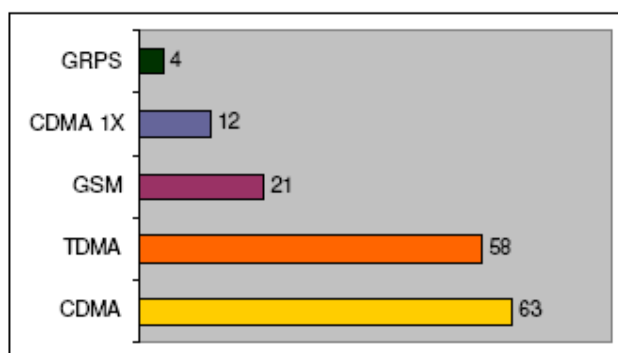
### 2.2.1.3. Tecnologia CDMA

O CDMA é um sistema de celular digital que funciona transformando a voz ou dados transmitidos pelo usuário de seu celular em um sinal da rádio codificado, que é recebido pelas antenas e transformado novamente para o receptor [PALUDO, 2003].

A sigla CDMA vem do inglês *Code Division Multiple Access*, que quer dizer "Acesso Múltiplo por Divisão de Código". Um dos padrões utilizados nas redes digitais de telefonia móvel, que usa a tecnologia de espalhamento espectral para a utilização de toda a largura de transmissão. Assim, um grande número de pessoas acessa, simultaneamente, um único canal da antena celular sem que haja interferência.

Manteve-se durante muito tempo restrito a aplicações militares, aproveitando suas características de privacidade (dificuldade de interceptação) e resistência a sinais interferentes (intencionais ou não). Apenas na década de 80 essa tecnologia começou a ser explorada comercialmente [PALUDO, 2003].

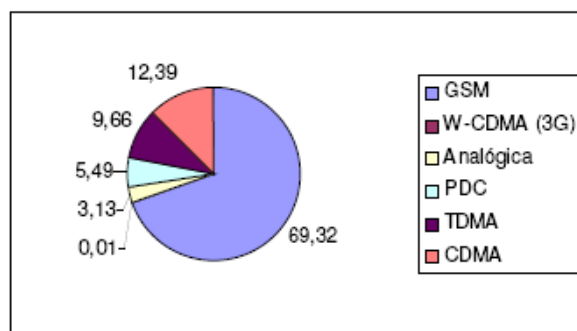
No Brasil, as tecnologias de celulares, em porcentagem nas 100 empresas mais ligadas em tecnologia, segundo pesquisa da revista Info Exame do abril de 2003 estava assim distribuída mostrada na figura 4 [PALUDO, 2003]:



Fonte: Paludo, 2005

**Figura 4 – Tecnologias celulares utilizadas no Brasil (em %).**

As redes celulares no mundo estavam assim distribuídas, em porcentagem, segundo o Northstream Research em maio de 2003, conforme figura 5 (apud Paludo, 2005):



Fonte: Paludo, 2005.

**Figura 5 - Redes celulares no Mundo (em %).**

### 2.3. Tecnologia *Bluetooth*

A interface de comunicações sem fio *Bluetooth* foi desenvolvida como uma especificação aberta para comunicação de voz e dados, originalmente, projetada para comunicação em uma direção entre um dispositivo de rádio e um computador. Ela funciona como alternativa à interface de comunicação unidirecional com radiação infravermelha, oferecida pela *Infrared Data Association (IrDA)*. A utilização do infravermelho na transmissão limita o raio de ação do dispositivo a 1m, na ausência de qualquer obstrução [ALENCAR, *apud* Paludo, 2003].

A interface incorpora alguma inteligência, permitindo a troca de dados automaticamente, sem intervenção do usuário. As especificações dessa tecnologia são coordenadas pelo *Bluetooth Special Interest Group*, que tem entre seus fundadores empresas como Ericsson, IBM, Intel, Nokia e Toshiba.

O *Bluetooth* é voltado para a comunicação entre dispositivos com baixo poder de processamento e pouca energia, substituindo completamente as conexões físicas.

### 2.4. Utilizações e Aplicações.

Nesta seção serão apresentados alguns exemplos de aplicações, de diferentes áreas do conhecimento, que utilizam dispositivos móveis para auxiliar o aprendizado.

#### 2.4.1. Utilização de Dispositivos Móveis para o Ensino de Ciências

Luchini et al. [*apud* Barros, 2005] apresenta o projeto *Mobile Learning Tools for Science* (MaLTS). Este trabalho tem como principal objetivo apresentar orientações para o desenvolvimento de soluções educacionais utilizando-se dispositivos móveis. O projeto foi dividido em três fases.

Na primeira fase, foi analisado material de pesquisa relacionado a softwares educativos em geral. Inicialmente, foram estudadas as estratégias cognitivas dos alunos de ciências. Depois, ferramentas educacionais para computadores pessoais foram analisadas.

Por último, foram pesquisados requisitos de usabilidade em aplicações para dispositivos móveis. Os estudos identificaram duas características principais em aplicações com alto grau de usabilidade em dispositivos móveis: aplicações sucintas, que apresentam somente a informação necessária e de forma altamente acessível; e, aplicações



automatizadas, que reduzem a necessidade do usuário digitar textos ou executar comandos. Entretanto, as principais contribuições dessa fase são as duas orientações para projetos de soluções educacionais em dispositivos móveis, que são:

- Telas baseadas em tarefas. A atividade da aprendizagem deve ser decomposta em tarefas e para cada uma deve ser elaborada uma tela contendo as informações necessárias específicas.
- Elementos com propósito duplo. As telas devem incluir componentes de interface que tenham funções educacionais e operacionais, como imagens que apresentam um conteúdo e funcionam como *links* para outras telas, por exemplo: numa imagem de um corpo humano onde cada parte abriria uma outra tela com informações relacionadas à parte selecionada.

A segunda fase apresentou três softwares educativos desenvolvidos para *handhelds Pocket PC* [LEH, 2006], que foram construídos utilizando-se as conclusões da fase anterior. E na fase três, foram apresentados os resultados da aplicação das soluções desenvolvidas em alunos da oitava série. A avaliação didática da solução foi baseada nos critérios apresentados por Quintana et al. [*apud* Barros, 2005]. O trabalho destaca o sucesso na utilização de softwares educativos móveis baseados em tarefas, seguindo a linha instrucionista (tipo de modelo de ensino), porém alerta que atividades complexas podem dificultar a usabilidade e comprometer a aprendizagem.

#### **2.4.1.2. Computação móvel em museus. MuseuM – uma aplicação de m-learning com realidade virtual**

Essa aplicação emprega recursos de realidade virtual em dispositivos móveis para auxílio à aprendizagem, apresentando um museu e suas obras virtuais, com o intuito de aumentar a motivação para o conhecimento de obras e propiciar excursões virtuais interativas.

Segundo [MARÇAL, SANTOS, VIDAL, ANDRADE, RIOS, 2004] como um ambiente de apoio à aprendizagem, os ambientes virtuais disponibilizam aos educadores, a oportunidade de possibilitar aos alunos o aprendizado por experimentação, pois o aluno poderá movimentar-se, ouvir, ver e manipular objetos, como se estivesse no mundo real. Uma das vantagens mais perceptíveis da representação virtual de museus é a possibilidade de o usuário conhecer e aprender culturas de diferentes regiões sem ter de se deslocar para um museu específico. Outras vantagens também identificadas são:

- Motivação para uma maior exploração do acervo do museu, através da possibilidade de aproximação e manipulação das obras;
- Visualização e simulação de experimentos nos museus de ciências, para que os alunos possam fixar os conceitos teóricos aprendidos;
- Maior poder de ilustração já que a realidade virtual, possibilitando a visão em três dimensões, permite o emprego de múltiplas camadas de referência (profundidade, tamanho, posição, etc.) no aprendizado.

#### **2.4.1.3. Computação Móvel aplicada ao estudo da genética**

O laboratório EDGE (*Exploring Dynamic Groupware Enviroments*) da Universidade Dalhousie no Canadá desenvolveu um programa para auxiliar as crianças a explorarem os conceitos de genética, denominado Geney® [BARROS, 2005].

Os pesquisadores queriam investigar como atividades de estudo em grupo poderiam ser estruturadas utilizando-se dispositivos móveis. Então surgiram algumas dúvidas:

- Como os PDAs poderiam ser adaptados para serem usados por crianças, tendo em vista terem sido projetados para adultos?
- Como os dispositivos móveis poderiam ser utilizados em atividades colaborativas?

O Geney® foi desenvolvido em um processo iterativo e centrado no usuário. O programa simula uma população de peixes representando um reservatório. Os peixes são distribuídos entre os PDAs, de forma que cada aluno/PDA fica com um tanque de peixes.

Os peixes mais velhos morrem com o tempo. Os alunos podem cruzar os peixes e, conseqüentemente, obter novos peixes. Os alunos também podem trocar peixes com os colegas via infravermelho, possibilitando um aumento da variedade genética no seu tanque.

O objetivo deste jogo é provocar uma cooperação entre os alunos a fim de gerar uma espécie de peixe com características particulares. Essas características são definidas no início da aprendizagem. Os alunos devem cooperar entre si, porque somente assim eles podem alcançar o objetivo principal.

Os pesquisadores desenvolveram um trabalho aprofundado que envolveu os seguintes etapas:

- Análise de requisitos usando cenários e modelos em tamanho real.
- Validação dos requisitos juntamente aos usuários, na qual se analisou como as crianças interagiriam com um *palmtop*.
- Desenvolvimento de protótipos.
- Teste dos protótipos com os usuários.
- Elaboração de uma especificação completa da solução.

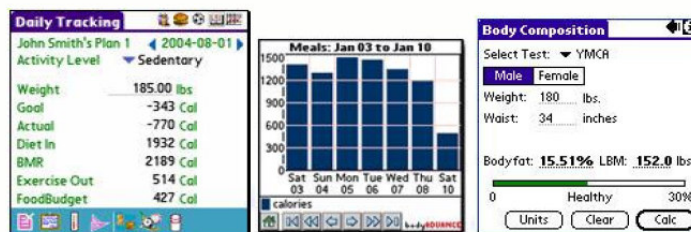
*Mandryk et al.* [apud Barros, 2005] destacam como principal característica do projeto *Geney®* a grande interação promovida, aumentando sensivelmente o grau de motivação dos alunos.

#### 2.4.1.4. Dispositivos móveis na Educação Física

Computadores de mesa têm um papel importante ao auxiliar o ensino, porém limitam-se ao laboratório ou à sala de aula onde estão instalados. Essa limitação se torna mais crítica ao se tratar da educação física, na qual os alunos podem estar aprendendo em ginásios ou pistas de corrida. Nesses casos, os dispositivos móveis surgem como uma alternativa interessante.

Aplicações em educação física permitem a coleta de dados, análise de informações e visualização de exames em campo. Essas possibilidades podem ser tanto usadas pelos professores para monitorar seus alunos como para o próprio estudante fazer um automonitoramento.

A Figura 6 apresenta telas de aplicações de educação física em dispositivos móveis [PALMGear, *apud* Barros, 2005]. As telas apresentadas nessa figura mostram informações que podem auxiliar professores e alunos de educação física e instrutores pessoais (*personal trainers*) tanto na aprendizagem como na execução de suas atividades.



Fonte: Barros, 2005

**Figura 6 - Telas de aplicações de educação física em *palmtops* [PALMGear, *apud* Barros, 2005].**

## CAPÍTULO 3

### **MOBILE LEARNING**

Neste capítulo será abordada a aplicabilidade da computação móvel na aprendizagem. Será apresentada a história e o conceito de *Mobile Learning* mostrando uma visão geral e destacando vantagens e desvantagens de dispositivos móveis no ensino como sua utilização na aprendizagem.

#### **3.1. Visão Geral da Computação Móvel**

A computação móvel pode ser definida como a utilização de dispositivos computacionais portáteis com capacidade de acessar informações, aplicações e serviços em qualquer lugar e a qualquer momento. Por isso, a computação móvel está fortemente associada à mobilidade de *hardware*, de *software* e da informação. As tecnologias de computação móvel abrangem não somente os computadores móveis designados por PDAs (*Personal Digital Assistants*) [PDA, 2006], *handhelds* ou *palmtops* [LEH, 2006], mas também outros dispositivos que refletem a convergência entre os computadores móveis e os telefones celulares. Esses últimos designam-se normalmente por telefones inteligentes (*smartphones*) citado no capítulo 2, já que a ênfase destes recai nas potencialidades para as comunicações remotas, às quais se adicionam as funcionalidades típicas de um dispositivo móvel. As seguintes limitações também caracterizam esses dispositivos: poder de processamento reduzido, tela pequena (às vezes monocromática), energia limitada (dependente de baterias) e comunicação com taxas de transmissão geralmente menores do que as das redes fixas [BARROS, 2005].

Os dispositivos móveis permitem que os usuários tenham acesso a serviços, independentemente de onde estão situados ou de mudanças de localização, o que caracteriza a mobilidade. Dessa forma a computação móvel amplia o conceito tradicional de computação distribuída. Isso é possível graças à comunicação sem fio [Wireless, 2006], que elimina a necessidade de o usuário manter-se conectado a uma infra-estrutura fixa e, em geral, estática. Com a diminuição dos custos desses dispositivos, a computação móvel tornar-se-á viável não somente para o segmento empresarial como também para os usuários finais [BARROS, 2005].

Algumas características importantes que distinguem os dispositivos móveis de outros tipos de computadores são [BARROS, 2005].

- A utilização de telas sensíveis ao toque, combinadas com a utilização de canetas especiais, como meio principal de entrada da informação. O uso da caneta permite selecionar informação no visor e escrever símbolos que são interpretados como letras.
- A inclusão de botões de navegação rápida que dão acesso às aplicações mais utilizadas, como agenda e calculadora.
- A disponibilização de uma gama abrangente de opções de comunicação, principalmente sem fio, incluindo a utilização de cabos seriais e USB (*Universal Serial Bus*) [USB, 2006], infravermelho [IrDA, 2006] e *Bluetooth* (tecnologia de comunicação local - seção 3.3, capítulo 3) para a troca de dados dos dispositivos entre si ou com um microcomputador, e o *IEEE (Institute of Electrical and Electronic Engineers) 802.11*(padrão de comunicação) [TERMOS#,2006] para integrar o dispositivo móvel a uma rede local sem fio. Além disso, merece destaque a possibilidade de transmissão de dados a longas distâncias através da rede de telefonia celular.
- A capacidade de sincronizar o conteúdo da memória do dispositivo móvel com um computador de mesa ou com um *laptop*, permitindo, assim, armazenar e manter uma cópia de segurança das informações existentes no dispositivo móvel. O processo de sincronização funciona de modo a assegurar que, ao final da operação, tanto o dispositivo móvel quanto o computador possuam a mesma informação.
- A utilização de cartões de memória, que permite expandir a capacidade de armazenamento limitada, tais como os cartões *Secure Digital (SD)* [SDC, 2006], *Compact Flash (CF)* [CF, 2006] e *Memory Stick (MS)* [MS, 2006]. Esses cartões permitem ainda transferir informação de um dispositivo móvel para outro, ou diretamente para um computador de mesa ou um *laptop*.
- A utilização de baterias recarregáveis que, dependendo da tecnologia, permite tempos de utilização eficientes que variam de algumas horas a alguns dias.
- A utilização de sistemas operacionais, especificamente, desenvolvida para dispositivos móveis, que gerenciam com mais eficiência o consumo de energia,

bem como as características específicas do hardware, tais como o Palm OS (Palm, 2004) [LEH, 2006], o Microsoft Windows Mobile [WM, 2006], Symbian OS [SYMBIAN OS, 2006] e o Linux OS [LINUX OS, 2006].

- A capacidade de instalar e executar aplicações com características específicas para dispositivos móveis, isto é, que sejam concebidas para operar dentro das limitações de processamento, *display*, memória e energia.
- A utilização de câmeras digitais acopladas, que permite o registro de eventos, através de fotos ou vídeos.

A combinação, no mesmo dispositivo, das várias tecnologias mencionadas acima, proporciona oportunidades únicas para o desenvolvimento de diferentes aplicações de gestão de informação, com muita utilidade nas mais variadas áreas da atividade humana, incluindo a área empresarial, o entretenimento, a saúde e o ensino [BARROS, 2005].

### **3.2. *Mobile Learning***

O crescimento e a popularização da internet vêm tornando possível utilizar novas estratégias e ferramentas para apoiar a aprendizagem a distância, oferecendo novas possibilidades para o processo de ensino-aprendizagem. Com isso, a primeira forma de ensino a distância que se utilizava basicamente de mídias impressas, rádio e televisão está sendo substituída pelo desenvolvimento do ensino a distância on-line (*e-learning*) [NTEL, 2006] [TAROUCO, FABRE, KONRATH, GRANDO, 2004], através do surgimento e disseminação de tecnologia de informação e comunicação em redes informatizadas que possibilitam a ampliação de sua infra-estrutura para o escopo do ensino a distância.

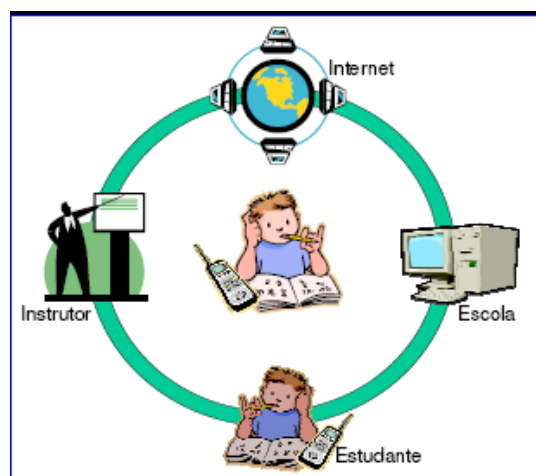
Dessa forma o *Mobile Learning* ou *m-learning* surgiu a partir do *e-learning*, ou seja, com o avanço da educação a distância, uma outra modalidade de ensino, que tem tido um avanço surpreendente é o da aprendizagem móvel, que permite acessar dados via Internet com dispositivos portáteis, dentre eles os computadores portáteis (como *laptops*, *palmtops*, etc.) e os telefones celulares [BERNARDO, BIELAWSKY, 2003].

Esse novo paradigma surge aproveitando-se da disponibilidade de dispositivos móveis e considerando necessidades específicas de educação e treinamento [NYIRI, *apud* Marçal, 2005].

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 [PELISSOLI, LOYOLLA, 2004].

Basicamente, o *m-learning* faz uso das tecnologias de redes sem fio, dos novos recursos fornecidos pela telefonia celular, da linguagem XML [XML, 2006], da linguagem JAVA [JAVA, 2006], da linguagem WAP [WAP, 2006], dos serviços de correio de voz, serviços de mensagens curtas (SMS) [SMS, 2006], da capacidade de transmissão de fotos, serviços de e-mail, *multimídia message service* (MMS) [MMS, 2006] e provavelmente em pouco tempo estará disponível o uso de vídeo sob demanda [PELISSOLI, LOYOLLA, 2004].

Um dos pontos chaves ao sucesso do *m-learning* é a construção de materiais atrativos e de fácil utilização pelo aluno/treinando. Quanto maior for a facilidade de uso e a interatividade com o material, maior será a possibilidade de sucesso. A Figura 7 demonstra a interação do estudante com as tecnologias no processo *m-learning*, onde o aluno utiliza a internet na busca de informações para seu aprendizado e se comunica com pessoas do mundo inteiro e a utilização do celular como instrumento também de auxílio através das aplicações inseridas no mesmo.



Fonte: Marçal, Andrade, Rios (2004).

**Figura 7 – Interação do estudante com o *m-learning*.**

### 3.2.1. Aprendizagem utilizando dispositivos móveis

A aprendizagem é fruto de um processo interno que surge da interação do sujeito com o meio. É uma mudança persistente no potencial humano. A aprendizagem significativa

está baseada na idéia fundamental da Psicologia cognitiva de Ausubel [*apud* TAROUCO, FABRE, KONRATH, GRANDO, 2004], na qual estabelece que a aprendizagem ocorra por assimilação de novos conceitos e proposições na estrutura cognitiva do aprendiz. Novas idéias e informações são aprendidas na medida em que existam pontos de ancoragem. Segundo Ausubel [*apud* TAROUCO, FABRE, KONRATH, GRANDO, 2004], o ponto de ancoragem é a ponte entre o conhecimento prévio do sujeito e as novas informações a serem integradas a sua estrutura cognitiva [*Idem, Ibidem*].

Estamos diante de um demasiado volume de informações disponíveis a cada novo dia. Nosso desafio educacional é saber organizar essas informações, dando prioridade ao que é mais importante para que sejam compreendidas, internalizadas. Podemos organizar as informações de variadas maneiras através do uso das novas tecnologias [TAROUCO, FABRE, KONRATH, GRANDO, 2004].

A interação e a valorização dos saberes dos sujeitos, assim como os desafios pedagógicos lançados através da exploração e vivência de novas experiências proporcionadas pela comunicação mediada pelo computador, podem assegurar uma aprendizagem cognitiva. Essa aprendizagem é aquela em cujo processamento predomina os elementos de natureza intelectual tais como percepção, o raciocínio, a atenção e a memória. É uma aprendizagem ideativa. Nessa aprendizagem pode-se distinguir a aprendizagem de habilidades: conhecimento, informações, linguagem, compreensão, resolução de problemas, aplicação de regras, análise síntese, avaliação e estratégias cognitivas [UNOESTE, 2006] no processo de educação continuada. Para oferecer educação continuada com qualidade é necessário utilizar sistemas de ensino aprendizagem altamente interativos [TAROUCO, FABRE, KONRATH, GRANDO, 2004].

A interação é proporcionada a partir de inserção de objetos de aprendizagem no contexto, que são materiais educacionais com objetivos pedagógicos, utilizados em diferentes meios inclusive dispositivos móveis [TAROUCO, FABRE, KONRATH, GRANDO, 2004].

O *m-Learning* surge como uma importante alternativa de ensino e treinamento, na qual podem ser destacados os seguintes objetivos [BARROS, 2005]:

- Melhorar os recursos para o aprendizado do aluno, que poderá contar com um dispositivo computacional para execução de tarefas, anotação de idéias, consulta



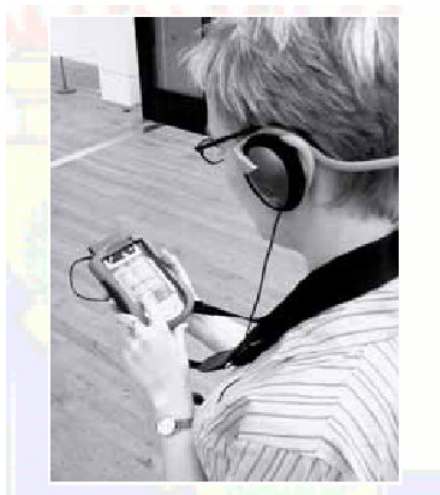
de informações via *Internet*, registro de fatos através de câmera digital, gravação de sons e outras funcionalidades existentes.

- Prover acesso aos conteúdos didáticos em qualquer lugar e a qualquer momento, de acordo com a conectividade do dispositivo.
- Incrementar e incentivar a utilização dos serviços providos por uma instituição, através do aumento das possibilidades de acesso ao conteúdo.
- Expandir as estratégias de aprendizado disponíveis, por meio de novas tecnologias que dão suporte tanto à aprendizagem formal como à informal.
- Expandir os limites internos e externos da sala de aula;
- Fornecer meios para o desenvolvimento de métodos inovadores de ensino e de treinamento, utilizando os novos recursos de computação e de mobilidade.

Em particular, dispositivos de comunicação sem fio correspondem a uma extensão natural da educação a distância via computadores (Lehner et al., *apud* Marçal, 2005), 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. A Figura 8 mostra um usuário estudando, utilizando um PDA com recursos de som.

Ao longo dos dez últimos anos, pesquisadores em educação têm mapeado a aprendizagem como uma atividade constante e vitalícia. Criou-se então o conceito CoLL (*Contextual Life-long Learning*) [SHARPLES ET AL., *apud* Marçal, 2005], cujas principais idéias estão diretamente associadas a *Mobile Learning*, entre elas:

- A aprendizagem não está restrita a lugares e horários pré-estabelecidos, acontece quando há uma quebra no fluxo da rotina e a pessoa reflete sobre uma situação para solucionar um problema, compartilhar uma idéia ou obter algum conhecimento;
- A educação formal não fornece ao aluno o conhecimento e as habilidades necessárias ao longo da vida.



Fonte: Barros, 2005

**Figura 8 - Usuário estudando com um PDA [SHARPLES ET AL., *apud* Barros, 2005].**

Assim, o *m-Learning* surge no meio como um novo conceito de aprendizagem, possibilitando ao aluno: acessar conteúdos remotamente; executar experiências em campo; capturar e armazenar eventos (com som e imagem) do dia-a-dia; e comunicar e compartilhar pesquisas com colegas e especialistas em todo o mundo[BARROS, 2005].

O instituto de pesquisa SRI (*Stanford Research Institute*) [SRI, 2006] realizou uma pesquisa sobre a utilização de dispositivos móveis nas escolas [CRAWFORD & VAHEY, *apud* MARÇAL; ANDRADE; RIOS, 2005]. Foram pesquisadas mais de 100 instituições de ensino nos Estados Unidos, do ano de 2000 a 2002. O estudo concluiu que os dispositivos móveis podem oferecer benefícios únicos aos alunos.

Os professores participantes dessa análise demonstraram uma grande aceitação da computação móvel em suas salas de aula. A seguir são apresentados alguns números que demonstram essa aceitação:

- 89% disseram que descobriram nos dispositivos móveis eficientes ferramentas de ensino;
- 93% acreditam que os dispositivos móveis podem ter um impacto positivo na aprendizagem dos alunos;
- 90% pretendem continuar a utilizar os dispositivos móveis em suas aulas;
- 75% dos professores que permitiram que os alunos levassem os dispositivos móveis para casa, constataram um aumento na conclusão dos trabalhos de casa;
- Entre os alunos, 66% acharam confortável o uso do dispositivo móvel.

- Quase a totalidade dos professores afirmou que a utilização de softwares educativos apropriados e acessórios foi de fundamental importância na aprendizagem, ao complementar os recursos básicos dos dispositivos móveis.

Em geral, a maioria dos professores afirmou que a introdução da computação móvel na sala de aula tanto aumentou a motivação dos estudantes para aprender, quanto aumentou a colaboração e a comunicação entre eles.

### **3.2.2. Vantagens e Desvantagens de dispositivos móveis no ensino**

Os dispositivos móveis, mais baratos e simples de usar, possuindo muitas das ferramentas encontradas num PC (acesso ao *e-mail* e à *Internet*, processador de texto, máquina fotográfica, leitor de MP3 [MP3, 2006], entre outras) evidenciam novas possibilidades e oportunidades para alargar o âmbito do ensino, visando incentivar e facilitar o acesso ao conhecimento.

Uma das grandes vantagens dos dispositivos móveis é de estarem permanentemente acessíveis aos utilizadores, comprovado pelo fato dos PDAs e telefones celulares poderem ser facilmente transportados (no bolso, na carteira, etc.), facultando um acesso instantâneo à aprendizagem, conceito denominado por JITL (*Just-In-Time Learning*) [JITL, 2006].

Embora a utilização dos PCs na área do ensino esteja massificada, a realidade é que os telefones celulares e mais recentemente os PDAs seguem o mesmo caminho, tendendo, eventualmente, a superar essa utilização e a convergir muito brevemente para se tornarem num objeto pessoal indispensável, conforme evidenciam os estudos efetuados no qual se enfatiza a sua utilização pela maioria dos jovens [COSTA,2005] . Dessa forma e sendo fundamental motivar e incentivar ao conhecimento, a criação de conteúdos pedagógicos para PDAs e telefones celulares torna-se primordial e um fator motivacional ao ensino e à aprendizagem[COSTA,2005].

Os recursos destes dispositivos incentivam a uma crescente mobilidade, facultando o acesso a conteúdos pedagógicos, promovendo a colaboração a distancia, a qualquer hora e sem qualquer restrição geográfica (transportes públicos, casa, etc.), considerando a permanente expansão e atualização das redes móveis (*Wireless*). Contudo, além da velocidade de transmissão das redes móveis ainda ficar distante das praticadas pelas redes fixas atuais, os dispositivos móveis apresentam, também, características próprias

que deverão ser consideradas em relação à criação de conteúdos pedagógicos de forma que esses tenham uma boa aceitação pelos alunos.

No Quadro 1 sintetizam-se algumas das vantagens e desvantagens na utilização de PDAs e telefones celulares no ensino.

**Quadro 1 - Vantagens e Desvantagens na utilização de PDAs e telefones celulares no ensino**

Vantagens	Desvantagens
Fornecem maior liberdade de interação aos alunos, não obrigando a permanecer em locais fechados.	Possuem monitores pequenos, dificultando a realização de determinadas tarefas.
São leves e fáceis de transportar.	Apresentam capacidade de memória e poder de processamento mais limitado.
Ocupam pouco espaço, permitindo a sua utilização em qualquer local (transportes públicos, casa, etc.).	Contêm baterias com autonomia reduzida (cerca de 3 a 4 horas).
Alguns modelos possuem ferramentas para reconhecimento de escrita manual, tornando a escrita mais natural face à escrita por teclado.	Encontra-se em constante atualização, pelo que podem ficar mais facilmente obsoletos.
Possibilitam um acesso JITL ( <i>Just in time learning</i> ), ou seja, um acesso instantâneo a conteúdos pedagógicos em qualquer lugar e a qualquer hora.	A sua permanente evolução cria dificuldades no desenvolvimento de aplicações informatizadas (o tamanho dos <i>monitores</i> e as funcionalidades estão em permanente mudança).
Possibilitam o desenho de diagramas e imagens diretamente sobre os monitores.	São menos robustos e mais fáceis de perderem.
Facilitam tirar apontamentos numa aula.	Mais limitado quando se utilizam animações (devido à falta de memória).
Possuem vários protocolos de comunicação sem fio (infravermelhos e <i>Bluetooth</i> ) facultando uma partilha de informação entre utilizadores de forma a promover o trabalho em grupo e a colaboração na realização de tarefas.	Mais difíceis de atualizar e expandir em termos de hardware.
Incentivam à aprendizagem pelo fato da maioria dos jovens possuírem e demonstrarem interesse na sua utilização.	As plataformas de comunicação (redes Wireless) possuem velocidade mais baixas e custos mais elevados comparativamente às redes fixas.
Na generalidade, são mais baratos comparativamente aos PCs, possibilitando combater a exclusão social.	Ainda existem poucas aplicações para o ensino à distância.

Fonte: Costa, 2005(adaptado pela Autora).

## CAPÍTULO 4

### TECNOLOGIAS E ARQUITETURA DE DESENVOLVIMENTO DE APLICAÇÕES PARA CELULAR

Neste capítulo serão apresentadas algumas tecnologias e arquitetura utilizada para o desenvolvimento de uma aplicação para celular. Este capítulo servirá de referencial teórico para apresentarmos o desenvolvimento da aplicação. Será feito uma breve descrição das tecnologias disponíveis para aplicações para celulares, das quais será mais detalhada a tecnologia escolhida. Será especificado o tipo de banco de dados a ser usado e a arquitetura, entre outras etapas importantes e indispensáveis para o desenvolvimento.

#### 4.1. Tecnologias disponíveis

As principais tecnologias disponíveis, atualmente, para a implementação de soluções móveis são: BREW e Java 2 *Micro Edition* (J2ME). Neste trabalho será feito uma breve explicação sobre a tecnologia BREW, daremos mais ênfase na J2ME a qual será utilizada na aplicação a ser proposta.

##### 4.1.1. Tecnologia BREW

A plataforma de desenvolvimento de aplicações móveis *Binary Runtime Environment for Wireless* (ambiente binário de tempo de execução para aplicativos sem fio), BREW, da Qualcomm [QUALCOMM, 2006], oferece uma solução completa e aberta para configuração de dispositivos e desenvolvimento, distribuição, faturamento e pagamento de aplicativos [PALUDO, 2003].

Ele apresenta alguns atrativos que a diferencia de sistemas como o J2ME, recursos multimídia e grande riqueza gráfica.

O custo de desenvolvimento no BREW é mais alto, pois toda aplicação tem que ser testada e certificada pela Qualcomm para garantir que não se introduza na rede aplicações mal intencionadas ou simplesmente mal feitas e com isso pondo em risco toda a credibilidade da plataforma BREW.

O BREW está presente em todos os chips CDMA fabricados pela Qualcomm e comercialmente foi lançado por algumas operadoras no mundo. Não há compatibilidade com os aparelhos TDMA e GSM.

#### 4.1.2. Tecnologia Java

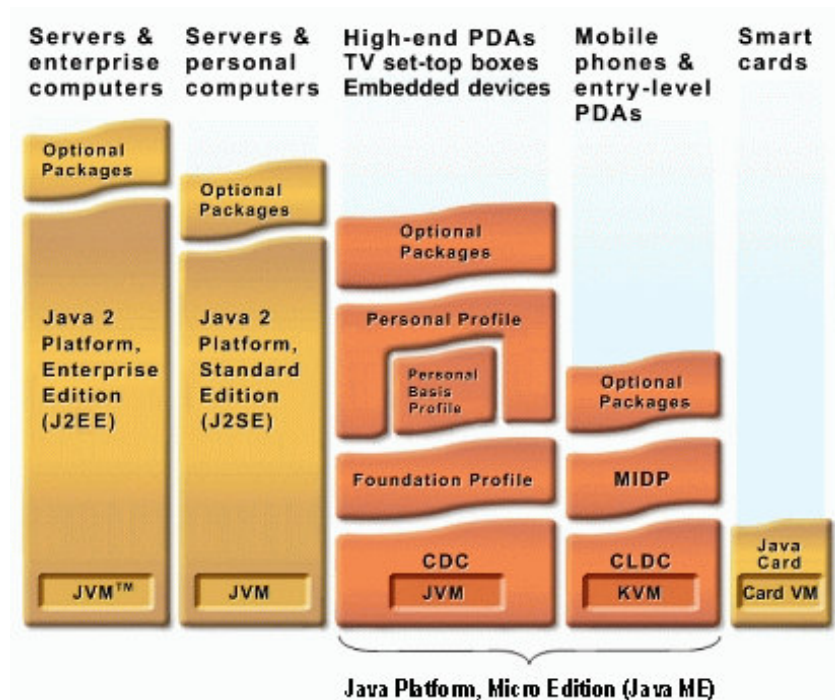
Java é uma linguagem de programação que possui algumas características particulares e importantes [SUN, 2006]: os seus programas podem ser interpretados e executados em qualquer sistema através de um interpretador Java, ou *Java Virtual Machine* (JVM), ou máquina virtual; uma aplicação desenvolvida em Java pode utilizar várias bibliotecas nativas desenvolvidas pela Sun, e o conjunto dessas bibliotecas Java é conhecido como *Java Application Programming Interface* (API), ou interface de programação de aplicação Java. Juntas, a JVM e a Java API constituem o ambiente de execução de Java ou ainda, a plataforma Java [SCHAEFER, 2004].

O Java, atualmente, encontra-se na versão 5. Quando a Sun lançou a versão 1.2 e as modificações eram tão grandes que resolveu mudar o nome para Java 2. Com isso houve também uma reestruturação dos pacotes (não confundir com packages da API), que agora contam com o prefixo J2, seguido do tipo da versão ( como em J2SE, por exemplo ) [PALUDO, 2003].

Java conta com 3 edições ou pacotes:

- J2SE (*Java 2 Standard Edition*): É a versão básica, destinada ao desenvolvimento da maior parte das aplicações de desktop e estações de trabalho. [SUN, 2006]
- J2EE (*Java 2 Enterprise Edition*): Versão destinada ao desenvolvimento de aplicações de grande porte, e servidores de aplicações. [SUN, 2006]
- J2ME (*Java 2 Micro Edition*): Destinada ao desenvolvimento de programas para periféricos móveis ou de pequeno porte, como celulares e relógios.

As três edições da Plataforma Java estão representadas na Figura 9 abaixo:



Fonte: [SUN, 2006]

**Figura 9 - Plataforma Java , Java 2 Micro Edition.**

Algumas definições importantes que serão utilizados neste trabalho [PALUDO, 2003]:

- JSDK (*Java Software Development Kit*), ou simplesmente JDK, é o ambiente de desenvolvimento Java. Entre outras coisas, incluem: Java Virtual Machine, compilador, códigos de exemplo e bibliotecas básicas do Java.
- JSP (*Java Server Pages*): Tecnologia baseada em Java para o desenvolvimento de aplicações Web.
- JDBC (*Java DataBase Connectivity*): Especificação que deve ser seguida para oferecer suporte ao acesso a Banco de Dados em Java.

Segundo Leal e Soares [apud Amorim, Borges, 2005] a tecnologia J2ME proporciona o melhor custo /benefício na medida em que:

- Profissionais da linguagem Java podem ser aproveitados;
- Amplamente adotada pelos fabricantes e operadoras de telefonia móvel;
- Compatibilidade de plataformas: aplicativos Java são escritos uma vez e rodam em diferentes dispositivos com sistemas operacionais diferentes;
- Permite programar criptografia extra para aplicativos de acesso a bancos entre outros;

- De acordo com Almeida (*apud* Amorim, Borges, 2005), em 2002 já existiam 50 milhões de aparelhos celulares com a máquina virtual Java em operação no mundo. Ele estima que até 2007, cem por cento dos celulares serão compatíveis com J2ME.

Tendo em vista que a máquina Java tende a ser incorporada em aparelhos das redes 2,5Geração, a tecnologia J2ME torna-se a principal ferramenta de desenvolvimento para aplicativos web móvel e muitos benefícios destas pesquisas estão surgindo [PALUDO, 2003].

Altos investimentos de fabricantes como a Motorola, Nokia, Siemens tornando essa plataforma a grande promessa para oportunidades emergentes [PALUDO, 2003].

#### **4.1.3. Tecnologia Java 2 *Micro Edition* - J2ME**

A tecnologia J2ME é um segmento JAVA, destinado ao mercado de dispositivos móveis. É um conjunto de especificações que disponibiliza uma JVM (máquina virtual Java), API (Interface para programação da aplicação) e ferramentas para equipamentos móveis e qualquer outro dispositivo com processamento menor do que os computadores de mesa atuais.

Com a tecnologia J2ME é possível criar aplicações especiais para tirar proveito das vantagens e particularidades dos aparelhos móveis, aplicações integradas ao serviço das operadoras, que tenham a interatividade esperada para um pequeno aparelho e que respondam de acordo com a situação e a atenção do usuário [PALUDO, 2003].

Possui uma série de vantagens em relação às demais tecnologias *wireless*, a portabilidade, ou seja, a capacidade de desenvolver aplicações que executem em qualquer dispositivo de qualquer fabricante, independente de tecnologia de operadora ou fabricante, os recursos gerais de Java podem ser utilizados, programação orientada a objetos, suporte à manipulação da tela gráfica e teclado dos dispositivos, maior suporte à conectividade com outros dispositivos, e programação em rede com o protocolo HTTP (HyperText Transfer Protocol ou seja, Protocolo de Transferência de Hipertexto) [PALUDO, 2003].

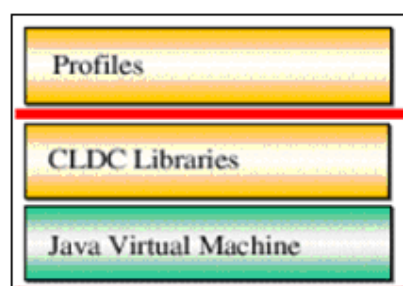
A arquitetura J2ME define configurações, perfis e pacotes opcionais como elementos para construir ambientes de execução completos que preencham os requisitos para um bom número de dispositivos e mercados-alvos [SCHAEFER, 2004]. Existem 2 tipos de configurações, o CLDC (*Connected Limited Device Configuration*), que regem as



configurações para aparelhos bem pequenos como celulares ou PDA's, e o CDC (*Connected Device Configuration*) que regem as configurações para aparelhos um pouco maiores. Será especificada nas próximas seções a configuração CLDC a qual será utilizada na aplicação proposta para celular.

A arquitetura é um modelo de camadas como mostra a figura 10.

- a) camada de perfil (*Profile*);
- b) camada de configuração (*Configuration*);
- c) camada representada pela máquina virtual J2ME (KVM).

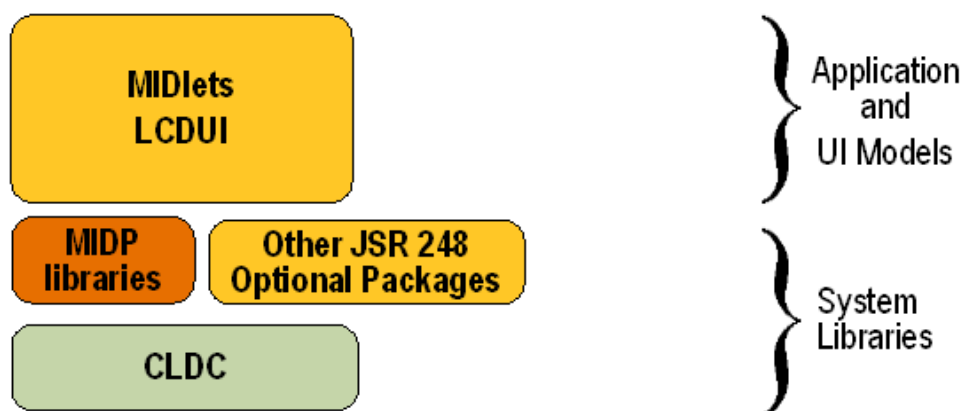


**Figura 10 – Camadas do J2ME**

#### 4.1.2.1 – Camada de Perfil

O *Profile*, ou perfil, especifica um conjunto de APIs que se moldam a um determinado tipo de dispositivo. Segundo Corbera (2003 *apud* SCHAEFER, 2004), a diferença entre Configuração e Perfil é que a Configuração descreve de forma geral uma família de dispositivos, enquanto o Perfil fica mais específico para um tipo particular de aparelho em uma dada família. O Perfil tão somente acrescenta funcionalidades àquele aparelho. Alguns dos perfis existentes são: *Mobile Information Device Profile* (MIDP), *Foundation Profile* (FP), *Personal Profile* (PP), etc.

O perfil MIDP é voltado especificamente aos dispositivos portáteis. Esse perfil especifica interface com o usuário, entrada e persistência de dados, manipulação de eventos, modelo de armazenamento orientado a registro, rede, mensagens e aspectos relacionados ao ambiente de programação para esses dispositivos (DEPINÉ, *apud* SCHAEFER, 2004). Combinado com o CLDC, fornece um ambiente de execução Java completo que alavanca a capacidade dos dispositivos e minimiza o consumo de memória e energia, sendo estas aplicações chamadas de *Midlet*, conforme apresentada na Figura 11. Seu funcionamento será explicado na p seção 4.1.2.4.



Fonte: [SUN, 2006]

**Figura 11 – Aplicações Midlet**

#### 4.1.2.2. – Camada de Configuração

A camada de configuração é classificada de acordo com as capacidades de cada dispositivo. Atualmente, existem duas configurações no J2ME: *Connected Limited Device Configuration* (CLDC) e *Connected Device Configuration* (CDC) como foi citado acima.

A CLDC é a menor das duas configurações, designada para dispositivos com conexões de rede intermitentes. Contém uma API (*Application Programming Interface*) mínima para poder rodar aplicativos em dispositivos móveis com processadores lentos e memória limitada (como celulares, PDAs).

Segundo Corbera [*apud* SCHAEFER, 2004], para o caso específico dos celulares, a configuração CLDC deve apresentar as seguintes características:

- a) o hardware do dispositivo deve ter no mínimo 160Kb para Java, um processador de no mínimo 16 bits com baixo consumo (típicas baterias de um celular) e conexão de rede *wireless* de 9.6Kbps, 144Kbps ou 2Mbps;
- b) o software deve incluir suporte a um subconjunto da linguagem Java e a um subconjunto da máquina virtual Java que definam um núcleo de funções permitindo o desenvolvimento de aplicações móveis.

As aplicações para CLDC devem ser projetadas para consumir o mínimo de recurso possível, e devem permitir ao usuário sair do aplicativo a qualquer hora sem que informações sejam perdidas.

A CDC foi feita para dispositivos que possuem mais memória, processadores e conexões mais rápidos. A maioria dos dispositivos compatíveis com CDC possui processadores de 32 bits e um mínimo de 2MB de memória disponíveis para a implementação da plataforma Java e suas aplicações.

#### **4.1.2.3. – Camada do Interpretador**

Esta camada é uma implementação da máquina virtual Java customizada para dispositivos específicos, a KVM.

Segundo Kuhnen [*apud* SCHAEFER, 2004], a *Kilo Virtual Machine (KVM)* é uma nova implementação da JVM para dispositivos portáteis com recursos limitados.

A KVM foi desenhada para ser pequena, para uso de memória estática entre 40 a 80Kb.

A quantidade mínima de memória requerida pela KVM é 128Kb, incluindo o interpretador, um mínimo de bibliotecas especificadas pela configuração e algum espaço para rodar aplicativos.

Características fundamentais para o sucesso de aplicações *wireless* podem ser atendidas plenamente com o uso de Java 2 *Micro Edition*.

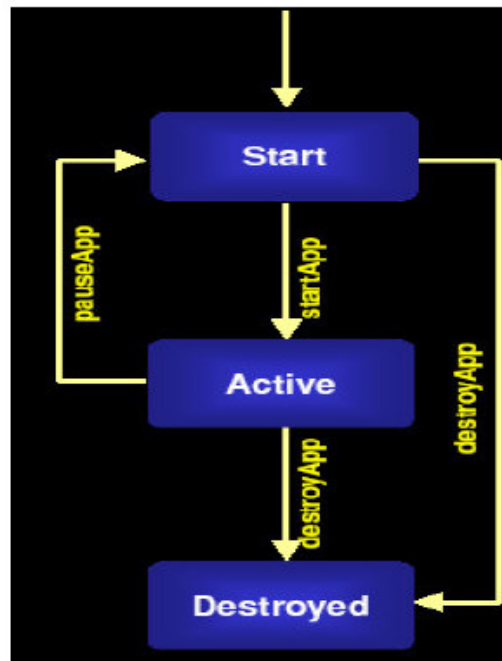
#### **4.1.2.4. – Funcionamento do MIDLET**

MIDLET são aplicações do perfil MID. São empacotadas, colocadas dentro de um container, como arquivo JAR. Possuem arquivo descritor da aplicação - JAD que acompanha o arquivo JAR, no qual estão inseridas as classes do aplicativo. O arquivo descritor contém informações, tais como: nome do vendedor, versão, tamanho, nome de cada MIDlet, ícones, imagens. [PALUDO, 2003].

O *Application Manager (AM)* de cada dispositivo é quem vai controlar os aplicativos a serem instalados, onde e como serão armazenados e como serão executados. [J2ME, 2005]

Assim que a *MIDlet* é invocada, o AM invoca o método *startApp()*, o qual coloca a *midlet* no estado *Active*. Enquanto ela estiver executando o AM pode pausar ela invocando o método *pauseApp()* no caso de uma chamada sendo recebida, ou SMS[SMS, 2006] chegando. A aplicação pode pausar a si mesma, bastando invocar

*notifyPaused()*. Assim como a AM pode pausar a aplicação e esta a si mesma, ocorre o mesmo com o *DestroyApp()* que é invocado pela AM para fechar a aplicação ou até mesmo pode ser fechada através da própria aplicação invocando o *notifyDestroyed()*. A Figura 12 mostra como funciona o Midlet, seu ciclo de vida. [J2ME, 2005]



Fonte: Paludo, 2003

**Figura 12 - Ciclo de vida de um Midlet**

#### 4.1.2.5. Interface

As MIDlets devem poder ser executadas em qualquer dispositivo, contendo a máquina virtual, sem alterações, porém isso torna-se bastante difícil na parte de interface com usuário, pois dispositivos variam de tamanho de tela, cores, teclados.

As aplicações são desenvolvidas com certa abstração de tela, pois os comandos e inserção de dados são feitos através dos botões do celular, e isso não é sabido previamente. As aplicações descobrem isto em *Runtime (tempo de execução)* e se comportam de maneira apropriada a cada celular. Já no desenvolvimento de jogos a aplicação é bem mais específica, porque o desenvolvedor precisa conhecer o dispositivo previamente para melhor aproveitamento de recursos, como disposição em tela, por exemplo [J2ME, 2005].

A tela do dispositivo é representada por uma instância da classe *Display*, a qual é obtida pelo método *getDisplay()*, geralmente, contida no método *startApp()*, o método

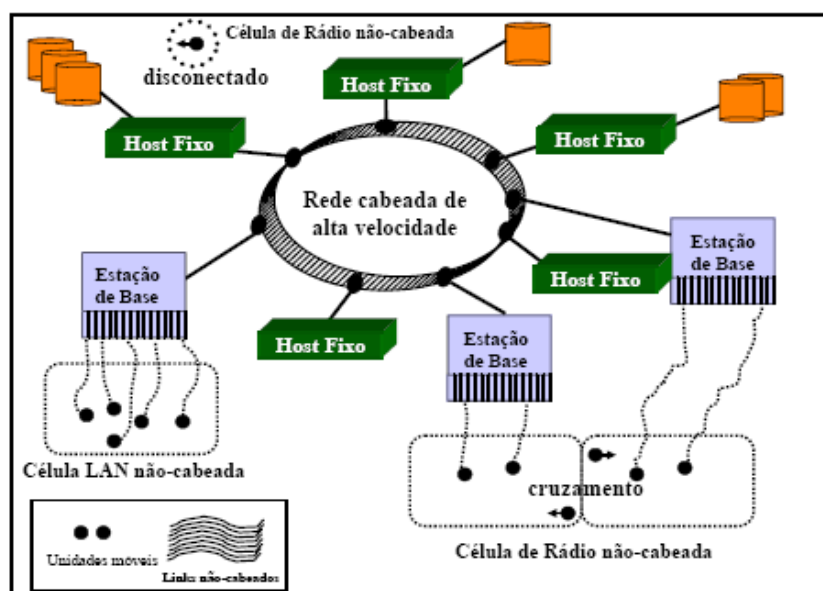
`getDisplay()`, somente fica disponível após o início da Aplicação (`startApp()`). Nessa instância de `Display` são inseridos heranças de `Displayable`.

Na tela, são mostrados componentes, objetos já instanciados, que são chamados pelo método `setCurrent()` o qual seta o objeto que será mostrado.

As ações do usuário são gerenciadas por comandos (*commands*), os quais são adicionados a componentes visuais. São usados para interação do usuário com a aplicação.

#### 4.2. Arquitetura de um sistema móvel

A arquitetura genérica de uma plataforma móvel consiste em uma arquitetura distribuída [Ozsu and Valduriez, *apud* CÔRTEZ, LIFSCHITZ, 2006], na qual computadores, geralmente, conhecidos como Hosts Fixos (*FS - Fixed Hosts*) e Estações de Base (*BS - Base Stations*), são interligados através de uma rede com fio de alta velocidade. Hosts fixos são computadores de finalidade genérica que não são equipados para gerenciar unidades móveis, mas podem ser configurados de forma a fazê-lo. Estações de Base são equipadas com interfaces para as redes sem fio e pode comunicar-se com as unidades móveis para suportar o acesso a dados [Elmasri and Navathe, *apud* CÔRTEZ, LIFSCHITZ, 2006]. A arquitetura está demonstrada na figura 13.



Fonte: CÔRTEZ, LIFSCHITZ, 2006.

**Figura 13 - Arquitetura de um Sistema de Computação Móvel**

#### 4.2.1. Unidades Móveis (*Mobile Units*)

São computadores portáteis movidos à bateria, movimentando-se livremente em um domínio geográfico de mobilidade, uma área que é restringida pela limitada amplitude de banda dos canais de comunicação sem fios [CÔRTEZ, LIFSCHITZ, 2006].

Para gerenciar a mobilidade dessas unidades, o domínio de mobilidade geográfica é dividido em domínios menores chamados de células. A computação móvel requer que o movimento de unidades seja irrestrito dentro do domínio de mobilidade geográfica (movimento intercelular), ao mesmo tempo em que possuir contigüidade de acesso durante o movimento, garante que o movimento de uma unidade móvel através dos limites das células não terá nenhum efeito sobre o processo de recuperação de dados [Elmasri and Navathe, *apud* CÔRTEZ, LIFSCHITZ, 2006].

#### 4.2.2. Unidades Móveis e Estações de Base

As unidades móveis e as estações base se comunicam através de canais sem fio que possuem largura de banda significativamente menor do que aquelas de uma rede com fio. Um canal de conexão do tipo *downlink* é utilizado para enviar dados das Estações de Base para as Unidades Móveis, e um canal de conexão do tipo *uplink* é utilizado para enviar dados das Unidades Móveis para as Estações de Base [Elmasri and Navathe, *apud* CÔRTEZ, LIFSCHITZ, 2006].

#### 4.2.3. Arquitetura Física

A arquitetura física implementa o serviço de *m-learning* pela integração do sistema de telefonia celular e internet.

Nessa arquitetura, o usuário acessa a aplicação utilizando o telefone celular que acessa o conteúdo do servidor da instituição.

O telefone celular estabelece uma conexão com a antena de telefonia celular mais próxima e o sistema de telefonia celular estabelece uma conexão com a Internet.

Conforme citado no capítulo 3, os celulares possuem tecnologias para transmissão de dados específicos para cada tipo de aparelho. Segundo a GSM World [GSM World, 2005] milhões de usuários estão adquirindo aparelhos com tecnologia GSM que do

ponto de vista do consumidor a vantagem-chave são os serviços novos com baixos custos. A tecnologia GSM utiliza a tecnologia GPRS, que o serviço é “sempre ativo”, ou seja, ele é um modo no qual os recursos somente são atribuídos a um usuário quando for necessário enviar ou receber dados. Essa técnica permite que vários usuários compartilhem os mesmos recursos, aumentando assim a capacidade da rede e permitindo uma gerência, razoavelmente, eficiente dos recursos. Isso permite às operadoras GPRS disponibilizar acesso à Internet móvel em alta velocidade e a um custo razoável, pois a cobrança é feita pela quantidade de pacotes de dados transmitidos e não pelo tempo de conexão à rede. A maioria das operadoras trabalha com uma tarifa que gira em torno de R\$5,00 (informação adquirida dia 20/10/06) por cada megabyte movimentado [CM, 2006].

Satisfeita o processo de autenticação, entra em atividade a arquitetura lógica de funcionamento do sistema que passa a interagir com o usuário.

#### 4.2.4. Arquitetura Lógica

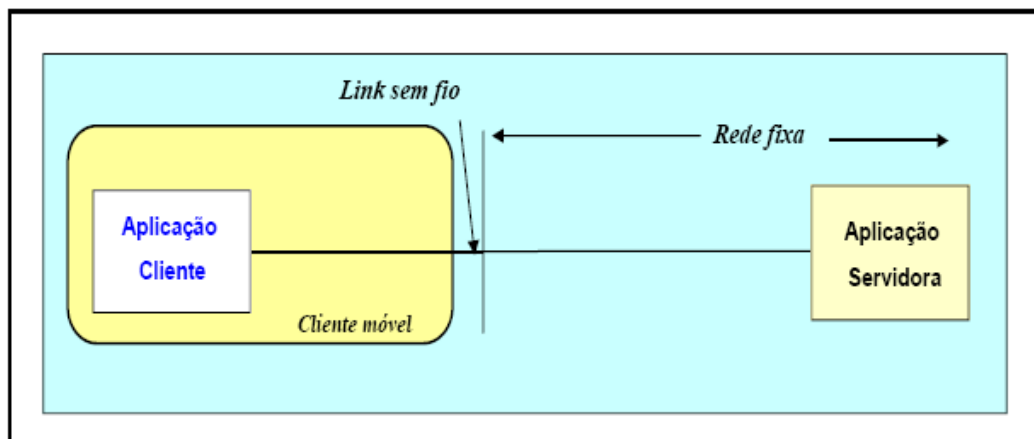
A arquitetura lógica utiliza o acesso WAP cujo objetivo básico é trazer diversos conteúdos da Internet (páginas Web) e outros serviços de dados para os telefones celulares digitais e outros terminais móveis sem fio, como: PDAs (*Personal Digital Assistants*) e *Laptops*, e o HTTP quando é trabalhado com uma aplicação inserida no celular cuja conexão serve para buscar os dados no servidor ligado ao SGBD ( sistema de gerência de banco de dados, ou *System database manager*).

O modelo (arquitetura) *cliente-servidor* supõe uma estrutura básica que consiste de muitos microcomputadores (PCs) e estações de trabalho, bem como um número menor de equipamentos de grande porte, conectados através de sistemas de redes locais e outros tipos de sistemas de rede de computadores. Um **cliente** nesta estrutura é geralmente um equipamento de um usuário que possui capacidades de interface e processamento local. Quando um **cliente** solicita acesso a funcionalidades adicionais, como por exemplo, acesso a um banco de dados e esse não reside neste equipamento, ele se conecta a um **servidor** que fornece a funcionalidade necessária. Um **servidor** é um equipamento que pode fornecer serviços para as estações clientes, como impressão, armazenamento de dados ou acesso ao banco de dados [Elmasri and Navathe, *apud* CÔRTEZ, LIFSCHITZ, 2006].

A idéia da arquitetura *cliente-servidor* é dividir as funcionalidades que precisam ser fornecidas aos usuários do SGBD em duas camadas, a camada “servidor” e a camada “cliente”, ambas com funcionalidades distintas. Essa arquitetura facilita o gerenciamento da complexidade dos SGBDs atuais [Ozsu and Valduriez, *apud* CÔRTEZ, LIFSCHITZ, 2006]. A arquitetura cliente/ servidor está incorporada aos Sistemas de Gerencia de Banco de Dados (SGBD). Em SGBDs relacionais, através de sua uma linguagem padrão chamada *Structured Query Language (SQL)*, criou-se um ponto de divisão lógica entre o cliente e o servidor do banco de dados. No cliente, estão os programas de aplicação e a interface do usuário, enquanto que as funcionalidades de consultas e transações permaneceram no lado do servidor. Nesse caso podemos dizer que o servidor é um servidor de consulta ou servidor de transação, pois fornece essas duas funcionalidades. Os programas da interface do usuário e da aplicação podem ser executados no lado do cliente. Quando é necessário acesso ao banco de dados, o programa estabelece uma conexão com o SGBD, que está no lado do servidor, e uma vez que a conexão é estabelecida, os programas no cliente podem se comunicar com o SGBD. Um exemplo de como efetivar essa conexão é através do padrão chamado *Open Database Connectivity (ODBC)* que é uma especificação de interface para acesso a dados. Essa especificação provê funções para conectar e desconectar fontes de dados, preparar e executar comandos, processar erros e processar transações [ODBC, 2006].

O ODBC fornece uma Application Programming Interface (API), permitindo que programas no lado do cliente chamem o SGBD, contanto que os equipamentos do cliente e do servidor tenham instalado os softwares necessários. Os resultados das consultas são enviados de volta para os programas no cliente, que podem processar ou exibir os resultados conforme necessário. Um outro exemplo de conexão é a utilização de programação JAVA, chamado JDBC, definido na seção que trata da tecnologia Java, permitindo que programas de clientes escritos na linguagem JAVA acessem o SGBD através de uma interface padrão [Elmasri and Navathe, *apud* CÔRTEZ, LIFSCHITZ, 2006]. A Figura 14 apresenta um ambiente de computação móvel em uma arquitetura cliente servidor tradicional.





Fonte: Cortês, Lifschitz, 2006.

**Figura 14 - Ambiente de computação móvel em uma arquitetura Cliente/Servidor convencional.**

#### 4.3. Banco de Dados Móveis

Bancos de Dados e suas tecnologias vêm tendo um grande impacto no crescimento do uso de computadores. Podemos afirmar que os bancos de dados desempenham um papel extremamente importante em todas as áreas da ciência onde a computação é utilizada. A expressão banco de dados tem sido tão amplamente utilizada que precisamos iniciar com a definição do que seja um banco de dados. Segundo [Elmasri and Navathe, *apud* CÔRTEZ, LIFSCHITZ, 2006], um banco de dados é uma coleção de dados relacionados.

Podemos ver o ambiente de computação dos SGBDs (sistema gerenciador de banco de dados) móveis como uma extensão dos sistemas distribuídos.

Os bancos de dados móveis podem ser distribuídos sob dois aspectos [Ozsu and Valduriez, *apud* CÔRTEZ, LIFSCHITZ, 2006]:

1. Todo o banco de dados é distribuído, principalmente, entre os componentes sob a rede com fio, possivelmente com replicação total ou parcial. Assim, uma estação de base gerencia seu próprio banco de dados com as funcionalidades inerentes aos Sistemas de Gerência de Banco de Dados (SGBDs), com funcionalidades adicionais para localizar unidades móveis e características adicionais de gerência de consultas e transações, para atender aos requisitos de ambientes móveis;
2. O banco de dados é distribuído entre os componentes sob a rede com fio e com os componentes sob a rede sem fio. A responsabilidade sobre a gerência de dados é compartilhada entre estações de base e unidades móveis.

## CAPÍTULO 5

### METODOLOGIA

Este capítulo apresenta os métodos utilizados para alcançar os objetivos gerais e específicos citados no capítulo 1 dessa proposta.

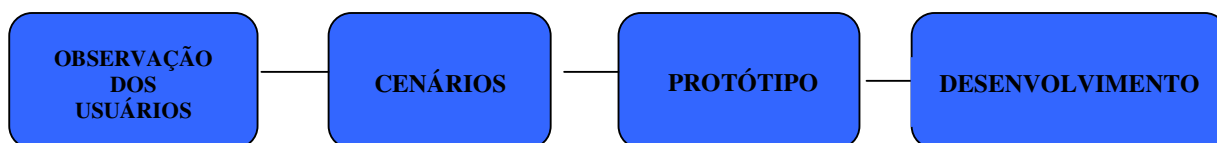
A inserção de tecnologia na educação tem demonstrado resultados positivos, como foi visto no capítulo 2, na seção 2.4, onde dispositivos móveis são utilizados em algumas áreas demonstrando um aumento no surgimento de novidades nesta área de *m-learning*. Com base nisto será descrito a metodologia utilizada para o desenvolvimento de uma aplicação para *m - learning*.

Como instrumento de coleta de dados para desenvolvimento da aplicação proposta será utilizado a observação do usuário. Após estas observações serão criados cenários atuais e futuros, onde a partir destes, retiram-se os requisitos necessários para realização da aplicação. Logo após será desenvolvido os protótipos. Com base no protótipo, vem a fase de escolha da ferramenta a ser utilizada na implementação da aplicação.

Serão utilizados diagramas no padrão UML [UML 2005] para representar cenários, representações do sistema entre outras necessidades neste trabalho.

#### 5.1. Observação, cenários e protótipos.

Conforme apresentada na figura abaixo, serão trabalhadas na metodologia da aplicação estas etapas:



**Figura 15 – Etapas principais da metodologia**

##### 5.1.1. Observação dos usuários

Esta etapa é essencial, pois é onde capturamos os requisitos necessários para concepção da aplicação.

A observação é uma pesquisa realizada no campo de trabalho onde as atividades ocorrem freqüentemente, são analisadas as situações e dessa forma compreendemos o contexto da atividade do usuário, seu ambiente e suas interações. O resultado ajuda a

fazer um levantamento dos requisitos básicos para o assunto em questão propondo melhorias e inovações para melhorar o desenvolvimento da atividade do usuário. [PASSOS, 2006].

Para alcançar essa etapa do processo foi observado:

- a frequência da realização das provas simuladas;
- o fator emocional dos usuários testados;
- o processo desde a elaboração da prova até a sua correção;
- a receptividade dos usuários ao receberem os resultados das provas.

### 5.1.2. Cenários

Cenários são histórias - passadas ou futuras - sobre pessoas e suas atividades, as quais permitem raciocinar sobre situações de uso de um artefato mesmo antes destas situações existirem de fato [Carroll, *apud* FALCÃO, 2004]. Essas histórias são descrições narrativas informais, que vêm sendo usadas em ciências humanas e sociais para descrever e analisar o comportamento humano nas mais diversas situações [Carroll, *apud* FALCÃO, 2004]. Atualmente, cenários são muito utilizados no projeto e desenvolvimento de artefatos tecnológicos e aplicações computacionais. Eles provêm uma visualização concreta das várias possibilidades futuras de uso do novo produto, permitindo identificar requisitos e possíveis dificuldades [Carroll, *apud* FALCÃO, 2004]. O contexto de uso do futuro sistema pode ser descrito, em forma de cenários, logo no início do processo de desenvolvimento, através de diversos episódios de uso. O foco destas descrições não são as operações do sistema, mas o modo de utilização do mesmo pelas pessoas para realizar suas atividades [Carroll, *apud* FALCÃO, 2004]. Dessa forma, o projeto será orientado pelas ações e experiências das pessoas que irão de fato utilizar a tecnologia [Carroll, *apud* FALCÃO, 2004].

Os elementos característicos de um cenário são:

- **Ambiente:** descreve um estado inicial do ambiente onde o episódio acontece, caracteriza-se o ambiente fisicamente, como as pessoas estão nele presentes.
- **Atores ou agentes:** aqueles que participam do episódio descrito, interagem com o ambiente influenciando ou sendo influenciados
- **O roteiro:** sequência de ações e

eventos representando o que os atores fazem durante o episódio, o que lhes acontece e que mudanças ocorrem no ambiente.

Alguns exemplos de cenário para utilização de dispositivos móveis no processo de aprendizagem são apresentados a seguir, onde será explorado mais a telefonia celular, conforme citado na introdução de dispositivos móveis [PELISSOLI, LOYOLLA, 2004]:

Cenário celular-1 - O aluno ouve pequenas gravações com as explicações do professor sobre um determinado conteúdo, no qual os instrutores abordam o assunto através de uma estrutura de começo, meio e fim bem demarcados.

Cenário celular-2 - A conferência por telefone também é um instrumento interessante, pois o aluno que está, temporariamente, deslocado do grupo pode participar das discussões através de seu telefone celular e assim interagir com os demais membros do grupo.

Cenário celular-3 - O teste em formato de jogo cria um ambiente de competitividade entre os alunos ou grupos de alunos e motiva a obtenção de mais informações, e conseqüentemente uma maior compreensão dos assuntos abordados. Os testes podem ser divididos em categorias ao invés de disciplinas assim, assuntos multidisciplinares podem ser abordados, auxiliando as ações interdisciplinares.

A partir das observações dos usuários foram elaborados cenários da situação atual e futura sobre a prática dos alunos e professores no ambiente do contexto. Esses cenários serão demonstrados sob a forma de diagramas de atividades.

### **5.1.3. Requisitos**

Segundo Pressman, os requisitos são objetivos ou restrições estabelecidas por clientes e usuários do sistema que definem as diversas propriedades do sistema. Os requisitos de software são, obviamente, aqueles dentre os requisitos de sistema que dizem respeito a propriedades do software. [AER, 2006].

Um conjunto de requisitos pode ser definido como uma condição ou capacidade necessária que o software deve possuir [AER, 2006]:

- Para que o usuário possa resolver um problema ou atingir um objetivo;
- Para atender as necessidades ou restrições da organização ou dos outros componentes do sistema

Os requisitos de software são separados em requisitos funcionais e não-funcionais. Os requisitos funcionais são a descrição das diversas funções que clientes e usuários querem ou precisam que o software faça. Eles definem a funcionalidade desejada do software. O termo função é usado no sentido genérico de operação que pode ser realizada pelo sistema, seja através comandos dos usuários, ou seja, pela ocorrência de eventos internos ou externos ao sistema. Requisitos não-funcionais são as qualidades globais de um software, como manutenibilidade, usabilidade, desempenho, custos e várias outras. Normalmente, esses requisitos são descritos de maneira informal, de maneira controversa e são difíceis de validar [AER, 2006]. Os requisitos são obtidos após os cenários.

#### **5.1.4. Prototipagem de baixa fidelidade**

Protótipo é uma valiosa ferramenta no projeto de aplicações. Eles podem ajudar na avaliação de alternativas de *design* a qualquer estágio do processo de desenvolvimento [OQP, 2006]. Durante a fase conceitual os elementos básicos do *design* podem ser explorados e testados junto com os usuários. Ao serem projetadas as telas, menus e levantadas as formas de interação em maior detalhe, estas questões podem ser discutidas, avaliadas e testadas. Mais adiante modelos mais elaborados podem ser utilizados para dar uma visão antecipada da aplicação final

A opinião recebida dos usuários, obtida a partir da utilização de um modelo do sistema com o qual eles podem interagir, facilita a obtenção de respostas e avaliação dos usuários, podendo modificar os requisitos originais bem como elaborar novas especificações. Foram realizadas duas etapas na prototipação. Primeira etapa foi criada telas de papel e a partir daí passamos para segunda etapa inserindo funcionalidades.

A prototipação de papel pode ser considerada como um método de projeto, criação, teste e comunicação de interfaces gráficas com usuários. Segundo a definição de Snyder [apud AZEVEDO, 2005] prototipação em papel é uma variação de testes de usabilidade que usuários representativos executam tarefas reais através da interação com uma versão de papel da interface manipulada por uma pessoa “atuando como um computador”, que não explica como a interface foi projetada para funcionar. A autora ainda salienta que essa técnica é independente de plataforma, podendo ser utilizada para

aplicações para internet, softwares, dispositivos *handhelds*, tudo que possui uma interface homem-computador pode ser objeto de prototipação em papel.

Essa etapa da metodologia visa criar um ambiente virtual de provas simuladas para preparação para vestibular. O Projeto baseia-se no princípio de que esta avaliação beneficiará o desenvolvimento da aprendizagem do aluno. Visa oferecer mais um suporte de auxílio ao processo de ensino-aprendizagem, tanto para o professor, pois somará como mais um instrumento de incremento neste processo, como aos alunos que ganharão mais um objeto de auxílio à inserção de conhecimento num período altamente estressante e diversificado de conteúdos. Ao concluir a prototipação de papel inicia-se a implementação da aplicação dando funcionalidade ao protótipo com construções dos diagramas e escolha das ferramentas usadas no processo de criação.

## **5.2. Diagramas UML**

UML (*Unified Modeling Language*) é uma linguagem para visualização, especificação, construção e documentação de artefatos de um software em desenvolvimento [UML, 2006]. Será usado na modelagem da aplicação do celular como para o servidor da aplicação (PC) o diagrama de caso de uso e o diagrama de classes e para a ilustração dos cenários será usado o diagrama de atividades.

### **5.2.1. Diagrama de Caso de Uso**

Segundo [FUR *apud* Gomes, 2003] diagramas de caso de uso fornecem um modo de descrever a visão externa do sistema e suas interações com o mundo exterior, representando uma visão de alto nível de funcionalidade intencional mediante o recebimento de um tipo de requisição do usuário [UML, 2006].

### **5.2.2. Diagrama de Classe**

Segundo [FUR *apud* Gomes, 2003] o diagrama de classe é a essência da UML, que trata de uma estrutura lógica estática em uma superfície de duas dimensões mostrando uma coleção de elementos declarativos de modelo, como classes, tipos e seus respectivos conteúdos e relações.

### 5.2.3. Diagrama de Atividades

O objetivo do diagrama de atividades é mostrar o fluxo de atividades em um único processo. O diagrama mostra como uma atividade depende uma da outra.

### 5.3. Modelagem do Banco de Dados

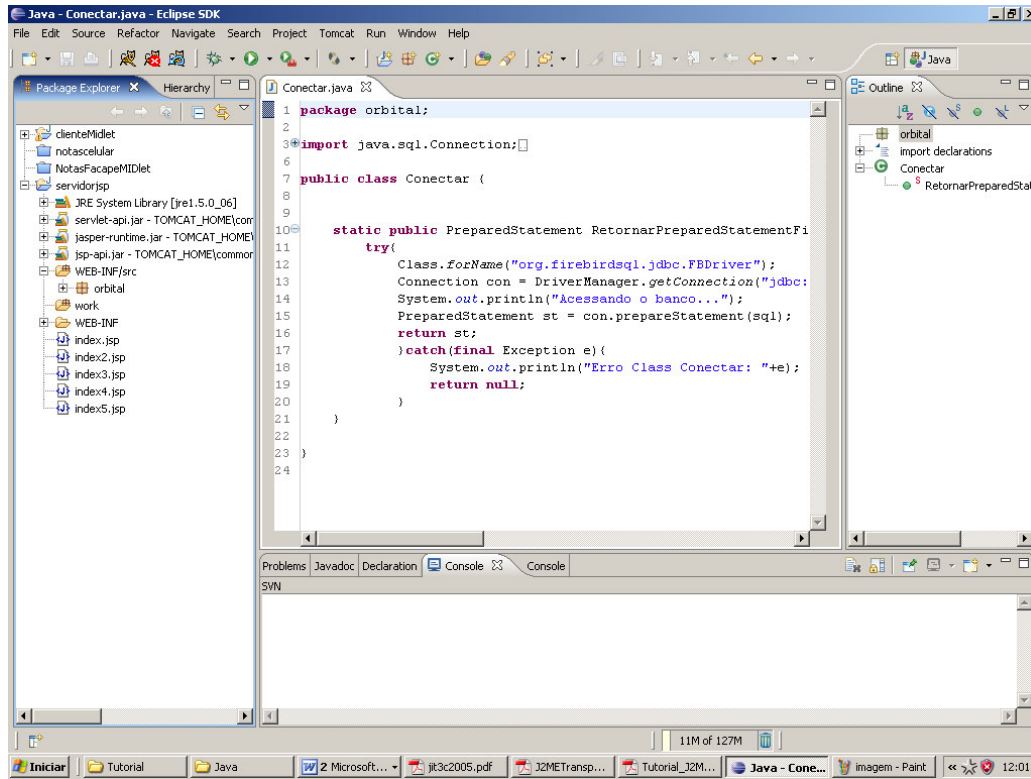
A modelagem do banco de dados será desenvolvida de acordo com o modelo relacional. Serão criadas as tabelas na ferramenta específica e a partir daí cria-se o banco de dados e registrando – o para sua utilização no servidor da aplicação.

### 5.4 Técnicas e Ferramentas utilizadas

Para implementação do protótipo do celular será utilizada a plataforma J2ME com o perfil MIDP 1.0 conforme citado na seção 4.1.3.1 do capítulo 4. A ferramenta utilizada será o Eclipse SDK 3.2, um ambiente de desenvolvimento de aplicações Java mostradop na Figura 16, com os plugins J2ME Wireless Toolkit, Edit JSP e Control Tomcat servindo de ambiente para a compilação e execução de aplicações J2ME através de emuladores de dispositivos móveis e páginas JSP. O J2ME Wireless Toolkit requer o J2 SDK e foi utilizado na versão 5.0.\_06 O emulador escolhido foi o *DefaultGrayPhone* [DGP , 2006], mostrado na Figura 17.

O servidor da aplicação será desenvolvido na ferramenta eclipse SDK na linguagem Java. O Apache Tomcat é um servidor de aplicações Java para web. É distribuído como software livre e desenvolvido como código aberto dentro do conceituado projeto Apache Jakarta e oficialmente endossado pela Sun [SUN BRASIL, 2006] para as tecnologias JavaServer Pages (JSP) citado no capítulo 4 seção 4.1.2. O Tomcat tem a capacidade de atuar também como servidor web/HTTP. O Apache Tomcat será utilizado como ambiente servidor para o servidor da aplicação[TOMCAT, 2006].

O banco de dados será desenvolvido na ferramenta SQL Manager 2005 [ SQLM,2006]e o banco utilizado foi Firebird 1.5 [Firebird, 2006 ].



**Figura 16 - Eclipse SDK 3.2**



Fonte: [SCHAEFER, 2004]

**Figura 17 - Emulador DefaultGrayPhone**



## CAPÍTULO 6

### DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo, serão mostrados os resultados do trabalho desenvolvido. O referencial teórico em que o trabalho se baseou está focado no capítulo 4 - Tecnologias e arquiteturas para o desenvolvimento de aplicações para celular e foram empregadas técnicas de projeto de pesquisa e a construção de cenários e prototipação entre outras etapas conforme citado no capítulo 5.

#### 6.1. Observação do Usuário

Com base em observações feitas por usuários (estudantes) do ensino preparatório para vestibulares que fazem testes simulando as provas seletivas nos cursos respectivos, notou-se que:

- Os usuários absorvem este processo com muita realidade, levando o fator emocional a interferir em seus resultados;
- Alguns usuários têm receio de participar temendo que seu resultado seja inferior ao esperado, prejudicando seu próprio desenvolvimento.

#### 6.2. Descrição do Cenário atual

Atualmente nos cursos preparatórios para vestibulares são realizados testes simulando provas do processo seletivo para o ingresso de alunos nas universidades. Esse processo é realizado no decorrer do ano com o intuito de demonstrar ao aluno como será a real situação dele diante do processo seletivo. Com isso é avaliado o aprendizado, o tempo gasto na resolução desses testes como também o sistema emocional diante desta situação. Para o professor é muito importante a realização dessas provas, visto que é uma forma de forçar o aluno a estudar mais e mostrá-lo o resultado para, diante disso, o aluno buscar um esforço ainda maior para alcançar no final o resultado almejado. As concorrências desses processos seletivos estão aumentando a cada ano e o nível dos candidatos também, com isso, é positiva a realização destas etapas de provas que são divididas de acordo com o conteúdo dado e de acordo com o sistema de cada instituição.

A seguir demonstramos através do diagrama de atividade e cenários, a realização dessas ações em relação ao professor e ao aluno.

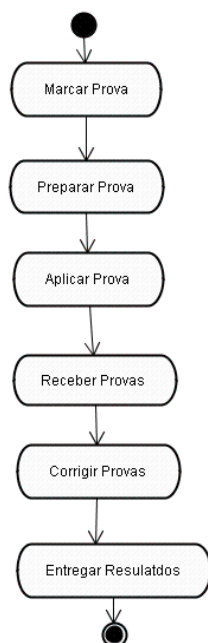
### 6.2.1. Cenário 1 - *Realização de um simulado por parte do Professor*

O cenário atual descreve todos os passos para a realização de um simulado, desde a etapa que o professor marca o dia da prova até sua aplicação e recebimento para entrega dos resultados, conforme mostrado na figura 17.

**Ator:** Professor

**Ambiente:** Sala de aula

**Roteiro:** O professor deseja realizar um simulado Então ele marca o dia da realização do simulado. Ao chegar o dia, o professor entra na sala de aula, aguarda todos os alunos entrarem e sentarem nos seus devidos lugares e começa a entregar as provas. Ao concluir, comunica aos alunos que o tempo será marcado e após concluída a hora determinada recolherá as provas. O professor comunica que não serão permitidas consultas e tampouco a utilização de aparelhos e/ou máquinas para realização dos testes. O professor aguarda os alunos resolverem as respectivas provas e ao concluí-las no o tempo, ele recolhe, e levando-as para serem corrigidas para poder dar o resultado ao aluno.



**Figura 18 - Diagrama de Atividades do professor**

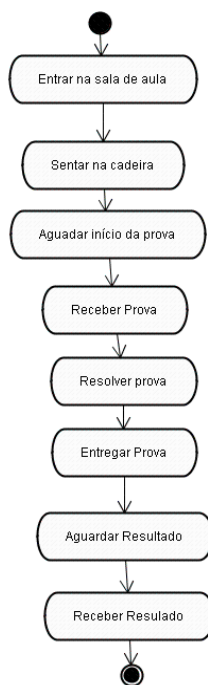
### 6.2.2. Cenário 2 – Realização de um simulado por parte do Aluno

O cenário atual descreve como o aluno realiza a prova conforme mostrado na figura 18.

**Ator:** Aluno

**Ambiente:** Sala de aula

**Roteiro:** O aluno deseja realizar um simulado. Ele entra na sala de aula, senta em seu lugar e aguarda ansioso o professor entregar a prova. Logo após, recebê-la ele aguarda o sinal do professor para começar a resolvê-la. O aluno sabe que seu tempo é limitado para resolver a prova, com isso fica nervoso, são muitas questões para resolver, o receio de não dar tempo, são fatores que podem atrapalhar seu desempenho. Terminada a prova ele ainda tem que passar as respostas para o gabarito. Quando o professor comunica que o tempo terminou o aluno entrega imediatamente e sai da sala. O aluno aguarda o resultado ansioso. Com o passar de alguns dias o aluno recebe o resultado.



**Figura 19 - Diagrama de Atividades do Aluno.**

### 6.3. Sistema Proposto

Com base nas observações dos usuários há uma diminuição de frequência nestas etapas de realização de provas devido à carga de *estress* devido à aproximação das provas reais. O aluno não assimila a idéia de um teste apenas e o receio de resultados negativos afasta ainda mais prejudicando o processo de ensino-aprendizagem. Com base no estudo realizado propõe-se a implementação de uma aplicação para celular que disponibilize provas para que o aluno possa realizá-las a qualquer hora, em qualquer lugar. Essas provas servirão para aumentar o nível de aprendizado do aluno. Inicialmente a disciplina escolhida é química, serão inseridas três provas focando os conteúdos da matéria que é dividido em tópicos, química geral, físico-química e química orgânica respectivamente relacionados à prova 1, 2 e 3, contendo cinco questões objetivas cada totalizando 15 questões. O intuito desta aplicação não é substituir o contexto atual e sim adicionar mais um incremento no processo de ensino-aprendizado, visto que a inserção de tecnologia tem contribuído muito nesta área educacional conforme pesquisa realizada e focada no capítulo 3.

A seguir será demonstrada através de cenários a aplicação proposta.

#### 6.3.1 Cenários Futuros

O papel do professor neste cenário é passar para o operador da aplicação as provas para serem inseridas no banco de dados. O cenário a ser descrito será apenas do aluno que utilizará a aplicação.

#### 6.3.2. Cenário Futuro – *Realização do simulado pelo aluno através aplicação.*

**Ator:** Aluno

**Ambiente:** Ambiente virtual do celular.

**Roteiro:** O aluno está em um veículo, no banco do passageiro, indo para casa. Ele aproveita este tempo e pega seu celular e acessa as provas e começa a resolver. À medida que o aluno vai resolvendo as questões, vai obtendo o resultado, se está acertando ou errando. Com isso ele percebe se está tendo algum aproveitamento das aulas, ao mesmo tempo observa que as questões vão sendo assimiladas. Ao parar o veículo o aluno sai da aplicação.

#### **6.4. Requisitos não funcionais**

Esta seção apresenta requisitos desejáveis respectivamente a aplicações educacionais em dispositivos móveis.

**RNF01** – O treinamento pode ser realizado em qualquer lugar e a qualquer momento, e de uma forma ininterrupta;

**RNF02** – A interatividade é outra característica importante presente nos treinamentos desenvolvidos, nos quais o aluno tem participação ativa.

**RNF03** – A aplicação permite um aprendizado baseado em experiências, na qual o aluno pode controlar ativamente o ambiente e observar diretamente o comportamento dos objetos virtuais.

**RNF04** – Facilidade de uso.

**RNF05** – Funcionamento em qualquer dispositivo móvel que suporte a tecnologia J2ME e que possibilite conexão com Internet.

#### **6.5.Requisitos funcionais**

Para o desenvolvimento de aplicações educacionais que satisfaçam as características requeridas na seção anterior, a aplicação deve atender a determinados requisitos funcionais:

**RF01** – Efetuar login – Permitir a verificação dos usuários cadastrados no sistema.

**RF02** – Selecionar Provas

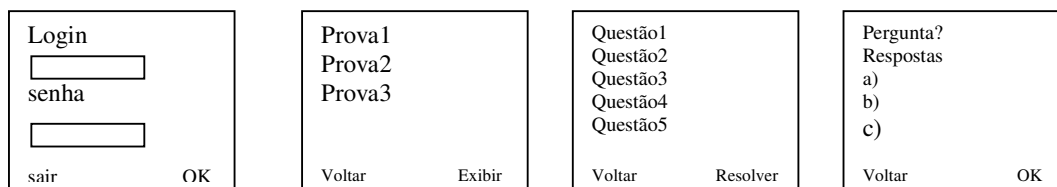
**RF03** – Escolher Questões

**RF04** – Responder Questões

#### **6.6. Protótipo de baixa fidelidade 1**

O protótipo, inicialmente, foi montado através de telas de papel para termos uma visão de como poderia ser a aplicação, descrevendo os passos necessários visando à usabilidade do usuário conforme foi citado no capítulo 5 seção 5.1.4.

A seguir demonstraremos as telas iniciais em papel na figura 20:



**Figura 20 – Principais telas de papel da aplicação.**

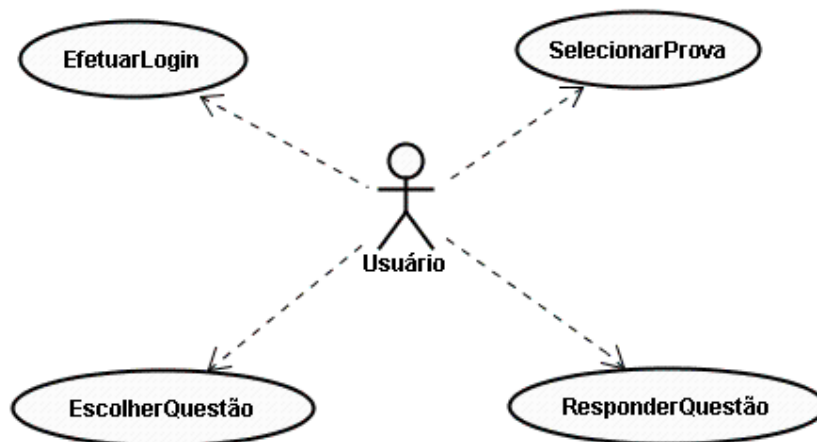
## 6.7. Modelagem do Protótipo2

Após observações dos usuários ao primeiro protótipo, criou-se o segundo protótipo, inteiramente funcional, ou seja, não há apenas telas estáticas, mas um completo acesso a banco de dados. Ilustramos a modelagem do sistema através do padrão UML [UML, 2006], onde serão apresentados os itens abaixo:

- Diagrama de Caso de Uso
- Diagrama de Classe

### 6.7.1. Diagrama de Caso de Uso

O diagrama de caso de uso do sistema será descrito abaixo:



**Figura 21 - Diagrama de caso de uso da aplicação.**

Os casos de uso indicados no diagrama são os seguintes:

- ***EfetuarLogin***: Permite a verificação dos usuários cadastrados no sistema, liberando o acesso às provas.
- ***SelecionarProva***: Permite a disponibilização das provas, com seu respectivo título.

- **EscolherQuestão:** Permite a liberação das questões da respectiva prova, para que o usuário escolha a qual deseja responder.
- **ResponderQuestão:** Permite o acesso a pergunta com as respectivas respostas para que o usuário possa responder.

### 6.7.2. Diagrama de Classe

A figura 21 mostra as classes utilizadas no projeto do protótipo proposto (Celular).

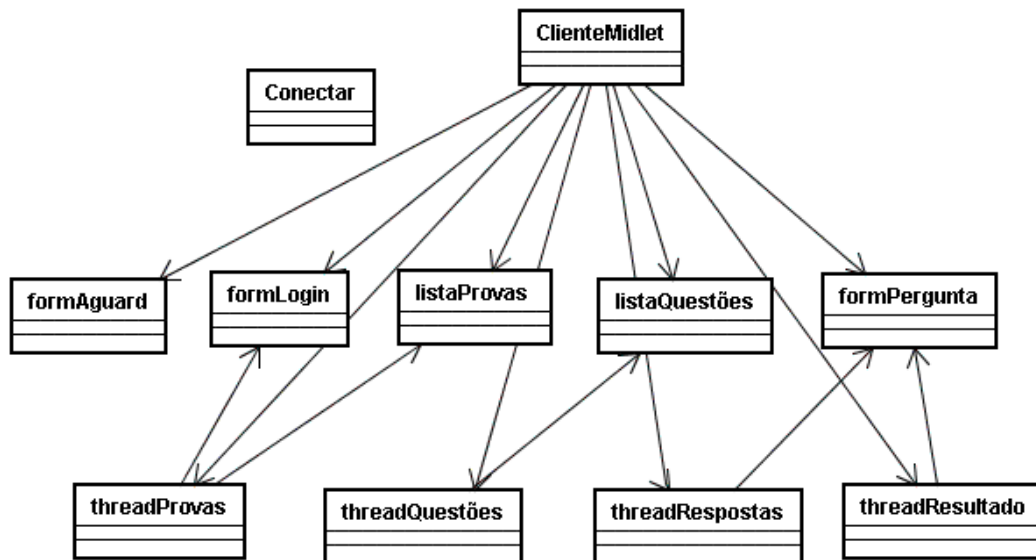
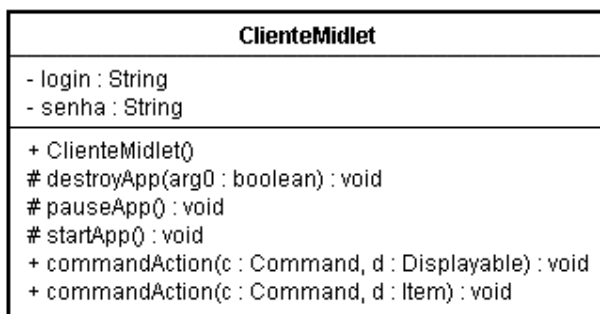


Figura 22 - Diagrama de Classes



**Classe ClienteMidlet** – Esta é a classe principal da aplicação pois é responsável pelo gerenciamento das ações dos forms e listas da aplicação conforme descrito no capítulo 4. Os métodos mostrados nesta classe: destroyApp(), pauseApp() e startApp() são métodos

específicos da classe Midlet para que a aplicação funcione conforme descrito no capítulo 4 seção 4.1.2.4.

<b>Conectar</b>
- <u>urlServidor : String = "127.0.0.1"</u>
- <u>portaServidor : String = "8080"</u>
+ <u>getPortaServidor() : String</u>
+ <u>getUrlServidor() : String</u>

**Classe Conectar** – É a classe que guarda o endereço (IP) e a porta do servidor, conforme mostrado na figura acima.

<b>formAguarde</b>
+ formAguarde(listener : CommandListener)
+ getCmdCancelar() : Command

**Classe FormAguarde** – Informa ao usuário para aguardar enquanto faz a verificação da possibilidade de conexão com o servidor

<b>formLogin</b>
+ formLogin(listener : CommandListener)
+ getCmdExit() : Command
+ getCmdOk() : Command
+ getlogin() : TextField
+ getsenha() : TextField

**Classe formLogin** – Classe que trata da entrada do login e senha para controle da classe ClienteMidlet.

<b>listaProvas</b>
+ listaProvas(listener : CommandListener, mensagem : String)
+ getCmdExibir() : Command
+ getCmdVoltar() : Command

**Classe listaProvas** – Classe que disponibiliza a lista de provas para controle da classe ClienteMidlet.

<b>listaQuestoes</b>
+ listaQuestoes(listener : CommandListener, mensagem : String)
+ getCmdResolver() : Command
+ getCmdVoltar() : Command

**listaQuestoes** – Classe que disponibiliza a lista de questões para controle da classe ClienteMidlet.



<b>formPergunta</b>
+ formPergunta(listener : ItemCommandListener) + setChoice(lista : ChoiceGroup) : void + getChoice() : ChoiceGroup + getPergunta() : StringItem + setPergunta(pergunta : StringItem) : void + getCmdOK() : Command + getCmdVoltar() : Command

**FormPergunta** – Esta classe disponibiliza as perguntas e respostas das questões listadas. É controlada pela classe ClienteMidlet.

Para serem exibidas, cada form e lista se ligará as classes Threads respectivas das suas classes citadas acima conforme apresentada no diagrama de classe para poderem ser manipuladas em tempo de execução (*Runtime*).

<b>threadProvas</b>
- URL : String
+ threadProvas(display : Display, formLogin : Form, lista : List, login : String, senha : String) + run() : void + <u>urlEncode(s : String) : String</u> + pararConexao() : void + retornarHtml() : HttpConnection

**ThreadProvas** – Esta classe controla a exibição das classes formLogin e listaProvas, mas é controlada pela ClassMidlet..

<b>threadQuestoes</b>
- URL : String
+ threadQuestoes(display : Display, listaQuestoes : List, login : String, senha : String, prova : String) + run() : void + <u>urlEncode(s : String) : String</u> + pararConexao() : void

**ThreadQuestões** – Esta classe controla a exibição da classe listaQuestões, mas é controlada pela classe ClienteMidlet.

<b>threadResposta</b>
- URL : String
+ threadResposta(display : Display, formPergunta : Form, login : String, senha : String, questao : String) + run() : void + <u>urlEncode(s : String) : String</u> + pararConexao() : void

**ThreadRespostas** – Esta classe controla a exibição da classe formPergunta mas é controlada pela classe ClienteMidlet.

<b>threadResultado</b>
- URL : String ~ certo : String
+ threadResultado(display : Display, formPergunta : Form, login : String, senha : String, resposta : String) + run() : void <u>+ urlEncode(s : String) : String</u> + pararConexao() : void

**ThreadResultado** – Esta classe controla a exibição do alertas de certo ou errado da classe formPergunta referente as respostas das questões. É controlada pela classe ClienteMidlet.

Parte do código fonte mostrando a classe ClinteMidlet Instanciando os formulários e as listas que serão usadas na aplicação ( no final desse trabalho estará anexo o código fonte completo).

```
public ClienteMidlet() {
    super();

    display = Display.getDisplay(this);
    formLogin = new formLogin(this);
    formaguarde = new formAguarde(this);
    formPergunta=new formPergunta(this);
    listaProvas = new listaProvas(this,null);
    listaQuestoes = new listaQuestoes(this,null);
    display.setCurrent(formLogin);// mostra na tela

}
```

Parte do Código Fonte mostrando o método que fica responsável pelo controle das ações dos itens do formulário de resposta e mostra como habilita o botão OK das respostas das perguntas e chama o resultado (no final desse trabalho estará anexo o código fonte completo).

```

public void commandAction(Command c, Item d) {

    if(c==formPergunta.getCmdOK()){// habilita o
botão Ok

        threadResultado = new
threadResultado(display, formPergunta, login,senha,
formPergunta.getChoice().getString(formPergunta.getChoice().
getSelectedIndex()),listaQuestoes);

        Thread thread4 = new
Thread(threadResultado);

        //inicia a Thread
        thread4.start();
        display.setCurrent(formaguarde);
    }

public void commandAction(Command c, Item d) {

    if(c==formPergunta.getCmdOK()){// habilita o
botão Ok das respostas das perguntas e chama o resultado

        threadResultado = new
threadResultado(display, formPergunta, login,senha,
formPergunta.getChoice().getString(formPergunta.getChoice().
getSelectedIndex()),listaQuestoes);

        Thread thread4 = new
Thread(threadResultado);

        thread4.start();
        display.setCurrent(formaguarde);
    }

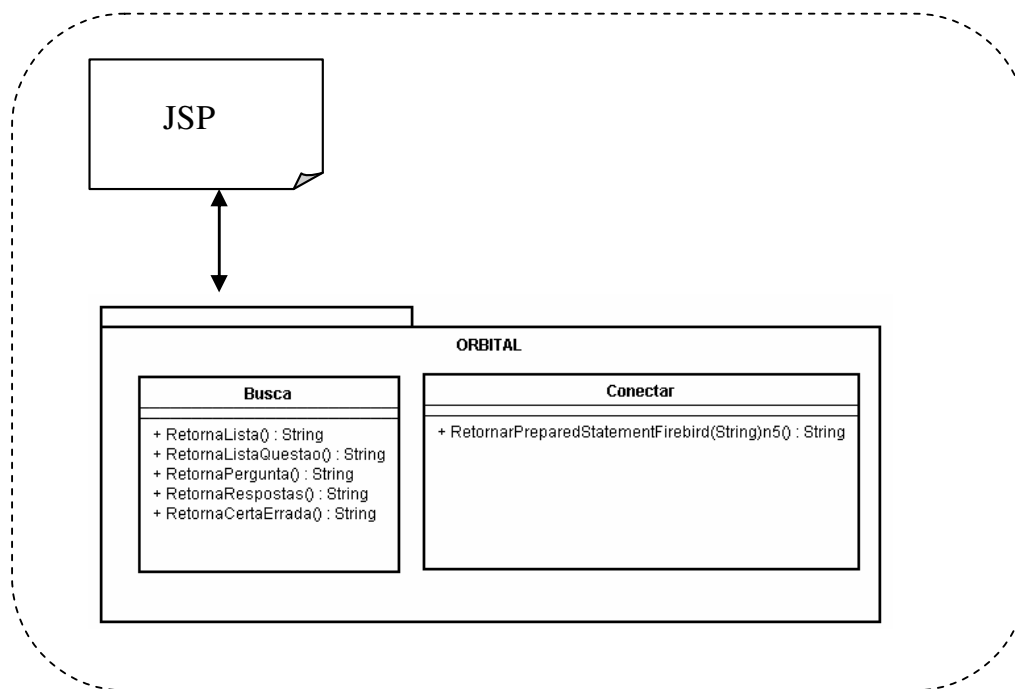
(threadResultado);

    thread4.start();
    display.setCurrent(formaguarde);
}

```

## 6.8. Modelagem do Servidor da Aplicação

O servidor da aplicação foi construído em um pacote chamado Orbital contendo duas classes, a classe Busca e a Classe Conectar. A classe conectar trata da conexão do servidor com o Banco de Dados. Foi utilizada a tecnologia JSP para fazer o tratamento das requisições recebidas e enviadas entre a aplicação cliente e o servidor através da Classe Busca. A figura 23 apresenta o diagrama do Servidor da Aplicação conforme citado no capítulo 5 seção 5.2.:



**Figura 23 – Diagrama do Servidor da Aplicação**

**JSP** – possui método responsável pelo recebimento de parâmetros da classe clienteMidlet e enviado para classe busca a qual vai fazer as transações com o banco de dados.

**Classe Busca** – responsável pelo tratamento dos parâmetros recebidos através do JSP para fazer a verificação no banco de dados.

**Classe Conectar** – responsável pela conexão com o banco de dados.

## 6.9. Arquitetura da Aplicação

A aplicação proposta foi baseada no capítulo 4 seção 4.2.3 e 4.2.4., sendo arquitetura Cliente Servidor conforme mostra a figura 24:

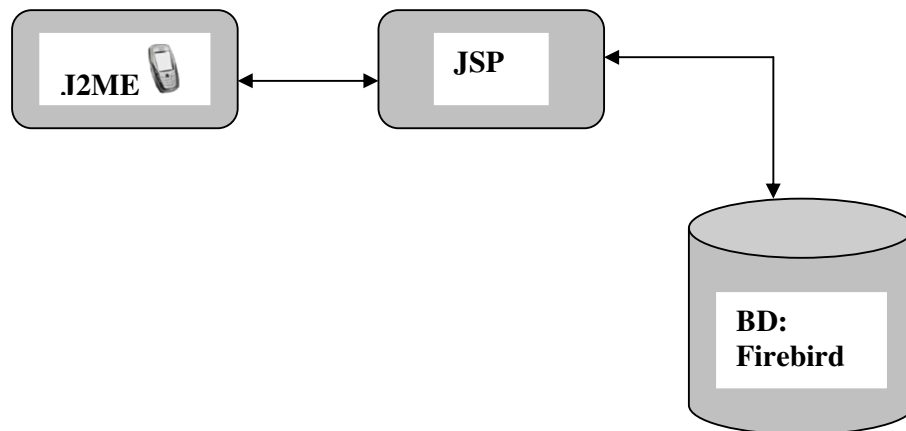


Figura 24 – Arquitetura da Aplicação

## 6.10. Modelagem do Banco de Dados

Na Figura 25 abaixo é mostrada a modelagem do banco com as respectivas tabelas.

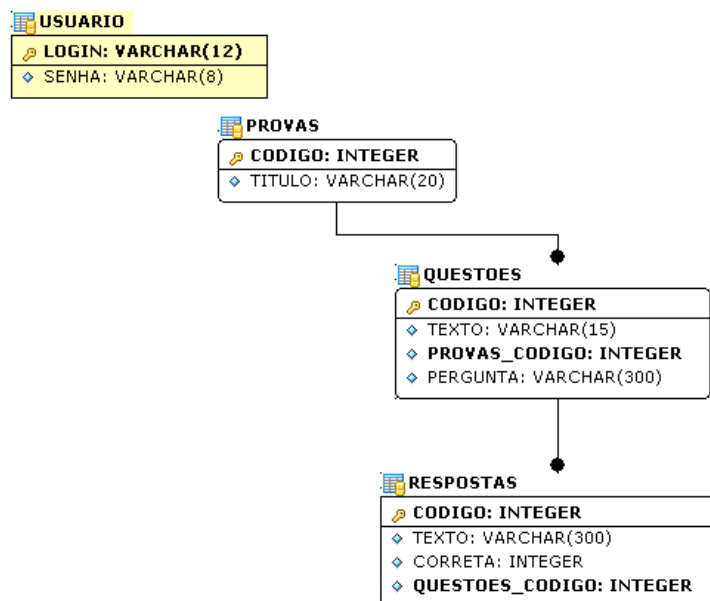


Figura 25 - Modelagem do Banco de Dados

### 6.10.1. Tabelas

#### USUÁRIO

CAMPO	TIPO	TAMANHO	CHAVE
LOGIN	C	12	*
SENHA	C	08	

#### PROVAS

CAMPO	TIPO	TAMANHO	CHAVE
CÓDIGO	N	03	*
TÍTULO	C	20	

#### QUESTÕES

CAMPO	TIPO	TAMANHO	CHAVE
CÓDIGO	N	03	*
TEXTO	C	20	
COD_PROVAS	N	03	
PERGUNTA	C	300	

#### RESPOSTAS

CAMPO	TIPO	TAMANHO	CHAVE
CÓDIGO	N	03	*
TEXTO	C	20	
COD_QUESTÕES	N	03	
CORRETA	N	01	

### 6.10.2. Script do Banco de Dados

```
CREATE TABLE USUARIO (
  LOGIN VARCHAR(12) CHARACTER SET NONE NOT NULL COLLATE NONE,
  SENHA VARCHAR(8) CHARACTER SET NONE COLLATE NONE);
```

```
ALTER TABLE USUARIO ADD CONSTRAINT USUARIO_PK PRIMARY KEY
(LOGIN);
```

```
CREATE TABLE PROVAS (
  CODIGO INTEGER NOT NULL,
  TITULO VARCHAR(20) CHARACTER SET NONE COLLATE NONE);
```

```
ALTER TABLE PROVAS ADD CONSTRAINT PROVAS_PK PRIMARY KEY
(CODIGO);
```

```

CREATE TABLE QUESTOES (
  CODIGO INTEGER NOT NULL,
  TEXTO VARCHAR(15) CHARACTER SET NONE COLLATE NONE,
  PROVAS_CODIGO INTEGER NOT NULL,
  PERGUNTA VARCHAR(300) CHARACTER SET NONE COLLATE NONE);

ALTER TABLE QUESTOES ADD CONSTRAINT QUESTOES_PK PRIMARY KEY
(CODIGO);

ALTER TABLE QUESTOES ADD CONSTRAINT FK_PROVAS FOREIGN KEY
(PROVAS_CODIGO) REFERENCES PROVAS (CODIGO);

CREATE TABLE RESPOSTAS (
  CODIGO INTEGER NOT NULL,
  TEXTO VARCHAR(300) CHARACTER SET NONE COLLATE NONE,
  CORRETA INTEGER,
  QUESTOES_CODIGO INTEGER NOT NULL);

ALTER TABLE RESPOSTAS ADD CONSTRAINT RESPOSTAS_PK PRIMARY
KEY (CODIGO);

ALTER TABLE RESPOSTAS ADD CONSTRAINT FK_QUESTOES FOREIGN
KEY (QUESTOES_CODIGO) REFERENCES QUESTOES(CODIGO);

```

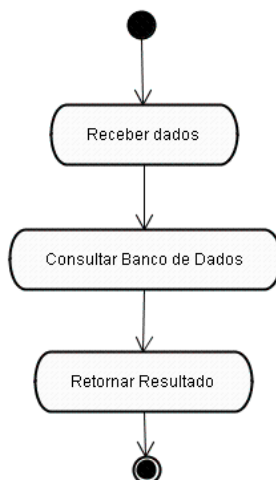
### 6.11. Operacionalidade da Implementação

Nesta seção, é apresentado o funcionamento da aplicação, a parte do servidor (PC) como a parte do Cliente (celular) e o funcionamento do protótipo (celular).

#### Aplicação

Servidor: A operacionalidade ocorre conforme diagrama descrito na Figura 25:

- Recebe os dados da aplicação cliente;
- Consulta o banco de dados;
- Retorna o *resultado*.

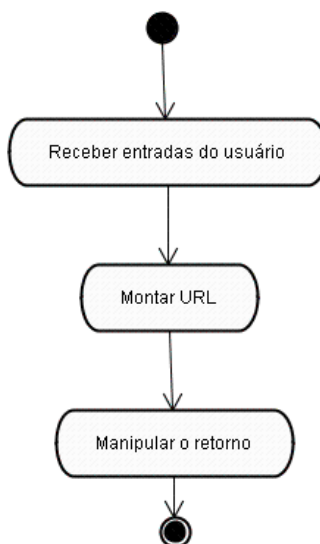


**Figura 26 – Diagrama de atividades do servidor (PC) da aplicação**

O servidor retorna uma string com um separador “-”.

Cliente: A operacionalidade ocorre conforme diagrama descrito na Figura 26:

- Recebe as entradas dos usuários;
- Monta a URL (*Universal Resource Locator*, ou seja, endereço de um recurso disponível na Internet) e envia para o Servidor; (Ex. `http://servidor:8080/index.jsp?login= "" senha= ""`) [URL, 2006].
- Manipula o retorno



**Figura 27 – Diagrama de Atividades do cliente (celular) da aplicação**



O Cliente J2ME separa a string em várias sub-string e organiza em uma lista.

O protocolo utilizado pela aplicação é o HTTP. É um protocolo *stateless* [STATELESS, 2006] de comunicação cliente-servidor: o cliente envia uma requisição para o servidor, esse processa a requisição e devolve uma resposta para o cliente.

Parte do Código do Servidor – Este código faz parte da classe busca do pacote Orbital conforme citado na seção 6.8 deste capítulo. Mostra método que recebe os parâmetros login e senha, faz a verificação, e retorna a lista de provas (no final desse trabalho estará em anexo o código fonte completo).

```
public class Busca {

    public String RetornaLista(String login, String senha)
    throws SQLException{

        try{
            // monta o SQL
            String sql = "select * from USUARIO where
            usuario.LOGIN='"+login+"' and usuario.SENHA='"+senha+"'";
            PreparedStatement st =
            Conectar.RetornarPreparedStatementFirebird(sql);
            // retorna um o resultado da busca
            ResultSet rs = st.executeQuery();

            String retorno = "";

            //Verifica se houve retorno (se o usuario
            existe)
            if(rs.next()){
                PreparedStatement st2 =
                Conectar.RetornarPreparedStatementFirebird(
                "select Provas.TITULO from
                PROVAS");
                ResultSet rs2 = st2.executeQuery();
                //Monta uma String com a lista das
                provas separadas por "-"
                while (rs2.next()){
                    retorno = retorno + rs2.getString(1) +
                    "-";
                }
            }else{
                retorno = null;
            }
            return retorno;

        }catch(SQLException e){System.out.print("Erro Class
        Busca: "+e);}
        return null;
    }
}
```

Código Fonte da classe conectar que faz parte do pacote orbital conforme citado na seção 6.8 deste capítulo. Esta classe é responsável pela conexão com o banco de dados (no final desse trabalho estará em anexo o código fonte completo).

```
package orbital;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class Conectar {

    static public PreparedStatement
    RetornarPreparedStatementFirebird(String sql){
        try{

            Class.forName("org.firebirdsql.jdbc.FBDriver");// Seta
            o drive de conexão do Banco de DADOS
            //Seta o caminho onde se encontra o banco de
            dados
            Connection con =
            DriverManager.getConnection("jdbc:firebirdsql:localhost:C:/D
            ocuments and
            Settings/Micro/Desktop/Projeto_celular/DB/SIMULADO.FDB", "SYS
            DBA", "masterkey");
            System.out.println("Acessando o banco...");
            //gera o PreparedStatement
            PreparedStatement st =
            con.prepareStatement(sql);
            //retorna o PreparedStatement
            return st;
        }catch (final Exception e){
            System.out.println("Erro Class
            Conectar: "+e);
            return null;
        }
    }

}
```

O funcionamento do protótipo (celular). É composto de sete telas:

- a) Tela de login;
- b) Tela de erro de login;
- c) Tela de aviso (*It is OK to Use Airtime?*);
- d) Tela de lista de provas;
- e) Tela de lista de questões da respectiva prova;
- f) Tela da pergunta com as respostas;

g) Tela de aviso (certa ou errada).

Na Figura 27 é mostrada o protótipo com a tela de login.

a) Nesta **tela de login** o usuário insere as informações referentes ao seu login e senha já cadastrados na instituição. As setas direcionais do botão SELECT (mostrado abaixo na figura 27) servem para a navegação entre os campos. As opções disponíveis nos botões são:

- OK – quando o usuário clicar neste botão então a ocorrerá a verificação do login e senha, se não estiver cadastrado ou digitado errado vai para tela de erro.
- SAIR. Saída da aplicação

b) a **tela de erro** é exibida com a mensagem “Não foi possível fazer a conexão... por favor, verifique a senha e o login” e, logo retorna para a tela de login onde o usuário corrigirá.

c) Confirmado o login e senha efetua-se uma conexão ao servidor e envia a informação do login e senha cadastrada. Logo após vem uma **tela de aviso** (“*It is OK to use airtime?*”), sendo esta uma tela padrão para avisar o usuário que a aplicação utilizará a Internet para enviar e receber dados e pergunta se o usuário está pronto para enviar na hora, dando duas opções: *Yes* (permite fazer a conexão com servidor) ou *No* (encerra a aplicação).



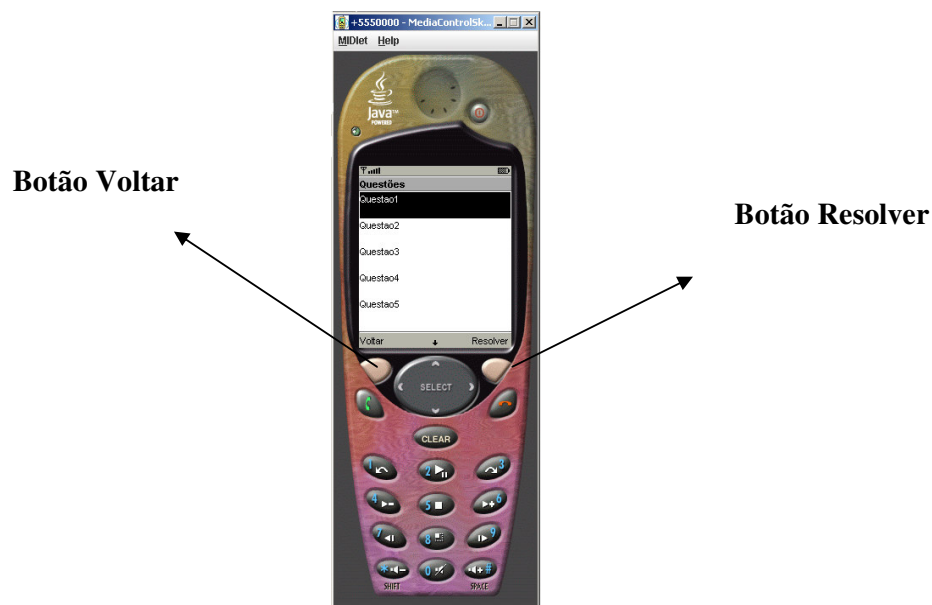
**Figura 28- Tela de login**

d) Se a opção for positiva aparece a tela com a lista de provas. Nesta tela, aparece dois botões “*voltar*” que retorna a tela de login e “*exibir*” que mostra a próxima tela com a lista de questões. A figura, abaixo mostra o protótipo com a lista de provas



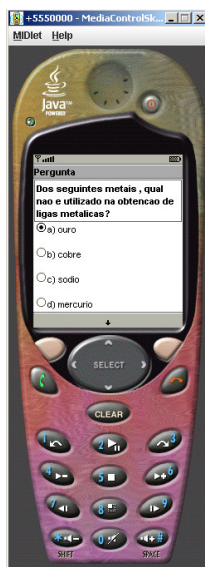
**Figura 29- Tela de lista de provas**

e) Na tela de lista de questões, aparecerá dois botões “*voltar*” que retorna a lista de provas e “*resolver*” que exibe a pergunta com suas alternativas de respostas. Na figura abaixo, mostra o protótipo com a lista de questões.



**Figura 30- Tela de lista de questões**

f) Na tela da pergunta, com o botão SELEC, o aluno seleciona a alternativa da resposta desejada e clica no botão OK aparecendo a resposta na tela de aviso (certa ou errada). A figura abaixo mostra o protótipo com a pergunta e resposta de uma questão.



**Figura 31 - Tela de Pergunta com as respectivas respostas**

g) Na tela de aviso, se estiver errada, aparecerá a tela de aviso com alerta “errado”, se estiver correta, aparecerá a tela de aviso com o alerta “certa”. As figuras 29 e 30 mostram as telas de erro e acerto respectivamente.



**Figura 32- Tela de Aviso - “errado”**



**Figura 33 – Tela de aviso - “certa”**

## CAPÍTULO 7

### CONCLUSÃO

Neste trabalho, foi possível a realização de um estudo sobre *mobile Learning*, uma tecnologia inovadora voltada para aprendizagem através de dispositivos móveis. Essa tecnologia surgiu a partir do *e-learning*, ou seja, educação a distância, utilizando tecnologias computacionais para disseminar a educação trazendo oportunidades e criando novas possibilidades de mudança no processo ensino-aprendizagem. Com a utilização de dispositivos móveis, acrescentou-se um novo incremento nesta área em busca de novos desafios neste processo além de aumentarmos a funcionalidade destes equipamentos.

Foi possível também conhecer ainda mais estes dispositivos, constatando-se suas limitações em relação às aplicações *Wireless* para dispositivos sem fio, apresentando ainda muitos desafios aos profissionais da computação, principalmente em relação às aplicações padrões para PC. Existem desafios no campo sociológico, como o fato de que as atividades realizadas com um celular ainda são fundamentalmente diferentes das realizadas com um PC e a resistência dos usuários pelo fato de que a interação com o aplicativo no celular (como telas e teclado) é muito mais limitada que num PC. Além disso, tem o custo para os usuários, que é outro fator influente na aquisição destas aplicações.

A realização deste trabalho trouxe muitas contribuições, pois além de acrescentar conhecimentos sobre o assunto estudado foi possível demonstrar uma aplicação voltada para aprendizagem através de um celular que o aluno pode acessar um banco de provas de qualquer lugar e resolvê-las, assimilando conteúdos e colocando em prática através desse exercício a aprendizagem. Uma dificuldade encontrada foi em relação às questões colocadas no banco de dados, visto que, por ser um dispositivo que apresenta uma tela pequena, com espaço limitado de memória, as questões escolhidas foram objetivas, não foram inseridas questões com cálculos e com respostas grandes. E também como a disciplina escolhida foi química, existem algumas fórmulas com carbonos que não puderam ser utilizadas pela dificuldade de inserção no banco.

Conclui-se que apesar de muitas dificuldades é possível o desenvolvimento de aplicações para esta área de ensino como para outras áreas, a qual no decorrer deste

trabalho foi possível observar através de exemplos citados, e que com a evolução das tecnologias computacionais e também de aparelhos celulares novas possibilidades serão criadas para a utilização destes dispositivos para o *m-Learning*.

## **TRABALHOS FUTUROS**

Sugerem-se para trabalhos futuros pesquisas relacionadas à aceitabilidade destes dispositivos neste processo de ensino aprendizagem, tanto pelos alunos como pelos professores.

Aperfeiçoar esta aplicação adicionando resultados estatísticos de acertos e erros e tempo de resolução das provas. Inserir outras disciplinas também para serem testadas como: biologia, português e línguas estrangeiras. A inserção de imagens nas questões também pode ser trabalhada como inovação nesta aplicação.

Sugere-se o desenvolvimento de um estudo de caso real de usabilidade para ser validado junto a alunos de instituições de ensino de primeiro, segundo grau e pré-vestibulandos ou com profissionais de campo. Neste trabalho, seria necessário o envolvimento de profissionais da educação para definição de aspectos pedagógicos necessários e de acordo com o público alvo escolhido.

Na expectativa de novos modelos, com mais recursos como áudio, vídeo, banda larga entre outros onde será possível incrementar cada vez mais as aplicações tornando a receptividade dos usuários mais aguçados diante destas inovações.

Por fim sugere-se a criação de uma aplicação que permita ao professor criar seu simulado, podendo inserir no banco de dados as provas.

## REFERÊNCIAS BIBLIOGRÁFICAS

[AER, 2006] PRESSMAN, Roger. Análise e Especificação de Requisitos. Disponível em: <http://www.dimap.ufrn.br/~jair/ES/es991c3.html>. Acesso em novembro de 2006.

[ANATEL, 2006] Telefonía móvel se aproxima das 97 milhões de habilitações em outubro. Disponível em: [http://www.anatel.gov.br/Tools/frame.asp?link=biblioteca/releases/2006/release\\_20\\_11\\_2006mm](http://www.anatel.gov.br/Tools/frame.asp?link=biblioteca/releases/2006/release_20_11_2006mm). Acesso em novembro de 2006

[AMORIM, BORGES, 2005] AMORIM, Andrea Rodrigues de; BORGES, Karen Selbach – Desenvolvimento de aplicações móveis com J2ME. Disponível em: <http://karen.confrariadojava.org/bib/jit3c2005.pdf>. Acesso em setembro de 2006.

[AZEVEDO, 2005] AZEVEDO, Filipe Levi Barros de. Prototipação rápida no ciclo de design interativo de aplicações multimídia para formação de professores. Trabalho de conclusão de Curso de Ciência da Computação – Centro de Informática (UFPE). Disponível em: [www.cin.ufpe.br/flba](http://www.cin.ufpe.br/flba). Acesso em dezembro de 2006.

[BARROS, 2005], Edgar M. Filho - VirTraM: Um Framework para o Desenvolvimento de Treinamentos Utilizando Realidade Virtual em Dispositivos Móveis, 2005. <http://www.mcc.ufc.br/disser/EdgarMarcal.pdf>. Acesso em agosto de 2006

[BERNARDO, BIELAWSKY, 2003] BERNARDO, Hugo de Magalhães; BIELAWSKY, Larry. O Futuro do *e-Learning*. <http://www.spi.pt/madilearning/manual6/OFuturodoeLearning-formando.pdf>. Acesso em agosto de 2006.

[BLUETOOTH, 2006] O que é *Bluetooth*? Disponível em: <http://www.boadica.com.br/layoutdica.asp?odigo=383>. Acesso em novembro de 2006.

[CF, 2006] *Compact Flash*. Disponível em: <http://en.wikipedia.org/wiki/compactflash>. Acesso em novembro de 2006.

[CM, 2006] Conectividade Móvel. Disponível em: [http://www.mobilezone.com.br/conect\\_wwan.htm](http://www.mobilezone.com.br/conect_wwan.htm). Acesso em outubro de 2006.

[CÔRTEZ, LIFSCHITZ, 2006] CÔRTEZ, Sérgio da Costa; LIFSCHITZ, Sérgio. Banco de Dados para um Ambiente de Computação Móvel. Disponível em: <http://www.unifev.edu.br/canais/docentes/luis/BDAmbientesCompMovel.pdf>. Acesso em outubro de 2006.

[COSTA, 2006] COSTA, Ricardo. *Tele-Experimentação Móvel (Mobile Remote Experimentation)* Considerações sobre uma área emergente no ensino à distância. Disponível em: <http://www.laboris.isep.ipp.pt/rjc/documentos/mre.pdf>. Acesso em setembro de 2006.



- [DGP, 2006] *DefaultGrayPhone*. Disponível em: <http://www.j2mepolish.org/devices/Generic/DefaultGrayPhone.html#multimedia> . Acesso em dezembro de 2006.
- [FALCÃO, 2004] FALCÃO, Taciana Pontual da Rocha. Modelagem de Soluções Ubíquas para uso em salas de aula no Ensino Fundamental. [http://www.cin.ufpe.br/~tg/2004-1/tprf\[1\].pdf](http://www.cin.ufpe.br/~tg/2004-1/tprf[1].pdf). Acesso em novembro de 2006
- [FIREBIRD, 2006] Firebird 1.5 – Disponível em: [http://www.comunidade-firebird.org/cflp/html\\_docs/001\\_FacSheet/Firebird%201.5%20Factsheet.htm](http://www.comunidade-firebird.org/cflp/html_docs/001_FacSheet/Firebird%201.5%20Factsheet.htm) . Acesso em dezembro de 2006.
- [GOMES, 2003] GOMES, Marco Antonio Pereira. Estudo e Prototipação de ferramenta para gerência de projetos, enfocando a gerência de aquisições. <http://www.ulbra.tche.br/~tcc-canoas/2003-2/marco.pdf> . Acesso em outubro de 2006.
- [GSM World, 2005] Estatística de Assinantes de Telefonia Celular. Disponível em: <http://www.gsmworld.com/news/statistics/index.shtml>. Acesso em novembro de 2006.
- [HTTP, 2006] *HyperText Transmission Protocol* – Protocolo de Transmissão por Hipertexto. Disponível em: [http://en.wikipedia.org/wiki/Protocolo\\_de\\_Transfer%C3%A7%C3%A3o\\_de\\_Hipertexto](http://en.wikipedia.org/wiki/Protocolo_de_Transfer%C3%A7%C3%A3o_de_Hipertexto). Acesso em dezembro de 2006.
- [IrDA, 2006] Termos técnicos GdH. Disponível em: <http://www.guiadohardware.net/termos/irda>. Acesso em 18/11/06. Acesso em novembro de 2006.
- [JAVA, 2006] *The Source for Java Developers*. Disponível em: <http://java.sun.com>. Acesso em novembro de 2006.
- [J2ME, 2005] CARNIEL, Juliano, TEIXEIRA, Clóvis – Apostila J2ME – versão1. 1- <http://www.engcomp.ufrn.br/~evebat/tutorialj2me1.pdf> . Acesso em agosto 2006.
- [JITL, 2006] Just in time Learning. Disponível em: <http://www.computerworld.com/news/2000/story/0,11280,44312,00.html>. Acesso em novembro de 2006.
- [KUHNEN, 2003] KUHNEN, Alex. Protótipo de uma Aplicação Lbs utilizando Gps conectado em Celular para consultar dados georeferenciados. <http://www.inf.furb.br/~pericas/orientacoes/WAPLBS2003.pdf> . Acesso em novembro de 2006.
- [LEH, 2006] Os livros eletrônicos e os handhelds. Disponível em: <http://ebookeyult.com.br>. Acesso em novembro de 2006.
- [LINUX OS, 2006] *Mobile Linux OS aims at lower cost allphones*. Disponível em: <http://linuxdevices.com>. Acesso em novembro de 2006.

[MARÇAL, SANTOS, VIDAL, ANDRADE, RIOS, 2005] MARÇAL, Edgar; ANDRADE, Rossana; RIOS, Riverson – Aprendizagem utilizando Dispositivos Móveis com Sistemas de Realidade Virtual. [http://www.cinted.ufrgs.br/renote/maio2005/artigos/a51\\_realidadevirtual\\_revisado.pdf](http://www.cinted.ufrgs.br/renote/maio2005/artigos/a51_realidadevirtual_revisado.pdf) . Acesso em setembro de 2006.

[MARÇAL, SANTOS, VIDAL, ANDRADE, RIOS, 2004] MARÇAL, Edgar; SANTOS, Raniery; VIDAL, Creto; ANDRADE, Rossana; RIOS, Riverson - museuM: Uma Aplicação de m-Learning com Realidade <http://www.lia.ufc.br/~great/artigos/museuM.pdf> . Acesso em setembro de 2006.

[MEIRELLES, TAROUÇO, ALVES, 2006] MEIRELLES, Luiz Fernando T.; TAROUÇO, Liane Margarida R.; ALVES, Carlos Vinicius R.- Telemática aplicada a aprendizagem com mobilidade. [http://www.cinted.ufrgs.br/renote/nov2004/artigos/a13\\_telematica-aplicada.pdf](http://www.cinted.ufrgs.br/renote/nov2004/artigos/a13_telematica-aplicada.pdf) . Acesso em setembro de 2006.

[MMS, 2006] *Multimidea Messaging Service* Disponível em: <http://en.wikipedia.org/wiki/mult>. Acesso em outubro de 2006.

[MP3, 2006] MPEG-1/2 Áudio Layer 3. Disponível em: <http://pt.wikipedia.org/wiki/MP3>. Acesso em novembro de 2006.

[MS, 2006] *Memory Stick*. Disponível em: [http:// pt.wikipédia.org/wiki/memory\\_stick](http://pt.wikipédia.org/wiki/memory_stick) . Acesso em novembro de 2006.

[NTEL, 2006] Novas Tecnologias no *e-learning*: Desafios e Oportunidades para o Design. Disponível em: [http:// www.abed.org.br](http://www.abed.org.br). Acesso em 18/11/06.

[OQP, 2006] O que é prototipação? Disponível em: <http://www.black-beans.com.br/prototyping.htm>. Acesso em novembro de 2006.

[ODBC, 2006] Uma visão geral sobre ODBC. Disponível em: <http://www.pr.gov.br/batebyte/edicoes/1996/bb53/odbc.htm>. Acesso em 01/12/06

[PALUDO, 2003], PALUDO, Lauriana – Um estudo sobre as tecnologias Java de desenvolvimento de aplicações móveis, 2003. [www.inf.ufsc.br/~leandro/ensino/esp/monografiaLaurianaPaludo.pdf](http://www.inf.ufsc.br/~leandro/ensino/esp/monografiaLaurianaPaludo.pdf) . Acesso em agosto de 2006.

[PASSOS, 2006] PASSOS, Jocélio de Oliveira Dantas. Design interativo de ferramenta de manipulação de objetos de aprendizagem de ambientes virtuais de ensino a distância. Disponível em: [http://www.facape.br/jocelio/arquivos/DissertacaoLMS\\_final.pdf](http://www.facape.br/jocelio/arquivos/DissertacaoLMS_final.pdf). Acesso em novembro de 2006.

[PELISSOLI, LOYOLLA, 2004] PELISSOLI, Luciano; LOYOLLA, Waldomiro. Aprendizado Móvel: dispositivos e cenários <http://www.abed.org.br/congresso2004/por/htm/074-TC-C2.htm> . Acesso em setembro de 2006.

[PDA, 2006] *Personal Digital Assistant*. Disponível em: [http://en.wikipedia.org/wiki/Personal\\_digital\\_assisten](http://en.wikipedia.org/wiki/Personal_digital_assisten) . Acesso em novembvro de 2006

[QUALCOMM, 2006] Qualcomm. Disponível em: <http://brew.qualcomm.com/brew/en/>. Acesso em novembro de 2006.

[SCHAEFER, 2004] SCHAEFER, Carine - Protótipo de aplicativo para transmissão de dados a partir de dispositivos móveis aplicado a uma empresa de transportes. <http://www.inf.furb.br/~pericas/orientacoes/J2METransporte2004.pdf>  
Acesso em agosto de 2006.

[STATELESS, 2006] Protocolo http. Disponível em: <http://www.webopedia.com/TERM/H/HTTP.html> . Acesso em dezembro de 2006.

[SUN BRASIL, 2006] Tecnologia Java. Disponível em: <http://java.sun.com>. Acesso em novembro de 2006.

[SRI, 2006] Stanford Research Institute. Disponível em: <http://siher.stanford.edu>. Acesso em novembro de 2006.

[SDC, 2006] *Secure Digital Card*. Disponível em: [http://pt.wikipedia.org/wiki/secure\\_digital\\_card](http://pt.wikipedia.org/wiki/secure_digital_card). Acesso em 18/11/06.

[SMS, 2006] O que é SMS. Disponível em: <http://w3.interlize.com.br/modules/xfsection/article.php?.Acesso> em outubro de 2006

[SQLM, 2006] SQL Manager 2005 for Interbase/Firebird. Disponível em: <http://www.sqlmanager.net/products/ibfb/manager> . Acesso em dezembro de 2006

[SYMBIAN OS, 2006]. *SymbianOS*. Disponível em: <http://en.wikipedia.org/wiki/symbian>. Acesso em novembro de 2006.

[TAROUCO, FABRE, KONRATH, GRANDO, 2004]. TAROUCO, Liane M. R.; FABRE, Marie-Christine Julie M.; KONRATH, Mary Lúcia P.; GRANDO, Anita Raquel – Objetos de Aprendizagem M-Learning, 2004. [http://www.cinted.ufrgs.br/CESTA/objetosdeaprendizagem\\_sucesu.pdf](http://www.cinted.ufrgs.br/CESTA/objetosdeaprendizagem_sucesu.pdf) . Acesso em setembro de 2006.

[TDMA, 2006] Time Division Multiple Access. Disponível em: <http://pt.wikipedia.org/wiki/TDMA> Acesso em novembro de 2006.

[TERMOS#,2006] Termos#I\_ Termos técnicos GdH . Disponível em: <http://www.guiadohardware.net/termos/i.html>. Acesso em novembro de 2006.

[TOMCAT, 2006] Apache Tomcat. Disponível em: [http://pt.wikipedia.org/wiki/Apache\\_Tomcat](http://pt.wikipedia.org/wiki/Apache_Tomcat). Acesso em dezembro de 2006.

[USB, 2006] *Universal Serial Bus*. Disponível em: <http://www.infowester.com/usb.php>. Acesso em novembro de 2006.

[UML 2005] Object Management Group, Inc (OMG) Unified Modeling Language™ (UML™). Disponível em <http://www.omg.org/uml/>. Acesso em novembro de 2006.

[UML, 2006] UML (Unified Modeling Language). Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/uml.htm> . Acesso em dezembro de 2006.

[URL, 2006] Universal Resource Locator. Disponível em: [www.pt.wikipedia.org/wiki/URL](http://www.pt.wikipedia.org/wiki/URL). Acesso em dezembro de 2006.

[WAP, 2006] *Wireless Application Protocol*. Disponível em: <http://pt.wikipédia.org/wiki/wap>. Acesso em novembro de 2006.

[WAP Forum, 2004] "WAP Forum". Disponível em <http://www.wapforum.org>. Acesso em outubro de 2006.

[WIRELESS, 2006] O que é Wireless? Disponível em: [http:// sistema.com.br/materiais/idmat002959.htm](http://sistema.com.br/materiais/idmat002959.htm). Acesso em novembro de 2006.

[WM, 2006] Windows Mobile. Disponível em: <http://www.microsoft.com>. Acesso em novembro de 2006.

[XML, 2006] *Extensible Markup Language*. Disponível em: [http:// www.w3.org/xml](http://www.w3.org/xml). Acesso em novembro de 2006.

## ANEXOS

### Anexo I – CÓDIGO FONTE

// Aplicação: Cliente (celular)

#### Classe ClienteMidlet

```
import java.io.IOException;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.ChoiceGroup;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Item;
import javax.microedition.lcdui.ItemCommandListener;
import javax.microedition.midlet.MIDlet;
import
javax.microedition.midlet.MIDletStateChangeException;

//extends de MIDlet para que o Application Manager possa
//interpretar esse
//aplicativo como sendo um aplicativo MIDP e impelementa
//CommandListener
//para que a classe tenha que tratar ações de botões.

public class ClienteMidlet extends MIDlet implements
CommandListener, ItemCommandListener{

    private Display display;
    private formLogin formLogin;
    private listaProvas listaProvas;
    private listaQuestoes listaQuestoes;
    private formPergunta;
    private String login;
    private String senha;
    private formAguarde formaguarde;
    private threadProvas threadProvas;
    private threadQuestoes threadQuestoes;
    private threadResposta threadResposta;
    private threadResultado threadResultado;

    public ClienteMidlet() {
        super();
        // Instanciando os formulários e as listas que serão
        usadas na aplicação
        display = Display.getDisplay(this);
        formLogin = new formLogin(this);
```

```

        formaguarde = new formAguarde(this);
        formPergunta=new formPergunta(this);
        listaProvas = new listaProvas(this,null);
        listaQuestoes = new listaQuestoes(this,null);
        display.setCurrent(formLogin);// mostra na tela
    }
    //método chamado quando a aplicação for encerrada,
    //não há necessidade de ser implementada

    protected void destroyApp(boolean arg0) throws
MIDletStateChangeException {

    }
    //método executado se a MIDlet foir pausada por ela mesma
    //ou por um evento externo

    protected void pauseApp() {
        // TODO Auto-generated method stub
    }

    //método executado ao inicio de Execução da MIDlet
    protected void startApp() throws
MIDletStateChangeException {

    }
    //Esse método fica responsável pelo controle das ações das
    //listas e dos forms da aplicação
    public void commandAction(Command c, Displayable d) {
        if(c == formLogin.getCmdOk()){//habilita o botão OK
para quando o usuário digitar a senha e o login
        // fazer a verificação de confirmação ou não

            login = formLogin.getlogin().getString();
            senha = formLogin.getsenha().getString();
            System.out.println("login: " + login);
            System.out.println("Senha: " + senha);
            if(senha.equals("") || login.equals("")){

                }// Chama lista de provas
                threadProvas = new threadProvas(display,
formLogin, listaProvas,login, senha);
                Thread thread = new Thread(threadProvas);
                thread.start();
                System.out.println("Entrou no
threadProvas...");
                display.setCurrent(formaguarde);
            }
        }
    }

```

```

        if(c == formLogin.getCmdExit()){//Sair da Aplicação
            try{
                this.destroyApp(false);
                this.notifyDestroyed();
            }catch (Exception e) {
                e.printStackTrace();
            }
        }

        if(c == listaProvas.getCmdExibir()){// Exibe a
        lista de questões
            //Monta a Thread
            threadQuestoes = new threadQuestoes(display,
        listaQuestoes,login, senha,
        listaProvas.getString(listaProvas.getSelectedIndex()));
            Thread thread2 = new Thread(threadQuestoes);
            //Inicia a thread
            thread2.start();
            display.setCurrent(formaguarde);

            System.out.println(listaProvas.getString(listaProvas.g
        etSelectedIndex()));
        }

        if(c == listaQuestoes.getCmdResolver()){// Mostra
        as perguntas e respostas da questão
            threadResposta = null;
            threadResposta = new threadResposta(display,
        formPergunta,login, senha,
        listaQuestoes.getString(listaQuestoes.getSelectedIndex()));
            Thread thread3 = new Thread(threadResposta);
            thread3.start();
            display.setCurrent(formaguarde);
        }

        if(c == formaguarde.getCmdCancelar()){// Erro de
        conexão, testando conexão
            try {
                threadProvas.pararConexao();
                threadQuestoes.pararConexao();
            } catch (IOException e) {
                e.printStackTrace();
            }catch (NullPointerException e) {
                e.printStackTrace();
            }catch (SecurityException e) {
                e.printStackTrace();
            }finally{

```

```

        display.setCurrent(formLogin);
    }
}

    if(c == listaProvas.getCmdVoltar()){// Aciona
comando voltar do form Lista de Provas
        display.setCurrent(formLogin);
    }
    if(c == listaQuestoes.getCmdVoltar()){
        display.setCurrent(listaProvas);
    }

}

//Esse método vai ficar responsável pelo controle das
ações dos itens do formulário de resposta
public void commandAction(Command c, Item d) {

    if(c==formPergunta.getCmdOK()){// habilita o
botão Ok das respostas das perguntas e chama o resultado
        threadResultado = new
threadResultado(display, formPergunta, login, senha,
formPergunta.getChoice().getString(formPergunta.getChoice()
.getSelectedIndex()), listaQuestoes);
        Thread thread4 = new
Thread(threadResultado);
        thread4.start();
        display.setCurrent(formaguarde);
    }

    if(c==formPergunta.getCmdVoltar()){// Habilita o
botão voltar do form das respostas da pergunta da Questão
        display.setCurrent(listaQuestoes);

    }

}

}

```



**Classe Conectar**

```
public class Conectar {  
    // Guarda o endereço do servidor e a porta  
    private static String urlServidor = "127.0.0.1";  
    private static String portaServidor = "8080";  
  
    static public String getPortaServidor() {  
        return portaServidor;  
    }  
  
    public static String getUrlServidor() {  
        return urlServidor;  
    }  
  
}
```

**Classe FormLogin**

```

import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.TextField;

public class formLogin extends Form{
    private TextField login, senha;
    private Command cmdOk,cmdExit;

    public formLogin(CommandListener listener) {
        super("Simulado");

        login = new TextField("login
", "", 12, TextField.ANY);
        senha = new TextField("Senha
", "", 10, TextField.PASSWORD+TextField.NUMERIC);
        // Comandos do Formulário
        cmdOk = new Command ("OK",Command.OK,1);
        cmdExit = new Command ("Sair",Command.EXIT,0);
        //adiciona os comandos
        this.addCommand(cmdOk);
        this.addCommand(cmdExit);

        //adiciona-se os componentes ao Form Login

        this.append(login);
        this.append(senha);
        this.setCommandListener(listener);
    }

    public Command getCmdExit() {
        return cmdExit;
    }

    public Command getCmdOk() {
        return cmdOk;
    }

    public TextField getlogin() {
        return login;
    }

    public TextField getsenha() {
        return senha;
    }
}

```

**Classe listaProvas**

```

import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.List;

public class listaProvas extends List{

    private Command cmdExibir,cmdVoltar;

    public listaProvas(CommandListener listener, String
mensagem) {
        super("Provas",List.IMPLICIT); //lista implícita
de provas

// Comandos da tela da lista de Provas
        cmdExibir = new Command("Exibir",Command.OK,0);
        cmdVoltar = new Command("Voltar",Command.BACK,0);
// adiciona os comandos
        this.addCommand(cmdExibir);
        this.addCommand(cmdVoltar);

        this.setCommandListener(listener);// Os objetos
são setados por este comando

    }
    public Command getCmdExibir() {
        return cmdExibir;
    }

    public Command getCmdVoltar() {
        return cmdVoltar;
    }
}

```

**Classe listaQuestões**

```
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.List;

public class listaQuestoes extends List{

    private Command cmdResolver,cmdVoltar;

    public listaQuestoes(CommandListener listener, String
mensagem) {
        super("Questões",List.IMPLICIT);//lista implícita
de questões

        cmdResolver = new
Command("Resolver",Command.OK,0);
        cmdVoltar = new Command("Voltar",Command.BACK,0)
// adição dos comandos aos objetos e setados pelo comando
//Listener
        this.addCommand(cmdResolver);
        this.addCommand(cmdVoltar);
        this.setCommandListener(listener);

    }
    public Command getCmdResolver() {
        return cmdResolver;
    }

    public Command getCmdVoltar() {
        return cmdVoltar;
    }
}
```

**Classe formPergunta**

```

import javax.microedition.lcdui.ChoiceGroup;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.ItemCommandListener;
import javax.microedition.lcdui.StringItem;

public class formPergunta extends Form{

    private Command cmdVoltar,cmdOK;
    public ChoiceGroup listaRespostas = new
ChoiceGroup("",ChoiceGroup.EXCLUSIVE);
    private StringItem pergunta = new
StringItem(null,null);
    private ItemCommandListener listener2;
    // Formulário das perguntas
    public formPergunta(ItemCommandListener listener) {
        super("Pergunta");
        listener2 = listener;
        // Comandos do formulário
        cmdOK = new Command("OK",Command.OK,0);
        cmdVoltar = new Command("Voltar",Command.BACK,1);
        // adiciona a pergunta
        this.append(pergunta);
        // adiciona os comandos na lista de respostas
        this.listaRespostas.addCommand(cmdOK);

this.listaRespostas.setItemCommandListener(listener);
    }
    //
    public void setChoice(ChoiceGroup lista){
        this.listaRespostas = lista;
        this.listaRespostas.addCommand(cmdOK);
        this.listaRespostas.addCommand(cmdVoltar);
        this.append(lista);

        this.listaRespostas.setItemCommandListener(listener2);
    }

    public ChoiceGroup getChoice(){
        return this.listaRespostas;
    }

    public StringItem getPergunta() {
        return pergunta;
    }

    public void setPergunta(StringItem pergunta) {
        this.pergunta = pergunta;
    }
}

```

```
        public Command getCmdOK() {  
            return cmdOK;  
        }  
  
        public Command getCmdVoltar() {  
            return cmdVoltar;  
        }  
    }  
}
```

**Classe threadProvas**

```

import java.io.IOException;
import java.io.InputStream;
import javax.microedition.io.ConnectionNotFoundException;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.List;

public class threadProvas implements Runnable {

    private Display display;
    private Form formLogin;
    private List lista;
    private String URL;
    private HttpConnection http;
    private Alert alerta = new Alert("Não Conectado...",
"Não foi possivel fazer a conexao...",null,null);
    public threadProvas(Display display,Form
formLogin,List lista, String login, String senha){
        this.display = display;
        this.formLogin = formLogin;
        this.lista = lista;
        // monta a URL
        this.URL =
urlEncode("http://" + Conectar.getUrlServidor() + ":" + Conectar.
getPortaServidor() + "/servidorjsp/index.jsp?&login="+login+"
&senha="+senha);

        public void run() {
try{

            http = (HttpConnection)
Connector.open(URL,Connector.READ_WRITE); // Faz a conexão
            http.setRequestMethod(HttpConnection.GET);
            if (http.getResponseCode() ==
HttpConnection.HTTP_OK){
                InputStream iStrm =
http.openInputStream();
                int length = (int) http.getLength();
                if (length > 0) {
                    byte servletData[] = new
byte[length];

                    iStrm.read(servletData);
                    String dado_retorno = new
String(servletData);

```

```

String listadesaida[] = new
String[10];

int contLista = 0;
listadesaida[contLista] = "";

for(int cont = 0; cont <
dado_retorno.length(); cont++){

    String x =
String.valueOf(dado_retorno.charAt(cont));

    if(x.equals("-")){
        contLista++;
        listadesaida[contLista]
= "";
    }else{
        if(x!= null){
            listadesaida[contLista] = listadesaida[contLista] + x;}
        }

    }

    //Zerando o tipo List
    int w = lista.size();
    for(int x=0; x < w;x++){
        lista.delete(0);}

    //Preenchendo a lista de Provas
    for(int cont2 = 0; cont2 <
listadesaida.length; cont2++){

        if(listadesaida[cont2]!=null){
            String x =
listadesaida[cont2];
            lista.append(x.trim(),
null);
        }else{
        }
    }

    //controle de erro... login e senha
    if((lista.size()-1)==0){
        Alert alerta2 = new Alert("Erro de
login", "Não foi possivel fazer a conexão...\n Por favor
verifique a senha e o login...",null,null);

        display.setCurrent(alerta2,formLogin);
        alerta2.setTimeout(3000);
    }else{
        display.setCurrent(lista);
    }
}

```



```

    }
    http.close();// fecha a conexão
    http = null;
} catch (ConnectionNotFoundException e) { //trata
as exceções
    display.setCurrent(alerta,formLogin);
    alerta.setTimeout(3000);
    e.printStackTrace();
} catch (SecurityException e) {
    display.setCurrent(alerta,formLogin);
    alerta.setTimeout(3000);
    e.printStackTrace();
} catch (Exception e) {
    display.setCurrent(alerta,formLogin);
    alerta.setTimeout(3000);
    e.printStackTrace();
} finally{

    }

}

// Tira os espaços em branco da URL e coloca o sinal +
e tira os espaços vazios da direita e esquerda
public static String urlEncode(String s) {
    s = s.trim();
    s= s.replace(' ', '+');
    return s;
}
//Fecha conexão
public void pararConexao() throws IOException{
    http.close();
}

}

```

**Classe threadQuestões**

```

import java.io.IOException;
import java.io.InputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.List;

public class threadQuestoes implements Runnable{

    private Display display;
    private List listaQuestoes;
    private String URL;
    private HttpConnection http;

    public threadQuestoes(Display display, List
listaQuestoes, String login, String senha, String prova){
        this.display = display;
        this.listaQuestoes = listaQuestoes;
        this.URL =
urlEncode("http://" + Conectar.getUrlServidor() + ":" + Conectar.
getPortaServidor() + "/servidorjsp/index2.jsp?&login=" + login +
"&senha=" + senha + "&prova=" +
        prova); //monta a URL
        System.out.println("URL Questoes: " + URL);
    }

    public void run() {
        try{
            http = (HttpConnection)
Connector.open(URL, Connector.READ_WRITE);
            http.setRequestMethod(HttpConnection.GET);
            if (http.getResponseCode() ==
HttpConnection.HTTP_OK){
                InputStream iStrm =
http.openInputStream();
                int length = (int) http.getLength();
                if (length > 0) {
                    byte servletData[] = new
byte[length];

                    iStrm.read(servletData);
                    String dado_retorno = new
String(servletData);

                    String listadesaida[] = new
String[10];

                    int contLista = 0;
                    listadesaida[contLista] = "";

```

```

        for(int cont = 0; cont <
dado_retorno.length(); cont++){

            String x =
String.valueOf(dado_retorno.charAt(cont));

            if(x.equals("-")){//
Identifica "-" e organiza em Lista
                contLista++;
                listadesaida[contLista]
= "";
            }else{
                if(x!= null){
                    listadesaida[contLista]
= listadesaida[contLista] + x;}
            }

            int w = listaQuestoes.size();
            for(int x=0; x < w;x++){
                listaQuestoes.delete(0);}

        for(int cont2 = 0; cont2 <
listadesaida.length; cont2++){

            if(listadesaida[cont2]!=null){
                String x =
listadesaida[cont2];

                String y = x.trim();

                listaQuestoes.append(y+"\n",null);// monta a lista de
questões

            }
        }
        // mostra na tela lista de
Questões

        display.setCurrent(listaQuestoes);
        //Retorno da informação.
    }

    }
    http.close();
} catch (Exception e) {
    e.printStackTrace();
}

}

```

```
public static String urlEncode(String s) {  
    s = s.trim();  
    s= s.replace(' ', '+');  
    return s;  
}  
  
public void pararConexao() throws  
IOException{  
    http.close();  
}  
  
}
```

**Classe threadResposta**

```

import java.io.IOException;
import java.io.InputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.ChoiceGroup;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.List;
import javax.microedition.lcdui.StringItem;

public class threadResposta implements Runnable{

    private Display display;
    private String URL;
    private HttpConnection http;
    private formPergunta formPergunta;
    private ChoiceGroup listaRespostas = new
ChoiceGroup("",ChoiceGroup.EXCLUSIVE);
    private StringItem pergunta; // = new
StringItem(null,null);

    public threadResposta(Display display, Form
formPergunta, String login, String senha, String questao){
        this.display = display;
        this.formPergunta = (formPergunta)formPergunta;
        this.formPergunta.deleteAll();
        this.URL =
urlEncode("http://" + Conectar.getUrlServidor() + ":" + Conectar.
getPortaServidor() + "/servidorjsp/index4.jsp?&questao=" + ques
tao);

        System.out.println("URL respostas: " + URL);

    }

    public void run() {
        try{
            http = (HttpConnection)
Connector.open(URL,Connector.READ_WRITE);
            http.setRequestMethod(HttpConnection.GET);
            if (http.getResponseCode() ==
HttpConnection.HTTP_OK){
                InputStream iStrm =
http.openInputStream();
                int length = (int) http.getLength();
                if (length > 0) {
                    byte servletData[] = new
byte[length];

                    iStrm.read(servletData);

```

```

        String dado_retorno = new
String(servletData);

        String listadesaida[] = new String[10];
        int contLista = 0;
        listadesaida[contLista] = "";

        for(int cont = 0; cont <
dado_retorno.length(); cont++){

            String x =
String.valueOf(dado_retorno.charAt(cont));

            if(x.equals("-")){
                contLista++;
                listadesaida[contLista] = "";
            }else{
                if(x!= null){
                    listadesaida[contLista] = listadesaida[contLista] +
x;
                }
            }
        }

        for(int cont2 = 0; cont2 < listadesaida.length; cont2++){

            if(listadesaida[cont2]!=null){
                String x = listadesaida[cont2];
                String y = x.trim();
                if ( cont2 == 0){
                    pergunta = new StringItem(y,null);
                }else{
                    listaRespostas.append(y+"\n",null);
                }
            }
        }

        this.formPergunta.append(pergunta);

        listaRespostas.delete(listaRespostas.size()-1);
        try{
            this.formPergunta.setChoice(listaRespostas);
        }catch (Exception e) {
            e.printStackTrace();
        }

        display.setCurrent(this.formPergunta);
    }
    http.close();
} catch (Exception e) {

```

```
        e.printStackTrace();
    }

    }

    public static String urlEncode(String s) {
        s = s.trim();
        s= s.replace(' ', '+');
        return s;
    }

    public void pararConexao() throws IOException{
        http.close();
    }

    }
```

**Classe threadResultado**

```

import java.io.IOException;
import java.io.InputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.ChoiceGroup;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.List;
import javax.microedition.lcdui.StringItem;

public class threadResultado implements Runnable{

    private Display display;
    private String URL;
    private HttpConnection http;
    private Form formPergunta;
    private ChoiceGroup listaRespostas = new
ChoiceGroup("",ChoiceGroup.EXCLUSIVE);
    private StringItem pergunta;
    private Alert alertaerrado = new Alert("Esta
Errado","Errado",null,AlertType.ERROR);
    private Alert alertacerto = new Alert("Esta
Certo","Certo",null,AlertType.CONFIRMATION);
    private List listaQuestoes;
    String certo;

    public threadResultado(Display display, Form
formPergunta, String login, String senha, String resposta,
List listaQuestoes){
        this.display = display;
        this.formPergunta = formPergunta;
        this.listaQuestoes = listaQuestoes;
        // monta a URL
        this.URL =
urlEncode("http://" + Conectar.getUrlServidor() + ":" + Conectar.
getPortaServidor() + "/servidorjsp/index5.jsp?&resposta="+res
posta);
    }

    public void run() {
        try{
            http = (HttpConnection)
Connector.open(URL,Connector.READ_WRITE); // Faz a conexão
            http.setRequestMethod(HttpConnection.GET);
            if (http.getResponseCode() ==
HttpConnection.HTTP_OK){

```



```

        InputStream iStrm = http.openInputStream();
        int length = (int) http.getLength();
        if (length > 0) {
            byte servletData[] = new byte[length];
            iStrm.read(servletData);
            String dado_retorno = new String(servletData);

            String listadesaida[] = new String[10];
            int contLista = 0;
            listadesaida[contLista] = "";

            for (int cont = 0; cont < dado_retorno.length(); cont++){
                String x =
                String.valueOf(dado_retorno.charAt(cont));

                if(x.equals("0")){
                    certo = "0";
                }else if (x.equals("1")) {
                    certo = "1";
                }

            }

        }

        http.close();
        // Verifica se a resposta está certa ou errada
        if(certo.equals("0")){
            display.setCurrent(alertaerrado, formPergunta);
        }else{
            display.setCurrent(alertacerto, listaQuestoes);
        }

        }catch (Exception e) {
            e.printStackTrace();
        }

    }

    public static String urlEncode(String s) {
        s = s.trim();
        s= s.replace(' ', '+');
        return s;
    }

    public void pararConexao() throws IOException{
        http.close();
    }

}

```

**//Aplicação: Servidor (PC)****JSP**

//JSP com os parâmetros que vão ser enviados para classe busca

**Index.jsp**

```
<%@ page import="orbital.Busca"%>

<%
String login = request.getParameter("login");
String senha = request.getParameter("senha");
%>

<%
    Busca busca = new Busca();
    String lista = busca.RetornaLista(login, senha);
%>
<%=lista%>
```

**Index2.jsp**

```
<%@ page import="orbital.Busca"%>

<%
String prova = request.getParameter("prova");
%>

<%
    Busca busca = new Busca();
    String lista = busca.RetornaListaQuestao(prova);
%>
<%=lista%>
```

**Index3.jsp**

```
<%@ page import="orbital.Busca"%>

<%
String questao = request.getParameter("questao");
%>

<%
    Busca busca = new Busca();
    String lista = busca.RetornaPergunta(questao);
%>
<%=lista%>
```

**Index4.jsp**

```
<%@ page import="orbital.Busca"%>

<%
String questao = request.getParameter("questao");
```

```
%>

<%
    Busca busca2 = new Busca();
    String lista2 = busca2.RetornaPergunta(questao);
%>
<%=lista2%>-
```

```
<%
    Busca busca = new Busca();
    String lista = busca.RetornaRespostas(questao);
%>
<%=lista%>
```

### **Index5.jsp**

```
<%@ page import="orbital.Busca"%>

<%
String resposta = request.getParameter("resposta");
%>

<%
    Busca busca = new Busca();
    String lista = busca.RetornaCertaErrada(resposta);
%>
<%=lista%>
```

**Classe Busca**

```

package orbital;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

// Esta classe recebe os parâmetros do JSP
public class Busca {
    //O metodo recebe dois parametros, login e senha
    public String RetornaLista(String login, String senha)
    throws SQLException{

        try{
            // monta o SQL
            String sql = "select * from USUARIO where
usuario.LOGIN='"+login+"' and usuario.SENHA='"+senha+"'";
            PreparedStatement st =
Conectar.RetornarPreparedStatementFirebird(sql);
            // retorna um o resultado da busca
            ResultSet rs = st.executeQuery();

            String retorno = "";

            //Verifica se houve retorno (se o usuario
existe)
            if(rs.next()){
                PreparedStatement st2 =
Conectar.RetornarPreparedStatementFirebird(
                    "select Provas.TITULO from
PROVAS");
                ResultSet rs2 = st2.executeQuery();
                //Monta uma String com a lista das
provas separadas por "-"
                while (rs2.next()){
                    retorno = retorno +
rs2.getString(1) + "-";
                }
            }else{
                retorno = null;
            }
            return retorno;

        }catch(SQLException e){System.out.print("Erro
Class Busca: "+e);}
        return null;
    }

    //método que recebe como parâmetro "prova"
    public String RetornaListaQuestao(String prova) throws
SQLException{

```

```

        try{
            // monta o SQL
            String sql = "select QUESTOES.TEXTO from
PROVAS,QUESTOES where provas.TITULO ='"+prova+ "'and
QUESTOES.PROVAS_CODIGO = PROVAS.CODIGO";
            PreparedStatement st =
Conectar.RetornarPreparedStatementFirebird(sql);

            ResultSet rs = st.executeQuery();

            String retorno = "";
            // verifica o retorno e coloca o separador
            “-” na lista de questões
            while (rs.next()){
                retorno = retorno +
rs.getString(1) + "-";
            }

            return retorno;

        }catch(SQLException e){System.out.print("Erro
Class Busca: "+e);}
        return null;
    }

    //método que recebe como parâmetro “questão”
    public String RetornaPergunta(String questao) throws
SQLException{

        try{
            // monta o SQL
            String sql = "select QUESTOES.PERGUNTA from
QUESTOES where QUESTOES.TEXTO = '"+questao+"'";
            PreparedStatement st =
Conectar.RetornarPreparedStatementFirebird(sql);

            ResultSet rs = st.executeQuery();

            String retorno = "";
            // Verifica o retorno da questão
            while (rs.next()){
                retorno = retorno +
rs.getString(1);
            }

            return retorno;

        }catch(SQLException e){System.out.print("Erro
Class Busca: "+e);}
        return null;
    }

    // Método que recebe como parâmetro “questão”

```

```

        public String RetornaRespostas(String questao) throws
        SQLException{

            try{
                //monta o SQL
                String sql = "select RESPOSTAS.TEXTO from
                QUESTOES,RESPOSTAS where QUESTOES.TEXTO = '"+questao+"' and
                RESPOSTAS.QUESTOES_CODIGO = QUESTOES.CODIGO";
                PreparedStatement st =
                Conectar.RetornarPreparedStatementFirebird(sql);
                ResultSet rs = st.executeQuery();

                String retorno = "";
                // Verifica o retorno da resposta da questão
                while (rs.next()){
                    retorno = retorno + rs.getString(1) + "-";
                }

                return retorno;

            }catch(SQLException e){System.out.print("Erro
            Class Busca: "+e);}
            return null;
        }
        // método que recebe como parâmetro " resposta"
        public String RetornaCertaErrada(String resposta) throws
        SQLException{
            try{
                monta o SQL
                String sql = "select RESPOSTAS.CORRETA from RESPOSTAS where
                RESPOSTAS.TEXTO = '"+resposta+"'";
                PreparedStatement st =
                Conectar.RetornarPreparedStatementFirebird(sql);
                ResultSet rs = st.executeQuery();
                String retorno = "";
                //Verifica o retorno da resposta da questão (certa/errada)
                while (rs.next()){
                    retorno = retorno + rs.getString(1);
                }
                return retorno;
            }catch(SQLException e){System.out.print("Erro
            Class Busca: "+e);}
            return null;
        }
    }
}

```

**Classe Conectar**

```

package orbital;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class Conectar {

    static public PreparedStatement
    RetornarPreparedStatementFirebird(String sql){
        try{

            Class.forName("org.firebirdsql.jdbc.FBDriver");// Seta
            o drive de conexão do Banco de DADOS
            //Seta o caminho onde se encontra o banco de dados
            Connection con =
            DriverManager.getConnection("jdbc:firebirdsql:localhost:C:/
            Documents and
            Settings/Micro/Desktop/Projeto_celular/DB/SIMULADO.FDB","SY
            SDBA","masterkey");
                //gera o PreparedStatement
                PreparedStatement st = con.prepareStatement(sql);
                //retorna o PreparedStatement
                return st;
            }catch(final Exception e){
                return null;
            }

        }

    }
}

```