

Clésio Rubens de Matos

**ReMoS – Um Software para a Gerência de Servidores
Linux em Dispositivos Móveis usando a Linguagem de
Programação J2ME e o Protocolo GPRS/EDGE**

Monografia de Pós-Graduação “Lato Sensu”
apresentada ao Departamento de Ciência da
Computação para obtenção do título de
Especialista em “Administração em Redes
Linux”

Orientador
Prof. Msc. Joaquim Quinteiro Uchôa

LAVRAS
MINAS GERAIS - BRASIL
2009

Clésio Rubens de Matos

**ReMoS – Um Software para a Gerência de Servidores
Linux em Dispositivos Móveis usando a Linguagem de
Programação J2ME e o Protocolo GPRS/EDGE**

Monografia de Pós-Graduação “Lato Sensu”
apresentada ao Departamento de Ciência da
Computação para obtenção do título de
Especialista em “Administração em Redes
Linux”

Aprovada em 19 de Abri de 2009

Prof. Herlon Ayres Camago

Prof. Marluce Rodrigues Pereira

Prof. Msc. Joaquim Quinteiro Uchôa
(Orientador)

LAVRAS
MINAS GERAIS - BRASIL
2009

Agradecimentos

Prestamos o nosso primeiro e principal agradecimento ao grandioso Deus, que de forma ilustre e invisível nos conduziu, mostrando o caminho para conclusão deste projeto.

Ao Prof. Joaquim Uchôa, orientador deste trabalho, pelo seu apoio e confiança em nós depositados, oportunizando a execução desta.

Ao prof. Heitor Costa, co-orientador deste trabalho, pelo incentivo e apoio disponibilizados na orientação inicial.

Aos demais professores do curso, que compartilharam conhecimento em cada disciplina estudada durante a especialização, deixando assim a sua contribuição para a realização deste trabalho.

Prestamos também agradecimentos aos colegas do curso, pelo compartilhamento e ajuda nos estudos, e nos pouquíssimos momentos de lazer.

À noiva Caroline Martins pela cumplicidade, força e apoio dedicados a nós durante todo o desenvolvimento deste trabalho, tendo um papel especial em todo o processo.

Por fim, prestamos um agradecimento todo especial aos nossos familiares, nas pessoas de Rubens Matos, Eliene Matos e Gabriela Matos, que conseguiram estar como um porto seguro para suprir todas nossas dificuldades durante esta caminhada. E assim compartilhamos agora com todos vocês o regozijo no término deste projeto.

Sumário

1. Introdução.....	1
1.1. Motivação	3
1.2. Objetivo	5
1.3. Metodologia	6
1.4. Estrutura do Trabalho	9
2. Plataforma Java	11
2.1. Considerações Iniciais	11
2.2. Plataforma de Desenvolvimento Java.....	12
2.2.1. Edição Java J2SE	13
2.2.2. Edição Java J2EE.....	15
2.2.3. Edição Java J2ME.....	18
2.3. Considerações Finais	21
3. Tecnologia Móvel	23
3.1. Considerações Iniciais	23
3.2. Sistema Operacional Móvel	24
3.2.1. Sistema Operacional Mobile Symbian	25
3.2.2. Sistema Operacional Windows CE (Mobile)	26
3.3. Tecnologias para Celular	27
3.3.1. 1º Geração	27
3.3.2. 2º Geração	28
3.3.2.1. Tecnologia de Transmissão de Dados GPRS	29
3.3.2.2. Tecnologia de Transmissão de Dados EDGE ...	30
3.3.3. 3º Geração	30
3.4. Considerações Finais	31
4. Trabalhos Relacionados	32
4.1. Considerações Iniciais	32
4.2. Desenvolvimento de Software Móvel	33
4.3. Redes Wireless e Móveis	34
4.4. Automação de Tarefas Através de Aplicações Móveis	35

4.5.	Considerações Finais	36
5.	ReMoS – <i>Remote Mobile Server</i>	37
5.1.	Considerações Iniciais	37
5.2.	Modelagem do ReMoS	38
5.2.1.	Requisitos do ReMoS	38
5.2.2.	Diagrama de Casos de Uso	39
5.2.3.	Diagrama de Atividades.....	42
5.2.4.	Diagrama de Implantação	44
5.3.	Funcionamento do ReMoS.....	44
5.3.1.	Módulo Mobile.....	44
5.3.1.1.	Acesso ao ReMoS	46
5.3.1.2.	Funções de Controle de Processos	48
5.3.1.3.	Funções de Controle de Serviços	48
5.3.1.4.	Funções de Sistema	50
5.3.1.5.	Funções Configuráveis.....	51
5.3.2.	Módulo Server	51
5.3.2.1.	Execução de Tarefas	52
5.3.2.2.	Comunicação Cliente x Servidor	55
5.4.	Considerações Finais	56
6.	Considerações Finais.....	57
6.1.	Conclusões	57
6.2.	Contribuições	58
6.3.	Trabalhos Futuros	59
	BIBLIOGRAFIA.....	60

Lista de Figuras

Figura 2-1 – Divisão da Plataforma Java (Fonte: Andreão, 2008) ...	14
Figura 2-2 – Divisão da Plataforma Java J2SE (Fonte: SUN, 2008d)	16
Figura 2-3 – Ambiente J2EE (Fonte: SUN, 2008a)	19
Figura 2-4 – Arquitetura da Plataforma J2ME (Fonte: Nogueira <i>et al</i>, 2005).....	20
Figura 2-5 - Arquitetura de Execução da Plataforma J2ME (Fonte: IBM, 2008).....	22
Figura 5-1 – Diagrama de Casos de Uso do Módulo Mobile.....	40
Figura 5-2 – Diagrama de Casos de Uso do Módulo Server.....	41
Figura 5-3 – Diagrama de Atividades do módulo Mobile.....	43
Figura 5-4 – Diagrama de Atividades do Módulo Server	45
Figura 5-5 - Diagrama de Implantação do ReMoS	46
Figura 5-6 – Controle de Acesso ao ReMoS.....	47
Figura 5-7 – <i>Menu</i> de Funções do ReMoS	47
Figura 5-8 – a) <i>Menu</i> do ReMoS b) Relatório de Uso da Memória .	49
Figura 5-9 – a) <i>Menu</i> do ReMoS b) Visualização da Tabela ARP... 	50
Figura 5-10 – a) Chamado do <i>Script</i> pessoal uso_hd. b) Resultado da Execução do <i>Script</i> no Servidor.....	52
Figura 5-11 – Estrutura do <i>Script</i> a ser Executado no Módulo Server	54
Figura 5-12 – Código para Implementar Conexão com o Módulo Sever.....	55
Figura 5-13 – Código para Implementar Transferência de Dados... 	56

Lista de Tabelas

Tabela 5-1 – Requisitos Funcionais do ReMoS..... 38

Tabela 5-2 – Requisitos Funcionais do ReMoS..... 39

..

Resumo

Para manter controle efetivo de servidores de redes de computadores, o administrador necessita de ferramentas eficientes, que possam lhe oferecer, constantemente, monitoramento e controle de falhas destas máquinas. Assim sendo, o objetivo deste trabalho é usar tecnologias de desenvolvimento de *software* para dispositivos móveis, na criação de um aplicativo *mobile* para o controle remoto de servidores Linux. Este aplicativo tem alta disponibilidade no diagnóstico e na resolução de falhas de serviços com soluções pré-programadas. O *software*, denominado de ReMoS (Remote Mobile Server), foi desenvolvido usando a plataforma Java *Micro Edition* (J2ME) e os protocolos de conexão com a Internet via celular. Porém, apenas algumas funções foram disponibilizadas aos administradores de rede, principalmente, por causa do custo de transmissão de dados entre o dispositivo móvel e o servidor. As funções implementadas foram escolhidas mediante entrevista (conversa informal) com alguns administradores de rede, para identificar quais funções são importantes no seu ponto de vista e viáveis de implementação. Desta forma, o ReMoS é um aplicativo que pode ser muito útil na administração remota de redes de computadores.

Palavras Chaves: Celular, GPRS, EDGE, J2ME, Shell Script, *Mobile*, *Wireless*.

1. Introdução

A popularização dos dispositivos móveis, em particular os aparelhos celulares, vem caminhando a passos largos, constituindo hoje uma quantidade na casa de bilhões de usuários no mundo. Este fato ocorre em consequência principalmente do avanço tecnológico, da miniaturização, do aumento de poder de processamento desses dispositivos e da necessidade crescente do ser humano estar informado a qualquer hora e em qualquer lugar [Bottentuiut, 2008].

Com o acréscimo da quantidade de celulares, a tecnologia *wireless* (sem fio) tem sido mais explorada e utilizada com grande frequência, tendo grande relevância para aplicações que utilizam este meio de comunicação. Dentre as aplicações, pode-se citar: informações sobre o trânsito, meteorologia, informações financeiras e publicidade em geral, permitindo ao usuário a mobilidade na busca por estes serviços, pelo fato de não estarem limitados à conexão de um cabo para seu funcionamento, dispondo destas informações em dispositivos móveis, como é o caso dos celulares [Pereira, 2008].

Pelo fato da tecnologia celular prover um mercado crescente, há constantes descobertas de novas tecnologias que fazem os aparelhos cada vez mais úteis, informativos e atraentes. Como prova disso, nos celulares mais modernos tem-se a possibilidade de executar aplicativos apenas existentes em computadores *desktops*, aumentando o leque de possibilidades para os usuários. Estes aplicativos podem ser desde uma simples agenda a um *software* de relatórios gerenciais de uma empresa.

Atualmente, existe pequena limitação quanto ao poder de processamento e a quantidade de memória destes aparelhos, questões estas que vem sendo resolvidas nos novos aparelhos produzidos, principalmente com aumento da capacidade de armazenamento, possibilitando armazenar arquivos como músicas, vídeos e *software* [Bottentuiut, 2008].

Diante dessa grande oportunidade, os gigantes da informática estão investindo cada vez mais nesta área. Um exemplo é a Sun Microsystem, que formulou uma versão de sua plataforma de desenvolvimento Java, exclusivamente para dispositivos móveis como o celular. O nome dado foi J2ME (*Java 2 Micro Edition*). O Java se destaca de outras linguagens de programação, pois além de dispor de recursos de Orientação a Objetos, permite o desenvolvimento de códigos portáteis, reutilizáveis, robustos e seguros. A plataforma de desenvolvimento Java viabiliza a implementação de aplicativos para qualquer dispositivo, desde *SmartCard*¹, passando por celulares e computadores até grande servidores, uma vez que eles possuam uma máquina virtual Java (JVM – *Java Virtual Machine*) [SUN, 2008c].

Um fator de grande valia para incrementar este cenário é a possibilidade de aparelhos celulares se conectarem com a rede mundial de computadores (Internet), fazendo o acesso a *sites*, possibilitando a troca de dados com qualquer computador pertencente a esta rede, ou seja, tornando-se membro dela. Assim sendo, há a possibilidade de executar nestes dispositivos, a mesmas funcionalidades de uma máquina com

¹ *SmartCard* – Cartão criptográfico de Certificados Digitais.

maior poder de processamento e de armazenamento, isto é, com poucas limitações [Cruz, 2008].

Com a combinação do acesso a Internet via celular e da possibilidade de construção de *software* para estes aparelhos, pode-se implementar aplicativos que realizem duas operações nos servidores de rede: i) consulta; e ii) controle. Estas operações são atrativas àqueles que desempenham funções que exigem constante mobilidade, por exemplo consultores, supervisores e auditores de informática.

1.1. Motivação

Segundo a fabricante de aparelhos celulares Samsung, as vendas globais de celulares atingiram 291,5 milhões de aparelhos no primeiro trimestre do ano de 2008, atingindo neste mesmo ano o total de 3,66 bilhões de assinantes de telefones celulares no mundo [TELECO, 2009a; Tan, 2009]. O Brasil, com venda de mais de 1,3 milhões de aparelhos celulares em janeiro de 2009, aproxima-se da marca de 152 milhes de celulares em funcionamento, segundo dados da Agência Nacional das Telecomunicações – Anatel [IDGNOW, 2009].

Essas estatísticas permitem verificar que o mercado de aparelhos celulares torna-se um dos maiores na atualidade e em constante crescimento. Com tal expansão, o mercado está voltando sua atenção para a comunicação de dados na telefonia celular, produzindo aparelhos capazes de enviar fotos e vídeos em tempo real com menor custo, fazendo com que o interesse dos desenvolvedores de *software* seja despertado [Monte, 2008]. A construção de *software* para dispositivos móveis, em

particular aparelhos celulares, vem a ser de grande valia para incrementar o leque de informações disponibilizadas, a qual, ao se aliar à possibilidade de conexão com a Internet, traz ao homem moderno uma nova ferramenta completa, móvel e de grande disponibilidade para auxiliá-lo nas diversas áreas do conhecimento [Monte, 2008].

Seguindo tais expectativas, este trabalho é focado no desenvolvimento de um *software* dirigido a esta fatia do mercado. O diferencial deste *software* consiste na agregação de valores junto ao aparelho celular, mostrando a possibilidade de efetuar o gerenciamento básico de servidores através deste dispositivo, de forma rápida e segura, aliado ao conforto e à alta disponibilidade que estes aparelhos móveis podem oferecer. O administrador de redes pode utilizar este *software* em seu celular, efetuando conexões ao servidor da empresa para diagnosticar e resolver pequenos problemas da rede de forma remota, sem a necessidade de um computador ligado à internet para efetuar a tarefa.

A criação de *software* para aparelhos celulares traz uma gama de oportunidades e um grande mercado em potencial, com implementações que facilitam o cotidiano das pessoas. Estes aplicativos podem ser utilizados para diversos fins, por exemplo, acesso a contas bancárias e realização de transações como pagamentos, extratos, consulta de saldos, compras pela internet e facilidades na área de saúde como marcação de consultas em clínicas que disponibilizem o serviço. A construção de *software* com esta funcionalidade é baseada em princípios discutidos e explanados neste trabalho, podendo este servir como base para tais

implementações, colaborando para sua utilidade no comércio e na economia [Pereira, 2008].

Além da importância econômica e comercial do resultado deste trabalho, pode-se também citar sua relevância acadêmica, tornando-se uma fonte de pesquisa para aqueles que desejarem conhecer particularidades da programação para aparelhos celulares e Internet e do controle básico de servidores de rede.

1.2. Objetivo

O objetivo geral deste trabalho é desenvolver um software com arquitetura cliente/servidor, utilizando a linguagem de programação Java (plataforma J2ME) e protocolo GPRS/EDGE (*General Packet Radio Service/Enhanced Data Rates for Global Evolution*), de forma que este *software* funcione em um aparelho celular, realizando conexão com um servidor localizado na Internet. Assim, será possível realizar consultas e enviar comandos para serem executadas no servidor.

De acordo com o objetivo geral do projeto, os objetivos específicos definidos são:

- Construir um *software* para acesso remoto via aparelho celular aos servidores de rede de uma empresa;
- Demonstrar vantagens existentes na consulta e no processamento de dados por meio de dispositivos móveis em detrimento dos dispositivos fixos;
- Construir um *software* alternativo de baixo custo para substituir os *laptops*, utilizando os populares aparelhos celulares com acesso a rede

GPRS/EDGE da operadora, no processo de gerenciamento de servidores;

- Gerar alta disponibilidade no acesso remoto de servidores, proporcionando formas rápidas e seguras de análise e de resolução de pequenos problemas em máquinas e serviços de uma rede, minimizando a perda de tempo na execução destas tarefas.

1.3. Metodologia

A finalidade deste trabalho é a construção de um *software* para aparelhos celulares, usando a tecnologia Java, que se conecte a um servidor na Internet, possibilitando a consulta e envio de comandos para serem executados. Para isso, a tecnologia J2ME e o GPRS/EDGE serão usados para conexão com Internet em aparelhos celulares GSM (*Global System for Mobile Communications*).

Os *Midlets* são programas Java, aplicações executadas nos aparelhos celulares para exercer uma função desejada. A plataforma J2ME possibilita o desenvolvimento de programas segundo a abordagem orientada a objetos, sendo que o processamento das consultas e a gravação das informações ocorrem no servidor, pois os aparelhos celulares são dispositivos limitados no armazenamento e no processamento das informações [JCP, 2009].

O modelo de comunicação Cliente/Servidor foi utilizado como arquitetura, tendo como cliente o aplicativo do celular e como servidor um *software* executado no nó² servidor na Internet.

O projeto foi dedicado à tecnologia GSM de telefonia celular, com testes e aprovações operadas em celulares nela inclusos. Para conexão com a Internet, foram utilizados a tecnologia GPRS/EDGE integrada a GSM e o protocolo de conexão HTTP (*Hypertext Transfer Protocol*).

O *software* no servidor foi implementado usando a distribuição Java J2EE (*Java Enterprise Edition*) específica para aplicações corporativas e Internet, abrangendo apenas as funções necessárias para atender às solicitações do aparelho celular e executar os comandos. O *software* servidor é executado com poderes de administrador (*root*), possibilitando a execução das tarefas solicitadas pelo *software* cliente [SUN, 2008a].

O *software* servidor atuou como uma camada intermediária entre o cliente e a máquina, recebendo requisições do cliente, executando os comandos desejados e reportando ao cliente o resultado os comandos realizados. Estes comandos foram executados usando arquivos de *Shell Scripts* predeterminados no servidor, permitindo ao *software* um grande leque de funções, pois se pode implementar *Scripts* para desempenhar funções diferentes, sendo estas executadas via *software* cliente [Neves, 2008].

² Nó – Qualquer equipamento autônomo capaz de realizar algum tipo de processamento em uma rede.

Para a configuração e implantação do *software* servidor, foi utilizado o servidor Web TomCat, produzido pela Fundação Apache, que permite a execução de aplicações para Web construídas na linguagem de programação Java, como os *Servlets*³ desenvolvidos nesta implementação [TOMCAT, 2009].

A análise e o desenvolvimento do módulo de *software* foram essencialmente orientados a objetos, adotando a UML (*Unified Modeling Language*) para especificação e documentação dos casos de uso implementados no aplicativo. Algumas ferramentas de especificação UML foram pesquisadas, adotando o Umbrello como padrão de uso, pois fornece suporte na geração automática de código Java, como classes, atributos e objetos, fornecendo agilidade e segurança na implementação, além do conhecimento prévio da ferramenta pelos desenvolvedores do projeto [Medeiros, 2008].

Para controle e gerenciamento do desenvolvimento do *software*, foi adotado o ciclo de vida em espiral através da metodologia YP (*Easy Process*). Esta metodologia permite ao desenvolvedor entender e reagir aos riscos em cada etapa evolutiva do processo. Outro ponto importante desta metodologia é o uso constante de prototipação, possibilitando aplicar a abordagem de prototipação em qualquer etapa da evolução do produto, reduzindo os riscos antes que se tornem problemáticos [YP, 2008].

³ *Servlets* – Aplicativos implementados usando a linguagem programação Java e a plataforma J2EE.

Foi usada a *suite* J2ME baseada no *Java Standart Edition* (J2SE) para aplicações *desktop*, no entanto, com algumas particularidades e recursos reduzidos, destinados aos dispositivos com pouca capacidade de armazenamento e processamento, devido ao tamanho de seus *displays* e pouca quantidade de memórias ROM (*Read Only Memory*) e RAM (*Random Access Memory*), como é o caso dos aparelhos celulares.

O Wireless Toolkit versão 2.5.1 da Sun foi usado, para geração de código e pré-visualização, simulando o *display* do aparelho celular e evitando custos de conexões com a Internet na realização de testes durante a implementação. Ele possui um repositório para simulação de alguns tipos de aparelhos celulares, incluindo aparelhos do fabricante Motorola. O NetBeans, com o pacote para desenvolvimento de aplicativos moveis Mobility Pack, foi usado como editor de código, por ser uma IDE (*Integrated Development Environment*) de desenvolvimento robusta, rápida e de fácil uso [SUN, 2008b; NETBEANS, 2008].

O ambiente de desenvolvimento foi configurado na plataforma Linux distribuição Ubuntu na versão 8.04. Os aplicativos escolhidos para o desenvolvimento do projeto são de livre distribuição e com bons conceitos de desempenho.

1.4. Estrutura do Trabalho

Este trabalho está dividido em seis capítulos. Neste primeiro capítulo, é introduzido o assunto tema da pesquisa, sendo abordados também a motivação, o objetivo e a metodologia.

O Capítulo 2 apresenta o referencial teórico do estudo, abrangendo as principais plataformas de desenvolvimento de aplicativos para dispositivos portáteis, com análise de suas arquiteturas e modelos de desenvolvimento.

O Capítulo 3 contém breve introdução sobre os sistemas operacionais de dispositivos móveis, abrangendo as tecnologias de aparelhos celulares existentes e os meios de conexão para transmissão de dados (Internet), utilizados por estes aparelhos.

O Capítulo 4 contém um estudo sobre outros trabalhos relacionados à linha de pesquisa, mostrando a atualidade e relevância do tema proposto.

O Capítulo 5 é dedicado à explicação de funcionamento do *software* ReMoS, apresentando a modelagem, a documentação, as técnicas usadas na implementação e o uso do ReMoS.

O Capítulo 6 discute-se as contribuições, as perspectivas futuras deste trabalho e as conclusões alcançadas.

2. Plataforma Java

2.1. Considerações Iniciais

O objetivo deste capítulo é explicar as principais plataformas de desenvolvimento para celular, tendo como foco principal a linguagem de programação Java, demonstrando as particularidades de programação desta tecnologia.

Apesar dos celulares serem bem conhecidos e comercializados para usuários, há pequena quantidade de profissionais no mercado que desenvolvem aplicações para esses dispositivos, existindo quantidade limitada de linguagens de programação destinadas a este uso. Estes aparelhos se encaixam na tecnologia móvel, onde estão os PDAs (*Personal Digital Assistant*), Pocket PCs e Smartphones [MOBILEZONE, 2008].

Existem basicamente três plataformas de desenvolvimento para aplicativos móveis: C++, Brew e Java, sendo este último o mais utilizado atualmente por questões de portabilidade, suporte e facilidade de programação. Há outras linguagens de programação para celulares de cunho específico para jogos, como é caso da plataforma ExEn desenvolvida pela empresa *In-Fusion*, utilizada em larga escala pelas operadoras e desenvolvedores europeus e chineses. Nesse nicho, há também a plataforma Mophun desenvolvida pela empresa *Synergenix*, considerada uma das mais avançadas e completa solução de

desenvolvimento de jogos para dispositivos móveis, construída ao longo dos anos [Nogueira *et al*, 2005].

Complementando o grupo de plataformas de programação para celulares, podem ser citadas algumas linguagens desenvolvidas para tecnologias específicas, como é o caso do Phyton for S60. Ela foi criada exclusivamente para uma serie de celulares compatíveis com o sistema operacional Symbian S60 e a plataforma .Net da empresa Microsoft, que permite o desenvolvimento de aplicativos para dispositivos com o sistema operacional Windows Mobile [Haddad, 2008; Caraciolo, 2008].

2.2. Plataforma de Desenvolvimento Java

Segundo Deitel (2001), uma das contribuições importantes da revolução dos microcomputadores até esta data é o fato dela ter tornado possível o desenvolvimento de computadores pessoais que logo puderam chegar a milhares em todo mundo. Os computadores pessoais tiveram um profundo impacto sobre as pessoas e a maneira como as organizações conduzem e gerenciam seus negócios.

Reconhecendo isso, a Sun Microsystems financiou a pesquisa corporativa interna com o codinome Green em 1991. O projeto resultou no desenvolvimento de uma linguagem baseada em C e C ++ que seu criador, James Gosling, chamou de Oak (carvalho), em homenagem à paisagem vista da janela do seu escritório na Sun. Descobriu-se depois que existia uma linguagem de programação com esse nome. Então, ela foi renomeada para Java, marca de um café que tomavam na discussão do novo nome [Deitel, Deitel, 2001].

Java é uma linguagem de paradigma orientada a objeto que trabalha exclusivamente com manipulação de objetos. Além de ser usada para fazer programas pessoais de diversos ramos de empresas e negócios, Java é empregada para criar páginas Web com conteúdo dinâmico e interativo, desenvolver aplicativos corporativos de larga escala, aprimorar a funcionalidade de servidores e fornecer aplicativos para dispositivos ao consumidor final (como telefones celulares, *paggers*, PDAs) [Horstmann, Cornell, 2001].

A plataforma Java possui três núcleos diferenciados de desenvolvimento (Figura 2-1):

- J2SE (*Java Standard Edition*) – Contém as classes principais da plataforma Java, sendo também chamado de *Core Java Platform*. Desenvolvido especialmente para aplicativos *desktop*;
- J2EE (*Java Enterprise Edition*) – É um conjunto de tecnologias que fornecem APIs⁴ (*Application Programming Interface*) e um ambiente para desenvolvimento e execução de aplicações corporativas. Específico para desenvolvimento de servidores corporativos;
- J2ME (*Java Micro Edition*) – Padrão Java para o desenvolvimento de *software* de sistemas embarcados e dispositivos móveis.

2.2.1. Edição Java J2SE

⁴ API é um conjunto de rotinas e padrões estabelecidos por um software para utilização de sua funcionalidade.

Este é o *suite*⁵ da plataforma Java específica para desenvolvimento de aplicações para computadores pessoais, *notebooks* e arquiteturas com poder de processamento e memória consideráveis, sendo assim a edição Java mais utilizada. Existe uma quantidade grande de APIs que acompanham esta versão para desempenhar as mais diversas funções, gerando mais facilidade de segurança para o desenvolvedor.



Figura 2-1 – Divisão da Plataforma Java (Fonte: Andreão, 2008)

Java 2 Standard Edition possui duas divisões [SUN, 2008d]:

- JDK (*Java Development Kit*) ou SDK (*Standard Development Kit*) é um conjunto de APIs construídas especialmente para o

⁵ Neste caso específico, *suite* significa uma plataforma de desenvolvimento de software, contendo APIs e funcionalidade completas.

desenvolvimento de aplicações, possuindo ferramentas aprimoradas para tal tarefa;

- JRE (*Java Runtime Edition*) é uma versão mais compacta que a anterior, pois possui funcionalidade apenas para preparar o ambiente computacional para a execução do aplicativo Java, sendo utilizada apenas para executar aplicativos construídos com a SDK.

A Figura 2-2 exhibe minuciosamente a complexidade da arquitetura J2SE, demonstrando os recursos disponíveis, bem como a divisão desta edição Java. J2SE possui diversas características importantes. A primeira delas é a portabilidade, inerente a plataforma Java. Assim, seus aplicativos tornam-se independentes do sistema operacional utilizado, característica esta, interessante para o desenvolvimento de sistemas para *desktop*, foco da edição J2SE. Pode-se ainda destacar as APIs específicas nela contidas, como a JDBC (*Java Database Connectivity*) que atua exclusivamente no tratamento e no gerenciamento da conexão com banco de dados, sendo portátil para qualquer banco de dados relacional, a API específica para programação com modelagem CORBA⁶ (*Common Object Request Broker Architecture*), além do desenvolvimento Java Security, que busca implementar a segurança do sistema com controle de permissões e autenticações para uso [Deitel, Deitel, 2001; SUN, 2008d].

2.2.2. Edição Java J2EE

Para o desenvolvimento de aplicações corporativas e/ou para Internet, normalmente é necessário o uso de recursos avançados como

⁶ Padrão para comunicação entre objetos distribuídos. Provê formas para interligar objetos desenvolvidos em linguagens de programação diferentes.

objeto distribuídos, mapeamento objeto-relacional para persistência de dados, controle de transação, manipulação XML (*Extensible Markup Language*), *Web Services*⁷, entre outros [Hall, 2005]. A edição J2EE foi exatamente construída para este ramo do desenvolvimento de *software*, garantindo várias facilidades para o desenvolvedor. As aplicações produzidas nesta edição são chamadas de *Servlets* e só podem ser executadas em máquinas com servidor Web configuradas que suportem a tecnologia, como é o caso do servidor Web *TomCat* da fundação Apache [Hall, 2005].

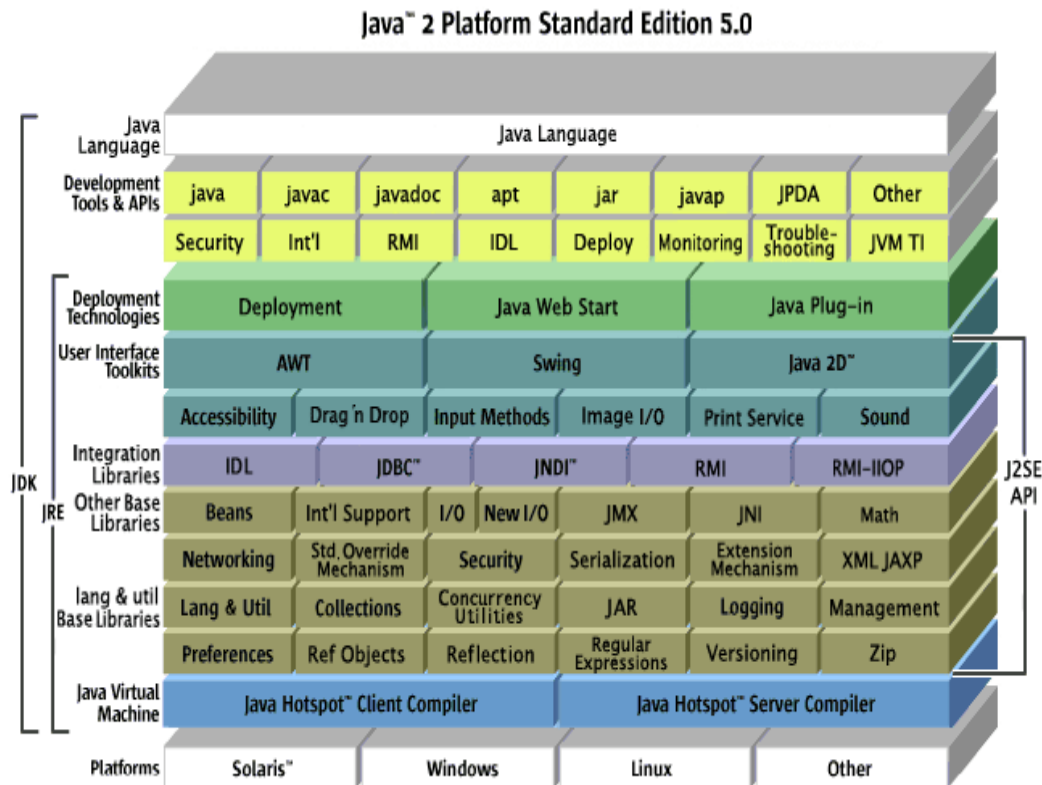


Figura 2-2 – Divisão da Plataforma Java J2SE (Fonte: SUN, 2008d)

⁷ Componente de software que troca dados com outra aplicação em uma rede.

Com esta edição, grande parte dos recursos não precisam ser implementados pelo programador. A infra-estrutura para gerenciar os aspectos complexos fica a cargo de um servidor de aplicações ou de um servidor *Web*. Qualquer componente do J2EE precisa ser executado dentro de um servidor, portanto, existe uma inversão de controle, na qual o desenvolvedor deve usar uma API padronizada e implementar a funcionalidade que será invocada pelo servidor [SUN, 2008a].

Para exemplificar a infra-estrutura descrita por esta plataforma, podem-se citar as atividades exercidas pelo servidor, descentralizando responsabilidades do desenvolvedor, como o gerenciamento da escalabilidade da aplicação, onde o servidor é responsável pela quantidade de objetos e quando estes devem ser instanciados. Isso permite que o desenvolvedor se preocupe somente com o domínio do problema, ou seja, a implementação das regras de negócio da aplicação. O uso das APIs de forma correta possibilita o desenvolvimento de aplicações escaláveis e fáceis de integrar com outros sistemas e componentes reutilizáveis [SUN, 2008a; Hall, 2005].

Os serviços da plataforma J2EE destinam-se especialmente a suprir as necessidades de aplicações empresariais distribuídas, com flexibilidade no controle das regras de negócio e na disponibilização de acesso para os diferentes tipos de dispositivos clientes – aplicativos *desktop*, *browsers*, dispositivos móveis [SUN, 2008e].

As aplicações J2EE são comumente compostas de uma camada cliente, que faz a interface com o usuário, uma camada intermediária, que

implementam a parte funcional do sistema, provendo serviço ao cliente, e a camada servidora, chamada de *Enterprise Information System* ou EIS, formada por componentes diversos instalados em *containers* e em bancos de dados [SUN, 2008e].

Os *containers* são divididos em dois grupos:

- Web Container: Destinado a executar componentes Web tais como: HTML, *Servlet*, JSP (*JavaServer Pages*). Implementa funções para aplicações Web, tendo como principal cliente o *browser*;
- EJB Container: Provê a infra-estrutura para execução de componentes distribuídos. Usando o protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*), ele gerencia a comunicação entre objetos, podendo ser integrando a outras tecnologias de computação distribuídas como RMI⁸ (*Remote Method Invocation*) e CORBA [SUN, 2008e].

A Figura 2-3 apresenta a arquitetura de funcionamento da plataforma J2EE.

2.2.3. Edição Java J2ME

O J2ME é uma versão reduzida da plataforma Java, que permite a criação de aplicativos para dispositivos móveis com as diversas vantagens da plataforma Java, como portabilidade, robustez, orientação a objetos e heranças. Com o J2ME, é possível que dispositivos móveis, com poucos recursos de memória, tela e processamento, possam fazer *download* de

⁸ Interface de programação que permite a execução de chamadas remotas de procedimentos em máquinas distribuídas.

aplicações Java, semelhante à tecnologia *Java Applets*. Apesar das semelhanças, um *Java Applets* utiliza API J2SE, enquanto que um aplicativo J2ME utiliza a API J2ME [SUN, 2008c].

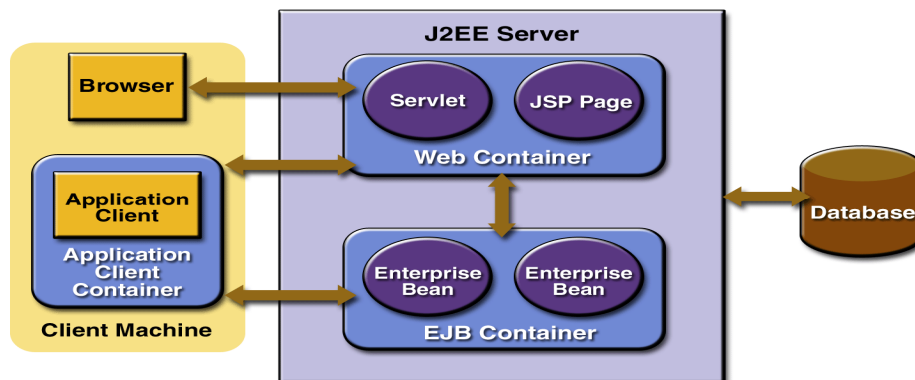


Figura 2-3 – Ambiente J2EE (Fonte: SUN, 2008a)

J2ME é mais utilizada no desenvolvimento de aplicativos móveis. Os *MIDlets* são executados no próprio aparelho e, para isso, necessitam da memória livre e o poder de processamento do dispositivo. A idéia inicial baseava-se no fato dos aplicativos Java serem compatíveis com os tipos de plataforma e aparelhos, como prega um dos princípios do Java. Porém, como cada fabricante disponibiliza uma API com os recursos dos aparelhos celulares, sendo criadas particularidades em cada aparelho, minimizando a portabilidade de J2ME [MOBILEZONE, 2008].

A plataforma J2ME consiste basicamente de dois módulos, Configuração e Perfil, organizados como ilustrado na Figura 2-4. O módulo de configuração define as características similares de *hardware* de um grupo de dispositivos, uma configuração mínima requerida. Como exemplo, pode-se citar as exigências de quantidade de memória e

processamento para um grupo de aparelhos [IMB, 2008]. O módulo de perfil é responsável por designar famílias de aparelhos por características de *software* e recursos, por exemplo, famílias que suportam as mesmas APIs. Este módulo garante a interoperabilidade entre uma família de dispositivos [IMB, 2008].

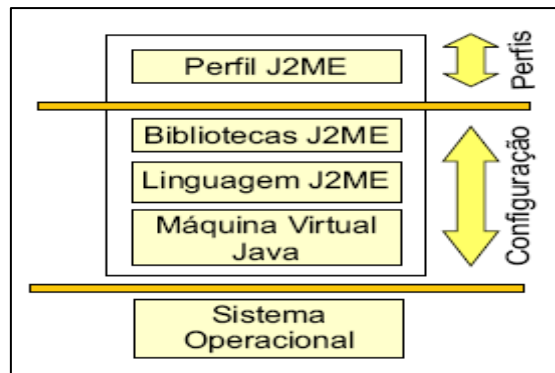


Figura 2-4 – Arquitetura da Plataforma J2ME (Fonte: Nogueira *et al*, 2005)

Além da interoperabilidade entre dispositivos da mesma família, existe outra vantagem da plataforma, a distribuição gratuita, em que cada desenvolvedor pode disponibilizar seus produtos livremente, sem a necessidade de autorizações ou certificados por parte da empresa criadora. Apesar dessa plataforma possibilitar independência ao desenvolvedor, as prevenções contra pirataria são mais frágeis, sendo o desenvolvedor o responsável pela publicidade e pela distribuição do próprio aplicativo [IBM, 2008].

Outra vantagem de J2ME é em relação à quantidade de dispositivos que a suportam e a compatibilidade existente entre os modelos móveis

que comportam a linguagem, sendo independente de sistema operacional dos dispositivos (Nogueira *et al*, 2005).

A Figura 2-5 ilustra a arquitetura de execução, contendo a Máquina Virtual, as Bibliotecas e os Perfis. A biblioteca de configuração CLDC (*Connected Limited Device Configuration*) e o perfil MPID (*Mobile Information Device Profile*) são utilizados para a programação de celulares dentro do J2ME [Nogueira *et al*, 2005].

2.3. Considerações Finais

Java é uma linguagem de programação atual e robusta que busca atuar com excelência em diversos setores do mercado, utilizando suas versões de plataformas, para abranger a cada área de forma específica. Dentre estas plataformas, pode-se destacar J2ME, que atua no mercado de desenvolvimento de *software* para pequenos dispositivos (dispositivos móveis), possibilitando a criação de aplicações sofisticadas, desde jogos *on-line* até programas de uso corporativo.

Outra vantagem de Java é a independência de sistema operacional, sendo possível que o mesmo *software* escrito na linguagem de programação Java seja executado em dispositivos com sistemas operacionais Symbian, Windows CE, Pocket PC, Palm OS entre outros. Para isso, estes sistemas operacionais devem ter a JVM correspondente.

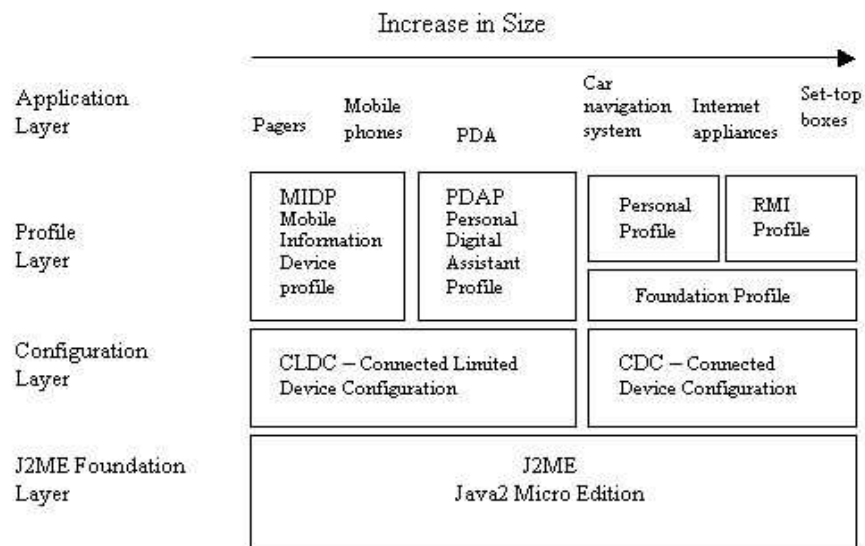


Figura 2-5 - Arquitetura de Execução da Plataforma J2ME (Fonte: IBM, 2008)

3. Tecnologia Móvel

3.1. Considerações Iniciais

O objetivo deste capítulo é abordar os sistemas operacionais utilizados nos telefones celulares, as tecnologias para a conexão com as operadoras e os recursos usados para transmissão de dados (conexão com a Internet).

A comunicação móvel sofre hoje uma mudança de utilização. Aos poucos, a transferência de voz, motivo para o qual foi criado, vem se tornando apenas uma funcionalidade básica nos celulares entre tantas outras possibilidades e variedade de serviços. Neste cenário, serviços digitais personalizados oferecem novas opções e oportunidades para as operadoras, introduzindo a transferência de dados Multimídia (texto, áudio, vídeo) e Internet como parte dos serviços, e criando a cada dia novas tecnologias para melhoria da comunicação móvel [Luca, 2009].

Estas evoluções tecnológicas sofridas pela comunicação tendem a ter grande importância na sociedade, tornando a comunicação mais pessoal e expressiva, elevando a dimensão a um panorama de alta qualidade e versatilidade. Para isso, a tecnologia móvel conta com padrão de mensagens MMS (*Multimedia Messaging Service*), com a possibilidade de transmitir texto, imagens e clipes de áudio usando celulares e *e-mails*, bem como a tecnologia *Streaming*, possibilitando a transferência de dados em tempo real como rádios *on-line* e canais de TV ao vivo em dispositivos móveis [Luca, 2009].

3.2. Sistema Operacional Móvel

O sistema operacional de dispositivos móveis exerce a mesma função de um sistema operacional comum, ou seja, ele gerencia os recursos do aparelho como memória livre, o armazenamento de dados e o ambiente gráfico, implementa o escalonamento de processos controlando aplicativos e gerencia os dispositivos de *hardware* como teclado, tela, *bluetooth*, infra-vermelho e demais dispositivos de conexão [Ferrari, 2009].

Atualmente, existem vários sistemas operacionais para dispositivos móveis, por exemplo, PalmOS, Pocket PC, Brew, Symbian e Windows Mobile. Os dois últimos são os mais utilizados em telefones celulares e *smartphones*. Estes sistemas possuem diversas classificações, criando versões para atender a mais grupos de modelos aparelhos com configurações e semelhantes [Ferrari, 2009].

Neste nicho de mercado, existem sistemas com pouca utilização como é o caso do LiMo⁹, um sistema para dispositivos portáteis baseado no sistema operacional Linux, que está em execução em modelos de fabricação das empresas Motorola e Panasonic. Outro exemplo é o sistema Android¹⁰ da empresa Google, que busca disponibilizar um sistema multitarefa, livre, com diversos aplicativos interessantes [LINUXDEVICES, 2009]. Para um aprofundamento melhor no tema é sugerida a leitura de Ferrari (2009).

⁹ <http://www.limofoundation.org/>

¹⁰ <http://code.google.com/intl/pt-BR/android/>

3.2.1. Sistema Operacional Mobile Symbian

O Symbian¹¹ é um sistema operacional multitarefa desenvolvido com base no sistema EPOC 32 que equipava os dispositivos da empresa Psion¹² por volta de 1998, no início da era de equipamentos portáteis. A empresa Symbian era constituída pelos próprios fabricantes de celulares, como a própria Psion, Nokia, Motorola e Ericsson, detentores da grande fatia do mercado de sistemas operacionais de aparelhos portáteis. Mas, com o crescimento do uso de aparelhos celulares, a Symbian iniciou o desenvolvimento de sistemas para outras empresas que atuavam no mesmo ramo [PALMLAND, 2009a].

Apesar da incompatibilidade com outros sistemas de dispositivos móveis, como PalmOS e Pocket PC, o Symbian é utilizado na maioria dos aparelhos celulares modernos, pelo fato ser um sistema totalmente modular e incremental. Cada empresa pode criar ou modificar sua própria interface, criando uma nova aparência do sistema. Assim, o Symbian comporta módulos de telas monocromáticas até módulos complexos e robustos, como os encontrados nos *palms* e *smartphones* [SYMBIANBRASIL, 2009].

O sistema Symbian construído usando padrões abertos, com alta qualidade de suporte por meio de uma rede global de competência. O sistema trabalha com tecnologias atuais como J2ME, *bluetooth*, MMS e IPv6. A plataforma mais conhecida do Symbian é a S60 utilizada principalmente pela Nokia [SYMBIANBRASIL, 2009].

¹¹ <http://www.symbian.com/index.asp>

¹² <http://www.psionteklogix.com/>

Atualmente, o sistema tem participação de praticamente todos os grandes fabricantes de aparelhos celulares como a Nokia, a Ericsson, a Sony-Ericsson, a Panasonic, a Siemens e a Samsung [PALMLAND, 2009a].

3.2.2. Sistema Operacional Windows CE (Mobile)

Este sistema operacional é uma versão compacta do sistema operacional tradicional Windows, para execução em aparelhos portáteis. Ele opera principalmente em dispositivos do tipo PDAs, sendo usado com menor frequência em aparelhos celulares. Sua execução é compatível com sistema Pocket PC, tornando-se possível criar aplicativos que executam em ambas as plataformas, com algumas restrições para aplicativos com interface gráfica complexa [PALMLAND, 2009b].

O Windows Mobile comporta-se como um sistema de recursos multimídia e corporativo de uso fácil e intuitivo, mas, é considerado um sistema instável em comparação com os demais sistemas operacionais de dispositivos móveis como Symbian e PalmOS [PALMLAND, 2009b].

Existem diversos aplicativos para o desenvolvimento de aplicações nesta plataforma, contendo recursos complexos e ótimas soluções de mobilidade. Além disso, o desenvolvedor pode contar com a padronização e a interoperabilidade entre as versões deste sistema, trazendo aos aplicativos construídos uma vida útil bastante longa. A principal desvantagem está no fato deste ser um sistema proprietário, de código fechado e com custo elevado, fatores que até hoje não tem

influenciado na sua popularidade, existindo inúmeras marcas e modelos de dispositivos moveis operados por ele [Ferrari, 2009].

O Windows Mobile tem previsões de um futuro promissor com as versões 6 e 7, pelas ótimas ferramentas que o sistema contém e pela presença eminente da Microsoft no mercado global. No Brasil, a comercialização de dispositivos móveis equipados com Windows Mobile ainda é restrita a aparelhos da classe dos PDAs e Pocket PCs, assumindo pequena parcela na classe de celulares e *smartphones* [Ferrari, 2009].

3.3. Tecnologias para Celular

Os modelos de conexão via celular sofreram evoluções de tecnologia nos últimos anos, pela necessidade crescente de dispositivos móveis e sem fio de alta qualidade. Atualmente, as operadoras oferecem serviços que operam em boas velocidades de conexão para o envio dados multimídia usando a tecnologia 3G (3º geração). As gerações e tecnologias de celular são abordadas de forma breve nas seções a seguir, pois estes não estão no enfoque principal deste trabalho.

3.3.1. 1º Geração

Representada pela tecnologia AMPS (*Advanced Mobile Phone Service*), foi desenvolvida em laboratórios nos anos 80, com a operação analógica de voz na frequência de faixa dos 800 MHz.

Este sistema começou a ser desativado a partir do ano de 2008 inicialmente pelos EUA e tende a ser extinto em outros países nos próximos anos, buscado utilizar o seu espectro para tecnologias mais

novas, como a extensão de espectro para o 3G com HSDPA (*High-Speed Downlink Packet Access*) [Martinez, 2008].

3.3.2. 2º Geração

Com a necessidade de maior capacidade e qualidade de transmissão de voz digital, atingindo maiores frequências para comunicação de dados, e de implementações de segurança como criptografia, surgiram as tecnologias de segunda geração. Os dois maiores grupos desta geração são as tecnologias CDMA (*Code Division Multiple Access*) e GSM, que deram suporte para a evolução para a próxima geração padrões de celulares [TELECO, 2008a]. As tecnologias inclusas na segunda geração são [Ayres, 2008; TELECO, 2008a]:

- TDMA (*Time Division Multiple Access*). Esta tecnologia é configurada com uma divisão de tempo para cada usuário fazer seu acesso, ou seja, acesso múltiplo por divisão do tempo. No tocante a transmissão de dados, esta tecnologia não diferencia voz e dados. Dessa forma, uma conexão com a Internet (dados) é executada como uma ligação de voz. Normalmente, este serviço é caro e lento. Somente para estabelecer a conexão leva-se 8 segundos e, a transferência de dados dá-se a 9.600 bps (*bits per second*);
- CDMA (*Code Division Multiple Access*). Esta tecnologia é configurada com uma divisão de códigos para cada usuário. Cada usuário faz a conexão ao mesmo tempo utilizando um pouco do recurso, isto é, acesso múltiplo por divisão de códigos. Nessa tecnologia, há separação entre voz e dados, usando uma conexão 100% ativa, ou seja, não existe *delay* de conexão, toda informação

requisitada vem diretamente pela conexão aberta. Com esta tecnologia, a taxa de transferência de dados pode chegar a 256 Kbps (*kilobits per second*), usando o padrão CDMA 1xRTT;

- GSM (*Global System for Communication*). É configurado por divisão de tempo como a TDMA, operando uma separação entre voz e dados com conexão 100% ativa, como a CMDA, porém utiliza da tecnologia GPRS ou EDGE para transmissão de dados. Normalmente, os aparelhos desta tecnologia contêm um *chip* com as informações que podem ser transferidas facilmente para qualquer aparelho celular pertencente à tecnologia.

3.3.2.1. Tecnologia de Transmissão de Dados GPRS

O GPRS é um serviço não baseado em voz que permite o envio e a recepção de informações por meio de uma rede telefônica móvel. Nesta tecnologia, os pacotes de dados são enviados usando múltiplos *slots*¹³ de tempo, todavia não existe reserva. Os *slots* são alocados conforme a demanda dos pacotes enviados ou recebidos. Desta forma, consegue-se um serviço de dados com conexão permanente (*always on*) sem a necessidade de reservar permanentemente *slots* de tempo para o transporte de dados [TELECO, 2008a].

Esta tecnologia possui taxa de transporte de dados máxima de 26 a 40 kbit/s, podendo chegar, em teoria, a 171,2 kbit/s, com uma conexão de dados sem necessidade de estabelecer um circuito telefônico, permitindo

¹³ Espaços de tempo determinados.

a cobrança por uso e não por tempo de conexão, em um sistema sempre disponível para o usuário (*always on*) [TELECO, 2008a; Cruz, 2008].

3.3.2.2. Tecnologia de Transmissão de Dados EDGE

O EDGE é uma tecnologia de transmissão de dados de alta velocidade que, teoricamente, pode atingir até 384 kbps. Em caso de taxa de transferência média, entre 110 e 120 kbps, ele pode operacionalizar bem serviços avançados de dados multimídia (áudio e vídeo) [Ayres, 2008].

Como o GPRS, esta tecnologia é baseada em um sistema de pacotes de conexão permanente de transferência dados, oferecendo melhoria de até quatro vezes na velocidade e no *throughput* atingido pela anterior. Por este motivo, foi adotada uma segunda nomenclatura para este padrão E-GPRS (*Enhanced GPRS*) ou GPRS ampliada [TELECO, 2008a].

3.3.3. 3º Geração

Uma das vantagens da tecnologia 3G é a maior capacidade, podendo suportar uma quantidade maior de usuários com serviços de voz e dados. Esta característica se torna muito interessante para as áreas urbanas, que aglomeram grande quantidade de aparelhos celulares [TELECO, 2008b].

A tecnologia utiliza um espectro de bandas diferenciado das demais gerações, atingindo conexões com velocidade de até 384 kbit/s em sistemas móveis e até 2 Mbit/s para sistemas estacionários [TELECO,

2008b]. As principais tecnologias desta geração são [TECNOLOGIA3G, 2009; TELECO, 2009b]:

- WCDMA (*Wideband Code Division Multiple Access*). Tecnologia digital de comunicação de banda larga, que possibilita o aumento da capacidade da rede móvel, com a redução de custo para serviços de voz e dados em comparação com as tecnologias anteriores. Uma vantagem é o baixo custo de migração para as operadoras do sistema 2G. Este padrão pode atingir velocidades de dados até de 2 Mbit/s, com novos serviços compostos de diferentes tipo de mídia como vídeo, áudio digital, imagens e animações coloridas. A tecnologia foi nomeada pelo ITU (*International Telecommunication Union*) com *IMT-2000 direct spread*;
- HSDPA (*High Speed Download Packet Access*). Tecnologia criada a partir de uma evolução rede WCDMA, proporcionando velocidade até quatro vezes maior que esta, de forma a atingir, na teoria, a marca de 10 Mbit/s no enlace, em bandas de 5 MHz para transmissão. Esta tecnologia trabalha sobre um novo canal físico transporte de dados (HS-DSCH – *High Speed Downlink Shared channel*), com diferenças do WCDMA como: reserva rápida de pacotes, modulação e codificação com adaptação ao meio e uso de dois modos de modulação (QPSK e 16-QAM), conseguindo operacionalizar qualidades de serviços bem superiores aos outros padrões da tecnologia 3G.

3.4. Considerações Finais

A evolução da tecnologia de aparelhos celulares gerou avanços no setor de telecomunicação. Inicialmente, este avanço atingiu a tecnologia digital, criando a possibilidade de tráfego de dados entre aparelhos móveis, usando conexão com a Internet e resultando em uma verdadeira revolução na cultura do uso destes aparelhos. Para tanto, as diversas novas funções agregaram aos dispositivos móveis facilidades cada vez mais interessantes e úteis, respaldado pela qualidade de serviço prestada pela tecnologia 3G, em evidência atualmente.

Introduzidos neste contexto de constantes evoluções, os sistemas operacionais de aparelhos celulares também implementaram novas funções atraentes com o foco nos usuários, como por exemplo os aparelhos celulares com *touch screen* e GPS (*Global Positioning System*).

4. Trabalhos Relacionados

4.1. Considerações Iniciais

Vários trabalhos relacionados com tecnologia móvel e *wireless* têm sido desenvolvidos ultimamente, por tratar de uma ciência nova, emergente e com um grande mercado potencial.

Pesquisas sobre novas tecnologias para conexão de aparelhos celulares, buscando transferência mais eficiente de voz e, principalmente, de dados, trabalhos sobre o desenvolvimento de jogos para dispositivos móveis, projetos de *software* para automação e gestão de tarefas diversas com mobilidade e disponibilidade e pesquisas sobre acesso e segurança

em redes sem fio são exemplos de ramos de pesquisa que buscam responder às necessidades de tecnologia acessível e segura exigidas pelo mundo moderno.

4.2. Desenvolvimento de Software Móvel

Os trabalhos existentes relacionados com o meio móvel, em particular, o desenvolvimento de *software* para celulares, abrange aspectos diferenciados, como é o caso do trabalho de Renato Miranda e Valeska Marcondes do Instituto Nacional de Telecomunicações (Inatel-MG) que buscou demonstrar a aplicação da tecnologia Java para dispositivos portáteis, focando a construção de um manual de desenvolvimento J2ME, o qual mostra as especificações da linguagem, arquitetura e ferramentas para o desenvolvimento, além de apresentar exemplo de codificação da implementação de aplicações com esta tecnologia [Miranda, Marcondes, 2009].

Outro trabalho relacionado é o artigo de Wallace Nogueira e Emerson Cavalcante da Universidade Federal de Alagoas e Hyggo Almeida da Universidade Federal de Campina Grande. Ambos trataram sobre as linguagens de desenvolvimento de jogos para celulares abrangendo a plataforma Java na edição J2ME, fazendo um comparativo entre as linguagens vigentes, mostrando vantagens e desvantagens na aplicação do desenvolvimento de jogos para dispositivos móveis em particular os celulares [Nogueira *et al*, 2005].

Existem trabalhos relacionados no tocante ao desenvolvimento de ferramentas para criação de aplicativos para celulares, como se percebe

no projeto final de curso dos alunos Priscila Lugon e Thiago Rossato da Universidade Federal de Santa Catarina. Este trabalho sugeriu a construção de um *framework* de persistência de objetos para aplicações desenvolvidas em J2ME. A ferramenta implementa a camada de persistência do sistema, sendo responsável por encapsular a lógica necessária para salvar e recuperar objetos em um repositório. Desta forma, é possível eliminar parte da complexidade do desenvolvimento de aplicações no ambiente de dispositivos móveis. [Lugon, Rossato, 2009].

4.3. Redes Wireless e Móveis

Na área de redes *wireless*, existe grande tendência aos temas que envolvem a segurança dos dados trafegados nas redes sem fio e as possibilidades de intervenção alheia nestas informações.

Como exemplo de trabalhos com esse foco, tem-se o artigo de um grupo de engenheiros da Faculdade Federal do Rio de Janeiro que apresenta um estudo dos processos de segurança em Sistemas Móveis de Terceira Geração, buscando analisar desde os requisitos de confiabilidade do sistema até a implementação de técnicas e protocolos para especificar a segurança nesta rede [Ribeiro *et al*, 2009].

Ainda neste nicho, tem-se o artigo do analista de sistema Nelson Murilo, que explanou sobre os aspectos de segurança das redes *wireless* corporativas em um estudo de caso prático, apontando os principais riscos existentes nesta tecnologia. Pode-se citar também, o artigo de Marco Aurélio, que focou seu trabalho nas principais formas de ataque em redes sem fio, demonstrando pontos de vulnerabilidades, por meio da

exploração de ferramentas para a identificação destes ataques [Aurélio, 2009; Murilo, 2009].

4.4. Automação de Tarefas Através de Aplicações Móveis

Pela necessidade de automação e de monitoramento constante de processos nas diversas áreas do conhecimento, existe hoje, grande tendência de construção de trabalhos que envolvem os dispositivos móveis na busca da resolução de problemas que precisam de mobilidade e de disponibilidade na coleta e/ou análise de dados. Para isso, são implementadas aplicação diversas para celulares e outros dispositivos, no intuito de realizar esta automação de forma descentralizada e remota, contando a vantagem da comunicação sem fios.

Como exemplo disto, tem-se o projeto da Faculdade de Engenharia da PUCRS, que utilizou a computação móvel, PDAs e celulares, na melhoria das rotinas de manutenção de redes elétricas. Objetivo principal foi fazer uso de um *software* móvel para auxiliar no planejamento da manutenção de sistemas de distribuição de energia elétrica, além da coleta e da validação de dados de eventos não programados destas redes, em tempo real [Pretto *et al*, 2009].

Ainda com este foco, pode-se citar o artigo de Ramon Rabello da Universidade da Amazônia, que implementou, usando J2ME, uma solução de *software* para o meio acadêmico. O objetivo era facilitar e automatizar a tarefa cotidiana do docente de atualizar as frequências e as notas dos alunos, permitindo que esta tarefa seja realizada remotamente

de um dispositivo móvel para a base de dados da instituição de ensino [Rabello *et al*, 2009].

Comprovando, também, a utilização de dispositivos móveis na geração de automação de tarefas, tem-se o trabalho de Doutorado de Paulo Salomão da Universidade Federal de São Paulo, que implementou um sistema para registros de prontuário médicos eletrônico para dispositivos móveis. O projeto buscou comprovar a necessidade de manter e de consultar, usando dispositivos portáteis, as informações de pacientes em casos de atendimento médico externo ao ambiente hospitalar ou via telefone, comprovando sua utilização com profissionais da área medica [Salomão, 2009].

4.5. Considerações Finais

Os modelos de trabalhos descritos nos tópicos anteriores mostram a atualidade e a necessidade de desenvolvimento ligado ao tema proposto. Tais artigos, apesar de se assemelhar ao presente trabalho e trabalhar com as mesmas plataformas e tecnologias de desenvolvimento, têm seu foco diferenciado, pois este projeto busca demonstrar a funcionalidade e a facilidade existentes na criação de aplicativos para celulares, utilizando da conexão com a internet efetuada por esses dispositivos, aplicada ao gerenciamento remoto de serviços básicos de servidores em rede Linux.

5. ReMoS – *Remote Mobile Server*

5.1. Considerações Iniciais

Este capítulo é dedicado à apresentação da ferramenta proposta neste trabalho que consiste em um *software* móvel para controle remoto de funções básicas de computadores servidores de rede intitulado de ReMoS. O ReMos apresenta algumas vantagens de uso sobre os sistemas convencionais de acesso remoto, principalmente em caso de necessidade urgente de acesso, tais como: i) alta disponibilidade; ii) facilidade e simplicidade de operação; iii) mobilidade para os usuários; iv) acesso seguro; e v) baixo custo.

Os requisitos implementados no ReMoS foram definidos mediante breve entrevista com gerentes de redes que lidam diretamente com servidores, análise de situações reais e alguns anos de experiência no gerenciamento de servidores de rede. Foram definidas para o escopo do ReMoS apenas as necessidades principais para a operacionalização básica do servidor, não implementando todas as funções, por causa de limitações de memória e de processamento dos aparelhos celulares. O ReMoS é constituído de dois módulos:

- Módulo Mobile – executa no celular. Este módulo possui interface com o usuário, de forma a modelar as requisições a serem enviadas ao servidor e interpretar as respostas recebidas para exibição ao usuário;
- Módulo Server – executa no servidor. Este módulo tem a função de receber as requisições do módulo Mobile, executar os comandos, processar respostas e enviá-las de volta ao dispositivo móvel para a

apresentação ao usuário. Este módulo tem a possibilidade de gerenciar mais de um cliente, controlando separadamente as requisições enviadas.

O Diagrama de Casos de Uso, o Diagrama de Atividades e o Diagrama de Implantação da UML são usados para apresentar parte da modelagem e da documentação do ReMoS.

5.2. Modelagem do ReMoS

5.2.1. Requisitos do ReMoS

As regras de negócio do ReMoS podem ser entendidas como seus requisitos funcionais. Elas consistem de funções que o aplicativo deve conter para satisfazer os principais motivos de sua existência. A Tabela 5-1 apresenta os requisitos funcionais do ReMoS.

Tabela 5-1 – Requisitos Funcionais do ReMoS

Requisitos	Descrição	Descrição do Fluxo
RF1	Uso do endereço IP da máquina servidora a ser conectada.	1) Ao entrar no sistema é requisitado o endereço do servidor de conexão. 2) A conexão é efetuada.
RF2	Uso de autenticação de conexão (<i>login</i> e senha) para disponibilizar o acesso.	1) Na tela inicial, é requisitado <i>login</i> e senha do usuário. 2) Os dados são validados no servidor.
RF3	Discriminação de erros de comunicação com servidor.	1) O software solicita comunicação com servidor. 2) Caso ocorra algum erro de comunicação durante o processo, é gerada mensagem de erro explicativa.
RF4	Reinicializar e desligar servidor.	1) Usuário escolhe a funcionalidade desligar/reiniciar. 3) Função é executada pelo servidor.
RF5	Controle básico de processos e serviços no servidor.	1) Usuário escolhe a funcionalidade desejada para controle de processos ou serviço.

		2) Servidor executa operação desejada. 3) Servidor retorna resultado ao cliente.
RF6	Execução de <i>script</i> pessoal.	1) Usuário entra com o nome do <i>script</i> pessoal a ser executado. 2) Servidor executa <i>script</i> desejado. 3) Servidor retorna resultado ao cliente.

Os requisitos não funcionais são entendidos como características operacionais do *software*, ou seja, restrições que se colocam sobre como o ReMoS deve realizar seus requisitos funcionais. Estes requisitos podem ser definidos em categorias como usabilidade, confiabilidade, desempenho e segurança. A Tabela 5-2 apresenta os requisitos não funcionais do ReMoS.

Tabela 5-2 – Requisitos Não Funcionais do ReMoS

Requisitos	Descrição	Descrição do Fluxo
RNF1	A ferramenta deve ser de fácil manuseio.	Usabilidade
RNF2	O retorno de informações ao usuário deve ser rápido.	Desempenho
RNF3	A discriminação de erros de operação deve ser efetiva (entrada de dados errados).	Segurança
RNF4	A segurança dos dados deve ser garantida.	Segurança
RNF5	Tráfego de dados entre cliente e servidor deve ser pequeno para menor custo de operações.	Operacional

5.2.2. Diagrama de Casos de Uso

Um caso de uso pode ser entendido como um documento que descreve a interação existente entre os atores e o ReMoS na execução dos processos. O ator é uma entidade externa, um usuário ou mesmo outro módulo do sistema que, de alguma maneira, participa do caso de uso. Assim, o Diagrama de Casos de Uso apresenta a relação entre cada ator e os casos de uso executados por ele. De modo geral, este diagrama fornece uma

forma para descrever a visão externa do ReMoS e suas interações com o mundo exterior, como mostram a Figura 5-1 e a Figura 5-2, módulo Mobile e módulo Server respectivamente.

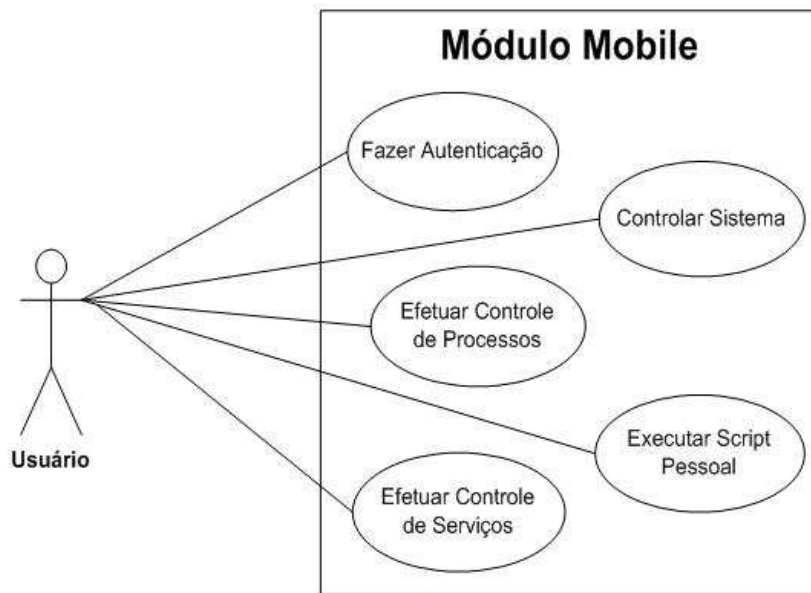


Figura 5-1 – Diagrama de Casos de Uso do Módulo Mobile

O ReMoS tem apenas dois atores: i) o usuário, que atua diretamente com o *software* no dispositivo móvel (módulo Mobile); e ii) módulo Mobile, no caso do módulo Server, que atua diretamente com servidor fazendo requisições. Apesar dos dois módulos conterem casos de usos semelhantes, os atores interagem de forma diferenciada com os casos de uso definidos, constituindo a funcionalidade de cada módulo.

No módulo Mobile, o ator usuário interage com os seguintes casos de uso:

- Fazer Autenticação. A autenticação é efetuada via *login* e senha do usuário pré-cadastrados no servidor;

- Efetuar Controle de Processos. O usuário pode escolher entre três funções: i) processos com maior utilização de processador; ii) processos com maior utilização de memória; e iii) matar processos pela numeração PID (*Process Identifier*);
- Efetuar Controle de Serviços. O usuário pode escolher entre duas funções: i) reiniciar o serviço de *proxy* (Squid¹⁴); e ii) recuperar a tabelar ARP (*Address Resolution Protocol*) do servidor no momento;
- Controlar Sistema. O usuário pode escolher entre duas funções: i) reiniciar servidor; e ii) desligar Servidor;
- Executar Script Pessoal. O usuário pode executar *scripts* diversos configurados por ele no servidor.

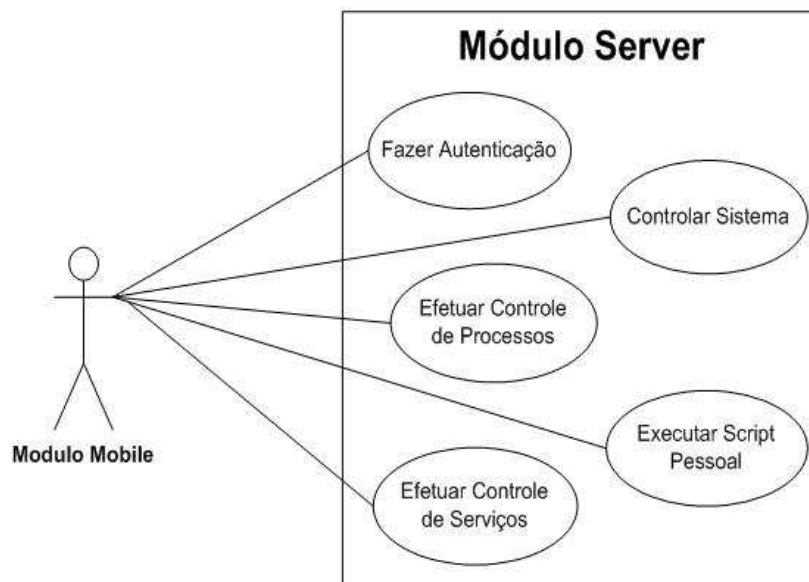


Figura 5-2 – Diagrama de Casos de Uso do Módulo Server

¹⁴ Software específico para serviço de *proxy* e *cache* para sistemas *like Unix* e *Windows*.

No Diagrama de Casos de Uso do módulo Server, o ator Módulo Mobile faz interação com os seguintes casos de uso:

- Fazer Autenticação. A validação do usuário é realizada pela base de dados do servidor, podendo ser habilitado vários usuários para o uso do ReMoS;
- Efetuar Controle de Processos. A funcionalidade solicitada é identificada, executando os comandos necessários para realizar a função. Em seguida, o resultado é enviado ao cliente;
- Efetuar Controle de Serviços. Segue a mesma linha de execução do caso de uso Efetuar Controle de Processos;
- Controlar Sistema. O ator recebe uma mensagem de encerramento da conexão, sendo esta mensagem apresentada após a realização das funções: reiniciar servidor e desligar servidor;
- Executar Script Pessoal. O *script* pessoal solicitado é identificado e executado após a solicitação do ator. As respostas geradas pelo *script* são enviadas ao ator.

5.2.3. Diagrama de Atividades

Usando o Diagrama de Atividades, pode-se mostrar o funcionamento do ReMoS de acordo com as transações de estados efetivadas durante sua execução. Com a definição dos estados em que o ReMoS pode se encontrar, são traçadas interligações, condicionais ou não, demonstrando as possibilidades de seu funcionamento. O Diagrama

de Atividades está descrito na Figura 5-3 e na Figura 5-4, módulo Mobile e módulo Server respectivamente.

O módulo Mobile possui oito funções principais. Elas estão divididas em dois grupos: i) funções diretas, estas funções permitem solicitação ao servidor diretamente após a escolha do usuário (Utilização Processador, Utilização Memória, Reiniciar Serviço *Squid*, Tabela ARP, Reiniciar Sistema e Desligar Sistema); e ii) funções com dados, estas funções dependem da entrada de uma informação do usuário para emitir requisições ao servidor (Matar Processo e Executar *Script* Pessoal).

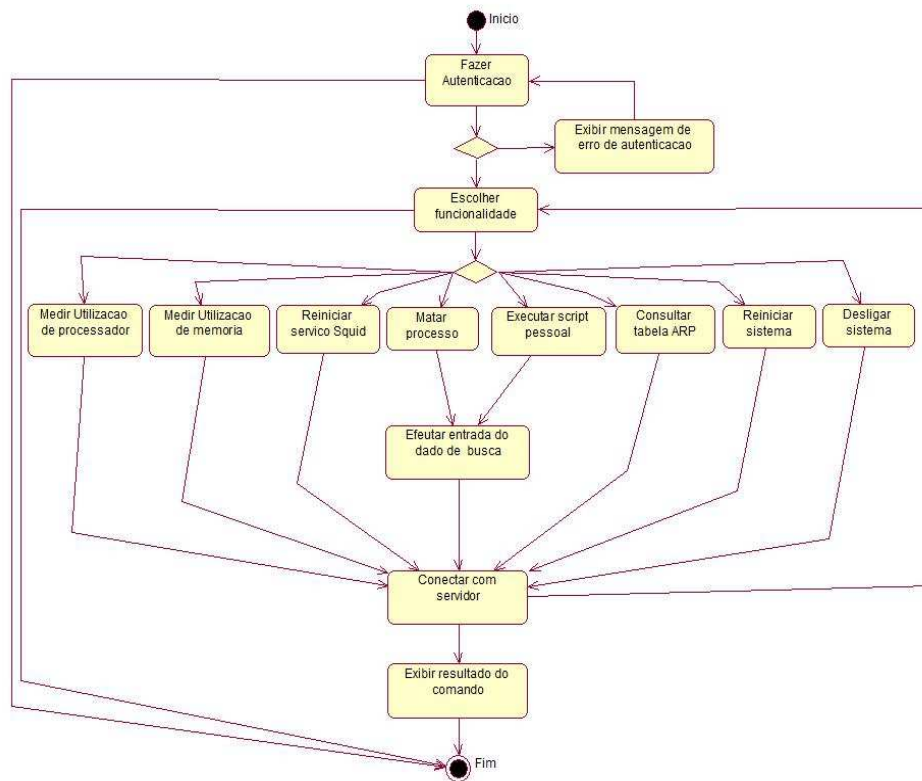


Figura 5-3 – Diagrama de Atividades do módulo Mobile

O módulo Server possui dois serviços definidos a partir das informações enviadas pelo módulo Mobile: i) autenticação de usuários, esta autenticação responde ao módulo Mobile sobre a possibilidade ou não da efetivação de *login* no servidor; e ii) execução de funções, esta consiste na identificação da função desejada e na execução do *script* pré-configurando correspondente.

5.2.4. Diagrama de Implantação

Este diagrama é utilizado especialmente na descrição de sistemas distribuídos. Ele permite apresentar a topologia de rede das máquinas do ReMoS, demonstrando onde cada componente executável (processos) é executado na rede. O sistema cliente/servidor é um tipo particular de sistema distribuído. O lado cliente interage com o exterior (usuários) e lado servidor realiza o processamento e o armazenamento de informações. A Figura 5-5 ilustra o Diagrama de Implantação do ReMoS.

5.3. Funcionamento do ReMoS

5.3.1. Módulo Mobile

Pelo fato deste módulo ter comunicação direta com os usuários, buscou-se utilizar uma interface amigável, fácil e de execução rápida para garantir a boa operacionalização dos usuários.

As funções do ReMoS abrangem as atividades básicas realizadas pelos gerentes de rede (usuários da aplicação), focando na agilidade de uso e no tráfego na rede de poucos dados. O cuidado para reduzir o tráfego na rede, além de gerar agilidade na operacionalização do ReMoS,

é de grande relevância para o barateamento de custos no uso do aplicativo, pois o serviço de conexão com a Internet, realizada pelo protocolo GPRS ou EDGE, é tarifada pela quantidade de dados trafegados durante a conexão. Assim, quanto menor for a transação de dados entre aplicativo e servidor, menor será o custo na execução das atividades do *software*.

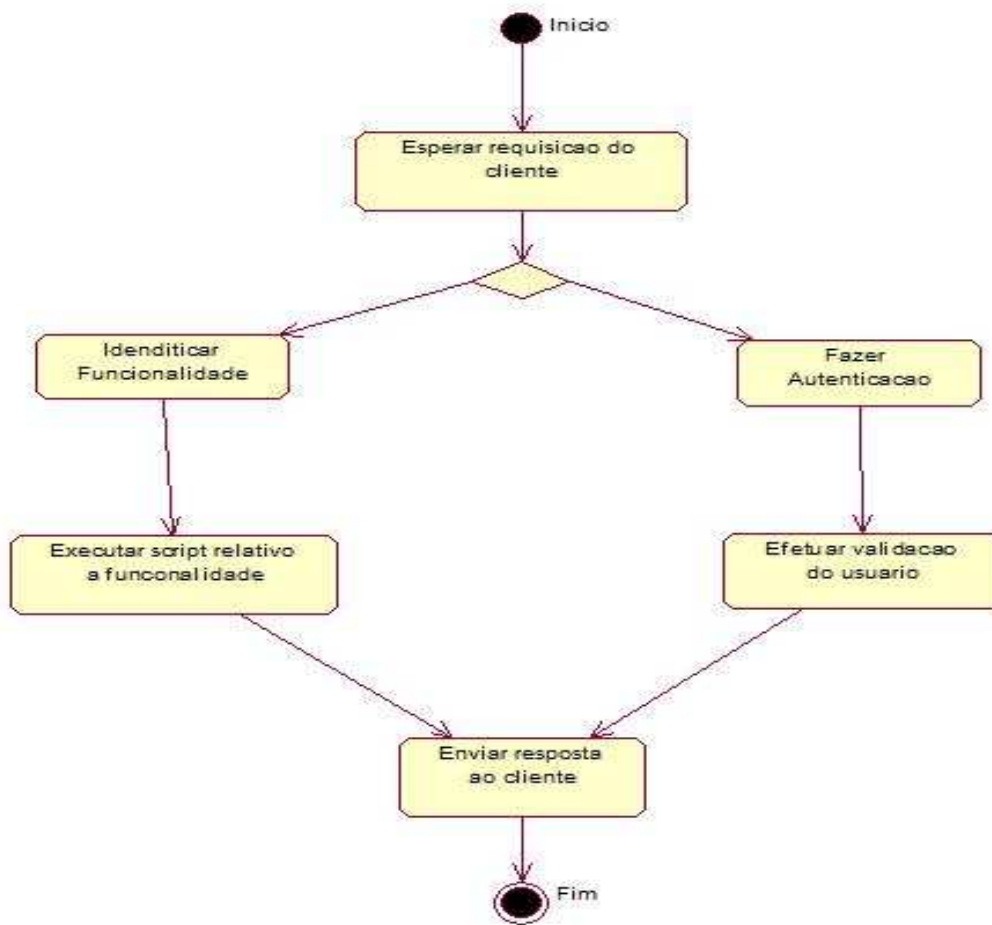


Figura 5-4 – Diagrama de Atividades do Módulo Server

O módulo cliente foi desenvolvido com a tecnologia J2ME, específico para aparelhos com configuração CLCD versão 2.0 e perfil MIDP 2.0 ou superior.

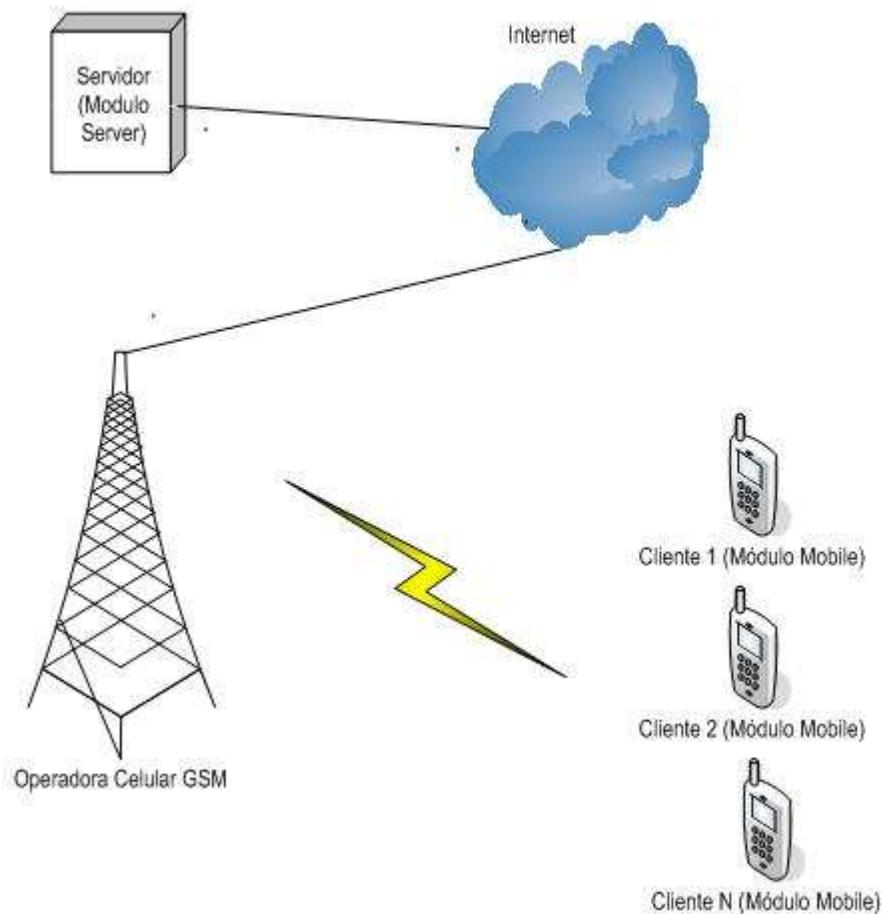


Figura 5-5 - Diagrama de Implantação do ReMoS

5.3.1.1. Acesso ao ReMoS

Inicialmente, deve ser informado o endereço IP do servidor para conexão. Com isto, é possível efetuar o controle de vários servidores independentes com um único aplicativo. No próximo passo, é ativado o

nível de segurança, com o controle de acesso ao *software*, solicitando *login* e senha ao usuário para autenticação (Figura 5-6).

A validação do usuário é efetuada pelo servidor, que reporta ao cliente a possibilidade de acesso às funções do ReMoS. Após obter acesso ao ReMoS, é exibido o *menu* de serviços, onde o usuário pode escolher uma das funções organizadas em quatro categorias (Figura 5-7): i) Processos; ii) Serviços; iii) Sistema; e iv) Pessoal.

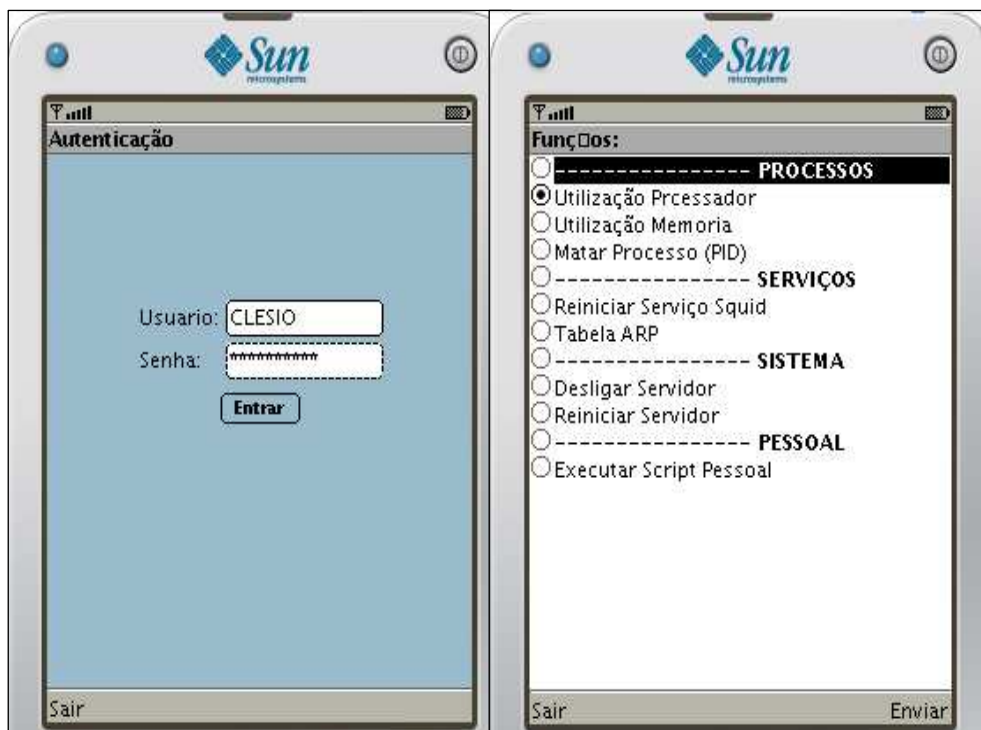


Figura 5-6 – Controle de Acesso ao ReMoS

Figura 5-7 – Menu de Funções do ReMoS

5.3.1.2. Funções de Controle de Processos

A partir da análise dos requisitos, foram definidas três funções básicas relativas ao controle de processos em execução no servidor. Com este grupo de funções, buscou-se resolver pequenos problemas de afogamento de processamento e memória por processos indesejados.

Assim, o usuário pode emitir relatórios e monitorar processos com maior uso do processador e da memória, sendo capaz de encerrá-los usando o seu PID. Nos relatórios deste grupo de funções, os seis primeiros processos em uso de recursos podem ser visualizados, em obediência às limitações de processamento, de tela e de pouco tráfego em rede dos dispositivos móveis.

A

Figura 5-8 apresenta o *menu* das funções de processos (

Figura 5-8a) e um exemplo de relatório de uso da memória no servidor (

Figura 5-8b). A tela de resultados também exibe relatório de erros, caso exista algum problema na execução dos comandos no servidor. Com o processamento efetuado com sucesso, o ReMoS apresenta a tela inicial.

5.3.1.3. Funções de Controle de Serviços

Os servidores normalmente são dedicados a serviços específicos, que variam em cada máquina da rede de acordo com a necessidade do ambiente. Desta forma, seria necessário inserir muitas funções diferenciadas ao ReMoS para atender a todos os serviços possíveis em servidores.

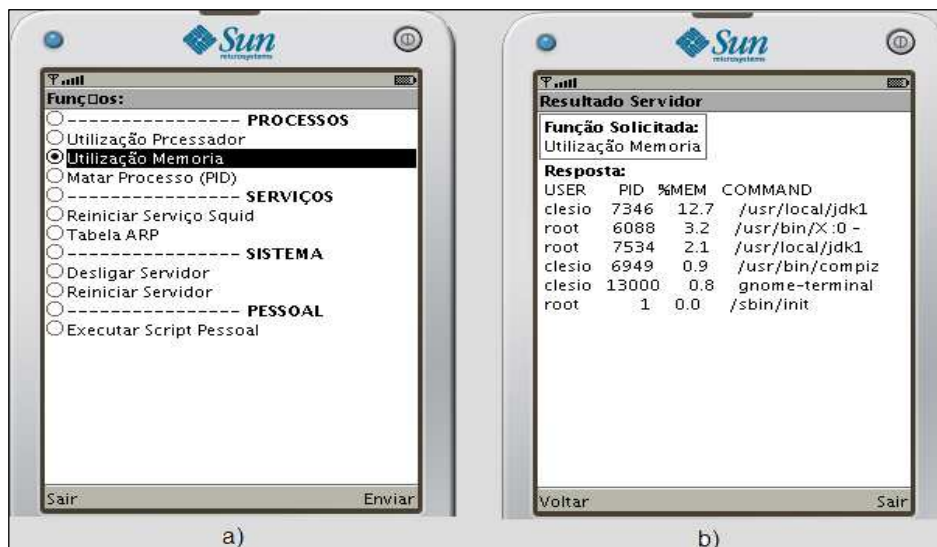


Figura 5-8 – a) Menu do ReMoS

b) Relatório de Uso da Memória

Como esta necessidade afeta diretamente a complexidade do ReMoS e os fatores de programação, por motivos de escassez de recursos de memória e de processamento dos dispositivos portáteis, foram definidas duas funções básicas: controle de serviço de *proxy* e identificação de máquinas em atividade na rede. Estas funções são largamente usadas em pelos profissionais da área.

A função do *proxy* refere-se ao *software Squid*, possibilitando reiniciar este serviço, quando for necessário. No tocante a identificação de máquinas ativas na rede, existe a possibilidade de recuperar a tabela ARP do servidor, com identificação de *Host Name*, IP e MAC (*Media Access Control*) dos computadores. A Figura 5-9 apresenta o *menu* de funcionalidade de serviços (Figura 5-9a) e a visualização da tabela ARP de um servidor (Figura 5-9b).

Para implementar os requisitos de generalização de uso do aplicativo nos diversos tipos de servidores, foi criada uma função, *script* pessoal, discutida nos próximos tópicos.

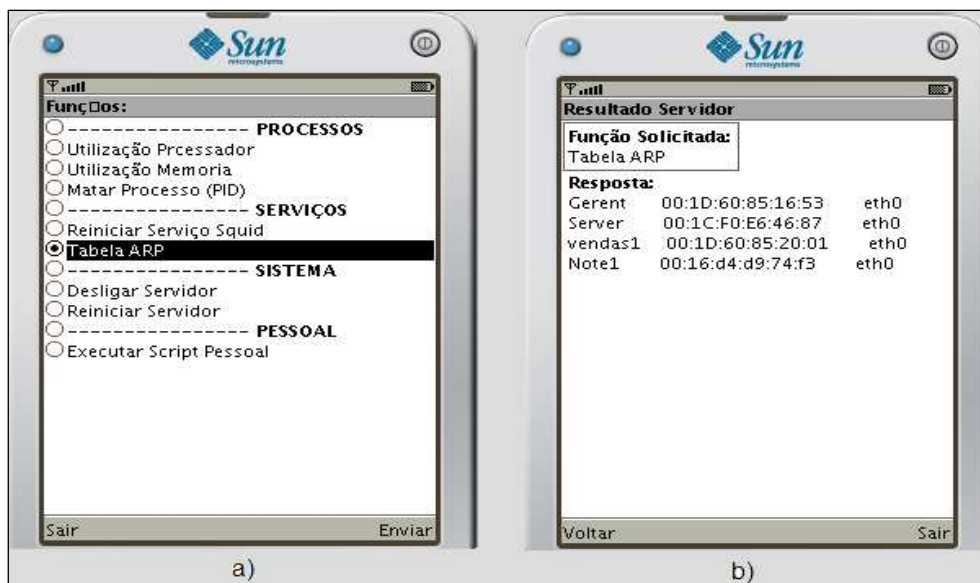


Figura 5-9 – a) Menu do ReMoS

b) Visualização da Tabela ARP

5.3.1.4. Funções de Sistema

De acordo com os requisitos definidos no ReMoS, foram identificadas duas funções: reiniciar sistema e desligar sistema. Essas funções são utilizadas em caso de pane de serviços ou da necessidade de reconfiguração do servidor, implicando em reiniciar para ativar as novas configurações.

Estas funções possibilitam apenas o simples manuseio do servidor, sendo de grande valia ao administrador de redes, pois é possível resolver alguns problemas que necessitam reiniciar o sistema operacional.

5.3.1.5. Funções Configuráveis

Com a necessidade de deixar o aplicativo mais utilizável e atraente aos usuários, foi implementada a possibilidade de criar *scripts* pessoais no acesso remoto via ReMoS. Assim, um conjunto de funções diversas criadas pelos usuários no servidor, usando *shell scripts*, pode ser executado pelo módulo Mobile, fazendo o uso de parâmetros para a entrada de dados no *script*. Ao final da execução, o ReMoS apresenta os resultados vindos do servidor.

Estas funções possibilitam uma generalização do uso do ReMoS, deixando-o configurável a cada usuário, para atender às suas necessidade de acesso remoto a servidores. A Figura 5-10 apresenta a chamada do *script* pessoal `uso_hd.sh` (Figura 5-10a) e a apresentação dos dados de resposta do servidor (Figura 5-10b).

5.3.2. Módulo Server

A operacionalização do ReMoS é feita usando este módulo. Ele recebe as requisições do módulo Mobile, efetua o processamento solicitado e envia a resposta ao módulo Mobile. O módulo Server não implementa interface com usuário, executando apenas processos internos de comunicação via rede.

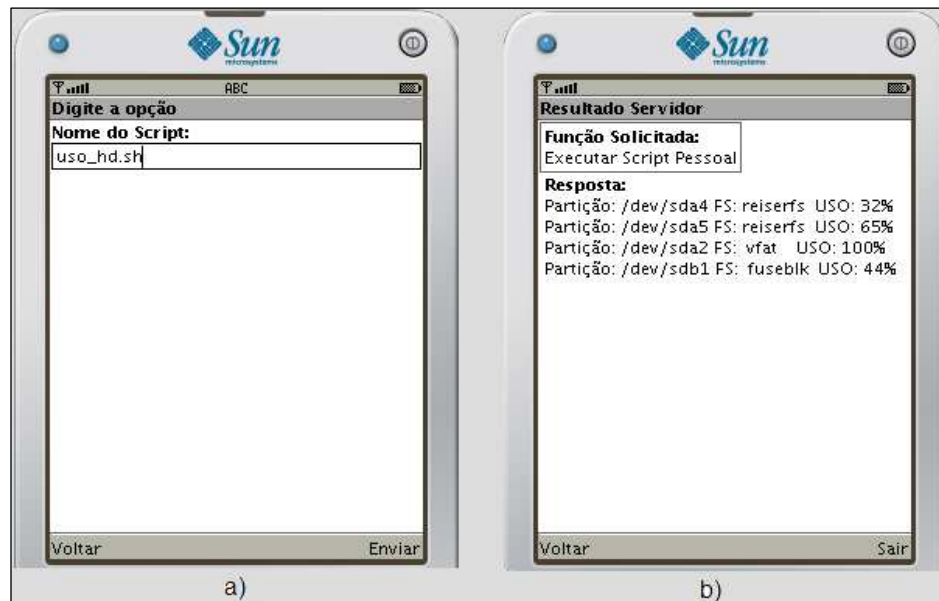


Figura 5-10 – a) Chamado do Script pessoal uso_hd.sh b) Resultado da Execução do Script no Servidor

O módulo Server necessita de uma máquina com recursos de servidor Web e com a tecnologia *Java Enterprise Edition (Servlets)* instalada para ser executado. A conexão cliente/servidor é feita usando o protocolo HTTP e *sockets*. O módulo Server aguarda por uma conexão com o módulo Mobile para efetuar o processamento. Este módulo é capaz de gerenciar mais de uma conexão por vez, pois a implementação de *thread* possibilita o gerenciamento da concorrência entre vários clientes solicitantes, sem a preocupação de perda ou desatualização de dados.

5.3.2.1. Execução de Tarefas

A partir do recebimento de uma solicitação, o módulo Server decide a função a ser executada, baseado nos parâmetros recebidos na conexão,

coordenando a troca de mensagens com o módulo Mobile. As funções são implementadas usando *shell scripts* do Linux.

O *shell scripts* é uma ferramenta poderosa para a administração de redes, com a possibilidade de realizar todas funções desejadas, pois ele não se limita apenas à execução de comandos de sistema, sendo composto pelas melhores propriedades das linguagens de programação robustas. Com base nesta propriedade, o módulo Server pode ganhar recursos diversos, configuráveis pelos usuários, implementando *scripts* de uso específico.

Cada *script* tem o objetivo de executar os comandos programados para realizar a função planejada e de gerar resultados a serem enviados ao módulo Mobile. Estes resultados são gravados em dois arquivos: `arq_resultado.tmp` e `arq_erro.tmp`. O primeiro arquivo contém o resultado das atividades e o segundo arquivo contém erros ocasionados por situações adversas. A partir do conteúdo destes arquivos, o módulo Server decide qual informação envia ao módulo Mobile, ou seja, decide se envia resultados ou informações de erro.

Como o módulo Server pode atender requisições simultâneas de vários módulos Mobile, foi permitida a concorrência na leitura e na escrita dos dois arquivos mencionados anteriormente. Esta concorrência foi implementada adicionado ao nome dos arquivos um número randômico controlado pelo módulo Server, especificando os arquivos temporários de cada conexão.

Além disso, foi utilizado o recurso de sincronização para os processos iniciados pelo módulo Server (leitura e gravação de arquivos e chamadas de sistema), sendo implementado com *flags*. Este controle coloca um processo em estado de espera até o término dos outros processos ligados à sua execução. Para evitar que, em casos de erro no *script*, o processo principal fique sempre bloqueado, foi especificado um tempo máximo para este estado. Caso este tempo seja atingido, o erro é reportado. Deste modo, o módulo Server pode emitir respostas completas ao módulo Mobile.

A Figura 5-11 apresenta a estrutura necessária de um *script*, para ser executado pelo módulo Server do ReMoS.

```
#!/bin/sh

# Script a ser executado pelo servidor ReMoS

#Parâmetros recebidos
FLAG=$1      # Flag para sincronização
RESULTADO=$2 # Arquivo de resultado
ERRO=$3      # Arquivo de erro

#..... 1º passo
# Seta Flag, para colocar processos concorrentes (no acesso
# dos arquivos de resultado e erro) em modo de espera
echo "0" > $FLAG

#..... 2º passo
#Bloco de comandos
ifconfig eth0 1> $RESULTADO 2> $ERRO

#..... 3º passo
# Seta Flag, para liberar a leitura dos arquivos de resultado e erro
echo "1" > $FLAG
```

Figura 5-11 – Estrutura do *Script* a ser Executado no Módulo Server

5.3.2.2. Comunicação Cliente x Servidor

A comunicação entre o módulo Mobile e o módulo Server é estabelecida usando o protocolo HTTP com conexão do tipo leitura e escrita (READ_WRITE), por meio do método POST deste protocolo. Este método implementa segurança de transmissão, encapsulando os dados a serem transmitidos no corpo da mensagem, não deixando estes parâmetros de forma visível aos usuários. Para esta conexão, foi utilizado o pacote Java HttpURLConnection da plataforma J2ME.

A Figura 5-12 apresenta um trecho de código do módulo Mobile que cria a conexão HTTP com o servidor.

```
HttpConnection con = null;

con = (HttpConnection) Connector.open(url, Connector.READ_WRITE);
con.setRequestMethod(HttpConnection.POST);
```

Figura 5-12 – Código para Implementar Conexão com o Módulo Sever

Com a conexão estabelecida, inicia-se a transferência de dados, que acontece usando um *array* de bytes, contendo as informações a serem transmitidas. Este *array* é processado nos dois lados da comunicação, para obter os parâmetros em separado e realizar as funções previstas no ReMoS.

Para tanto, foram utilizadas as classes `InputStream` e `OutputStream`, cuja propriedade é manipular fluxos de entrada e de saída de dados de uma conexão. Isto pode ser observado no trecho de código do

cliente, apresentado na Figura 5-13, onde este recurso é utilizado para obter dados da conexão, por meio da função `read` da classe `InputStream`.

```
InputStream is = null;

is = con.openInputStream();
byte[] dado_recebido = new byte[700];
is.read(dado_recebido);
```

Figura 5-13 – Código para Implementar Transferência de Dados

5.4. Considerações Finais

A proposta do ReMoS é munir o administrador de redes com uma conexão de alta disponibilidade aos servidores com acesso seguro, a partir da utilização dos dispositivos móveis. A disponibilidade do ReMoS é afetada apenas em setores ainda não cobertos pela telefonia celular, áreas que sofrem um processo de diminuição gradativa, ocasionada pela competitividade entre operadoras de celulares.

O ReMoS, além da alta disponibilidade, é configurável às verdadeiras necessidades de cada computador servidor controlado, possibilitando a agregação de funções diversas para cada servidor, na mesma ferramenta. Para isso, *scripts* pessoais são usados.

6. Considerações Finais

Este capítulo apresenta as conclusões deste trabalho, bem como as contribuições no âmbito acadêmico e comercial obtidas com o seu desenvolvimento. Além disso, são sugeridas propostas de trabalho futuros a serem implementados para a continuação deste trabalho.

6.1. Conclusões

Atualmente, com os avanços gradativos da tecnologia, os dispositivos portáteis se tornam cada vez mais poderosos, oferecendo aos usuários formas novas e fascinantes de comunicação e de administração para seus negócios. Java é uma tecnologia que possibilita aos desenvolvedores promover esta evolução, estendendo as aplicações existentes em computadores, para estes dispositivos. Esta tecnologia tem evoluído constantemente, buscando acompanhar o crescimento dos dispositivos móveis e seus novos recursos, com o objetivo de garantir a eficiência e portabilidade, inerente à tecnologia.

Com esta plataforma, este trabalho verificou a utilidade do uso dos dispositivos móveis como agente principal da automação de tarefas remotas. Isso foi alcançado com a construção de um *software* específico para estes dispositivos. Deste modo, o ReMoS pôde trazer uma boa solução a administração remota de servidores com alta disponibilidade de acesso.

Com a utilização do ReMoS, pode-se concluir que problemas de rede, em muitas vezes delineados como questões críticas como parada de

serviços ou afogamento de processador ou memória, podem ser diagnosticados e resolvidos em tempo hábil, usando um dispositivo móvel. Além disso, pôde-se perceber que, com planejamento preditivo de falhas, isto é, implementação de *script* para solução de prováveis erros, é possível efetuar o controle de falhas de servidores usando o ReMoS.

Portanto, as tecnologias *wireless* e *mobile* vêm a cada dia se consolidando como recursos indispensáveis ao homem moderno, gerando informações precisas, atuais e relevantes ao dia-a-dia destes. É preciso apenas vislumbrar e criar soluções que atinjam estas necessidades, pois os recursos tecnológicos estão disponíveis gratuitamente, gerando boas oportunidades de informação *on-line*, sem fio e móvel.

6.2. Contribuições

Com este trabalho, foram exploradas características da plataforma J2ME, oferecendo leitura de referência aos que se interessam pelo desenvolvimento de soluções computacionais aplicadas ao ramo dos dispositivos móveis e Internet. Ressalta-se, que não há a necessidade de investimento financeiro para tanto, pois boas ferramentas de desenvolvimento são gratuitamente encontradas.

Além da relevância acadêmica, gerando conhecimentos teóricos e práticos, a principal contribuição se refere à criação do acesso remoto a servidores via dispositivos móveis, com uma proposta de aplicação de baixo custo, independentemente da plataforma operacional destes dispositivos. O sistema foi testado em ambiente real, contribuindo de

maneira satisfatória para o controle imediato de diversas falhas de servidores com uma abordagem pré-programada de soluções.

6.3. Trabalhos Futuros

Com este trabalho, por se situar em um ramo novo e emergente da informática, é possível vislumbrar vários projetos para a sua continuação, construindo soluções completas para automação remota de tarefas diversas.

Como complemento, sugere-se a construção de módulos de *software* móveis para controlar serviços específicos, munidos com as funções necessárias para atender com excelência o gerenciamento de tal serviço. Por exemplo, um aplicativo para controle de *firewall*, dispondo de funções para a manutenção das regras, o controle de portas e as permissões de tráfego de pacotes, e *kits* de regras pré-programadas para determinadas funções, facilitando o manuseio do *software* pelos usuários.

Ainda complementando este trabalho, pode-se apresentar uma solução móvel para recuperar *logs* de acesso e de operação de servidores, permitindo avaliação completa das atividades efetuadas na máquina. Nesta solução, pode ser inserida a escolha de *logs* de diversos serviços e dos registros de manipulação de usuários, com opções de análise em tela ou redirecionamento para *e-mail*.

BIBLIOGRAFIA

Andreão, C. F. Conceitos Básicos das Plataformas Java e J2ME.

Disponível em:

<<http://www.devmedia.com.br/articles/viewcomp.asp?comp=6484>>

Acessado em: 03 Dez. 2008.

Aurélio, M. *Ataque Passivo em Redes Wireless*. Disponível em:

<http://www.malima.com.br/article_read.asp?id=224>. Acessado em:

20 Fev. 2009.

Ayres, M. CDMA, GSM, TDMA, GPRS: saiba o que essas siglas significam. Disponível em:

<<http://tecnologia.uol.com.br/ultnot/2006/11/24/ult2870u201.jhtm>>

Acessado em: 01 Dez. 2008.

Bottentuiut, J. Júnior. Dispositivos e Tecnologias Móveis Para Empresas:

Possibilidades para os Negócios na Área da Informação. Disponível

em: <<http://publica.fesp.br/index.php/rnti/article/viewFile/55/71>>.

Acesso em: 01 dez. 2008.

Caraciolo, M. P. Desenvolvendo Aplicativos Python para Symbian OS.

Disponível em:

<<http://www.devmedia.com.br/articles/viewcomp.asp?comp=11644>>

Acessado em: 03 Dez. 2008.

Cruz, R. Celulares Querem Disputar Acesso à Internet. Disponível em:

<<http://www.wirelessbrasil.org/wirelessbr/colaboradores/alancarvalho/gprs.html>>.

Acessado em: 01 Dez. 2008.

Deitel .H. M.; Deitel P. J. *Java, como programar*. 3. ed. Porto alegre: Bookman, 2001.

Ferrari, G. Sistema operacional para Celular. Disponível em: <<http://www.sosestudante.com/informatica/sistema-operacional-de-celular.html>>. Acessado em: 01 Mar. 2009.

Haddad, R. Saiba porquê desenvolver aplicações para telefones celulares com .NET. Disponível em: <<http://www.linhadecodigo.com.br/Artigo.aspx?id=171>>. Acessado em: 01 Dez. 2008.

Hall, M.; Brown, L. *Core Servlets e JavaServer Pages - Volume 1: Tecnologias Core*. Rio de Janeiro, Ciência Moderna: 2005.

Horstmann, C. S.; Cornell, G. *Core Java2: Volume II – Recursos Avançados*. São Paulo: Makrom Book, 2001.

IDGNOW. Disponível em: <http://computerworld.uol.com.br/telecom/2009/02/18/brasil-tem-cerca-de-152-milhoes-de-telefone-celulares-diz-anatel/> Acessado em: 20 fev. 2009.

IMB. Think small with J2ME. Disponível em: <<http://www.ibm.com/developerworks/library/wi-j2me/>> Acessado em: 04 Dez. 2008.

JCP. Disponível em: < <http://jcp.org/en/jsr/detail?id=30>>. Acessado em: 10 Jan 2009.

LINUXDEVICES. Mobile software merger combines Android, LiMo forces. Disponível em: <<http://www.linuxdevices.com/news/NS7949959944.html> > Acessado em: 01 Mar. 2009.

Luca, L. Conheça 8 serviços de transmissão ao vivo de vídeos pela internet. Disponível em: <<http://idgnow.uol.com.br/internet/2009/02/19/conheca-8-servicos-de-transmissao-ao-vivo-de-videos-pela-internet/>> Acessado em: 01 Mar. 2009.

Lugon, P. T.; Rossato, T. Floggy: Framework de Persistência para J2me/Midp. Disponível em: <http://projetos.inf.ufsc.br/arquivos_projetos/projeto_284/Monografia_Floggy_Priscila_Thiago.pdf>. Acessado em: 15 Fev. 2009.

Martinez, C. Celulares analógicos serão desativados nos Estados Unidos. Disponível em: <<http://allgsm.blogspot.com/2008/02/celulares-analgicos-sero-desativados.html>> Acessado em: 01 Dez. 2008.

Medeiros, J. P. S. Usando o Umbrello como ferramenta de modelagem de software. Disponível em <<http://www.dicas-l.com.br/dicas-l/20060523.php> > Acessado em: 01 Dez. 2008.

Miranda, R. B.; Marcondes, V. P. P. Prime: uma solução Java para acesso móvel a informações utilizando GSM/GPRS. Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v3.1/art05.pdf>> Acessado em: 19 Fev. 2009.

- MOBILEZONE. Disponível em:
<http://www.mobilezone.com.br/artigo_flashlite.htm>. Acesso em: 22
Dez. 2008.
- Monte, F. *Vinton Cerf, pai da Web, fala do futuro da Internet*. Disponível
em:
[http://wnews.uol.com.br/site/noticias/materia_especial.php?id_secao=1
7&id_conteudo=249](http://wnews.uol.com.br/site/noticias/materia_especial.php?id_secao=17&id_conteudo=249). Acessado em: 10 Dez 2008.
- Murilo, N. *Aspectos de segurança em redes Wi-Fi*. Disponível em:
<http://www.malima.com.br/article_read.asp?id=287>. Acessado em:
19 Fev. 2009.
- NETBEANS. Disponível em: < <http://www.netbeans.org/features/>>.
Acessado em: 01 Dez. 2008.
- Neves, J. C. *Programação Shell Script*. 7ed. Rio de Janeiro: Braspot,
2008.
- Nogueira, W. F.; Loureiro Filho, E. C.; Almeida, H. O. de. *Plataformas
para Desenvolvimento de Jogos para Celulares*. Universidade Federal
de Campina Grande (UFCG), 2005.
- PALMLAND. Disponível em: <<http://www.palmland.com.br/symbian/>>.
Acessado em: 01 Mar. 2009a.
- PALMLAND. Disponível em: < <http://www.palmland.com.br/wince/>>.
Acessado em: 01 Mar. 2009b.

Pereira, R. Resultado de exames via celular. Disponível em:
<<http://www.parana-online.com.br/canal/tecnologia/news/201384/>>.
Acessado em: 02 Dez. 2008.

Pretto, C. O. et al. Utilização de computação móvel para qualificação de rotinas de operação e manutenção de redes de distribuição. Disponível em:
<http://www.scielo.br/scielo.php?pid=S0103-17592006000400006&script=sci_arttext&tlng=en>. Acessado em: 15 Fev. 2009.

Rabello, R. R. *et al.* Integrando a Tecnologia J2me no Âmbito Acadêmico. Disponível em:
<<http://www.unibratec.com.br/jornadacientifica/diretorio/NovoINT.pdf>> Acessado em: 15 Fev. 2009.

Ribeiro, F. J. L.; Lopes, J. C. R.; Pedroza, A. de C. P. Análise dos Processos de Segurança em Sistemas Móveis de 3a Geração. Disponível em:
<<http://www.gta.ufrj.br/ftp/gta/TechReports/RiLoPe03a.pdf>>
Acessado em: 20 Fev. 2009

Salomão, P. L. Utilização do computador de mão integrado à telefonia celular no atendimento médico: desenvolvimento de sistema e avaliação. Disponível em:
<http://www.disacad.unifesp.br/pg/..%5Csapg%5Carquivos%5Carq_9.pdf>. Acessado em: 15 Fev. 2009.

SUN. Disponível em: <<http://java.sun.com/javaee/>>. Acessado em: 11 Dez. 2008a.

SUN. Disponível em: <<http://java.sun.com/products/sjwtoolkit/>>. Acessado em: 01 Dez. 2008b.

SUN. Disponível em: <<http://java.sun.com/javame/>>. Acessado em: 12 Dez. 2008c.

SUN. Disponível em: <<http://java.sun.com/j2se/1.5.0/docs/index.html>>. Acessado em: 12 Dez. 2008d.

SUN. Disponível em: <<http://java.sun.com/j2ee/1.3/docs/tutorial/doc/>>. Acessado em: 12 Dez. 2008e.

SYMBIANBRASIL. Disponível em: <<http://www.symbianbrasil.com/sobre/>>. Acessado em: 01 Mar. 2009.

Tan J. Disponível em: <http://br.tecnologia.yahoo.com/article/17062008/5/noticias-tecnologia-samsung-ve-alta-9-nas-vendas-globais-celulares.html> Acessado em: 10 jan. 2009.

TECNOLOGIA3G. Frequências. Disponível em: <<http://www.tecnologia3g.com.br/frequencias.php>> Acessado em: 20 fev. 2009.

TELECO. Telefonia Celular. Disponível em: <<http://www.teleco.com.br/tecnocel.asp>> Acessado em: 01 Dez. 2008a.

TELECO. 3G. Disponível em: <http://www.teleco.com.br/3g_tecnologia.asp>. Acessado em: 20 fev. 2008b.

TELECO. Telefonia Celular. Disponível em:
<<http://www.teleco.com.br/pais/celular.asp>> Acessado em: 20 fev.
2009a.

TELECO. HSDPA. A tecnologia. Disponível em:
<http://www.teleco.com.br/tutoriais/tutorialhsdpad/pagina_3.asp/>
Acessado em: 20 fev. 2009b.

TOMCAT. Disponível em: < <http://tomcat.apache.org/>>. Acessado em:
12 jan. 2009.

YP. Disponível em: <http://dsc.ufcg.edu.br/~yp/principal.htm>. Acessado
em: 01 Dez. 2008.