

UMA ABORDAGEM SOBRE J2ME

Júlio César Jardim Júnior¹, Elio Lovisi Filho (Orientador)¹

¹Ciência da Computação – Universidade Presidente Antônio Carlos (UNIPAC)
Rua Palma Bageto Viol, s/n – Barbacena - MG.
jardimj@yahoo.com.br, professor_elio@nextwave.com.br

Resumo: *Este artigo vem mostrar conceitos e fundamentos do Java 2 Micro Edition (J2ME), com ênfase para o desenvolvimento de aplicações para telefones celulares, bem como apresentar ferramentas disponíveis para este fim, sem o intuito de esgotar o assunto, apresentando ainda a proposta de uma aplicação desenvolvida com a tecnologia J2ME, a aplicação consiste de uma sistemática de aprovações de documentos pelo celular.*

Palavra Chave: *Java, J2ME, Midlet, Dispositivos Móveis.*

1 – Introdução

A computação móvel tem crescido muito nos últimos anos. Com a chegada ao mercado e diminuição de preço dos *Personal Digital Assistants* (PDA's ou *Handheld's*) que são computadores de dimensões reduzidas mas com grande poder de processamento, e principalmente a evolução e agregação de funcionalidades destes dispositivos aos telefones celulares, criou-se um novo campo de desenvolvimento para a área da computação, o desenvolvimento de software para dispositivos móveis.

Esta é uma área que está em plena expansão e já existe em menor escala sistemas móveis implementados como acessórios para automóveis e pesquisadores há muito tempo falam de sistemas móveis implementados até em eletrodomésticos, o que já ocorre com sistemas encontrados, por exemplo, em receptores de sistemas de TV a cabo ou por satélite [KEULLER 2005].

A *Sun Microsystems* disponibilizou o J2ME, plataforma que nos permite trabalhar com desenvolvimento de aplicações para estes dispositivos. Esta frente vem sendo apoiada por diversos fabricantes como a Nokia, Siemens, Motorola, LG, Palm e outros. Como outros segmentos da plataforma Java, o J2ME vem crescendo com o total apoio destes fabricantes [KEULLER 2005].

2 – O J2ME

O J2ME é uma plataforma Java voltada para micro aplicativos que executam em micro processadores, assim como os dos celulares [CLÓVIS E JULIANO 2003]. Os dispositivos que suportam o J2ME, tem embutido em seu sistema operacional uma máquina virtual Java (JVM).

Dispositivos móveis tem características em comum, como processamento reduzido, limitações de memória e armazenamento, o J2ME é a ferramenta que permite a criação de aplicações que executam nestes dispositivos. O J2ME possui API's de criação de interface, armazenamento e conexões de rede. Assim como as demais plataformas Java o J2ME prove portabilidade nas aplicações desenvolvidas, tão importante, pois é grande a diversidade de dispositivos móveis que o mercado oferece.

A Figura 1 mostra as divisões da plataforma JAVA, dividido por poder de processamento e memória. Ela apresenta o *Java 2 Enterprise Edition* (J2EE), que é a plataforma de desenvolvimento para máquinas com maior poder de processamento como servidores e *workstations*, nesta categoria normalmente está o desenvolvimento para web, aplicações distribuídas e com volume de transações seguras. O *Java 2 Standard Edition* (J2SE) é a categoria normalmente utilizada para o desenvolvimento de sistemas cliente/servidor n-camadas, nela está contida por exemplo as API's de desenvolvimento de telas como a AWT ou *Swing* e desenvolvimento de *Applets*. Por último, vem o J2ME atendendo ao desenvolvimento, quando se trata de dispositivos móveis. Todas as categorias trabalham integradas fornecendo um conjunto completo de soluções para o desenvolvimento cliente/servidor, suportadas pelo núcleo da linguagem JAVA.



Figura 1. Plataformas Java e dispositivos suportados [Sun Microsystems 1]

3 – Arquitetura J2ME

A arquitetura J2ME é dividida em dois componentes básicos, *Configurations* e *Profiles*.

3.1 – Configurations

As *Configurations*, são as configurações de cada dispositivo que são agrupadas pelas características da JVM, que refletem o poder de processamento, recursos de memória e armazenamento do dispositivo. As *Configurations* são divididas em dois grupos CDC e CLDC [KEULLER 2005].

A Figura 2 mostra os limites das *Configurations* de acordo com o poder de processamento e memória dos diversos dispositivos móveis, mostrando aparelhos semelhantes que podem estar em uma categoria ou outra já que temos dispositivos que ultrapassam limites de suas características originais por exemplo, temos PDA's com funcionalidades de estações de trabalho e celulares com funcionalidades de PDA's e as vezes com alguns recursos encontrados em estações de trabalho como acesso a internet por *browser* comum e acesso a um *suite* de aplicações para escritório.

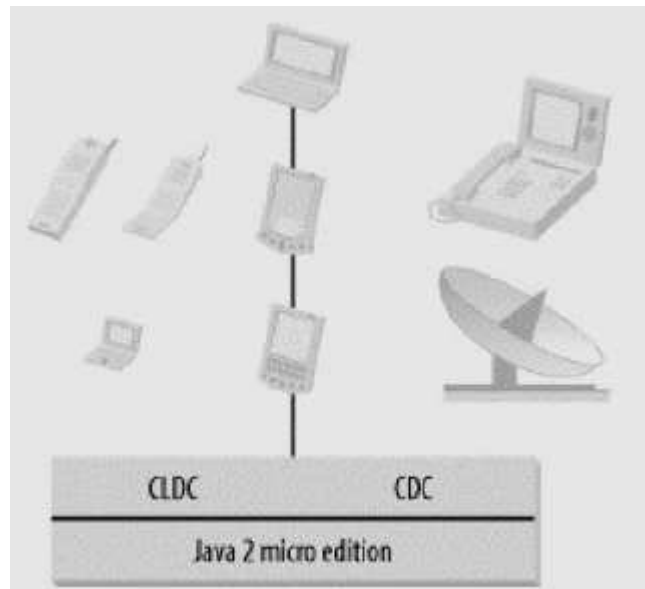


Figura 2. J2ME X CDC/CLDC [Sun Microsystems 1]

3.1.1 – CDC – Connected Device Configuration

Neste grupo de configuração estão os dispositivos com maior poder de processamento e maior capacidade de memória como por exemplo computadores de bordo, TV com suporte a internet, e celulares do tipo comunicadores [KEULLER 2005].

As Principais características destes dispositivos são processador de 32 bits, capacidade de 2MB ou mais para aplicações JAVA e conectividade com algum tipo de rede externa. As vezes possuem algum tipo de interface com o usuário.

3.1.2 – CLDC – Connected Limited Device Configuration

Neste grupo de configuração estão os dispositivos com menor poder de processamento e menor capacidade de memória, a maioria dos dispositivos com suporte ao J2ME estão neste grupo, e neste grupo estão a maioria dos telefones celulares e PDA's [KEULLER 2005].

As principais características destes dispositivos são poder de processamento limitado, capacidade entre 160Kb e 512Kb para aplicações JAVA, conectividade geralmente por Wireless, ou seja redes sem fio.

Normalmente possuem interface com o usuário, mas a maioria das vezes com um *display* de tamanho muito reduzido.

3.2 – Profiles

Um *Profile* é um conjunto de API's que reside dentro da JVM do aparelho e isso varia de *Configuration* para *Configuration*. Cada fabricante adiciona na sua *Profile* além da API Core do J2ME as API's de suporte a sua *Configuration*. A principal *Profile* do J2ME é a *Mobile Information Device Profile* (MIDP) [CHRISTIANO 2005].

3.2.1 – MIDP

Esta *Profile* foi projetada para operar em uma configuração CLDC para uso em aparelhos celulares e PDA's. A MIDP está na versão 2.0 e fornece um conjunto de classes para criação de interface, armazenamento de dados e conexão externa [CHRISTIANO 2005].

4 – Aplicações com MIDP

Para a criação de uma aplicação MIDP, tem que se utilizar de uma classe principal que estenda da classe *MIDLet*. Tecnicamente uma aplicação para celular é chamada de *Midlet* assim como os *Applets* e *Servlets*. Um *Midlet* é distribuído em um pacote, um arquivo “.jar”. Um *Midlet* possui um ciclo de vida que é gerenciado pela JVM [ROSFRAN 2004].

4.1 – Estrutura de um Midlet

A classe que estenda da classe *MIDLet* deve sobrescrever os métodos *startApp()*, *pauseApp()* e *destroyApp()*. O Quadro 1 mostra a estrutura da classe principal de uma aplicação *Midlet*.

```
import javax.microedition.lcdui.Display;
```

```
import javax.microedition.midlet.MIDlet;

public class Sag extends MIDlet {
    protected void startApp() {

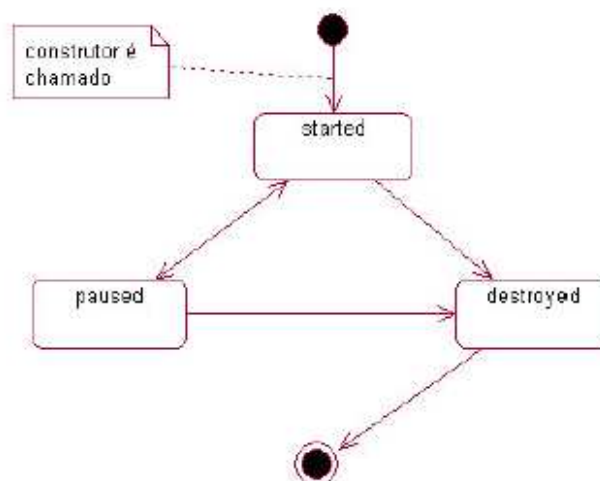
        Display d = Display.getDisplay(this);
    }
    protected void pauseApp() {}
    protected void destroyApp(boolean u) {}
}
```

Quadro 1. Código com a estrutura básica de um Midlet

A tela do dispositivo é representada por uma instância da classe Display, a qual é obtida pelo método `getDisplay()`, geralmente contida no método `startApp()`, pois o método `getDisplay()`, somente fica disponível após o início da Aplicação [ROSFRAN 2004].

A Figura 3 é um diagrama de estados que representa o ciclo de vida de um Midlet. O método `startApp()` é invocado quando se inicia uma aplicação Midlet, o `pauseApp()` é invocado quando acontece uma pausa na aplicação por exemplo se o celular recebe uma ligação enquanto um Midlet está executando e o `destroyApp()` é executado quando se finaliza a aplicação.

Figura 3. Diagrama de estados de um Midlet



4.2 – Interface em um Midlet

Geralmente, uma aplicação *Midlet* consiste em montar um *Display*, esperar a ação do usuário e decidir qual *Display* será exibido em seguida. Os *Displays* não são nada mais do que formulários, contendo componentes visuais de interação com o usuário. As ações do usuário são gerenciadas por comandos (*Commands*), os quais são adicionados a componentes visuais e estes componentes visuais por sua vez são adicionados ao *Display* [CLÓVIS E JULIANO 2003].

A Figura 4 mostra a hierarquia de classes disponíveis para a montagem de uma interface com o MIDP. A hierarquia inicializa pela classe *Display*, cada *Midlet* possui apenas um *Display*, assim parte por *Displayable*, para classes de tela como a *Screen* que contém classes de formulário como *Form* e *TextBox* ou *Canvas* que é a parte de desenho livre em tela muito utilizada em jogos.

Criar uma interface adequada pode representar um problema uma vez que o *display* da maioria dos dispositivos é de tamanho reduzido, nos celulares típicos a tela é de 96x54 pixels [CLÓVIS E JULIANO 2003].

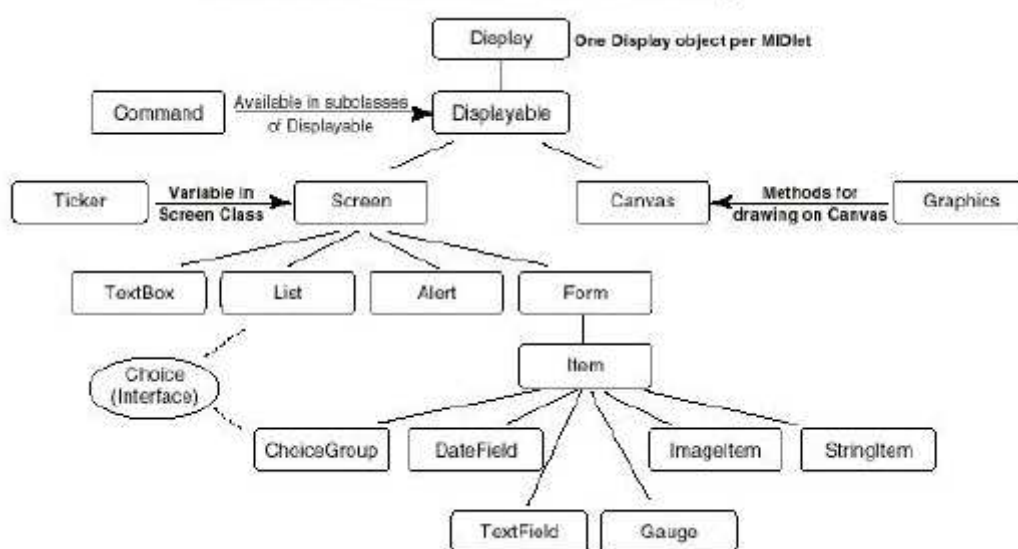


Figura 4. Hierarquia de Classes do MIDP – Interface [Core J2ME 1]

4.3 – Conexões

Além da API de interface, o MIDP fornece em sua API classes de conexão a uma rede, a classe *HttpConnection*. Os dispositivos móveis normalmente oferecem conexão *Over The Air* (OTA) via *General Packet Radio Service* (GPRS) que é uma tecnologia que aumenta as taxas de transferência de dados nas redes de telefonia existentes, permitindo o transporte de dados por pacotes [Wikipedia].

No MIDP a principal forma de comunicação é via HTTP (classe *HttpConnection*), o que permite que a parte do servidor que responderá ao *Midlet* seja implementada em qualquer tecnologia que trabalhe com o protocolo HTTP [CLÓVIS E JULIANO 2003].

A Figura 5 mostra a hierarquia de classes para conexão do MIDP, partindo da interface *Connection* e através de interfaces de leitura e escrita, normalmente feitas *Byte a Byte* até chegar na classe *HttpConnection* que realmente abre a comunicação implementando este protocolo no Midlet.

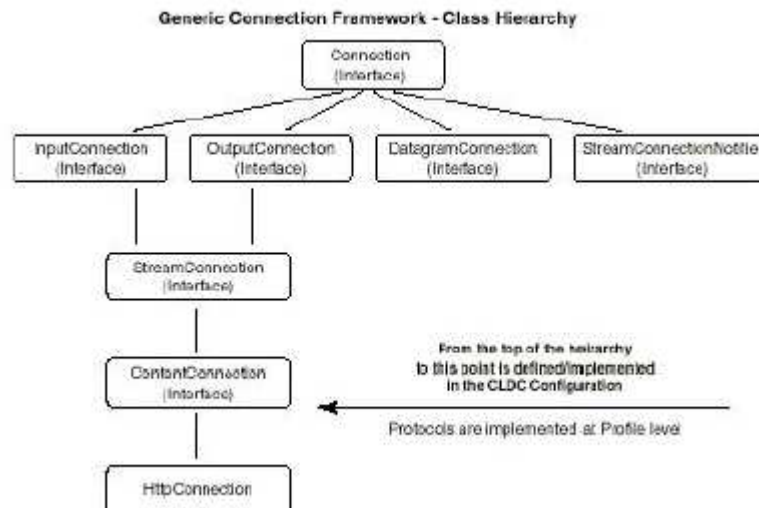


Figura 5. Hierarquia de Classes do MIDP – Conexão [Core J2ME 2]

4.4 – Armazenamento

O MIDP oferece também uma classe de armazenamento de dados em memória através da classe *RecordStore*. O *RecordStore* provê uma forma bem simples de armazenamento apresentando cada registro como um ID (do tipo inteiro) e um *array* de Bytes, para dividir um registro em campos é necessário a manipulação do *array* de Bytes colocando um separador de *string's*, exemplo para armazenar nome e telefone deve-se utilizar, "Nome|Telefone", tendo "|" como separador de *string* [CLÓVIS E JULIANO 2003].

5 – Ferramentas de Desenvolvimento

No mercado encontramos ferramentas disponíveis para auxiliar o ciclo de desenvolvimento de aplicações J2ME.

A *Sun Microsystems* disponibiliza o "J2ME - *Wireless ToolKit*" que compila, empacota e testa, via emulador um *Midlet*, disponível em <<http://java.sun.com/products/sjwtoolkit/>>.

Para a instalação do *Wireless ToolKit* é necessário a instalação do Java JDK versão 1.4 ou superior, disponível em <http://java.sun.com/products/archive/j2se/1.4.1_07/>.

Para edição de código pode ser utilizado como IDE o Eclipse da IBM, disponível em <<http://www.eclipse.org/downloads/>>. O Eclipse possui um *plugin* específico para J2ME e comunica diretamente com o *Wireless ToolKit*, disponível em <http://sourceforge.net/project/showfiles.php?group_id=86829>.

Estas ferramentas são gratuitas e estão disponíveis nos sites dos fabricantes acima citados [KEULLER 2005].

6 – Aplicação SAG

Para a testar o J2ME foi implementado um protótipo de uma aplicação denominada Sistema de Aprovações Gerenciais (SAG). O sistema consiste de uma aplicação que prove aprovações de documentos via celular. Por exemplo, se uma empresa determina que seus gestores aprovem documentos de

pagamentos diversos a empregados, via sistema, este gestor recebe no celular dados do documento, analisa e aprova ou não este pagamento.

No sistema SAG a idéia é que o gestor aprove documentos de pagamento de bolsa de estudos de seus empregados.

6.1 – Arquitetura da Aplicação SAG

No lado do cliente foi implementado o SAG.jar, um *Midlet* em J2ME que faz requisições a um servidor que trabalha com o protocolo HTTP.

A Figura 6 mostra o diagrama de componentes da arquitetura da aplicação SAG mostrando o lado do cliente o *Midlet* SAG.jar e do servidor a página Back_Side.asp.

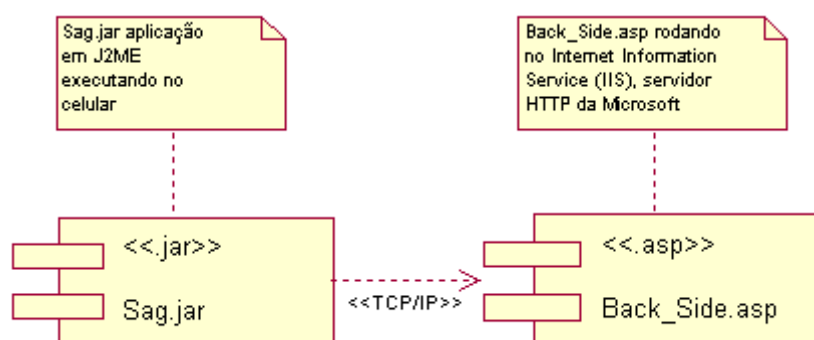


Figura 6. Diagrama de Componentes da Arquitetura do SAG

6.1.1 – SAG.jar

O Sag.jar foi desenvolvido da seguinte forma:

- 1 – Criada uma classe Sag que estende a classe *MIDLet* e caracteriza uma aplicação *Midlet*;
- 2 – A classe Sag instancia a *GuiPrincipal*, que monta a tela principal de Interface com o usuário;

3 – A classe GuiPrincipal instancia a classe GuiLista que através da classe Conexao conecta no servidor. Esta conexão obtém a lista de documentos preenchendo uma composição da classe BeanAprovacao que contem as características dos documentos a serem analisados.

4 – A classe GuiLista pode instanciar a classe GuiDetalhe que mostra detalhes dos documentos da lista, ou pode submeter a aprovação (ou reprovação) os documentos selecionados na lista através da classe Conexao.

A Figura 7 mostra o diagrama de classes do sistema SAG, o diagrama está simplificado mostrando métodos e atributos mais relevantes.

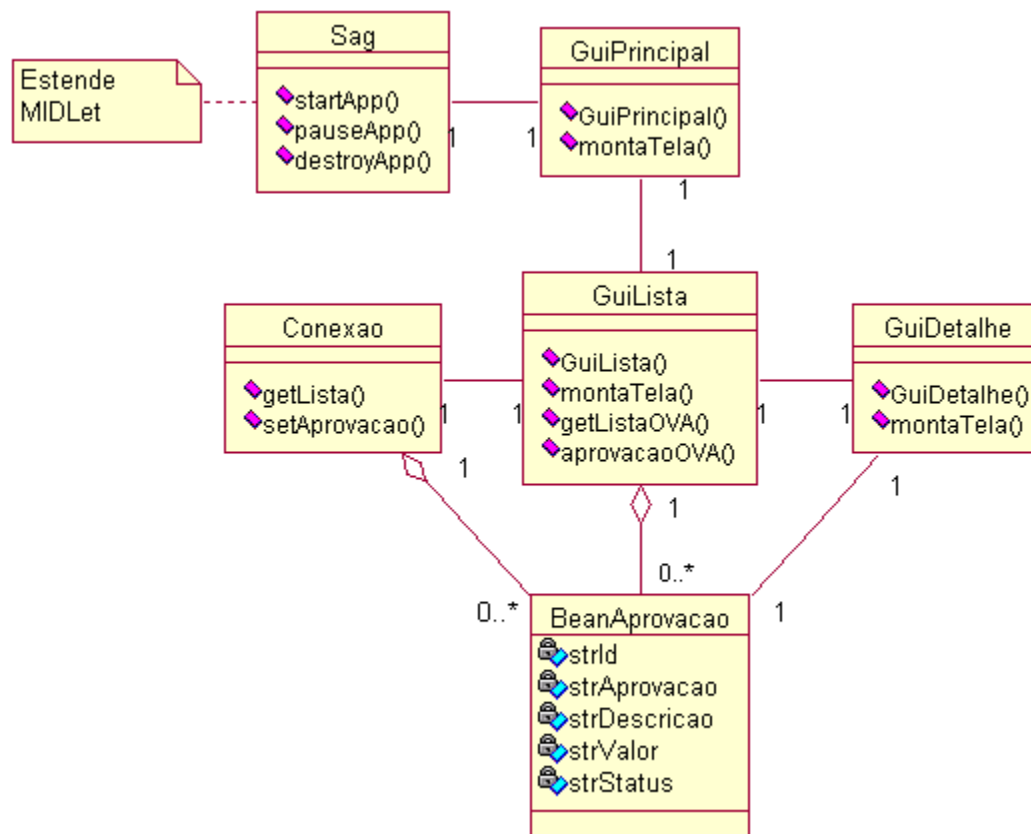


Figura 7. Diagrama de Classes do SAG

6.1.2 – Back_Side.asp

O lado servidor utiliza o *Internet Information Service* (IIS) da Microsoft, que é um servidor que trabalha com o protocolo HTTP, que suporta páginas *Active Server Pages* (ASP). O ASP é a linguagem de *script* da Microsoft para construção de páginas para a internet e trabalha com os objetos *response* e *request*. O ASP possui também poder de instanciar componentes *Activex*, o que prove por exemplo acesso a objetos de conexão com banco de dados e email.

A página *back_side.asp* do lado do servidor é responsável por receber requisições do SAG.jar e devolver a lista de documentos, ou processar aprovações. Dentro da página que representa a camada de negócio da aplicação está contida a lógica de acesso a uma base de dados (*sag.mdb* – em MS Access) e a lógica de montagem da lista de retorno e o processo de gravação de aprovações. A página está hospedada no endereço <www.artigoj2me.vistait.com.br>. A tecnologia escolhida para implementar o lado servidor foi com o ASP mas poderia ter sido qualquer outra como por exemplo com um *servlet*.

6.1.3 – Operação do SAG

No SAG os documentos a serem analisados podem ter três *status*, aprovados, reprovados e pendentes.

A Figura 8 – Tela 1 mostra a tela principal do SAG onde o usuário escolhe quais os documentos devem ser retornados para análise por *status* e qual opção de processamento, sair do sistema ou carregar a lista.

A Figura 8 – Tela 2 mostra a lista de documentos retornados pelo servidor e qual opção de processamento, ver detalhes de um documento, aprovar ou reprovar em lote e voltar a tela anterior.

A Figura 8 – Tela 3 mostra os detalhes de um documento caso o usuário tenha optado por detalhe na tela anterior. Esta tela é navegacional através do *menu* que tem as opções de próximo, anterior e voltar.



Figura 8. Telas do SAG

7 – Considerações Finais

Apesar de toda a mobilidade dos *Notebooks*, com pontos de acesso em hotéis e aeroportos, a conectividade de um celular é mais abrangente. A maioria dos celulares inclusive os mais em conta, saem hoje com suporte ao Java, isto é um atrativo para o desenvolvimento nesta área.

O J2ME se mostrou uma ferramenta funcional, poderosa e estável, com ela é possível implementar desde jogos a sistemas corporativos, em um celular.

O mercado oferece ferramentas que realmente facilitam o ciclo do desenvolvimento de software nesta área.

O desenvolvimento de aplicações para dispositivos móveis é um campo promissor de trabalho na área de Ciências da Computação.

O Java é mais do que uma linguagem de programação é um ambiente de desenvolvimento de aplicações que oferece orientação a objetos, portabilidade, dinamismo e alto desempenho, abrangendo através do J2EE, J2SE e J2ME a possibilidade de desenvolvimento para praticamente qualquer plataforma hoje existente.

Como sugestão para trabalhos futuros fica estender a aplicação SAG melhorando principalmente a interface com o usuário e implementando a

camada do servidor com J2EE/Servlet's utilizando assim todo o potencial oferecido pela arquitetura Java.

8 – Referências Bibliográficas

- [CLÓVIS E JULIANO 2003] CARNIEL, Juliano; TEIXEIRA, Clóvis. **Apostila de J2ME**. Disponível em: <www.portaljava.com.br/home/modules.php?name=News&file=article&sid=524>. Acesso em: 26 nov. 2005.
- [CHRISTIANO 2005] PINHEIRO, Christiano. **J2ME - Java para os portáteis**. Disponível em: <<http://www.imasters.com.br/artigo.php?cn=1539&cc=20>>. Acesso em: 26 nov. 2005.
- [ROSFRAN 2004] BORGES, Rosfran. **J2ME na Prática**. Disponível em: <http://www.ucb.br/java/JavaDays/J2ME_RosfranBorges.pdf>. Acesso em: 26 nov. 2005.
- [KEULLER 2005] MAGALHÃES, Keuller. **Criando Aplicações J2ME com Eclipse**. Disponível em: <http://www.cafeh.com.br/artigos/12/Aplicacoes_J2ME_com_Eclipse.pdf>. Acesso em: 26 nov. 2005.
- [Sun Microsystems 1] SUN Microsystems. **Java Platform, Micro Edition (Java ME) Overview**. Disponível em: <<http://java.sun.com/javame/overview.html>>. Acesso em: 26 nov. 2005.
- [Sun Microsystems 2] SUN Microsystems. **Mobile Information Device Profile (MIDP); JSR 37, JSR 118**. Disponível em: <<http://developers.sun.com/techtopics/mobility/midp/index.html>>. Acesso em: 20 maio. 2006.
- [Sun Microsystems 3] SUN Microsystems. **J2SE**. Disponível em: <http://java.sun.com/products/archive/j2se/1.4.1_07>. Acesso em: 20 maio 2006.
- [Sun Microsystems 4] SUN Microsystems. **J2ME - Wireless Toolkit**. Disponível em: <<http://java.sun.com/products/sjwtoolkit/>>. Acesso em: 20 maio 2006.
- [Wikipedia] Wikipedia. **GPRS**. Disponível em: <<http://pt.wikipedia.org/wiki/GPRS>>. Acesso em: 20 maio. 2006.
- [Core J2ME 1] Core J2ME. **Core J2ME Book**. Disponível em: <<http://www.corej2me.com/CoreJ2MEBook/Hierarchy/Core%20J2ME%20Technology%20and%20MIDP.pdf>>. Acesso em: 20 maio 2006.
- [Core J2ME 2] Core J2ME. **Core J2ME Book**. Disponível em: <<http://www.corej2me.com/CoreJ2MEBook/Hierarchy/Core%20J2ME%20Technology%20and%20GCF.pdf>>. Acesso em: 20 maio 2006.
- [ECLIPSE] ECLIPSE. **ECLIPSE**. Disponível em: <<http://www.eclipse.org/downloads/>>. Acesso em: 26 nov. 2005.

[Sourceforge] Sourceforge. **ECLIPSE and J2ME plugin**. Disponível em:
<http://sourceforge.net/project/showfiles.php?group_id=86829>. Acesso em:
20 maio 2006.