

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO**

**PROTÓTIPO DE UMA APLICAÇÃO LBS UTILIZANDO GPS**  
**CONECTADO EM CELULAR PARA CONSULTAR DADOS**  
**GEOREFERENCIADOS**

**ALEX KUHNEN**

**BLUMENAU**  
**2003**

**2003/2-02**

**ALEX KUHNEN**

**PROTÓTIPO DE UMA APLICAÇÃO LBS UTILIZANDO GPS  
CONECTADO EM CELULAR PARA CONSULTAR DADOS  
GEOREFERENCIADOS**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do curso de Ciência  
da Computação — Bacharelado.

Prof. Francisco Adell Péricas – Orientador

**BLUMENAU  
2003**

**2003/2-02**

**PROTÓTIPO DE UMA APLICAÇÃO LBS UTILIZANDO GPS  
CONECTADO EM CELULAR PARA CONSULTAR DADOS  
GEOREFERENCIADOS**

Por

**ALEX KUHNEN**

Trabalho aprovado para obtenção dos créditos  
na disciplina de Trabalho de Conclusão de  
Curso II, pela banca examinadora formada  
por:

Presidente:

---

Prof. Francisco Adell Péricas. – Orientador, FURB

Membro:

---

Prof. Sérgio Stringari, FURB

Membro:

---

Prof. Jomi Fred Hübner, FURB

Blumenau, 11 de dezembro de 2003

Dedico este trabalho aos meus pais, Domingos e Vera, à minha irmã, Kalinca, e à minha namorada, Vanessa, pelo incentivo na realização deste trabalho.

## **AGRADECIMENTOS**

À Deus, pelas oportunidades e desafios, e por sempre estar junto a mim guiando meus passos.

À minha família, que mesmo longe, sempre esteve presente.

Aos meus amigos, pelos empurrões e cobranças.

À minha empresa, Quick Soft, que me apoiou e incentivou para a realização deste trabalho.

Ao meu orientador, Francisco Adell Péricas, por ter acreditado na conclusão deste trabalho.

A todos os professores do colegiado que com paciência e dedicação repassaram seu conhecimento e experiência na certeza da formação de grande profissional.

## RESUMO

Este Trabalho de Conclusão de Curso apresenta um estudo sobre Sistemas de Informação Geográfica (SIG), mais especificamente *Location Based Services* (LBS) que permita obter uma aplicação para dispositivos móveis baseado na plataforma Java 2 Micro Edition para realizar consultas via web a dados geoprocessados através da coleta de dados de um GPS.

Palavras chaves:LBS, GPS, J2ME, *Servlets*, PDAs.

## **ABSTRACT**

This work aims to carry a study on Geographical Information System (GIS), more specifically Location Based Services (LBS) that allows to get as final result an application for mobile devices based on Java 2 Micro Edition to consult by Web a Geographic data, through a GPS device.

Key-Words: LBS, GPS, J2ME, Servlets, PDAs.

## LISTA DE ILUSTRAÇÕES

FIGURA 1: Arquitetura de uma aplicação LBS.....	15
FIGURA 2: Estrutura do sistema GPS .....	17
FIGURA 3: LBS com A-GPS .....	18
FIGURA 4: Tecnologia AOA .....	20
FIGURA 5: Tecnologia TDOA .....	21
FIGURA 6: A plataforma Java.....	25
FIGURA 7: Camadas da arquitetura J2ME.....	26
FIGURA 8: Processo de servidor para rodar <i>Servlets</i> .....	31
FIGURA 9: Especificação genérica do protótipo.....	34
FIGURA 10: Fluxograma do processo A .....	35
FIGURA 11: Fluxograma do processo B .....	36
FIGURA 12: Fluxograma do processo C .....	37
FIGURA 13: Fluxograma do processo D.....	38
FIGURA 14: J2ME Wireless Toolkit.....	40
FIGURA 15: Emulador DefaultColorPhone .....	41
FIGURA 16: Ferramenta JDeveloper.....	42
FIGURA 17: Tela para pesquisa de farmácias .....	48
FIGURA 19: Consulta retornando uma mensagem de erro .....	50
FIGURA 20: Confirmação do serviço.....	51
FIGURA 21: Pesquisa pela localização da Farmácia Essencial.....	53
FIGURA 22: Pesquisa pela localização da Farmácia Big .....	54
FIGURA 23: Pesquisa pela localização da Farmácia Camila .....	55
FIGURA 24: Pesquisa com valores inválidos .....	56
QUADRO 1: Implementação do processo A (Efetua Pesquisa) .....	43
QUADRO 2: Implementação do processo B (Busca Localização).....	44
QUADRO 3: Implementação do processo C (Verifica Pesquisa).....	45
QUADRO 4: Implementação do processo D (Verifica Estabelecimento Mais Próximo) .....	46

## LISTA DE TABELAS

Tabela 1: Sub-processos do protótipo .....	33
Tabela 2: Farmácias e suas respectivas localizações.....	52



## LISTA DE SIGLAS

LBS – *Location Based Services*  
SIG – *Sistema de Informações Geográficas*  
GPS – *Global Positioning System*  
SMS – *Short Message Service*  
IDC – *International Data Corporation*  
J2ME – *Java 2 Micro Edition*  
PDA – *Personal Digital Assistant*  
A-GPS – *Assisted-GPS*  
Cell ID – *Cell Identification*  
AOA – *Angle of Arrival*  
TDOA – *Time Difference of Arrival*  
FTC – *Federal Trade Commission*  
JVM – *Java Virtual Machine*  
MIDP – *Mobile Information Device Profile*  
PDAP – *Personal Digital Assistant Profile*  
CLDC – *Connected Limited Device Configuration*  
CDC – *Connected Device Configuration*  
GCF – *Generic Connection Framework*  
KVM – *Kilo Virtual Machine*  
URI – *Uniform Resource Indicator*  
WAP – *Wireless Application Protocol*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>11</b>
1.1 OBJETIVOS DO TRABALHO .....	12
1.2 ESTRUTURA DO TRABALHO .....	12
<b>2 SISTEMA DE INFORMAÇÃO GEOGRÁFICA .....</b>	<b>14</b>
2.1 LOCATION BASED SERVICE - LBS .....	15
2.1.1 Formas de implementação do LBS .....	15
2.1.2 Tecnologias de Localização .....	16
2.1.2.1 Global Positioning System .....	16
2.1.2.2 Assisted-GPS .....	17
2.1.2.3 Cell Identification .....	18
2.1.2.4 Angle of Arrival.....	19
2.1.2.5 Time Difference of Arrival .....	20
2.1.3 Aplicações de LBS .....	21
2.1.4 Telefonia Celular.....	22
<b>3 JAVA.....</b>	<b>24</b>
3.1 JAVA 2 MICRO EDITION.....	26
3.1.1 Camada de Configuração .....	27
3.1.2 Camada de Perfil .....	28
3.1.3 Camada do Interpretador.....	29
3.1.4 Generic Connection Framework .....	29
3.1.5 Aplicações <i>wireless</i> .....	30
3.2 SERVLETS .....	31
<b>4 DESENVOLVIMENTO DOS PROTÓTIPOS .....</b>	<b>32</b>
4.1 REQUISITOS PRINCIPAIS DO PROTÓTIPO .....	32
4.2 ESPECIFICAÇÃO DO PROTÓTIPO .....	33
4.3 IMPLEMENTAÇÃO .....	38
4.3.1 Ferramentas utilizadas.....	39
4.3.2 Principais funções do protótipo.....	42
4.3.3 Operacionalidade do software cliente .....	46
4.4 TESTES E VALIDAÇÃO.....	51
<b>5 CONCLUSÕES.....</b>	<b>57</b>
5.1 DIFICULDADES ENCONTRADAS .....	58

5.2 LIMITAÇÕES.....	58
5.3 EXTENSÕES .....	58
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>60</b>

## 1 INTRODUÇÃO

A constante evolução na área tecnológica está proporcionando grandes avanços no mundo da geotecnologia. Prova disso é a grande interatividade dos sistemas de informações geográficas atuais. O que antes era uma realidade apenas para centros de pesquisas ambientais, hoje atinge os mais variados públicos de diversas maneiras. Uma delas, e provavelmente a mais recente, é disponibilização dos *Location Based Services* (LBS).

Em uma visão futurista, Smyth (2000 apud Câmara, 2000) citou o seguinte exemplo: imagine um mundo onde todos os componentes relevantes (pessoas e coisas) possuam uma identidade e se quiserem, possam tornar sua localização acessível a qualquer momento. Algo como um aparelho de *Global Positioning System* (GPS) com um código de identificação pessoal acoplado a um relógio de pulso. Neste cenário, o indivíduo poderá estar num corredor de supermercado e ouvir de um pacote de cereais a excitante pergunta: "Sr. X, não vais me comprar esta semana? Não estás esquecendo nada?". Segundo Câmara (2000), esta visão descreve que com o amadurecimento e a expansão dos dispositivos de conexão sem-fio (*wireless*), o grande mercado das geotecnologias passará da simples produção de mapas para os serviços baseados em localização.

A fusão entre dispositivos móveis e a internet está causando uma grande mudança cultural na sociedade. Graças a tecnologias como o *Short Message Service* (SMS), os LBS estão começando a serem usados nas mais diversificadas áreas. Em São Paulo já é comum os táxis estarem munidos com equipamentos de navegação capazes de traçar rotas por menor caminho e, até mesmo, de determinar rotas alternativas para desviar de tráfegos congestionados. Segundo o *International Data Corporation* (IDC), a receita gerada somente pelos serviços de localização deve ser superior a US\$ 5 bilhões em 2004.

Vindo ao encontro destas tecnologias, a plataforma *Java 2 Micro Edition* (J2ME), caso específico do ambiente Java, está se tornando a solução preferida pela grande maioria dos fabricantes de dispositivos para telefonia celular. Um reflexo disso é a quantidade de celulares que já dispõem da máquina virtual Java (JVM) instalada. Além destes aparelhos, outros dispositivos *wireless* como *paggers* de duas vias, *Personal Digital Assistant* (PDAs) também dão suporte a J2ME.

Usando a plataforma J2ME, o presente trabalho desenvolve uma solução LBS baseada em informações geo-personalizadas, capaz de identificar e listar quais são os estabelecimentos mais próximos da localização do usuário no momento da consulta.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver um protótipo de software para telefone celular que permitir ao usuário fazer consultas via WEB a uma base de dados georeferenciada.

Os objetivos específicos do trabalho são:

- a) utilizar os recursos de conectividade existentes na API J2ME;
- b) estabelecer comunicação entre o GPS e o software para capturar as informações da localização atual;
- c) fazer com que o software efetue consultas em um servidor WEB passando como parâmetros os dados colhidos do GPS;
- d) tratar as consultas recebidas no servidor WEB, retornando apenas as informações condizentes com a localização informada;
- e) apresentar de forma gráfica ou textual os valores retornados na consulta, demonstrando assim, como seria um aplicativo LBS.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em seis capítulos, estando assim distribuídos:

No primeiro capítulo, encontra-se a introdução e os objetivos a serem alcançados com o desenvolvimento do trabalho.

No segundo capítulo é apresentado um estudo sobre sistema de informação geográfica. Incluindo ainda algumas seções referentes aos serviços baseados em localização.

O terceiro capítulo trata da tecnologia Java. Apresenta ainda um estudo mais aprofundado na plataforma J2ME.

No quarto capítulo trata de uma pesquisa sobre telefonia celular, onde é apresentado um breve histórico desta área, bem como a realidade atual e as perspectivas futuras.

O quinto capítulo apresenta informações sobre o desenvolvimento do protótipo, demonstrando as ferramentas utilizadas, a especificação e a implementação do mesmo.

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

## 2 SISTEMA DE INFORMAÇÃO GEOGRÁFICA

Câmara (1996) define Sistema de Informação Geográfica (SIG) como sistema automatizado usado para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é relevante e/ou indispensável. O seu surgimento se deve principalmente da necessidade que se tinha de reduzir os custos na produção e manutenção de mapas. Com a possibilidade de armazenar e representar informações geográficas em ambientes computacionais, deu-se o aparecimento do geoprocessamento.

Os primeiros sistemas foram criados na década de 70, no Canadá, mas como nesta década os recursos tecnológicos ainda eram muito precários, estes sistemas ganharam sustentação somente a partir da década seguinte.

No Brasil, o responsável pela introdução do geoprocessamento foi o professor Jorge Xavier da Silva da Universidade Federal do Rio de Janeiro (UFRJ), no início dos anos 80. A partir deste ponto, começaram a surgir vários grupos de estudo interessados em estudar e desenvolver esta tecnologia, os quais desenvolveram vários sistemas de informação geográfica, tais como: Sistema de Análise Geo-Ambiental (SAGA) do grupo do Laboratório de Geoprocessamento do Departamento de Geografia da UFRJ, MaxiDATA da AeroSul e Sistema para Processamento de Informações Geográficas (SPRING) da Divisão de Processamento de Imagens do Instituto Nacional de Pesquisas Espaciais (INPE).

Os SIG's possuem uma gama muito grande de aplicações que podem ser classificadas em áreas distintas como: sócio-econômicas, envolvendo o uso da terra, seres humanos e a infra-estrutura existente; ambientais, enfocando o meio ambiente e o uso dos recursos naturais; e de gerenciamento, envolvendo a realização de estudos e projeções que determinam onde e como alocar recursos para remediar problemas ou garantir preservação de determinadas características (CAMARA,1996).

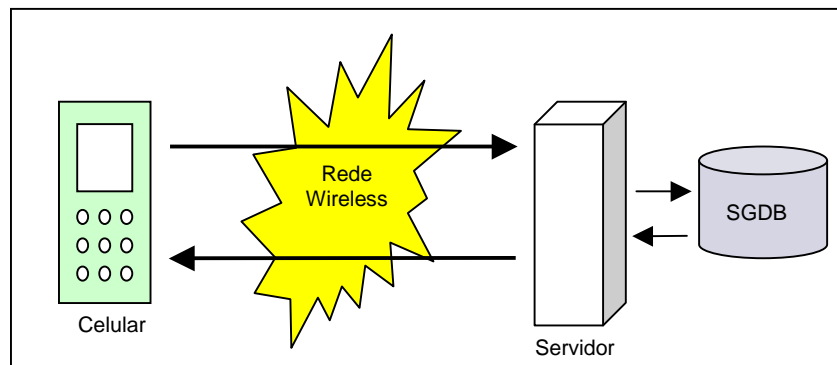
Existem vários tipos de SIG, cada um apresentando propósitos distintos e servindo a variados tipos de tomadas de decisão. Uma grande quantidade de nomes vem sendo aplicada ao SIG, distinguindo por suas funções e papéis. Dentre estes, um dos mais recentes é o *Location Based Services* (LBS), onde a sua principal função é prover conteúdo baseado em localização (CONCEIÇÃO, 2001).

## 2.1 LOCATION BASED SERVICE - LBS

A tecnologia sem-fio atinge hoje um estágio tão avançado de desenvolvimento, que está se tornando um dos tópicos mais instigantes da atualidade. Isso porque ela tem o poder de derrubar as fronteiras entre comunicação, informação, mídias e entretenimento, oferecendo todos estes serviços na palma da mão, em qualquer lugar e a qualquer hora.

Embora tudo isso ainda seja algo novo para muitas pessoas, as empresas envolvidas com esta tecnologia já estão implantando vários serviços para dispositivos *wireless*, dentre eles os *Location Based Services*. Os LBSs permitem aos usuários de dispositivos móveis utilizarem serviços baseados em sua posição ou localização geográfica (ROCHA, 2001). Com base nestas informações é possível que empresas comerciais forneçam serviços personalizados capazes de atender as necessidades de seus usuários dependendo de onde estiverem.

O LBS surgiu graças à evolução das redes de comunicação sem fio, ao aumento da mobilidade e à necessidade de novas aplicações e tem sua origem ligada à convergência de múltiplas tecnologias como: sistema de informação geográfico, internet, comunicação sem fio e dispositivos portáteis. A figura 1 demonstra a arquitetura de uma aplicação LBS.



Fonte: Adaptado de Miranda (2003)

FIGURA 1: Arquitetura de uma aplicação LBS

### 2.1.1 Formas de implementação do LBS

Os serviços de localização podem utilizar soluções baseadas em *handset* (*handset-based*) ou rede (*network-based*), dependendo da base geográfica ou da base de clientes. As soluções *handset-based* especificam que o dispositivo terá que ter um receptor GPS para fazer o processo de localização. Já as soluções *network-based* fazem com que a localização seja



feita de acordo com a área de cobertura das células. Com isto, as operadoras de telefonia celular ficam responsáveis por disponibilizar informações sobre a localização dos usuários.

A principal vantagem de se utilizar soluções *handset* é de não ter que pagar o serviço de localização prestado pela operadora. De outro lado, a desvantagem é que como o celular estará conectado a um GPS, este também estará consumindo a bateria do aparelho. Já as soluções baseadas em rede proporcionam a seus usuários maior liberdade na hora comprar um aparelho celular, visto que este não terá necessariamente que vir com um GPS.

A união de soluções *handset-based* com *network-based* dá o surgimento de uma terceira solução chamada híbrida, onde a tarefa de realizar a localização é dividida entre o dispositivo móvel e a operadora. O uso dela dá uma garantia maior de que o serviço de localização funcionará corretamente. Porém o seu custo é mais elevado, tendo em vista que a informação da localização será feita pela operadora e/ou pelo dispositivo móvel.

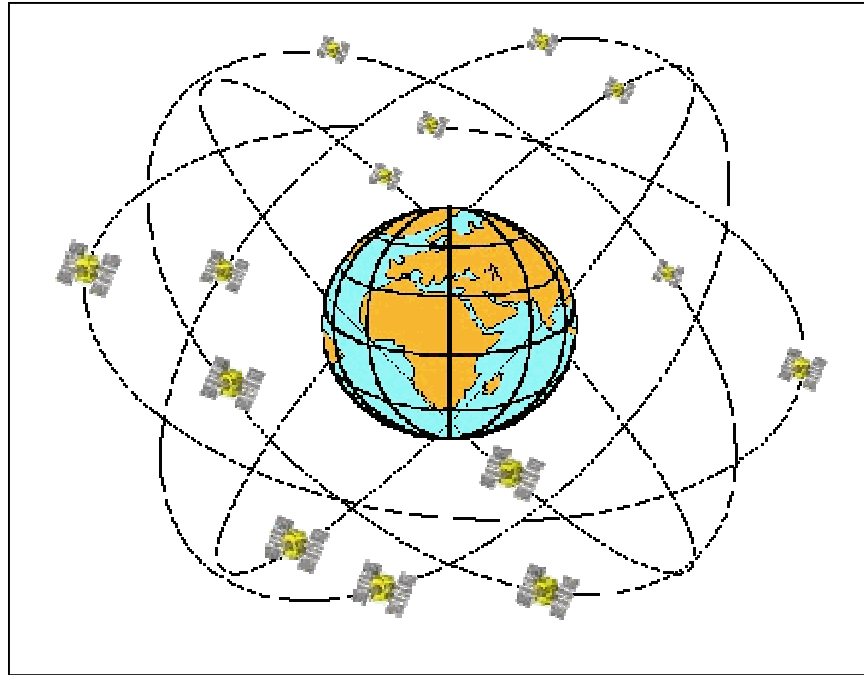
### 2.1.2 Tecnologias de Localização

Atualmente existem várias tecnologias de localização que podem ser utilizadas por serviços baseados em localização. O uso de cada uma dependerá principalmente do tipo de solução que se pretende adotar, ou seja, inicialmente deve-se definir se a solução será *handset-based*, *network-based* ou híbrida.

#### 2.1.2.1 Global Positioning System

Em 1973, o Departamento de Defesa dos Estados Unidos proveu a fusão de dois projetos, TIMATION (*Time Navigation*) da Marinha e 621B da Força Aérea para a concepção de um novo sistema denominado NAVSTAR/GPS. O *Navigation Satellite with Time and Ranging* (NAVSTAR) ou *Global Position System* foi projetado para fornecer a posição instantânea bem como a velocidade de um ponto sobre a superfície da Terra ou próximo a ela, num referencial tridimensional (JÚNIOR, 2003). Este sistema é baseado em satélites que enviam mensagens específicas que são interpretadas pelo receptor GPS.

Em sua estrutura final o sistema conta com 24 satélites espalhados por 6 órbitas diferentes. Esta configuração faz com que qualquer ponto na superfície da Terra, ou próxima a ela, tenha a cobertura de no mínimo 4 satélites. A figura 2 demonstra a estrutura do sistema GPS.



Fonte: Júnior (2003)

FIGURA 2: Estrutura do sistema GPS

O GPS determina a localização calculando o tempo que ele leva para capturar os sinais individuais de pelo menos três satélites. Baseando-se em cada um destes intervalos de tempo e multiplicado-os pela velocidade da luz, o receptor calcula a distância que se encontra de cada satélite. A partir deste momento o receptor obtém sua localização usando a Lei de Cosine – complexo processo similar à triangulação (SIMPSON, 1997).

As aplicações dos receptores GPS variam em precisão e funcionalidade. Alguns possuem programas que fazem transformações entre sistemas de coordenadas ou contém dados de saída compatíveis com sistemas de SIG comuns no mercado. Outros dispositivos permitem a leitura com os receptores em movimento e são bastante úteis para realizar mapeamento de terrenos com veículos (CÂMARA, 1996).

As vantagens de se utilizar receptores GPS são a precisão da informação e o tamanho destes dispositivos, que podem ser encontrados até mesmo em relógios de pulso. A desvantagem do uso destes receptores é que eles não conseguem ter muita precisão em ambientes fechados (*indoor*) (ROCHA, 2001).

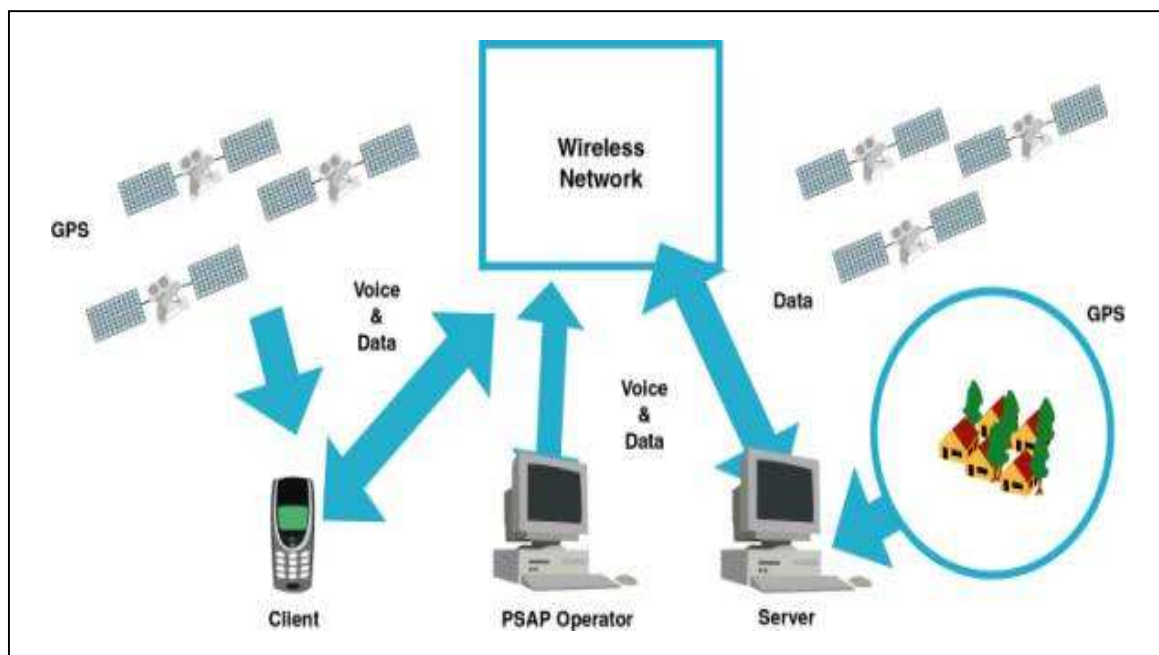
#### 2.1.2.2 Assisted-GPS

Segundo Zurstrassen (2003) a idéia do *Assisted-GPS* (A-GPS) é distribuir as tarefas de posicionamento entre os dispositivos móveis e o servidor de serviços de localização. O

servidor se comunica com os satélites da rede mapeando sua região de controle e passa estas informações para o dispositivo móvel. Desta forma este dispositivo não tem necessidade de fazer todos os cálculos de sua localização sozinho, pois recebe estas informações do servidor pré-processadas.

Os principais benefícios do A-GPS são a redução do tempo de localização, economia de energia e simplificação dos requisitos do receptor de GPS. A desvantagem é que este método é mais caro em relação ao GPS.

Um produto que utiliza esta tecnologia é o SnapTrack's Location on Demand (QUALCOMM, 2003). Ele é basicamente uma tecnologia de localização com marketing de precisão. Caso o usuário opte por não ser localizado, o serviço é suspenso e ainda permite que o consumo da bateria seja reduzido (ZURSTRASSEN, 2003). A figura 3 demonstra um exemplo de uma aplicação baseada nesta tecnologia.



Fonte: Zurstrassen (2003)

FIGURA 3: LBS com A-GPS

### 2.1.2.3 Cell Identification

A *Cell Identification (Cell ID)* é uma tecnologia onde a localização é determinada através da célula onde o usuário se encontra. Cell ID ou localização baseada em células se dá da seguinte maneira: um servidor armazena as informações de cada célula, latitude e longitude da posição da torre, ângulo inicial e final dos setores da estação e o raio máximo de

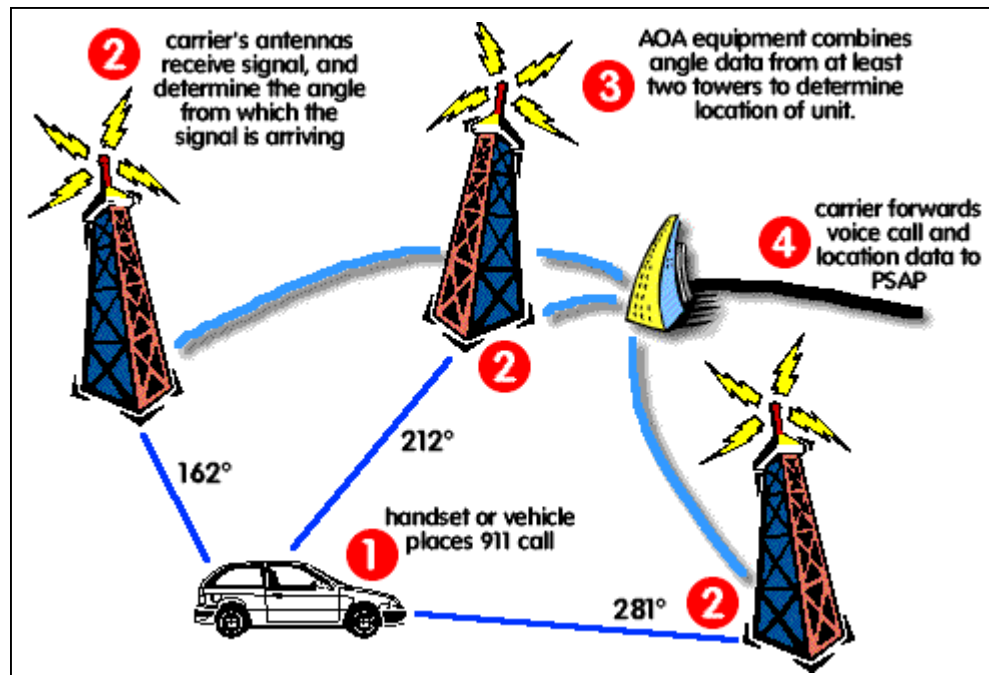
cobertura. Com posse destas informações o servidor poderá calcular o centróide de cada setor (ZURSTRASSEN, 2003).

Todas estas informações servirão para o cálculo da localização dos assinantes quando esta for solicitada ao servidor de aplicações LBS. Com isto, informações referentes ao posicionamento do usuário serão diretamente proporcionais ao tamanho da célula onde se encontram. Sendo que esta variação poderá ser de 150m a até 20Km. Embora esta seja a grande desvantagem desta tecnologia, o fato de a localização ser muito rápida e de fácil disponibilização ainda a torna uma boa opção quando se fala em soluções *network-based*.

#### 2.1.2.4 Angle of Arrival

O *Angle of Arrival* (AOA) é uma das mais conhecidas abordagens ao problema da localização. Seu princípio está no cálculo do ângulo com que determinado sinal chega às antenas. O AOA é determinado pela variação de fase dos sinais recebidos ao longo de um grupo de antenas. A diferença de fase do sinal entre antenas deste grupo resulta em ângulos recebidos, e isto pode ser referenciado em relação a qualquer direção fixa. O conhecimento do AOA de pelo menos dois locais, fornece a informação necessária para a localização da fonte do sinal (SILVA, 2003).

Esta tecnologia é muito menos dispendiosa que a do GPS, por aproveitarem os sinais transmitidos pelos aparelhos móveis, dispensando a modificação quer aos aparelhos, quer à rede de comunicações. Uma das desvantagens desta tecnologia é que a precisão da localização é degradada sempre que existir interferência no sinal (ROCHA, 2001). A figura 4 exemplifica o uso da tecnologia AOA.



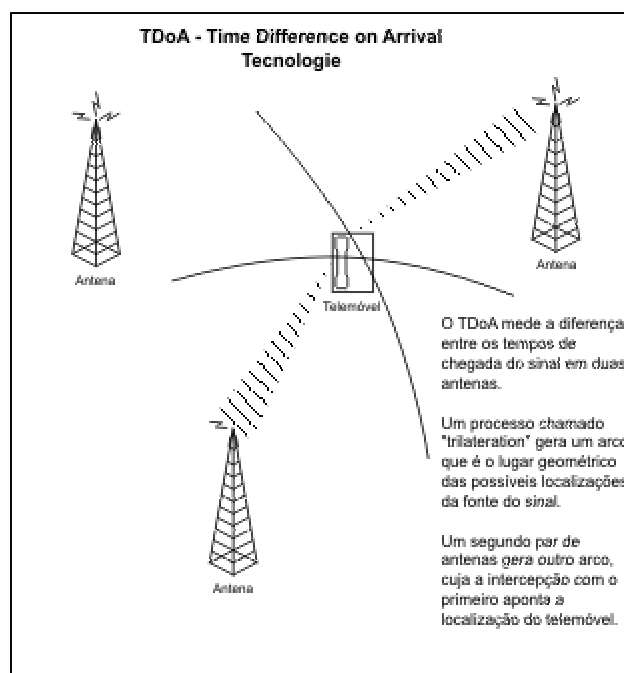
Fonte: Zurstrassen (2003)

FIGURA 4: Tecnologia AOA

#### 2.1.2.5 Time Difference of Arrival

O *Time Differential of Arrival* (TDOA) usa receptores de rádio frequência que são instalados em vários lugares da antena para capturar sinais. Nesta abordagem, a posição do dispositivo móvel é calculada pela medição das diferenças em tempo do sinal recebido em cada receptor (ROCHA, 2001).

A precisão deste tipo de tecnologia é determinada pelas limitações de largura de banda do sinal, bem como a precisão do sistema e pelo ambiente em que se propaga. Outras desvantagens incluem a extrema precisão temporal necessária em cada receptor, a necessidade de recalibrar estes mesmos receptores devida a diferentes influencias no sinal. A vantagem desta tecnologia é o fato de não ser necessário modificar os dispositivos móveis (SILVA, 2003). A figura 5 demonstra o uso da tecnologia TDOA.



Fonte: Zurstrassen (2003)

FIGURA 5: Tecnologia TDOA

### 2.1.3 Aplicações de LBS

Atualmente, uma das áreas mais interessantes no uso dos LBS é a de assistência em emergências. De tal maneira que nos EUA, a *Federal Trade Commission* (FTC) elaborou uma lei denominada E911Act destinada a padronizar e aprimorar o serviço de emergência 911 através de dispositivos móveis. Esta lei está em vigor desde 1999 e seu objetivo é melhorar o tempo de resposta nas chamadas de emergência feitas por usuários de telefones celulares (DEITEL, 2003).

No Brasil, já é possível encontrar aplicativos LBS baseados em soluções *handset*. Um deles é o Apontador Duo (WEBRASKA, 2003), um guia de ruas com recursos de voz e GPS. Este software foi desenvolvido para um dispositivo do tipo Palm Pilot, onde são armazenados os mapas de acordo com a rota determinada pelo usuário.

Outra área que terá um grande impulso a partir do uso de serviços de localização será a do comércio eletrônico móvel (*m-business*). Com a combinação entre os dispositivos móveis e os LBS, o *m-business* terá como elaborar estratégias de marketing extremamente poderosas. Pode-se fazer com que uma pessoa que tenha um aparelho celular receba uma mensagem informando que o restaurante que está a 500m de distância fará uma promoção relâmpago em meia-hora.

#### 2.1.4 Telefonia Celular

Desde 1973, quando foi feita a primeira ligação de aparelho celular até os dias de hoje, o mercado de telecomunicações vem se destacando a cada ano que passa. Prova disto é o número de usuários de telefones celulares – são mais de um bilhão em todo o mundo (CARDOZO, 2003).

No começo, as deficiências tecnológicas, o tamanho e o custo dos aparelhos fizeram com que o mercado de telefonia celular ficasse restrito a um grupo de pessoas que gozavam de um poder aquisitivo bastante alto. Nesta primeira geração (1G), os escandinavos foram os primeiros a montar uma rede que abrangesse múltiplos países. Em seguida, vários países lançaram seus serviços de telefonia celular. Como as diversas redes eram incompatíveis entre si, 26 companhias européias formaram, em 1982, o *Groupe Speciale Mobile* (GSM) com a intenção de aumentar a capacidade do sistema através de multiplexação e adotar um padrão único para o uso dos mesmos aparelhos em toda a Europa. Mais tarde a sigla GSM passou a ser *Global System for Mobile Communications*. Desta maneira começava a surgir a segunda geração, ou 2G (GREGO, 2003).

No Brasil, os telefones celulares começaram a aparecer em 1990, mas eram caros e de difícil acesso. Esta situação só melhorou em 1998, quando as empresas privadas tiveram acesso ao mercado de telecomunicações, até então dominado pelas estatais. Com isto surgiram as primeiras redes digitais no Brasil e o celular começou a ser mais acessível.

Os dois padrões adotados inicialmente no Brasil foram o *Time Division Multiple Access* (TDMA) e o *Code Division Multiple Access* (CDMA). Ambos referem-se ao método de multiplexação.

Em 1999, a União Internacional de Telecomunicações definiu especificações para telefonia celular de terceira geração (3G), onde a meta era permitir tráfego de dados em alta velocidade, possibilitando a transmissão de vídeo e música. Devida a grande diferença existente entre o 2G e o 3G, foram criados padrões intermediários denominados de 2,5G. A velocidade de uma rede de 2G é de no máximo 14,4Kbps, enquanto a de 2,5G pode chegar a 144Kbps e a de 3G chega a 2Mbps. No Brasil os serviços de 2,5G chegaram em 2001 enquanto que neste mesmo ano, Japão e Coreia do Sul já tinham serviços de 3G (GREGO, 2003).

Segundo Reggiani (2003), a previsão dos laboratórios da indústria de celulares é de que em 7 anos estará implantada uma revolução nos hábitos de todos os habitantes deste planeta. A idéia é que com a consolidação da rede 3G, o celular passe a ser utilizado de várias formas, como por exemplo, acoplado ao carro para detectar falhas, localizar serviços de manutenção e determinar qual a melhor rota para se chegar lá. Assim, pretende-se que o acesso móvel seja o canal direto das empresas com o indivíduo, entregando informações relacionadas ao local e confirmando os serviços baseados em localização como fator decisivo e indissociável ao sucesso da telefonia celular.



### 3 JAVA

Em 1991 a Sun Microsystems, diante da concepção de que a próxima área em que os microprocessadores teriam um impacto profundo seria a de dispositivos eletrônicos inteligentes destinados ao consumidor final, financiou uma pesquisa para o estabelecimento de uma linguagem capaz de atender os requisitos destes dispositivos. Esta pesquisa tinha o codinome *Green* e o seu resultado foi o desenvolvimento de uma linguagem baseada em C e C++ que seu criador, James Gosling, chamou de Oak (carvalho) em homenagem a uma árvore que dava para a janela de seu escritório na Sun. Mais tarde descobriu-se que já havia uma linguagem de computador com este nome. Foi quando uma equipe da Sun que estava numa cafeteria local sugeriu o nome Java, que é uma cidade de onde um tipo específico de café é importado (DEITEL, 2001).

Mas como o mercado naquela época não se desenvolveu tão rápido como era esperado, e neste mesmo período a *World Wide Web* estava tendo um grande impulso, a Sun vislumbrou o potencial de utilizar Java para criar páginas da Web com o chamado conteúdo dinâmico. Foi desta forma que Java ganhou força e vem se consolidando como uma das linguagens mais conhecidas e utilizadas mundialmente (DEITEL, 2003).

Atualmente Java está distribuída em três plataformas: Java 2 *Enterprise Edition* (J2EE), Java 2 *Standard Edition* (J2SE) e Java 2 *Micro Edition* (J2ME). Estas três plataformas formam um conjunto que possibilita o desenvolvimento de aplicações completas, robustas e portáteis.

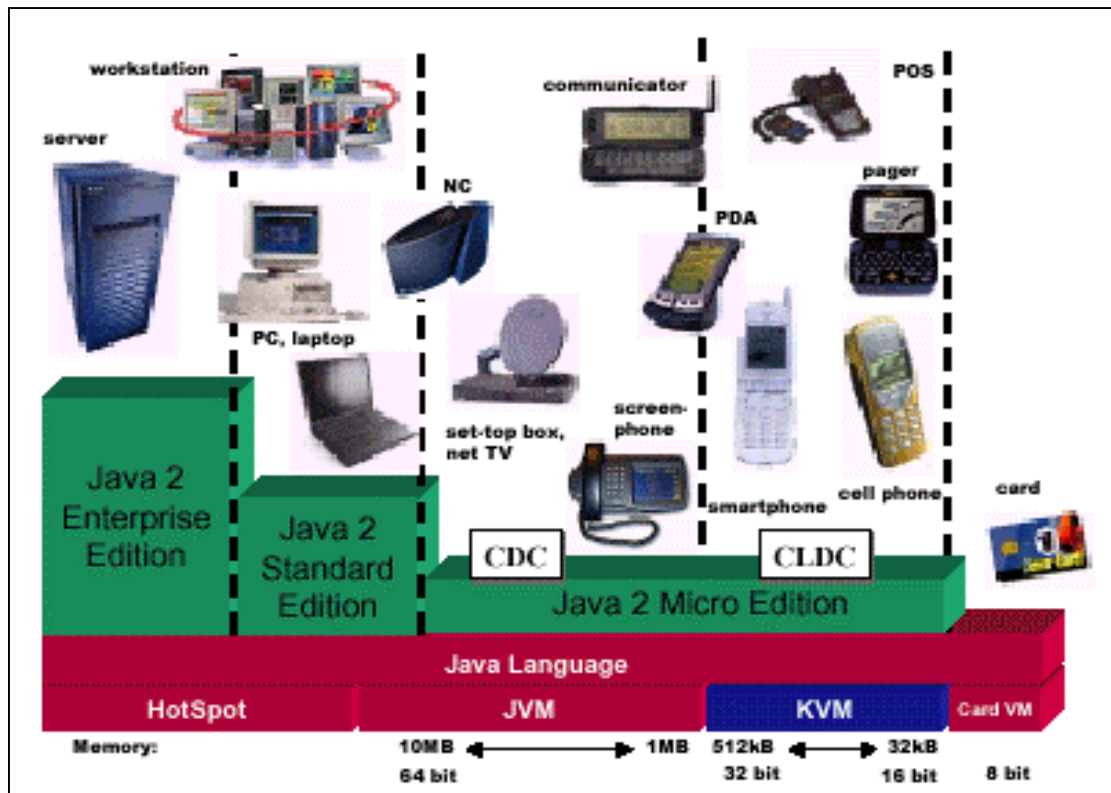
A J2SE foi a primeira edição criada da Java 2 *Platform*. Esta plataforma dá suporte para o desenvolvimento de aplicações para *desktop* e estações de trabalho. J2SE é a versão básica do Java com *Applications Programming Interface* (API) padrão (DEPINÉ, 2002).

Segundo SouJava (2003), J2EE é um conjunto de especificações coordenadas, objetivando o desenvolvimento de aplicações corporativas. As especificações do J2EE são criadas um grupo muito grande de empresas, entre elas Sun, IBM, BEA, Oracle, Borland, Apache, e para a qual existem dezenas de produtos compatíveis hoje no mercado.

A mais recente edição é a J2ME. Esta tecnologia é direcionada ao mercado de pequenos dispositivos. Seu conjunto de especificações tem por objetivo disponibilizar uma

JVM, API e ferramentas para equipamentos como: telefones celulares, *paggers*, *handhelds*, vídeo games, sistemas embutidos, etc (SOUJAVA, 2003).

A figura 6 representa as três edições de Java, bem como os dispositivos para os quais cada camada foi implementada.



Fonte: Sun (2003)

FIGURA 6: A plataforma Java

Os principais fatores que tornam Java uma linguagem de destaque são (ORACLE, 2002):

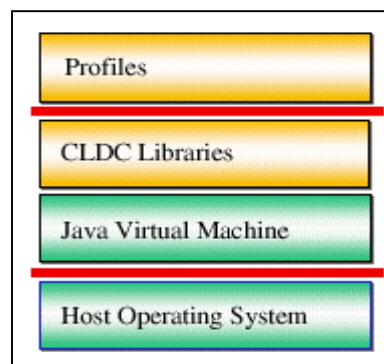
- Orientação a Objetos: a programação orientada a objetos é um paradigma natural e poderoso usado no desenvolvimento de aplicativos. Pesquisas apontam que o uso deste paradigma permite uma sobrevivência mais prolongada às mudanças que acompanham o crescimento de um negócio;
- Portabilidade e independência de plataforma. Programas Java são interpretados por uma máquina nativa (Java Virtual Machine - JVM) em tempo de execução. Pelo fato destes programas executarem debaixo do controle da JVM, os programas Java podem ser utilizados em qualquer sistema operacional que tenha uma JVM;

- c) Facilidade de distribuição. Classes Java podem ser baixadas dinamicamente na internet quando requerido. Adicionalmente, Java provê suporte para cliente-servidor e programação distribuída;
- d) *Multithreading*. Java disponibiliza primitivas de concorrência para o programador. Isto significa que é possível desenvolver aplicativos que executam múltiplas tarefas em paralelo;
- e) Robustez e segurança. Java tem a capacidade interna de impedir a corrupção de memória. Automaticamente administra os processos de alocação de memória e a checagem de limite de array.

### 3.1 JAVA 2 MICRO EDITION

Em 1999 a Sun Microsystems introduziu no mercado uma edição de Java capaz de ser executada em dispositivos de baixa capacidade de processamento e pouca memória disponível, conhecida como Java 2 Micro Edition (FONSECA, 2002). Esta edição nasceu de um projeto chamado *Spotless System*, cujo principal objetivo era criar uma JVM portátil que desse suporte a sistemas embarcados (KROLL, 2002).

Segundo a Sun (2003), a arquitetura J2ME é modular e escalável. Esta modularidade e escalabilidade é definida pela tecnologia J2ME em um modelo de 3 camadas embutidas sobre o sistema operacional do dispositivo: a camada de perfil (*Profile*), a camada de configuração (*Configuration*) e a camada representada pela máquina virtual Java (*Java Virtual Machine*), juntamente com o sistema operacional presente no dispositivo (*Host Operation System*). A aplicação encontra-se logo acima da camada de perfil. Estas três camadas podem ser observadas na figura 7.



Fonte: Sun (2003)

FIGURA 7: Camadas da arquitetura J2ME

Como a configuração é responsável por determinar os requisitos básicos, o perfil funciona como um complemento para atender detalhes específicos do dispositivo. Alguns dos perfis existentes são: *Mobile Information Device Profile* (MIDP), *Personal Digital Assistant Profile* (PDAP), PersonalJava etc.

Os recursos de conectividade de *Java 2 Micro Edition* existentes no CLDC estão especificados no *Generic Connection Framework* (GCF). Este *framework* oferece suporte a vários protocolos de comunicação como HTTP/HTTPS, TCP, UDP, serial e infravermelho.

Uma das características mais interessantes para aplicações MIDP, é a capacidade destas aplicações responderem a eventos de conectividade, mesmo que a máquina virtual Java não esteja ativa.

A versão suportada na maioria dos dispositivos celulares no Brasil é a CLDC 1.0 e o MIDP 1.0. O suporte comercial ao MIDP 2.0 está previsto para o final de 2003, quando deverão ser lançados os primeiros modelos baseados nessa nova versão do perfil (CORBERA, 2003).

### 3.1.1 Camada de Configuração

A camada de configuração é classificada de acordo com as capacidades de cada dispositivo. A configuração *Connected Limited Device Configuration* (CLDC) contém uma API mínima para poder rodar aplicativos em dispositivos móveis com limites de processamento e memória, tais como, telefones celulares, *smartphones*, *paggers* e PDAs. Opostamente, a *Connected Device Configuration* (CDC) pressupõe dispositivos com mais capacidades como os *PocketPCs*.

Para a configuração CLDC os aparelhos devem apresentar as seguintes características (CORBERA, 2003):

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

A segurança, que a CLDC define, está baseada no chamado “*Sandbox Security Model*”, isto é, a máquina virtual terá um espaço de memória independente do restante das aplicações do celular (tais como agenda, tons, imagens, configuração, *browser* WAP, etc). Nesse modelo de segurança restringem-se as operações que podem ser executadas a um conjunto de funções da API tão somente. Nenhuma outra operação é permitida. Com isso a aplicação não pode acessar a área de memória do calendário ou agenda de contatos do aparelho celular. Outro ponto importante no que tange ao Java é o fato que não é permitido carga de classes definidas pelo usuário de forma a garantir a segurança de qualquer aplicação desenvolvida com a API CLDC.

As regras de design de uma aplicação CLDC são simples: manter tudo pequeno e o mais simples possível. As aplicações para CLDC devem ser projetadas para consumir o mínimo de recurso possível, e deve permitir ao usuário sair do aplicativo a qualquer hora sem que informações sejam perdidas (KROLL, 2002).

### 3.1.2 Camada de Perfil

No topo da camada de CLDC existe a camada de software chamada *Profile*. O *Profile*, ou perfil, especifica um conjunto de APIs que moldam-se a um determinado tipo de dispositivo.

Um perfil funciona como o complemento da configuração, em que esta define o mínimo de funcionalidades e o perfil atende detalhes específicos do dispositivo. Ou seja, a configuração descreve de forma geral uma família de dispositivos, enquanto o perfil fica mais específico para um tipo particular de dispositivo em uma dada família. Alguns dos perfis existentes são: *Mobile Information Device Profile* (MIDP), *RMI Profile*, *Foundation Profile*, *PDA Profile*, *Personal Java*, etc.

O perfil MIDP é voltado especificamente aos dispositivos portáteis. Este perfil especifica interface com o usuário, entrada e persistência de dados, manipulação de eventos, modelo de armazenamento orientado a registro, rede, mensagens e aspectos relacionados ao ambiente de programação para esses dispositivos (DEPINÉ, 2002).

Muitos fabricantes adicionam ao MIDP algumas outras APIs, tais como suporte a uma forma simplificada de AWT (*Abstract Window Toolkit*) e controles de vibração, som e

display. Com isto torna-se difícil portar aplicações entre fabricantes, a menos que se use sempre o conjunto mínimo especificado no MIDP 1.0 (CORBERA, 2003).

As aplicações Java baseadas na configuração CLDC e no perfil MIDP, e portanto voltadas a dispositivos móveis, são chamadas de *Midlets*.

### 3.1.3 Camada do Interpretador

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

A *Kilo Virtual Machine* (KVM) é uma nova implementação da JVM para dispositivos portáteis com recursos limitados. A KVM aceita o mesmo conjunto de *bytecodes* e o mesmo formato para o arquivo “.class” que a máquina virtual clássica.

A KVM foi desenhada para ser pequena, para uso de memória estática entre 40 a 80Kb. A quantidade mínima de memória requerida pela KVM é 128Kb, incluindo o interpretador, um mínimo de bibliotecas especificadas pela configuração e algum espaço para rodar aplicativos.

### 3.1.4 Generic Connection Framework

Em Java 2 Standard Edition, as classes para manipulação de conexões de rede estão localizadas no pacote java.net. Este pacote contém mais de 20 classes, interfaces e classes de exceção. Isto garante ao menos uma classe para cada tipo de conexão.

A quantidade de classes e interfaces que seriam necessárias para suportar conexões de rede seria muito grande para serem adaptadas em Java 2 Micro Edition. Assim, o *Connected Limited Device Configuration* tem uma aproximação uniforme para conexões de rede chamado *Generic Connection Framework* (GCF). O GCF contém apenas a classe genérica *Connector* (KROLL, 2002). Nesta classe o *Uniform Resource Indicator* (URI) é utilizado para criar os objetos implementadores da interface *Connection* especializados no protocolo específico.

Com o GCF, o Java 2 Micro Edition garante que funcionalidades tão importantes como as de pesquisa e envio de mensagens interagindo com serviços localizados em outras

máquinas possam ser realizadas. E isto no meio wireless envolve fatores como a escolha de protocolo de comunicação, formato de dados, largura de banda e latência de rede (MIRANDA, 2003).

Para aplicações MIDP, a característica mais importante é a baixa latência de rede, ou seja, quanto menor o tempo transcorrido desde o início de uma transmissão até a origem até o início do seu recebimento pelo destino melhor. Porém a largura de banda que pode ser definida como a quantidade de informação que pode ser transmitida em uma única conexão já não é tão importante para este tipo de aplicação. Isto porque estas aplicações não têm a necessidade de transmitir grande quantidade de informação, até mesmo porque isso poderia levar muito tempo e conseqüentemente também afetaria a bateria do dispositivo.

### 3.1.5 Aplicações *wireless*

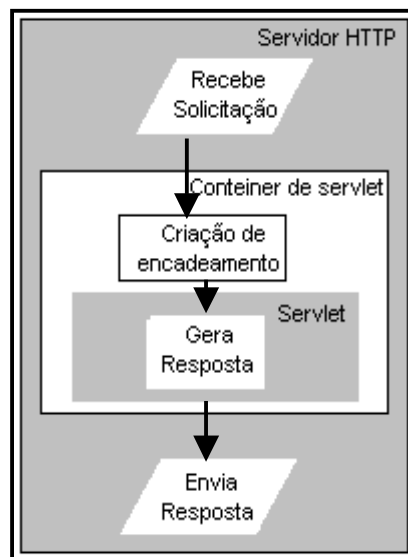
O mercado de aplicações *wireless*, em especial o de telefonia celular, começou já há algum tempo. O início foi marcado pela noção de que os dispositivos móveis seriam um micro-computador de bolso. A intenção que se tinha era a de transportar vários serviços dados pelos PCs diretamente para os celulares. Imaginava-se que seria possível acessar a internet com a mesma facilidade que no computador pessoal. Foi dessa maneira que surgiu o *Wireless Application Protocol* (WAP). Ele foi desenvolvido para facilitar a introdução e a padronização do acesso à internet sem-fio (DEITEL, 2003). Porém, neste início não foi considerado que a forma de utilização dos dispositivos *wireless* não era a mesma que as dos PCs.

Quando se acessa a internet em um dispositivo móvel, deve-se lembrar que ele possui várias características que o distinguem de um micro-computador. Além das limitações do tamanho, bateria, falta de teclado entre outras, os momentos em que estes aparelhos são utilizados são totalmente diferentes dos momentos em os PCs são utilizados. Um computador pessoal geralmente lembra algo mais estático como, por exemplo, ficar atrás de uma mesa fazendo trabalhos de faculdade. Usuários de equipamentos *wireless* geralmente estão em trânsito, com a atenção compartilhada entre o aparelho e outras atividades. Estas características fundamentais para o sucesso de aplicações *wireless* podem ser atendidas plenamente com o uso de *Java 2 Micro Edition*.

### 3.2 SERVLETS

Um *Servlet* é um programa escrito em linguagem Java que é carregado dinamicamente para atender às solicitações de um servidor web, ou seja, é uma extensão acrescentada ao servidor que aumenta a sua funcionalidade. Os servidores web respondem às solicitações dos usuários, geralmente, usando o protocolo HTTP através do envio de documentos escritos em HTML (GATTASS, 2000).

Os *Servlets* de Java têm um modelo de programação similar aos *scripts* de CGI, na medida em que recebem uma solicitação de HTTP de um navegador da web como entrada e espera-se que localizem e/ou construam o conteúdo apropriado para a resposta do servidor (FIELDS, 2000). A figura 8 demonstra onde o *Servlet* fica armazenado no servidor.



Fonte: Fields (2003)

FIGURA 8: Processo de servidor para rodar *Servlets*

Um *Servlet* pode ser carregado em qualquer servidor, desde que este tenha a máquina virtual Java em seu sistema. Os *Servlets* podem ser usados para o processamento de formulários (HTML), interagir com bancos de dados, ou de uma maneira geral atuar como uma camada intermediária em uma arquitetura de três camadas (GATTASS, 2000).



## 4 DESENVOLVIMENTO DOS PROTÓTIPOS

Os protótipos de software desenvolvidos neste trabalho formam uma aplicação de um serviço baseado em localização, onde se pretende identificar qual o estabelecimento mais próximo da localização informada. Esta aplicação está dividida em duas partes: o protótipo do software cliente e o protótipo do software servidor. O primeiro, é executado em um emulador de celular. O segundo, é executado em um servidor web.

O software cliente deve ser utilizado em dispositivos móveis que suportem a tecnologia J2ME, com a configuração CLDC e o perfil MIDP. As consultas efetuadas pelo software cliente são realizadas via web ao software servidor, onde um *Servlet* recebe os parâmetros da localização e por meio de um método de pesquisa identifica qual o estabelecimento mais próximo. Em seguida, o *Servlet* envia a resposta ao software cliente, onde então é mostrado o resultado ao usuário.

O protótipo cliente não foi executado em um celular pela falta do cabo DKU-5, que faz a comunicação entre o computador e o celular Nokia 2280, o qual seria utilizado neste trabalho. A aquisição deste cabo se tornou inviável pelo fato de ele ser importado e não se ter certeza que a compra do mesmo chegasse a tempo de se concluir este trabalho.

### 4.1 REQUISITOS PRINCIPAIS DO PROTÓTIPO

Alguns requisitos devem estar presentes no protótipo para ser realizada a especificação. Estes requisitos correspondem a algumas características que o protótipo precisa ter para alcançar o resultado esperado.

Os requisitos principais do protótipo são:

- a) buscar a localização através de um GPS para criar a consulta;
- b) permitir a comunicação com serviços residentes em servidores (*Servlets*), sendo estes responsáveis pelo processamento das consultas;
- c) efetuar consultas a dados geoprocessados, através de um servidor web, para obter um determinado estabelecimento mais próximo da posição localizada;
- d) apresentar os valores retornados nas consultas.

## 4.2 ESPECIFICAÇÃO DO PROTÓTIPO

Para a especificação do protótipo adotou-se a fluxogramação para demonstrar o fluxo dos principais processos executados nos protótipos.

Na tabela 1 são enumerados os sub-processos juntamente com a descrição de cada um deles.

Tabela 1: Sub-processos do protótipo

<b>Nome</b>	<b>Descrição</b>	<b>Ambiente de Execução</b>
Processo A	Efetua Pesquisa	Emulador de Celular
Processo B	Busca Localização	Emulador de Celular
Processo C	Verifica Pesquisa	Servidor web
Processo D	Verifica Estabelecimento Mais Próximo	Servidor web

A figura 9 apresenta uma especificação genérica da aplicação.

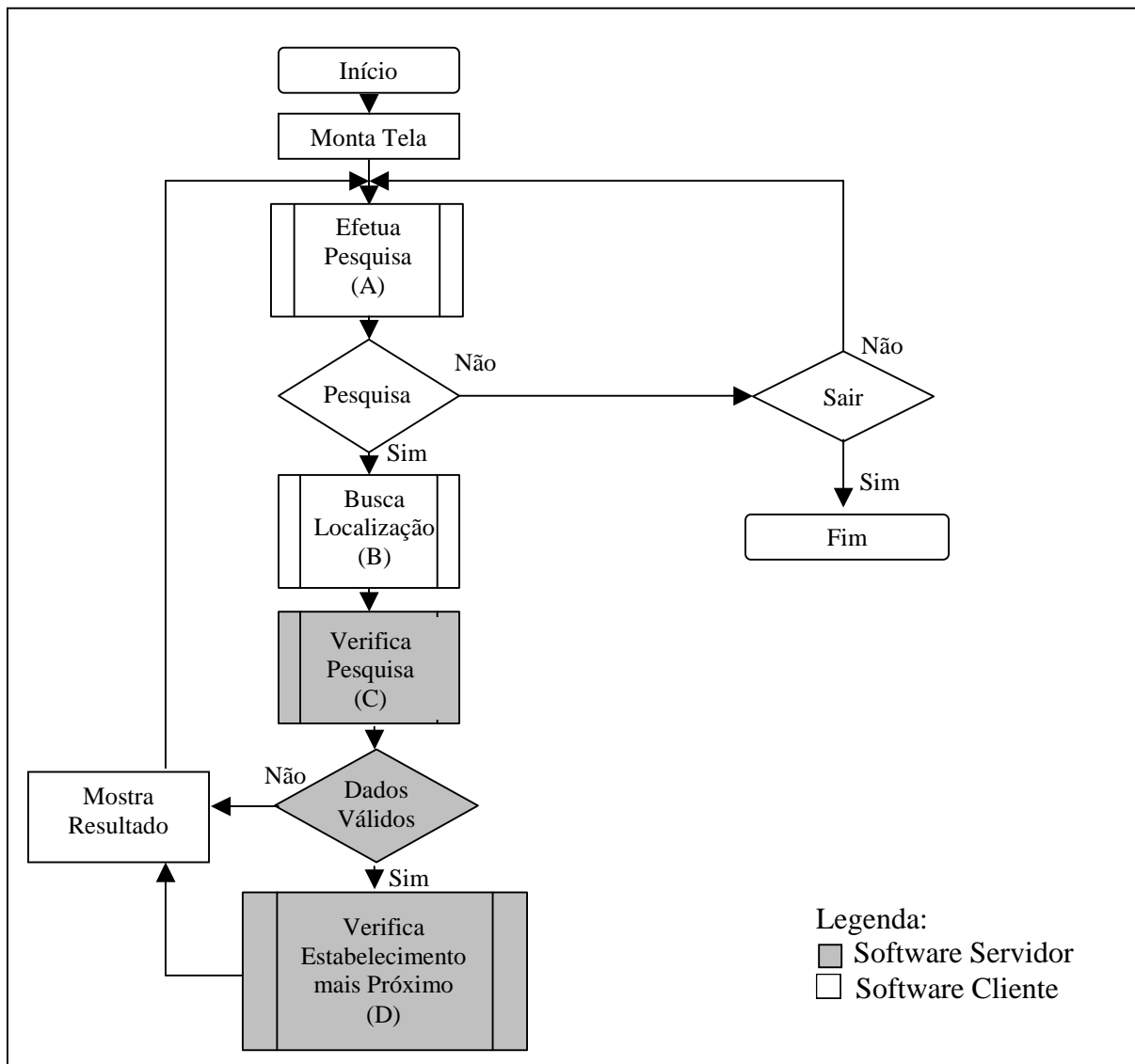


FIGURA 9: Especificação genérica do protótipo

A seguir são apresentados os diagramas correspondentes aos sub-processos da aplicação com suas respectivas descrições.

Na figura 10 observa-se o detalhamento do processo A, que busca os parâmetros para a consulta através do Processo B e inicia o processo de pesquisa.

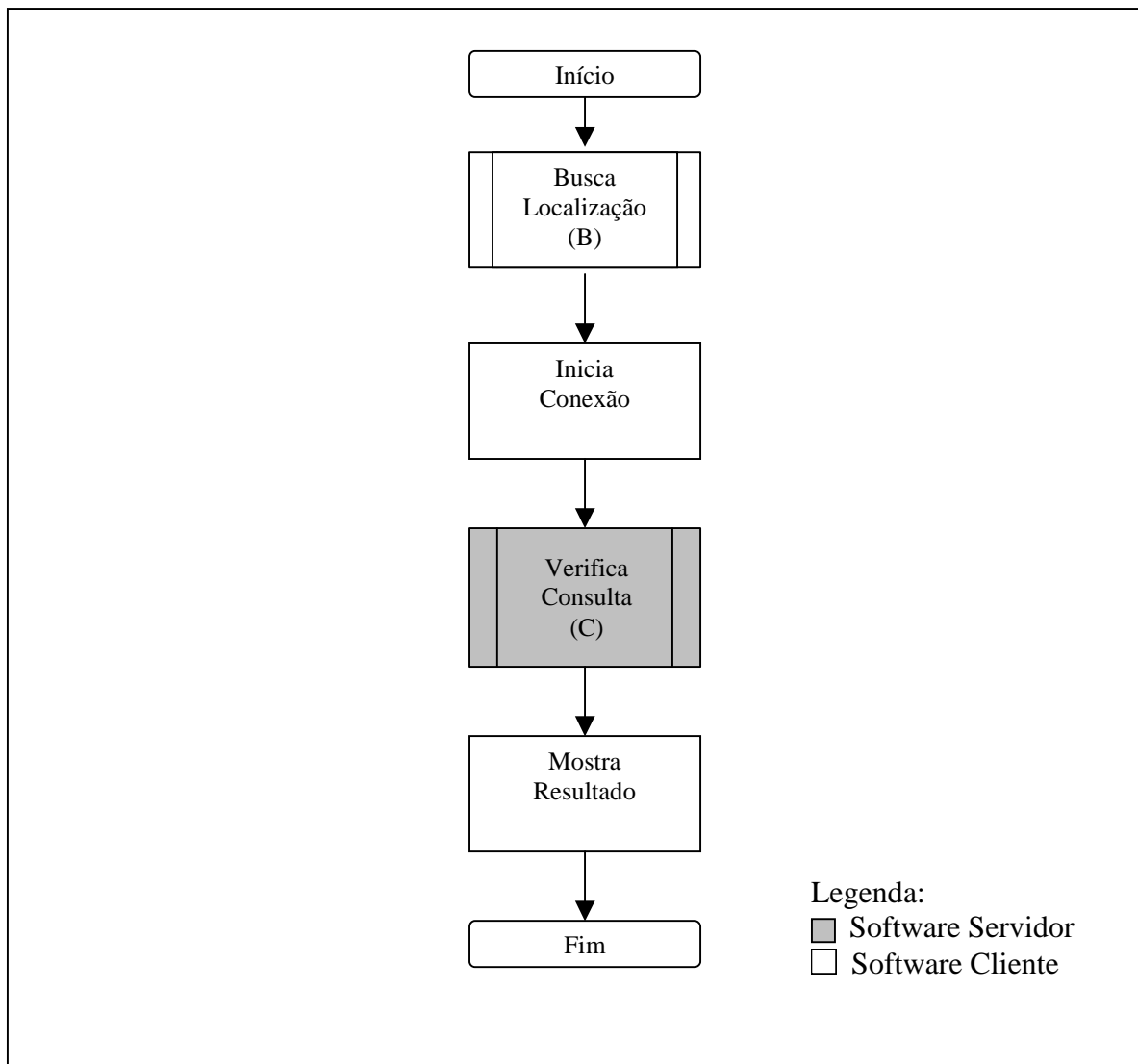


FIGURA 10: Fluxograma do processo A

A figura 11 apresenta o detalhamento do processo B, que é executado no software cliente. Nele é determinada a localização do usuário através da conexão com o GPS.

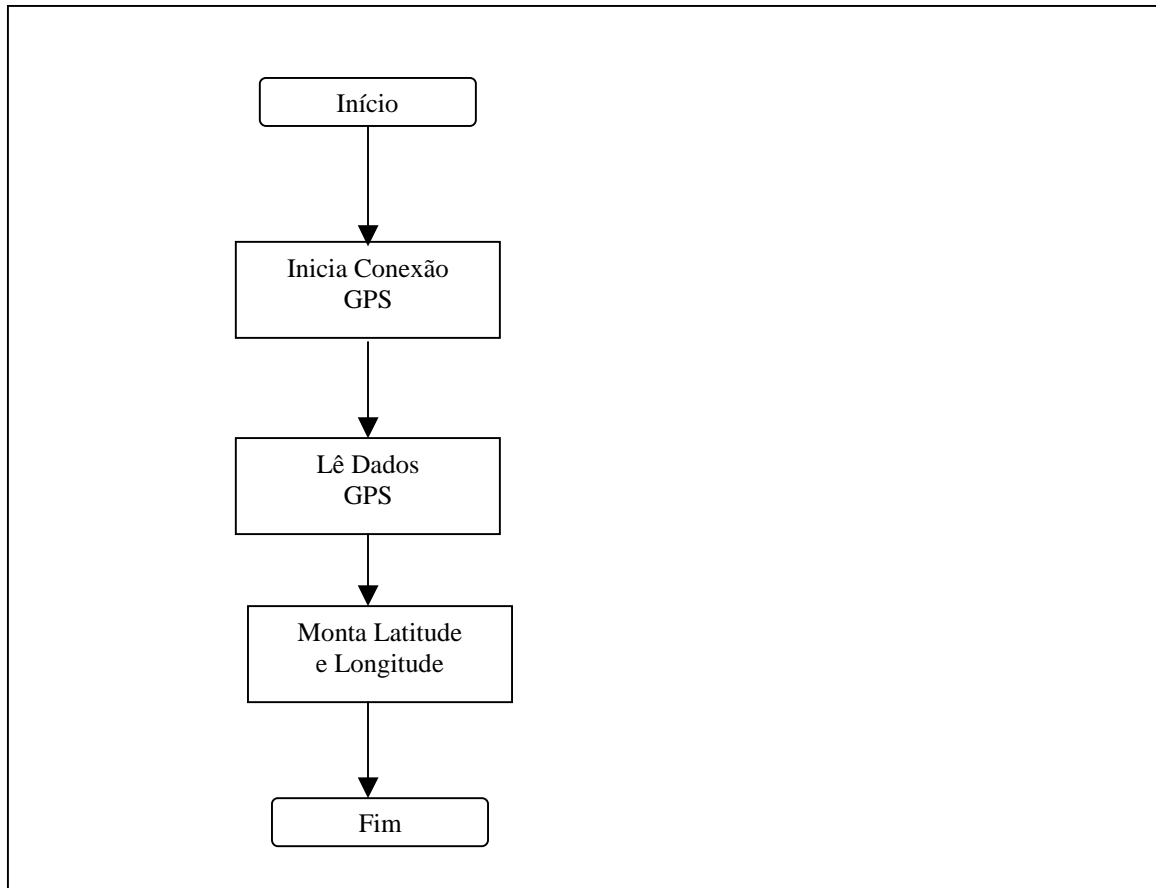


FIGURA 11: Fluxograma do processo B

A figura 12 apresenta o detalhamento do processo C, que é executado no software servidor. Nele é feita uma verificação da pesquisa, validando os parâmetros recebidos do software cliente. Caso os parâmetros estejam corretos, o processo D é chamado. Este processo é descrito ainda neste capítulo.

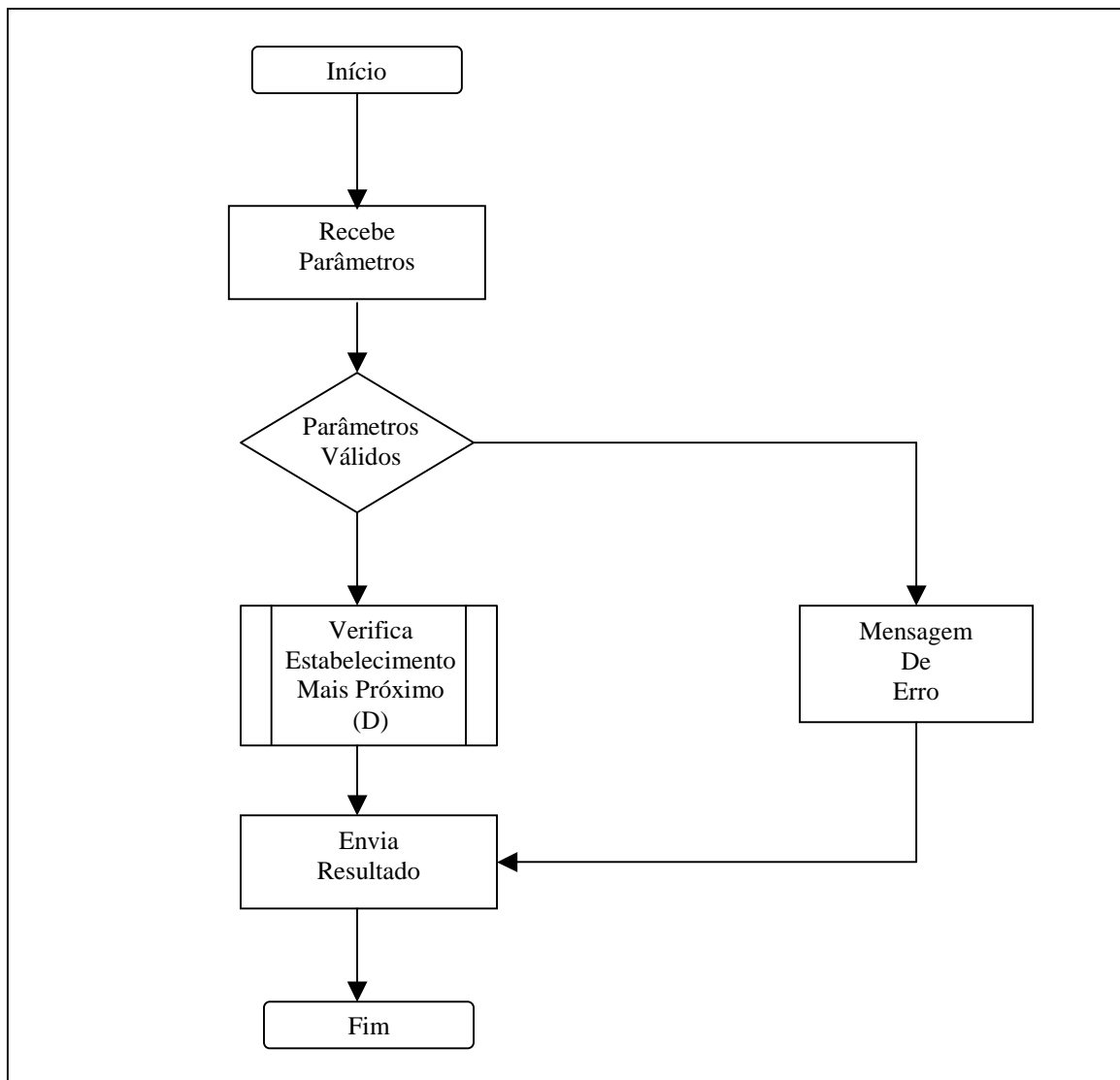


FIGURA 12: Fluxograma do processo C

A figura 13 apresenta o detalhamento do processo D, que é executado no software servidor. Este processo é responsável por pesquisar o estabelecimento mais próximo da posição pesquisada. Nele, um laço é responsável por percorrer por todos os estabelecimentos cadastrados no sistema, fazendo-se um cálculo das diferenças entre a latitude e longitude pesquisadas com a latitude e longitude dos estabelecimentos do sistema. O resultado apresenta o estabelecimento que obter a menor diferença (menor distância).

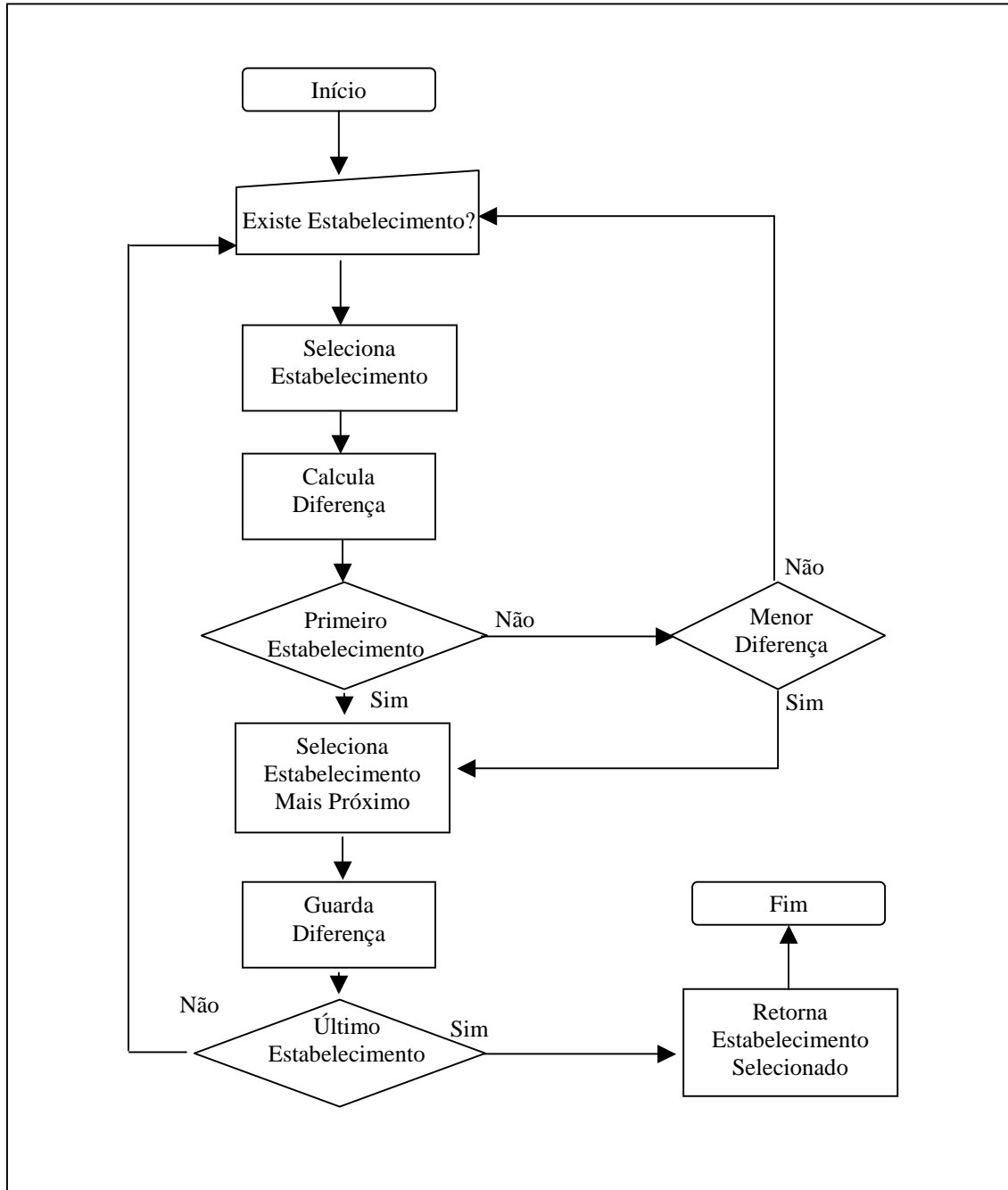


FIGURA 13: Fluxograma do processo D

### 4.3 IMPLEMENTAÇÃO

Para ilustrar o desenvolvimento dos protótipos, optou-se por criar um estudo de caso: consulta à farmácia mais próxima. Para tanto, foram cadastradas as latitudes e longitudes de

três farmácias localizadas na região da FURB: Farmácia Essencial, Farmácia do Big e Farmácia Camila.

Estas farmácias estão cadastradas no software servidor em vetor chamado *farmacias*. Nele um *Servlet* chamado *Consulta*, fica aguardando requisições que representam as consultas originadas pelo dispositivo móvel. Ainda no software servidor, criou-se a classe *Farmacia*, que serve para delimitar em um único objeto as informações da posição (latitude e longitude) e do nome das farmácias.

Para o software cliente, determinar a localização, criou-se uma conexão com a porta COM1 (através da RS232), onde é conectado o receptor GPS. Por esta conexão é passado um conjunto de 56 caracteres no formato TEXT OUT pré-estabelecido pelo receptor GPS Garmin ETREX, onde o carácter da posição 14 indica se é latitude Norte (N) ou Sul (S) e os caracteres da posição 15 a 21 determinam o valor em graus da latitude. O carácter da posição 22 indica se é longitude Leste (E) ou Oeste (W) e os caracteres da posição 23 a 30 determinam o valor em graus da longitude.

O software cliente é representado por um MIDlet chamado *LocaFarmaMIDlet*. Ele está presente no dispositivo móvel, e é responsável por coletar as informações da latitude e longitude do dispositivo, que são gravadas respectivamente nos atributos de classe *lat* e *lon* após a execução do método *makePoints*. Este método é responsável por determinar no conjunto de caracteres recebido do GPS qual o valor da latitude a ser gravado no atributo *lat* e qual o valor da longitude a ser gravado no atributo *lon*. Com as informações de localização, o software cliente faz uma pesquisa ao software servidor passando a latitude e longitude como parâmetros. O retorno desta pesquisa mostra qual a farmácia mais próxima da localização informada.

#### 4.3.1 Ferramentas utilizadas

Para a implementação do software cliente foi utilizada a linguagem J2ME, com a configuração CLDC 1.1 e o perfil MIDP 2.0, e o ambiente de desenvolvimento J2ME Wireless Toolkit v.2.0\_01, representado pela figura 14.



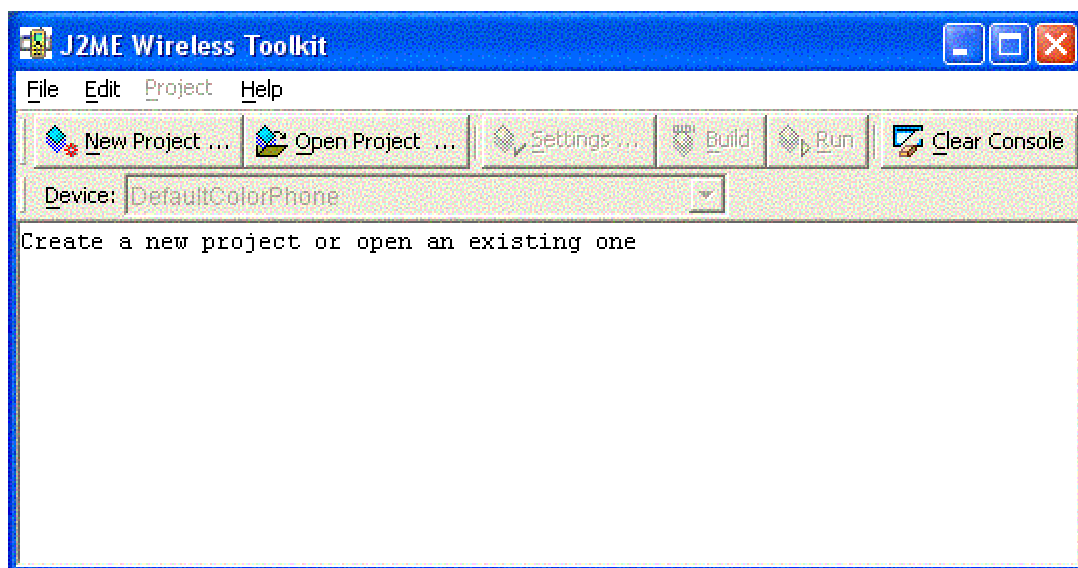


FIGURA 14: J2ME Wireless Toolkit

O J2ME Wireless Toolkit requer o J2 SDK, Standard Edition, instalado. A versão utilizada foi a 1.4.0. Esta ferramenta serve como ambiente para a compilação e a execução de aplicações J2ME através de emuladores de dispositivos móveis. O emulador DefaultColorPhone foi o escolhido para o desenvolvimento do protótipo do software cliente e está representado na figura 15. Apesar de apresentar o emulador, a ferramenta não contém um editor de texto integrado, sendo necessário utilizar um editor convencional ou algum outro produto especializado para escrever o código java. Esta ferramenta apresenta para cada projeto criado uma tela para configuração, onde podem ser estabelecidos diversos parâmetros, inclusive parâmetros que podem ser utilizados pelo software durante sua execução.



FIGURA 15: Emulador DefaultColorPhone

Para o desenvolvimento do software servidor foi utilizada a ferramenta JDeveloper da Oracle, apresentada na figura 16. Esta ferramenta pode ser utilizada com fins de pesquisa estudantil, mas não pode ser utilizada para fins comerciais sem que se compre uma licença. A escolha pelo JDeveloper deve-se principalmente a duas razões: a familiaridade e a facilidade na depuração de aplicações web. Esta ferramenta contém o *container* OC4J embutido, onde fica executando o *Servlet* Consulta e a classe Farmacia, que são os dois fontes existentes no software servidor. O JDeveloper serviu também como editor para o software cliente, porém a compilação e a emulação deste foi totalmente realizado pelo Wireless Toolkit.

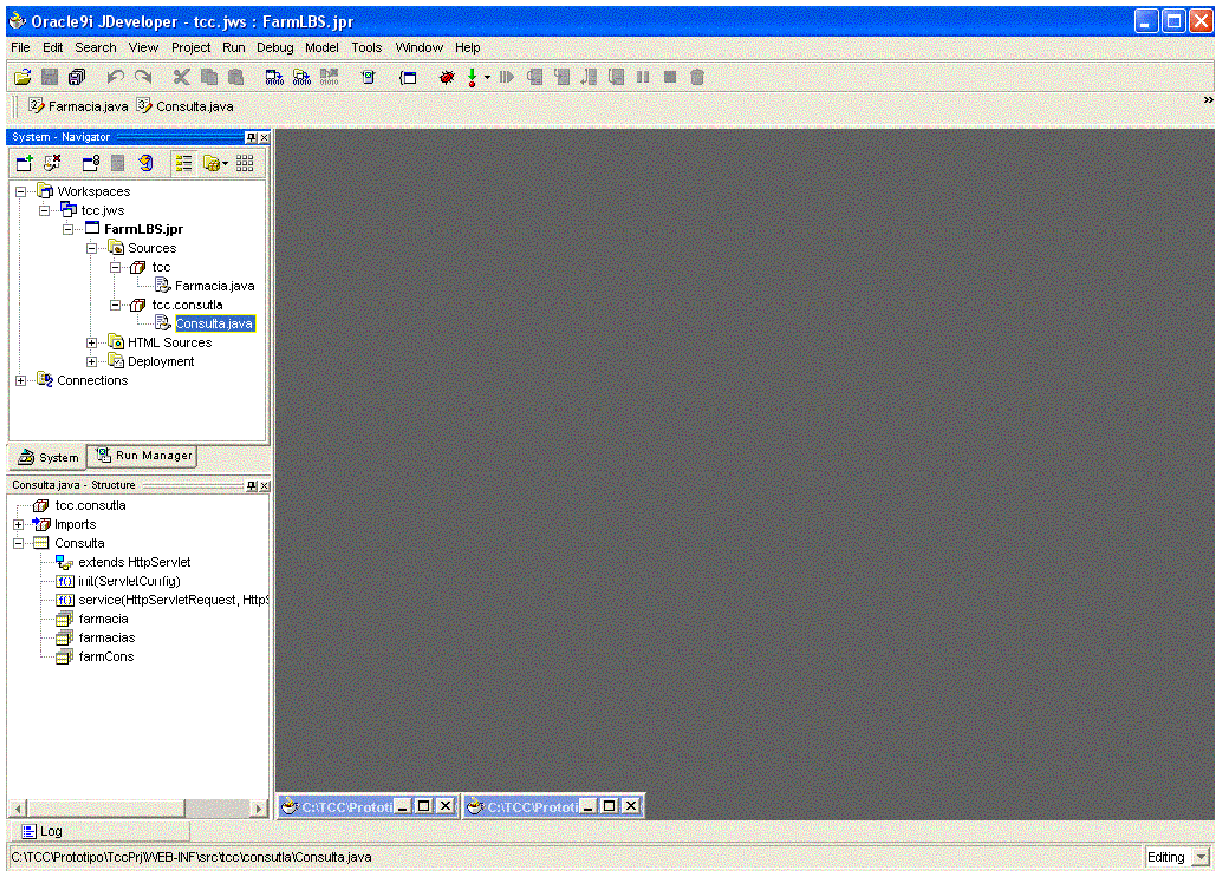


FIGURA 16: Ferramenta JDeveloper

#### 4.3.2 Principais funções do protótipo

As funções apresentadas nos quadros seguintes correspondem aos três sub-processos do protótipo.

No quadro 1 pode-se visualizar o código do processo A (Efetua Pesquisa). A especificação deste código-fonte pode ser vista na Figura 10.

```

private void connect() {
    HttpConnection hc = null;
    InputStream in = null;
    //URL definida na configuracao da aplicacao
    String url = getAppProperty("LocaFarmaMIDlet.URL");

    connectSerial();

    // Adiciona os parametros para Lat. e Lon.
    url += "posX="+lat+"&posY="+lon;
    mMessageItem.setText(url);
    try {
        // Inicia a conexao
        hc = (HttpConnection)Connector.open(url);
        // in: contem o retorno da requisicao
        in = hc.openInputStream();

        int contentLength = (int)hc.getLength();
        byte[] raw = new byte[contentLength];
        int length = in.read(raw);

        // Mostra a resposta ao usuario.
        String s = new String(raw, 0, length);

        in.close();
        hc.close(); // Fecha a conexao http.

        mMessageItem.setText(s);
    }
    catch (IOException ioe) {
        mMessageItem.setText(ioe.toString());
    }

    mDisplay.setCurrent(mMainForm);
}

```

QUADRO 1: Implementação do processo A (Efetua Pesquisa)

No quadro 2 pode-se visualizar o código do processo B (Busca Localização). A especificação deste código-fonte pode ser vista na figura 11.

```

private void connectSerial()
{
    CommConnection cc = null;
    InputStream is = null;
    StringBuffer inputBuffer = new StringBuffer();
    int data = 0;

    try{
        cc = (CommConnection) Connector.open
            ("comm:com1;baudrate=9600;bitsperchar=8;parity=none");

        is = cc.openInputStream();

        while (data != -1) {
            try {
                data = is.read();
                if (data == -1) {
                    break;
                }
                if ('\r' == (char)data) {
                    break;
                } else {
                    inputBuffer.append((char)data);
                }
            } catch (IOException ex) {
                System.err.println(ex);
                return;
            }
        }

        is.close();
        cc.close(); // Fecha a conexao serial.

        makePoints(inputBuffer.toString());

    } catch (IOException ioe) {
        System.out.println("Erro: " + ioe.toString());
        jMenuItem.setText(ioe.toString());
    }
}

```

QUADRO 2: Implementação do processo B (Busca Localização)

No quadro 3 pode-se visualizar o código do processo C (Verifica Pesquisa). A especificação deste código-fonte pode ser vista na figura 12.

```
farmCons = new Farmacia();

String message = "A Farmácia mais próxima é:\n";
float posX = 0;
float posY = 0;

String posXstr = req.getParameter("posX");
String posYstr = req.getParameter("posY");

try{
    posX = Float.parseFloat(posXstr);
    posY = Float.parseFloat(posYstr);

    message += farmCons.farmaciaMaisProxima(farmacias,posX,posY);
} catch (NumberFormatException e) {
    message = "Erro: Os campos devem ser preenchidos apenas com
números.";
}

PrintWriter out = resp.getWriter();
resp.setContentType("text/plain");
resp.setContentLength(message.length());

out.println(message);
```

QUADRO 3: Implementação do processo C (Verifica Pesquisa)

No quadro 4 pode-se visualizar o código do processo D. A especificação deste código-fonte pode ser vista na figura 13.

```

/**
 * Soma as diferenças entre a latitude e a longitude de cada farmacia
 * em relação latitude e longitude pesquisadas.
 * A menor diferença representará a farmácia mais próxima.
 *
 * @param newFarm - vetor com as farmacias cadastradas
 * @param newPesqLatitude - latitude a ser pesquisada
 * @param newPesqLongitude - longitude a ser pesquisada
 *
 * @return a farmacia mais próxima da latitude/longitude pesquisada
 */
public String farmaciaMaisProxima(Vector newFarm,
                                   float newPesqLatitude,
                                   float newPesqLongitude)
{
    String nmFarmaciaMaisProx = "Pesquisa não retornou valores!";
    Farmacia farmacia = null;
    float menorDiferenca = 0,
          diferenca = 0;

    for (int i= 0 ; i < newFarm.size(); i++ )
    {
        farmacia = (Farmacia) newFarm.get(i);
        diferenca = Math.abs( farmacia.getLatitude() - newPesqLatitude );
        diferenca += Math.abs( farmacia.getLongitude() - newPesqLongitude
    );

        if ( i == 0 )
        {
            menorDiferenca = diferenca;
            nmFarmaciaMaisProx = farmacia.getNmFarmacia();
        } else {
            if ( diferenca < menorDiferenca )
            {
                menorDiferenca = diferenca;
                nmFarmaciaMaisProx = farmacia.getNmFarmacia();
            }
        }
    }

    return nmFarmaciaMaisProx;
}

```

QUADRO 4: Implementação do processo D (Verifica Estabelecimento Mais Próximo)

#### 4.3.3 Operacionalidade do software cliente

Basicamente, o software cliente tem duas telas:

- a) pesquisar farmácia: esta tela permite ao usuário efetuar pesquisas pelo estabelecimento mais próximo. Na opção Pesquisar, o protótipo obtém a localização do GPS e faz a consulta na web, passando estas informações como parâmetro para o *Servlet* Consulta;

- b) *ok to send information*: esta é uma tela padrão quando se está prestes a conectar na internet ou ao GPS. Ela serve para alertar ao usuário que o serviço pode ser pago e determina quatro ações possíveis:
- Yes, always. Don't ask again*: nesta opção o usuário aceita o serviço e faz com que na próxima pesquisa a tela não apareça,
  - This time. Ask me next time*: nesta opção o usuário aceita o serviço e pede para ser questionado novamente na próxima pesquisa,
  - Not this time. Ask me next time*: nesta opção o serviço não é autorizado e é solicitado que seja questionado novamente numa próxima pesquisa,
  - No. Shut off LocaFarma*: nesta opção o serviço não é autorizado e o usuário sai do sistema.

A figura 17 apresenta a tela para pesquisar farmácias. As opções disponíveis nos botões são:

- pesquisar: conecta-se à Internet enviando as informações da latitude e longitude obtida do GPS. Após o clique e a confirmação de conexão, o resultado da consulta é apresentado na tela;
- sair: saída do sistema.





FIGURA 17: Tela para pesquisa de farmácias

O botão SELECT serve para selecionar os campos e suas setas direcionais para a navegação entre os mesmos.

A figura 18 apresenta a tela de pesquisa de farmácia após a execução da consulta.



FIGURA 18: Resultado de uma consulta

A figura 19 apresenta a tela de com uma mensagem de erro.



FIGURA 19: Consulta retornando uma mensagem de erro

A figura 20 apresenta a tela para confirmação do serviço.

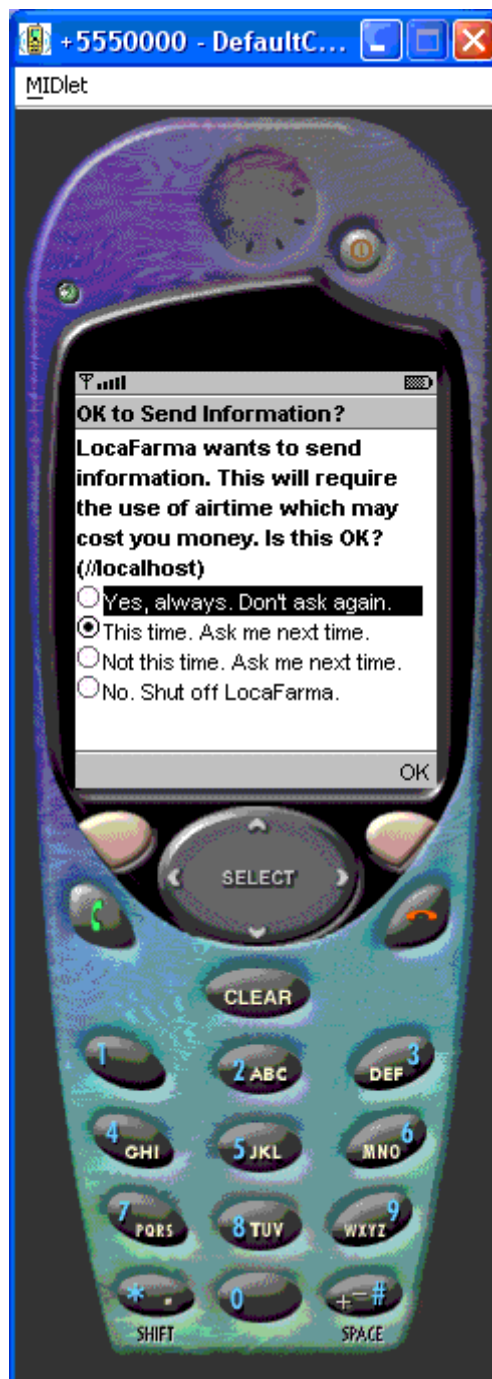


FIGURA 20: Confirmação do serviço

#### 4.4 TESTES E VALIDAÇÃO

Nos testes aplicados aos protótipos foram feitas várias pesquisas passando valores numéricos positivos e negativos e foram avaliadas também, as possibilidades de entrada de valores inválidos.

Para a realização dos testes, foi usado como exemplo a simulação com os dados da tabela 2. Nela são destacadas 3 farmácias com seus respectivos nomes e localizações. Onde cada um dos dados referentes à localização foi obtido manualmente, visitando-se cada estabelecimento e armazenando sua localização. Em cada visita, obteve-se também pontos próximos aos estabelecimentos, possibilitando assim, a realização de testes de validação, demonstrados ainda neste capítulo.

Tabela 2: Farmácias e suas respectivas localizações

Nome	Latitude	Longitude
Farmácia Essencial	26°54'36'' Sul	049°04'52'' Oeste
Farmácia Big	26°54'41,5'' Sul	049°04'53,4'' Oeste
Farmácia Camila	26°54'16,3'' Sul	049°05'34,7'' Oeste

Na figura 21 observa-se a pesquisa pelos pontos de localização da Farmácia Essencial.



Na figura 22 observa-se a consulta por valores que não constam na tabela 2, porém indicam que dentre as opções existentes, a farmácia mais próxima é a Farmácia Big.









FIGURA 24: Pesquisa com valores inválidos

Os testes apresentados nesta seção validam a implementação dos protótipos especificados e comprovam sua eficácia. Desta forma, os objetivos específicos do presente trabalho, descritos na seção 1.1, foram atingidos.

## 5 CONCLUSÕES

O presente trabalho permitiu a realização de um estudo sobre a tecnologia J2ME, verificando suas características e conceitos. Permitiu ainda usufruir de seu *framework* de conectividade para efetuar a comunicação de um MIDlet com um *Servlet*, mostrando que a união entre a internet e os dispositivos móveis é uma forma de ampliar as funcionalidades dos aplicativos. Esta ampliação pôde ser constatada também através da comunicação do MIDlet com o GPS, que uniu duas tecnologias e mostrou a utilidade que isto pode dar a sistemas embarcados.

Durante o decorrer dos estudos observou-se que desenvolver aplicações *wireless* para dispositivos sem fio apresenta muitos desafios aos programadores, projetistas, arquitetos e engenheiros de software, pois as arquiteturas destes dispositivos são bem mais limitadas que as de um PC. E além desta limitação, existe ainda o fato de que a interação com a internet através de um dispositivo móvel é totalmente diferente de com um computador pessoal, pois as atividades realizadas com um celular são fundamentalmente diferentes das realizadas em frente de um PC.

Observou-se que as tecnologias de comunicação móvel estão dando um grande salto evolutivo. Visto que no início, quando surgiram os primeiros celulares analógicos, sua funcionalidade se resumia praticamente a efetuar ligações, hoje em dia, com o advento da 3G, o celular continua tendo esta função, mas em muitos casos ela está deixando de ser a mais importante. Está tendo que concorrer com funções de vídeo-conferência, jogos em 3D, câmeras de vídeo, chaves de automóveis, entre outras.

As ferramentas utilizadas foram adequadas para a obtenção do resultado, citando como exemplo a linguagem Java e o *container* de aplicações OC4J. A ferramenta utilizada para auxiliar no desenvolvimento das rotinas em Java foi a JDeveloper da ORACLE. Esta ferramenta satisfaz as necessidades. O Wireless Toolkit também se mostrou um boa ferramenta para a compilação e emulação do MIDlet, mas o fato de não possuir um ambiente de edição trouxe grandes dificuldades na depuração do aplicativo.

Os protótipos desenvolvido neste trabalho atingiram seus objetivos. Neles foi possível estabelecer a comunicação entre o software do celular e o GPS, baseando-se numa conexão serial e uma comunicação com um Servlet, baseando-se numa conexão HTTP. O uso destas

duas conexões possibilitou um bom estudo sobre o *framework* de conectividade do J2ME. Através da comunicação do software do celular com o GPS, para buscar a localização, e posterior comunicação com o *Servlet* o protótipo pôde cumprir seus requisitos e mostrar o conceito de uma aplicação de serviços baseados em localização.

### 5.1 DIFICULDADES ENCONTRADAS

Foram encontradas algumas dificuldades no decorrer deste trabalho, dentre as quais pode-se destacar:

- a) pouca bibliografia sobre J2ME e LBS;
- b) dificuldade de depuração do protótipo;
- c) falta de um ambiente de desenvolvimento gratuito;
- d) impossibilidade de aquisição do cabo de comunicação de dados DKU-5 da Nokia, ocasionado pela falta de tempo disponível, pois o produto deve ser importado.

### 5.2 LIMITAÇÕES

Os protótipos apresentam algumas limitações, dentre as quais pode-se destacar:

- a) o protótipo do software cliente não apresenta o resultado das pesquisas de forma gráfica, apenas de forma textual;
- b) como o objetivo do trabalho não era coletar uma grande base de dados sobre a localização dos estabelecimentos, o protótipo do software servidor não possui uma base de dados em um banco de dados, pois as informações georeferenciadas são poucas, e não necessitaram de um armazenamento físico.

### 5.3 EXTENSÕES

Sugere-se como extensão deste trabalho uso do aplicativo LBS para consultar as cotas de enchentes de Blumenau, o que poderia ser realizado em parceria com o Instituto de Pesquisas Ambientais da FURB.

Ainda como sugestão, o protótipo do software servidor poderia ser dotado de uma lógica de busca de melhor caminho, o que agregaria mais valor ao aplicativo e o deixaria mais completo. Dando a opção de apresentar os resultados obtidos também através de imagens.

Para as sugestões acima, seria interessante ter uma boa base de dados georeferenciados. A partir desta, os protótipos podem apresentar mais opções de consulta, aumentando assim a funcionalidade do aplicativo.

## REFERÊNCIAS BIBLIOGRÁFICAS

CÂMARA, G. *et al.* **Anatomia de sistemas de informação geográfica**. Rio de Janeiro: SBC, 1996. Disponível em : <<http://www.dpi.inpe.br/gilberto/livro/anatomia.pdf>>. Acesso em: 03 jun. 2003.

CÂMARA, Gilberto. **Perspectivas ao norte do equador**, Savannah, out. 2000. Disponível em: < <http://www.giscience.org/GIScience2000/camara.html> >. Acesso em: 15 mai. 2003.

CONCEIÇÃO, Arlindo Flávio da. **Location based services**. [S.l.],[2001]. Disponível em: <<http://www.ime.usp.br/~arlindo/slides/lbs.ppt>>. Acesso em: 28 set. 2003.

CORBERA, Rodrigo Garcia. **Tutorial de programação J2ME**. [S.l.],[2003]. Disponível em: <[http://geocities.yahoo.com.br/brasilwireless/tutorial\\_j2me/j2me\\_01/j2me\\_01.html](http://geocities.yahoo.com.br/brasilwireless/tutorial_j2me/j2me_01/j2me_01.html)>. Acesso em: 8 out. 2003.

CARDOZO, André. O mapa das tecnologias. **Info Exame**, São Paulo, v.1, n.212,p.23, set. 2003.

DEITEL, H. M. **Internet & world wide web**, como programar. Porto Alegre: Bookman, 2003.

DEITEL, H. M. **Java**, como programar. Porto Alegre: Bookman, 2001.

DEPINÉ, Fábio Marcelo. **Protótipo de software para dispositivos móveis utilizando Java ME para cálculo de regularidade em rally**. 2002. 55 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FIELDS, Duane. **Desenvolvendo na Web com Java Server Pages**. Rio de Janeiro: Ciência Moderna, 2000.

FONSECA, Jorge Cavalcanti. **Portando a KVM**. 2002. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Informática, Universidade Federal de Pernambuco, Recife.

GATTASS, Marcelo. **Servlets e COM para a Visualização de Dados Geográficos na Web**, Rio de Janeiro, [2000]. Disponível em: <[www.dpi.inpe.br/nsf-cnpq/servlets.pdf](http://www.dpi.inpe.br/nsf-cnpq/servlets.pdf)>. Acesso em: 8 out. 2003.

GREGO, Maurício. Do telefone tijolo ao 3G. **Info Exame**, São Paulo, v.1, n.212,p.23, set. 2003.

JÚNIOR, Edvaldo Simões da Fonseca. **Global positioning system**. São Paulo, [2003]. Disponível em: <[http://www.ptr.usp.br/Cursos/SIG\\_GPS/gps/home.htm](http://www.ptr.usp.br/Cursos/SIG_GPS/gps/home.htm)>. Acesso em 29 set. 2003.

KROLL, Michael; HAUSTEIN, Stefan. **Java 2 micro edition application development**. Indianápolis: Sams, 2002.

MIRANDA, Cláudio. Conectividade com MIDP. **Java Magazine**, Rio de Janeiro, v.1, n.6,p.23, mai. 2003.

ORACLE. **Java programming**. World Headquarters: Oracle Corporation, 2002.

QUALCOMM. **Snap Track**. 2003. Disponível em: <<http://www.snaptrack.com/index.jsp>>. Acesso em: 04 out. 2003.

REGGIANI, Lúcia. Admirável mundo sem fio. **Info Exame**, São Paulo, v.1, n.212,p.23, set. 2003.

ROCHA, Mauro Nacif. **Serviços baseados em localização**. Viçosa, [2001]. Disponível em: <<http://www.dpi.ufv.br/~nacif/cmovei/Aula16.pdf>>. Acesso em: 04 out. 2003.

SILVA, Miguel Reis Castilho da. **Uma introdução sobre sistemas de localização**. Évora, [2003]. Disponível em: <<http://alunos.uevora.pt/~112058/es/t3/>>. Acesso em: 04 out. 2003.

SIMPSON, Stu; D'ANTONI Ed. **Global positioning system ou simplesmente GPS**. [S.l.],[1997]. Disponível em: <<http://www.abordo.com.br/~lincoln/>>. Acesso em: 29 set. 2003.

SOUJAVA. **Sociedade de usuários Java**. [S.l.], [2003]. Disponível em: <<http://www.soujava.org.br/>>. Acesso em: 11 out. 2003.

SUN. **Java 2 plataforma**, micro edition. [S.l.],[2003]. Disponível em: <<http://java.sun.com/j2me>>. Acesso em: 11 out. 2003.

ZURSTRASSEN, Leonardo. **Location based services**. [S.l.],[2003]. Disponível em: <[http://www.wirelessbrasil.org/wirelessbr/colaboradores/zurstrassen/lbs\\_01.html](http://www.wirelessbrasil.org/wirelessbr/colaboradores/zurstrassen/lbs_01.html)>. Acesso em: 28 set. 2003.

WEBRASKA. **Apontador Duo**. 2003. Disponível em: <<http://www.apontadorduo.com.br/>>. Acesso em: 30 ago. 2003.