

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística
Curso de Ciências Da Computação

Sistema Web gerador de formulários de pesquisa para
dispositivos móveis

Aguilar Figueira Dias

Florianópolis, 2007

Aguilar Figueira Dias

Sistema Web gerador de formulários de pesquisa para
dispositivos móveis

Monografia apresentada à Universidade
Federal de Santa Catarina, como parte dos
requisitos para a obtenção do grau de Bacharel
em Ciências da Computação sob Orientação do
Prof. Dr. João Bosco da Mota Alves

Florianópolis, 2007

Aguilar Figueira Dias

Sistema Web gerador de formulários de pesquisa para
dispositivos móveis

Monografia aprovada em, xx/xx/2007,
como requisito para obtenção do Grau de
Bacharel em Ciências da Computação.

Banca Examinadora

Prof. Dr. João Bosco da Mota Alves
Orientador

Prof. Dr. Rosvelter Coelho da Costa
Membro

Prof. Dr. Vitório Bruno Mazzola
Membro

Florianópolis, 2007

Sumário

Lista de Figuras	Pág 7
Lista de Tabelas	Pág 9
Lista de Abreviaturas e Siglas	Pág 10
Resumo	Pág 11
Abstract	Pág 12
1 Introdução	Pág 13
1.1 Objetivo Geral	Pág 14
1.2 Objetivo Específico	Pág 14
1.3 Modelo gráfico dos sistemas	Pág 15
2 TECNOLOGIAS UTILIZADAS	Pág 18
2.1 Especificação do sistema móvel(J2ME)	Pág 18
2.1.1 Introdução	Pág 18
2.1.2 Configuration	Pág 18
2.1.3 Profiles	Pág 18
2.1.4 Segurança	Pág 18
2.1.5 Tecnologias Wireless	Pág 22
2.1.6 Diferença entre MIDP 1.0 e 2.0	Pág 23
2.1.7 API's	Pág 24
2.1.8 Outras restrições	Pág 24
2.1.9 Ciclo de Vida de uma MIDlet	Pág 24
2.1.10 Interface	Pág 26

2.1.11 Armazenamento em celulares (RecordStores) – RMS	Pág 26
2.1.12 KXML(leitor de arquivos XML)	Pág 27
2.2 Especificação do sistema Web	Pág 29
2.2.1 Linguagem de programação	Pág 29
2.2.2 PHP	Pág 29
2.2.3 O Design Pattern MVC	Pág 33
2.2.4 M – Lumine	Pág 34
2.2.5 V – Smarty	Pág 36
2.2.6 C – PHPMVC	Pág 37
3 Projeto Sistema Web	Pág 38
3.1 Informações Preliminares	Pág 38
3.2 Domínio de negócios	Pág 38
3.3 Requisitos Funcionais	Pág 38
3.4 Requisitos não-funcionais	Pág 39
4 Implementação do sistema Web	Pág 40
4.1 Ferramentas utilizadas	Pág 40
4.2 Descrição do sistema	Pág 40
4.3 Diagrama do Banco de Dados	Pág 47
5 Projeto Sistema Móvel(J2ME)	Pág 48
5.1 Informações Preliminares	Pág 48
5.2 Domínio de negócios	Pág 48
5.3 Requisitos Funcionais	Pág 48
5.4 Requisitos não-funcionais	Pág 49

6 Implementação do sistema J2ME	Pág 50
6.1 Ferramentas utilizadas	Pág 50
6.2 Descrição do sistema	Pág 50
7 Conclusão	Pág 61
8 Trabalhos Futuros	Pág 62
Referências Bibliográficas	Pág 63
Referências das Figuras	Pág 64
Referências das Tabelas	Pág 65
Anexo A - Código Fonte Sistema Web	Pág 66
Anexo B - Código Fonte Sistema J2ME	Pág 94

Lista de Figuras

Figura 1: Plataforma Java	Pág 19
Figura 2: Dispositivos para as plataformas Java	Pág 20
Figura 3 : Ciclo de vida de uma Midlet	Pág 24
Figura 4: API para construção de interface gráfica	Pág 25
Figura 5: Estrutura de uma Record Store	Pág 26
Figura 6: Gráfico do uso das linguagens de programação	Pág 30
Figura 7: Modelo MVC	Pág 33
Figura 8: Como uma ferramenta Objeto-Relacional trabalha	Pág 34
Figura 9: Tela principal sistema Web	Pág 40
Figura 10: Formulário inserção pesquisa	Pág 40
Figura 11: lista de perguntas para uma pesquisa	Pág 41
Figura 12: Formulário inserção de uma pergunta em um pesquisa	Pág 42
Figura 13: Formulário inserção de uma pergunta com opções avançadas em um pesquisa	Pág 43
Figura 14: Formulário para alteração de uma pergunta	Pág 44
Figura 15: Diagrama do banco de dados	Pág 46
Figura 16: Tela de entrada do sistema móvel(J2ME)	Pág 50
Figura 17: Lista de pesquisas disponíveis	Pág 51
Figura 18: Tela principal de uma pesquisa	Pág 53
Figura 19: Opções disponíveis na tela principal	Pág 54
Figura 20: Realização de uma pesquisa	Pág 54

Figura 21: Tela de aviso, descarte de pesquisa	Pág 55
Figura 22: Tela de aviso, responda todas as perguntas	Pág 56
Figura 23: Tela principal, exibição dados atualizados	Pág 57
Figura 24: Lista de pesquisas, ênfase na opção “Relatório”	Pág 58
Figura 25: Estatísticas geradas para cada resposta de uma pergunta ...	Pág 59

Lista de Tabelas

Tabela 1: APIs para leitura de arquivos XML Pág 27

Tabela 2: Utilização das linguagens de programação em 2006/2007 Pág 29

Lista de Abreviaturas

API: Application Programming Interface

PDA - Personal digital assistants

CLDC - Connected Limited Device Configuration

CDC - Connected Limited Configuration

TDMA - time division multiple access

CDMA - code division multiple access

GSM - global system for communication

PHP - Hypertext Preprocessor

XML - Extensible Markup Language

SQL - Structured Query Language

J2EE - Java 2 Platform Enterprise Edition

HTTP - Hiper Text Tranfer Protocol

EJB - Enterprise Java Beans

Resumo

O objetivo do trabalho é automatizar uma tarefa repetitiva, que precise de muita intervenção humana e conseqüentemente adiciona muitas possibilidades de erro.

Esta tarefa é a realização de pesquisas, como por exemplo o CENSO do Brasil, pesquisas eleitorais, e quaisquer tipo de pesquisa que tenha o padrão de perguntas seguidas por respostas.

Isso será feito com o desenvolvimento de dois sistemas, um web, que será o responsável por gerar e gerenciar as pesquisas ficando hospedado em qualquer servidor na Internet, e um sistema para ser realizada as pesquisas, este sistema será feito para dispositivos móveis mais especificamente para celulares. Com esses dois sistemas toda a tarefa de criação de formulários de pesquisa em papel, preenchimento dos formulários com as respostas e geração de estatísticas serão totalmente automatizadas, reduzindo assim o tempo de trabalho para obtenção dos resultados de uma pesquisa e praticamente anulando possíveis erros na manipulação dos dados coletados.

Palavras-chave: J2ME, sistemas web, pesquisas

Abstract

The goal of the work is to automatize an repetitive task, that need to much human intervention and consequently add many errors possibilities.

This task is an research work, like CENSO in Brazil, electoral research, and any other kinds of research that follow a standard with questions follow by answers.

This work will be the development os two systems, one web, which will be responsible to generate and manage the researchs, that system will be hosted in any internet server, and another system to really do the work of research, this system will be made for mobile devices, more specifically cell phones.

With this two systems all the work to create the forms os research in paper, fulfilling of the forms with the answers and statistics generation will be totally automatized, reducing the work time to get the results of one research and practically annulling posible errors in manipulate the collected datas.

Key words: J2ME, web systems, researchs

1 Introdução

A automatização de tarefas teve um grande aumento com o advento dos computadores, trabalhos repetitivos nos quais pode-se definir um padrão de etapas a serem seguidas, são sempre alvo da automação, pode-se automatizar desde trabalhos administrativos até trabalhos mecânicos, existem diversos exemplos bem sucedidos de trabalhos que foram automatizados e que trouxeram grandes benefícios, podemos citar uma criação Brasileira que teve um grande impacto na sociedade e desperta interesse mundial, a urna eletrônica para votação trouxe muitos benefícios, agilidade e segurança na apuração dos votos, são muito superiores se comparado ao sistema antigo manual, outro exemplo são as montadoras de veículos, que ganharam em produção e qualidade com a utilização de robôs para auxiliar a montagem dos veículos. Todos esses trabalhos dependem de sistemas computacionais para serem executados, na verdade os sistemas computacionais são os grandes responsáveis pelo sucesso ou fracasso em se tentar automatizar um tarefa.

Com base no que foi dito procurou-se algo ainda não automatizado, algo que necessitasse de uma grande quantidade de pessoas para sua realização, essa tarefa pode ser o famoso CENSO no Brasil, uma pesquisa que coleta informações de todo país, essa coleta de informação é feita de forma totalmente manual, um funcionário do governo que chega em uma casa faz algumas perguntas, anota em um papel e posteriormente adiciona esses dados em algum sistema

computacional para a geração de resultados, podemos citar vários exemplos de pesquisas, pesquisa eleitoral, pesquisa de produtos, etc... O que elas tem em comum? Todas são feitas em com a ajuda do bom e velho papel e caneta. Provavelmente um funcionário que trabalhe 8 horas diárias utiliza 4 horas para fazer a pesquisa e as outras 4 horas para incluir os dados coletados em um sistema computacional que irá gerar resultados e armazenar os dados coletados. E se pudéssemos fazer com que em tempo de coleta dos dados os mesmos fossem enviados e armazenados no devido sistema computacional, poderíamos diminuir pela metade o numero de funcionários ou dobrar a produtividade dos mesmos, para automatizar essas tarefas seria necessário um sistema computacional, com isso a cada nova pesquisa necessitaria desenvolver um novo sistema. Porém todos os sistemas seguem o mesmo padrão, uma pergunta seguido de uma resposta, isso é repetido até ser satisfeito algum critério estabelecido. Então vem a pergunta e se pudéssemos gerar um gerador de formulários de pesquisa.

1.1 Objetivo Geral

Construir um sistema computacional, no qual será possível gerar formulários de pesquisa para serem executados em dispositivos móveis como por exemplo celular, Palm Tops, etc...

1.2 Objetivo Específico

Explorar a plataforma de desenvolvimento Java J2ME e a integração com um sistema web. O sistema gerador de formulários estará acessível através de um servidor web, o sistema será desenvolvido em PHP, e irá gerar um arquivo XML contendo o formulário de pesquisa, então o sistema J2ME interpretará esse arquivo XML e montará o formulário de pesquisa, para isso uma máquina virtual Java deverá estar instalada nos dispositivos móveis, essa máquina virtual é instalada pelo fabricante do aparelho e no presente momento (2006) a maioria dos aparelhos satisfaz esse requisito.

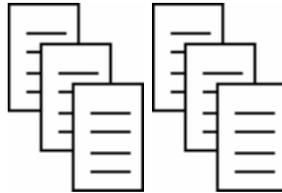
1.3 Modelo gráfico dos sistemas

Modelo sem a utilização dos sistemas propostos

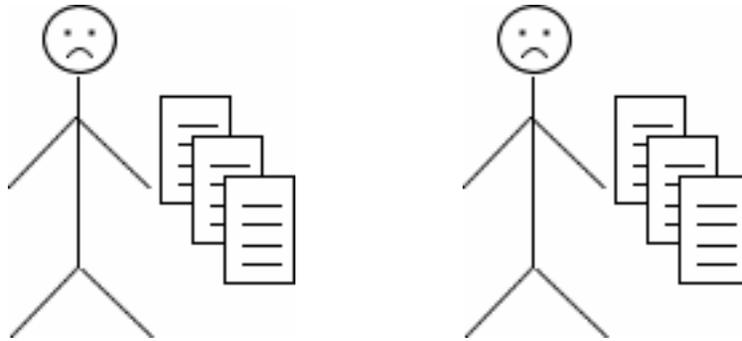
1 - Criação do formulário de pesquisa em papel



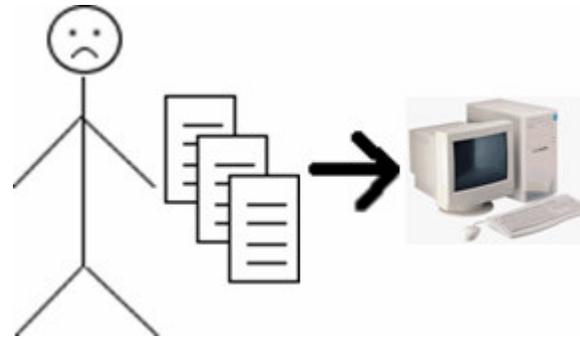
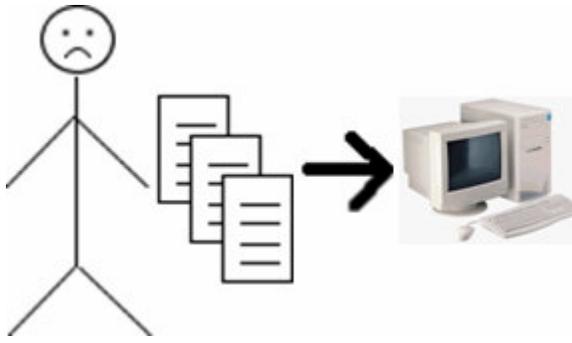
2 – Impressão de cópias em papel



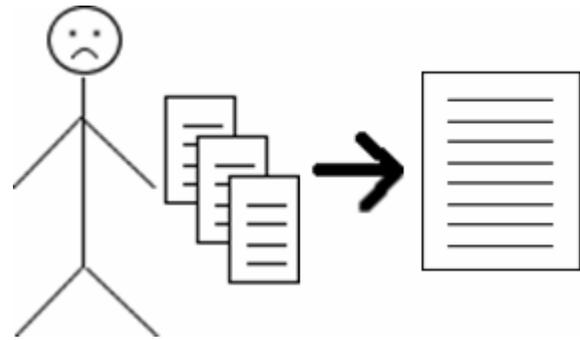
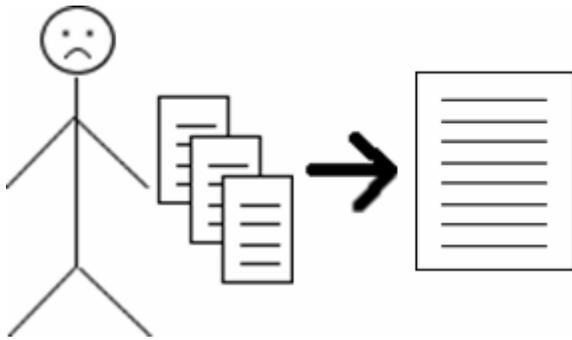
3 – Funcionários realizam a pesquisa e anotam os resultados em papel



4 – Dados coletados são inseridos manualmente em um sistema computacional que irá gerar as estatísticas, ou as estatísticas são geradas manualmente.



ou

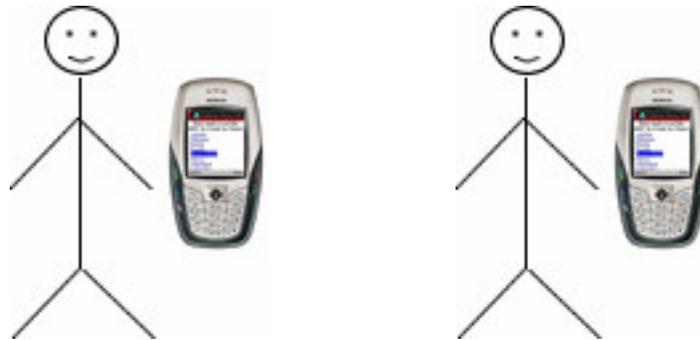


Modelo com a utilização dos sistemas propostos

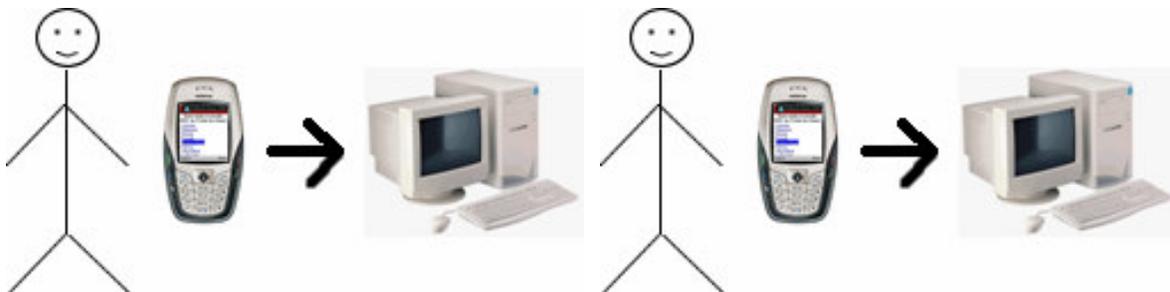
1 - Criação do formulário de pesquisa no sistema Web



2 – Funcionários utilizam um aparelho de celular para realizar as pesquisas que estão disponíveis e anotam os resultados no próprio dispositivo



4 – Dados coletados são enviados para o sistema Web através do sistema utilizado para realizar a pesquisa, economizando assim tempo de trabalho e possíveis erros de manipulação dos dados



2 TECNOLOGIAS UTILIZADAS

2.1 Especificação do sistema móvel(J2ME)

2.1.1 Introdução

Java 2 Micro Edition é uma API Java voltada para micro aplicativos que rodam em micro processadores assim como os dos celulares e PDA s.

J2ME basicamente é dividido em configurations, profiles e API's opcionais.

2.1.2 Configuration

A configuration mais básica do J2ME é o CLDC (Connected Limited Device Configuration) e também a mais importante pois ela da base para a outra configuration, CDC (Connected Limited Configuration) a qual prove algumas funcionalidades a mais, e requer um pouco mais de hardware. Configuration basicamente provê algumas bibliotecas básicas e a VM (virtual machine), ou seja, a parte mais básica que necessita estar presente em todo e qualquer dispositivo que suporta essa tecnologia.

2.1.3 Profiles

Um profile, nada mais é que uma série de API s padrões que combinadas com alguma configuration, neste caso o CLDC, prove um serviço completo para que aplicações possam ser rodadas. A profile utilizada no trabalho é a MIDP.

Celulares, em sua quase totalidade hoje no mercado trabalha com o MIDP1.0 e CLDC1.0.

Profiles são mais específicos que configurations, e fazendo analogia a um velho exemplo, uma abstração sobre o que é um carro e como ele é fabricado (configuration) e como um Chevrolet é fabricado (profile), mais tecnicamente falando profile é baseado em configuration e ainda acima dos profiles estão as API's que na nossa analogia seria um modelo específico da Chevrolet.

Como já foi citado, existem dois "configurations", um configuration é o CLDC (Connected, Limited Device Configuration), que rege as configurações para aparelhos bem pequenos como celulares ou PDA's, o qual fica acima das diretrizes J2ME juntamente com CDC (Connected Device Configuration) o que rege as configurações para aparelhos um pouco maiores, mas mesmo assim pequenos.

Podem haver vários Profiles vamos citar dois aqui os quais são os mais importantes, MIDP (Mobile Information Device Profile) e também o PDAP (Personal Digital Assistant Profile) e ambos estão acima do CLDC.

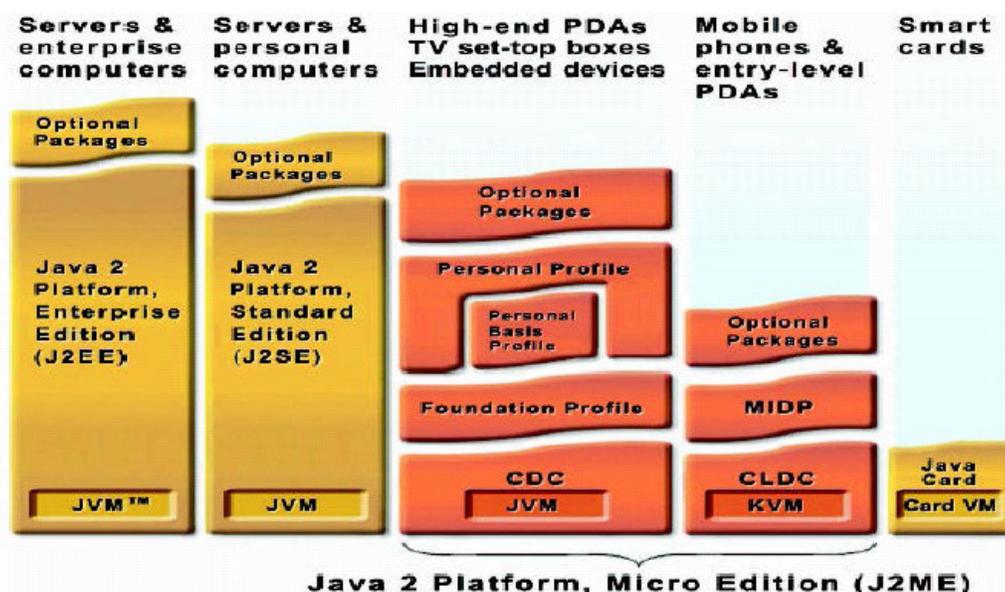


Figura 1: Plataforma Java

CLDC rege as configurações para aparelhos extremamente pequenos, ele foi desenvolvido para dispositivos de no mínimo 160KB e algumas dezenas de MHz de processamento. o processamento é muito fraco, mas muito mesmo, pois no mínimo 16MHz são requeridos, e se tratando de aparelhos tops de linha podem chegar até 300MHz ou um pouco mais, porém é muito difícil dizer ao certo pois o acesso a informações desse tipo nas especificações dos fabricantes é bem restrito, e também a conexão lenta, tipicamente de 9.600bps para aparelhos TDMA/CDMA e de 128Kbps para GPRS/CDMA 1xRTT.

MIDP tem as seguintes características:

- Mínimo de 160KB de memória não-volátil para JAVA.
- Um processador de 16 bits ou 32 bits com um clock de no mínimo 16MHz.
- 32KB de memória volátil para tempo de execução.
- Pelo menos 192KB livres para Java.
- 8KB de memória não-volátil de para armazenamento de dados.
- Uma tela de pelo menos 96x54 pixels
- Capacidade de entrada de dados seja por teclado (do celular), teclado externo ou mesmo Touch-screen.
- Possibilidade de enviar e receber dados em conexão possivelmente intermitente e banda reduzida.

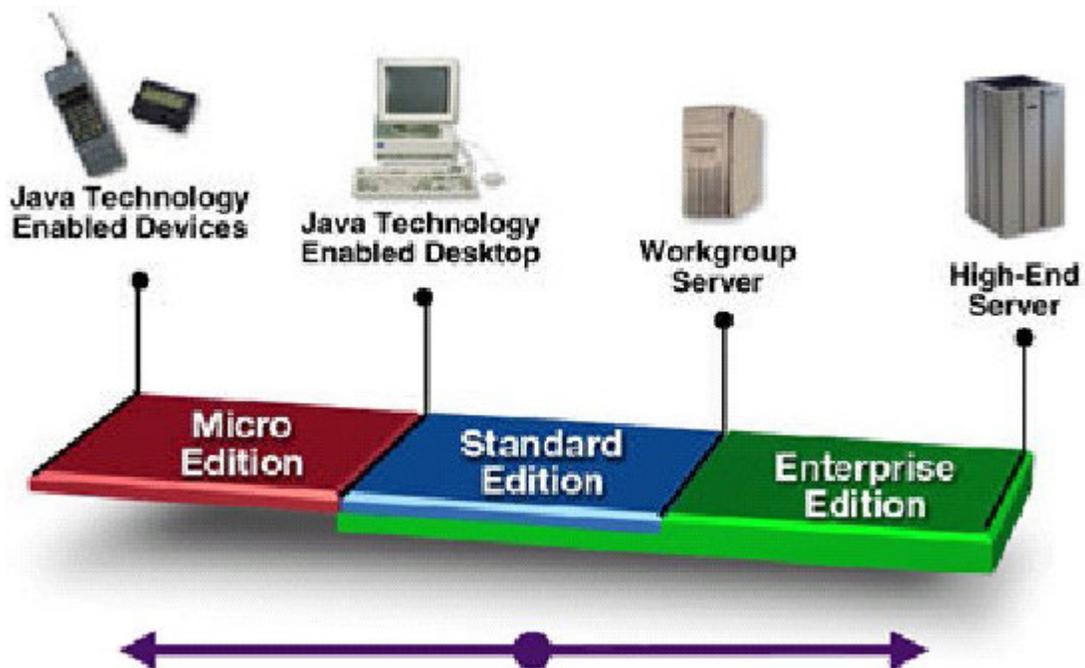


Figura 2: Dispositivos para as plataformas Java

2.1.4 Segurança

Uma pergunta freqüentemente feita é: Agora com programas rodando nos celulares não iremos ter problemas de vírus, ou programas maliciosos? Por meio do J2ME, muito difícil, porém celulares já foram atacados por vírus que exploram falhas do Sistema operacional e não de uma aplicação J2ME.

No caso dos jogos de celulares, cada fabricante disponibiliza uma API específica para os jogos, podendo assim aproveitar melhor o desempenho do aparelho, porém cai por terra a forte característica de vida do JAVA, a (WORA) "Write Once, Run Anywhere!", apesar de ser mais seguro, perde-se um pouco a funcionalidade. A maquina virtual tem um espaço independente de memória (sand-box), e não pode acessar a memória correspondente às aplicações nativas do celular assim como é feito com os Applets, um pouco diferente do J2SE.

2.1.5 Tecnologias Wireless

TDMA: (time division multiple access, ou acesso múltiplo por divisão do tempo), ou seja, ele não diferencia voz e dados então para navegar na internet(dados) você paga o mesmo preço que pagaria por fazer uma ligação de voz. Quem tem algum celular tdma sabe o quão caro e lento é isso. Somente para estabelecer a conexão leva-se uns 8 segundos e a transferência de dados dá-se a 9.600bps(conseguindo-se taxa máxima).

CDMA: (code division multiple access, ou acesso múltiplo por divisão de códigos), ou seja, é separado voz de dados, e tanto esta tecnologia quando a GSM são conexão 100% ativa, ou seja, não existe este delay de conexão você requisita uma informação e ela vem diretamente. Além de a taxa de transferência que é maior chegando a 256Kbps usando CDMA 1xRTT, que é o que esta em vigor.

GSM: (global system for communication, ou sistema global de comunicação), ou seja, tem as mesmas características da tecnologia CDMA, porém opera usando GPRS, que aumenta as taxas de transferência de dados, GPRS permite o transporte de dados por pacotes (Comutação por pacotes), sendo assim oferece uma taxa de transferência de dados muito mais elevada que as taxas de transferência das tecnologias anteriores, que usavam comutação por circuito, que eram em torno de 12kbps. Já o GPRS, em situações ideais, pode ultrapassar a marca dos 170kbps. No entanto na prática, essa taxa está em torno dos 40 kbps.

2.1.6 Diferença entre MIDP 1.0 e 2.0

Existem algumas diferenças entre as versões. Além desses métodos a mais, e algumas facilidades maiores da 2.0 como classes para jogos, e tratamento de sons, temos a diferença na comunicação de dados que é o mais interessante aqui como, por exemplo, as conexões no CLDC1.0 são feitas através do protocolo http que no caso é inseguro no envio de informações, já na 2.0 está implementado o https (ssl) conexão com criptografia.

Packages disponíveis para as versões.

java.lang

java.lang.ref (somente CLDC1.1)

java.io

java.util

javax.microedition.io

javax.microedition.lcdui

javax.microedition.lcdui.game (somente na 2.0)

javax.microedition.media (somente na 2.0)

javax.microedition.media.control (somente na 2.0)

javax.microedition.midlet

javax.microedition.rms

javax.microedition.pki (somente na 2.0)

Opcionalmente, fabricantes podem fornecer API's JAVA para acesso a partes específicas de cada aparelho. Com relação à segurança, o CLDC1.1 traz bastante

recursos, e já se aproxima mais do J2SE, com Permissions Types, Protection Domains baseados em IP's e PKI's.

2.1.7 API's

A CLDC 1.0 não possui suporte a ponto flutuante, ou seja, para trabalhar com contas, e valores reais terá que manuseá-los em código mesmo. A versão 1.1 do CLDC já possui suporte a ponto flutuante, como no trabalho o dispositivo não realizara cálculos com a informação coleta, essa restrição não afetara o desenvolvimento.

2.1.8 Outras restrições

- Sem user classLoading
- Sem finalização de objetos (finalize() de java.lang.Object) Garbage collector existe, porém não executa método automático.
- Sem RMI e sem JINI
- Sem métodos nativos, a não ser os fornecidos pela JVM nativa.
- MultiThreading (sem interrupt(), pause(), resume() e stop())
- Sem thread groups e thread Deamons

2.1.9 Ciclo de Vida de uma MIDlet

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. As classes de cada aplicativo estão em um arquivo JAR, o qual vem acompanhado de um descritor JAD.

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 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()`.

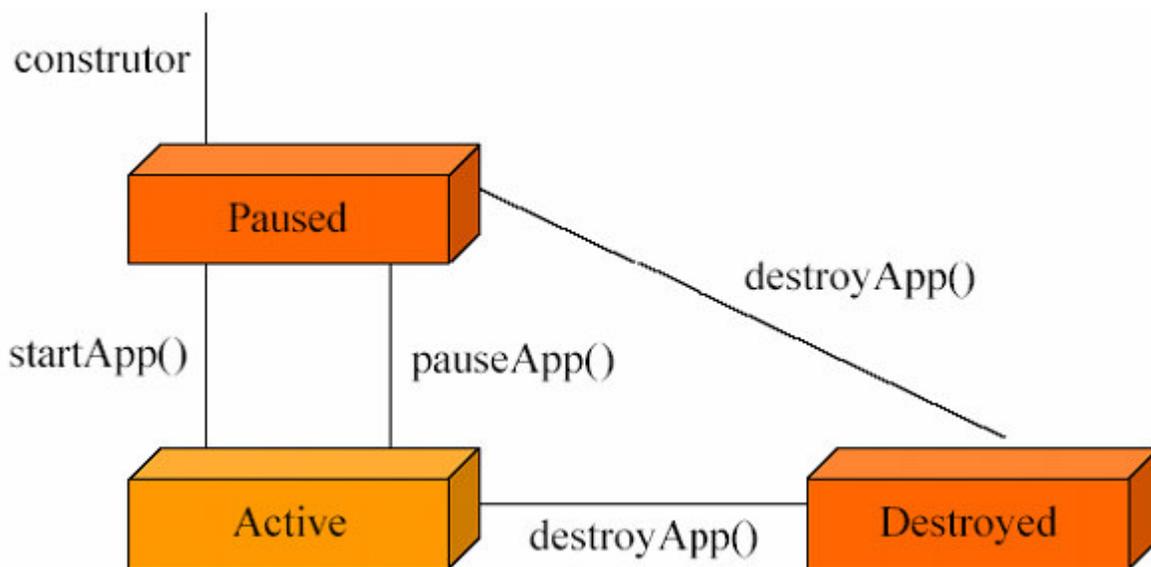


Figura 3: Ciclo de vida de uma Midlet

2.1.10 Interface

As MIDlets devem poder ser executadas em qualquer dispositivo, contendo a VM, sem alterações porém isso torna-se bastante difícil na parte de Interface com usuário, pois dispositivos variam de tamanho de tela, cores, teclados, touch-Screens e outros aspectos.

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

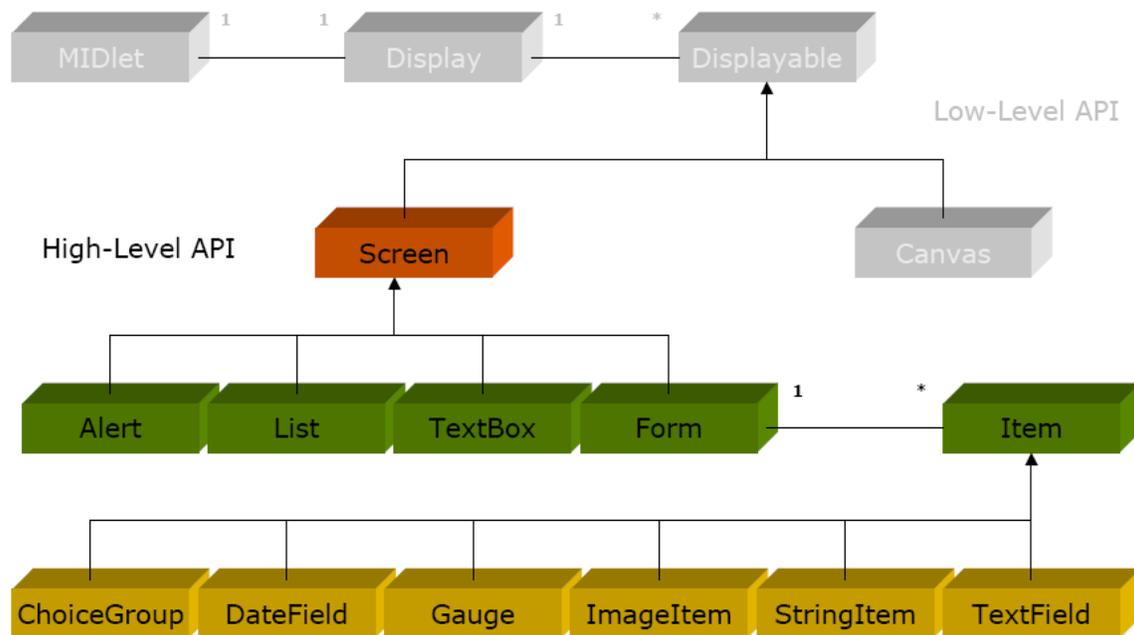


Figura 4: API para construção de interface gráfica

2.1.11 Armazenamento em celulares (RecordStores) - RMS

A persistência em celulares é tratada com RecordStores que são conjuntos de registros e instâncias de `javax.microedition.rms.RecordStore`. Cada recordStore é identificado por um nome que deve ser único, sua criação é feita de forma simples e bem sugestiva, da seguinte forma:

```
RecordStore.openRecordStore( NomeQualquer , true);
```

O parâmetro boolean da chamada do método indica que se true o RecordStore será aberta e se não existir será criada. O false apenas abre uma RecordStore existente.

A estrutura de Armazenamento de um RecordStore é bem simplória, possui apenas um id e um array de Bytes como local para armazenagem de dados.

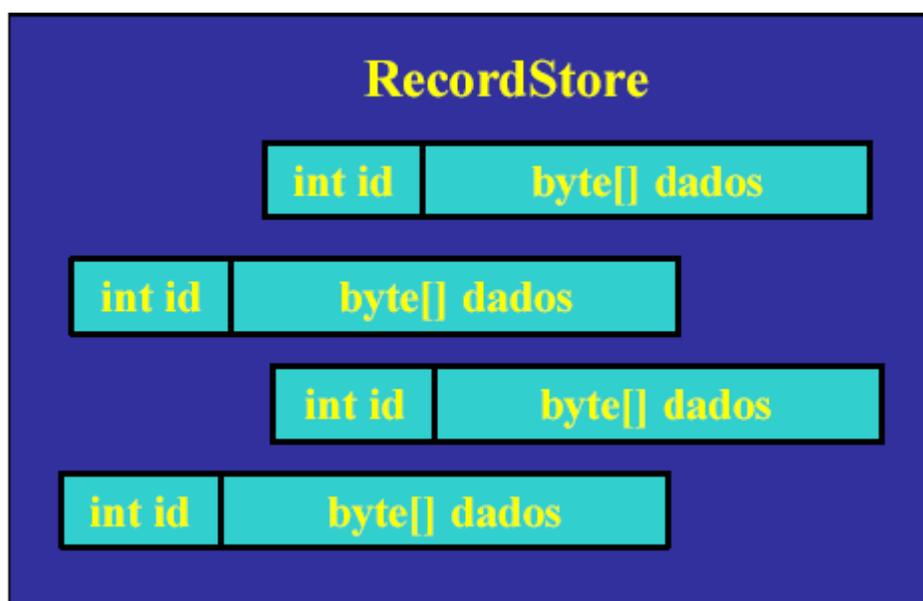


Figura 5: Estrutura de uma Record Store

2.1.12 KXML(leitor de arquivos XML)

Arquivos XML são muito utilizados para intercâmbio de dados entre sistemas, como no trabalho será feita a comunicação entre um sistema web e um J2ME, o uso de XML é uma excelente alternativa, além disso uma possível utilização dos dados por um outro sistema seria possível sem nenhuma alteração no que já foi desenvolvido.

Foram pesquisadas algumas API's para leitura de arquivos XML em J2ME, as mais utilizadas são:

Name	License	Size	MIDP
ASXMLP 020308	Modified BSD	6 kB	yes
kXML 2.0 alpha	EPL	9 kB	yes
kXML 1.2	EPL	16 kB	yes
Xparse-J 1.1	GPL	6 kB	yes

Tabela 1: APIs para leitura de arquivos XML

Foi escolhida a biblioteca kXML 2.0. A seguir um exemplo de um arquivo XML e como é feita a leitura desta API

```
<elements>
  <text>text1</text>
  <text>text2</text>
</elements>
```

Código de leitura:

```
parser.nextTag();
parser.require(XmlPullParser.START_TAG, null, "elements");

while(parser.nextTag() == XmlPullParser.START_TAG) {
  parser.require(XmlPullParser.START_TAG, null, "text");

  // handle element content
  System.out.println("text content: "+ parser.nextText());

  parser.require(XmlPullParser.END_TAG, null, "text");
}

parser.require(XmlPullParser.END_TAG, null, "elements");
```

2.2 Especificação do sistema Web

2.2.1 Linguagem de programação

2.2.2 PHP

Em primeiro lugar gostaria de comentar que o sistema web foi originalmente feito em Java, utilizando frameworks como Struts, Spring, Freemarker, porém pela semelhança entre Java e PHP comecei a programar em PHP e notei um grande ganho de produtividade, após pouco tempo de uso, já percebi como a linguagem é muito parecida com Java na questão de Orientação à Objetos, e como esse é o paradigma de programação mais utilizado no momento pois traz inúmeras vantagens ao desenvolvimento de software, resolvi migrar para essa linguagem, então a seguir será mostrado as principais características de PHP, como será visto o desenvolvimento web em PHP não perde em nada para Java, e na minha opinião desenvolver em PHP é muito mais produtivo que em Java para aplicações deste porte.

PHP surgiu em 1994, porém só na versão 3 foi incluído o modelo de orientação a objetos, esse modelo ainda era muito primitivo, na versão 4 da linguagem o modelo foi melhorado, atualmente a maioria dos sistemas web e servidores web, foram feitos e dão suporte a essa versão, porém em junho de 2004 foi lançada a versão 5 da linguagem, que é a última versão, nessa versão o modelo de Orientação à Objetos foi totalmente reescrito, resultando em uma linguagem com praticamente todos os recursos existentes em outras linguagens como por exemplo Java que é uma referência, atualmente a linguagem mais utilizada junto

ao C++, abaixo um gráfico das linguagens mais utilizadas em projetos no ano de 2006/2007

Posição Jun 007	Posição Jun 2006	Variação de Posição	Linguagem de programação	Status
1	1	=	Java	A
2	2	=	C	A
3	3	=	C++	A
4	4	=	(Visual) Basic	A
5	5	=	PHP	A
6	6	=	Perl	A
7	8	↑	C#	A
8	7	↓	Python	A
9	10	↑	JavaScript	A
10	19	↑↑↑↑↑↑↑↑	Ruby	A
11	13	↑↑	PL/SQL	A
12	11	↓	SAS	A
13	9	↓↓↓↓	Delphi	A
14	18	↑↑↑↑	D	A
15	15	=	Lisp/Scheme	B
16	21	↑↑↑↑↑	ABAP	B
17	16	↓	Ada	B
18	12	↓↓↓↓↓	FoxPro/xBase	B
19	14	↓↓↓↓↓	COBOL	B
20	22	↑↑	Fortran	B

Tabela 2: Utilização das linguagens de programação em 2006/2007

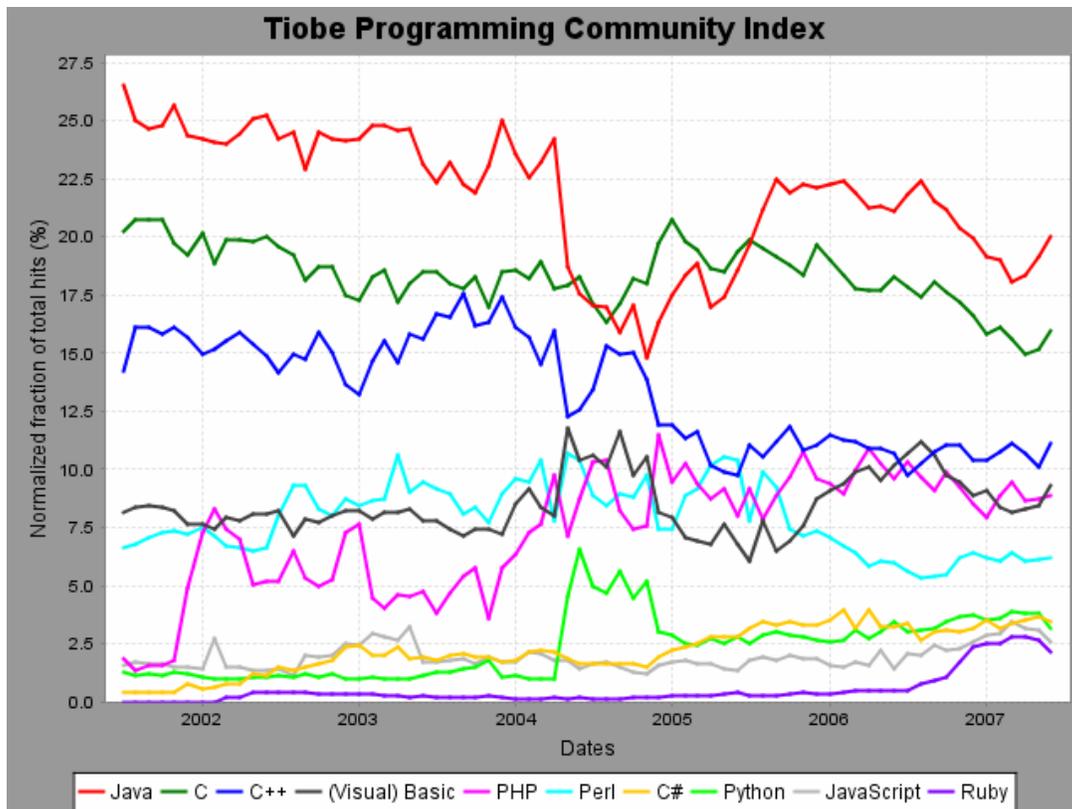


Figura 6: Gráfico do uso das linguagens de programação

Bom então será mostrado as principais características de PHP versão 5, que foi a utilizada no trabalho

- Nomes fixos para os construtores e destrutores

Em PHP 5 utilizamos nomes pré-definidos para os métodos construtores e destrutores. Para se criar um objeto chamamos o método `__construct()` e para destruir `__destruct()`, em PHP não é preciso destruir os objetos assim como também não é preciso fazê-lo em Java porém caso queiramos realizar algo a finalização de um objeto podemos implementar a funcionalidade dentro do método `__destruct()`;

- Acesso public, private e protected a propriedades e métodos

São os modificadores de acesso habituais da POO.

- Uso de interfaces

As interfaces são utilizadas para definir um conjunto de métodos que uma classe deve implementar. Uma classe pode implementar várias interfaces ou conjuntos de métodos. Na prática, o uso de interfaces é utilizado muitas vezes para suprir a falta de herança múltipla assim como em Java.

- Métodos e classes final

Indicar que um método é "final" significa não se permitir sobrescrever esse método em uma nova classe que o herde. Se a classe é "final" isso indica que esta classe não permite ser herdada por outra classe.

- Operador instanceof

Utiliza-se para saber se um objeto é uma instância de uma classe determinada.

- Atributos e métodos static

Atributos e métodos podem ser "static" isso significa que podem ser acessados

sem a necessidade de haver instanciado um objeto de tal classe.

- Classes e métodos abstratos

Também é possível criar classes e métodos abstratos.

As classes abstratas não podem ser instanciadas, os métodos abstratos não podem ser chamados.

- Constantes de classe

Pode-se definir constantes dentro da classe, podendo ser acessadas através da própria classe.

PHP é a sigla de Hypertext Preprocessor, uma linguagem que nasceu para o desenvolvimento web, hoje já existe um versão para desenvolvimento desktop chamada PHP GTK, porém as características dessa linguagem não serão abordados aqui pois não foi utilizada no trabalho.

Uma vez definida a linguagem, começou-se a procurar frameworks para auxiliar e organizar o desenvolvimento, como o sistema já tinha sido feito em Java com os frameworks citados, foi feito uma procura por frameworks semelhantes em PHP para não impactar tanto na migração de uma linguagem para outra, e felizmente encontrou-se frameworks no mínimo iguais aos de Java, porém na minha opinião são ainda melhores que os de Java, então para começar, antes de mais nada o desenvolvimento web é feito com o modelo MVC

2.2.3 O Design Pattern MVC

No design pattern MVC, M significa Model, V-View, C-Controller.

Nesse sistema o fluxo da aplicação é mediado por um controlador central (Controller). O controlador delega requisições para um dos tratadores apropriados, que estão localizados no modelo (Model), que representa a lógica de negócio e o estado da aplicação. A requisição então é respondida, através do controlador, e

apresentada na visão (View), da maneira adequada. Esse modelo independente da linguagem de programação é um padrão de projeto.

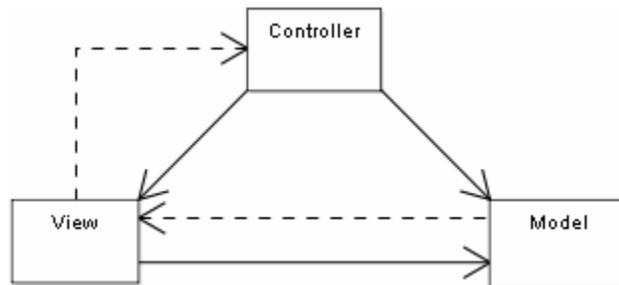


Figura 7: Modelo MVC

No sistema desenvolvido nesse trabalho, será utilizado os seguintes frameworks para cada camada:

M – Lumine

V – Smarty

C – PHPMVC

2.2.4 M – Lumine

Camada Model, essa camada de software é responsável pela persistência dos dados, foi utilizado o framework Lumine, o qual é um projeto Brasileiro, muito utilizado na comunidade de desenvolvedores PHP,

O objetivo do Lumine é facilitar a construção de aplicações PHP dependentes de bases de dados relacionais, particularmente, facilitar o desenvolvimento das consultas e atualizações dos dados. O uso de ferramentas de mapeamento objeto relacional, como o Lumine diminuem a complexidade resultante da convivência de

modelos diferentes; o modelo orientado a objetos (da linguagem PHP) e o relacional (da maioria dos SGBDs).

O Lumine é responsável apenas pelo mapeamento das tabelas do modelo relacional para classes da linguagem PHP. As questões relacionadas ao gerenciamento de transações e a tecnologia de acesso à base de dados são de responsabilidade de outros elementos da infraestrutura da aplicação.

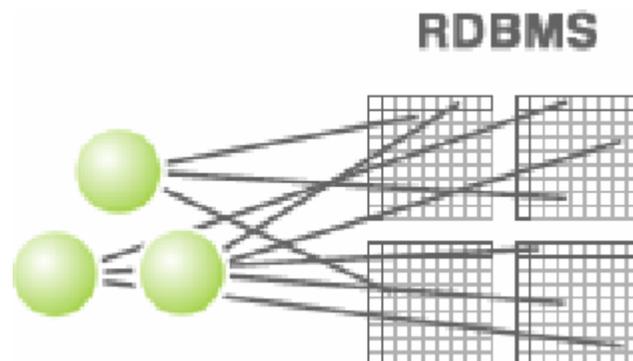


Figura 8: Como uma ferramenta Objeto-Relacional trabalha

Ferramentas de mapeamento objeto relacional nada mais são do que "tradutores" entre duas linguagens totalmente diferentes, a de Banco de Dados e a Orientada à Objetos.

No gráfico acima o que queria se mostrar é que por exemplo se temos um objeto que tem uma propriedade que é um array ou um conjunto de valores, quando esse objeto(bolinhas verdes) precisa ser salvo em banco, a ferramenta precisa mapear esse objeto para diferentes tabelas, como essa lista de valores precisará ser salva em outra tabela ele precisa fazer esse mapeamento adequadamente.

2.2.5 V – Smarty

O Smarty é um sistema de templates para PHP. Mais especificamente, ele fornece uma maneira fácil de controlar a separação da aplicação lógica e o conteúdo de sua apresentação. Isto é melhor descrito em uma situação onde o programador da aplicação e o designer do template executam diferentes funções, ou na maioria dos casos não são a mesma pessoa. Por exemplo, digamos que está se criando uma página para web para mostrar um artigo de um jornal. O autor, a manchete, a conclusão e o corpo do artigo são elementos de conteúdo, eles não contém informação alguma sobre como eles devem ser mostrados. Eles são enviados ao Smarty pela aplicação, então o designer do template edita o template e usa uma combinação de tags HTML e tags de templates para formatar a apresentação destes elementos (tabelas HTML, cores de fundo, tamanhos de fontes, folhas de estilos, etc.). Se algum dia o programador precisar alterar a maneira como o conteúdo do artigo é tratado (uma mudança na lógica da aplicação). Esta mudança não afeta o design do template, o conteúdo será enviado ao template exatamente da mesma forma. De modo semelhante, se o designer do template quiser redesenhar completamente os templates, não é necessária nenhuma alteração na lógica da aplicação. Sendo assim, o programador pode fazer mudanças na lógica da aplicação sem a necessidade de reestruturar os templates, e o designer do template pode fazer mudanças nos templates sem alterar a lógica da aplicação.

Algumas das características do Smarty:

- Ele é extremamente rápido.
- Ele é eficiente visto que o interpretador do PHP faz o trabalho mais pesado.
- Sem elevadas interpretações de template, apenas compila uma vez.
- Ele está atento para só recompilar os arquivos de template que foram mudados.
- Pode-se criar funções customizadas e modificadores de variáveis customizados, de modo que a linguagem de template é extremamente extensível.
- Os construtores `if/elseif/else/endif` são passados para o interpretador de PHP, assim a sintaxe de expressão `{if ...}` pode ser tanto simples quanto complexa da forma que se queira.
- É possível embutir o código PHP diretamente em seus arquivos de template, apesar de que isto pode não ser necessário (não recomendado) visto que a ferramenta é totalmente customizável.
- Suporte de caching embutido

2.2.6 C – PHPMVC

Este framework é o concentrador do sistema, todas as requisições passam por ele, ele controla o fluxo do sistema, nele são inseridas as chamadas a camada de Persistência e posteriormente a camada de Visão, embora ele seja muito importante é o mais simples de todos os frameworks.

3 Projeto Sistema Web

3.1 Informações Preliminares

Desenvolver um sistema web para administração das pesquisas. Será possível analisar as estatísticas de cada pesquisa.

3.2 Domínio de negócios

Com esse sistema será possível criar e gerenciar pesquisas, funcionários, grupos de funcionários, após uma pesquisa ser criada será possível visualizá-la no dispositivo móvel do funcionário, que irá efetuar a pesquisa, uma vez concluída a pesquisa no dispositivo móvel este envia os resultados coletados para esse sistema web, que por sua vez irá gerar todas as estatísticas definidas no sistema.

O sistema possui 2 visões

- Administrador: será responsável pelo gerenciamento das pesquisas, funcionários, grupos de funcionários.
- Funcionário: este irá efetuar a pesquisa com o dispositivo móvel.

3.3 Requisitos Funcionais

Função do Administrador

Login com nome de usuário e senha

Exibir uma lista com as pesquisas criadas

Exibir uma pesquisa

Alterar um pesquisa
Remover uma pesquisa
Exibir as estatísticas de uma pesquisa
Cadastrar um funcionário
Alterar um funcionário
Remover um funcionário
Criar um grupo de funcionários
Alterar um grupo de funcionários
Remover um grupo de funcionários
Vincular um grupo de funcionários à uma pesquisa

Função do Funcionário

Login com nome de usuário e senha
Exibir uma lista com as pesquisas que está participando
Exibir as estatísticas de uma pesquisa
Alterar dados cadastrais

3.4 Requisitos não-funcionais

Segurança: através de níveis de permissão só as pessoas autorizadas pelo sistema poderão usá-lo.

Usabilidade: o sistema será de fácil compreensão, as pessoas que trabalham nessa área já estão acostumadas com a utilização de sistemas, então não terão dificuldades.

Portabilidade: o sistema será feito em PHP, dos servidores web que dão suporte a linguagem de programação, os que possuem o interpretador PHP instalado são a maioria.

Desempenho: por se tratar de um sistema web, vários fatores influenciam o desempenho para o usuário, dentre eles o tipo de conexão do usuário à internet e o servidor web, em relação ao sistema a parte de geração das estatísticas é o ponto de maior processamento e poderá ser afetado.

Confiabilidade: os servidores web fazem backup diariamente

4 Implementação do sistema Web

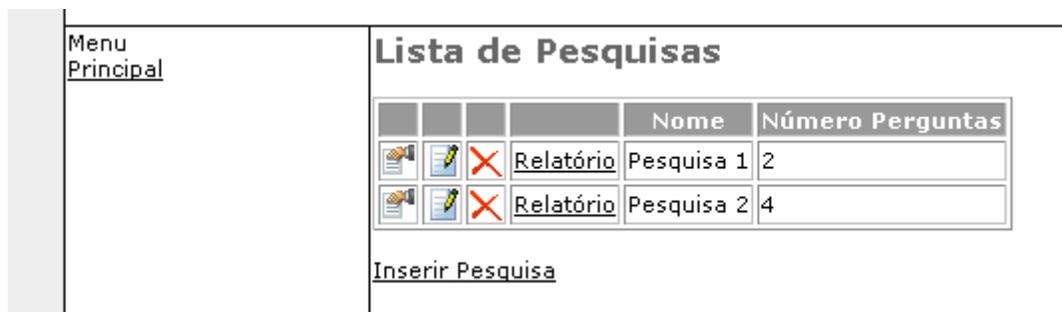
4.1 Ferramentas utilizadas

- Apache 2.2.3: servidor web
- PHP 5: interpretador PHP
- Mysql 5.0: Banco de dados relacional gratuito
- IDE Eclipse (versão 3.2) + Plugin para PHP
- Frameworks: PHPMVC, Lumine, Smarty

4.2 Descrição do sistema

Visão do Administrador

O administrador entrará no sistema com seus dados de acesso(usuário e senha), e será exibido uma lista com as pesquisas criadas, conforme imagem abaixo



The screenshot shows a web interface for an administrator. On the left, there is a vertical menu with the text "Menu Principal". The main content area is titled "Lista de Pesquisas" and contains a table with two rows of search data. Each row has three icons (a document, a pencil, and a red X) in the first three columns, followed by the search name and the number of questions. Below the table is a link labeled "Inserir Pesquisa".

				Nome	Número Perguntas
			Relatório	Pesquisa 1	2
			Relatório	Pesquisa 2	4

[Inserir Pesquisa](#)

Figura 9: Tela principal sistema Web

ele poderá então criar uma pesquisa, uma pesquisa só possui o campo nome, então ao clicar no link "Inserir Pesquisa" irá aparecer a janela abaixo

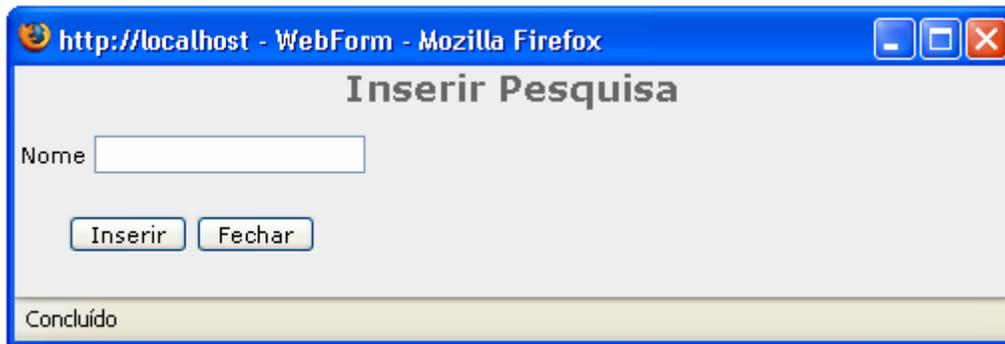


Figura 10: Formulário inserção pesquisa

Uma vez criada uma pesquisa, ele poderá alterar o nome, apagar e inserir perguntas para a pesquisa, ao clicar em visualizar a pesquisa o sistema exibirá a seguinte tela:



Figura 11: lista de perguntas para uma pesquisa

Nessa tela o Administrador poderá gerenciar a pesquisa, criando, alterando e removendo perguntas para a mesma. Ao clicar no link “Inserir Pergunta”, a seguinte tela será exibida:

The screenshot shows a web browser window with the title "http://localhost - WebForm - Mozilla Firefox". The main content area is titled "Inserir Pergunta". It contains a large text input field labeled "Pergunta". Below it are two checkboxes: "Permitir múltipla escolha" and "Marcar respostas certas?". Underneath these is a section labeled "Respostas - Digite as respostas:" followed by three text input fields. Each of these three fields has a red "X" icon to its right, indicating a delete function. Below the third field is a link that says "+ resposta". At the bottom of the form area are two buttons: "Inserir" and "Fechar". A status bar at the very bottom of the window displays the word "Concluído".

Figura 12: Formulário inserção de uma pergunta em um pesquisa

Nessa tela o Administrador insere a pergunta e as respostas nos respectivos campos, ele pode clicar em "+resposta" para que um novo campo de resposta apareça ou poderá clicar na imagem  para remover a resposta.

Ainda é possível configurar a pergunta com 2 opções, são elas:

- Permitir múltipla escolha: essa opção serve para que quando o Funcionário estiver fazendo a pergunta a uma pessoa, ele possa marcar mais de uma resposta para a pergunta.

-Marcar Respostas certas: ao selecionar essa opção o Administrador poderá informar quais resposta estão certas, essa é outra forma de pergunta, uma pergunta na qual existe um ou mais respostas certas, então para que ele possa selecionar as certas o sistema adiciona um campo no começo de cada resposta para que o Administrador selecione as certas, conforme imagem abaixo:

http://localhost - WebForm - Mozilla Firefox

Inserir Pergunta

Pergunta

Sua pergunta aqui

Permitir múltipla escolha

Marcar respostas certas?

Respostas - Digite as respostas:

sua resposta 1

sua resposta 2

sua resposta 3

[+ resposta](#)

javascript:removerResposta(1);

Figura 13: Formulário inserção de uma pergunta com opções avançadas em um pesquisa

Posteriormente ele poderá alterar uma pesquisa reconfigurando a mesma com todas as opções apresentadas

The screenshot shows a web browser window with the address bar displaying 'http://localhost - WebForm - Mozilla Firefox'. The main content area is titled 'Alterar Pergunta'. It contains a text input field with the value 'pergunta1'. Below this are two checked checkboxes: 'Permitir múltipla escolha' and 'Marcar respostas certas?'. The 'Respostas - Digite as respostas:' section features three text input fields labeled 'resposta1', 'resposta2', and 'resposta3'. Each field has a checkbox to its left and a red 'X' icon to its right. The first checkbox is checked, while the others are unchecked. Below the third field is a '+ resposta' link. At the bottom of the form are two buttons: 'Alterar' and 'Fechar'. A status bar at the very bottom of the window displays the word 'Concluído'.

Figura 14: Formulário para alteração de uma pergunta

Uma vez que as pesquisas estão criadas com as respectivas perguntas, já podem ser visualizadas no dispositivo móvel para que se possa efetuar a pesquisa com o público alvo.

Além do gerenciamento das pesquisas o Administrador do sistema deverá, criar os funcionários que irão realizar as pesquisas e adicioná-los em grupos de funcionários, quando o funcionário “logar” no sistema pelo dispositivos móvel, o sistema irá se comunicar com esse sistema web que irá retornar a lista de pesquisas que o mesmo pode realizar.

Também deve ser feito o vínculo do grupo de funcionário com cada pesquisa, isso serve para uma empresa poder criar várias equipes de funcionários, cada equipe/grupo realizará diferentes pesquisas, porém todas estarão dentro da mesma visão do Administrador.

4.3 Diagrama do Banco de Dados

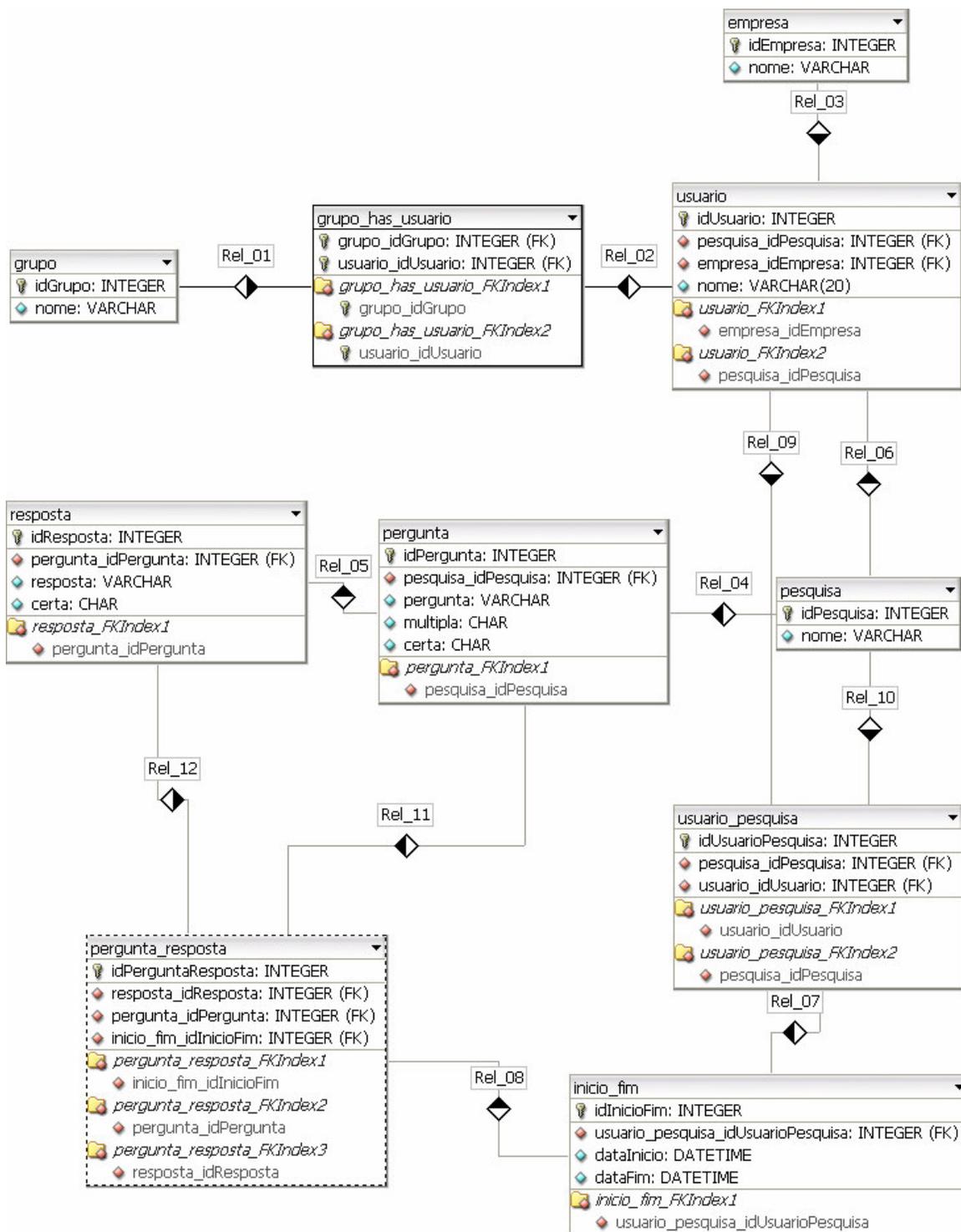


Figura 15: Diagrama do banco de dados

5 Projeto Sistema Móvel(J2ME)

5.1 Informações Preliminares

Desenvolver um sistema em J2ME, para a realização de uma pesquisa, substituindo o papel e caneta comumente utilizados.

5.2 Domínio de negócios

Com esse sistema será possível realizar uma pesquisa, respondendo as perguntas da mesma, e salvando os resultados no próprio dispositivo, em qualquer momento o Funcionário encarregado pela pesquisa poderá enviar os dados coletados para o sistema web que irá gerar as estatísticas com os resultados da pesquisa, o processo de envio é feito usando a Internet, esse é um dos grandes ganhos da utilização do sistema, ele acaba com qualquer possível erro de entrada dos dados para serem analisados e agiliza o processo.

5.3 Requisitos Funcionais

Login com nome de usuário e senha

Exibir uma lista com as pesquisas disponíveis para o Funcionário

Exibir uma pesquisa

Iniciar uma pesquisa

Salvar os dados de resposta de uma pesquisa

Enviar resposta da pesquisa para o servidor web

Excluir respostas das pesquisas

5.4 Requisitos não-funcionais

Segurança: através do login e senha o aplicativo só pode ser utilizado por pessoas que foram previamente cadastradas no sistema web pelo Administrador

Usabilidade: o sistema será de fácil gerenciamento, as pesquisas poderão ser efetuadas no aparelho móvel(celular) pessoais dos próprios funcionários, então eles já estarão habituados com o dispositivo e a navegação pelas teclas.

Portabilidade: o sistema será feito em J2ME, todos os celulares fabricados hoje já possuem a Máquina Virtual instalada. O sistema será feito em MIDP 1.0 e CLDC 1.0 que é a configuração mais básica.

Desempenho: o sistema será rápido o bastante para não causar nenhum atraso na execução da pesquisa, o que poderá ser afetado é o envio e recebimento dos dados ao servidor, mas isso dependerá do plano utilizado com a operadora.

Confiabilidade: os resultados das pesquisas serão salvas no dispositivos móvel e só serão marcadas como enviadas quando receber uma notificação do servidor web, garantido assim a integridade dos dados

6 Implementação do sistema J2ME

6.1 Ferramentas utilizadas

- WTK 2.2: Wireless Tool Kt da Sun, acompanha um simulador de dispositivos móveis

- RMS: persistência em dispositivos móveis

- kXML: leitor de arquivos XML

- IDE Eclipse (versão 3.2) + Plugin EclipseME

6.2 Descrição do sistema

O sistema J2ME é instalado no dispositivo móvel simplesmente fazendo o download de um arquivo “jad”, que está disponível no servidor web, quando se inicia o sistema uma tela de login é apresentada, conforme imagem abaixo.



Figura 16: Tela de entrada do sistema móvel(J2ME)

O login é autenticado no sistema web, o nome de usuário e senha utilizado pelo Funcionário para entrar no sistema web é o mesmo utilizado aqui, após o usuário se autenticar no sistema com os dados corretos, é exibida uma lista com as pesquisas que ele poderá realizar, essas pesquisas foram liberadas pelo administrador do sistema web

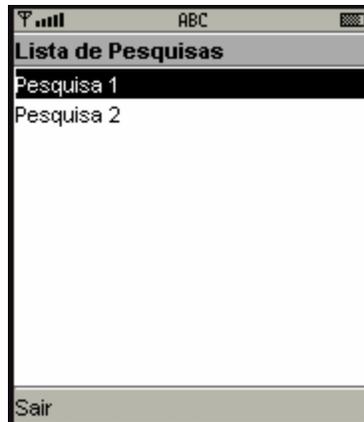


Figura 17: Lista de pesquisas disponíveis

Essas pesquisas são carregadas através de uma requisição ao servidor web, que responde com um arquivo XML com o seguinte formato

```
<?xml version="1.0" encoding="iso-8859-1"?>
<pesquisas>
  <pesquisa>
    <idPesquisa>10</idPesquisa>
    <nome>Pesquisa 1</nome>
  </pesquisa>
  <pesquisa>
    <idPesquisa>1</idPesquisa>
    <nome>Pesquisa 2</nome>
  </pesquisa>
</pesquisas>
```

Foi escolhido usar um arquivo XML pois esses dados poderiam ser enviados para qualquer outro sistema, o qual poderia interpretar o arquivo XML sem problemas.

Uma vez recebido a aplicação extrai os dados como o nome da pesquisa e exibe uma lista, conforme imagem exibida acima. O usuário então escolhe uma pesquisa e uma nova requisição http é feita pelo sistema móvel, para o servidor web, essa requisição é feita para trazer as perguntas da pesquisa, uma vez que cada pesquisa é carregada de uma vez, por uma questão de espaço de armazenamento no dispositivo. O servidor web responde com o seguinte XML

```

<?xml version="1.0" encoding="iso-8859-1"?>
<perguntas>
  <pergunta>
    <idPergunta>17</idPergunta>
    <pergunta>pergunta1</pergunta>
    <multipla>1</multipla>
    <certa>1</certa>
    <respostas>
      <resposta>
        <idResposta>226</idResposta>
        <resposta>resposta 1</resposta>
        <certa>1</certa>
      </resposta>
      <resposta>
        <idResposta>227</idResposta>
        <resposta>resposta2</resposta>
        <certa>0</certa>
      </resposta>
      <resposta>
        <idResposta>228</idResposta>
        <resposta>resposta3</resposta>
        <certa>0</certa>
      </resposta>
    </respostas>
  </pergunta>
  <pergunta>
    <idPergunta>18</idPergunta>
    <pergunta>pergunta2</pergunta>
    <multipla>0</multipla>
    <certa>0</certa>
    <respostas>
      <resposta>
        <idResposta>229</idResposta>
        <resposta>resposta 3</resposta>
        <certa>0</certa>
      </resposta>
      <resposta>
        <idResposta>230</idResposta>
        <resposta>resposta 4</resposta>
        <certa>0</certa>
      </resposta>
      <resposta>
        <idResposta>231</idResposta>
        <resposta>resposta 5</resposta>
        <certa>0</certa>
      </resposta>
    </respostas>
  </pergunta>
</perguntas>

```

</pergunta>
</perguntas>

Após receber os dados o sistema J2ME, exibe a seguinte tela

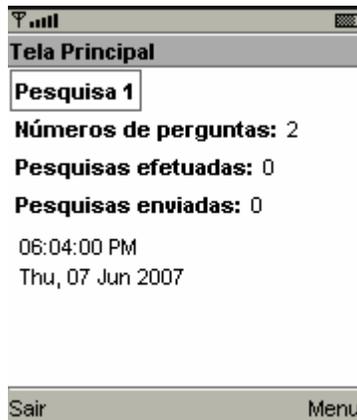


Figura 18: Tela principal de uma pesquisa

São exibidos:

- Nome da pesquisa
- Número de perguntas da pesquisa
- Número de pesquisas efetuadas e que estão salvas no dispositivo
- Número de pesquisas enviadas para o sistema web
- Data atual

O Usuário poderá selecionar o menu no canto direito da tela que será exibida as seguintes opções:

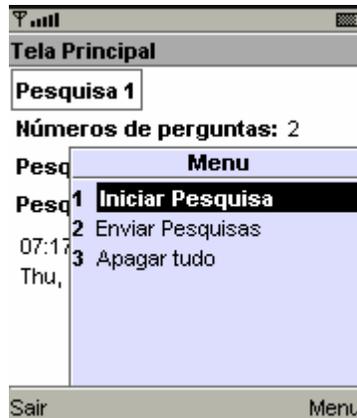


Figura 19: Opções disponíveis na tela principal

- Iniciar uma pesquisa: são exibidas então as perguntas recebidas do sistema web, pelo arquivo XML mostrado acima.
- Enviar Pesquisas: Envia as pesquisas efetuadas para o servidor web
- Apagar tudo: apaga todos os registros salvos no dispositivo

A seguir será exibida a tela da pesquisa propriamente dita, toda a idéia deste trabalho foi pensando em se chegar nesta tela, isto substitui o papel e caneta comumente utilizado para a realização de uma pesquisa, esta tela é carregada quando se seleciona a opção 1 do menu(Iniciar Pesquisa)

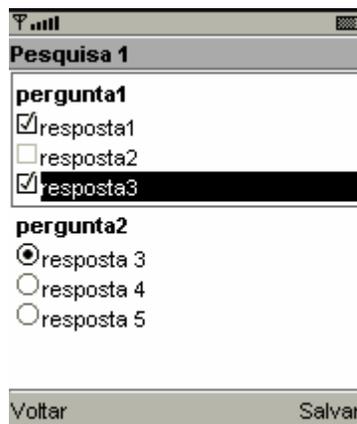


Figura 20: Realização de uma pesquisa

No exemplo existem 2 perguntas, a primeira conforme foi configurada no sistema web, foi abilitada para múltipla escolha, então podemos escolher mais de uma resposta. Na segunda pergunta essa opção não foi abilitada então só é permitido uma resposta.

Na parte inferior da tela temos 2 opções

- Voltar
- Salvar

Caso se selecione Voltar sem salvar os dados o sistema mostra um aviso alertando que os dados não serão salvos.

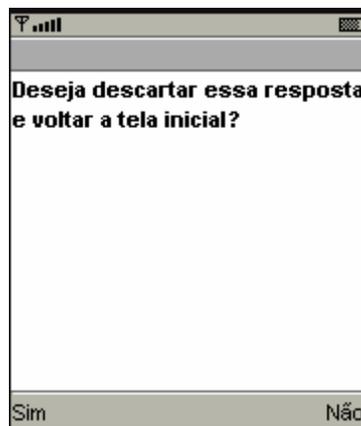


Figura 21: Tela de aviso, descarte de pesquisa

Caso o usuário tente salvar a pesquisa sem responder à uma pergunta o sistema também exibe um alerta

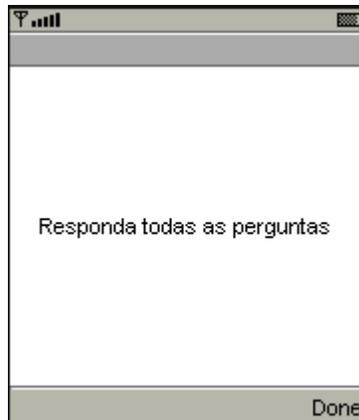


Figura 22: Tela de aviso, responda todas as perguntas

Quando se seleciona opção de salvar os dados da pesquisa o sistema salva os dados coletados utilizando um arquivo XML e salvando em um registro da estrutura RMS, já explicada, a estrutura do XML é semelhante ao recebido pelo sistema web, porém são adicionadas e retiradas algumas tags

Tags retiradas

- <pergunta>
- <múltipla>
- <certa>
- <resposta>
- <certa>

Tags adicionadas

- <dataInicio>: informa a data/hora em que uma pesquisa começou a ser realizada
- <dataFim>: informa a data/hora em que uma pesquisa foi finalizada
- <enviada>: controle para saber quais pesquisas foram enviadas, o sistema só envia para o servidor as pesquisas ainda não enviadas

```

<perguntas>
  <pergunta>
    <idPergunta>17</idPergunta>
    <respostas>
      <resposta>
        <idResposta>226</idResposta>
      </resposta>
      <resposta>
        <idResposta>227</idResposta>
      </resposta>
    </respostas>
  </pergunta>
  <pergunta>
    <idPergunta>18</idPergunta>
    <respostas>
      <resposta>
        <idResposta>229</idResposta>
      </resposta>
    </respostas>
  </pergunta>
  <dataInicio> 2007-06-07 19:47:25</dataInicio>
  <dataFim> 2007-06-07 19:52:25</dataFim>
  <enviada>0</enviada>
</perguntas>

```

Uma vez salva a pesquisa o sistema volta para a tela principal e atualiza os dados de controle

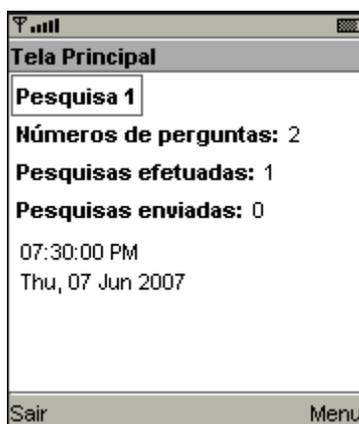


Figura 23: Tela principal, exibição dados atualizados

Uma vez existindo pesquisas efetuadas, é possível enviá-las para o servidor, selecionando o Menu > EnviarPesquisas, conforme foi exibido acima. O sistema J2ME simplesmente envia o XML salvo no dispositivo para o sistema web.

Por fim será exibido o sistema web, que recebeu os dados e poderá gerar as estatísticas para a pesquisa

Tratamento dos resultados da pesquisa pelo sistema web

De volta a tela principal do sistema

				Nome	Número Perguntas
			Relatório	Pesquisa 1	2
			Relatório	Pesquisa 2	4

[Inserir Pesquisa](#)

Figura 24: Lista de pesquisas, ênfase na opção “Relatório”

Ao clicar no link “Relatório”, é exibida a estatística de votação para cada resposta

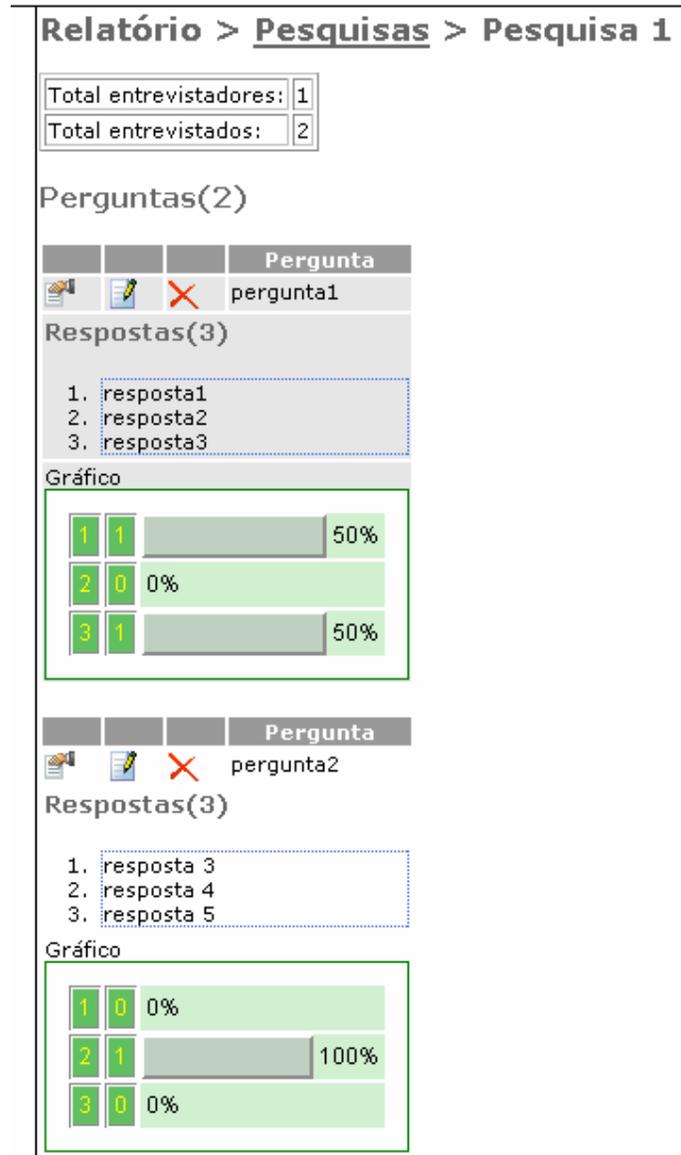


Figura 25: Estatísticas geradas para cada resposta de uma pergunta

Aqui é exibido

- Total Entrevistadores: número de funcionários que enviaram resultados da pesquisa para o servidor
- Total entrevistados: somatório de todas as pessoas entrevistadas.
- Abaixo de cada pergunta é exibido um gráfico com o número de vezes que cada resposta foi escolhida no sistema J2ME

7 Conclusão

Com a adoção desse sistema empresas podem economizar significativamente em vários aspectos, podem reduzir o número de pessoas envolvidas nas pesquisas, uma vez que essas pessoas só terão o trabalho de preencher um formulário eletrônico, todo o trabalho de inserção dos dados coletados e geração das estatísticas foi eliminado, isso passou a fazer parte do trabalho do sistema, com isso ganha-se em tempo e segurança já que possíveis erros na manipulação dos dados coletados são anulados com o sistema.

Outro fator facilitador da adoção desse sistema é o custo, o sistema que irá executar a pesquisa poderá rodar no telefone celular dos próprios empregados, uma vez que praticamente todas as pessoas que estão no mercado de trabalho possuem um celular, além disso hoje(2007) o custo de um aparelho de telefone celular é insignificante comparado com o ganho que a adoção do sistema traz.

Então uma vez atingidos os objetivos no desenvolvimento do sistema, mostrando-se eficiente na prática, as empresas que realizam esse tipo de trabalho de pesquisas, poderiam adotar essa solução e comprovar o ganho na prática.

8 Trabalhos Futuros

Com a utilização do sistema na prática várias melhorias e funcionalidades poderiam ser realizadas, dentre elas podemos citar:

- Utilização do protocolo HTTPS para a comunicação entre o sistema web e o móvel(J2ME) uma vez que os dados trafegam na Internet, para isso necessitaria mudar o configuration para CLDC1.1, pois na versão utilizada no trabalho CLDC 1.0 só existe suporte ao protocolo http

- Já que o sistema web exporta o formulário criado para um arquivo XML, poderia ser utilizado outros dispositivos para a realização da pesquisa, como por exemplo um PDA, um próprio formulário web.

- Todos os funcionários que estão realizando as pesquisas estão cadastrados no sistema web com o seu respectivo número do aparelho de celular, então poderia ser adicionada uma nova opção no sistema web para envio de SMS para os celulares dos funcionários uma vez que estes podem ser monitorados em tempo real desde de que enviem a resposta de uma pesquisa com uma grande freqüência.

- Poderia ser adicionado uma funcionalidade de envio de foto junto com o resultado de uma pesquisa, logicamente os celulares necessitariam de uma câmera digital, essa foto poderia ser utilizada na interpretação dos dados pelos seres humanos uma foto poderia revelar as condições do ambiente em que a pesquisa foi feita.

Referências Bibliográficas

1. DEITEL, HARVEY M.; DEITEL; PAUL J. Java: como programar, Bookman, 2002.
2. MUCHOW, John W. Core J2ME Technology & Midp
3. WHITE, James e HEMPHILL David. J2ME - Java in small things
4. MCGRAW, Hill. J2ME The Complete Reference
5. JULIANO NIEDERAUER, PHP 5: Guia de Consulta Rápida
6. TIM CONVERSE & JOYCE PARK, PHP: a Bíblia
7. Apostila de J2ME,
<http://portaljava.com/home/modules.php?name=Content&pa=showpage&pid=24>
8. Um pouco mais que “Hello World”,
http://www.guj.com.br/content/articles/j2me_hello/j2me_helloworld.pdf

Referências das Figuras

Figura 1 – Referências Bibliográficas(7)

Figura 2 – Referências Bibliográficas(7)

Figura 3 – Referências Bibliográficas(7)

Figura 4 – Referências Bibliográficas(8)

Figura 5 – Referências Bibliográficas(7)

Figura 6 – <http://www.tiobe.com/tpci.htm>

Figura 7 – <http://pt.wikipedia.org/wiki/MVC>

Figura 8 – <http://www.db4o.com/about/productinformation/>

Referências das Tabelas

Tabela 1 – <http://developers.sun.com/mobility/midp/articles/parsingxml/>

Tabela 2 – <http://www.tiobe.com/tpci.htm>

Anexo A - Código Fonte Sistema Web

AdministradorAction.php

```
<?
require_once("util/Action.php");
require_once("dao/PesquisaDAO.php");

class AdministradorAction extends Action {

    function AdministradorAction() {
        session_start();
        parent::__construct();
    }

    function listarPesquisa() {
        $listaPesquisa = PesquisaDAO::listarPesquisa();

        $this->smarty->assign('listaPesquisa', $listaPesquisa);
        $this->smarty-
>display('paginas/administrador/listarPesquisa.php');
    }
    function listarGrupo() {
        $listaGrupo = PesquisaDAO::listarGrupo();

        $this->smarty->assign('listaGrupo', $listaGrupo);
        $this->smarty-
>display('paginas/administrador/listarGrupo.php');
    }
    function listarFuncionario() {
        $listaFuncionario = PesquisaDAO::listarFuncionario();

        $this->smarty->assign('listaFuncionario',
$listaFuncionario);
        $this->smarty-
>display('paginas/administrador/listarFuncionario.php');
    }

    function exibirPesquisa() {
        $idPesquisa = $_REQUEST['idPesquisa'];
        $pesquisa = BaseDAO::exibir("Pesquisa",$idPesquisa);

        $this->smarty->assign('pesquisa', $pesquisa);
        $this->smarty-
>display('paginas/administrador/exibirPesquisa.php');
    }
    function exibirRelatorioPesquisa() {
        $idPesquisa = $_REQUEST['idPesquisa'];
        $pesquisa = BaseDAO::exibir("Pesquisa",$idPesquisa);
```

```

        $this->smarty->assign('pesquisa', $pesquisa);
        $this->smarty-
>display('paginas/administrador/exibirRelatorioPesquisa.php');
    }

    function exibirInserirPesquisaPopUp() {
        //$this->verificarSessao(4);
        $this->smarty-
>display('paginas/administrador/inserirPesquisaPopUp.php');
    }
    function exibirInserirPerguntaPopUp() {
        //$this->verificarSessao(4);
        $_POST['idPesquisa'] = $_REQUEST['idPesquisa'];
        $this->smarty-
>display('paginas/administrador/inserirPerguntaPopUp.php');
    }
    function exibirInserirGrupoPopUp() {
        //$this->verificarSessao(4);
        $this->smarty-
>display('paginas/administrador/inserirGrupoPopUp.php');
    }
    function exibirInserirFuncionarioPopUp() {
        //$this->verificarSessao(4);
        $this->smarty-
>display('paginas/administrador/inserirFuncionarioPopUp.php');
    }

    function inserirPesquisa() {
        //$this->verificarSessao(4);
        $pesquisa = Util::Import("bean.Pesquisa");
        $pesquisa->nome = $_REQUEST['nome'];
        $pesquisa->dataCadastro = time();

        //bug nao sei pq insere 2 vezes
        if ($pesquisa->nome==null)
            echo 1;
        else
            $pesquisa->save();

        $this->smarty->assign('msg', "Pesquisa inserida com
sucesso");
        echo "<script>".
            "window.opener.location.reload();" .
            "</script>";

        $this->smarty-
>display('paginas/administrador/inserirPesquisaPopUp.php');
    }
    function inserirPergunta() {
        //$this->verificarSessao(4);
        $pergunta = Util::Import("bean.Pergunta");
        $pergunta->pesquisa = $_REQUEST['idPesquisa'];

```

```

$pergunta->pergunta = $_REQUEST['pergunta'];
$pergunta->multipla = $_REQUEST['multipla'];
$pergunta->certa = $_REQUEST['certa'];
if (!isset($_REQUEST["multipla"]))
    $pergunta->multipla = '0';
if (!isset($_REQUEST["certa"]))
    $pergunta->certa = '0';

$textoResposta = $_REQUEST["textoResposta"];
$respostaCerta = $_REQUEST["respostaCerta"];

$listaResposta = array();
for ($i=0; $i < sizeof($textoResposta); $i++) {
    $resposta = new Resposta();
    $resposta->resposta = $textoResposta[$i];
    $resposta->certa = '0';

    if ($pergunta->certa=='1') {
        if (in_array($i, $respostaCerta))
            $resposta->certa = '1';
    }

    $listaResposta[] = $resposta;
}
$pergunta->listaResposta = $listaResposta;
//bug nao sei pq insere 2 vezes
if ($pergunta->pergunta==null)
    echo 1;
else
    $pergunta->save();

$this->smarty->assign('msg', "Pergunta inserida com
sucesso");
echo "<script>".
"window.opener.location.reload();" .
"</script>";

$this->smarty-
>display('paginas/administrador/inserirPerguntaPopUp.php');
}
function inserirGrupo() {
    //$this->verificarSessao(4);
    $grupo = Util::Import("bean.Grupo");
    $grupo->nome = $_REQUEST['nome'];

    $grupo->save();

    $this->smarty->assign('msg', "Grupo inserido com
sucesso");
    echo "<script>".
"window.opener.location.reload();" .
"</script>";
}

```

```

        $this->smarty-
>display('paginas/administrador/insertarGrupoPopUp.php');
    }
    function insertarFuncionario() {
        //$this->verificarSessao(4);
        $funcionario = Util::Import("bean.Usuario");
        $funcionario->nome = $_REQUEST['nome'];
        $funcionario->funcionario = '1';
        $funcionario->administrador = '0';

        if ($funcionario->nome == null)
            return;
        $funcionario->save();

        $this->smarty->assign('msg', "Funcionario inserido com
sucesso");
        echo "<script>".
            "window.opener.location.reload();".
            "</script>";

        $this->smarty-
>display('paginas/administrador/insertarFuncionarioPopUp.php');
    }

    function exibirAlterarPesquisaPopUp() {
        //$this->verificarSessao(4);
        $idPesquisa = $_REQUEST['idPesquisa'];
        $pesquisa = BaseDAO::exibir('Pesquisa',$idPesquisa);

        $this->smarty->assign('pesquisa', $pesquisa);
        $this->smarty-
>display('paginas/administrador/alterarPesquisaPopUp.php');
    }
    function exibirAlterarPerguntaPopUp() {
        //$this->verificarSessao(4);
        $idPergunta = $_REQUEST['idPergunta'];
        $pergunta = BaseDAO::exibir('Pergunta',$idPergunta);

        $this->smarty->assign('pergunta', $pergunta);
        $this->smarty-
>display('paginas/administrador/alterarPerguntaPopUp.php');
    }
    function exibirAlterarGrupoPopUp() {
        //$this->verificarSessao(4);
        $idGrupo = $_REQUEST['idGrupo'];
        $grupo = BaseDAO::exibir('Grupo',$idGrupo);

        $this->smarty->assign('grupo', $grupo);
        $this->smarty-
>display('paginas/administrador/alterarGrupoPopUp.php');
    }
}

```

```

function exibirAlterarFuncionarioPopUp() {
    //$this->verificarSessao(4);
    $idFuncionario = $_REQUEST['idFuncionario'];
    $funcionario =
BaseDAO::exibir('Usuario',$idFuncionario);

    $this->smarty->assign('funcionario', $funcionario);
    $this->smarty-
>display('paginas/administrador/alterarFuncionarioPopUp.php');
}

function alterarPergunta() {
    //$this->verificarSessao(4);
    $idPergunta = $_REQUEST['idPergunta'];
    $pergunta = BaseDAO::exibir("Pergunta",$idPergunta);
    $pergunta->pergunta = $_REQUEST['pergunta'];
    $pergunta->multipla = $_REQUEST['multipla'];
    $pergunta->certa = $_REQUEST['certa'];
    if (!isset($_REQUEST["multipla"]))
        $pergunta->multipla = '0';
    if (!isset($_REQUEST["certa"]))
        $pergunta->certa = '0';

    $textoResposta = $_REQUEST["textoResposta"];
    $respostaCerta = $_REQUEST["respostaCerta"];

    $listaResposta = array();
    for ($i=0; $i < sizeof($textoResposta); $i++) {
        $resposta = new Resposta();
        $resposta->resposta = $textoResposta[$i];
        $resposta->pergunta = $pergunta->idPergunta;
        $resposta->certa = '0';

        if ($pergunta->certa=='1') {

            if (in_array($i, $respostaCerta))
                $resposta->certa = '1';
        }

        $listaResposta[] = $resposta;
    }

    $pergunta->removeAll('listaResposta');
    $pergunta->listaResposta = $listaResposta;

    //bug nao sei pq insere 2 vezes
    if ($pergunta->pergunta==null)
        echo 1;
    else
        $pergunta->save();
}

```

```

        $this->smarty->assign('pergunta', $pergunta);
        $this->smarty->assign('msg', "Pergunta alterada com
sucesso");
        echo "<script>".
        "window.opener.location.reload();" .
        "</script>";

        $this->smarty-
>display('paginas/administrador/alterarPerguntaPopUp.php');
    }
    function alterarPesquisa() {
        //$this->verificarSessao(4);
        $idPesquisa = $_REQUEST['idPesquisa'];
        $pesquisa = BaseDAO::exibir("Pesquisa", $idPesquisa);
        $pesquisa->nome = $_REQUEST['nome'];

        //bug nao sei pq insere 2 vezes
        if ($pesquisa->nome==null)
            echo 1;
        else
            $pesquisa->save();

        $this->smarty->assign('pesquisa', $pesquisa);
        $this->smarty->assign('msg', "Pesquisa alterada com
sucesso");
        echo "<script>".
        "window.opener.location.reload();" .
        "</script>";

        $this->smarty-
>display('paginas/administrador/alterarPesquisaPopUp.php');
    }
    function alterarGrupo() {
        //$this->verificarSessao(4);
        $idGrupo = $_REQUEST['idGrupo'];
        $grupo = BaseDAO::exibir("Grupo", $idGrupo);
        $grupo->nome = $_REQUEST['nome'];

        $grupo->update();

        $this->smarty->assign('grupo', $grupo);
        $this->smarty->assign('msg', "Grupo alterado com
sucesso");
        echo "<script>".
        "window.opener.location.reload();" .
        "</script>";

        $this->smarty-
>display('paginas/administrador/alterarGrupoPopUp.php');
    }
    function alterarFuncionario() {

```

```

        // $this->verificarSessao(4);
        $idFuncionario = $_REQUEST['idFuncionario'];
        $funcionario =
BaseDAO::exibir("Usuario", $idFuncionario);
        $funcionario->nome = $_REQUEST['nome'];

        $funcionario->update();

        $this->smarty->assign('funcionario', $funcionario);
        $this->smarty->assign('msg', "Funcionário alterado com
sucesso");
        echo "<script>".
        "window.opener.location.reload();" .
        "</script>";

        $this->smarty-
>display('paginas/administrador/alterarFuncionarioPopUp.php');
    }

    function removerPesquisa() {
        $idPesquisa = $_REQUEST['idPesquisa'];
        $pesquisa = BaseDAO::exibir("Pesquisa", $idPesquisa);
        $pesquisa->delete();

        header("Location:
Main.php?do=administradorAction&action=listarPesquisa");
    }
    function removerGrupo() {
        $idGrupo = $_REQUEST['idGrupo'];
        $grupo = BaseDAO::exibir("Grupo", $idGrupo);
        $grupo->delete();

        header("Location:
Main.php?do=administradorAction&action=listarGrupo");
    }
    function removerFuncionario() {
        $idFuncionario = $_REQUEST['idFuncionario'];
        $funcionario =
BaseDAO::exibir("Usuario", $idFuncionario);
        $funcionario->delete();

        header("Location:
Main.php?do=administradorAction&action=listarFuncionario");
    }
    function removerPergunta() {
        $idPergunta = $_REQUEST['idPergunta'];
        $pergunta = BaseDAO::exibir("Pergunta", $idPergunta);
        $pergunta->delete();

        header("Location:
Main.php?do=administradorAction&action=exibirPesquisa&idPesquisa="
.$pergunta->pesquisa);
    }

```

```
}  
}  
?>
```

MEAction.php

```
<?  
require_once("util/Action.php");  
require_once("dao/PesquisaDAO.php");  
require_once("dao/UsuarioDAO.php");  
  
class MEAction extends Action {  
  
    function MEAction() {  
        session_start();  
        parent::__construct();  
    }  
  
    function login() {  
        $usuario = $_REQUEST['usuario'];  
        $senha = $_REQUEST['senha'];  
  
        $usuario = UsuarioDAO::exibirUsuarioLogin($usuario,  
$senha);  
        if($usuario->idUsuario == null) {  
            echo "0";  
        }  
        else {  
            echo SEPARADOR1.$usuario->idUsuario;  
        }  
    }  
    function listarPesquisa() {  
        $listaPesquisa = PesquisaDAO::listarPesquisa();  
  
        $string1 = "";  
        foreach ($listaPesquisa as $pesquisa)  
            $string1 .= $pesquisa->  
>idPesquisa.SEPARADOR1.$pesquisa->nome.SEPARADOR2;  
  
        $string1 = substr($string1,0,strlen($string1)-1);  
        echo $string1;  
    }  
    function listarPesquisaXML() {  
        header("Content-type: text/xml; charset=iso-8859-1");  
        $listaPesquisa = PesquisaDAO::listarPesquisa();  
  
        $this->smarty->assign('listaPesquisa', $listaPesquisa);  
    }  
}
```

```

        $this->smarty-
>display('paginas/me/listarPesquisaXML.php');
    }

function exibirPesquisa() {
    $idPesquisa = $_GET['idPesquisa'];
    $pesquisa = BaseDAO::exibir("Pesquisa", $idPesquisa);

    $string2 = "";

    $listaPergunta = $pesquisa->getLink('listaPergunta');
    foreach ($listaPergunta as $pergunta) {
        $string1 = $pergunta-
>idPergunta.SEPARADOR1.$pergunta->multipla.SEPARADOR1.$pergunta-
>pergunta.SEPARADOR2;
        $listaResposta = $pergunta-
>getLink('listaResposta');
        foreach ($listaResposta as $resposta)
            $string1 .= $resposta-
>idResposta.SEPARADOR1.$resposta->certa.SEPARADOR1.$resposta-
>resposta.SEPARADOR2;

            $string1 = substr($string1,0,strlen($string1)-1);
            $string2 .= $string1.SEPARADOR3;
        }

    $string2 = substr($string2,0,strlen($string2)-1);
    echo $string2;
}

function exibirPesquisaXML() {
    header("Content-type: text/xml; charset=iso-8859-1");
    $idPesquisa = $_GET['idPesquisa'];
    $pesquisa = BaseDAO::exibir("Pesquisa", $idPesquisa);

    $this->smarty->assign('pesquisa', $pesquisa);

    $this->smarty-
>display('paginas/me/exibirPesquisaXML.php');
}

function inserirResultadoPesquisa() {
    //$idPesquisa = $_POST['id'];
    //$pergunta = $_POST['p'];
    $idPesquisa = $_REQUEST['id'];
    $idUserio = $_REQUEST['u'];
    $pergunta = $_REQUEST['p'];
    //verificar se usuario_pesquisa ja existe, caso sim, usar ele
    $usuarioPesquisa =
PesquisaDAO::exibirUsuarioPesquisa($idUserio, $idPesquisa);
    if ($usuarioPesquisa->idUserioPesquisa==null) {
        $usuarioPesquisa =
Util::Import("bean.UsuarioPesquisa");
    }
}

```

```

        $usuarioPesquisa->usuario = $idUserario;
        $usuarioPesquisa->pesquisa = $idPesquisa;
        $usuarioPesquisa->save();
    }
    //--
//verificar se primeiro inicio e fim ja existe, caso sim, essa
pesquisa ja foi enviada
    $array1 = explode(SEPARADOR2,$pergunta[0]);
    $temp1 = $array1[0];
    $array2 = explode(SEPARADOR1,$temp1);
    $inicioFim = PesquisaDAO::exibirInicioFim($array2[0],
$array2[1]);
    if ($inicioFim->idInicioFim!=null) {
        echo "Pesquisa já enviada!, usando navegador";
        return;
    }
    //--

    for ($i=0; $i<count($pergunta); $i++) {

        $array1 = explode(SEPARADOR2,$pergunta[$i]);
        $temp1 = $array1[0];
        $array2 = explode(SEPARADOR1,$temp1);

        $inicioFim = Util::Import("bean.InicioFim");
        $inicioFim->usuarioPesquisa = $usuarioPesquisa-
>idUserarioPesquisa;
        $inicioFim->dataInicio = $array2[0];
        $inicioFim->dataFim = $array2[1];
        $inicioFim->save();
    for ($j=1; $j<count($array1); $j++) {

        $temp1 = $array1[$j];
        $array2 = explode(SEPARADOR1,$temp1);
        $idPergunta = $array2[0];
        for ($k=1; $k<count($array2); $k++) {
        $perguntaResposta = Util::Import("bean.PerguntaResposta");
            $perguntaResposta->usuarioPesquisa =
$usuarioPesquisa->idUserarioPesquisa;
            $perguntaResposta->inicioFim =
$inicioFim->idInicioFim;
            $perguntaResposta->pergunta =
$idPergunta;
            $perguntaResposta->resposta =
$array2[$k];
            $perguntaResposta->save();
        }
    }
}

    echo "tudoCerto";
}

```

```
}  
?>
```

PesquisaDAO.php

```
<?  
require_once("util/BaseDAO.php");  
require_once("bean/Resposta.php");  
require_once("bean/UsuarioPesquisa.php");  
  
class PesquisaDAO extends BaseDAO {  
  
    function exibirUsuarioPesquisa($idUserio, $idPesquisa) {  
        $obj = Util::Import('bean.UsuarioPesquisa');  
        $obj->whereAdd("idUserio=".$idUserio);  
        $obj->whereAdd("idPesquisa=".$idPesquisa);  
        $obj->find();  
        $obj->fetch();  
        return $obj;  
    }  
  
    function exibirInicioFim($dataInicio, $dataFim) {  
        $obj = Util::Import('bean.InicioFim');  
        $obj->  
>whereAdd("UNIX_TIMESTAMP(dataInicio)=".$dataInicio);  
        $obj->whereAdd("UNIX_TIMESTAMP(dataFim)=".$dataFim);  
        $obj->find();  
        $obj->fetch();  
        return $obj;  
    }  
  
    function exibirSomatorioPerguntaResposta($idPesquisa,  
$idPergunta, $idResposta) {  
        //LumineLog::setLevel(3);  
        //LumineLog::setOutput();  
  
        $lista = Util::Import('bean.PerguntaResposta');  
        $lista->selectAdd();  
        $lista->selectAdd("count(idPerguntaResposta)");  
        $usuarioPesquisa = new UsuarioPesquisa();  
        $usuarioPesquisa->pesquisa = $idPesquisa;  
        $lista->joinAdd($usuarioPesquisa);  
        $lista->whereAdd("idPergunta=".$idPergunta);  
        $lista->whereAdd("idResposta=".$idResposta);  
        $lista->find();  
        $lista->fetch();  
        $lista2 = $lista->toArray();  
        return array_pop($lista2);  
    }  
}
```

```

}
function exibirSomatorioInicioFim($idPesquisa) {
    $lista = Util::Import('bean.InicioFim');
    $lista->selectAdd();
    $lista->selectAdd("count(idInicioFim)");
    $usuarioPesquisa = new UsuarioPesquisa();
    $usuarioPesquisa->pesquisa = $idPesquisa;
    $lista->joinAdd($usuarioPesquisa);
    $lista->find();
    $lista->fetch();
    $lista2 = $lista->toArray();
    return array_pop($lista2);
}
function exibirTotalEntrevistadores($idPesquisa) {
    $lista = Util::Import('bean.UsuarioPesquisa');
    $lista->selectAdd();
    $lista->selectAdd("count(distinct(idUsuario))");
    $lista->pesquisa = $idPesquisa;
    $lista->find();
    $lista->fetch();
    $lista2 = $lista->toArray();
    return array_pop($lista2);
}
function listarPesquisa() {
    $lista = Util::Import('bean.Pesquisa');
    $lista->orderBy("dataCadastro desc");
    $lista->find();

    $lista2 = array();
    while ($lista->fetch())
        array_push($lista2,
BaseDAO::exibir('Pesquisa', $lista->idPesquisa));

    return $lista2;
}
function listarGrupo() {
    $lista = Util::Import('bean.Grupo');
    $lista->orderBy("nome asc");
    $lista->find();

    $lista2 = array();
    while ($lista->fetch())
        array_push($lista2,
BaseDAO::exibir('Grupo', $lista->idGrupo));

    return $lista2;
}
function listarFuncionario() {
    $lista = Util::Import('bean.Usuario');
    $lista->whereAdd("funcionario='1'");
    $lista->orderBy("nome asc");
    $lista->find();
}

```



```

        <tr>
            <td><input type="checkbox" name="certa" value="1" {if
$pergunta->certa}checked{/if} onclick="marcarCerta();" > Marcar
respostas certas?</td>
        </tr>
        <tr>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td><b>Respostas</b> - Digite as respostas:
                <table>
                    <tbody id="tabelaResposta">
                        {foreach from=$pergunta-
>getLink('listaResposta') item=resposta name=resposta}

                            <tr
id="tr{$smarty.foreach.resposta.iteration}">
                                <td>
                                    {if $pergunta->certa}
                                        <span
name='checkRespostaCerta' id="checkRespostaCerta"><input
type='checkbox' name='respostaCerta[]' {if $resposta-
>certa}checked{/if}></span>
                                        {else}
                                        <span
name='checkRespostaCerta' id="checkRespostaCerta"></span>
                                        {/if}
                                        <textarea
name='textoResposta[]' class="w350h50">{$resposta-
>resposta}</textarea><a
href="javascript:removerResposta({$smarty.foreach.resposta.iterati
on});" title="Remover"></a></div>
                                </td>
                            </tr>
                        {/foreach}
                    </tbody>
                </table>
                <a
href="javascript:exibirInserirResposta($('formulario'),'listaRespo
sta');">+ resposta</a>
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td align="center">
                <input type='submit' value="Alterar">
                <input type='button' value="Fechar"
onclick="javascript>window.close();" >
            </td>

```



```

{assign var="listaPergunta" value=$pesquisa-
>getLink('listaPergunta')}
<h3>Perguntas({$listaPergunta|@count})</h3>

<table border="0">
{foreach from=$listaPergunta item=pergunta name=pergunta}
    {if $smarty.foreach.pergunta.index%2==0}
        <tr class="par">
    {else}
        <tr class="impar">
    {/if}

    <th>&nbsp;</th><th>&nbsp;</th><th>&nbsp;</th><th>Pergunta</th>
>
</tr>
{if $smarty.foreach.pergunta.index%2==0}
    <tr class="par" valign="top">
{else}
    <tr class="impar" valign="top">
{/if}
{if}
    <td>
        <!--<a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birPergunta&idPergunta={$pergunta->idPergunta}',500,400);"></a-->
        <a href="javascript:alert('Nã
implementado');"></a>
        </td>
    <td><a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birAlterarPerguntaPopUp&idPergunta={$pergunta-
>idPergunta}',500,500,1);"></a></td>
        <td><a href="javascript:confirmarRemover('{ $pergunta-
>pergunta}', 'Main.php?do=administradorAction&action=removerPergunt
a&idPergunta={$pergunta->idPergunta}');" title="Remover"><a/></td>
        <td>{$pergunta->pergunta}</td>
    </tr>
{if $smarty.foreach.pergunta.index%2==0}
    <tr class="par">
{else}
    <tr class="impar">
{/if}
    <td colspan="5" align="left">
        {assign var="listaResposta" value=$pergunta-
>getLink('listaResposta')}
        <h4>Respostas({$listaResposta|@count})</h4>
        <ul id="listaResposta">

```

```

                                {foreach from=$listaResposta item=resposta
name=resposta}
                                <li>{$resposta->resposta}</li>
                                {/foreach}
                            </ul>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="5">&nbsp;</td>
                    </tr>
                {/foreach}
            </table>

```

```

<br>
<a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birInserirPerguntaPopUp&idPesquisa={$pesquisa-
>idPesquisa}',500,500,1);">Inserir Pergunta</a>

```

```
{include file="layout/rodape.php"}
```

exibirRelatorioPesquisa.php

```
{include file="layout/cabecalho.php"}
```

```

<h2>
Relatório > <a
href="Main.php?do=administradorAction&action=listarPesquisa">Pesqu
isas</a> > {$pesquisa->nome}
</h2>

```

```

<table border="1">
<tr>
    <td>Total entrevistadores:</td>
    <td>
        {php}
            $pergunta = $this->get_template_vars('pesquisa');
            $totalEntrevistadores =
PesquisaDAO::exibirTotalEntrevistadores($idPesquisa);
            echo $totalEntrevistadores;
        {/php}
    </td>
</tr>
<tr>
    <td>Total entrevistados:</td>
    <td>
        {php}
            $somatorioInicioFim =
PesquisaDAO::exibirSomatorioInicioFim($idPesquisa);
            echo $somatorioInicioFim;
        {/php}
    </td>
</tr>

```

```

        </td>
</tr>
</table>

{assign var="listaPergunta" value=$pesquisa-
>getLink('listaPergunta')}
<h3>Perguntas({$listaPergunta|@count})</h3>

<table border="0">
{foreach from=$listaPergunta item=pergunta name=pergunta}
    {if $smarty.foreach.pergunta.index%2==0}
        <tr class="par">
    {else}
        <tr class="impar">
    {/if}

    <th>&nbsp;</th><th>&nbsp;</th><th>&nbsp;</th><th>Pergunta</th>
>
</tr>
{if $smarty.foreach.pergunta.index%2==0}
    <tr class="par" valign="top">
{else}
    <tr class="impar" valign="top">
{/if}
    <td>
        <!--<a
href="javascript:popup('Main.php?do=administradorAction&action=exi
birPergunta&idPergunta={ $pergunta->idPergunta}', 500, 400);"></a-->
        <a href="javascript:alert('Nã
implementado');"></a>
    </td>
    <td><a
href="javascript:popup('Main.php?do=administradorAction&action=exi
birAlterarPerguntaPopUp&idPergunta={ $pergunta-
>idPergunta}', 500, 500, 1);"></a></td>
        <td><a href="javascript:confirmarRemover('{ $pergunta-
>pergunta}', 'Main.php?do=administradorAction&action=removePergunt
a&idPergunta={ $pergunta->idPergunta}');" title="Remover"><a/></td>
        <td>{ $pergunta->pergunta}</td>
    </tr>
{if $smarty.foreach.pergunta.index%2==0}
    <tr class="par">
{else}
    <tr class="impar">
{/if}
    <td colspan="5" align="left">

```

```

                {assign var="listaResposta" value=$pergunta-
>getLink('listaResposta')}
                <h4>Respostas({$listaResposta|@count})</h4>
                <ul id="listaResposta">
                    {foreach from=$listaResposta item=resposta
name=resposta}
                        <li>{$resposta->resposta}</li>
                    {/foreach}
                </ul>
            </td>
        </tr>
        {if $smarty.foreach.pergunta.index%2==0}
            <tr class="par">
        {else}
            <tr class="impar">
        {/if}
        <td colspan="5">
            Gráfico
            <script src="/conteudo/js/graphs2.js"
language="JavaScript" ></script>
            {literal}
            <script>
                graph = new BAR_GRAPH("hBar");
                graph.values = new Array();
            {/literal}

            {php}
                $pergunta = $this-
>get_template_vars('pergunta');
                $listaResposta = $this-
>get_template_vars('listaResposta');
                foreach ($listaResposta as $resposta) {
                    $somatorioPerguntaResposta =
PesquisaDAO::exibirSomatorioPerguntaResposta($pergunta->pesquisa,
$pergunta->idPergunta, $resposta->idResposta);
                    echo
"graph.values[graph.values.length]=".$somatorioPerguntaResposta.";
\n";
                }
            {/php}
            {literal}
            graph.graphBGColor = "white";
            graph.graphBorder = "1px solid green";
            graph.graphPadding = 10;
            graph.barColor = "#C0D0C0,#80D080";
            graph.barBGColor = "#D0F0D0";
            graph.labelColor = "yellow";
            graph.labelBGColor = "#60C060";

            graph.showValues = 1;
            document.write(graph.create());
        </script>

```



```

        <td><input type="checkbox" name="multipla" value="1">
Permitir múltipla escolha</td>
    </tr>
    <tr>
        <td><input type="checkbox" name="certa" value="1"
onclick="marcarCerta();" > Marcar respostas certas?</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><b>Respostas</b> - Digite as respostas:

        <table>
            <tbody id="tabelaResposta">
                <tr id="tr1">
                    <td>
                        <span name="checkRespostaCerta"
id="checkRespostaCerta"></span>
                        <textarea name='textoResposta[]'
class="w350h50"></textarea><a
href="javascript:removerResposta(1);" title="Remover"></a>
                    </td>
                </tr>
                <tr id="tr2">
                    <td>
                        <span name="checkRespostaCerta"
id="checkRespostaCerta"></span>
                        <textarea name='textoResposta[]'
class="w350h50"></textarea><a
href="javascript:removerResposta(2);" title="Remover"></a>
                    </td>
                </tr>
                <tr id="tr3">
                    <td>
                        <span name="checkRespostaCerta"
id="checkRespostaCerta"></span>
                        <textarea name='textoResposta[]'
class="w350h50"></textarea><a
href="javascript:removerResposta(3);" title="Remover"></a>
                    </td>
                </tr>
            </tbody>
        </table>

        <a
href="javascript:exibirInserirResposta($('formulario'),'listaRespo
sta');">+ resposta</a>
    </td>

```


listarFuncionario.php

```
{include file="layout/cabecalho.php"}

<h2>Lista de Funcionários</h2>

<table border="1">
<tr>
    <th>&nbsp;</th><th>&nbsp;</th><th>&nbsp;</th><th>Nome</th><th>
>Grupo(s)</th>
</tr>
{foreach from=$listaFuncionario item=funcionario name=funcionario}
    <tr>
        <td><a
href="Main.php?do=administradorAction&action=exibirFuncionario&idF
uncionario={$funcionario->idUsuario}"></a></td>
        <td><a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birAlterarFuncionarioPopUp&idFuncionario={$funcionario-
>idUsuario}',500,400);"></a></td>
        <td><a href="javascript:confirmarRemover('{funcionario-
>nome}','Main.php?do=administradorAction&action=removeFuncionario
&idFuncionario={$funcionario->idUsuario}');" title="Remover"><a/></td>
        <td>{funcionario->nome}</td>
        <td>
            {assign var="listaGrupo" value=$funcionario-
>getLink('listaGrupo')}
            {listaGrupo|@count}
        </td>
    </tr>
</foreach>
</table>

<br>
<a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birInserirFuncionarioPopUp',400,200,1);">Inserir funcionário</a>

{include file="layout/rodape.php"}
```

listarGrupo.php

```
{include file="layout/cabecalho.php"}

<h2>Lista de Grupos</h2>

<table border="1">
<tr>
```

```

        <th>&nbsp;</th><th>&nbsp;</th><th>&nbsp;</th><th>Nome</th><th>
>Número funcionários</th>
</tr>
{foreach from=$listaGrupo item=grupo name=grupo}
    <tr>
        <td><a
href="Main.php?do=administradorAction&action=exibirGrupo&idGrupo={
$grupo->idGrupo}"></a></td>
        <td><a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birAlterarGrupoPopUp&idGrupo={$grupo->idGrupo}',500,400);"></a></td>
        <td><a href="javascript:confirmarRemover('{ $grupo-
>nome}', 'Main.php?do=administradorAction&action=removerGrupo&idGru
po={$grupo->idGrupo}');" title="Remover"><a/></td>
        <td>{$grupo->nome}</td>
        <td>
            {assign var="listaUsuario" value=$grupo-
>getLink('listaUsuario')}
            {$listaUsuario|@count}
        </td>
    </tr>
</foreach>
</table>

<br>
<a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birInserirGrupoPopUp',400,200,1);">Inserir grupo</a>

{include file="layout/rodape.php"}

listarPesquisa.php
{include file="layout/cabecalho.php"}

<h2>Lista de Pesquisas</h2>

<table border="1">
<tr>
    <!--<th>&nbsp;</th><th>&nbsp;</th><th>&nbsp;</th>--
><th>&nbsp;</th><th>&nbsp;</th><th>&nbsp;</th><th>&nbsp;</th><th>N
ome</th><th>Número Perguntas</th>
</tr>
{foreach from=$listaPesquisa item=pesquisa name=pesquisa}
    <tr>
        <td><a
href="Main.php?do=administradorAction&action=exibirPesquisa&idPesq
uisa={$pesquisa->idPesquisa}"></a></td>

```

```

        <td><a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birAlterarPesquisaPopUp&idPesquisa={$pesquisa-
>idPesquisa}',500,400);"></a></td>
        <td><a href="javascript:confirmarRemover('{Pesquisa-
>nome}','Main.php?do=administradorAction&action=removerPesquisa&id
Pesquisa={$Pesquisa->idPesquisa}');" title="Remover"><a/></td>
        <td><a
href="Main.php?do=administradorAction&action=exibirRelatorioPesqui
sa&idPesquisa={$Pesquisa->idPesquisa}">Relatório</a></td>
        <!--<td><a
href="Main.php?do=administradorAction&action=gerarPesquisa&idPesqu
isa={$Pesquisa->idPesquisa}">Gerar Pesquisa</a></td>
        <td><a
href="Main.php?do=administradorAction&action=gerarClassePesquisa&i
dPesquisa={$Pesquisa->idPesquisa}">Gerar Classe Pesquisa</a></td>
        <td><a
href="Main.php?do=administradorAction&action=rodarAnt&idPesquisa={
Pesquisa->idPesquisa}">Rodar Ant</a></td>-->
        <td>{Pesquisa->nome}</td>
        <td>
                {assign var="listaPergunta" value=$pesquisa-
>getLink('listaPergunta')}
                {$listaPergunta|@count}
        </td>
    </tr>
</foreach>
</table>

<br>
<a
href="javascript:popUp('Main.php?do=administradorAction&action=exi
birInserirPesquisaPopUp',500,500,1);">Inserir Pesquisa</a>

{include file="layout/rodape.php"}

```

Anexo A - Código Fonte Sistema J2ME

EnviarPesquisaAction.java

```
package action;

import java.io.InputStream;
import java.io.OutputStream;
import java.util.Vector;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

import view.ExibirIndexPesquisa;
import view.Main;
import dao.PesquisaDAO;

public class EnviarPesquisaAction extends Thread {

    private Main main;
    private ExibirIndexPesquisa exibirIndexPesquisa;
    private Vector listaPesquisa;
    private int idPesquisa;
    private int idUsuario;

    public EnviarPesquisaAction(Main main, ExibirIndexPesquisa
exibirIndexPesquisa, Vector listaPesquisa, int idPesquisa, int idUsuario)
    {
        this.main = main;
        this.exibirIndexPesquisa = exibirIndexPesquisa;
        this.listaPesquisa = listaPesquisa;
        this.idPesquisa = idPesquisa;
        this.idUsuario = idUsuario;
    }

    public void run() {
        try {
            HttpConnection con =
(HttpConnection)Connector.open("http://localhost/webForm/Main.php?do=meAc
tion&action=inserirResultadoPesquisa");
            con.setRequestMethod(HttpConnection.POST);
            con.setRequestProperty("Content-Type", "application/x-
www-form-urlencoded");
            con.setRequestProperty("Connection", "close");
            //con.setRequestProperty("Accept", "application/octet-
stream");

            byte post[] = this.gerarPost();
            con.setRequestProperty("Content-Length",
Integer.toString(post.length));
            OutputStream os = con.openOutputStream();
            os.write(post);
            os.close();
        }
    }
}
```

```

        int rc = con.getResponseCode();
        if (rc != HttpURLConnection.HTTP_OK) {
            System.out.println("erro, nada enviado");
        }
        else {
            InputStream is = con.openInputStream();
            int tamanhoResposta = (int)con.getLength();
            byte arrayResposta[] = new byte[tamanhoResposta];
            is.read(arrayResposta);
            String resposta = new String(arrayResposta);
            System.out.println(resposta);
            if (resposta.equals("tudoCerto")) {
                System.out.println("alterar");
                PesquisaDAO pesquisaDAO = new
PesquisaDAO(this.idPesquisa,this.idUsuario);
                pesquisaDAO.alterarPesquisaEnviada();
                this.exibirIndexPesquisa.montarLayout();
            }
            else {
                System.out.println("algo errado");
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public byte[] gerarPost() {
    String post = "&id="+this.idPesquisa+"&u="+this.idUsuario;
    String temp1;

    for (int i=0; i<this.listaPesquisa.size(); i++) {
        temp1 = listaPesquisa.elementAt(i).toString();
        //retira o caracter de controle de envio da pesquisa
        temp1 = temp1.substring(0, temp1.length()-2);
        post += "&p[]" +temp1;
        //System.out.println(lista.elementAt(i));
    }
    System.out.println(post);
    return post.getBytes();
}
}

```

ExibirIndexPesquisaAction.java

```
package action;

import java.io.InputStream;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

import util.Constantes;
import util.StringUtil;
import view.ExibirIndexPesquisa;
import view.Main;
import bean.Pergunta;
import bean.Pesquisa;
import bean.Resposta;

public class ExibirIndexPesquisaAction extends Thread {

    private Main main;
    private Pesquisa pesquisa;
    private int idUsuario;

    public ExibirIndexPesquisaAction(Main main, Pesquisa pesquisa, int
idUsuario) {
        this.main = main;
        this.pesquisa = pesquisa;
        this.idUsuario = idUsuario;
    }

    public void run() {
        try {
            HttpConnection con =
(HttpConnection)Connector.open("http://localhost/webForm/Main.php?do=meAc
tion&action=exibirPesquisa&idPesquisa="+pesquisa.getIdPesquisa());
            con.setRequestMethod(HttpConnection.GET);
            con.setRequestProperty("Content-Type", "application/x-
www-form-urlencoded");
            con.setRequestProperty("Connection", "close");
            //con.setRequestProperty("Accept", "application/octet-
stream");

            int rc = con.getResponseCode();
            if (rc != HttpConnection.HTTP_OK) {
                System.out.println("erro, nada enviado");
            }
            else {
                InputStream is = con.openInputStream();
                int tamanhoResposta = (int)con.getLength();
                byte arrayResposta[] = new byte[tamanhoResposta];
                is.read(arrayResposta);
                String resposta = new String(arrayResposta);
                //System.out.println(resposta);

                String[] string1, string2, string3;
```

```

        string1 = StringUtil.split(resposta,
Constantes.separador3);//lista perguntas/respostas
        Pergunta[] listaPergunta = new
Pergunta[string1.length];
        Resposta[] listaResposta;
        Pergunta perguntal;
        Resposta respostal;
        for (int i=0; i<string1.length; i++) {
Constantes.separador2);//pergunta/respostas
            string2 = StringUtil.split(string1[i],
1);//a primeira posicao eh a pergunta
            for (int j=0; j<string2.length; j++) {
Constantes.separador1);//pergunta ou resposta
                if (j==0) {//pergunta

                    perguntal.setIdPergunta(Integer.parseInt(string3[0]));

                    perguntal.setMultipla(Integer.parseInt(string3[1]));

                    perguntal.setPergunta(string3[2]);
                        listaPergunta[i] = perguntal;
                    }
                    else {//respostas
                        respostal = new Resposta();

                        respostal.setIdResposta(Integer.parseInt(string3[0]));

                        respostal.setCerta(Integer.parseInt(string3[1]));

                        respostal.setResposta(string3[2]);
                            listaResposta[j-1] = respostal;
                        }
                    }
                    perguntal.setListaResposta(listaResposta);
                }
                this.pesquisa.setListaPergunta(listaPergunta);

                /*for (int i=0; i<listaPergunta.length; i++) {

                    System.out.println(listaPergunta[i].getPergunta());
                        listaResposta =
listaPergunta[i].getListaResposta();
                            for (int j=0; j<listaResposta.length; j++) {

                                System.out.println(listaResposta[j].getResposta());
                                    }
                                System.out.println("-");
                            }*/

                    /*ExibirIndexPesquisa exibirIndexPesquisa = new
ExibirIndexPesquisa(this.main, listaPergunta);
                    exibirIndexPesquisa.montarLayout();*/

```

```

        ExibirIndexPesquisa exibirIndexPesquisa = new
ExibirIndexPesquisa(this.main, pesquisa, this.idUsuario);
        exibirIndexPesquisa.montarLayout();
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

ListarPesquisaAction.java

```

package action;

import java.io.InputStream;
import java.util.Vector;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

import util.Constantes;
import util.StringUtil;
import view.ListarPesquisa;
import view.Main;
import bean.Pesquisa;

public class ListarPesquisaAction extends Thread {

    private Main main;
    private int idUsuario;

    public ListarPesquisaAction(Main main, int idUsuario) {
        this.main = main;
        this.idUsuario = idUsuario;
    }

    public void run() {
        try {
            HttpConnection con =
(HttpConnection)Connector.open("http://localhost/webForm/Main.php?do=meAc
tion&action=listarPesquisa&idUsuario="+idUsuario);
            con.setRequestMethod(HttpConnection.GET);
            con.setRequestProperty("Content-Type", "application/x-
www-form-urlencoded");
            con.setRequestProperty("Connection", "close");
            //con.setRequestProperty("Accept", "application/octet-
stream");

            int rc = con.getResponseCode();
            if (rc != HttpConnection.HTTP_OK) {
                System.out.println("erro, nada enviado");
            }
        }
    }
}

```

```

    }
    else {
        InputStream is = con.openInputStream();
        int tamanhoResposta = (int)con.getLength();
        byte arrayResposta[] = new byte[tamanhoResposta];
        is.read(arrayResposta);
        String resposta = new String(arrayResposta);
        System.out.println(resposta);

        String[] string1 = StringUtil.split(resposta,
Constantes.separador2);
        String[] string2;
        Vector listaPesquisa = new Vector();
        Pesquisa pesquisal;
        for (int i=0; i<string1.length; i++) {
StringUtil.split(string1[i],Constantes.separador1);
                pesquisal = new Pesquisa();

                pesquisal.setIdPesquisa(Integer.parseInt(string2[0]));
                pesquisal.setNome(string2[1]);
                listaPesquisa.addElement(pesquisal);

        }
        ListarPesquisa listarPesquisa = new
ListarPesquisa(this.main, listaPesquisa);
        listarPesquisa.montarLayout();

        /*ParserListaPesquisaXML parser = new
ParserListaPesquisaXML();
        Vector listaPesquisa = parser.parser(is);
        ListarPesquisa listarPesquisa = new
ListarPesquisa(this.main, this.display, listaPesquisa);
        listarPesquisa.montarLayout();*/
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

LoginAction.java

```

package action;

import java.io.InputStream;
import java.io.OutputStream;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

import util.Constantes;

```

```

import view.Login;
import view.Main;

public class LoginAction extends Thread {

    private Main main;
    private Login login;
    private String usuario;
    private String senha;

    public LoginAction(Main main, Login login, String usuario, String
senha) {
        this.main = main;
        this.login = login;
        this.usuario = usuario;
        this.senha = senha;
    }

    public void run() {
        try {
            HttpURLConnection con =
(HttpURLConnection)Connector.open("http://localhost/webForm/Main.php?do=meAc
tion&action=login");
            con.setRequestMethod(HttpURLConnection.POST);
            con.setRequestProperty("Content-Type", "application/x-
www-form-urlencoded");
            con.setRequestProperty("Connection", "close");
            //con.setRequestProperty("Accept", "application/octet-
stream");

            String string1 =
"&usuario="+this.usuario+"&senha="+this.senha;
            byte post[] = string1.getBytes();
            con.setRequestProperty("Content-Length",
Integer.toString(post.length));
            OutputStream os = con.openOutputStream();
            os.write(post);
            os.close();

            int rc = con.getResponseCode();
            if (rc != HttpURLConnection.HTTP_OK) {
                System.out.println("erro, nada enviado");
            }
            else {
                InputStream is = con.openInputStream();
                int tamanhoResposta = (int)con.getLength();
                byte arrayResposta[] = new byte[tamanhoResposta];
                is.read(arrayResposta);
                String resposta = new String(arrayResposta);
                System.out.println(resposta);
                if (resposta.startsWith(Constants.separador1)) {
                    int idUsuario =
Integer.parseInt(resposta.substring(1, resposta.length()));
                    ListarPesquisaAction listarPesquisaAction =
new ListarPesquisaAction(this.main,idUsuario);
                    listarPesquisaAction.start();
                }
            }
        }
    }
}

```

```

        }
        else {
            this.login.usuarioInvalido();
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

/*public byte[] gerarPost() {
    String post = "&id="+this.principalView.getIdPesquisa();
    String temp1, temp2[], temp3;
    int dataInicio, dataFim;
    for (int i=0; i<lista.size(); i++) {
        temp1 = lista.elementAt(i).toString();
        if (i%(this.principalView.getNroPerguntas()+1)==0)
//em cada comeco de pesquisa
            dataInicio = Integer.parseInt(temp1.substring(0,
10));
            dataFim = Integer.parseInt(temp1.substring(10,
20));
            post += "&i[]="+dataInicio+"&f[]="+dataFim;
        }
        else {
            //ex 11_1_0
            //pega 11_1 para analisar ultimo dado eh pesquisa
se foi enviada
            temp3 = temp1.substring(0, temp1.length()-2);
            temp2 = this.split(temp3, "_");
            for (int j=0; j<temp2.length; j++) {
                if (j==0)
                    post += "&p[]="+temp2[j];
                else
                    post += "&r[]="+temp2[j];
            }
        }
        System.out.println(lista.elementAt(i));
    }
    System.out.println(post);
    return post.getBytes();
}
public byte[] gerarPost() {
    String post = "&id="+this.principalView.getIdPesquisa();
    String temp1, temp2;
    int dataInicio, dataFim;
    for (int i=0; i<lista.size(); i++) {
        temp1 = lista.elementAt(i).toString();
        if (i%(this.principalView.getNroPerguntas()+1)==0)
//em cada comeco de pesquisa

```

```

        dataInicio = Integer.parseInt(temp1.substring(0,
10));
        dataFim = Integer.parseInt(temp1.substring(10,
20));
        post += "&i[]="+dataInicio+"&f[]="+dataFim;
    }
    else {
        //ex 11_1_0
        //11_1 retira ultimo dado, pois esse eh para
controle interno do j2me
        temp2 = temp1.substring(0, temp1.length()-2);
        post += "&p[]="+temp2;
    }
    //System.out.println(lista.elementAt(i));

}
System.out.println(post);
return post.getBytes();
}*/

}

```

PesquisaDAO.java

```

package dao;

import java.util.Vector;

import util.Constantes;
import util.RecordStoreUtil;
import util.StringUtil;
import bean.Record;

public class PesquisaDAO {

    RecordStoreUtil rs;

    public PesquisaDAO(int idPesquisa, int idUsuario) {
        rs = new RecordStoreUtil(idPesquisa+"_"+idUsuario);
    }

    public void inserirPesquisa(String pesquisa) {
        this.rs.add(pesquisa);
    }

    public void removerTudo() {
        this.rs.deleteAll();
    }

    public int getNumeroPesquisasEfetuadas() {
        int int1 = 0;
        try {

```

```

        int1 = rs.nroRecords();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return int1;
}
public int getNumeroPesquisasEnviadas() {
    int int1 = 0;
    Vector listaPesquisa = this.rs.listarConteudo();
    String string1, string3;
    String[] string2;
    Record record1;
    for (int i=0; i<listaPesquisa.size(); i++) {
        record1 = (Record)listaPesquisa.elementAt(i);
        string1 = record1.getConteudo();
        string2 = StringUtil.split(string1,
Constantes.separador2);
        string3 = string2[string2.length-1]; //ultimo registro
eh o controle de envio
        if (string3.equals("1"))
            int1++;
    }
    return int1;
}
public Vector listarPesquisaNaoEnviada() {
    Vector listaPesquisa = this.rs.listarConteudo();
    Vector listaPesquisaNaoEnviada = new Vector();
    String string1, string3;
    String[] string2;
    Record record1;
    for (int i=0; i<listaPesquisa.size(); i++) {
        record1 = (Record)listaPesquisa.elementAt(i);
        string1 = record1.getConteudo();
        string2 = StringUtil.split(string1,
Constantes.separador2);
        string3 = string2[string2.length-1]; //ultimo registro
eh o controle de envio
        if (string3.equals("0"))
            listaPesquisaNaoEnviada.addElement(string1);
    }
    return listaPesquisaNaoEnviada;
}
public void alterarPesquisaEnviada() {
    Vector listaPesquisa = this.rs.listarConteudo();
    String string1, string3;
    String[] string2;
    Record record1;
    for (int i=0; i<listaPesquisa.size(); i++) {
        record1 = (Record)listaPesquisa.elementAt(i);
        string1 = record1.getConteudo();
        string2 = StringUtil.split(string1,
Constantes.separador2);
        string3 = string2[string2.length-1]; //ultimo registro
eh o controle de envio
        if (string3.equals("0")) {
            string1 = string1.substring(0, string1.length()-
1);

```

```

        string1 += "1";
        this.rs.update(record1.getIdRecord(), string1);
    }
}
}
}

```

Alerta.java

```

package util;

import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Image;

public class Alerta {

    public void exhibirAlerta(Display display, String titulo, String
mensagem, int tipoAlerta) {
        Image icone = null;
        /*try {
            switch (tipoAlerta) {
                case (Constantes.AlertaConfirmacao) :
                    icone =
Image.createImage("/confirmacao.png");
                    break;
                case (Constantes.AlertaInformacao) :
                    icone = Image.createImage("/aleta.png");
                    break;
                case (Constantes.AlertaErro) :
                    icone = Image.createImage("/erro.png");
            }
        }
        catch(IOException e) {
            e.printStackTrace();
            //A Imagem nao foi encontrada, o objeto Image ser null
        }*/
        Alert alert = new Alert(titulo, "      "+mensagem, icone,
null);
        alert.setTimeout(Alert.FOREVER);
        display.setCurrent(alert);
    }
}
}

```

RecordStoreUtil.java

```

package util;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.util.Vector;

```

```

import javax.microedition.rms.InvalidRecordIDException;
import javax.microedition.rms.RecordEnumeration;
import javax.microedition.rms.RecordStore;
import javax.microedition.rms.RecordStoreException;
import javax.microedition.rms.RecordStoreFullException;
import javax.microedition.rms.RecordStoreNotFoundException;
import javax.microedition.rms.RecordStoreNotOpenException;

import bean.Record;

public class RecordStoreUtil {

    protected String RS_NAME;
    protected RecordStore rs = null;

    public RecordStoreUtil(String RS_NAME) {
        this.RS_NAME = RS_NAME;
    }

    public void open() {
        try {
            rs = RecordStore.openRecordStore(RS_NAME, true);
        }
        catch (RecordStoreNotFoundException e) {
            System.out.println("-- RecordStore inexistente");
        }
        catch (RecordStoreException e) {
            System.out.println("-- Outro erro");
        }
    }

    public void close() {
        try {
            rs.closeRecordStore();
        }
        catch( RecordStoreNotOpenException e ){
            System.out.println("-- O Record Store esta fechado");
        }
        catch( RecordStoreException e ){
            System.out.println("-- Outro erro");
        }
    }

    public void listarRecordStores() {
        //this.open();
        String lista[] = RecordStore.listRecordStores();
        for (int i = 0; i < lista.length; i++)
            System.out.println(lista[i]);
        //this.close();
    }

    public int nroRecords() throws RecordStoreNotOpenException {
        this.open();
        int temp = rs.getNumRecords();
    }
}

```

```

        this.close();
        return temp;
    }

    public String getRecord(int id) {
        this.open();
        String toReturn = "";
        try {
            int recordSize = rs.getRecordSize(id);
            byte[] data = new byte[recordSize];
            ByteArrayInputStream bais = new
ByteArrayInputStream(data);
            DataInputStream dis = new DataInputStream(bais);
            int numBytes = rs.getRecord(id, data, 0);
            toReturn = dis.readUTF();
            bais.reset();
            bais.close();
            dis.close();
        }
        catch (IOException e) {
            this.close();
            System.out.println("-- Erro de IO");
        }
        catch (ArrayIndexOutOfBoundsException e) {
            this.close();
            System.out.println("-- Registro muito grande");
        }
        catch (InvalidRecordIDException e) {
            this.close();
            System.out.println("-- ID inexistente");
        }
        catch (RecordStoreNotOpenException e) {
            this.close();
            System.out.println("-- O Record Store esta fechado");
        }
        catch (RecordStoreException e) {
            this.close();
            System.out.println("-- Outro erro");
        }
        this.close();
        return toReturn;
    }

    public void add(String nome) {
        this.open();
        try {
            ByteArrayOutputStream baos = new
ByteArrayOutputStream();
            DataOutputStream dos = new DataOutputStream(baos);
            dos.writeUTF(nome);
            dos.flush();
            byte[] data = baos.toByteArray();
            int id = rs.addRecord(data, 0, data.length);
            System.out.println("id"+id);
            baos.close();
            dos.close();
        }
        catch (IOException e) {

```

```

        this.close();
        System.out.println("-- Erro de IO");
    }
    catch (RecordStoreFullException e) {
        this.close();
        System.out.println("-- Não existe espaço disponível");
    }
    catch (RecordStoreNotOpenException e) {
        this.close();
        System.out.println("-- O Record Store esta fechado");
    }
    catch (RecordStoreException e) {
        this.close();
        System.out.println("-- Outro erro");
    }
    this.close();
}

public void update(int id, String nome) {
    this.open();
    try {
        ByteArrayOutputStream baos = new
ByteArrayOutputStream();
        DataOutputStream dos = new DataOutputStream(baos);
        dos.writeUTF(nome);
        dos.flush();
        byte[] data = baos.toByteArray();
        rs.setRecord(id, data, 0, data.length);
    }
    catch (IOException e) {
        this.close();
        System.out.println("-- Erro de IO");
    }
    catch (ArrayIndexOutOfBoundsException e) {
        this.close();
        System.out.println("-- Registro muito grande");
    }
    catch (InvalidRecordIDException e) {
        this.close();
        System.out.println("-- ID inexistente");
    }
    catch (RecordStoreNotOpenException e) {
        this.close();
        System.out.println("-- O Record Store esta fechado");
    }
    catch (RecordStoreException e) {
        this.close();
        System.out.println("-- Outro erro");
    }
    this.close();
}

public void delete(int id) {
    this.open();
    try {
        rs.deleteRecord(id);
    }
    catch (RecordStoreFullException e) {

```

```

        this.close();
        System.out.println("-- Não existe espaço disponível");
    }
    catch (RecordStoreNotOpenException e) {
        this.close();
        System.out.println("-- O Record Store esta fechado");
    }
    catch (RecordStoreException e) {
        this.close();
        System.out.println("-- Outro erro");
    }
    this.close();
}
public void deleteAll() {
    try {
        RecordStore.deleteRecordStore(this.RS_NAME);
    }
    catch (RecordStoreNotFoundException e) {
        System.out.println("-- RecordStore inexistente");
    }

    catch (RecordStoreException e) {
        System.out.println("-- Outro erro");
    }
}
}

```

```

/*public int getNextRecordId() {
    this.open();
    int toReturn = 0;
    try {
        toReturn = rs.getNextRecordID();
    }
    catch (RecordStoreFullException e) {
        this.close();
        System.out.println("-- Não existe espaço disponível");
    }
    catch (RecordStoreNotOpenException e) {
        this.close();
        System.out.println("-- O Record Store esta fechado");
    }
    catch (RecordStoreException e) {
        this.close();
        System.out.println("-- Outro erro");
    }
    this.close();
    return toReturn;
}*/

```

```

/*public void listarConteudo() {
    this.open();

```

```

        try {
            for (int i = 1; i <= rs.getNumRecords(); i++) {
                System.out.println("-- getRecord = " +
getRecord(i));
            }
        }
        catch (RecordStoreNotOpenException ex) {
            this.close();
            System.out.println("-- O Record Store esta fechado");
        }
        this.close();
    }*/

    public Vector listarConteudo() {
        Vector lista = new Vector();
        this.open();
        try {
            RecordEnumeration re = rs.enumerateRecords(null, null,
false);

            Record record1;
            while (re.hasNextElement()) {
                record1 = new Record();
                record1.setIdRecord(re.nextRecordId());

                record1.setConteudo(getRecord(record1.getIdRecord()));
                lista.addElement(record1);
            }
        }
        catch (RecordStoreNotOpenException ex) {
            this.close();
            System.out.println("-- O Record Store esta fechado");
        }
        catch (RecordStoreException ex) {
            this.close();
            System.out.println("-- id");
        }
        this.close();

        return lista;
    }

    /*public void printEnum() {
        this.open();
        try {
            RecordEnumeration re = rs.enumerateRecords(null, null,
false);

            while (re.hasNextElement()) {
                System.out.println("-- getRecord = " +
getRecord(re.nextRecordId()));
            }
        }
        catch (InvalidRecordIDException ex) {
            this.close();
            System.out.println("-- ID inexistente");
        }
        catch (RecordStoreNotOpenException ex) {
            this.close();

```

```

        System.out.println("-- O Record Store esta fechado");
    }
    this.close();
}*/

/*public void printEnumOrder() {
    this.open();
    try {
        ComparatorString sc = new ComparatorString();
        RecordEnumeration re = rs.enumerateRecords(null, sc,
false);
        while (re.hasNextElement()) {
            System.out.println("-- getRecord = " +
getRecord(re.nextRecordId()));
        }
    }
    catch (InvalidRecordIDException ex) {
        this.close();
        System.out.println("-- ID inexistente");
    }
    catch (RecordStoreNotOpenException ex) {
        this.close();
        System.out.println("-- O Record Store esta fechado");
    }
    this.close();
}*/

/*public class ComparatorString implements RecordComparator {

    public int compare(byte[] rec1, byte[] rec2) {
        String s1 = new String(rec1);
        String s2 = new String(rec2);
        int comparation = s1.compareTo(s2);
        if (comparation == 0)
            return RecordComparator.EQUIVALENT;
        else
            if (comparation > 0)
                return RecordComparator.PRECEDES;
            else
                return RecordComparator.FOLLOWS;
    }
}*/

```

```

}

```

CancelaPesquisa.java

```

package view;

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

```

```

import javax.microedition.lcdui.StringItem;

public class CancelaPesquisa implements CommandListener {

    private Main main;
    private ExibirPesquisa exibirPesquisa;
    private Form fCancelaPesquisa;
    private Command cSim, cNao;

    public CancelaPesquisa(Main main, ExibirPesquisa exibirPesquisa) {
        this.main = main;
        this.exibirPesquisa = exibirPesquisa;
    }

    public void montarLayout() {
        fCancelaPesquisa = new Form("");

        //Image icone = null;
        /*try {
            icone = Image.createImage("/imagem/alerta.gif");
        }
        catch(IOException e) {
            e.printStackTrace();
            //A Imagem nao foi encontrada, o objeto Image será null
        }*/

        StringItem sil = new StringItem("Deseja descartar essa
resposta e voltar a tela inicial?", "");

        //fCancelaPesquisa.append(icone);
        fCancelaPesquisa.append(sil);

        cSim = new Command("Sim", Command.BACK, 0);
        cNao = new Command("Não", Command.OK, 0);

        fCancelaPesquisa.addCommand(cNao);
        fCancelaPesquisa.addCommand(cSim);

        fCancelaPesquisa.setCommandListener(this);
        this.main.getDisplay().setCurrent(fCancelaPesquisa);
    }

    public void commandAction(Command cmd, Displayable disp) {
        if (cmd == cSim) {

            this.main.getDisplay().setCurrent(this.exibirPesquisa.getExibirIndexPesquisa().getFExibirIndexPesquisa());
        }
        if (cmd == cNao) {

            this.main.getDisplay().setCurrent(this.exibirPesquisa.getFExibirPesquisa());
        }
    }
}

```

```
}
```

ExibirIndexPesquisa.java

```
package view;

import java.util.Date;
import java.util.TimeZone;
import java.util.Vector;

import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.DateField;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Item;
import javax.microedition.lcdui.StringItem;

import action.EnviaPesquisaAction;
import bean.Pesquisa;
import dao.PesquisaDAO;

public class ExibirIndexPesquisa implements CommandListener {

    private Main main;
    private PesquisaDAO pesquisaDAO;

    private Form fExibirIndexPesquisa;
    private Command cSair, cPesquisa, cEnviar, cApagarTudo,
aApagarEnviadas, cApagarNaoEnviadas;

    private Pesquisa pesquisa;
    private int idUsuario;

    public ExibirIndexPesquisa(Main main, Pesquisa pesquisa, int
idUsuario) {
        this.main = main;
        this.pesquisa = pesquisa;
        this.idUsuario = idUsuario;
        this.pesquisaDAO = new
PesquisaDAO(this.pesquisa.getIdPesquisa(),this.idUsuario);
    }

    public void montarLayout() {

        StringItem siNomePesquisa;
        StringItem siPesquisasEfetuadas = null;
        StringItem siPesquisasEnviadas = null;
        StringItem siNumeroPerguntas;
        DateField dateField;

        fExibirIndexPesquisa = new Form("Tela Principal");

        siNomePesquisa = new StringItem(this.pesquisa.getNome(),"");
```

```

        siNumeroPerguntas = new StringItem("Números de perguntas:
",String.valueOf(this.pesquisa.getListaPergunta().length));

        TimeZone timezone = TimeZone.getTimeZone("GMT-3");
        dateField = new DateField(null, DateField.DATE_TIME,
timezone);

        int nroPesquisasEfetuadas =
this.pesquisaDAO.getNumeroPesquisasEfetuadas();
        int nroPesquisasEnviadas =
this.pesquisaDAO.getNumeroPesquisasEnviadas();
        try {
            siPesquisasEfetuadas = new StringItem("Pesquisas
efetuadas: ", String.valueOf(nroPesquisasEfetuadas), Item.PLAIN);
            siPesquisasEnviadas = new StringItem("Pesquisas
enviadas: ", String.valueOf(nroPesquisasEnviadas), Item.PLAIN);

        } catch (Exception e) {
            e.printStackTrace();
        }
        Date data = new Date();
        dateField.setDate(data);

        /*TimeZone timezone = TimeZone.getTimeZone("GMT-3");
Calendar c = Calendar.getInstance(timezone);
dataString = new StringItem(" ",c.getTime().toString());*/

        fExibirIndexPesquisa.append(siNomePesquisa);
        fExibirIndexPesquisa.append(siNumeroPerguntas);
        fExibirIndexPesquisa.append(siPesquisasEfetuadas);
        fExibirIndexPesquisa.append(siPesquisasEnviadas);
        fExibirIndexPesquisa.append(dateField);
        //fPrincipal.append(siData);

        cPesquisa = new Command("Iniciar Pesquisa", Command.OK, 0);

        cEnviar = new Command("Enviar Pesquisas", Command.OK, 0);
        cApagarTudo = new Command("Apagar tudo", Command.OK, 0);
        cSair = new Command("Sair", Command.EXIT, 0);

        fExibirIndexPesquisa.addCommand(cPesquisa);
        fExibirIndexPesquisa.addCommand(cEnviar);
        fExibirIndexPesquisa.addCommand(cApagarTudo);
        fExibirIndexPesquisa.addCommand(cSair);

        fExibirIndexPesquisa.setCommandListener(this);
        this.main.getDisplay().setCurrent(fExibirIndexPesquisa);
    }

    public void commandAction(Command cmd, Displayable disp) {
        if (cmd == cSair) {
            this.main.destroyApp(false);
            this.main.notifyDestroyed();
        }
        if (cmd == cPesquisa) {
            ExibirPesquisa exibirPesquisa = new
ExibirPesquisa(this.main, this, this.pesquisa);

```

```

        exibirPesquisa.montarLayout();
    }
    if (cmd == cEnviar) {
        Vector listaPesquisa =
this.pesquisaDAO.listarPesquisaNaoEnviada();
        EnviarPesquisaAction enviarPesquisaAction = new
EnviarPesquisaAction(this.main, this, listaPesquisa,
this.pesquisa.getIdPesquisa(), this.idUsuario);
        enviarPesquisaAction.start();
        //pesquisaView.enviarPesquisa();
    }
    if (cmd == cApagarTudo) {
        this.pesquisaDAO.removeTudo();
        this.montarLayout();
    }
}

public Form getFExibirIndexPesquisa() {
    return this.fExibirIndexPesquisa;
}
public PesquisaDAO getPesquisaDAO() {
    return this.pesquisaDAO;
}
}

```

ExibirPesquisa.java

```

package view;

import javax.microedition.lcdui.Choice;
import javax.microedition.lcdui.ChoiceGroup;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

import util.Alerta;
import util.Constantes;
import bean.Pergunta;
import bean.Pesquisa;
import bean.Resposta;

public class ExibirPesquisa implements CommandListener {

    private Main main;
    //private PesquisaDAO pesquisaDAO;
    private Pesquisa pesquisa;

    private Form fExibirPesquisa;
    private Command cSalvar, cVoltar;
    private ChoiceGroup[] choiceGroup;

    private ExibirIndexPesquisa exibirIndexPesquisa;
    private long dataInicio, dataFim;

```

```

    public ExibirPesquisa(Main main, ExibirIndexPesquisa
exibirIndexPesquisa, Pesquisa pesquisa) {
        this.main = main;
        this.exibirIndexPesquisa = exibirIndexPesquisa;
        this.pesquisa = pesquisa;
        //this.pesquisaDAO = new PesquisaDAO();
    }

    public void montarLayout() {
        fExibirPesquisa = new Form(pesquisa.getNome());

        Pergunta[] listaPergunta = pesquisa.getListaPergunta();
        choiceGroup = new ChoiceGroup[listaPergunta.length];

        String[] respostas;
        Resposta[] listaResposta;
        Pergunta perguntal;
        for (int i=0; i<listaPergunta.length; i++) {
            perguntal = listaPergunta[i];
            listaResposta = perguntal.getListaResposta();
            respostas = new String[listaResposta.length];

            for (int j=0; j<listaResposta.length; j++)
                respostas[j] = listaResposta[j].getResposta();

            choiceGroup[i] = new
ChoiceGroup(perguntal.getPergunta(),
perguntal.getMultipla()==1?Choice.MULTIPLE:Choice.EXCLUSIVE, respostas,
null);
            fExibirPesquisa.append(choiceGroup[i]);
        }

        cVoltar = new Command("Voltar", Command.BACK, 0);
        fExibirPesquisa.addCommand(cVoltar);

        cSalvar = new Command("Salvar", Command.OK, 0);
        fExibirPesquisa.addCommand(cSalvar);

        fExibirPesquisa.setCommandListener(this);

        this.dataInicio = System.currentTimeMillis()/1000;

        this.main.getDisplay().setCurrent(fExibirPesquisa);
    }

    public void commandAction(Command cmd, Displayable disp) {
        if (cmd == cSalvar) {
            //validacao
            if (this.validarPesquisa()) {
                this.inserirPesquisa();
                this.fExibirPesquisa.removeCommand(cSalvar);
                Alerta alerta = new Alerta();
                alerta.exibirAlerta(this.main.getDisplay(), "",
"Pesquisa salva com sucesso", Constantes.AlertaConfirmacao);
                this.dataInicio = 0;
            }
            else {

```

```

        Alerta alerta = new Alerta();
        alerta.exibirAlerta(this.main.getDisplay(), "",
"Responda todas as perguntas", Constantes.AlertaErro);
    }
}
if (cmd == cVoltar) {
    //validacao
    if (dataInicio != 0) {
        CancelaPesquisa cancelaPesquisa = new
CancelaPesquisa(this.main, this);
        cancelaPesquisa.montarLayout();
    }
    else {
        //ExibirIndexPesquisa exibirIndexPesquisa = new
ExibirIndexPesquisa(this.main, this.exibirIndexPesquisa.gpesquisa);
        exibirIndexPesquisa.montarLayout();
    }
}

}

public boolean validarPesquisa() {
    boolean selecionou;
    boolean bool1[];
    for (int i=0; i < this.pesquisa.getListaPergunta().length;
i++) {
        bool1 = new boolean[choiceGroup[i].size()];
        choiceGroup[i].getSelectedFlags(bool1);

        selecionou = false;
        for (int j=0; j < bool1.length; j++)
            if (bool1[j]==true) {
                selecionou = true;
                break;
            }

        if (!selecionou)
            return false;
    }

    return true;
}
public void inserirPesquisa() {
    String string1, string2;

    this.dataFim = System.currentTimeMillis()/1000;
    string1 =
this.dataInicio+Constantes.separador1+this.dataFim+Constantes.separador2;
//1

    boolean bool1[];
    Pergunta[] listaPergunta = this.pesquisa.getListaPergunta();
    Pergunta perguntal;
    Resposta[] listaResposta;
    for (int i=0; i < listaPergunta.length; i++) {
        perguntal = listaPergunta[i];

```

```

        listaResposta = pergunta1.getListaResposta();

        string2 =
pergunta1.getIdPergunta()+Constantes.separador1;//2

        bool1 = new boolean[choiceGroup[i].size()];
        choiceGroup[i].getSelectedFlags(bool1);
        for (int j=0; j < bool1.length; j++) {
            if (bool1[j]==true) {
                string2 +=
listaResposta[j].getIdResposta()+Constantes.separador1;//3
            }
        }

        string2 = string2.substring(0, string2.length()-1);
        string2 += Constantes.separador2;
        string1 += string2;
    }
    string1 += "0";//4
    //dataInicio_dataFim_idPergunta_respostas_resposta nao
    enviada_idPergunta_respostas_resposta nao enviada
    //ex: 1111111111_1111111112_10_1_11_1_2_0
    //System.out.println(string1);

    this.exibirIndexPesquisa.getPesquisaDAO().inserirPesquisa(string1);
}

public Form getFExibirPesquisa() {
    return this.fExibirPesquisa;
}

public ExibirIndexPesquisa getExibirIndexPesquisa() {
    return this.exibirIndexPesquisa;
}
}

```

ListarPesquisa.java

```

package view;

import java.util.Vector;

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

import action.ExibirIndexPesquisaAction;
import bean.Pesquisa;

public class ListarPesquisa implements CommandListener {

    private Main main;
    private List lListarPesquisa;
    private Command cSair;

    private Vector listaPesquisa;
    private int idUsuario = 1;

```

```

    public ListarPesquisa(Main main, Vector listaPesquisa) {
        this.main = main;
        this.listaPesquisa = listaPesquisa;
    }

    public void montarLayout() {

        lListarPesquisa = new List("Lista de Pesquisas",
List.IMPLICIT);
        for (int i=0; i<this.listaPesquisa.size(); i++) {
            Pesquisa pesquisa =
(Pesquisa)this.listaPesquisa.elementAt(i);
            lListarPesquisa.append(pesquisa.getNome(), null);

        }

        cSair = new Command("Sair", Command.EXIT, 0);

        lListarPesquisa.addCommand(cSair);

        lListarPesquisa.setCommandListener(this);
        this.main.getDisplay().setCurrent(lListarPesquisa);
    }

    public void commandAction(Command cmd, Displayable disp) {
        if (cmd == cSair) {
            this.main.destroyApp(false);
            this.main.notifyDestroyed();
        }
        else {
            Pesquisa pesquisal =
(Pesquisa)this.listaPesquisa.elementAt(lListarPesquisa.getSelectedIndex()
);
            ExibirIndexPesquisaAction exibirIndexPesquisaAction =
new ExibirIndexPesquisaAction(this.main, pesquisal, this.idUsuario);
            exibirIndexPesquisaAction.start();
        }
    }
}

```

Login.java

```

package view;

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

import action.LoginAction;

public class Login implements CommandListener {

    private Main main;

```

```

private Form fLogin;
private Command cSair, cEntrar;
private TextField tUsuario;
private TextField tSenha;
private StringItem sMensagem;

public Login(Main main) {
    this.main = main;
}

public void montarLayout() {
    fLogin = new Form("Login");

    tUsuario = new TextField("Usuário", "", 10, 0);
    tSenha = new TextField("Senha", "", 10, 0);
    sMensagem = new StringItem("", "");

    fLogin.append(tUsuario);
    fLogin.append(tSenha);
    fLogin.append(sMensagem);

    cEntrar = new Command("Entrar", Command.OK, 0);
    cSair = new Command("Sair", Command.EXIT, 0);

    fLogin.addCommand(cEntrar);
    fLogin.addCommand(cSair);

    fLogin.setCommandListener(this);
    this.main.getDisplay().setCurrent(fLogin);
}

public void usuarioInvalido() {
    this.sMensagem.setLabel("Usuário ou senha inválido(s)");
}

public void commandAction(Command cmd, Displayable disp) {
    if (cmd == cSair) {
        this.main.destroyApp(false);
        this.main.notifyDestroyed();
    }
    if (cmd == cEntrar) {
        //ExibirPesquisa exibirPesquisa = new
ExibirPesquisa(this.main, this, this.pesquisa);
        //exibirPesquisa.montarLayout();
        LoginAction loginAction = new LoginAction(this.main,
this, this.tUsuario.getString(), this.tSenha.getString());
        loginAction.start();
    }
}
}
}

```

Main.java

```
package view;

import javax.microedition.lcdui.Display;
import javax.microedition.midlet.MIDlet;

public class Main extends MIDlet {

    private Display display;

    public Main() {
        this.display = Display.getDisplay(this);

        /*int idUsuario = 1;
        ListarPesquisaAction listarPesquisaAction = new
ListarPesquisaAction(this, idUsuario);
        listarPesquisaAction.start();*/

        Login login = new Login(this);
        login.montarLayout();
    }

    public Display getDisplay() {
        return this.display;
    }

    public void destroyApp(boolean arg0) {}
    public void pauseApp() {}
    public void startApp() {}

}
```