



Universidade Federal de Pernambuco  
Centro de Informática

Graduação em Ciência da Computação

## **Utilizando o Conceito de Localização em Jogos Móveis**

Renato César Ferraz Marcolino Bezerra

Trabalho de Graduação

Recife  
2007

Universidade Federal de Pernambuco  
Centro de Informática

Renato César Ferraz Marcolino Bezerra

## **Utilizando o Conceito de Localização em Jogos Móveis**

*Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.*

Orientador: *Carlos André Guimarães Ferraz*

Recife  
2007

# Assinaturas

Este Trabalho de Graduação é resultado dos esforços do aluno Renato César Ferraz Marcolino Bezerra, sob a orientação do professor Carlos André Guimarães Ferraz, na participação do projeto *Utilizando o Conceito de Localização em Jogos Móveis*, conduzido pelo Centro de Informática da Universidade Federal de Pernambuco. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

---

Renato César Ferraz Marcolino Bezerra

---

Carlos André Guimarães Ferraz

*À minha família*

# Agradecimentos

Gostaria de agradecer primeiramente à minha família. Aos meus pais, Deusdeth e Margarida, que me incentivaram do começo ao fim do curso, e aos meus irmãos, Ana e Deusdeth, que me ajudaram sempre que precisei e foram compreensivos quando monopolizei o computador para concluir este trabalho.

Também gostaria de agradecer à Rannuzia, minha namorada, pelo amor, companheirismo e por agüentar o meu mau humor durante a reta final do curso.

Ao meu orientador Carlos Ferraz, pelo incentivo e confiança depositados em mim e pelo suporte dado durante o desenvolvimento desse trabalho.

A Fernando Trinta, por ter me repassado parte dos resultados da sua tese de doutorado.

Também gostaria de agradecer aos meus amigos de faculdade, Lucindo, Pablo, Thierry, Renê, Assis, Vitor, Padovan e Goiaba, que no decorrer da graduação, participaram de vários projetos comigo.

E finalmente, aos meus ex-colegas de estágio, Alan, Alexandre, Thiago e Rubens, e em especial a Sebastião, que me convenceu a tomar a difícil decisão de deixar o estágio, tornando assim possível a conclusão deste trabalho.

# Resumo

Serviços Baseados em Localização (LBS) constitui uma área relativamente nova da computação, sendo impulsionada nos últimos anos principalmente devido ao rápido desenvolvimento de tecnologias que lhe dão suporte (computação móvel, redes sem fio e tecnologias de localização) e da crescente demanda comercial. A união entre os LBS e os jogos móveis deu origem aos chamados jogos baseados em localização. Estes jogos introduzem uma nova forma de interação entre os seus jogadores, onde estes interagem com o mundo virtual mudando a sua localização real, também podendo se encontrar e interagir uns com os outros no contexto do jogo. Este trabalho propõe o desenvolvimento de um serviço de localização para um *middleware* de suporte a jogos multiusuário multiplataforma, onde os jogadores móveis poderão utilizar sua localização física como um elemento integrante da dinâmica destes jogos.

**Palavras-chave:** Ciência de Contexto, Serviços Baseados em Localização, Jogos Baseados em Localização, *Middlewares* para Jogos

# Sumário

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>                          | <b>1</b>  |
| 1.1      | Objetivo                                   | 2         |
| 1.2      | Organização do Trabalho                    | 3         |
| <b>2</b> | <b>Ciência de Contexto em Jogos Móveis</b> | <b>4</b>  |
| 2.1      | Definição de Contexto                      | 4         |
| 2.2      | Ciência de Contexto                        | 5         |
| 2.3      | A Localização como Informação de Contexto  | 5         |
| 2.4      | Técnicas de Localização                    | 7         |
| 2.5      | Jogos Móveis                               | 8         |
| 2.6      | Jogos Baseados em Localização              | 9         |
| <b>3</b> | <b>Trabalhos Relacionados</b>              | <b>10</b> |
| 3.1      | Middleware de Suporte PM2G                 | 10        |
| 3.2      | MoCA                                       | 13        |
| 3.3      | Nimbus                                     | 15        |
| <b>4</b> | <b>Serviço de Localização</b>              | <b>17</b> |
| 4.1      | Modelagem                                  | 18        |
| 4.2      | Arquitetura Proposta                       | 20        |
| <b>5</b> | <b>Experimentos</b>                        | <b>26</b> |
| 5.1      | Prototipação do Cliente                    | 26        |
| 5.2      | Prototipação do Servidor                   | 28        |
| 5.3      | Descrição dos Experimentos                 | 29        |
| 5.3.1    | Detectar Adversários Próximos              | 29        |
| 5.3.2    | Detectar Áreas Mapeadas (AMs)              | 29        |
| 5.3.3    | Enviar Mensagem para Adversários Próximos  | 30        |
| <b>6</b> | <b>Conclusões</b>                          | <b>31</b> |
| 6.1      | Trabalhos Futuros                          | 32        |
| 6.2      | Considerações Finais                       | 32        |

# Lista de Figuras

|      |   |    |
|------|---|----|
| 3.1  | Arquitetura de Serviços PM2G  | 11 |
| 3.2  | MoCA ( <i>Mobile Collaboration Architecture</i> )   | 13 |
| 3.3  | <i>Framework Nimbus</i>   | 16 |
| 4.1  | Modelo de Contexto PM2G [Tri07]   | 18 |
| 4.2  | Estados de um jogador [Tri07]   | 19 |
| 4.3  | Diagrama de Classes - Classes relacionadas ao Contexto  | 19 |
| 4.4  | Diagrama de Seqüência - Serviço de Localização  | 20 |
| 4.5  | Arquitetura do Serviço de Localização   | 21 |
| 4.6  | Diagrama de Seqüência - Recebimento de uma notificação de mudança de contexto pelo Serviço de Localização | 22 |
| 4.7  | Diagrama de Seqüência - Detecção de jogadores próximos  | 23 |
| 4.8  | Diagrama de Classe - Eventos, e criadores de mensagens  | 24 |
| 4.9  | Configuração dos Eventos e Factories utilizados pelo Serviço de Localização                               | 24 |
| 4.10 | Diagrama de Seqüência - Envio de mensagem para adversários próximos                                       | 25 |
| 5.1  | Cliente: Telas de logon e alteração de status   | 27 |
| 5.2  | Cliente: Tela principal, jogadores próximos e enviar mensagem   | 28 |
| 5.3  | Servidor: Visualizar mapa   | 29 |

## CAPÍTULO 1

# Introdução

O aumento da popularidade das tecnologias móveis, a partir da proliferação de redes sem fio e do surgimento de dispositivos portáteis cada vez mais leves, com mais recursos e melhor capacidade de comunicação e processamento, fez surgir um aumento na demanda por serviços cada vez mais diversificados [RES<sup>+</sup>04] [dOJ05].

A oferta de serviços que levem em consideração o contexto<sup>1</sup> no qual o usuário está inserido, como os Serviços Baseados em Localização (*Location Based Services* - LBS), vem tornando possível o desenvolvimento de aplicações com a capacidade de adequar o seu funcionamento às necessidades dos usuários. Esta capacidade de detectar, interpretar e responder a aspectos do ambiente é conhecida como Ciência de Contexto [Wie05].

Serviços Baseados em Localização constitui uma área relativamente nova da computação, sendo impulsionada nos últimos anos devido ao rápido desenvolvimento de tecnologias que lhe dão suporte (computação móvel, redes sem fio e tecnologias de localização) e da crescente demanda comercial [Gon05].

A união entre os LBS e os jogos móveis deu origem aos chamados jogos baseados em localização. Estes jogos, ao contrário dos jogos tradicionais, que são jogados em frente a computadores ou consoles de videogame e utilizam o teclado ou *joystick* como dispositivo de entrada, necessitam que os jogadores se movam no mundo real. Os jogadores interagem com o jogo mudando a sua posição e visitando lugares de interesse do jogo. Esses jogadores podem se encontrar no mundo real e interagir uns com os outros no contexto do jogo [NPM01].

O sucesso de jogos como *Mogi* [Gam] e *Uncle Roy All Around You* [The] ilustra bem este novo fenômeno. No primeiro, o jogador caminha no mundo real coletando itens virtuais que aparecem em um mapa exibido na tela do seu telefone celular, também podendo ver, através do mesmo mapa, a localização de outros jogadores, que podem se encontrar fisicamente e negociar os objetos encontrados. Em *Uncle Roy All Around You*, o jogador tem de encontrar o misterioso

---

<sup>1</sup>Contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade [Dey01]. Maiores detalhes são apresentados no próximo capítulo.

*Uncle Roy* no máximo em 60 minutos, contando para isto com a ajuda de jogadores online, que enviam mensagens e instruções.

Uma outra tendência é o surgimento de jogos pervasivos ou ubíquos, que tirando proveito da visão de Computação Pervasiva ou Ubíqua, introduzido por Weiser [WGB99], estendem as experiências do jogo. Um jogo pervasivo pode ser definido como uma aplicação sempre presente, disponível aos jogadores através de diferentes dispositivos e redes [Tri07].

A concepção e o desenvolvimento destes jogos devem levar em conta a grande diversidade de plataformas, sendo necessária a sua adaptação às características do dispositivo utilizado pelo jogador em um dado instante. Uma das possíveis soluções para este problema é a criação de uma infra-estrutura que amenize o esforço requerido para o desenvolvimento destes jogos, reduzindo assim os seus custos e o tempo de desenvolvimento [dVP06].

Durante uma tese de doutorado no Centro de Informática, foi especificada e implementada uma arquitetura baseada em uma camada de *middleware* para jogos pervasivos multiusuário multiplataforma (*Pervasive Multiplayer Multiplatform Game* - PM2G) [Tri07]. Esta camada tem como objetivo agregar serviços de suporte relativos aos conceitos específicos desse tipo de jogo, como serviço de gerenciamento de contexto, facilitando assim a implementação e execução destes jogos.

## 1.1 Objetivo

O objetivo deste trabalho é realizar um estudo sobre computação ciente de contexto, serviços baseados em localização, jogos móveis e jogos baseados em localização, e, a partir do conhecimento adquirido, propor uma extensão para o *middleware* de suporte a jogos PM2G [Tri07] mencionado anteriormente.

Esta extensão, sob forma de um Serviço de Localização, permitirá que usuários de dispositivos móveis utilizem sua localização física, obtida via GPS (*Global Positioning System*) [Gov04], como um elemento integrante da dinâmica dos jogos. O foco deste trabalho está na utilização da localização do usuário em ambientes externos, entretanto, umas das sugestões de trabalhos futuros apresentados na conclusão é a sua utilização em ambientes internos, como locais que dispõem de uma rede sem fio, como redes IEEE 802.11 [oEE99].

Finalmente, de forma a validar o Serviço de Localização proposto, foram definidos cenários

que exploram algumas das possibilidades oferecidas por este tipo de serviço, e, a partir destes cenários, foi desenvolvido o protótipo de uma aplicação simples que faz uso da informação de localização física do usuário.

## **1.2 Organização do Trabalho**

Este trabalho está organizado da seguinte forma: O Capítulo 2 apresenta ao leitor os conceitos gerais relacionados à computação ciente de contexto, serviços baseados em localização, jogos móveis e jogos baseados em localização. No Capítulo 3 são apresentados alguns dos trabalhos relacionados encontrados durante o levantamento bibliográfico realizado. O Capítulo 4 descreve, em detalhes, o Serviço de Localização proposto. No Capítulo 5 são descritos os cenários e o protótipo desenvolvido para validar o Serviço de Localização. O Capítulo 6 finaliza este trabalho, apresentando as conclusões obtidas e as perspectivas de trabalhos futuros.

# Ciência de Contexto em Jogos Móveis

Antes que a ciência de contexto possa ser aplicada a jogos móveis, um claro entendimento dos diferentes conceitos relacionados deve estar estabelecido. Este capítulo está dividido em três partes. Primeiramente, são apresentados os conceitos relacionados à ciência de contexto. Em seguida, é apresentado o conceito de jogos móveis. E por fim, é apresentado o conceito de jogos baseados em localização, surgido a partir da união dos dois conceitos apresentados anteriormente.

## 2.1 Definição de Contexto

Na literatura podem ser encontradas diferentes definições para o termo contexto. Schilit e Theimer [ST94] o definem como localização, identidade de pessoas e objetos próximos e mudanças nesses objetos. Em uma definição similar, Brown et al. [BBC97] definem contexto como localização e identidade de pessoas próximas do usuário, adicionando as informações de hora, período e temperatura do ambiente. Ryan et al. [RPM98] definem contexto como localização e identidade do usuário e de objetos, informações físicas do ambiente e tempo.

Em [Dey01], é apresentada uma definição mais genérica para o termo, onde contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Esta entidade é uma pessoa, um lugar ou um objeto considerado relevante para a interação do usuário com a aplicação. Tal definição deixa a cargo do projetista da aplicação quais informações serão caracterizadas como contexto.

Por ser a mais abrangente e referenciada na literatura, essa definição será utilizada no decorrer deste trabalho.

Ainda segundo Dey e Abowd [DA99], os dados explicitamente fornecidos pelo usuário também podem ser considerados como informação de contexto. Por exemplo, é indiferente se a identidade de um usuário é detectada de algum modo automático ou informada ao se conectar ao sistema, pois esta informação é considerada contexto e pode ser utilizada para adaptar os

serviços do sistema às necessidades do usuário.

## 2.2 Ciência de Contexto

Computação ciente de contexto é um tema de pesquisa originário da área de computação ubíqua, que é um paradigma de interação usuário-computador em que a tecnologia é integrada de forma transparente a ambientes físicos, auxiliando as pessoas na realização de suas tarefas diárias de forma contínua e onipresente [Wei91].

O termo computação ciente de contexto (*context-aware computing*) foi definido por [ST94] como sendo o estudo de aplicações que se adaptam de acordo com a localização de uso, o conjunto de pessoas e objetos próximos, e as mudanças ocorridas com estes ao longo do tempo.

Alguns autores [SAW94] [BBC97] [DA99] [Dey01] destacam alguns grupos de informações contextuais, incluindo informações como localização, atividade, tempo e identidade. Tais grupos de informações contextuais, chamados de contexto primário, são considerados mais importantes porque, além de responderem a questões básicas para identificação de um contexto (o quê, quando, onde e quem), permitem a descoberta de informações adicionais.

## 2.3 A Localização como Informação de Contexto

A localização é a informação mais utilizada por aplicações cientes de contexto [P<sup>+</sup>03]. Um aspecto chave neste tipo de aplicação é como a informação de localização se relaciona com as ações a serem executadas por estas aplicações. Um exemplo clássico de aplicação que utiliza o conceito de localização são os guias turísticos digitais, que podem fornecer informações sobre pontos turísticos assim que o usuário se aproxima deles.

A inclusão da informação de localização em aplicações, sem que seja necessária a intervenção do usuário, cria uma nova classe de serviços, chamada de serviços baseados em localização [Wie05]. Serviços baseados em localização (*Location Based Services - LBS*) podem ser definidos como serviços que integram a posição ou a localização do dispositivo móvel com outras informações de forma a agregar mais valor às aplicações [Cor04].

Os serviços baseados em localização surgiram a partir da convergência de diversas tecnologias já existentes, tais como computação móvel, redes sem fio, tecnologias de localização e

Sistemas de Informações Geográficas<sup>1</sup> (SIG).

Quanto à sua aplicação, os serviços baseados em localização podem ser classificados em seis grupos básicos: [DdA04]:

- *Serviços de Informação*: podem fornecer instruções de como se chegar ao destino desejado, indicar onde encontrar um determinado estabelecimento ou pessoa, ou localizam usuários perdidos em um mapa.
- *Serviços de Tarifa Baseada na Localização*: permitem diferenciar as tarifas em função do local de uso do serviço. Operadoras de telefonia móvel, por exemplo, poderiam oferecer descontos em ligações efetuadas em locais pré-definidos pelo usuário, como, por exemplo, próximo ao seu local de trabalho.
- *Serviços de Emergência*: possibilitam um menor tempo de resposta a situações de emergência, como na área médica e de segurança, onde o fator tempo é determinante. A determinação da localização do evento permite o deslocamento da equipe mais próxima da sua ocorrência, diminuindo assim o tempo de atendimento.
- *Serviços de Rastreamento*: possui uma grande semelhança com a categoria anterior, porém, este serviço se diferencia por fazer um rastreamento contínuo, não dependendo de eventos externos para a sua ativação. Por ser bastante invasiva, esta categoria engloba aplicações onde a privacidade do usuário não é necessária ou desejada.
- *Serviços de Entretenimento*: jogos baseados em localização, como versões virtuais de "pique-esconde" e "caça ao tesouro", se encaixam nesta categoria. Também se encaixam nesta categoria aplicações com a finalidade de aproximar pessoas, que podem, por exemplo, emitir um aviso sobre a aproximação de um conhecido.
- *Serviços de M-Commerce*: enviam anúncios de ofertas e promoções aos celulares das pessoas que estão em um determinado raio de proximidade do anunciante, aumentando assim a eficiência dos anúncios e o potencial de venda.

Segundo dados publicados pelo Insight Research [Res], o mercado mundial de serviços baseados em localização deverá atingir 1,5 bilhões de dólares até o final do ano. Esta mesma publicação mostra que os serviços de localização já são bastante populares nos países asiáticos, onde as operadoras e fabricantes oferecem diversos tipos de produto.

---

<sup>1</sup>*Sistemas de Informações Geográficas* são sistemas destinados ao tratamento de dados referenciados espacialmente.

Existem várias técnicas diferentes que podem ser utilizadas para determinar a localização física de pessoas ou dispositivo, porém, em geral, nenhuma delas é auto-suficiente para atender de forma eficaz a qualquer tipo de ambiente. Na próxima seção serão apresentadas algumas das técnicas e tecnologias de apoio utilizadas na determinação de localização.

## 2.4 Técnicas de Localização

Uma técnica de localização nada mais é do que o processo utilizado para determinar a localização de um determinado dispositivo. O resultado desse processo pode ser representado em termos de coordenadas (latitude, longitude e altitude), pontos de referência (próximo a prédio conhecido) ou o nome de uma localidade (o nome de uma rua, por exemplo). Este resultado pode ser mais ou menos preciso, dependendo da técnica utilizada [Wie05].

Há um grande número de técnicas de localização existentes no mercado. Estas técnicas podem ser classificadas quanto ao local onde as coordenadas de posicionamento são coletadas e calculadas. São elas: as soluções baseadas em rede (*network-based*), as baseadas no dispositivo (*handset-based*) e soluções híbridas, que utilizam tanto a rede quanto o dispositivo para processarem os cálculos [LNN<sup>+</sup>06].

As técnicas de localização baseadas em rede permitem a utilização da infra-estrutura já existente, como as redes das operadoras de telefonia móvel e redes locais sem fio, para determinar a posição dos dispositivos. Nesta categoria, destacam-se as seguintes técnicas:

- *Cell-ID*: predominantemente usada nas redes GSM (*Global System for Mobile*), esta técnica identifica a célula na qual o usuário está conectado, e então mapeia esta informação em coordenadas, realizando assim a estimativa da localização do usuário.
- *Angle of Arrival (AOA)*: esta técnica usa um equipamento presente nas Estações Rádio Base (ERBs) para determinar o ângulo de chegada dos sinais de rádio. Com alguns cálculos geométricos, se pode determinar a localização do usuário com somente dois ERBs recebendo o sinal.
- *Time Difference of Arrival (TDOA)*: também conhecido como triangulação de antenas, esse método faz uso da diferença de chegada dos sinais de rádio emitidos pelo dispositivo às ERBs. É necessário pelo menos três ERBs para calcular a localização do usuário.

Nas técnicas de localização baseadas no dispositivo, o cálculo do posicionamento é feito

no próprio dispositivo móvel, que fornece a localização sem a ajuda de redes de comunicação móveis. O principal representante desta categoria é o GPS, um sistema de posicionamento desenvolvido para fins militares pelo Departamento de Defesa dos Estados Unidos.

O GPS utiliza uma rede de 24 satélites geo-estacionários para triangularizar a posição de um receptor e prover as coordenadas de latitude, longitude e altitude. Tal posicionamento pode ser implementado tanto via acessórios GPS, ou como uma solução integrada em um dispositivo móvel. Este sistema informa a localização com precisão de 15 a 100 metros, porém, em contrapartida, não funciona adequadamente em ambientes fechados.

Já as soluções híbridas, como o próprio nome sugere, unem as técnicas baseadas na rede e as técnicas baseadas no dispositivo. Nesta categoria, destaca-se o A-GPS (*Assisted Global Positioning System*): esta técnica vem sendo cada vez mais utilizada nas redes de telefonia móvel por funcionar adequadamente em ambientes fechados e com muitos obstáculos. O A-GPS utiliza as células da rede móvel para distribuir a informação da localização dos satélites, reduzindo assim o tempo de inicialização e o consumo de energia.

## 2.5 Jogos Móveis

Jogos móveis, como o próprio nome sugere, são jogos voltados para dispositivos móveis, como telefones celulares, *smartphones*, PDAs (*Personal Digital Assistants*). Partindo de aplicações simples, com poucas cores e som bastante limitado, estes jogos evoluíram ao ponto de hoje apresentarem uma grande variedade de recursos, tais como gráficos em 3D, sons polifônicos.

Os primeiros jogos para dispositivos móveis surgiram no início da década de 80, com o lançamento da série de jogos *Games & Watch* pela Nintendo [Nin]. Já no ano de 1997, a Nokia [Nok] desenvolveu o primeiro jogo para telefones celulares, o *Snake*, que embora fosse muito simples, tornou-se bastante popular, incentivando o desenvolvimento de outros jogos pela empresa. [Sem04]

Telefones celulares podem ser encarados como uma ótima plataforma de jogos, pois, apesar de apresentarem algumas restrições, como tela reduzida e baixo poder de processamento, estão sempre disponíveis, podendo ser utilizados em praticamente qualquer local e a qualquer hora.

O mercado de jogos móveis encontra-se em pleno crescimento. Segundo um relatório re-

centemente divulgado pelo Gartner, Inc. [Inca], o mercado de jogos móveis deverá movimentar 4,3 bilhões de dólares este ano, registrando uma expansão de 50% em relação ao ano passado, e deverá crescer para cerca de 9,6 bilhões de dólares no ano de 2011.

## 2.6 Jogos Baseados em Localização

Os jogos baseados em localização tiveram origem a partir da combinação entre os jogos móveis e os serviços baseados em localização. Estes jogos tiram proveito da popularização dos dispositivos móveis e da crescente oferta de tecnologias de localização para introduzir uma nova forma de interação entre seus jogadores, melhorando assim a experiência de jogo.

Jogos baseados em localização utilizam a localização física do jogador como um dos elementos integrantes da dinâmica do jogo, exigindo o seu deslocamento em lugares reais, como parques e ruas, de modo que as suas ações no mundo real - como encontrar alguém, objetos virtuais ou se esconder - venham a alterar o estado do mundo virtual<sup>2</sup>.

Alguns destes jogos ainda permitem a participação de jogadores on-line, que participam do jogo via computadores pessoais, e podem, por exemplo, visualizar o andamento do jogo em um mapa, além de enviar mensagens com instruções para os participantes móveis.

Um dos jogos baseados em localização mais conhecidos é o Pac-Manhattan [Uni], que transpõe o famoso Pac-Man [Nam] para as ruas de Nova York. Utilizando conexões Wi-Fi (*Wireless-Fidelity*) nos telefones celulares, os jogadores atuam como os personagens do jogo, perseguindo o personagem principal nas ruas da cidade, sendo representados na tela dos dispositivos.

O jogo *Alien Revolt* [Coo], lançado pela operadora Oi [Oi] no ano de 2005, foi o primeiro jogo do Brasil a utilizar a localização física dos jogadores. O *Alien Revolt* é um *Massive Multiplayer Role Playing Game* (MMRPG) para dispositivos móveis onde é possível visualizar na tela a posição de cada jogador a uma distância de até três quilômetros.

---

<sup>2</sup>O *estado do mundo virtual* representa o conjunto de informações sobre os personagens de cada jogador e os demais elementos do jogo em um dado instante.

## Trabalhos Relacionados

Este capítulo apresenta algumas soluções propostas no sentido de facilitar o desenvolvimento e execução de aplicações cientes de contexto e serviços baseados em localização.

O primeiro dos trabalhos apresentados neste capítulo, o *middleware* de suporte ao desenvolvimento de jogos PM2G [Tri07], foi utilizado no desenvolvimento do serviço de localização proposto, apresentado no Capítulo 4.

### 3.1 Middleware de Suporte PM2G

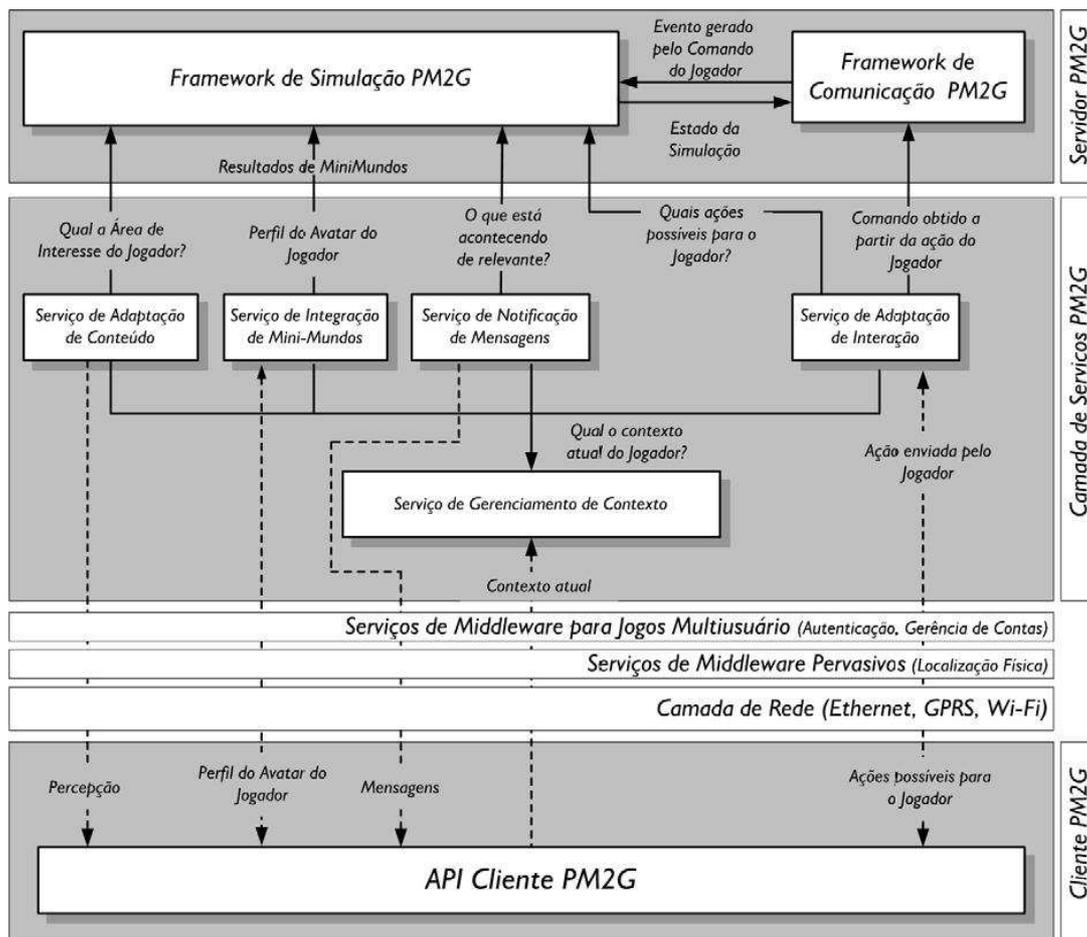
O *middleware* de suporte a jogos pervasivos multiusuário multiplataforma (*Pervasive Multiplayer Multiplatform Game* - PM2G) apresenta uma arquitetura baseada em serviços que ameniza o esforço exigido na sua implementação, reduzindo assim os custos e o tempo de desenvolvimento.

Aplicações PM2G impõem uma série de requisitos funcionais e não funcionais a serem tratados por um modelo de suporte. A proposta desta arquitetura é ampliar os serviços oferecidos pelas plataformas tradicionais de suporte a jogos multiusuário, que atacam serviços comuns - como gerência de contas e autenticação, escalabilidade e tolerância a falhas - agregando serviços de suporte relacionados a jogos PM2G.

A arquitetura PM2G segue o modelo Cliente/Servidor, onde o Servidor executa as simulações do jogo e o conjunto de serviços. Esta abordagem foi adotada principalmente pela necessidade de tratar a escalabilidade e propiciar um melhor controle sobre o gerenciamento do estado do jogo, além de ser muito resistente a jogadores trapaceiros.

A arquitetura é dividida em três blocos principais: o *Servidor*, a *Camada de Serviços* e o *Cliente*, como pode ser visto na Figura 3.1.

O primeiro bloco, o *Servidor*, é composto por dois *frameworks*: o *Framework de Simulação* e o *Framework de Comunicação*. A comunicação entre os jogadores e o mundo virtual simu-



**Figura 3.1** Arquitetura de Serviços PM2G [Tri07]

lado se dá através de eventos, que são recebidos pelo *framework de comunicação* e repassados ao *framework de simulação*. Por sua vez, o *framework de simulação* oferece aos clientes e demais serviços o estado corrente do jogo. Este estado representa o conjunto de informações sobre os personagens de cada jogador e demais elementos do jogo, como NPCs (*Non-player Character*) ou itens.

O segundo bloco, a *Camada de Serviços*, apresenta o conjunto de serviços oferecidos pela arquitetura, brevemente descritos a seguir.

- *Serviço de Gerenciamento de Contexto:* tem por finalidade realizar o monitoramento do contexto de execução dos jogadores. As informações de contexto incluem dados do dispositivo utilizado e as preferências pessoais do jogador. Este serviço tem um papel central em relação aos demais, uma vez que as informações de contexto de cada jogador guiam o funcionamento interno de cada um dos serviços.

- *Serviço de Adaptação de Conteúdo*: é responsável por transformar o estado corrente do jogo e repassá-lo ao jogador de acordo com o seu contexto atual. A partir da *área de interesse*<sup>1</sup> do jogador, o serviço realiza transformações que adequam as informações desta área de interesse de acordo com o contexto do jogador.
- *Serviço de Adaptação de Interação*: este serviço tem por objetivo facilitar a realização de tarefas por parte de jogadores que utilizam diferentes dispositivos, particularmente os móveis, uma vez que estes possuem uma série de limitações. Para alcançar este objetivo, o serviço filtra, a partir do estado da simulação e do contexto atual do jogador, quais as ações possíveis para o usuário em um dado momento do jogo. Em um processo inverso, o serviço transforma a ação escolhida em um ou mais comandos convencionais, que serão repassados ao framework de comunicação, e posteriormente ao framework de simulação para a sua execução no mundo virtual.
- *Serviço de Notificação de Mensagens*: este serviço tem a função de informar aos jogadores sobre acontecimentos de seu interesse, quando tais jogadores estiverem em um contexto móvel. Ele é o principal responsável por criar a idéia do jogador móvel estar sempre conectado ao jogo, sendo notificado através de diferentes meios de comunicação, como mensagens SMS (*Short Message Service*) ou e-mail. Este serviço visa criar uma abstração para que jogadores móveis possam ser alertados sobre diferentes eventos relativos tanto de sua área de interesse do mundo virtual, quanto aqueles do mundo real.
- *Serviço de Integração de Mini-Mundos*: este serviço diz respeito à integração ao jogo dos cenários formados pelo conceito de mini-mundo<sup>2</sup>. Cada partida paralela iniciada precisa repassar ao jogo informações de seus jogadores, localizações e resultados finais destas partidas.

Como mencionado anteriormente, estes serviços foram concebidos de modo a tratar questões específicas do domínio de aplicações PM2G.

E por fim, o terceiro bloco da arquitetura, o *Cliente*, representa a aplicação de acesso do jogador ao jogo. Esta aplicação recebe e envia informações tanto aos serviços quanto ao jogo, por meio de uma API (*Application Programming Interface*).

---

<sup>1</sup>*Área de interesse* é um subconjunto do mundo virtual, abrangendo objetos, jogadores e demais elementos do mundo com os quais o jogador pode interagir em um dado instante.

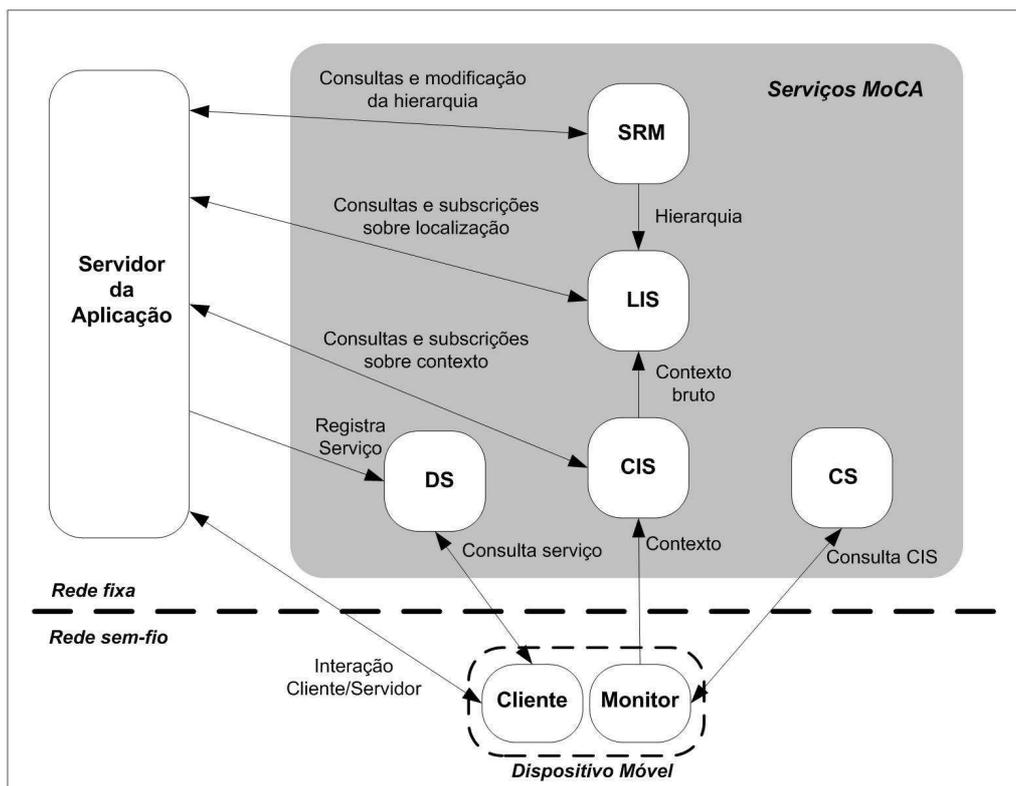
<sup>2</sup>*Mini-mundos* podem ser compreendidos como partidas isoladas entre jogadores móveis, porém com conseqüências no jogo massivo, de forma semelhante às conseqüências das interações entre jogadores dentro do mundo virtual

## 3.2 MoCA

MoCA (*Mobile Collaboration Architecture*) [SER<sup>+</sup>04] é uma arquitetura de *middleware* que oferece suporte ao desenvolvimento e execução de aplicações distribuídas cientes de contexto para redes locais sem fio infra-estruturadas, particularmente IEEE 802.11.

A arquitetura disponibiliza uma série de serviços que provêm meios para coletar e processar informações de contexto de dispositivos móveis, além de um conjunto de APIs de forma a facilitar o desenvolvimento de aplicações que interagem com esses serviços.

Uma típica aplicação desenvolvida utilizando MoCA é composta por três partes: *servidor*, *proxy* e *cliente da aplicação*. As duas primeiras partes são executadas na rede fixa e a última no dispositivo móvel. O *proxy* é responsável por intermediar a comunicação entre o cliente e o servidor, fornecendo funcionalidades como encriptação e compressão dos dados.



**Figura 3.2** MoCA (*Mobile Collaboration Architecture*) [SER<sup>+</sup>04]

Os componentes da MoCA, mostrados na Figura 3.2, são brevemente descritos a seguir, dando uma idéia de como podem ser utilizados no desenvolvimento de aplicações.

- *Context Information Service (CIS)*: recebe e processa as informações de contexto enviadas pelos monitores e eventualmente envia notificações sobre mudanças de contexto aos interessados.
- *Configuration Service (CS)*: armazena e gerencia informações de configuração de cada dispositivo utilizadas pelo CIS.
- *Monitor*: é um *daemon* executado em cada dispositivo móvel que tem a função de coletar informações a respeito do dispositivo e da rede, enviando-os periodicamente ao CIS, presente na rede fixa. Os dados coletados incluem ponto de acesso (*Access Point - AP*) e endereço IP (*Internet Protocol*) corrente, qualidade da conexão do enlace sem fio com todos os APs ao seu alcance, taxa de utilização da CPU, memória e energia disponível.
- *Location Inference Service (LIS)*: infere a localização simbólica aproximada de dispositivos móveis a partir de informações de contexto coletadas pelo CIS. Para obter a localização do dispositivo, o LIS compara o padrão de sinal RF (radiofrequência) corrente do dispositivo (de todos os APs 802.11 em sua vizinhança) com os padrões de sinal previamente coletados em pontos de referência pré-definidos. Portanto, antes de fazer a inferência, a base de dados do LIS deve ser povoada com as amostras de sinais RF em cada ponto de referência.
- *Symbolic Region Manager (SRM)*: disponibiliza uma interface para definir e consultar informações sobre hierarquias de regiões simbólicas<sup>3</sup>.
- *Context Privacy Service (CoPS)*: controla quando, como, e para quem informações de contexto coletadas e processadas pelo CIS e LIS serão reveladas. Este serviço implementa um mecanismo que avalia políticas de privacidade e checa se o acesso à informação de contexto de um usuário será permitido a um outro usuário ou aplicação.
- *Event-based Communication Interface (ECI)*: possibilita a comunicação assíncrona, usando a abordagem *publish/subscribe* [GHJ<sup>+</sup>95] baseada em eventos. Através de sua API, uma aplicação pode registrar seu interesse em eventos específicos, e desta forma, ser notificada da sua ocorrência.
- *Discovery Service (DS)*: armazena informações de aplicações e serviços registrados junto ao *middleware* MoCA, de forma a disponibilizar a sua descoberta por parte de seus clientes.

---

<sup>3</sup>Uma *região simbólica* é uma localização física com uma semântica ou identificação bem definida, como Universidade Federal de Pernambuco ou Centro de Informática.

- *Proxy Framework*: interage com os serviços oferecidos pela arquitetura, ocultando os detalhes desta interação e registrando-se como interessado em notificações sobre informações de contexto de interesse da aplicação. Permite associar ações a serem realizadas na ocorrência de eventos de contexto.
- *Client e Server APIs*: são utilizados para acessar os dados de qualquer dispositivo armazenados pelo CIS, ocultando do desenvolvedor da aplicação, os detalhes sobre o acesso aos serviços oferecidos pela arquitetura.

A arquitetura também provê transparência de mobilidade para as aplicações. Quando um dispositivo móvel migra para uma nova rede, o *proxy* é capaz de realizar tratamento de *handover*<sup>4</sup>, realizando a re-localização do dispositivo e determinando um outro proxy mais apropriado a ser usado pelo cliente, transferindo a sessão corrente para o mesmo.

### 3.3 Nimbus

O *Nimbus* [Rot04] é um *framework* flexível criado com o objetivo de apoiar o desenvolvimento de serviços baseados em localização. O *framework* provê uma interface comum de acesso às informações de localização e abstrai o mecanismo utilizado para a sua obtenção, permitindo também a troca do mecanismo de localização em tempo de execução em *runtime*.

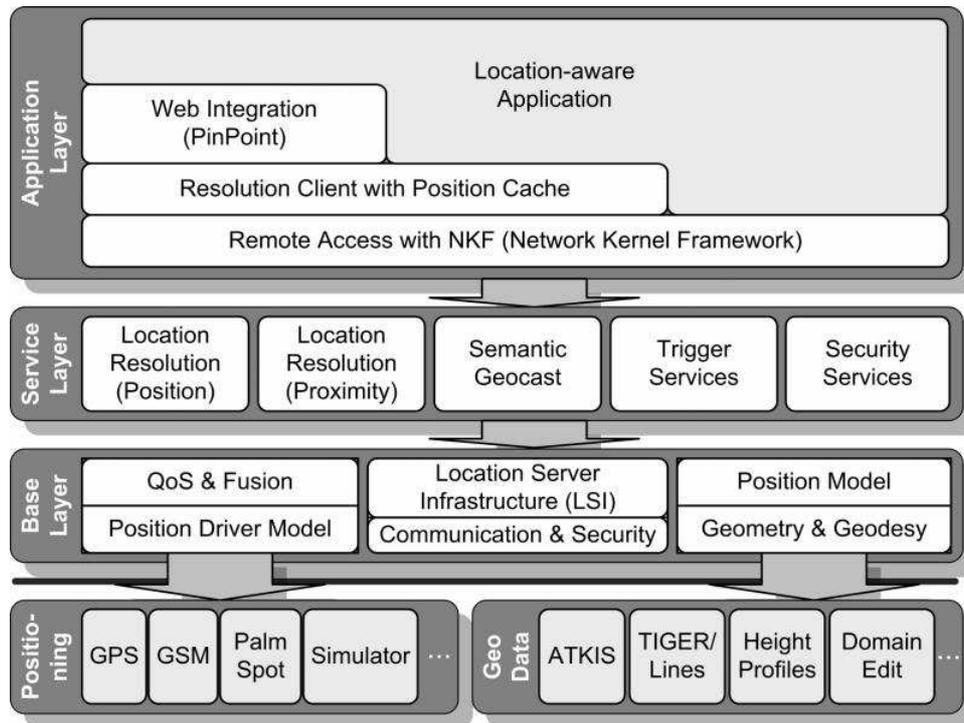
O *framework Nimbus* também faz uso do conceito de região simbólica, trazendo algumas facilidades como a possibilidade de realizar buscas por localidades através do seu nome.

A arquitetura do Nimbus é formado por três camadas: a *camada base (Base Layer)*, a *camada de serviço (Service Layer)* e a *camada de aplicação (Application Layer)*, como mostrado na Figura 3.3.

A *camada base* provê os serviços básicos relacionados com a tecnologia de localização, fornecendo suporte a sistemas de navegação por satélite (como GPS), sistemas baseados na infra-estrutura de rede celular e de sistemas de posicionamento indoor (como infravermelho e rádio). A tecnologia de localização é conectada ao sistema através de uma interface, permitindo assim a sua substituição em tempo de execução. Nessa camada, encontram-se ainda o *location model*, modelo este que oferece o suporte a regiões simbólicas, e a *Location Server Infrastructure*, que armazena os dados de localização e permite a sua recuperação.

---

<sup>4</sup>*Handover* é o procedimento empregado em redes sem fios para tratar a transição de um dispositivo móvel de uma célula para outra de forma transparente ao utilizador.



**Figura 3.3** *Framework Nimbus* [Rot04]

A *camada de serviço* provê serviços de localização de alto nível. Entre eles, o *Location Resolution*, que permite a uma aplicação requisitar a localização corrente de um usuário móvel. Se mais de uma tecnologia de localização estiver acessível, o *framework* seleciona a tecnologia apropriada de acordo com os parâmetros especificados. Outro importante serviço dessa camada é o *Semantic Geocast*, que informa a aplicação quando uma certa localidade for alcançada pelo usuário.

E por fim, a *camada de aplicação* é composta pela própria aplicação e por outros componentes, como o *Network Kernel Framework*, um *middleware* projetado para dispositivos móveis que oferece primitivas de comunicação para acesso aos servidores *Nimbus*, e o *PinPoint*, um componente de alto nível utilizado no desenvolvimento de aplicações baseadas em localização para a web.

## Serviço de Localização

Este capítulo descreve em detalhes o Serviço de Localização proposto neste trabalho, incluindo uma definição de seus componentes e a arquitetura proposta.

Como visto na Seção 2.3, a localização é a informação mais utilizada em aplicações cientes de contexto, sendo, inclusive, destacada por alguns autores [SAW94] [BBC97] [DA99] [Dey01] como parte do grupo de informações contextuais mais importantes, denominado *contexto primário*.

O serviço proposto tem como objetivo utilizar a informação de localização física dos jogadores móveis como um elemento integrante da dinâmica dos jogos, tendo um papel importante na proposta PM2G na medida em que dá suporte à detecção de jogadores que estejam nas proximidades do mundo real, auxiliando assim a formação dos mini-mundos [TFR06].

O serviço foi concebido para ser utilizado em ambientes externos, onde a localização é obtida via GPS. Porém, a princípio, qualquer método de localização que forneça a posição atual do dispositivo (latitude, longitude e altitude) pode ser utilizado, desde que seja mantido um certo grau de precisão.

Entre as inúmeras possibilidades do uso da informação de localização do jogador em um jogo PM2G, o serviço proposto neste trabalho oferece as seguintes funcionalidades:

- *Detecção de jogadores próximos*<sup>1</sup>: o serviço detecta a presença de adversários nas proximidades do jogador e notifica-o através de mensagens SMS (*Short Message Service*), facilitando, deste modo, a formação de mini-mundos, que são caracterizados pela necessária proximidade física dos jogadores.
- *Detecção de áreas mapeadas (AM)*: esta funcionalidade é semelhante a anterior, porém, o seu objetivo é notificar o jogador sobre a proximidade de *áreas mapeadas*, que nada mais são do que regiões onde o jogador pode utilizar a sua movimentação no mundo real

---

<sup>1</sup>Um objeto é considerado próximo quando está a uma distância de no máximo 3 km, seja ele um item virtual ou um jogador adversário.

para se mover no mundo virtual. É importante destacar que o serviço não dá suporte ao funcionamento destas áreas, mas somente à sua detecção.

- *Envio de mensagens para jogadores próximos:* o serviço permite que o jogador envie mensagens aos adversários que se encontram nas proximidades. Através desta funcionalidade, o jogador pode, por exemplo, desafiar estes adversários para a realização de mini-mundos, aumentando ainda mais a interação entre os jogadores.

Para que um jogador seja notificado e detectado pelo serviço, ele deve estar jogando através de um dispositivo móvel e indicar explicitamente nas suas preferências pessoais o desejo de ser localizado.

## 4.1 Modelagem

O Serviço de Localização (SL) atua de forma conjunta com dois serviços já existentes no *middleware*: o *Serviço de Gerenciamento de Contexto (SGC)* e o *Serviço de Notificação de Mensagens (SNM)*. O SL faz uso dos dois serviços consumindo as notificações de mudança de contexto produzidas pelo *SGC* e utilizando o *SNM* para enviar mensagens aos jogadores.

As informações de contexto utilizadas pelo *middleware* PM2G podem ser divididas em três partes: o dispositivo utilizado para acessar o jogo, as preferências pessoais do jogador, e a localização física do jogador, quando este estiver utilizando um dispositivo móvel e desejar ser localizado (Figura 4.1), sendo todas do interesse do Serviço de Localização.



**Figura 4.1** Modelo de Contexto PM2G [Tri07]

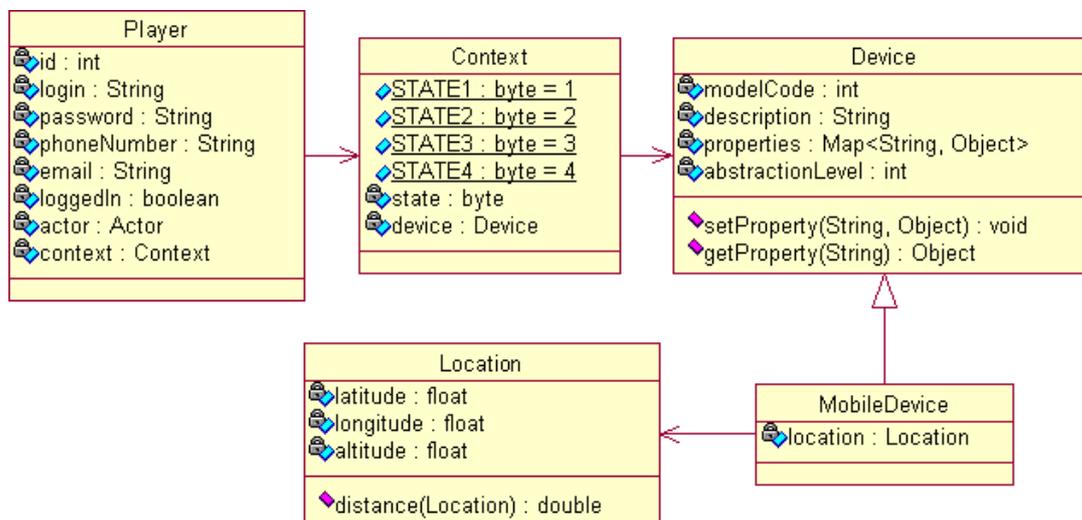
O dispositivo é o principal elemento de contexto utilizado pelo *middleware*, pois indica se o jogador está acessando o jogo através de um dispositivo móvel, dentre outras funcionalidades do *middleware*. Entre as preferências pessoais, está o comportamento associado a cada jogador (Figura 4.2), onde este pode optar por ser notificado sobre eventos do jogo, como da existência de adversários nas proximidades do mundo real. Por fim, a localização física do jogador, de

particular interesse neste trabalho, permite, entre outras possibilidades, que outros jogadores possam ter ciência da proximidade de potenciais adversários para a realização de mini-mundos.

| Estado | Notificações de Eventos | Mini-mundos  |
|--------|-------------------------|--------------|
| 1      | Disponível              | Disponível   |
| 2      | Disponível              | Indisponível |
| 3      | Indisponível            | Disponível   |
| 4      | Indisponível            | Indisponível |

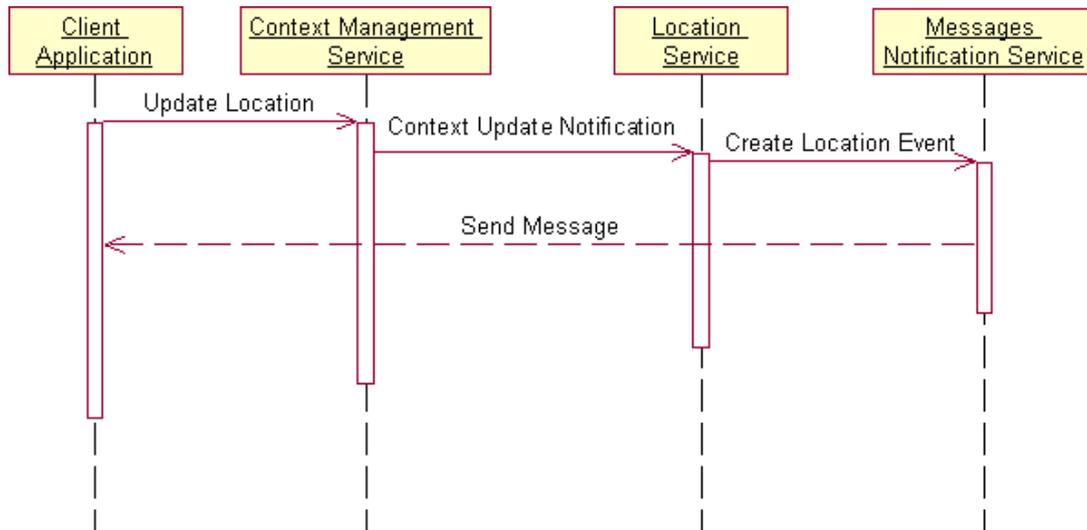
**Figura 4.2** Estados de um jogador [Tri07]

Cada jogador (classe *Player*) possui um contexto (classe *Context*), cujos principais atributos são um estado (atributo *state*) e um dispositivo (classe *Device*). Quando os jogadores estão utilizando dispositivos móveis, é utilizada a classe *MobileDevice*, que além das suas propriedades, possui uma localização (classe *Location*) associada. A classe *Location* é composta pelos atributos *latitude*, *longitude* e *altitude*, além do método *distance*, que realiza o cálculo da distância entre a localização corrente e qualquer outra passada como parâmetro. O relacionamento entre estas classes pode ser visto na Figura 4.3.



**Figura 4.3** Diagrama de Classes - Classes relacionadas ao Contexto

As mudanças de contexto são realizadas implícita ou explicitamente pelo jogador ao alterar as suas preferências pessoais ou se movimentar no mundo real, por exemplo. Estas informações são repassadas pela aplicação ao *S GC*, que notifica os serviços interessados sobre a mudança no contexto do jogador. A Figura 4.4 apresenta uma versão simplificada do diagrama de seqüência que ilustra a mudança de localização física do jogador.



**Figura 4.4** Diagrama de Seqüência - Serviço de Localização

O Serviço de Localização, ao receber uma notificação de mudança de contexto do *SGC*, processa estas informações e, de acordo com o seu resultado, cria um ou mais eventos e os envia para o *SNM*, responsável pela criação e envio das mensagens SMS aos jogadores.

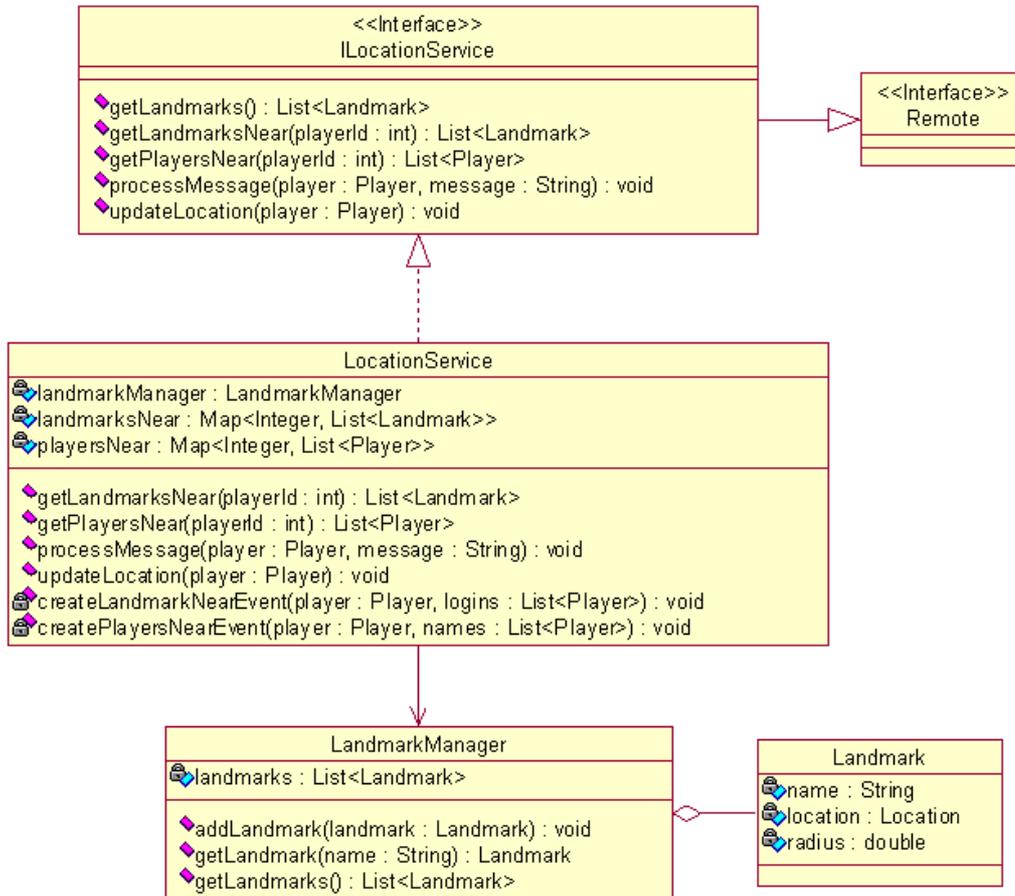
A classe *LocationService* armazena, para cada jogador, uma lista de adversários (atributo *playersNear*) e de áreas mapeadas (atributo *landmarksNear*) próximas, de modo a não ter que recalcular a lista de adversários próximos a cada chamada ao serviço, funcionando de modo semelhante à uma *cache*. Além disso, esta lista permite que o serviço identifique jogadores e AMs que não estavam próximos anteriormente, notificando o jogador sobre a sua proximidade.

Em relação às áreas mapeadas, foi criada a classe *Landmark*, que armazena o seu nome (atributo *name*), sua localização (atributo *location*) e o seu raio de alcance (atributo *radius*). O raio de alcance tem a função de informar ao serviço sobre a que distância estas áreas podem ser consideradas próximas do jogador, já que elas podem apresentar uma grande variação de tamanho.

## 4.2 Arquitetura Proposta

A Figura 4.5 apresenta a arquitetura proposta para o Serviço de Localização.

Como mencionado anteriormente, o *SGC* tem um papel central no funcionamento do *middleware* PM2G, indicando aos demais serviços a ocorrência de mudanças no contexto dos jo-



**Figura 4.5** Arquitetura do Serviço de Localização

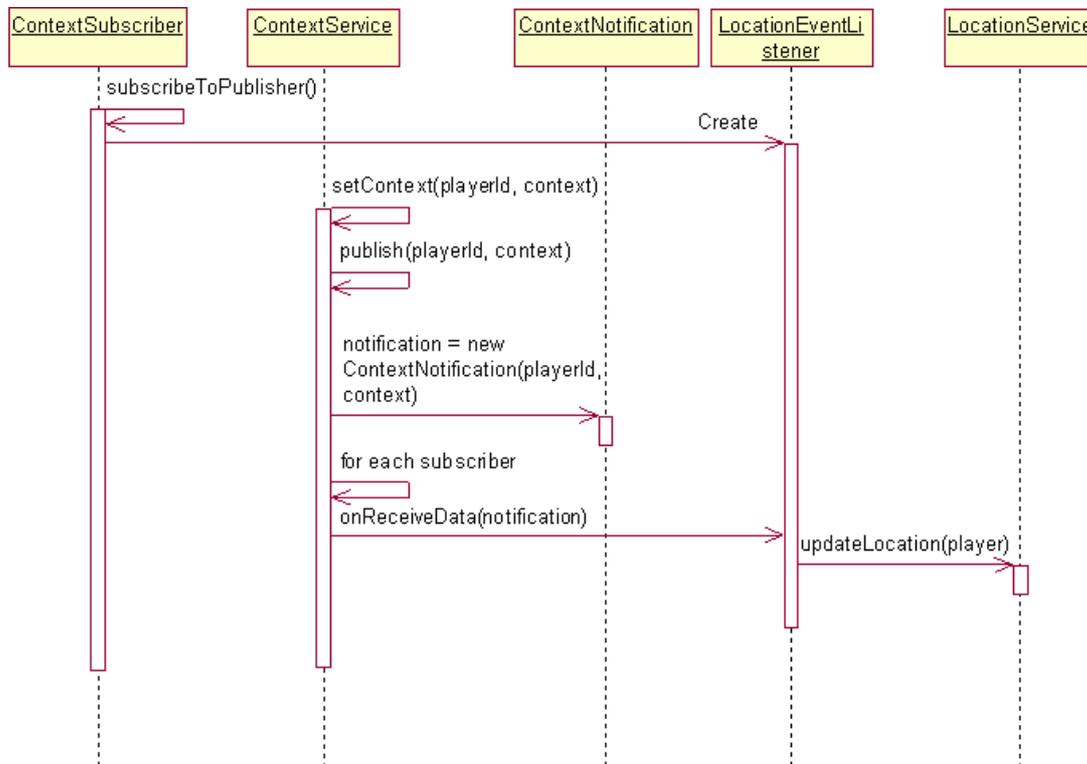
gadores. De modo a prover um canal de comunicação entre o serviço e os interessados nestas mudanças, o *SGC* faz uso do padrão de projeto *Observer* [GHJ<sup>+</sup>95], fornecendo a classe *ContextSubscriber*<sup>2</sup>, onde os interessados se registram para receber as notificações de mudança de contexto.

Seguindo este padrão de projeto, onde cada assinante especifica a ação a ser realizada ao ser notificado, foi criada a classe *LocationEventListener*, que estende a classe abstrata *ContextEventListener*, fornecida pela arquitetura. A classe *LocationEventListener* implementa o método herdado de sua superclasse, o *onReceiveData*, de forma a notificar o Serviço de Localização (classe *LocationService*) a respeito das mudanças de localização dos jogadores.

O *SGC*, serviço responsável pelo monitoramento do contexto, dispara as notificações a respeito destas mudanças, através do seu método *publish*, a todos os seus assinantes, entre eles,

<sup>2</sup>A classe *ContextSubscriber* foi modificada de modo a se adequar ao padrão *Observer*, descrito em [GHJ<sup>+</sup>95]

o Serviço de Localização. A Figura 4.6 ilustra estes passos através de um diagrama de seqüência.



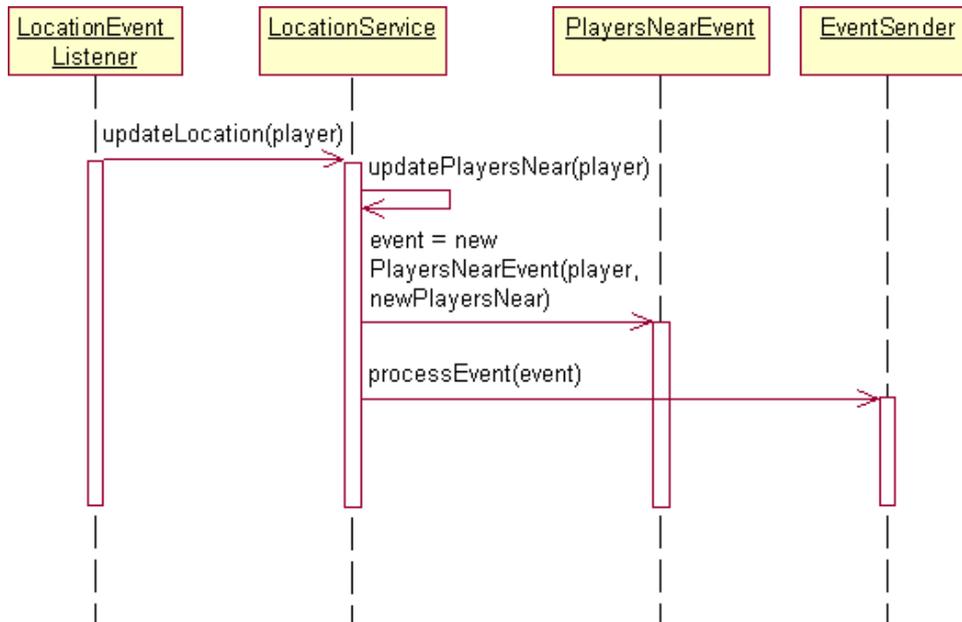
**Figura 4.6** Diagrama de Seqüência - Recebimento de uma notificação de mudança de contexto pelo Serviço de Localização

A partir do recebimento da notificação de mudança de contexto, através do método *updateLocation*, o Serviço de Localização (classe *LocationService*) recupera e processa a informação de localização do jogador, que se dá em duas etapas:

- Detecção de adversários próximos;
- Detecção de áreas mapeadas nas proximidades;

A primeira das etapas engloba (Figura 4.7) os procedimentos relacionados à detecção da proximidade de outros jogadores. Através do método *updatePlayersNear*, invocado a partir de *updateLocation*, o serviço atualiza a lista de jogadores próximos (atributo *playersNear*) e notifica o jogador sobre a presença destes adversários.

Por sua vez, a segunda etapa engloba os procedimentos relacionados à detecção de áreas mapeadas (método *updateLandmarksNear*), que, de modo semelhante à etapa anterior, é invocado a partir do método *updateLocation*. Neste ponto, o serviço atualiza a lista de AMs



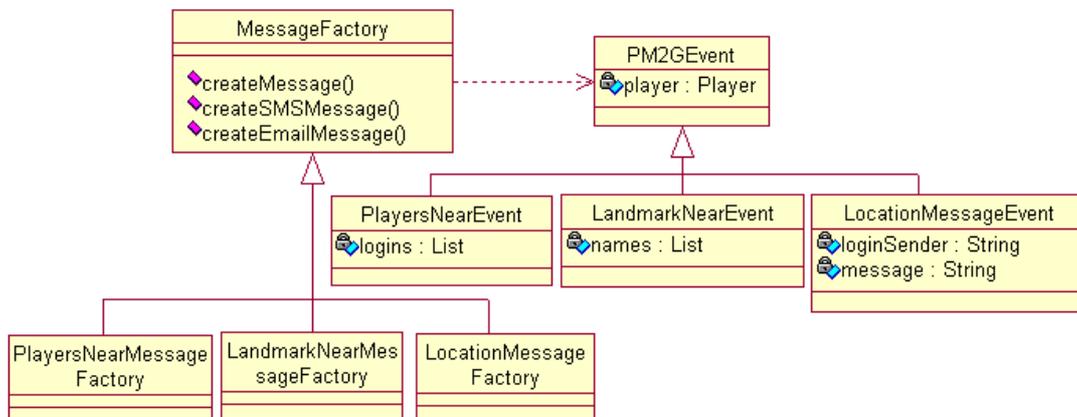
**Figura 4.7** Diagrama de Seqüência - Detecção de jogadores próximos

próximas, armazenadas no atributo *landmarksNear*, e notifica o jogador sobre a proximidade de outras AMs.

De forma a criar um evento específico para cada tipo de notificação, foram criadas as classes *PlayersNearEvent* e *LandmarkNearEvent*, ambas subclasses de *PM2GEvent*. Estes eventos armazenam, respectivamente, a lista de jogadores (atributo *players*) e a lista de AMs (atributo *landmarks*), sobre os quais o jogador deve ser notificado. Estes eventos são criados pelo serviço de localização e repassados ao *SNM*, que utiliza as classes *PlayersNearMessageFactory*, *LandmarkNearMessageFactory*, extensões da classe abstrata *MessageFactory*, para criar mensagens de acordo com as informações específicas de cada evento.

Para repassar os eventos criados ao *SNM*, o Serviço de Localização utiliza uma instância da classe *EventSender*. Esta classe possui o método *processEvent*, que ao receber os eventos, os armazena em um fila para serem tratados pelo serviço. O *SNM* é informado sobre a classe de tratamento associada ao evento através de um arquivo de configuração, cuja estrutura é apresentada na Figura 4.9, onde estão definidos os mapeamentos entre eventos e seus respectivos procedimentos de tratamento.

Por fim, a Figura 4.10 apresenta o diagrama de seqüência da funcionalidade que permite o envio de mensagens para os jogadores próximos.

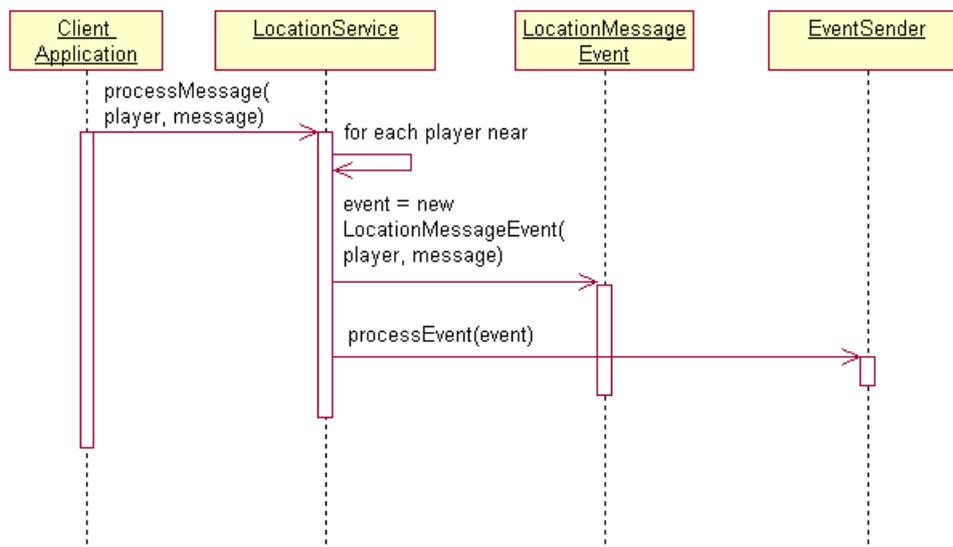


**Figura 4.8** Diagrama de Classe - Eventos, e criadores de mensagens

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<event_manager>
  <event_conf>
    <event_class>
      pm2g.services.location.events.PlayersNearEvent
    </event_class>
    <message_factory_class>
      pm2g.services.location.events.PlayersNearMessageFactory
    </message_factory_class>
  </event_conf>
  <event_conf>
    <event_class>
      pm2g.services.location.events.LandmarkNearEvent
    </event_class>
    <message_factory_class>
      pm2g.services.location.events.LandmarkNearMessageFactory
    </message_factory_class>
  </event_conf>
  <event_conf>
    <event_class>
      pm2g.services.location.events.LocationMessageEvent
    </event_class>
    <message_factory_class>
      pm2g.services.location.events.LocationMessageFactory
    </message_factory_class>
  </event_conf>
</event_manager>
  
```

**Figura 4.9** Configuração dos Eventos e Factories utilizados pelo Serviço de Localização



**Figura 4.10** Diagrama de Sequência - Envio de mensagem para adversários próximos

## Experimentos

A validação do serviço proposto neste trabalho foi realizada através do desenvolvimento de um protótipo. Neste capítulo são apresentados os detalhes da sua implementação e a descrição dos experimentos realizados.

O protótipo desenvolvido foi dividido em três partes:

- *Serviço*: consiste no serviço de localização descrito no Capítulo 4;
- *Cliente*: implementa a aplicação a ser executada no dispositivo móvel do usuário, fazendo uso da informação de localização;
- *Servidor*: componente que permite a comunicação entre o cliente e os serviços do middleware PM2G, entre eles, o de localização.

O servidor desenvolvido durante a prototipação é diferente do servidor do jogo, que é responsável, entre outras coisas, por receber os eventos enviados pelos clientes e demais serviços, repassar estes eventos ao *Framework de Simulação* e enviar o resultado da simulação aos clientes.

A utilização de um servidor para intermediar a comunicação entre os clientes e os serviços foi necessária devido à ausência de um componente da arquitetura PM2G que processe as informações transmitidas pelas aplicações, enviadas através de requisições HTTP (HyperText Transfer Protocol), o único protocolo de rede exigido pelo MIDP 2.0 (Mobile Information Device Profile) [SM06].

### 5.1 Prototipação do Cliente

Para a validação do serviço de localização, foi desenvolvida uma aplicação que procura simular parcialmente o cliente de um jogo PM2G. Esta aplicação é simples, mas suficiente para simular alguns dos cenários imaginados durante o desenvolvimento deste trabalho.

O cliente é responsável por transmitir ao servidor as informações de contexto do usuário, que incluem o dispositivo utilizado, as preferências pessoais e a sua localização física. Para que a aplicação inicie a transmissão destas informações, o usuário deve se autenticar e alterar o seu status para *disponível*.

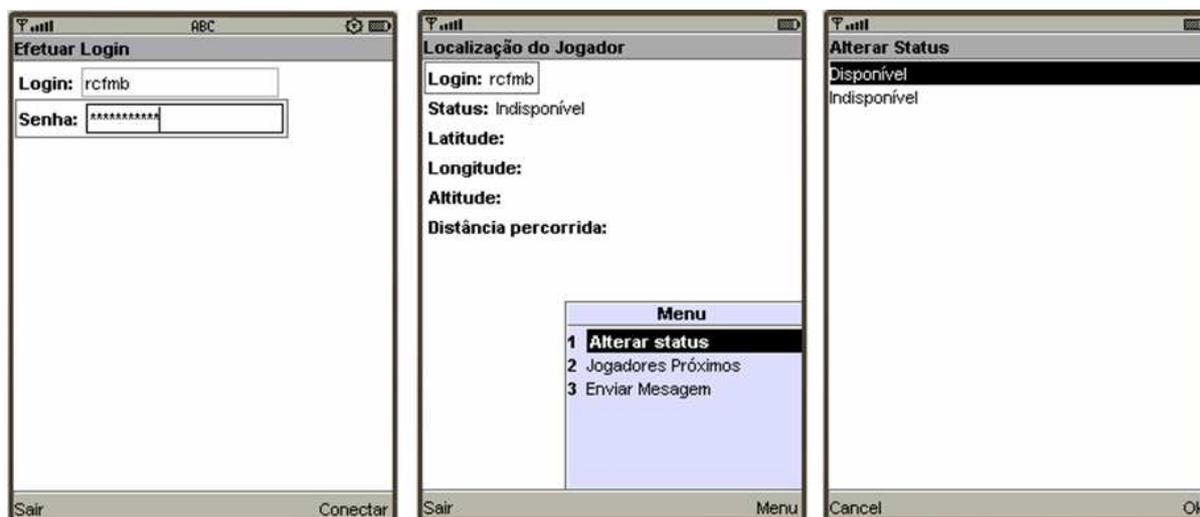


Figura 5.1 Cliente: Telas de logon e alteração de status

A partir deste momento, a aplicação passa a transmitir periodicamente (em intervalos de 20 segundos) as informações de contexto, exibindo na tela as coordenadas do dispositivo (obtidas via GPS) e a distância percorrida até então pelo usuário.

Entre as funcionalidades disponíveis no cliente estão:

- *Visualizar adversários*: permite a visualização do *login* dos usuários que estejam próximos;
- *Enviar mensagem*: possibilita o envio de mensagens aos usuários que estejam nas proximidades;

A aplicação foi totalmente desenvolvida na plataforma Java ME (*Java Platform, Micro Edition*) [SMb], utilizando a *Location API* (JSR 179) [Pro05] para obter a localização do dispositivo.

De forma a facilitar a realização dos experimentos, que exigiriam a utilização de dispositivos com receptor GPS integrado, foram utilizados simuladores de dispositivos móveis. Com o uso destes simuladores, foi possível a execução de múltiplas instâncias do cliente, além da entrada das coordenadas do GPS através de um arquivo XML (*Extensible Markup Language*).



**Figura 5.2** Cliente: Tela principal, jogadores próximos e enviar mensagem

## 5.2 Prototipação do Servidor

O servidor é responsável por receber as informações de contexto enviadas pelos clientes e repassá-las ao *Serviço de Gerenciamento de Contexto (SGC)*, que fica encarregado pela sincronização da ocorrência de mudanças de contexto para os demais serviços do *middleware*, entre eles, o de localização.

Como já mencionado, o servidor se comunica com os clientes através de requisições HTTP enviadas por estes. Já a sua comunicação com o *SGC* é feita através da classe *ContextServiceClient*, provida pelo API do *middleware*. A classe do servidor responsável por receber as requisições do cliente, a *ContextManagementServlet*, mantém no atributo *clients* uma instância da classe *ContextServiceClient* para cada um dos clientes online.

Além de prover a comunicação entre os clientes e os serviços do *middleware*, o servidor também permite que os usuários visualizem através de um navegador web um mapa com a sua localização e a de seus adversários mais próximos.

A maior parte do servidor foi desenvolvido na plataforma *Java Enterprise Edition (J2EE)* [SMA], e para a exibição dos mapas foi utilizada a Google Maps API [Incb], disponível em *Javascript*.



Figura 5.3 Servidor: Visualizar mapa

### 5.3 Descrição dos Experimentos

Nesta seção são apresentados brevemente os resultados da utilização da aplicação desenvolvida. Como mencionado anteriormente, os resultados foram obtidos através da execução do cliente em simuladores para dispositivos móveis, e da execução dos serviços da arquitetura PM2G e do servidor em computadores pessoais.

#### 5.3.1 Detectar Adversários Próximos

Durante a execução deste cenário foram utilizados seis clientes. Para cada um destes usuários, foi inserido, através do simulador, um arquivo XML com coordenadas aleatórias, de forma a fazê-los se encontrarem. O objetivo do experimento foi verificar se o Serviço de Localização cumpria com a sua função, que é de detectar e notificar cada usuário da presença dos outros. Foi verificado que o serviço funcionou adequadamente, enviando uma mensagem SMS para cada um deles.

#### 5.3.2 Detectar Áreas Mapeadas (AMs)

Neste cenário foram utilizados um usuário e quatro áreas mapeadas. O objetivo deste cenário foi verificar se o serviço notificava o jogador da proximidade destas AMs. De forma se-

melhante ao cenário anterior, o usuário se movimentou aleatoriamente de forma a se aproximar destes locais. O serviço se comportou como o esperado, notificando o jogador via mensagens SMS sobre a sua presença das AMs.

### **5.3.3 Enviar Mensagem para Adversários Próximos**

Neste cenário, os usuários utilizaram a aplicação para enviar mensagens de texto para os outros jogadores que se encontravam nas proximidades. Este cenário, de forma semelhante ao cenário Detectar Adversários Próximos, facilita a realização de mini-mundos. Um usuário pode, por exemplo, enviar uma mensagem convocando os jogadores que estejam nas proximidades para participar de uma seção de mini-mundo que está para ser iniciada. O serviço conseguiu entregar todas as mensagens corretamente.

## CAPÍTULO 6

# Conclusões

Neste trabalho foi realizada uma pesquisa sobre jogos baseados em localização, onde foram apresentados os conceitos de ciência de contexto, serviços baseados em localização e jogos móveis. Em seguida, foram analisados alguns trabalhos relacionados no sentido de facilitar o desenvolvimento e execução de aplicações que fazem uso da informação de contexto do usuário, entre elas, a sua localização.

O objetivo principal do trabalho foi o desenvolvimento de um novo serviço para o *middleware* de suporte a jogos PM2G [Tri07], o Serviço de Localização. Este serviço foi proposto com a intenção de permitir a utilização da informação de localização física do jogador na dinâmica de jogos PM2G, tendo um papel importante na proposta PM2G.

Entre as funcionalidades oferecidas pelo serviço estão: a possibilidade de detectar adversários nas proximidades, possibilitando o envio de mensagens para estes jogadores; e a detecção de áreas mapeadas, que são locais onde a movimentação no mundo real é utilizada para mover o avatar no mundo virtual. Vale ressaltar que o mapeamento da movimentação no mundo real para o virtual não foi implementado, e sim o mecanismo de inserção e detecção dessas áreas.

De forma a validar o serviço de localização proposto, foi desenvolvido o protótipo de uma aplicação que utiliza a localização do usuário, procurando, desta forma, simular parcialmente o cliente de um jogo. Esta aplicação é simples, mas foi suficiente para simular alguns dos cenários possíveis em jogos PM2G.

O serviço proposto ataca um subconjunto das possibilidades oferecidas pelo conhecimento da localização dos seus usuários, limitando-se a oferecer funcionalidades que servem apenas de apoio aos outros serviços já presentes na arquitetura PM2G, como a formação de mini-mundos.

Por se tratar de um protótipo, não foram realizados testes com o intuito de medir a sua eficiência ou escalabilidade, não sendo possível garantir a sua adequação e correto funcionamento em um ambiente real.

## 6.1 Trabalhos Futuros

Ao concluir este trabalho, pode-se identificar alguns encaminhamentos futuros no sentido de aperfeiçoar o Serviço de Localização proposto. Algumas sugestões são:

- Inserir a possibilidade de coletar objetos virtuais presentes no mundo real. A idéia é que o jogador seja notificado da presença destes objetos e consiga coletá-los ao se aproximar a uma certa distância. Esta funcionalidade começou a ser desenvolvida, mas não foi possível concluí-la a tempo.
- Utilizar outros métodos de localização de forma a diminuir dependência do GPS como fonte da localização. Uma das possibilidades é utilizar o método de localização via redes Wi-Fi, fornecida pela arquitetura MoCA [SER<sup>+</sup>04];
- Integrar o serviço de localização proposto ao *Location Inference Service (LIS)*, fornecido pela arquitetura MoCA, de modo a utilizar o conceito de região simbólica do LIS para implementar o mapeamento da movimentação do jogador no mundo real para o virtual.

## 6.2 Considerações Finais

Com base no que foi estudado neste trabalho, pode-se concluir que a utilização de informações de contexto, como a localização do usuário, permite a oferta de serviços mais sofisticados. Entre estes serviços, estão os chamados serviços baseados em localização (LBS), já bastante utilizados no exterior e que estão começando a experimentar um crescimento no país. A cada dia, operadoras de telefonia móvel e outras empresas lançam LBS buscando atender às necessidades não só dos clientes corporativos, mas também do público em geral, como o lançamento de jogos baseados em localização. A expectativa é que, a exemplo do que já acontece nos países europeus e asiáticos, em breve estes serviços comecem a fazer parte do dia-a-dia do brasileiro.

## Referências

- [BBC97] P. J. Brown, J. D. Bovey, and X. Chen. *Context-Aware Applications: From the Laboratory to the Marketplace*. IEEE Personal Communications, 1997.
- [Coo] M1nd Cooperation. Alien revolt. Disponível em: <http://www.alienrevolt.com/pt>.
- [Cor04] Adolfo Guilherme Silva Correia. *Aplicações e Serviços Baseados em Localização*. Pontifícia Universidade Católica do Rio de Janeiro, 2004.
- [DA99] A.K. Dey and G.D. Abowd. *Towards a Better Understanding of Context and Context-Awareness*. Georgia Institute of Technology, 1999.
- [DdA04] Luiz Eduardo Figueiredo Domiciano and Luiz Ricardo Rodrigues de Almeida. *Implementação de Serviços Alternativos de Localização Via Celular com Participação do Usuário*. Universidade de Brasília, 2004.
- [Dey01] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 2001.
- [dOJ05] Bruno Gonçalo de Oliveira Júlio. O impacto da localização de utilizadores, na construção de narrativas em serviços móveis. Technical report, Universidade Nova de Lisboa, September 2005.
- [dVP06] Davi de Vasconcelos Pedrosa. *Jogos Multiplataforma Multiusuário*. Universidade Federal de Pernambuco, 2006.
- [Gam] Newt Games. Mogi. Disponível em: <http://www.mogimogi.com>.
- [GHJ<sup>+</sup>95] Erich Gamma, Richard Helm, Ralph Johnson, et al. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., 1995.
- [Gon05] Kleder Miranda Gonçalves. Um framework para comunicação baseada em localização. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, April 2005.

- [Gov04] U.S. Government. *Global Positioning System*. 2004. Disponível em: <http://www.gps.gov>.
- [Inca] Gartner Inc. Disponível em: <http://www.gartner.com>.
- [Incb] Google Inc. *Google Maps API*. Google Inc. Disponível em: <http://www.google.com/apis/maps/>.
- [LNN<sup>+</sup>06] Elias F. Lopes, Jaime G. Nogueira, Thiego G. Nacif, et al. *Sistemas Baseados em Localização em um Mundo Sem-fio com J2ME e API Location*. Universidade da Amazônia, 2006.
- [Nam] Namco. Pac-man. Disponível em: <http://www.namco.co.jp>.
- [Nin] Nintendo. Disponível em: <http://www.nintendo.com>.
- [Nok] Nokia. Disponível em: <http://www.nokia.com>.
- [NPM01] Daniela Nicklas, Christoph Pfisterer, and Bernhard Mitschang. *Towards Location-based Games*. Universität Stuttgart, November 2001.
- [oEE99] Institute of Electrical and Electronics Engineers. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*. IEEE Std., 1999.
- [Oi] Oi. Disponível em: <http://www.oi.com.br>.
- [P<sup>+</sup>03] S. Pokraev et al. *Context-aware services, state-of-the-art*. Telematica Instituut, 2003.
- [Pro05] Java Community Process. *JSR 179: Location API for J2ME*. Java Community Process, 2005. Disponível em: <http://java.sun.com/javame>.
- [Res] Insight Research. Disponível em: <http://www.insight-corp.com>.
- [RES<sup>+</sup>04] Hana L. Rubinsztein, Markus Endler, Vagner Sacramento, et al. Support for context-aware collaboration. *First International Workshop on Mobility Aware Technologies and Applications*, 2004.
- [Rot04] J. Roth. *Novel Architectures for Location-based Services*. Annual Meeting for Information Technology and Computer Science, 2004.

- [RPM98] N. Ryan, J. Pascoe, and D. Morse. *Enhanced reality field work: the context-aware archaeological assistant*. In *Computer applications and quantitative methods in archeology*, Oxford press., 1998.
- [SAW94] B. Schilit, N. Adams, and R. Want. *Context-Aware Computing Applications*. 1st International Workshop on Mobile Computing Systems and Applications, 1994.
- [Sem04] Alexei Semenov. *Mobile Games*. Helsinki University of Technology, 2004.
- [SER<sup>+</sup>04] Vagner Sacramento, Markus Endler, Hana K. Rubinsztein, et al. *Moca: A middleware for developing collaborative applications for mobile users*. IEEE Distributed Systems Online, 2004.
- [SMa] Inc. Sun Microsystems. *Java 2 Platform, Enterprise Edition (J2EE)*. Sun Microsystems, Inc. Disponível em: <http://java.sun.com/javaee>.
- [SMb] Inc. Sun Microsystems. *Java ME Platform*. Sun Microsystems, Inc. Disponível em: <http://java.sun.com/javame>.
- [SM06] Inc. Sun Microsystems. *Mobile Information Device Profile (MIDP)*. Sun Microsystems, Inc., 2006. Disponível em: <http://java.sun.com/products/midp>.
- [ST94] B. N. Schilit and M. Theimer. *Disseminating Active Map Information to Mobile Hosts*. IEEE Network, 1994.
- [TFR06] Fernando Trinta, Carlos Ferraz, and Geber Ramalho. Uma proposta de cenários e serviços de suporte para jogos multiusuário multiplataforma pervasivos. *Simpósio Brasileiro de Sistemas Multimídia e Web: XII Simpósio Brasileiro de Sistemas Multimídia e Web*, 2006.
- [The] Blast Theory. Uncle roy all around you. Disponível em: <http://www.uncleroyallaroundyou.co.uk/>.
- [Tri07] Fernando Antonio Mota Trinta. *Definindo e Provendo Serviços de Suporte à Jogos Multiusuário e Multiplataforma: Rumo à Pervasividade*. PhD thesis, Universidade Federal de Pernambuco, 2007.
- [Uni] New York University. Pac-manhattan. Disponível em: <http://www.pacmanhattan.com/>.

- [Wei91] Mark Weiser. *The Computer for the Twenty-First Century*. Scientific American, 1991.
- [WGB99] M. Weiser, R. Gold, and J. S. Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM Syst. J*, 1999.
- [Wie05] Wouter Wiegmans. Location-based gaming. Technical report, University of Twente, January 2005.

