

**Implementação de Serviços Baseados em
Localização Utilizando Arquiteturas e Padrões
Abertos**

Grace Kelly de Castro Silva

Dissertação de Mestrado Profissional

Implementação de Serviços Baseados em Localização Utilizando Arquiteturas e Padrões Abertos

Grace Kelly de Castro Silva¹

11 de fevereiro de 2005 ✓

Banca Examinadora:

- Prof. Dr. Geovane Cayres Magalhães
Instituto de Computação, UNICAMP (Orientador)
- Prof. Dr. Célio Cardoso Guimarães
Instituto de Computação, UNICAMP
- Prof. Dr. Maurício Ferreira Magalhães
Faculdade de Engenharia Elétrica e Computação, UNICAMP
- Prof. Dr. Edmundo Roberto Mauro Madeira (Suplente)
Instituto de Computação, UNICAMP

¹Suporte financeiro do Fundo para o Desenvolvimento Tecnológico das Telecomunicações (FUNTEL).

UNIDADE	BC
Nº CHAMADA	Si38i
V	EX
TOMBO BCI	64893
PROC.	16-P-00086/05
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	21-05-07
Nº CPD	

Bib. id. 359246

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP

Si38i Silva, Grace Kelly de Castro
Implementação de serviços baseados em localização utilizando arquiteturas e padrões abertos / Grace Kelly de Castro Silva - Campinas, [S.P. :s.n.], 2005.

Orientador: Geovane Cayres Magalhães
Dissertação (Mestrado Profissional) - Universidade Estadual de Campinas, Instituto de Computação.

1. Computação móvel. 2. Sistema de informação geográfica. 3. ~~Web services.~~ I. Magalhães, Geovane Cayres. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

* Serviços na Web.

Implementação de Serviços Baseados em Localização Utilizando Arquiteturas e Padrões Abertos

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Grace Kelly de Castro Silva e aprovada pela Banca Examinadora.

v Campinas, 11 de fevereiro de 2005.



Prof. Dr. Geovane Cayres Magalhães
Instituto de Computação, UNICAMP
(Orientador)

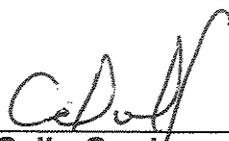
Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Trabalho Final Escrito defendido e aprovado em 11 de fevereiro de 2005, pela Banca Examinadora composta pelos Professores Doutores:



Prof. Dr. Mauricio Ferreira Magalhães
FEEC – UNICAMP



Prof. Dr. Celio Cardoso Guimarães
IC - UNICAMP



Prof. Dr. Geovane Cayres Magalhães
IC - UNICAMP

© Grace Kelly de Castro Silva, 2005.
Todos os direitos reservados.

*Dedico este trabalho aos meus pais, à minha
querida avó, à minha irmã e ao João Fabio,
com carinho.*

Agradecimentos

A Deus pela vida, saúde e trabalho, pela minha família maravilhosa e pelas boas oportunidades que sempre tive na vida.

Aos meus pais, que nunca mediram esforços para me oferecer sempre o melhor, sempre me dando oportunidades para que eu pudesse alcançar os meus objetivos, como a realização deste trabalho.

À minha irmã Alessandra, sempre tão compreensiva e carinhosa.

Ao João Fabio, pelo companheirismo, incentivo e paciência.

A toda a minha família, em especial à Beth, Rômulo e Rominho, que souberam compreender a minha ausência nas férias, feriados e almoços de domingo.

Ao meu orientador, Prof. Dr. Geovane Cayres Magalhães, pelo incentivo e por acreditar no meu trabalho, mesmo sabendo das dificuldades em se conciliar profissão e estudo.

Aos amigos da Fundação CPqD, pelo apoio e incentivo. Em especial a Eliana e Eliane pela oportunidade, ao Mário Harada pelas importantes dicas e a Pati, minha colega de projeto e “irmã gêmea”, cujo apoio foi essencial para a conclusão deste trabalho.

A todos os funcionários do Instituto de Computação da UNICAMP.

Resumo

Serviços Baseados em Localização (*Location-Based Service* (LBS)) são serviços que usam informações geográficas para obter e gerar informações úteis para os usuários, como por exemplo, a localização do posto de gasolina mais próximo ou a melhor rota da posição corrente do usuário até o hospital mais próximo.

Existem várias iniciativas na definição de padrões com o objetivo de aumentar a interoperabilidade entre aplicações baseadas em serviços de localização. Dentre as principais iniciativas podemos citar as especificações *Open Location Services* (OpenLS) do *Open Geospatial Consortium* (OGC) e *Mobile Location Protocol* (MLP) do *Open Mobile Alliance* (OMA).

O uso da tecnologia *Web Services* também é uma forma de garantir a interoperabilidade entre as aplicações de LBS, uma vez que ela permite que aplicações em diferentes plataformas e linguagens de programação se comuniquem via XML ou outros padrões Web.

Esta dissertação tem como objetivo apresentar um estudo sobre as tecnologias envolvidas no desenvolvimento de Serviços Baseados em Localização e propor uma arquitetura baseada em padrões abertos para oferecimento destes serviços. Esta arquitetura é validada através do desenvolvimento de um protótipo do Serviço de Apresentação de Mapas definido na especificação OpenLS. Este trabalho também contempla uma análise de alguns produtos disponíveis no mercado que oferecem aplicações LBS que utilizam padrões abertos.

Abstract

Location-Based Services (LBS) are services which use geographic information to obtain and generate useful information for users such as the location of the nearest gas station, for instance, or the best route for the nearest hospital by taking the user's current position as a starting point.

There are several initiatives for the definition of standards aiming at increasing the interoperability of applications, which are based upon location services. Among the main initiatives, the Open Location Services (OpenLS) of the Open Geospatial Consortium (OGC), and Mobile Location Protocol (MLP) of the Open Mobile Alliance (OMA) should be mentioned.

The use of the Web Services technology is also a way to assure the interoperability involving LBS applications, once it allows applications in different platforms and programming languages to communicate via XML or other Web standards.

This dissertation is aimed at presenting a study regarding the technologies involved in the development of Location-Based Services and proposing an architecture which is based on open standards to make these services available. This architecture is validated through the development of a prototype of the Presentation Service defined in the OpenLS specification. This study also comprises an analysis of some market-available products that offer LBS applications that use open standards.

Conteúdo

Agradecimentos	ix
Resumo	xi
Abstract	xiii
1 Introdução	1
1.1 Motivação	2
1.2 Objetivo	2
1.3 Organização do Trabalho	3
2 Serviços Baseados em Localização	5
2.1 Definição	5
2.2 Classificação	6
2.3 <i>Indoor</i> Location-Based Services	7
2.4 Tecnologias Envolvidas	8
2.4.1 Tecnologias de Posicionamento	10
2.4.2 Bases de Dados	16
2.4.3 <i>Web Services</i>	19
2.5 Considerações Finais	24
3 A Especificação <i>Open Location Services</i>	25
3.1 Arquitetura	25
3.2 Serviços OpenLS	27
3.2.1 Serviço de Diretório	28
3.2.2 Serviço de <i>Gateway</i>	31
3.2.3 Serviço de Geocodificação e Geocodificação Reversa	33
3.2.4 Serviço de Apresentação de Mapas	39
3.2.5 Serviço de Rotas	45
3.3 Outros Padrões Envolvidos	48

B Exemplos de Requisição e Resposta dos Serviços OpenLS	95
B.1 Serviço de Diretório	95
B.2 Serviço de <i>Gateway</i>	96
B.3 Geocodificação/Geocodificação Reversa	98
B.3.1 Serviço de Geocodificação	98
B.3.2 Serviço de Geocodificação Reversa	99
B.4 Serviço de Apresentação de Mapas	101
B.5 Serviço de Rotas	103
C Arquivo <i>deploy.wsdd</i> do Serviço de Apresentação de Mapas	107
D Mensagem SOAP enviada para o Serviço de Apresentação de Mapas	109

Lista de Tabelas

2.1	Precisão de Posicionamento Baseado em Cell-ID.	12
5.1	Tabela de Comparação dos Produtos.	83

Lista de Figuras

2.1	Organização de Células Urbanas.	12
2.2	Localização Através de <i>Enhanced Observed Time Difference</i> (E-OTD). . .	13
2.3	Localização Através de <i>Uplink Time Difference of Arrival</i> (U-TDOA). . . .	14
2.4	Localização Através de <i>Assisted-Global Positioning System</i> (A-GPS). . . .	15
2.5	Comunicação via <i>Web Services</i>	20
3.1	Padrão de Requisição/Resposta de Serviços OpenLS.	26
3.2	Requisição do Serviço de Diretório.	29
3.3	Resposta do Serviço de Diretório.	31
3.4	Requisição do Serviço de Gateway.	32
3.5	Resposta do Serviço de Gateway.	34
3.6	Requisição do Serviço de Geocodificação.	35
3.7	Elemento <code>StreetAddress</code> do Serviço de Geocodificação.	36
3.8	Elemento <code>StreetIntersection</code> do Serviço de Geocodificação.	36
3.9	Resposta do Serviço de Geocodificação.	37
3.10	Elemento <code>GeocodedAddress</code> do Serviço de Geocodificação.	37
3.11	Requisição do Serviço de Geocodificação Reversa.	39
3.12	Resposta do Serviço de Geocodificação Reversa.	40
3.13	Requisição do Serviço de Apresentação de Mapas.	41
3.14	Elemento <code>OutputType</code> do Serviço de Apresentação de Mapas.	42
3.15	Elemento <code>LayerType</code> do Serviço de Apresentação de Mapas.	43
3.16	Elemento <code>Overlay</code> do Serviço de Apresentação de Mapas.	43
3.17	Resposta do Serviço de Apresentação de Mapas.	44
3.18	Requisição do Serviço de Rotas.	46
3.19	Resposta do Serviço de Rotas.	48
4.1	Arquitetura Proposta.	55
4.2	Diagrama de Casos de Uso.	58
4.3	Organização dos Pacotes.	59
4.4	Classe <code>PresentationServiceLocator</code>	59

4.5	Classes Stub, Skeleton e Template de Implementação.	60
4.6	Classes do Pacote Presentation.	60
4.7	Classes do Pacote OpenGIS.	61
4.8	Diagrama de Sequência.	62
4.9	Página Inicial do Projeto Serviços e Aplicações Móveis (SAM).	68
4.10	Escolha do Agente para Visualização.	69
4.11	Mapa com a Localização do Agente.	70

Capítulo 1

Introdução

À medida que os preços de telefones celulares, *Personal Digital Assistants* (PDAs), computadores de mão, aparelhos de *Global Positioning System* (GPS) e acesso aos serviços oferecidos pelas operadoras de rede celular caem rapidamente, o número de usuários de aparelhos móveis aumenta de forma significativa em todo o mundo. Nos últimos anos, o aumento do número de usuários de telefones móveis excedeu o crescimento do número de linhas fixas, expandindo de 50 milhões para quase um bilhão em 2002. Mais de 90% dos países possuem rede sem fio e aproximadamente uma em cada seis pessoas possui um telefone celular [MDB03].

A redução do tamanho dos dispositivos móveis e melhorias na comunicação sem fio também são fatores que promovem a propagação dos serviços de computação em aparelhos móveis, antes disponíveis apenas em computadores *desktop*. Alguns tipos importantes de aplicações que surgem com esta mudança de comportamento incluem serviços de localização de dispositivos móveis, serviços de turismo, comércio eletrônico móvel, serviços de emergência, além de uma série de serviços na área de saúde e segurança públicas. Estas aplicações são coletivamente chamadas de Serviços Baseados em Localização (*Location-Based Service* (LBS)).

Serviços Baseados em Localização promovem uma mudança fundamental no uso da computação móvel, uma vez que adicionam a esta tecnologia a possibilidade de utilização de recursos de Sistemas de Informação Geográfica (*Geographic Information System* (GIS)¹) em campo, permitindo a interação direta do usuário de dispositivo móvel com o mundo ao seu redor.

Tradicionalmente, o processo de coleta e edição de dados em campo consome muito tempo e gera muitos erros. A informação geográfica muitas vezes é levada a campo na forma de mapas em papel, nos quais são feitas anotações e rascunhos. De volta aos

¹Apesar de existir o termo Sistema de Informação Geográfica (SIG) em português, nesta dissertação é utilizada a sigla em inglês GIS

escritórios, estes dados são inseridos manualmente na base de dados geográfica, gerando um resultado não tão atualizado e preciso como desejado. Conseqüentemente, análises das informações geográficas e tomadas de decisão são prejudicadas.

Algumas das principais atividades realizadas em campo que se beneficiam do uso da informação geográfica e de tecnologias de posicionamento são:

- Levantamento de patrimônio, que requer coleta de dados em campo.
- Manutenção de patrimônio, que requer atualização de atributos ou informações geográficas dos objetos.
- Inspeções, verificação das informações em campo para fins de auditoria e cumprimento de normas legais.
- Relato de incidentes e desastres, permitindo a visualização espacial de incidentes e desastres para análise e tomada de decisão.
- Atendimento de emergências, permitindo o atendimento de forma mais rápida e eficiente uma vez que a localização precisa do incidente pode ser identificada.

1.1 Motivação

Tendo como base estas várias atividades, muitas aplicações práticas podem se beneficiar do uso de Serviços Baseados em Localização tanto na indústria, como no mercado consumidor e governos. Diante deste contexto, vê-se a necessidade de estudo das tecnologia envolvidas no desenvolvimento destes serviços, analisando-se as diferentes possibilidades de implementação e procurando sempre garantir a ubiqüidade e interoperabilidade das aplicações LBS.

1.2 Objetivo

O objetivo desta dissertação é apresentar um estudo sobre as tecnologias envolvidas na implementação de Serviços Baseados em Localização para dispositivos móveis. Também é apresentada uma arquitetura baseada em padrões abertos para desenvolvimento de Serviços Baseados em Localização utilizando *Web Services*. Esta arquitetura é validada através do desenvolvimento de um protótipo do Serviço de Apresentação de Mapas que integra o projeto Serviços e Aplicações Móveis (SAM), em desenvolvimento na Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD). Com base na arquitetura proposta são analisados alguns produtos de mercado que oferecem soluções nesta área, visando destacar a utilização de padrões abertos.

1.3 Organização do Trabalho

O Capítulo 2 desta dissertação apresenta os principais conceitos de Serviços Baseados em Localização, identificando os diversos tipos de serviços existentes e descrevendo as suas características mais importantes. Em seguida são apresentadas as tecnologias envolvidas no desenvolvimento de aplicações de LBS, incluindo tecnologias de posicionamento, servidores de bancos de dados espaciais e *Web Services*.

O Capítulo 3 contempla um estudo da especificação OpenLS do OGC, apresentando os seus principais objetivos e uma visão geral da arquitetura para acesso aos serviços OpenLS. Este capítulo traz ainda uma descrição de cada um dos serviços tratados na especificação, incluindo um detalhamento das suas interfaces de requisição e resposta.

No Capítulo 4 é apresentada uma arquitetura baseada em padrões abertos para o desenvolvimento de Serviços Baseados em Localização utilizando *Web Services*. Para validação da arquitetura proposta são apresentados os resultados do desenvolvimento de um protótipo do projeto Serviços e Aplicações Móveis em andamento na Fundação CPqD, com foco no Serviço de Apresentação de Mapas descrito no Capítulo 3.

No Capítulo 5 são analisados alguns produtos de mercado que oferecem soluções na área de Serviços Baseados em Localização. São descritas as principais funcionalidades de cada solução, procurando destacar a utilização de padrões abertos.

Finalmente, no Capítulo 6 são apresentadas as conclusões finais deste trabalho e propostas de trabalhos futuros.

As siglas utilizadas nesta dissertação estão descritas no apêndice A.

Capítulo 2

Serviços Baseados em Localização

As primeiras idéias sobre comunicação através de dispositivos móveis surgiram no Bell Laboratories no início dos anos 70. Entretanto, os sistemas móveis celulares só foram utilizados para fins comerciais nos anos 80. Atualmente eles representam uma das áreas de maior crescimento no setor de telecomunicações [Sem97].

A evolução da tecnologia das redes de comunicação de dados sem fio e a redução do tamanho e dos preços dos dispositivos móveis, favoreceram o desenvolvimento de um novo tipo de aplicação que utiliza a informação geográfica para prover informações úteis e precisas para os usuários de dispositivos móveis. Tomando como base esta tendência observada, este capítulo apresenta os conceitos e tecnologias envolvidas no desenvolvimento de Serviços Baseados em Localização.

2.1 Definição

Serviços Baseados em Localização são serviços que usam informações geográficas, combinadas ou não com a posição do terminal móvel, para obter e gerar informações úteis para os usuários.

Em 1996, com a regulamentação do mandato E-911 [FCC04] pelo *Federal Communications Commission* (FCC), o qual determinava que as operadoras de rede sem fio dos Estados Unidos deveriam possuir mecanismos para localizar geograficamente as chamadas de emergência, a indústria de equipamentos para localização de dispositivos teve um grande crescimento.

Por volta de 1997 surgiram as primeiras aplicações de LBS. Nesta primeira fase (1997-2001), as aplicações tinham como principal foco extrair da rede celular a localização de dispositivos móveis [Spi03]. Durante este período não havia padronização das interfaces de acesso aos elementos da rede responsáveis por prover a localização de dispositivos, como os *Mobile Positioning Centers* (MPCs) nas redes *Code Division Multiple Access* (CDMA)

e *Gateway Mobile Location Centers* (GMLCs) nas redes *Global System for Mobile* (GSM). Tampouco havia uma interface padrão para acesso aos serviços de GIS. Cada produto de GIS tinha seu padrão XML aberto, mas utilizavam esquemas proprietários. Integração e interoperabilidade entre os sistemas existentes eram os maiores problemas enfrentados.

No final de 2000 e início de 2001, surge uma nova fase no desenvolvimento de aplicações de LBS, na qual o uso de padrões abertos é fator essencial. Dois padrões propostos visam resolver os problemas de interoperabilidade enfrentados na primeira fase. Estes padrões são o *Mobile Location Protocol* (MLP) [OMA02] do *Open Mobile Alliance* (OMA), para localização de dispositivos e *Open Location Services* (OpenLS) [OGC04a] do *Open Geospatial Consortium* (OGC) para processamento espacial.

No Brasil, a disputa por uma fatia cada vez mais lucrativa do mercado de telefonia móvel tem levado as operadoras a uma busca intensa por alternativas rentáveis de serviços destinados a usuários domésticos e corporativos.

As empresas brasileiras têm consciência do potencial dos Serviços Baseados em Localização para geração de novas receitas e conquista de mercado. De olho nesse potencial, várias operadoras já anunciaram iniciativas, buscando parcerias e desenvolvendo soluções que envolvem as tecnologias de localização. Em outubro de 2004, a VIVO[®] lançou a solução VIVO Encontra, que oferece no celular desde a busca de informações relacionadas a uma região até serviços que permitem a localização de outros aparelhos celulares VIVO[®]. As demais empresas brasileiras, no entanto, estão mais cautelosas e só devem investir em soluções de LBS a partir de 2005 [Pai04].

2.2 Classificação

Os Serviços Baseados em Localização podem ser classificados de acordo com o seu uso em diversas categorias [FT04]:

- **Serviços de informação:** Dedicados ao mercado consumidor, estes serviços incluem notícias, previsão do tempo, navegação e rotas, informações turísticas, propagandas baseadas em localização, páginas amarelas, informação de tráfego em tempo real, reservas de táxis e serviços para se localizar um amigo ou um ponto de interesse (*Point of Interest* (POI)) mais próximo. Todos estes serviços levam em consideração a localização corrente do dispositivo móvel, a fim de prover informações precisas e relevantes ao usuário. Estes tipos de serviços podem ser oferecidos tanto como *Short Message Service* (SMS), *Wireless Application Protocol* (WAP) ou voz.
- **Serviços de gerenciamento e rastreamento de frotas:** No mercado consumidor, estes serviços podem ser utilizados por pais para rastrear a localização dos filhos. De forma similar, podem ser usados para rastrear um parente mais velho que tenha

mal de *Alzheimer*, por exemplo. No caso do mercado de negócios, o gerenciamento de frotas pode ser utilizado para rastreamento de caminhões, táxis, etc, tornando possível a otimização das rotas e, assim, diminuir os tempos dos percursos e aumentar a segurança dos motoristas. Estes serviços também podem ser aplicados no gerenciamento de força de trabalho, para otimizar o envio de pessoal técnico, como eletricitistas, bombeiros e médicos, enviando a pessoa mais próxima ao local da solicitação.

- **Emergências:** O interesse pelas aplicações de LBS na área de serviços públicos de emergência surgiu porque as ambulâncias e corpos de bombeiros gastavam muito tempo para chegar ao local do socorro devido à imprecisão das informações sobre a localização da vítima. Além dos serviços públicos, também existem serviços de assistência privada que incluem serviços de monitoramento médico e assistência veicular.
- **Diversão:** Nesta categoria estão os jogos baseados em localização, tais como “atirar” em pessoas que estejam na mesma área, envio de vídeos ou fotos para atividades de lazer referentes ao local onde o usuário se encontra e brincadeiras de caça ao tesouro, no qual o usuário recebe perguntas ou dicas de acordo com a sua localização.

Uma outra classe de serviços são os LBSs baseados em alertas, também chamados *spatial-trigger services*, tais como serviços de localização de amigos e de produtos. Eles notificam usuários sobre eventos importantes que ocorrem nas suas proximidades, como quando um amigo entra em um determinado perímetro ou quando um produto está em promoção em uma loja pela qual o usuário está passando.

Além destas categorias, Serviços Baseados em Localização podem ser classificados em serviços do tipo *push* ou *pull*. Serviços *push* provêm informações baseadas na posição do usuário ou proximidade de algum lugar, produto ou serviço, enquanto serviços *pull* requerem que o usuário requisite a informação explicitamente, utilizando ou não a informação de sua localização como filtro.

2.3 Indoor Location-Based Services

Tecnologias de posicionamento *indoor* relacionam pessoas, objetos e eventos no espaço [Kol04]. A funcionalidade básica de tais aplicações é similar a todas as aplicações de LBS: encontrar um objeto, pessoa, lugar ou evento, porém dentro de um ambiente fechado.

Há aplicações de LBS que auxiliam pessoas cegas a caminhar até um determinado destino em um lugar desconhecido. Por exemplo, a aplicação orienta uma pessoa dentro de um edifício a caminhar do saguão até um auditório.

Tecnologias de posicionamento mais precisas podem orientar a localização de produtos dentro de uma loja. Usuários podem identificar produtos ao seu redor apontando para eles, neste caso, sensores que determinam direção devem ser integrados aos dispositivos móveis.

Conforme já observado por Kolodziej [Kol04], o desenvolvimento de sistemas de posicionamento que funcionam bem *indoor* é um desafio, porque os sinais refletidos nas paredes, pisos e tetos tendem a confundir os sensores e, normalmente, há obstáculos entre os sensores e os objetos sendo monitorados. O GPS e o posicionamento baseado na rede celular não são apropriados nestes casos devido ao bloqueio de sinal e existência de sombras. A tecnologia *Wireless Fidelity* (Wi-Fi) está surgindo como uma alternativa para localização de dispositivos quando o uso de GPS ou outras tecnologias *outdoor* não é adequado.

2.4 Tecnologias Envolvidas

O mercado de serviços de localização demanda tecnologias que têm como princípio a simplicidade, dado que estes serviços são principalmente utilizados por dispositivos móveis. O objetivo é fazer com que a complexidade das aplicações de GIS seja transparente para os provedores de aplicações de LBS. Além disso, devido a suas limitações de memória, capacidade de processamento e largura de banda, dispositivos móveis requerem soluções leves.

A tecnologia usada no desenvolvimento de Serviços Baseados em Localização deve prover mecanismos que suportem desenvolvimento rápido de novos serviços, facilidade de implantação e deve oferecer um sistema altamente escalável, confiável e com alta disponibilidade. Além disso, os serviços de localização de dispositivos devem considerar a existência de diferentes tipos de redes e variações de relevo. Estes e outros aspectos que devem ser considerados no desenvolvimento de Serviços Baseados em Localização estão descritos a seguir.

Escalabilidade e Desempenho Aplicações de LBS devem comportar grande volume de acessos a seus serviços e devem ter um desempenho adequado por se tratar de aplicações de tempo real, utilizando para isso recursos como balanceamento de carga. Serviços críticos como localização e atendimento de emergências possuem restrições em relação ao tempo de resposta.

Além do tempo de resposta, uma das medidas mais comuns de desempenho é a precisão da localização, já que a precisão é fácil de se medir e tradicionalmente é considerada um indicativo de qualidade da solução. Alguns exemplos reais de

requisitos de desempenho do serviço de emergência americano E-911 [FCC04] estão descritos a seguir:

Para tecnologias de localização baseadas nos dispositivos móveis:

- Precisão de até 50 m para 67% das chamadas
- Precisão de até 150 m para 95% das chamadas
- O tempo de resposta de localização do dispositivo deve ser de até 30 segundos

Para tecnologias de localização baseadas na rede celular:

- Precisão de até 100 m para 67% das chamadas
- Precisão de até 300 m para 95% das chamadas
- O tempo de resposta de localização do dispositivo deve ser menor que 30 segundos

Alta Disponibilidade e Confiabilidade A confiabilidade dos serviços baseados em localização é essencial para o sucesso deste tipo de aplicação. Não é desejável que o sistema indique para o usuário um endereço errado que possa levá-lo a um local perigoso ou uma rota que indique como caminho uma rua sem saída. O sistema também deve ser robusto o suficiente para estar sempre disponível e evitar quedas indesejáveis.

Segurança e Privacidade Privacidade e segurança são os pontos mais polêmicos das aplicações de LBS. Além de questões envolvendo autenticação e autorização dos usuários na utilização dos serviços, devem ser consideradas questões de privacidade na localização de dispositivos móveis e no envio de mensagens indesejadas para os terminais, como o envio de propagandas para dispositivos nas proximidades de *shoppings* ou lojas.

Interoperabilidade Devido a sua característica ubíqua, aplicações de LBS devem estar disponíveis em vários tipos de dispositivos, ter interface com sistemas e bancos de dados legados, além de contemplar uma variedade de tecnologias de infra-estrutura de rede. O uso de padrões abertos na definição das interfaces é uma forma de garantir a interoperabilidade entre os sistemas. A adoção da tecnologia *Web Services* também é uma forma de garantir a interoperabilidade, uma vez que ela tem como objetivo facilitar a comunicação entre as aplicações que residem em diferentes plataformas e utilizam diferentes linguagens de programação.

Desenvolvimento Simplificado O desenvolvimento de aplicações de LBS deve ser simples o suficiente para permitir a liberação de novos serviços de forma rápida a fim de atender as demandas do mercado.

A implementação de uma solução de LBS efetiva requer que operadoras de serviços de telecomunicações, provedores de infra-estrutura, integradores de sistemas e fabricantes de dispositivos móveis trabalhem juntos a fim de atingir os requisitos de implementação, desempenho e custo destas soluções em diferentes tipos de redes.

Tecnologias de diferentes áreas estão envolvidas na implementação de aplicações de LBS, desde equipamentos para localização dos dispositivos, bases de dados de localização e conteúdo, até arquiteturas para desenvolvimento dos serviços. Algumas destas tecnologias estão descritas a seguir.

2.4.1 Tecnologias de Posicionamento

Quando surgiram os primeiros sistemas celulares na década de 80, cada país desenvolvia o seu próprio sistema, o que causava uma série de inconvenientes já que os equipamentos só podiam operar dentro dos limites do país e o mercado para equipamentos móveis era limitado.

A fim de resolver estes problemas, foram definidos alguns padrões para os sistemas celulares, como o *Time Division Multiple Access* (TDMA) e o *Code Division Multiple Access* (CDMA). A tecnologia TDMA divide a frequência de rádio em faixas de tempo e aloca estas faixas para múltiplas chamadas. Uma única frequência comporta múltiplos canais de dados simultâneos. Ao contrário da tecnologia TDMA, CDMA não atribui uma faixa de frequência específica para cada usuário, cada canal usa todo o *spectrum* disponível. Dados individuais são codificados com uma seqüência digital randômica e, desta forma, podem ser diferenciados dos demais.

A tecnologia *Global System for Mobile* (GSM) [ETS96] foi construída com base no padrão TDMA. É considerada a tecnologia celular digital mais avançada e é extremamente popular na Europa e Ásia. O GSM tem como características a segurança e autenticação, além de dar ao usuário a possibilidade de trocar de telefone sem a necessidade de reconfigurar o novo aparelho, devido à existência do cartão *Subscriber Identity Module* (SIM).

O cartão SIM provê uma maior mobilidade, uma vez que o usuário tem acesso aos serviços independentemente do terminal que está utilizando. Ao inserir o cartão SIM em outro terminal GSM, o usuário pode receber e fazer chamadas neste terminal, além de acessar outros serviços a que tem direito [Sco97].

Outra tecnologia digital baseada no padrão TDMA é o *General Packet Radio Service* (GPRS). Ela é considerada uma tecnologia 2.5G, entre a segunda e terceira gerações [Wik04a]. Esta nova tecnologia possibilita que usuários façam ligações telefônicas e enviem

dados ao mesmo tempo. O principal benefício da tecnologia GPRS é que ela reserva recursos da rede apenas quando há dados para transferir.

O GPRS contempla uma grande variedade de larguras de banda, se mostrando eficiente quando a largura de banda é limitada. É utilizada tanto para envio e recebimento de pequenos volumes de dados, como e-mail ou navegação na Web, assim como grandes volumes.

Como as aplicações de LBS se baseiam na informação de localização do dispositivo móvel para prover os serviços para os usuários, são necessários mecanismos que forneçam o posicionamento do terminal com maior ou menor precisão de acordo com a necessidade. Um típico serviço que requer baixa precisão é a localização do restaurante mais próximo à posição corrente do usuário. Outros serviços como definição de rotas, gerenciamento de frotas e serviços de emergência, requerem uma precisão maior.

Além da precisão, o tempo de resposta e a confiabilidade da informação são fatores importantes para garantir o sucesso de uma solução de LBS.

Nem todas as tecnologias de posicionamento existentes funcionam em todos os tipos de rede celular descritos anteriormente. A seguir são apresentadas algumas das tecnologias de posicionamento mais utilizadas.

Cell-ID

Cell-ID provê a coordenada de localização do dispositivo baseado na célula na qual o usuário está localizado. A infra-estrutura de localização contém as coordenadas de localização do centróide de cada célula. Estas coordenadas é que são retornadas. Cell-ID opera em todas as redes e é o sistema de posicionamento mais utilizado [Ope04].

Como o terminal pode estar em qualquer lugar dentro da célula ou setor, a precisão deste método depende do tamanho da célula. O posicionamento geralmente é mais preciso em áreas urbanas com uma densa rede de células pequenas. Áreas rurais têm uma densidade menor de estações base, conseqüentemente, uma precisão menor.

A precisão de Cell-ID pode ser melhorada através do cálculo do tempo de transmissão do sinal do dispositivo para a estação base. Estas medidas podem ser utilizadas para se determinar a distância do dispositivo à estação, diminuindo assim o erro de posicionamento. Mesmo com estas melhorias, esta tecnologia é uma das mais inconsistentes e que apresenta menor precisão.

Enhanced Observed Time Difference (E-OTD)

E-OTD opera apenas em redes GSM. O dispositivo móvel mede as diferenças de tempo de transmissão de sinal para um conjunto de estações base na sua vizinhança. Esta informação é transmitida para um *Serving Mobile Location Center* (SMLC) que, através



Figura 2.1: Organização de Células Urbanas.

Área	Precisão
Urbana	100 a 400 metros
Subúrbio	400 a 2000 metros
Rural	1000 a 20000 metros

Tabela 2.1: Precisão de Posicionamento Baseado em Cell-ID.

de um esquema de triangulação e conhecendo as localizações das estações base, determina a área aproximada na qual o dispositivo está localizado [Sna03]. A Figura 2.2 ilustra o mecanismo de localização utilizado pelo E-OTD.

O dispositivo móvel deve medir as diferenças de tempo de transmissão de sinal para pelo menos três estações a fim de determinar a sua posição em duas dimensões (não são oferecidas medidas de altitude). A precisão do método depende da resolução das medidas dos tempos e da geometria das estações bases vizinhas.

Como a medida de tempo é crítica para o cálculo da localização do dispositivo, são necessários equipamentos *Location Measurement Units* (LMU) que forneçam informação precisa de tempo. A necessidade destes equipamentos exige mudanças na infra-estrutura da rede. Além disso, o dispositivo deve possuir software específico para prover E-OTD.

Uplink Time Difference of Arrival (U-TDOA)

O método U-TDOA opera apenas em redes *Universal Mobile Telecommunications Systems* (UMTS), baseadas na tecnologia CDMA. Este método calcula a localização de um dispositivo móvel medindo a diferença de tempo de chegada dos sinais em diferentes receptores LMU. Quando o dispositivo móvel está transmitindo, antenas em diferentes

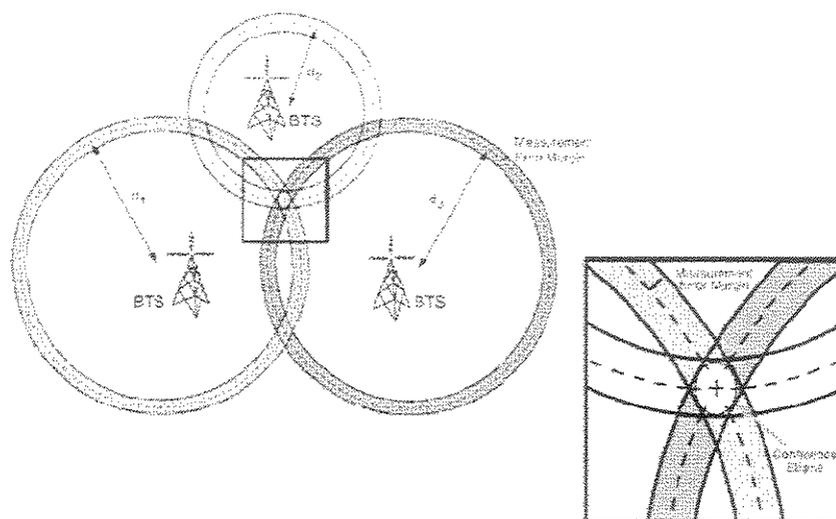


Figura 2.2: Localização Através de E-OTD.

estações base recebem o sinal. O tempo de detecção do sinal por um LMU particular é uma função da distância entre o terminal e cada antena. O método U-TDOA utiliza processamento de sinal digital para comparar os sinais em receptores LMU vizinhos e determinar a diferença do tempo de recepção. O equipamento U-TDOA calcula a latitude e longitude do dispositivo baseado nestas diferenças [Tru04].

Este método de localização produz resultados muito precisos, tipicamente na faixa de 50 metros. A tecnologia U-TDOA pode atingir estes resultados em situações na qual o GPS (ou mesmo A-GPS) simplesmente não funciona, particularmente em lugares fechados ou em ambientes densamente urbanizados. Além disso, com U-TDOA, a precisão pode ser melhorada ajustando-se o número de LMUs na rede.

Global Positioning System (GPS)

O GPS utiliza uma constelação de satélites controlada pelo Departamento de Defesa dos Estados Unidos para determinar a localização de qualquer coisa ou pessoa e prover uma referência de horário altamente precisa para quase todos os lugares da Terra [Wik04b].

O sistema GPS é dividido em três segmentos: espacial, controle e usuário. O segmento espacial compreende a constelação de satélites GPS. O segmento de controle compreende estações base espalhadas pelo mundo responsáveis por monitorar as rotas dos satélites, sincronizar seus horários e carregar dados anuais de calendários, ciclos lunares e informações estatísticas para transmissão pelos satélites. O segmento de usuário compreende os aparelhos GPS usados tanto militarmente como por aplicações civis. Um receptor

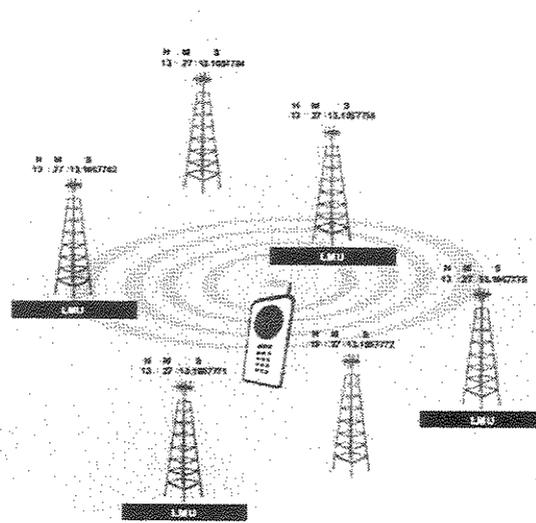


Figura 2.3: Localização Através de U-TDOA.

GPS decodifica tempos de transmissão de sinal a partir de vários satélites e calcula sua posição utilizando mecanismo de triangulação.

Assisted-Global Positioning System (A-GPS)

A-GPS também utiliza os satélites no espaço como pontos de referência para determinar a localização de dispositivos móveis. Além dos satélites, A-GPS utiliza informações de torres celulares da vizinhança, através de um servidor de localização [Sna03].

O servidor de localização informa ao dispositivo quais satélites deve utilizar para triangulação e também executa cálculos complexos para determinar a posição do dispositivo.

Esta tecnologia geralmente provê melhor precisão que GPS uma vez que combina informações da rede celular à localização por satélites.

Soluções Híbridas

A implementação mais direta de uma tecnologia híbrida é a combinação de A-GPS com Cell-ID [Sna03]. Esta solução melhora a cobertura em áreas onde A-GPS não funciona e oferece a precisão da tecnologia A-GPS nos outros casos. A cobertura e precisão de A-GPS é excelente na maioria dos locais, degradando apenas dentro de edifícios ou em áreas urbanas densas onde Cell-ID normalmente funciona melhor, já que estas áreas possuem alta densidade de células.

Uma outra alternativa é a combinação de A-GPS com E-OTD ou U-TDOA. Esta

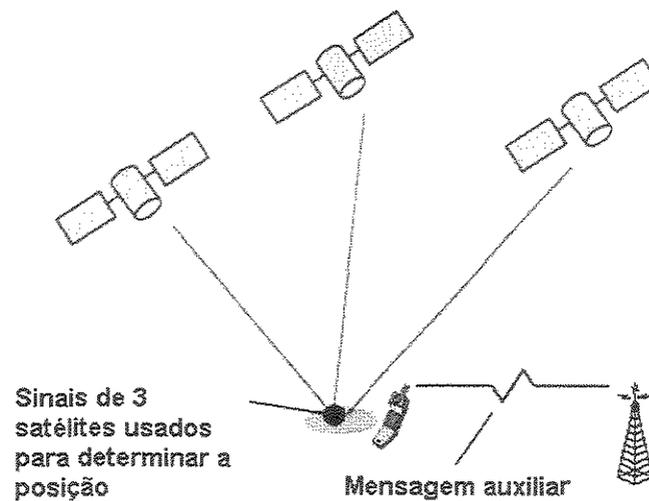


Figura 2.4: Localização Através de A-GPS.

alternativa requer apenas a instalação de E-OTD ou U-TDOA na rede, possibilitando a utilização de A-GPS na maior parte das áreas.

O uso de soluções híbridas melhora o oferecimento dos serviços à medida que uma tecnologia pode suprir a deficiência da outra e não prejudica o desempenho de forma significativa [Sna03].

Wi-Fi

O número cada vez maior de edifícios existentes nas grandes cidades é uma barreira para o uso de sistemas de posicionamento baseados em GPS, uma vez que os grandes blocos de concreto afetam os sinais nos quais a tecnologia GPS se baseia.

O sistema de posicionamento baseado em Wi-Fi, desenvolvido nos Estados Unidos e Inglaterra, preenche uma lacuna deixada por sistemas que utilizam GPS e tecnologias baseadas na rede celular: localização de dispositivos em grandes cidades e dentro de complexos fechados como *shopping centers* [Bie04]. Além do fator tecnológico, o baixo custo da tecnologia Wi-Fi em relação aos receptores GPS, leva a crer que Wi-Fi pode vir a ser a tecnologia central para as novas aplicações baseadas em serviços de localização.

Serviços de emergência têm um interesse especial nesta tecnologia e vêem o uso de Wi-Fi como uma alternativa quando a localização por GPS não é possível. De acordo com a *National Emergency Number Association* (NENA), mais de 30% das chamadas aos serviços 911 nos Estados Unidos são provenientes de celulares, podendo logo ultrapassar o número de chamadas utilizando telefones fixos [NEN].

Cada estação Wi-Fi tem um alcance de aproximadamente 100 metros. O sinal incorpora um código de endereço de rede único que identifica a estação base. Através da intensidade do sinal, a localização do usuário pode ser identificada através de triangulação de pelo menos três estações.

Os sistemas hoje existentes não são tão precisos como GPS, apresentando uma precisão de 20 a 30 metros, enquanto na média o GPS tem uma precisão de 8 a 10 metros. Porém, a utilização de algoritmos que levam em consideração a altura da estação base em relação ao solo ou os materiais das construções na vizinhança, pode levar a tecnologia Wi-Fi a ser compatível com GPS em áreas com boa cobertura.

Além da localização de dispositivos, Wi-Fi também tem como finalidade a conexão de dispositivos sem fio à Internet. É crescente o número de estações base instaladas em cafés, bibliotecas, universidades, aeroportos, cabines telefônicas e outros lugares públicos.

2.4.2 Bases de Dados

A informação associada à localização é essencial para um Serviço Baseado em Localização. Uma posição geográfica (Longitude e Latitude) somente se torna útil para o usuário quando associada a informações mais amigáveis, como endereços ou pontos de referência.

Além de dados de localização, aplicações de LBS utilizam dados dinâmicos para prover informações precisas para o usuário, tais como informações de tráfego, previsão do tempo, trechos em obras, etc. Todo este conteúdo deve ser armazenado em bases de dados e como muitos destes dados possuem informações geográficas associadas, as bases de dados devem ter suporte a dados espaciais.

A seguir são apresentados alguns Sistemas de Gerenciamento de Banco de Dados (SGBDs) que possuem suporte a dados espaciais.

Oracle Spatial

O SGBD Oracle possui duas opções para gerenciamento de dados espaciais: Oracle Locator e Oracle Spatial [Ora03]. O Oracle Locator faz parte da instalação padrão do banco de dados Oracle, tanto na versão *Standard* como *Enterprise*, oferecendo um conjunto de funcionalidades nativas. O Oracle Spatial é uma extensão do Locator e possui funcionalidades mais avançadas para desenvolvimento de um sistema GIS complexo, sendo disponível apenas para a versão *Enterprise*.

Como características do Oracle Locator e Oracle Spatial podem ser citadas a escalabilidade, segurança dos dados e capacidade de replicação e particionamento, possibilitando o gerenciamento de carga do sistema. O banco de dados Oracle também oferece suporte para transações curtas ou longas.

O Oracle Locator e Oracle Spatial possuem três formas geométricas básicas: pontos, linhas e polígonos. Estes tipos primitivos podem ser usados para representar objetos do mundo real como estradas, limites administrativos, facilidades das redes de telecomunicação ou de energia, prédios, dentre outros. Também possui ferramentas para gerenciamento de sistemas de coordenadas e projeções cartográficas, permitindo a conversão de dados entre diferentes sistemas.

Para determinar a interação entre os diferentes objetos da base de dados, existem alguns operadores espaciais como *contains*, *covers* e *anyinteract* (qualquer interação), além das consultas espaciais “*within distance*”, “*nearest neighbor*” e “*relate*”.

O Oracle possui um modelo de dados que armazena e mantém a conectividade de redes do tipo arco-nó, provendo funcionalidades como caminho mais curto e análise de conectividade da rede. Através deste modelo, o Oracle oferece funcionalidade para definição de rotas, determinando distâncias, tempos e percursos entre dois endereços ou pontos quaisquer. Esta funcionalidade inclui definição de preferências para rota mais rápida ou mais curta, resumo do roteiro ou informações detalhadas do percurso.

Na opção Oracle Spatial, está disponível um geocodificador, o qual determina a posição geográfica (longitude e latitude) de um dado endereço. O geocodificador é implementado em Java e armazenado dentro da base de dados como uma *Stored Procedure*. É fornecida uma *Application Program Interface* (API) PL/SQL para acesso à funcionalidade de geocodificação.

O Oracle Spatial é compatível com as recomendações da especificação *Simple Features* [OGC99] do OGC. Também possui suporte para a linguagem GML [OGC03a], além de ser compatível com as especificações OpenLS [OGC04a] e WMS [OGC04b].

PostgreSQL/PostGIS

O PostgreSQL [Pos04] é um SGBD Objeto-Relacional baseado no projeto POSTGRES, o qual foi iniciado em 1986 e desenvolvido no departamento de Ciência da Computação da Universidade da Califórnia, Berkeley. A primeira versão operacional foi liberada em 1987, e em 1988 o POSTGRES foi apresentado na conferência ACM-SIGMOD.

De modo a unificar os crescentes esforços necessários para evoluir o projeto POSTGRES, o projeto de Berkeley foi oficialmente encerrado e o produto passou a chamar-se Postgres95, sendo publicado na Internet.

Em 1996, com a rápida evolução e adoção, o Postgres95 foi renomeado para PostgreSQL, versão 6.0. A partir de então o PostgreSQL tem aumentado o número de usuários e colaboradores das comunidades de software livre, sendo atualmente o SGBD de Código Livre mais robusto e com o maior número de funcionalidades disponíveis.

O PostgreSQL implementa os padrões SQL:92 [ANS92] e SQL:99 [ANS99], bem como funcionalidades importantes: chaves estrangeiras, gatilhos, visões, integridade tran-

sacional, controle de concorrência com multi-versionamento, dentre outras. É possível também criar novos tipos de dados, funções, operadores, funções de agregação, índices e ainda linguagens procedimentais.

Existem também tipos geométricos, como *point*, *box*, *lseg*, *line*, *path*, *polygon*, e *circle*, além de inúmeras funções geométricas, operadores para relacionar estes tipos e funções para conversão de tipos.

O PostgreSQL implementa operadores que possibilitam recuperar a distância entre dois objetos, o ponto mais próximo de um objeto dentro de uma região, o centróide de um objeto, verificar se um objeto sobrepõe outro, se são paralelos, está contido, entre outros. Possui também funções que recuperam a área, diâmetro, número de pontos e raio de uma circunferência, além de oferecer funções de conversão de tipos manipulando polígonos, circunferências, pontos e segmentos de retas.

Há ainda o módulo adicional chamado PostGIS [Ref04], que adiciona uma série de tipos para manipulação de dados geográficos, permitindo que objetos GIS sejam armazenados na base de dados.

Os objetos de GIS implementados pelo PostGIS são os objetos especificados pelo OGC. O PostgreSQL/PostGIS, apesar de aceitar tipos de dados espaciais e um conjunto grande de operadores e funções espaciais, ainda não foi certificado como *OpenGIS compliant*.

MySQL

O SGBD MySQL [AB95b] surgiu a partir de um grupo de desenvolvedores que havia implementado rotinas de baixo nível (ISAM) e usavam o mSQL [Hug01] para conectar as tabelas. Depois de alguns testes foi verificado que o desempenho e a flexibilidade do mSQL não eram suficientes para as necessidades do grupo. Assim, algum tempo depois foi desenvolvida uma nova interface SQL para o banco de dados, mas mantendo praticamente intacta a interface do mSQL, de modo a facilitar a portabilidade de códigos de terceiros que haviam sido escritos para o mSQL. Com isso surgiu o MySQL, já existente há mais de dez anos.

Atualmente o MySQL é mantido pela empresa MySQL AB, na Suécia, e tem se firmado como o SGBD de código livre mais usado no mundo, especialmente para aplicações Web devido ao seu desempenho neste contexto.

O código do MySQL foi escrito em C e C++, testado em vários compiladores e funciona em diversas plataformas, como Linux, AIX, FreeBSD, HP-UX, SunOS, Windows, entre outras.

O MySQL não possui funcionalidades básicas que auxiliam o gerenciamento e administração de uma Base de Dados, como visões, gatilhos, código armazenado, integridade referencial, controle transacional, *full outer join* e cursores.

Esta limitação do MySQL pode ser resolvida com o produto MaxDB [AB95a], o qual adiciona ao MySQL as funcionalidades de um SGBD Relacional.

O MySQL possui suporte nativo a tratamento de dados espaciais a partir de versão 4.1, sem a necessidade de nenhuma instalação de módulos adicionais. Contudo, nas versões atuais (5.0 e 4.1) não é possível criar tabelas geo-referenciadas que suportam transação.

O MySQL implementa o suporte a dados espaciais seguindo a especificação do OGC. O OGC publicou uma especificação com vários conceitos para estender o suporte a dados espaciais no SQL, e o MySQL implementa um subconjunto dessas especificações (tipos geométricos, funções para recuperação de informações, etc).

Uma limitação ainda existente na versão atual (versão 5.0 de desenvolvimento) é que tabelas com dados espaciais não possuem controle de transação. Este fato não impede criação de GIS usando MySQL, porém restringe as possibilidades de sistemas que podem ser construídos usando este SGBD, já que não é possível criar um sistema multi-usuário com concorrência entre os processos.

2.4.3 *Web Services*

Web Services é uma tecnologia que provê interoperabilidade entre sistemas permitindo que aplicações em diferentes plataformas e linguagens de programação se comuniquem via *Extensible Markup Language* (XML) ou outros padrões *web* [MATH03].

Na tecnologia *Web Services*, a publicação e acesso aos serviços envolvem três elementos: consumidores de serviços, provedores de serviços e serviços de diretório.

A Figura 2.5 ilustra a comunicação entre provedores e consumidores de serviços. Os passos envolvidos em prover e consumir serviços são:

1. Um provedor de serviços descreve seu serviço utilizando a linguagem *Web Services Description Language* (WSDL). Esta definição é publicada em um serviço de diretório.
2. Um consumidor de serviços envia uma ou mais consultas ao diretório para localizar um serviço e determinar a forma de comunicação.
3. Parte do WSDL provido pelo provedor de serviço é passada para o consumidor. Isto informa ao consumidor como devem ser as requisições e respostas para o provedor de serviço.
4. O consumidor de serviço usa o XML baseado em WSDL para enviar uma requisição para o provedor.
5. O provedor fornece a resposta esperada para o consumidor.

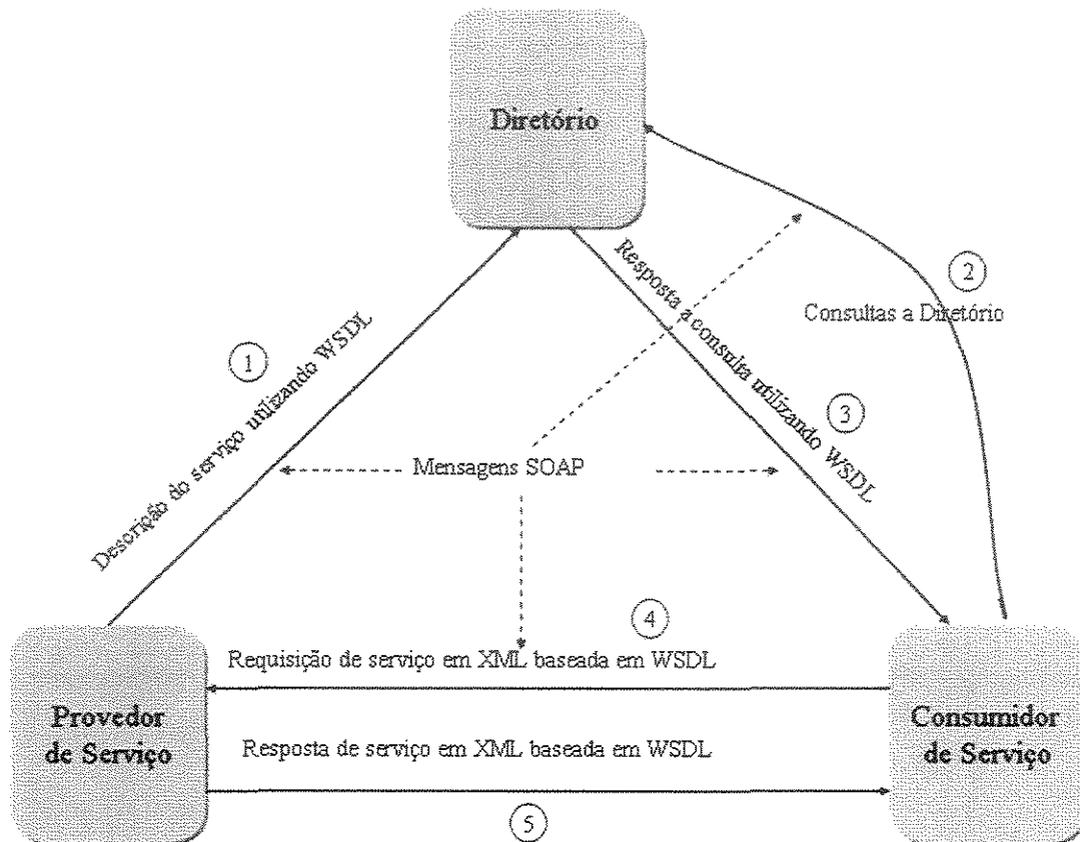


Figura 2.5: Comunicação via *Web Services*.

Para que a publicação e acesso aos serviços via *Web Services* possam ocorrer conforme descrito anteriormente e de forma a garantir a interoperabilidade entre os sistemas, foram definidos padrões para as três áreas envolvidas: comunicação, descrição de serviços e descoberta de serviços. A seguir estão descritas as especificações desenvolvidas para cada uma destas áreas.

Simple Object Access Protocol (SOAP)

O SOAP [W3C03] define um protocolo de comunicação baseado em XML para envio de mensagens e chamadas de procedimentos remotos (*Remote Procedure Call (RPC)*) em um ambiente heterogêneo e distribuído. Ao invés de definir um novo protocolo de transporte, o SOAP utiliza protocolos existentes como *Hypertext Transfer Protocol (HTTP)*, *Simple Mail Transfer Protocol (SMTP)* e *MQSeries* [CDK⁺02].

A estrutura de uma mensagem SOAP é muito simples: cada mensagem contém um elemento XML mais externo chamado Envelope, o qual contém dois elementos: o corpo da mensagem (*Body*) obrigatório e um cabeçalho (*Header*) opcional. O elemento *Body* possui o conteúdo da mensagem propriamente dito. Em uma mensagem de requisição, ele contém o nome e os parâmetros da operação a ser executada. O elemento *Header* consiste de múltiplos blocos, cada qual contendo meta-informação para o receptor da mensagem. Os blocos do cabeçalho também contêm informações adicionais como senha para autenticação.

A mensagem SOAP possui a seguinte estrutura:

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env='http://www.w3.org/2003/05/soap-envelope'>
  <env:Header>
    <!--Cabeçalho -->
  </env:Header>
  <env:Body>
    <!-- Corpo da mensagem -->
  </env:Body>
</env:Envelope>
```

Web Services Description Language (WSDL)

WSDL [W3C04] descreve os serviços do *Web Service* através de XML, fornecendo uma documentação do serviço para que possíveis clientes possam utilizá-lo de forma automatizada. O WSDL apresenta a definição de um *Web Service* em duas partes. A primeira representa uma definição abstrata independentemente do protocolo de transporte, enquanto a segunda representa uma descrição da ligação específica para o transporte na rede [IBM01].

Conforme descrito em [Amo04], um WSDL contém os seguintes elementos:

- Tipo de Porta (*portType*): Os elementos *portType* contêm um conjunto de operações representadas como elementos *operation* que possui um atributo *name* e um outro atributo opcional que especifica a ordem dos parâmetros usados nas operações que estão presentes em um *Web Service*. Além disso, ele pode ter apenas uma mensagem de entrada e uma mensagem de saída, da mesma maneira que uma chamada de método normal.
- Mensagem (*message*): Um elemento *message* contém uma definição dos dados a serem transmitidos. É semelhante ao parâmetro na chamada do método.

- Tipo (*type*): O elemento *type* contém os tipos de dados que estão presentes na mensagem.
- Ligação (*binding*): O elemento *binding* mapeia os elementos *operation* em um elemento *portType*, para um protocolo específico.
- Serviço (*service*) e Porta (*port*): Os elementos *service* e *port* (contido no *service*) incluem a localização da implementação do serviço na rede, ou seja, contém a informação para onde deve ser enviada a solicitação do serviço.

Segue um exemplo de um documento WSDL:

```

<message name='GetFlightInfoInput'>
  <part name='airlineName' type='xsd:string' />
  <part name='flightNumber' type='xsd:int' />
</message>

<message name='GetFlightInfoOutput'>
  <part name='flightInfo' type='fixed:FlightInfoType' />
</message>

<portType name='AirPortServicePortType'>
  <operation name='GetFlightInfo'>
    <input message='tns:GetFlightInfoInput' />
    <output message='tns:GetFlightInfoOutput' />
  </operation>
</portType>

<binding name='AirportServiceSoapBinding' type='TNS:AirPortServicePortType'>
  <soap:binding transport='http://schemas.xmlsoap.org/soap/http' />
  <operation name='GetFlightInfo'>
    <soap:operation style='rpc' soap:action='http://acme-travel/flightinfo' />
    <input>
      <soap:body use='encoded' namespace='http://acme-travel.com/flightinfo' encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
    </input>
    <output>
      <soap:body use='encoded' namespace='http://acme-travel.com/flightinfo' encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
    </output>
  </operation>
</binding>

```

```
        </operation>
    </binding>

    <service name='travelService'>
        <port name='travelServicePort' binding='tns: AirportServiceSoapBinding'
        >
            <soap:address location=http://acmetravel.com/travelService/>
        </port>
    </service>
```

Universal Description, Discovery, and Integration (UDDI)

Em seu artigo intitulado “*Unraveling the Web Services Web*”, Curbera [CDK⁺02] define o UDDI como um padrão que oferece ao usuário uma maneira única e sistemática de procurar provedores de serviços através de um registro centralizado, que é equivalente a uma “lista telefônica” de *Web Services*.

UDDI provê duas especificações básicas que definem a estrutura do registro de um serviço:

- Definição da informação que deve ser fornecida sobre cada serviço.
- API para consulta e atualização do registro, a qual descreve como a informação pode ser acessada e atualizada.

UDDI possui três tipos de informações associadas aos *Web Services*:

- Páginas Brancas: Contém o nome do provedor do serviço e informações para contato.
- Páginas Amarelas: Contém informações categorizadas por negócio ou tipo de serviço.
- Páginas Verdes: Contém informações técnicas sobre os serviços.

Através desta estrutura, consumidores e provedores de serviço podem acessar um registro único para procurar ou publicar serviços. O acesso ao registro é feito através de uma API SOAP padrão, tanto para consulta como para atualização das informações.

As especificações SOAP, WSDL e UDDI são básicas para o desenvolvimento de *Web Services* e já estão mais consolidadas, porém várias organizações como *World Wide Web Consortium (W3C)*, *Organization for the Advancement of Structured Information Standards (OASIS)*, *Liberty Alliance Project* e *Web Services Interoperability (WS-I)* desenvolvem especificações na área de automação de processos, segurança e confiabilidade de *Web Services* sem que haja um órgão unificador. Esta falta de consenso provoca incertezas nos desenvolvedores na escolha do padrão a ser adotado em uma aplicação. Além disso, vale

ressaltar que “implementações de *Web Services* de diferentes empresas ou organizações não são totalmente compatíveis e interoperáveis” [Lea04].

Outro fator a ser considerado na utilização da tecnologia *Web Services* é o desempenho das aplicações. Conforme já observado por Adam Bosworth [Bos03] em entrevista para a revista ACM Queue, “*Web Services* não são super eficientes”.

2.5 Considerações Finais

Vimos neste Capítulo que existem diversos tipos de Serviços Baseados em Localização, tanto para ambientes abertos como fechados. Algumas categorias de serviços baseados em localização são os serviços de localização de lugares ou pessoas, serviços de rotas, atendimento de emergências, gerenciamento de frotas ou força de trabalho, além de serviços de entretenimento.

Sendo a informação de localização do dispositivo móvel essencial para as aplicações de LBS, foram apresentadas algumas tecnologias de posicionamento que fornecem a posição do terminal com maior ou menor precisão. Algumas tecnologias são baseadas na rede celular, como Cell-ID, outras são baseadas no dispositivo móvel, como GPS e também existem soluções híbridas, que combinam a localização por GPS, por exemplo, com recursos da rede celular, que é o caso do A-GPS. A escolha da melhor tecnologia de posicionamento depende muito do tipo de serviço a ser oferecido. Serviços como localização de pontos de interesse não requerem uma alta precisão da localização, porém, serviços de emergência e determinação de rotas, exigem uma precisão maior.

Além das tecnologias de posicionamento, outras tecnologias envolvidas no desenvolvimento de Serviços Baseados em Localização são os Bancos de Dados Espaciais e *Web Services*. Os bancos de dados provêm as informações necessárias para a execução de um Serviço Baseado em Localização, como localização de endereços, pontos de interesse e informações em tempo real de tráfego e previsão do tempo.

O uso da tecnologia *Web Services* para acesso aos Serviços Baseados em Localização é um importante fator para garantia de interoperabilidade, uma vez que esta tecnologia permite que aplicações em diferentes plataformas e linguagens de programação se comuniquem através de padrões *web*. Porém, vale ressaltar que a tecnologia *Web Services* é muito recente. Ainda não existe um órgão que centralize o desenvolvimento de padrões e existem questões a serem resolvidas, como segurança, confiabilidade e desempenho.

Capítulo 3

A Especificação *Open Location Services*

Conforme já observado em um estudo feito em 2003 sobre o uso de padrões [Del03], há uma clara mudança de atitude em relação ao uso de padrões de *software*. Restrições econômicas e a necessidade de integração de sistemas em diferentes plataformas e linguagens de programação tornaram a adoção de padrões quase uma questão imperativa no desenvolvimento de *software*.

Além de garantir a interoperabilidade entre os sistemas, esta pesquisa também revelou que o uso de padrões diminui o custo de desenvolvimento de *software* uma vez que reduz a necessidade de aquisição de ferramentas proprietárias, aumenta a oferta de mão-de-obra qualificada no mercado, diminui a curva de aprendizado de novos funcionários e reduz o tempo de desenvolvimento, beneficiando-se dos esforços de análise e projeto já embutidos nos padrões.

A principal organização responsável pela definição de padrões abertos para Sistemas de Informações Geográficas é o *Open Geospatial Consortium*, um consórcio internacional, sem fins lucrativos, de mais de 255 empresas, agências de governo e universidades. Este capítulo descreve a especificação *Open Location Services*, aprovada em Janeiro de 2004 pelos membros do OGC, a qual define um conjunto de interfaces para o desenvolvimento de Serviços Baseados em Localização.

3.1 Arquitetura

A especificação OpenLS [OGC04a] define um conjunto de interfaces para o desenvolvimento de aplicações baseadas em serviços de localização com alto grau de interoperabilidade. Estas interfaces definem o acesso a serviços de diretório (como páginas amarelas), serviços de determinação de rotas, serviços de localização de dispositivos móveis, serviços

de geocodificação e geocodificação reversa e serviços de geração de mapas, todos utilizando protocolos padrão *web*.

A Figura 3.1 ilustra o padrão de acesso aos serviços OpenLS através de requisições e respostas baseadas em XML [W3C96].

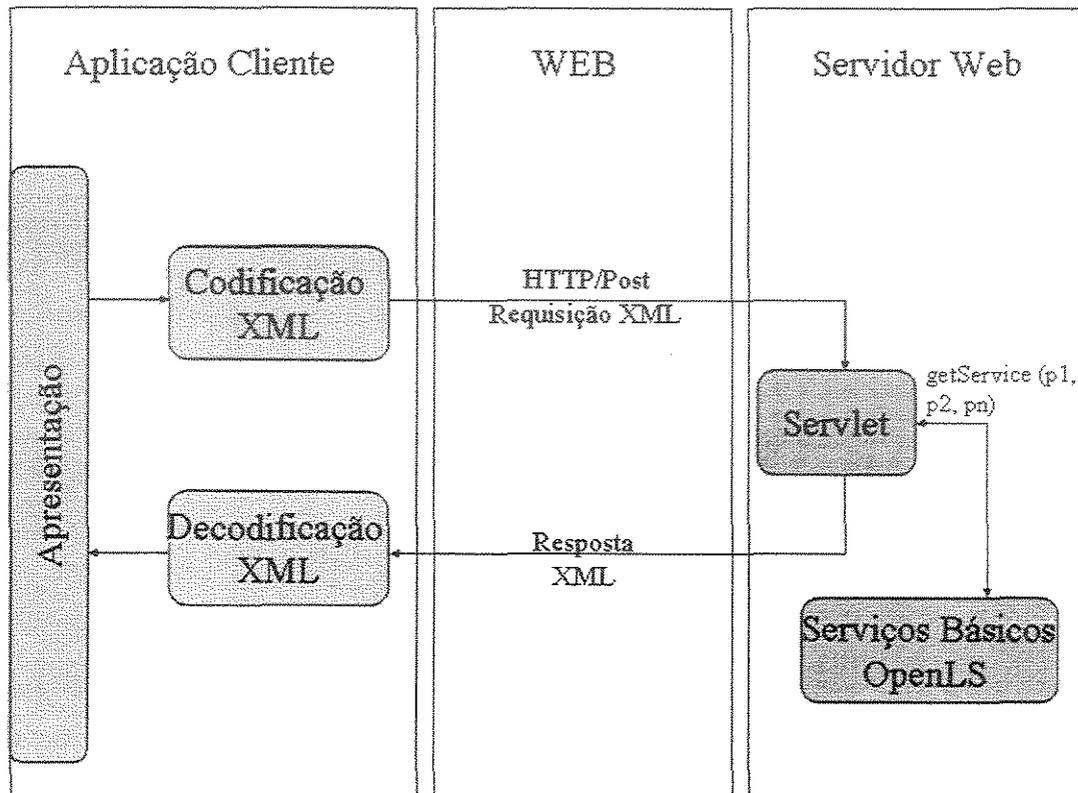


Figura 3.1: Padrão de Requisição/Resposta de Serviços OpenLS.

Um caso de uso padrão de utilização desta arquitetura se inicia com uma Aplicação Cliente processando a requisição de um serviço feita por um usuário. A Aplicação Cliente codifica a chamada ao serviço OpenLS como uma Requisição XML, enviando a requisição ao *Servlet* através do método HTTP/Post.

O *Servlet* analisa a Requisição XML e gera a chamada de função apropriada para o Serviço OpenLS. O Serviço OpenLS processa a Requisição e envia a Resposta de volta para o *Servlet*, que a codifica como uma Resposta XML e a envia para a Aplicação Cliente. Esta, por sua vez, decodifica a Resposta XML e aplica as funções de apresentação apropriadas para mostrar a resposta no dispositivo.

No caso de dispositivos muito simples, é possível que a Resposta seja processada no servidor e o conteúdo para exibição seja enviado diretamente para o dispositivo.

3.2 Serviços OpenLS

Os serviços básicos OpenLS trocam informações através de Tipos Abstratos de Dados (*Abstract Data Types* (ADT)) bem definidos:

- *Position* ADT - Localização de um ponto em um determinado sistema de coordenadas.
- *Address* ADT - Nome de rua ou cruzamento de ruas.
- *POI* ADT - Um determinado lugar, produto ou serviço.
- *Area of Interest (AOI)* ADT - Polígono, retângulo ou círculo que delimita a área de uma consulta.
- *Location* ADT - Uma localização (*Position*, *Address* ou *POI*).
- *Map* ADT - O desenho de mapas e elementos sobre eles (rotas e *POI*).
- *Route Summary* ADT - Metadados relativos a uma rota.
- *Route Geometry* ADT - Geometria de uma rota.
- *Route Maneuvers* ADT - Informações de navegação de uma rota.
- *Route Directions* ADT - Instruções de navegação passo a passo de uma rota.

As requisições e respostas aos serviços OpenLS são feitas em XML. O elemento raiz de uma requisição ou resposta é o *XLS*, o qual sempre possui os elementos *RequestHeader* e *Request* para uma requisição ou *ResponseHeader* e *Response* para uma resposta.

O elemento *RequestHeader* define o cabeçalho da mensagem de requisição e possui informações úteis para autenticação e identificação do cliente, como nome do cliente, senha, tipo do dispositivo e identificador da sessão. Já o elemento *Request* possui o corpo da mensagem de requisição do serviço, contendo informações tais como nome do método a ser chamado pelo serviço, versão da requisição implementada pelo cliente, identificador da requisição e número máximo de respostas que podem ser retornadas. Além destas informações comuns a toda requisição, o *Request* possui o elemento *_RequestParameters* que contém os parâmetros da requisição de cada serviço específico.

Assim como a requisição, a resposta possui um cabeçalho e o corpo da mensagem representados pelos elementos *ResponseHeader* e *Response*, respectivamente. O elemento *ResponseHeader* contém a lista de erros encontrados no processamento da requisição e o identificador da sessão, especificado pelo cliente no cabeçalho da requisição. A lista de erros também pode ser retornada dentro do corpo da mensagem. O elemento *Response* define a resposta retornada por um serviço. Ele contém a versão da resposta implementada pelo serviço, o identificador da requisição informado pelo cliente na requisição do serviço e o número de respostas geradas. A resposta de cada serviço específico é retornada no elemento *ResponseParameters*.

Os parâmetros de requisição e resposta de cada serviço OpenLS específico estão descritos nas seções a seguir. No apêndice B são apresentados exemplos de requisição e resposta para cada um dos serviços OpenLS.

3.2.1 Serviço de Diretório

Este serviço provê acesso a um diretório *online* para localização de um determinado lugar, produto ou serviço. Para especificar o lugar, produto ou serviço que se deseja localizar, o usuário pode informar seu nome, tipo, categoria, palavra chave, número de telefone ou algum outro identificador que seja de fácil entendimento. O usuário também deve especificar o tipo do diretório que se deseja localizar, por exemplo, páginas amarelas, guia de restaurantes, etc.

Uma posição geográfica também pode ser usada na consulta quando o usuário deseja localizar um lugar, produto ou serviço mais próximo de uma determinada localização, em uma localização específica ou dentro de uma determinada área. A posição geográfica neste caso pode ser a posição corrente do dispositivo móvel, determinada pelo serviço de *Gateway*.

O Serviço de Diretório retorna uma ou mais respostas para a consulta, com endereços e descrições completas do lugar, produto ou serviço desejado. As respostas são ordenadas segundo critérios da consulta.

Parâmetros da Requisição do Serviço de Diretório

A figura 3.2 ilustra os parâmetros da requisição do Serviço de Diretório. No elemento *DirectoryRequest* é possível informar dados do POI que se quer localizar e parâmetros de configuração da busca, conforme descrito a seguir.

- *POILocation* (opcional): Define o tipo de restrição para execução da busca. Pode conter um dos seguintes elementos:

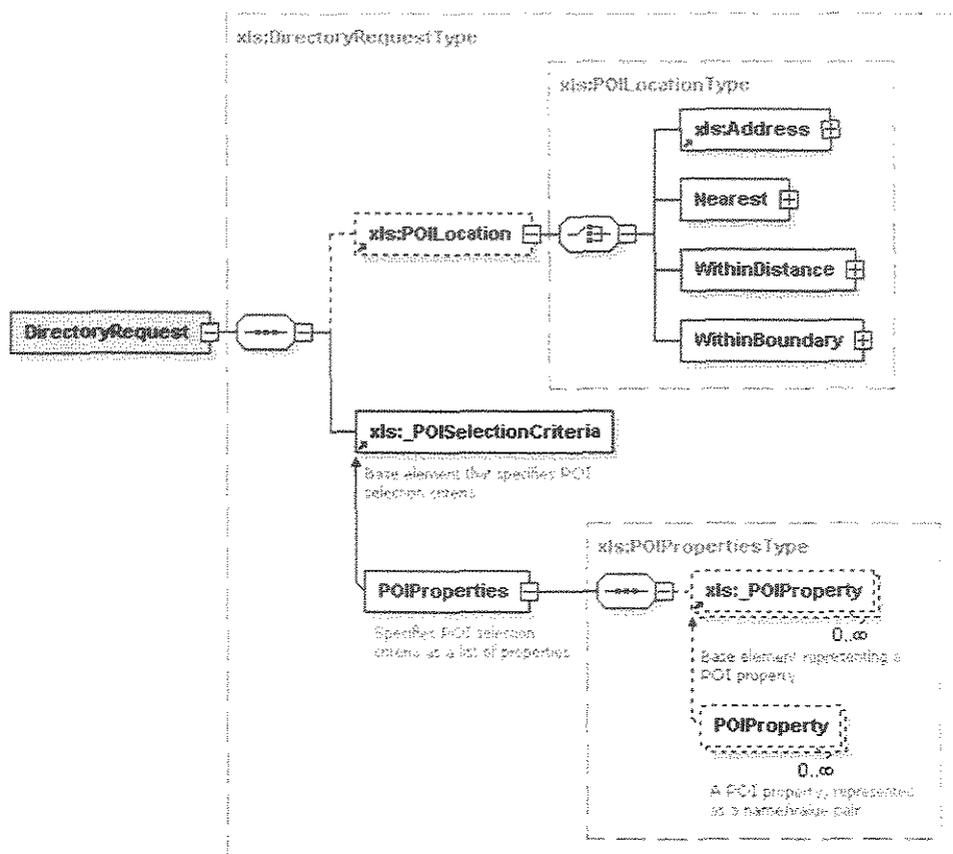


Figura 3.2: Requisição do Serviço de Diretório.

- *Address*: Endereço do POI. Este elemento será mais bem detalhado no serviço de geocodificação.
 - *Nearest*: Define um filtro espacial que seleciona o Ponto de Interesse mais próximo de uma dada localização.
 - *WithinDistance*: Define um filtro espacial que seleciona os Pontos de Interesse localizados a uma determinada distância de uma dada localização.
 - *WithinBoundary*: Define um filtro espacial que seleciona os Pontos de Interesse dentro de uma determinada área. Pode ser definido um círculo, polígono ou retângulo.
- *POIProperties*: Contém a lista de propriedades que serão usadas na busca. Possui o atributo *directoryType* e o elemento *POIProperty*:

- *directoryType*: Tipo de diretório a ser usado na busca, podendo assumir um dos seguintes valores: *White Pages*, *Yellow Pages*, *Green Pages*.
- *POIProperty*: Propriedade usada para procurar um POI, contém um nome e valor. As seguintes propriedades são válidas:
 - * ID
 - * POIName
 - * PhoneNumber
 - * Keyword (parâmetro genérico)
 - * NAICS_type¹
 - * NAICS_subtype
 - * NAICS_category
 - * SIC_type²
 - * SIC_subtype
 - * SIC_category
 - * SIC_code
 - * Outro (valor personalizado)

Parâmetros da Resposta do Serviço de Diretório

Um ou mais Pontos de Interesse podem ser retornados no elemento *POIContext* da resposta do Serviço de Diretório:

- *POIContext*: Uma lista de um ou mais POIs que satisfazem os critérios da requisição e suas distâncias da localização pedida. Contém os seguintes elementos:
 - POI: Ponto de Interesse encontrado. Cada ponto de interesse pode conter as seguintes informações:
 - * *POIAttributeList*: Lista de dados do POI na forma Nome/Valor.
 - * *Point*: Coordenada geográfica do POI.
 - * *Address*: Endereço do POI.
 - *Distance* (opcional): Distância da localização pedida. A distância será zero para a requisição com a opção *WithinBoundary*.

¹North American Industry Classification System (NAICS)

²Standard Industrial Classification (SIC)

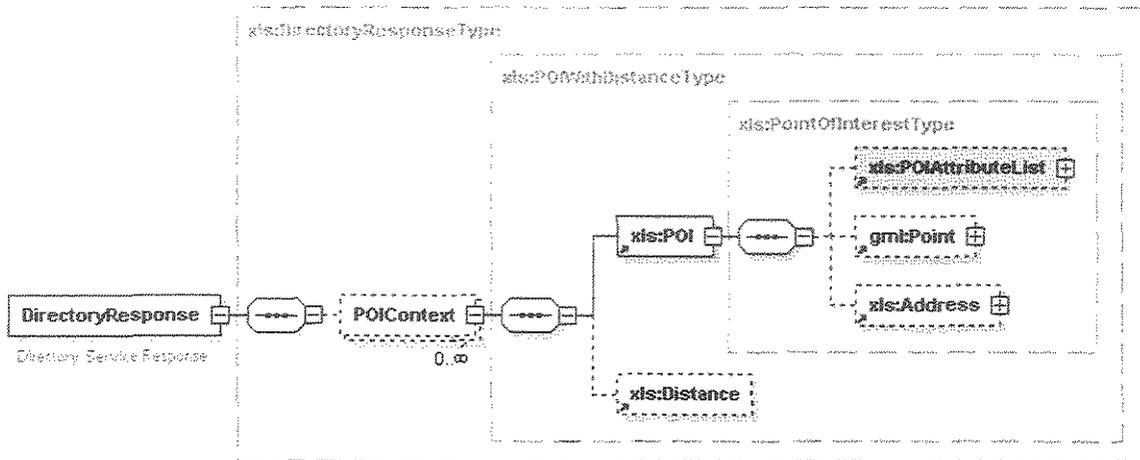


Figura 3.3: Resposta do Serviço de Diretório.

3.2.2 Serviço de *Gateway*

Este serviço identifica a posição geográfica de um determinado dispositivo móvel. Um Cliente envia uma requisição para o Serviço de *Gateway* a fim de determinar a posição do dispositivo. O *Gateway* calcula a posição e a envia de volta para o Cliente, o qual pode armazená-la pelo tempo que julgar necessário.

Parâmetros da Requisição do Serviço de *Gateway*

Na requisição do Serviço de *Gateway* devem ser informados dados do dispositivo que se quer localizar, além de parâmetros que definem a qualidade e tipo da busca, conforme detalhado a seguir:

- *locationType* (opcional) : define o tipo de localização desejada. Pode assumir um dos seguintes valores:
 - CURRENT: posição corrente do dispositivo. Se a posição corrente não puder ser determinada, é retornado um erro. Este é o valor *default*.
 - LAST: a última posição do dispositivo armazenada em *cache*.
 - CURRENT_OR_LAST: a última posição do dispositivo armazenada em memória mais recente do que um dado tempo, usualmente 15 min. Se não existir um registro, é retornada a posição corrente do dispositivo. Se nenhuma das duas alternativas for bem sucedida, é retornado um erro.

- *requestedcrsName* (opcional): sistema de coordenadas no qual deve ser retornada a posição do dispositivo.
- *Priority* (opcional): prioridade da requisição. Os valores válidos são NORMAL ou HIGH. *Default* é HIGH.

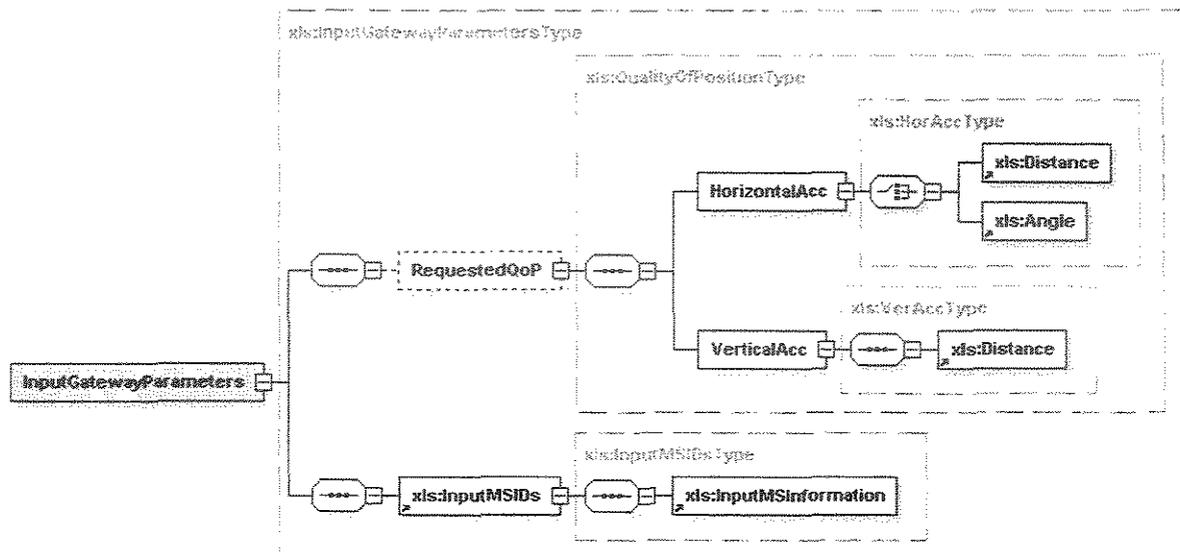


Figura 3.4: Requisição do Serviço de Gateway.

Além destes atributos a requisição possui os seguintes elementos:

- *RequestedQoP* (opcional): A qualidade da posição do dispositivo é dada em função da precisão horizontal e vertical, além do tempo de resposta.
- *InputMSID*: identificação do cliente móvel que se deseja localizar. O dispositivo móvel pode ser identificado através de uma das seguintes propriedades:
 - MSISDN: Mobile Station International ISDN Number
 - MIN: Mobile Identification Number
 - IMSI: International Mobile Subscriber Identity
 - IMEI: International Mobile Station Equipment Identity
 - MDN: Mobile Directory Number

- EME_MSID: Emergency MSID
- IPV4: Mobile station IP address (Versão 4)
- IPV6: Mobile station IP address (Versão 6)

Parâmetros da Resposta do Serviço de *Gateway*

A Resposta do Serviço de *Gateway* possui o elemento *OutputGatewayParameters*, o qual possui os parâmetros da resposta propriamente ditos:

- *locationType*: Mesmo atributo descrito na requisição.
- *requestedcrsName*: Mesmo atributo descrito na requisição.
- *Priority*: Mesmo atributo descrito na requisição.

Além dos atributos descritos acima, a resposta possui os elementos *RequestedQoP* e *OutputMSID*:

- *RequestedQoP*: Mesmo elemento descrito na requisição. Retorna a qualidade da localização obtida.
- *OutputMSID*: Identificação do cliente móvel, conforme descrito no elemento *InputMSInformation* da requisição, e sua posição geográfica, retornada no elemento *Position*.

3.2.3 Serviço de Geocodificação e Geocodificação Reversa

Este serviço determina uma posição geográfica, dado um nome de um lugar, endereço ou código postal. Ele também retorna a descrição completa do lugar, a qual pode ser útil quando o usuário não possui informações completas. Este serviço também funciona como um geocodificador reverso retornando um endereço completo, dada uma posição geográfica. Os dois serviços podem retornar nenhuma, uma ou várias respostas, dependendo das informações da requisição, do algoritmo utilizado e dos critérios de busca.

Parâmetros de Requisição do Serviço de Geocodificação

O Serviço de Geocodificação recebe em sua requisição uma coleção de um ou mais endereços, *Address*, que devem ser geocodificados.

O elemento *Address* pode ser um endereço com formato livre (*freeFormAddress*), um endereço estruturado (*StreetAddress*) ou uma interseção de ruas (*StreetIntersection*).

Além destes elementos obrigatórios, o *Address* possui os seguintes elementos opcionais:

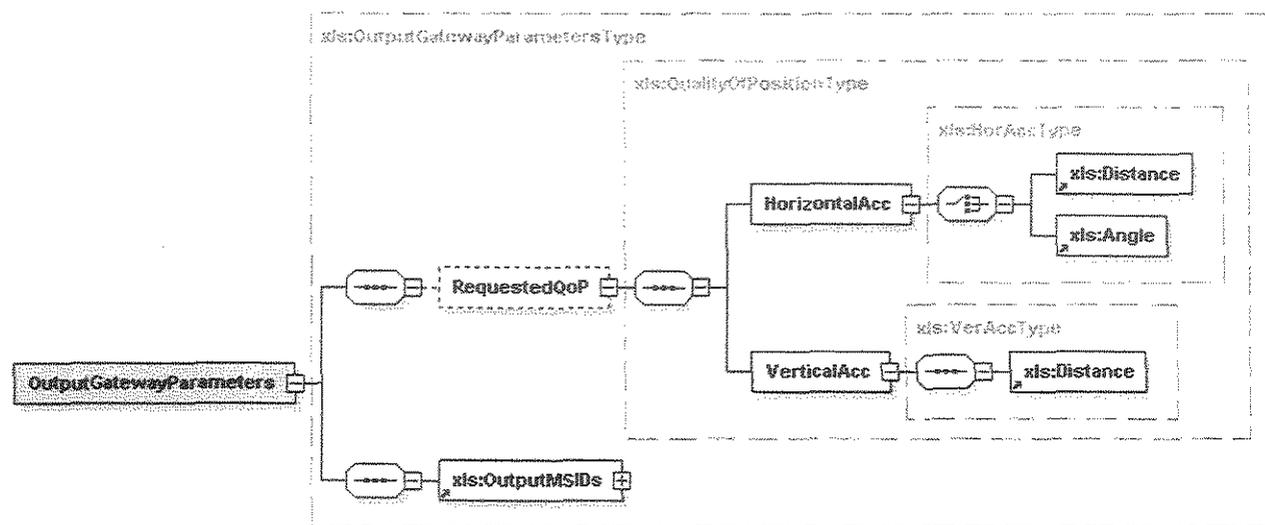


Figura 3.5: Resposta do Serviço de Gateway.

- *Place* (opcional): este elemento representa um conjunto hierárquico de regiões geográficas. Os níveis hierárquicos podem ter a seguinte classificação:
 - *CountrySubdivision*: uma divisão dentro do País, por exemplo, o Estado.
 - *CountrySecondarySubdivision*: uma segunda possível divisão dentro do País, por exemplo, o Município.
 - *Municipality*: Localidade.
 - *MunicipalitySubdivision*: subdivisão dentro da Localidade, por exemplo, o Bairro.
- *PostalCode* (opcional): código postal.

O elemento *Address* também possui dois atributos:

- *addressee* (opcional): destinatário.
- *countryCode*: código do país segundo padrão ISO 3166 Alpha-2.

Conforme mencionado anteriormente, o endereço informado na requisição do Serviço de Geocodificação pode ter um formato estruturado na forma de um nome de rua e número, conforme apresentado na Figura 3.7:

O elemento *StreetAddress* define um endereço estruturado através dos seguintes elementos:

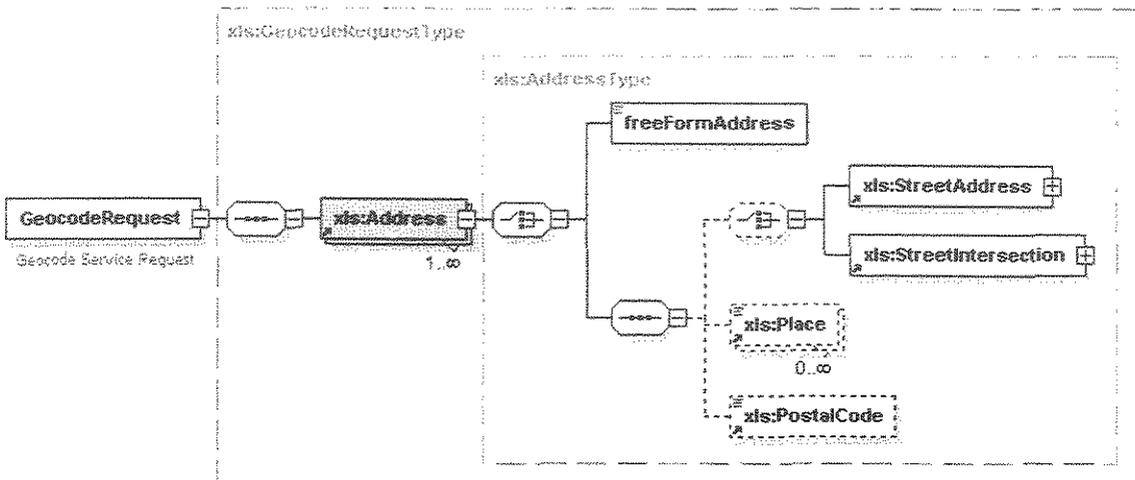


Figura 3.6: Requisição do Serviço de Geocodificação.

- *Street*: possui o nome estruturado de uma rua através dos seguintes atributos:
 - *directionalPrefix* (opcional): direção - prefixo.
 - *typePrefix* (opcional): tipo da rua - prefixo. Por exemplo: rua (R), avenida (Av).
 - *officialName* (opcional): nome oficial.
 - *typeSuffix* (opcional): tipo da rua - sufixo. Por exemplo: rua (R), avenida (Av).
 - *directionalSuffix* (opcional): direção - sufixo.
 - *munioctant* (opcional): localização do endereço em relação aos pontos cardeais: N, NE, NW, E, SE, S, SW, W.
- *Building*: um lugar endereçável, normalmente uma localização em uma rua, como por exemplo, um número, um apartamento, o nome do prédio, etc. Contém os seguintes atributos:
 - *number* (opcional): número do prédio ou residência.
 - *subdivision* (opcional): complemento do número. Por exemplo: conjunto, bloco, apartamento.
 - *buildingName* (opcional): nome do prédio.

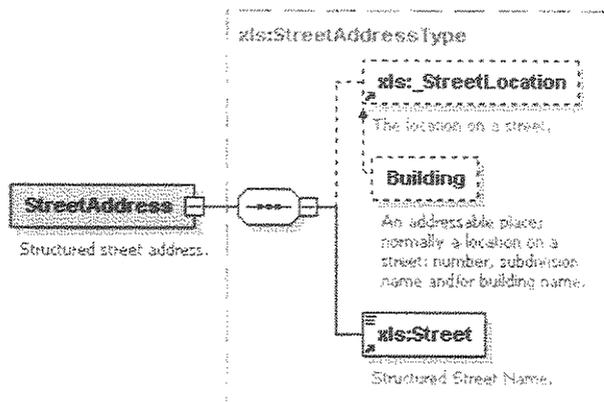


Figura 3.7: Elemento `StreetAddress` do Serviço de Geocodificação.

No elemento `Address` também pode ser informado a interseção de duas ruas através do elemento `StreetIntersection` representado na Figura 3.8.

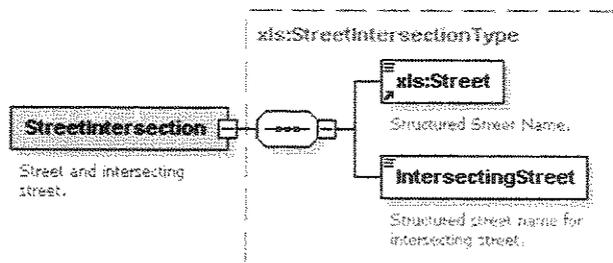


Figura 3.8: Elemento `StreetIntersection` do Serviço de Geocodificação.

Os elementos do `StreetIntersection` estão descritos a seguir:

- `Street`: nome estruturado de uma rua conforme já descrito anteriormente.
- `IntersectingStreet`: nome estruturado de uma rua que intercepta a rua informada no elemento `Street`.

Parâmetros da Resposta do Serviço de Geocodificação

Para cada endereço da requisição será retornada uma lista, `GeocodeResponseList`, dos endereços geocodificados.

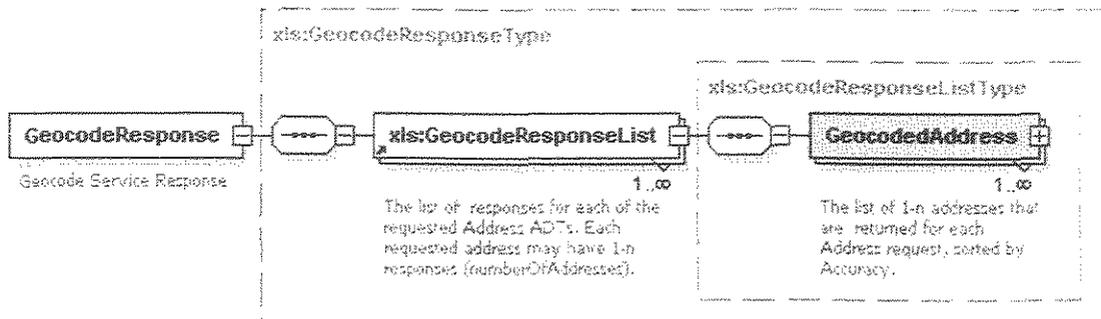


Figura 3.9: Resposta do Serviço de Geocodificação.

O elemento *GeocodeResponseList* possui o atributo *NumberOfGeocodedAddress*, além de uma lista de endereços geocodificados (*GeocodedAddress*) ordenados de acordo com a precisão:

- *NumberOfGeocodedAddress*: quantidade de endereços encontrados para o endereço informado na requisição. Mais de uma resposta pode ser retornada no caso do endereço informado na requisição não ser completo.

O elemento *GeocodedAddress* possui os seguintes parâmetros:

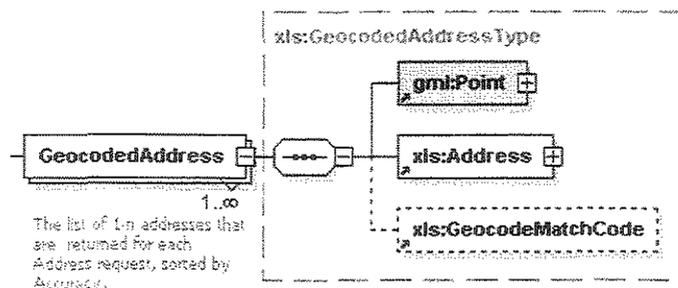


Figura 3.10: Elemento GeocodedAddress do Serviço de Geocodificação.

- *Point*: coordenada geográfica do endereço geocodificado.
- *Address*: endereço completo do endereço geocodificado.

- *GeocodeMatchCode*: provê informação da qualidade da geocodificação através de dois atributos:
 - *Accuracy* (opcional): valor que define a taxa de acerto de 0.0 a 1.0.
 - *matchType* (opcional): descreve o tipo de comparação usada pelo algoritmo de geocodificação.

Serviço de Geocodificação Reversa

O serviço de geocodificação reversa deve atender a uma série de requisitos:

- Dada uma Posição, deve retornar um ou mais endereços com suas respectivas localizações geográficas.
- O formato dos endereços retornados deve se basear nas preferências do usuário, definidas na requisição. O usuário pode especificar um nome de rua (*StreetAddress*), cruzamento de ruas (*StreetIntersection*) ou uma posição (*PositionOfInterest*).
- Deve ser capaz de retornar todas as informações de endereços dentro de uma área de interesse (AOI ADT - um círculo, polígono ou retângulo).
- Deve indicar o número de respostas (possivelmente zero) que satisfazem o critério de busca para uma dada requisição.

Parâmetros de Requisição do Serviço de Geocodificação Reversa

O elemento raiz da requisição do Serviço de Geocodificação Reversa é o *ReverseGeocodeRequest* que possui os elementos listados a seguir e ilustrados na Figura 3.11:

- *Position*: representa a coordenada geográfica da qual se deseja obter o endereço.
- *ReverseGeocodePreference* (opcional): Descreve o formato da resposta: *StreetAddress*, *IntersectionAddress* ou *PositionOfInterest* (lugar e/ou código postal). Se não for especificado, o endereço é retornado no formato *StreetAddress*.
- *SearchArea* (opcional): define uma área que restringe a busca do Serviço de Geocodificação Reversa. Pode ser um círculo, polígono ou retângulo.

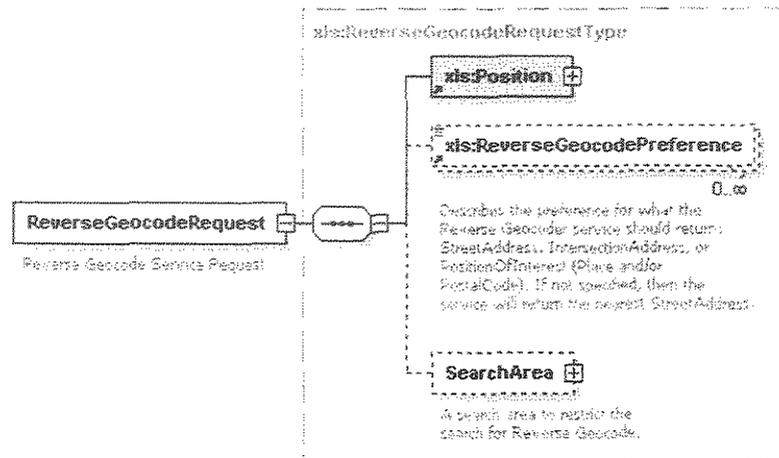


Figura 3.11: Requisição do Serviço de Geocodificação Reversa.

Parâmetros da Resposta do Serviço de Geocodificação Reversa

A resposta do Serviço de Geocodificação Reversa pode conter um ou mais endereços que atendem à requisição. Para cada endereço encontrado também são retornadas a sua posição exata e a sua distância da posição fornecida na requisição, conforme apresentado na Figura 3.12. Os elementos da resposta do Serviço de Geocodificação Reversa estão descritos a seguir:

- *Point*: posição exata do endereço localizado.
- *Address*: endereço encontrado na coordenada requisitada. O elemento *Address* foi descrito no Serviço de Geocodificação.
- *SearchCentreDistance* (opcional): distância da posição exata do endereço encontrado à posição fornecida na requisição.

3.2.4 Serviço de Apresentação de Mapas

Este serviço tem como objetivo apresentar informações geográficas no terminal móvel. Os mapas podem conter geometrias de rotas entre dois pontos, pontos de interesse (POI), áreas de interesse (AOI), localizações e/ou endereços. Este serviço também pode ser utilizado para a exibição de instruções de rotas retornadas pelo Serviço de Rotas.

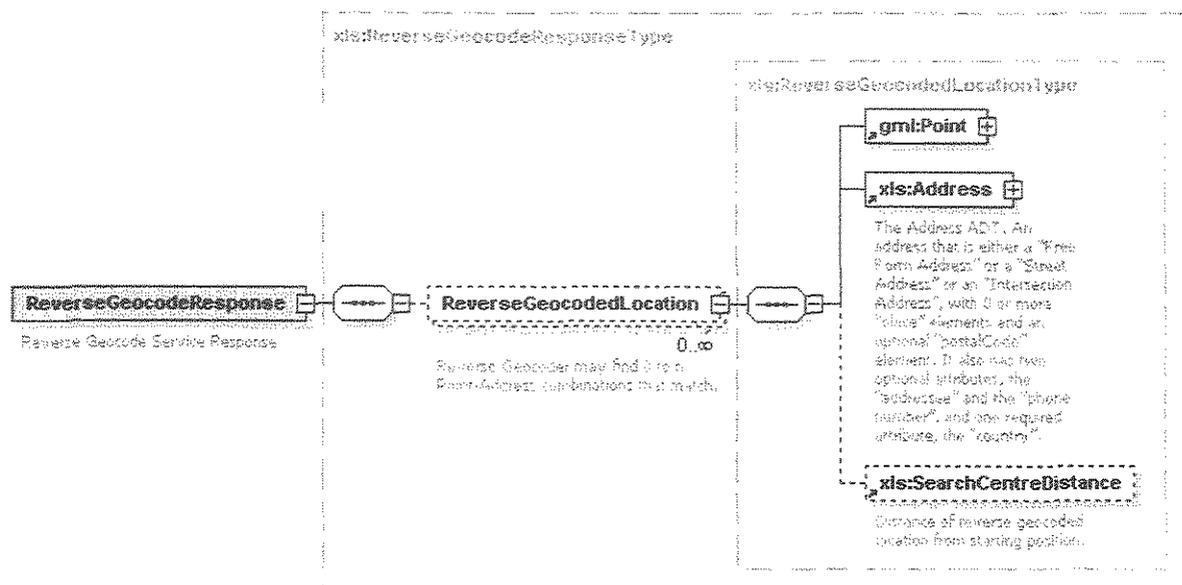


Figura 3.12: Resposta do Serviço de Geocodificação Reversa.

O propósito deste serviço é, em conjunto com outros serviços como o de Geocodificação, Rotas e *Gateway*, adicionar estilo às informações obtidas e apresentar nos dispositivos móveis mapas que sejam facilmente compreendidos pelo usuário.

Os requisitos deste serviço são:

- Possibilidade de definição de atributos de saída do mapa tais como largura, altura, formato, etc.
- Suporte a dados de contexto da visualização, permitindo a definição da área da terra que será exibida, sistema de coordenadas e escala de visualização.
- Definição das informações que serão exibidas, como por exemplo, uma rota entre o aeroporto e o hotel ou a localização de um hospital.
- Definição de um mapa base sobre o qual serão exibidas as informações solicitadas pelo usuário (desejável).
- Suporte a estilos que definem como as informações serão exibidas (não obrigatório).
- Suporte a estilos customizados pelo usuário (não obrigatório).
- Identificação da lista de recursos que a implementação contempla (não obrigatório).

Parâmetros de Requisição do Serviço de Apresentação de Mapas

A requisição do Serviço de Apresentação de Mapas, *PresentationRequest*, pode conter três elementos:

- *Output*: especifica a saída do mapa a ser gerado.
- *BaseMap* (opcional): especifica a lista de camadas que devem compor o mapa.
- *Overlay* (opcional): especifica a lista de tipos de dados que devem ser retornados sobre o mapa.

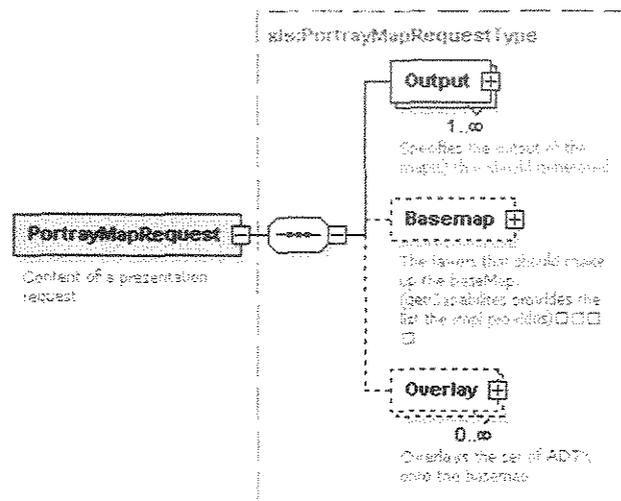


Figura 3.13: Requisição do Serviço de Apresentação de Mapas.

O elemento *Output* representado na Figura 3.14 contém os seguintes parâmetros:

- *width*: inteiro não negativo que define a largura do mapa em *pixels*.
- *height*: inteiro não negativo que define a altura do mapa em *pixels*.
- *format*: tipo *Multipurpose Internet Mail Extensions* (MIME) que define o formato do mapa. A função *getCapabilities* provê a lista de tipos que a implementação oferece, como por exemplo, PNG, JPG, etc.
- *BGColor* (opcional): cor de fundo do mapa no formato RGB. O valor *default* é "#FFFFFF" (branco).

- *transparent* (opcional): especifica a transparência do mapa. O valor *default* é *true*.
- *Content* (opcional): define como o mapa deve ser retornado. Assume um dos valores válidos: URL ou Data. *Default* é URL.

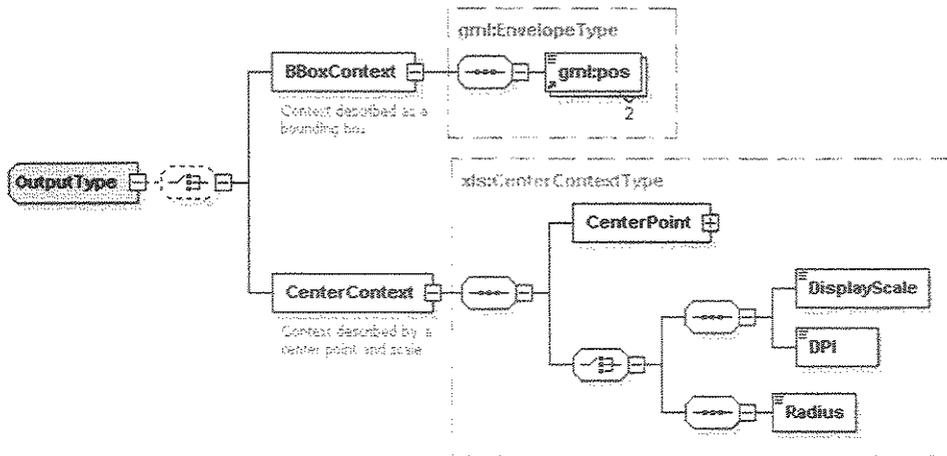


Figura 3.14: Elemento OutputType do Serviço de Apresentação de Mapas.

A área da terra a ser exibida pode ser especificada da seguinte forma:

- *BBoxContent*: retângulo definido por dois pontos em um determinado sistema de coordenadas que corresponde aos limites do mapa.
- *CenterContext*: define o centro do mapa em uma determinada escala e projeção. Possui as seguintes informações:
 - *CenterPoint*: ponto central do mapa especificado como uma coordenada lat/long no sistema de coordenadas WGS84³.
 - *Radius*: define o raio da área a ser exibida.
 - *DisplayScale*: define uma razão de distância. Por exemplo, 1:5000 é expresso como 5000, sempre expresso por 1 *pixel* da tela.
 - *DPI*: *Dots per Inch* do dispositivo.
 - *azimuth* (opcional): orientação do mapa.

³ World Geodetic System (WGS)

- SRS: Sistema de coordenadas no qual os dados serão projetados, no formato EPSG:XXXX. A função *getCapabilities* retorna a lista de valores oferecidos.

O elemento *BaseMap* da requisição é composto por um conjunto de camadas que devem ser incluídas ou excluídas do mapa, as quais podem ser informadas no elemento *LayerType* apresentado na Figura 3.15:

- *LayerType*: contém uma lista de 1 ou mais camadas que compõem o mapa, para cada camada é definido um filtro indicando se esta deve ser incluída ou excluída, além de um estilo que descreve como a camada deve ser exibida.

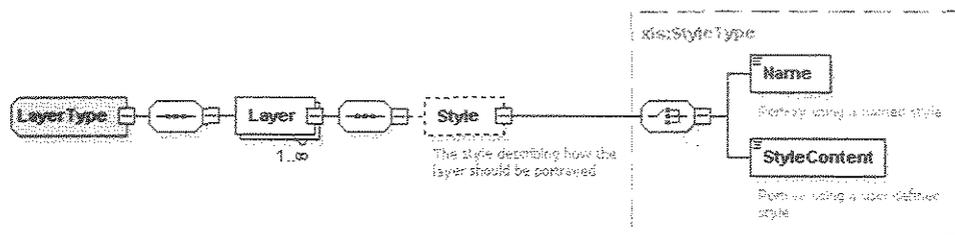


Figura 3.15: Elemento *LayerType* do Serviço de Apresentação de Mapas.

O elemento *Overlay* define os ADTs que devem ser exibidos sobre o mapa, podendo ser um dos seguintes:

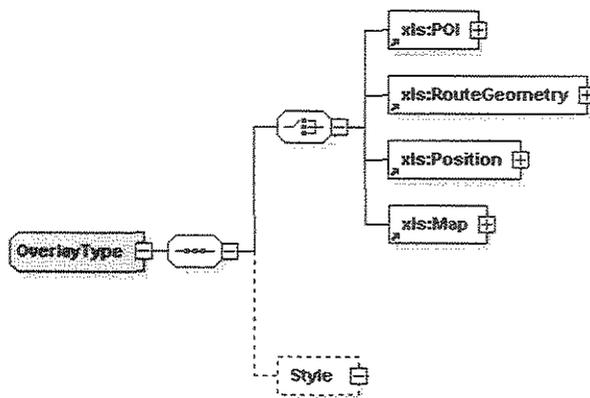


Figura 3.16: Elemento *Overlay* do Serviço de Apresentação de Mapas.

- *POI*: ponto de interesse, geralmente gerado a partir do Serviço de Diretório.
- *RouteGeometry*: geometria de uma rota, dada por uma seqüência de pontos.
- *Position*: uma coordenada geográfica.
- *Map*: um mapa.

Para cada ADT pode ser definido um estilo de exibição.

Parâmetros da Resposta do Serviço de Apresentação de Mapas

O Serviço de Apresentação de Mapas possui um Mapa como resposta, cujos elementos são os mesmos já descritos na requisição.

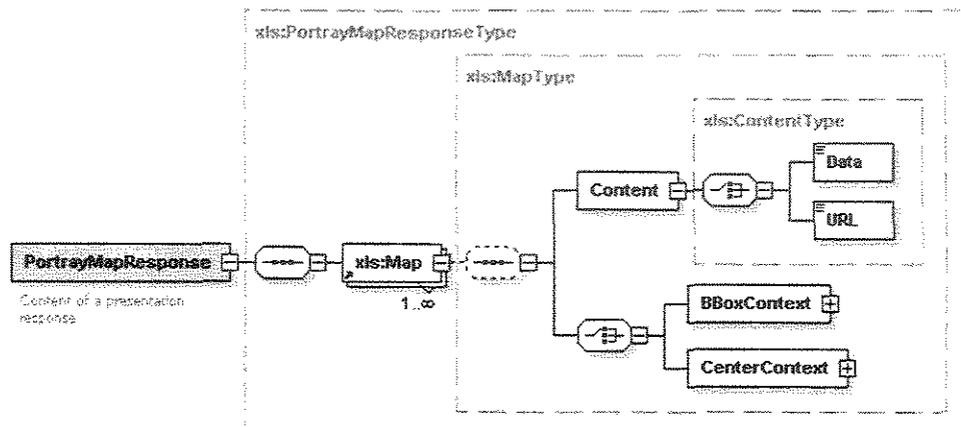


Figura 3.17: Resposta do Serviço de Apresentação de Mapas.

GetCapabilities

O Serviço de Apresentação de Mapas, além do *PresentationRequest*, possui um outro tipo de requisição, *GetPortrayMapCapabilitiesRequest*, que retorna informações a respeito dos serviços providos pela implementação, tais como:

- *Layers*: Lista de camadas providas pela implementação.
- *Style*: Lista de estilos providos pela implementação.
- *Format*: Lista de formatos providos pela implementação. *SRS*: Lista de Sistema de Coordenadas providos pela implementação, no formato EPSG:XXXX.

A resposta para esta requisição é retornada no elemento *GetPortrayMapCapabilities-Response*.

3.2.5 Serviço de Rotas

Este serviço determina a rota entre dois pontos informados pelo usuário. O usuário deve indicar um ponto de início (que pode ser a posição corrente do dispositivo móvel obtida através do serviço de *Gateway*) e o ponto final (qualquer localização, como um lugar do qual o usuário possui o endereço ou número do telefone, ou um lugar obtido do serviço de diretório). O usuário também pode, opcionalmente, informar pontos pelos quais a rota deve passar, rotas preferenciais (mais rápida, mais curta, menos tráfego, mais atrativa, etc.) e o modo de transporte. O usuário também pode armazenar a rota pelo tempo que desejar, sendo necessários recursos para recuperar uma rota armazenada.

O Serviço de Rotas deve prover as seguintes funcionalidades:

- Dado um conjunto de critérios, determinar uma rota entre dois pontos.
- Determinar uma nova rota, a partir dos critérios de uma rota existente. A nova rota pode ser diferente da primeira em função de mudanças em tempo real de tráfego, horário do percurso, restrições de transporte em função do horário ou alguma outra restrição.
- Após determinar a rota, retornar as seguintes informações:
 - Resumo das informações
 - Geometria da rota
 - Mapas da rota
 - Instruções de navegação passo a passo e advertências sobre a rota

Parâmetros de Requisição do Serviço de Rotas

O Serviço de Rotas recebe em sua requisição um elemento *DetermineRouteRequest* que contém os parâmetros da rota que se deseja determinar.

Os parâmetros da requisição estão descritos a seguir:

- *RouteInstructionsRequest* (opcional): possui o atributo *format* que especifica, na forma de um tipo MIME, o formato da apresentação das instruções de navegação e advertências geradas pelo Serviço de Rotas. O valor *default* é texto.
- *RouteGeometryRequest* (opcional): define os parâmetros para a determinação da geometria da rota. Possui as seguintes informações:

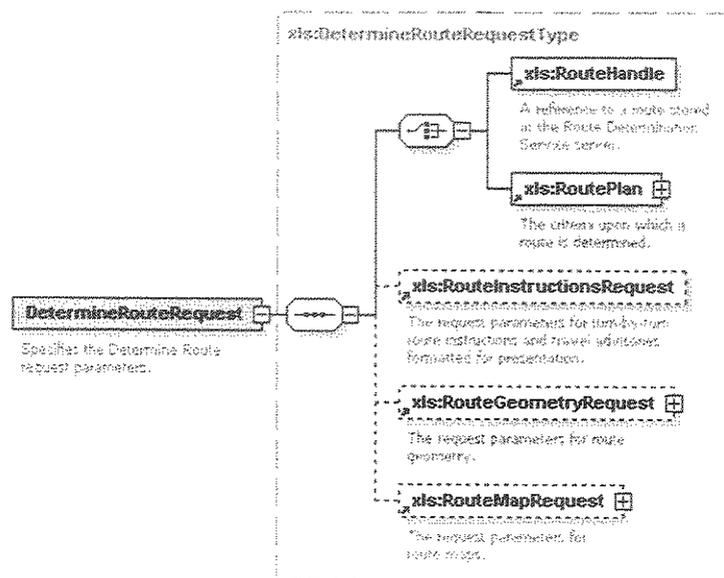


Figura 3.18: Requisição do Serviço de Rotas.

- *BoundingBox* (opcional): determina uma área retangular da rota para a qual se deseja visualizar a geometria. Se não for especificada, toda a rota será exibida.
- *scale* (opcional): escala máxima na qual a rota será exibida. Expressa na forma de unidades da terra por unidades do dispositivo. Por exemplo, 1:50000 deve ser expressa como 50000. Valor *default* é 1.
- *provideStartingPortion* (opcional): se o valor for *true*, retorna a geometria da porção inicial da rota contida dentro da área especificada até o número máximo de pontos definido. Se for *false*, retorna a geometria da rota completa contida dentro da área especificada, reduzindo a precisão da geometria, a fim de não exceder o número máximo de pontos definido.
- *maxPoints* (opcional): número máximo de pontos da geometria. O valor *default* é 100.
- *RouteMapRequest* (opcional): define os parâmetros do mapa a ser gerado para apresentar a geometria. Contém o elemento *Output* que possui as mesmas informações descritas no Serviço de Apresentação de Mapas.

Na requisição do Serviço de Rotas também deve ser informado um dos elementos *RouteHandle* ou *RoutePlan*:

- *RouteHandle*: referência para uma rota previamente armazenada no servidor do Serviço de Rotas. Na determinação desta nova rota, serão usados os critérios definidos para aquela rota previamente armazenada.
- *RoutePlan*: define os critérios de determinação da rota através dos seguintes parâmetros:
 - *RoutePreference*: preferência a ser levada em consideração na determinação da rota a fim de minimizar o tempo ou a distância da rota para veículo ou obter a melhor rota a pé.
 - *WayPointList*: lista de pontos pelos quais a rota deve passar. Definida em termos de um ponto inicial, zero ou mais pontos intermediários e um ponto final. Os pontos podem ser informados na forma de um endereço (*Address*), ponto de interesse (POI) ou uma posição geográfica (*Position*).
 - *AvoidList* (opcional): lista de áreas (AOI), localizações (endereços, POIs ou posições) e tipos de *features* pelos quais a rota deve evitar passar. Para os tipos de *features*, os seguintes valores podem ser informados:
 - * *HighWay* - minimizar o uso de auto-estradas
 - * *TollWay* - minimizar pedágios
 - *useRealTimeTraffic* (opcional): indica se devem ser usadas informações de tráfego em tempo real na determinação da melhor rota. O valor *default* é *false*.
 - *expectedStartTime* (opcional): especifica no formato YYYY-MM-DD HH:MM o dia e hora do início do trajeto. O valor *default* é a hora atual.

Parâmetros da Resposta do Serviço de Rotas

A resposta do Serviço de Rotas, *DetermineRouteResponse*, é uma coleção de um ou mais dos elementos apresentados na Figura 3.19 e descritos a seguir:

- *RouteHandle* (opcional): referência para a rota armazenada no servidor do Serviço de Rotas.
- *RouteSummary*: especifica as características da rota em função do tempo estimado do percurso completo da rota, distância total da rota e área retangular que cobre a rota completa.
- *RouteGeometry* (opcional): polilinha que define a geometria da rota.

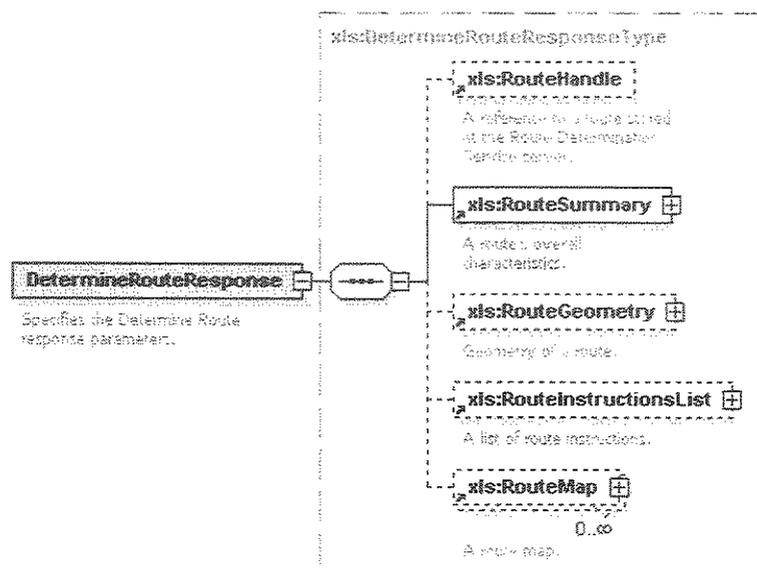


Figura 3.19: Resposta do Serviço de Rotas.

- *RouteInstructionList* (opcional): uma lista de instruções de navegação passo a passo e advertências sobre a rota, formatadas para apresentação. Para cada trecho da rota é dada uma instrução do caminho a ser percorrido, a sua duração e distância.
- *RouteMap* (opcional): define um mapa da rota.

3.3 Outros Padrões Envolvidos

O Serviços OpenLS utilizam outros padrões OGC na execução de suas funções, tais como o WMS utilizado pelo Serviço de Apresentação de Mapas para geração de mapas e o WFS utilizado pelo Serviço de Rotas e Serviço de Geocodificação para recuperação dos objetos da base de dados.

Estes e outros padrões OGC envolvidos no desenvolvimento dos Serviços OpenLS estão descritos a seguir.

3.3.1 *Geography Markup Language (GML)*

A especificação GML 3.0 [OGC03a] define um vocabulário XML específico para representação de objetos espaciais. São definidos mecanismos e sintaxes que representam in-

formações alfanuméricas e geográficas em XML, permitindo o armazenamento e troca de dados espaciais.

3.3.2 *Web Feature Service (WFS)*

A especificação WFS 1.0 [OGC02] define interfaces para que dados de um servidor de objetos espaciais (*features*) possam ser acessados de forma independente do esquema de armazenamento.

Um servidor WFS básico desempenha os seguintes papéis:

1. Informar quais as *features* disponíveis, as operações (consulta, inserção, atualização e remoção) que podem ser executadas e os tipos de operadores (AND, OR, WITHIN, etc.) oferecidos para definição de filtros.
2. Descrever a meta informação de um tipo de *feature* (*feature type*): atributos, relacionamentos, etc.
3. Retornar *features* especificadas por um filtro. Para descrição das *features*, o formato GML deve ser obrigatoriamente oferecido.

Estes serviços podem ser requisitados por uma aplicação cliente utilizando, respectivamente, as seguintes interfaces definidas pela especificação WFS: *GetCapabilities*, *DescribeFeatureType* e *GetFeature*.

3.3.3 *Web Map Service (WMS)*

A especificação WMS 1.3.0 [OGC04b] padroniza interfaces que devem ser utilizadas por clientes para requisitar os mapas aos servidores e também padroniza o modo como estes servidores devem descrever e retornar estes mapas.

Um servidor WMS básico é capaz de:

1. Gerar mapas georeferenciados (como uma imagem ou um conjunto de objetos gráficos).
2. Responder perguntas sobre o conteúdo de um mapa, retornando informações sobre um determinado objeto (*feature*) do mapa.
3. Descrever quais mapas ele pode produzir e quais podem ou não ser consultados, para que um cliente deste servidor saiba quais mapas podem ser requisitados.

Estes serviços podem ser requisitados pelo cliente utilizando as três interfaces definidas pela especificação WMS:

1. *GetMap* (obrigatória) para requisitar um mapa. Na requisição devem ser especificados parâmetros como o *layer*, a área que deve ser mapeada (*extent*), o sistema de coordenadas e nome do estilo.
2. *GetFeatureInfo* (opcional) para consultar o mapa. Na requisição deve ser especificada a coordenada em que deve ser feita a consulta.
3. *GetCapabilities* (obrigatória) para descrever os mapas.

3.3.4 *Web Coverage Service (WCS)*

A especificação WCS 1.0.0 [OGC03b] define interfaces para a troca de informações geográficas que representam fenômenos de variação espaço-temporal, como por exemplo, variação da área de uma floresta no decorrer do tempo.

Assim como as especificações WMS e WFS, WCS permite que clientes recuperem informações geográficas de um servidor com base em restrições espaciais ou outros critérios. Porém, ao contrário da especificação WMS que recupera os objetos e os retorna em mapas, WCS também retorna informações detalhadas associadas aos objetos, permite consultas complexas nestes dados e retorna informações semânticas sobre cada objeto, de forma que estas informações possam ser interpretadas e não apenas exibidas em um mapa.

Em comparação ao WFS, que retorna objetos espaciais discretos, *Web Coverage Service* retorna representações de fenômenos de variação do espaço que relacionam um domínio espaço-temporal a um conjunto de propriedades.

De maneira análoga às especificações WMS e WFS, WCS possui as seguintes interfaces: *GetCapabilities*, *GetCoverage* e *DescribeCoverage*.

3.4 Considerações Finais

Já é um consenso na indústria, em órgãos de pesquisa e setores públicos a importância da adoção de padrões no desenvolvimento de *software*. Além de garantir a interoperabilidade entre os sistemas, o uso de padrões traz uma série de benefícios como a redução do custo de desenvolvimento, o aumento da oferta de mão-de-obra qualificada no mercado, diminuição da curva de aprendizado de novos funcionários e redução do tempo de desenvolvimento.

Tendo como visão “um mundo no qual todas as pessoas se beneficiem da informação geográfica e serviços sejam oferecidos em qualquer rede, aplicação ou plataforma” [OGC94], o OGC vem, desde 1994, desenvolvendo padrões abertos para Sistema de Informação Geográfica. Em Janeiro de 2004, foi aprovada pelos membros do OGC a especificação Open Location Services que define um conjunto de interfaces para o desenvolvimento de Serviços Baseados em Localização.

Neste Capítulo foram descritos os serviços que integram a especificação OpenLS:

- Serviço de Diretório para busca por lugares, produtos ou serviços.
- Serviço de *Gateway* para localização de dispositivos móveis.
- Serviço de Geocodificação/Geocodificação Reversa para determinação do endereço a partir de uma coordenada geográfica ou vice-versa.
- Serviço de Apresentação de Mapas para visualização de mapas contendo posições, pontos de interesse e/ou rotas.
- Serviço de Rotas para determinação de rotas entre dois pontos.

A especificação OpenLS é muito recente e ainda apresenta algumas deficiências, por isso já existem iniciativas no OGC para desenvolvimento de uma nova versão da especificação. As principais evoluções dizem respeito à compatibilização desta especificação com as demais especificações WMS, WFS e WCS também do OGC e inclusão de suporte a SOAP.

Capítulo 4

Arquitetura Proposta

A Computação Orientada a Serviços (*Service-Oriented Computing* (SOC)) é um paradigma que utiliza serviços como elementos fundamentais para o desenvolvimento de aplicações [Pap03]. Para construir este modelo de serviços, a SOC se baseia na Arquitetura Orientada a Serviços (*Service-Oriented Architecture* (SOA)), que é uma forma de reorganização de aplicações de *software* em um conjunto de serviços que interagem entre si. Segundo definição de Papazoglou [Pap03], “os serviços são elementos auto-descritivos, multiplataforma que permitem a construção de aplicações distribuídas de forma rápida e com baixo custo”.

Outras arquiteturas como *Common Object Request Broker Architecture* (CORBA) [OMG04], JavaRMI (Java Remote Method Invocation) [Sun02] e *Distributed Component Object Model* (DCOM) [Mic96] também têm como objetivo promover a comunicação entre sistemas distribuídos, mas todas elas apresentam uma série de problemas como dificuldades de comunicação com o código legado de outras plataformas, a falta de suporte para linguagens de programação distintas e a complexidade das plataformas [Amo04].

Como os serviços provêm a distribuição da informação de forma uniforme e ubíqua para uma grande variedade de dispositivos de software (computadores de mão, PDAs, telefones celulares) e plataformas, eles constituem um conceito promissor na computação distribuída. Diferentes infra-estruturas podem ser utilizadas para facilitar a comunicação e interação entre serviços, uma vez que estes podem ser implementados em uma única máquina, podem ser distribuídos em uma rede local ou através da Internet. Um caso específico são os *Web Services* que utilizam a Internet para se comunicar através de padrões *web* abertos e são descritos e acessados através de padrões baseados em XML.

No caso dos Serviços Baseados em Localização, além da utilização da tecnologia *Web Services* como forma de garantir a interoperabilidade entre as aplicações, é importante a adoção de padrões abertos nas interfaces de acesso aos serviços. A especificação OpenLS do OGC descrita no Capítulo 3 define um conjunto de interfaces para o desenvolvimento

de aplicações de LBS com alto grau de interoperabilidade, todas baseadas em padrões *web*.

Neste Capítulo é apresentada uma proposta de arquitetura baseada em padrões abertos para o desenvolvimento de Serviços Baseados em Localização utilizando *Web Services*. Também são apresentados os resultados do desenvolvimento de um protótipo de validação desta arquitetura, tendo como foco o Serviço de Apresentação de Mapas descrito no Capítulo 3.

4.1 Descrição da Arquitetura

Na arquitetura ilustrada na Figura 4.1, o usuário de dispositivo móvel acessa, via *Web Services*, a Aplicação LBS responsável por prover os serviços definidos na especificação OpenLS. Além de oferecer acesso aos serviços básicos OpenLS, a aplicação principal também possui funções de suporte aos Serviços Baseados em Localização como autenticação de usuário, registro e contabilização de acesso aos serviços, para fins de faturamento.

Após tratar a requisição do usuário, a Aplicação LBS acessa os Serviços OpenLS (Rotas, Geocodificação, Diretório, Apresentação e *Gateway*), também via *Web Services*, através das APIs definidas na especificação. Estes serviços básicos, por sua vez, acessam as bases de dados para obter informações necessárias para executar suas funções.

As bases de dados contêm mapas, rotas, endereços, dados de suporte a navegação e diretórios com informações de lugares, produtos ou serviços. Estes dados são acessados através de interfaces OGC WFS, WMS e WCS ou outras interfaces padronizadas. As informações de posição de dispositivos móveis são obtidas através da requisição de serviços a um servidor GMLC/MPC. Caso o dispositivo móvel possua GPS embutido, a informação de localização pode ser enviada pelo próprio dispositivo para o Serviço de *Gateway*.

Vale ressaltar que na arquitetura aqui proposta os serviços básicos e as bases de dados podem estar distribuídos em servidores distintos.

4.2 Estudo de Caso

A fim de validar a arquitetura apresentada na seção anterior, foi desenvolvido um protótipo no âmbito do projeto Serviços e Aplicações Móveis (SAM) em andamento na Fundação CPqD [SSY+04].

Trata-se de um projeto de Pesquisa Aplicada financiado pelo Fundo para o Desenvolvimento Tecnológico das Telecomunicações (FUNTTEL) que conta com a participação de equipes de várias especialidades da Fundação CPqD.

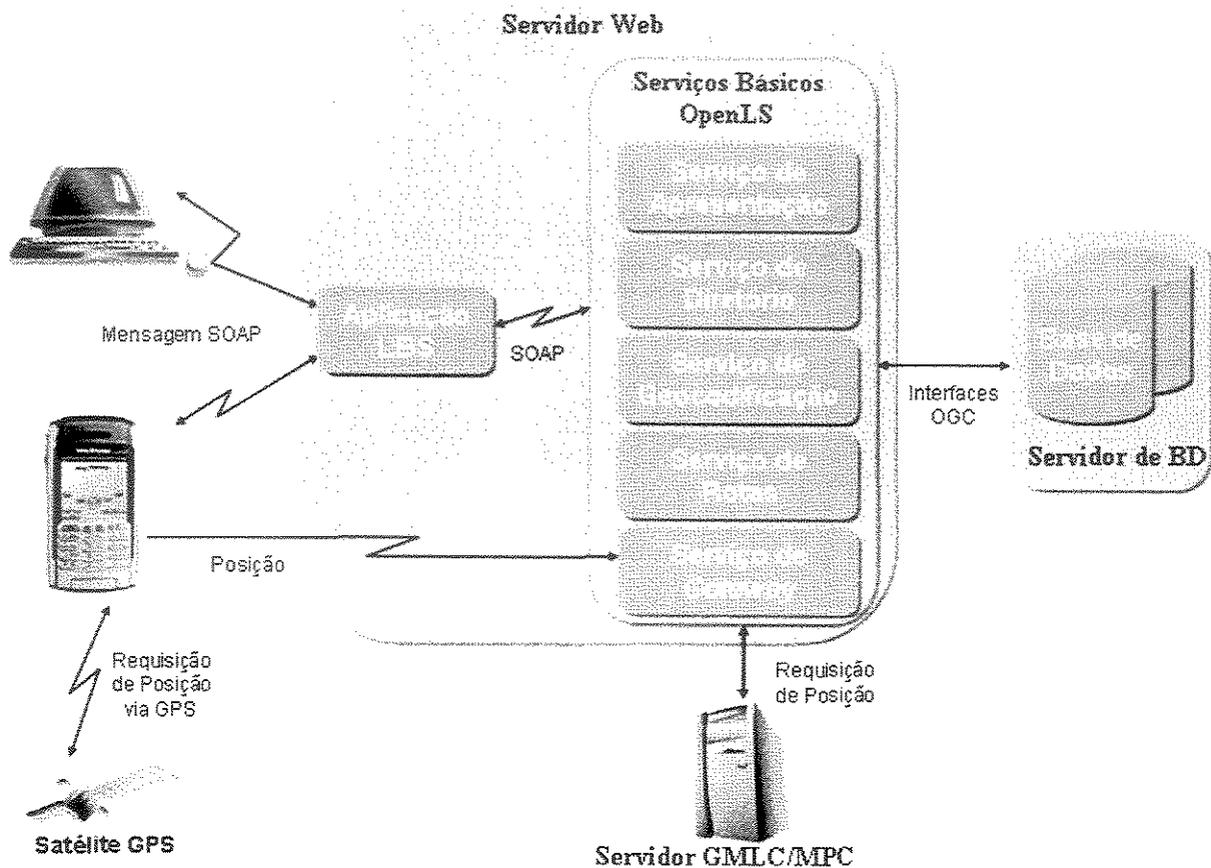


Figura 4.1: Arquitetura Proposta.

4.2.1 Descrição do Projeto

O projeto SAM tem como principal objetivo o desenvolvimento de uma arquitetura que tenha capacidade de prover serviços para administração pública, com forte apelo para o uso de *software* e padrões abertos. Além do uso de padrões abertos, foi definida a utilização da tecnologia *Web Services* para integração dos diferentes serviços que compõem o projeto.

A adoção de uma arquitetura orientada a serviços baseada em *Web Services* tem como objetivos reduzir os erros de transcrição de dados coletados em campo pelos agentes móveis, reduzir o tempo de divulgação das informações coletadas, garantir a precisão dos dados coletados através da utilização de informações georeferenciadas e dar maior poder de decisão aos agentes em campo, uma vez que estes passam a ter acesso a informações atualizadas em qualquer lugar, a qualquer hora, em qualquer dispositivo.

Estes objetivos são atingidos a partir da integração dos seguintes sistemas que compõem

o projeto SAM [SSY+04]:

- Sistema de Mobilidade para Serviços Públicos: responsável por oferecer serviços de sincronização de dados, acesso seguro a serviços, autenticação e autorização.
- Sistema de Geoposicionamento para Serviços Públicos: responsável por oferecer mapas georeferenciados, análises temáticas, definição de rotas otimizadas e visualização da localização dos agentes em campo.
- Sistema Gestor de Custos de Serviços de Dados Móveis: responsável por monitorar o uso dos serviços, gestão dos custos e gestão dos contratos.
- Sistema Gestor de Rede para Serviços Baseados em Localização: responsável por prover a localização dos dispositivos móveis, base de dados de localização e funções de gerenciamento de localização.

Esta dissertação tem como foco o Serviço de Apresentação de Mapas que integra o Sistema de Geoposicionamento para Serviços Públicos. Este serviço possibilita a geração de mapas georeferenciados para visualização das posições dos agentes móveis em campo e visualização de rotas otimizadas para envio dos agentes.

Conforme já mencionado, o projeto SAM se baseia na utilização de tecnologias e padrões abertos, sendo escolhidos os seguintes *softwares* e padrões para desenvolvimento do Serviço de Apresentação de Mapas: Apache AXIS como implementação do SOAP, Apache Tomcat como Servidor Web, MapServer como servidor de mapas, PostgreSQL com a extensão PostGIS como base de dados geográfica e a linguagem Java para implementação do serviço. As tecnologias utilizadas no desenvolvimento do protótipo estão descritas a seguir.

Apache AXIS

O AXIS da Apache [Apa00] é uma implementação modular, flexível e de bom desempenho do SOAP e foi adotado no projeto porque, dentre outras funcionalidades, possui extenso suporte a WSDL, pode ser utilizado em servidores de aplicação tais como Tomcat e possui ferramenta para geração de classes Java a partir do WSDL e vice-versa. Além disso, é do projeto Apache, especializado em fornecer soluções de código aberto com padrões de qualidade comercial.

Apache Tomcat

Desenvolvido pelo grupo Apache, o Tomcat [Apa99] é atualmente o servidor Web mais popular. Escrito em Java, é a implementação de referência das especificações das tecnologias de *Java Servlets* [Sun94a] e *Java Server Pages* [Sun94c] criadas pela Sun e pode

ser usado livremente seja para fins comerciais ou não. A escolha deste servidor também foi devido ao fato dele pertencer ao projeto Apache e possuir código aberto.

MapServer

O MapServer [Uni96] é um ambiente de desenvolvimento de código aberto para aplicações espaciais baseadas em Internet. Possui uma API em C e uma linguagem chamada *MapScript* que permite interação com outras linguagens como TCL, Perl, Python e Java. Há também módulos PHP/MapServer e Perl/MapServer, além da possibilidade de desenvolver aplicações através de *JavaScript*.

O servidor de mapas MapServer contempla os protocolos WMS e WFS do OGC. Ele oferece suporte a formatos vetoriais, como ESRI[©] shapefiles e ESRI[©] ArcSDE, e formatos *raster*, como TIFF/GeoTIFF, GIF, PNG e JPEG. O MapServer pode ser integrado aos bancos de dados Oracle e sua extensão Oracle Spatial e PostgreSQL com sua extensão PostGIS.

A adoção do MapServer neste projeto se deve ao fato de ser considerado um servidor de mapas mais estável, já que é utilizado por vários projetos, inclusive no Brasil.

PostgreSQL/PostGIS

O PostgreSQL com a extensão PostGIS apresentado no Capítulo 2, foi adotado neste projeto por ser um SGBD de código aberto. Além disso, é considerado mais robusto que o MySQL, dada sua arquitetura e as funcionalidades que implementa. Ele provê os recursos que o MySQL não possui de forma nativa, além de possibilitar transação sobre tabelas georeferenciadas. O PostgreSQL também possui o diferencial de ser um Sistema de Gerenciamento de Banco de Dados (SGBD) Objeto-Relacional, e como tal possui conceitos de herança de tabelas, por exemplo, que podem ajudar na modelagem de um sistema.

Java

A linguagem Java [Sun94b] foi considerada mais apropriada para o projeto devido a sua escalabilidade para implementação de soluções que atendam desde dispositivos móveis mais simples até soluções corporativas mais complexas [KR03]. Além disso, a linguagem Java, por ser multiplataforma, garante a ubiquidade das aplicações de LBS, permitindo que estas possam ser oferecidas em qualquer dispositivo.

4.2.2 Análise e Projeto

Nesta seção são apresentados os diagramas em UML (*Unified Modeling Language*) desenvolvidos durante a fase de Análise e Projeto do Serviço de Apresentação de Mapas. São apresentados os diagramas de Casos de Uso, diagramas de Classes e diagramas de Seqüência, sendo que cada um deles apresenta um nível de abstração diferente do sistema.

Diagrama de Casos de Uso

Os diagramas de casos de uso são utilizados para capturar os requisitos do sistema e especificar o seu comportamento. Eles representam uma visão alto nível do sistema na perspectiva de um ator externo.

Conforme ilustrado na Figura 4.2, no protótipo foram desenvolvidos os casos de uso de localização dos dispositivos móveis e visualização de um mapa com as posições dos agentes em campo, sendo que apenas o segundo é detalhado neste trabalho. Nos casos de uso em questão, a aplicação cliente requisita ao Serviço de *Gateway* a posição corrente dos agentes ou o histórico de localização informando o código IMEI dos dispositivos. Com base nas coordenadas obtidas do Serviço de *Gateway*, a aplicação cliente solicita ao Serviço de Apresentação de Mapas a geração de um mapa com as localizações dos dispositivos.

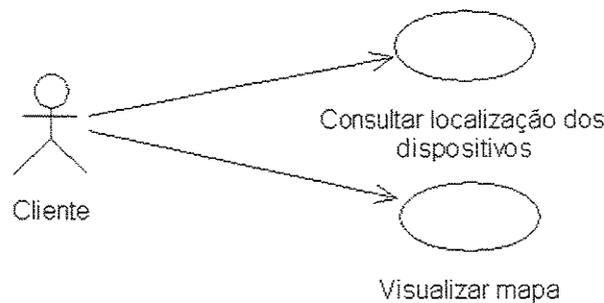


Figura 4.2: Diagrama de Casos de Uso.

Diagramas de Classes

O diagrama de classes oferece uma visão estática da estrutura do sistema, apresentando um conjunto de elementos tais como classes, interfaces e seus relacionamentos. As classes que compõem o Serviço de Apresentação de Mapas são organizadas em pacotes (Figura 4.3) de acordo com o seu propósito.

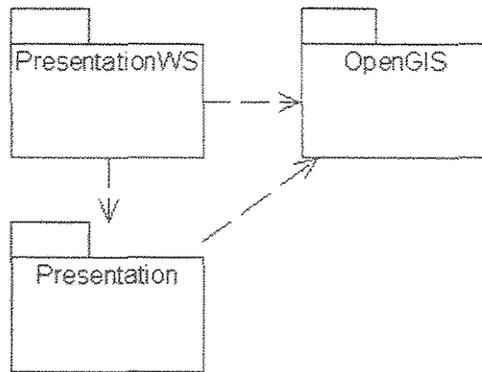


Figura 4.3: Organização dos Pacotes.

O pacote *PresentationWS* contém as classes que oferecem suporte para a comunicação entre consumidor e provedor do serviço via *Web Services*. Estas classes são geradas automaticamente pela ferramenta *WSDL2Java* do Axis a partir do documento WSDL do Serviço de Apresentação de Mapas.

A ferramenta gera a classe *PresentationServiceLocator* (Figura 4.4) usada pelo cliente para recuperar um *stub* (*PresentationServiceBindingStub*) para o serviço, a fim de se comunicar com um *skeleton* (*PresentationServiceBindingSkeleton*) localizado no servidor.

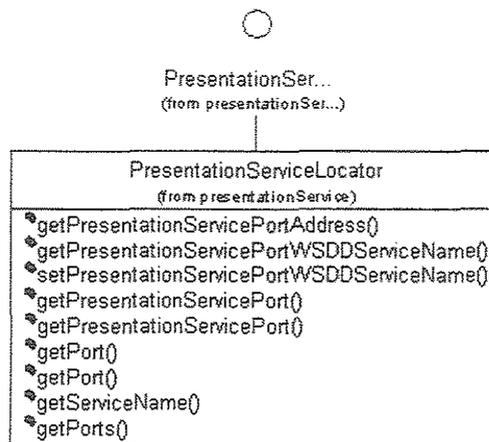


Figura 4.4: Classe *PresentationServiceLocator*.

A classe *PresentationServiceBindingSkeleton* (Figura 4.5) recebe a mensagem do cli-

ente e a encaminha para a classe *PresentationServiceBindingImpl*. Esta, por sua vez, apenas encaminha a mensagem para a classe que implementa o serviço propriamente dito no pacote *Presentation*.

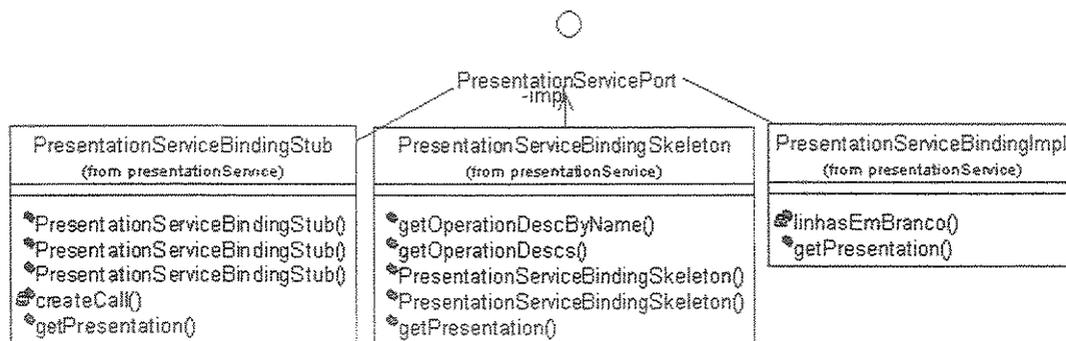


Figura 4.5: Classes Stub, Skeleton e Template de Implementação.

O pacote *Presentation* contém as classes que implementam o Serviço de Apresentação de Mapas propriamente dito.

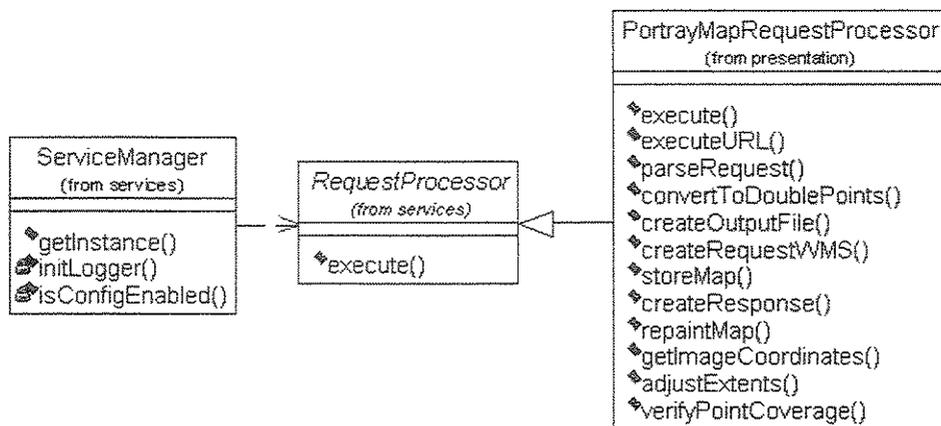


Figura 4.6: Classes do Pacote Presentation.

O pacote *OpenGIS* contém as classes correspondentes aos tipos de dados definidos na especificação OpenLS usados na requisição e resposta dos serviços. Estas classes também foram geradas pela ferramenta *WSDL2Java* do Axis a partir dos esquemas *XML Schema*

Definition (XSD) fornecidos na especificação OpenLS e incluídos no documento WSDL do Serviço de Apresentação de Mapas. Algumas classes deste pacote estão ilustradas na Figura 4.7.

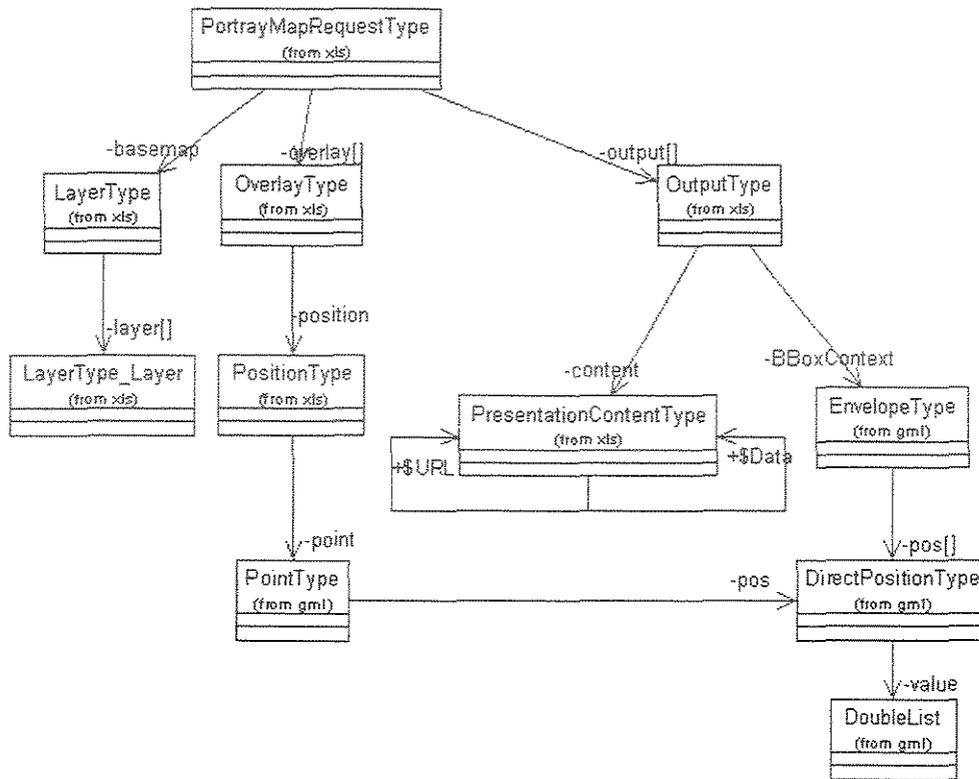


Figura 4.7: Classes do Pacote OpenGIS.

Diagrama de Sequência

Um Diagrama de Sequência é uma visão gráfica de um cenário de um caso de uso, mostrando a seqüência de interação entre os objetos e as mensagens trocadas entre eles. Diagramas de seqüência estabelecem os papéis dos objetos e ajudam a levantar informações essenciais para determinar as responsabilidades das classes e suas interfaces.

No caso de uso de visualização das localizações dos agentes em campo, o Serviço de Apresentação de Mapas recebe na requisição os parâmetros do mapa a ser gerado, como altura, largura, área da terra a ser visualizada, formato, além das posições dos agentes a serem desenhadas sobre o mapa. Com base nas informações recebidas, é construída uma

requisição WMS para o Servidor de Mapas. Este, por sua vez, acessa a base de dados geográfica para recuperar as informações necessárias, gera o mapa de acordo com os dados da requisição e o retorna para o Serviço de Apresentação de Mapas. Após receber o mapa no formato solicitado, o Serviço de Apresentação de Mapas desenha sobre a imagem as posições dos dispositivos móveis e retorna o mapa para o cliente. A Figura 4.8 apresenta o diagrama de seqüência para o caso de uso descrito acima.

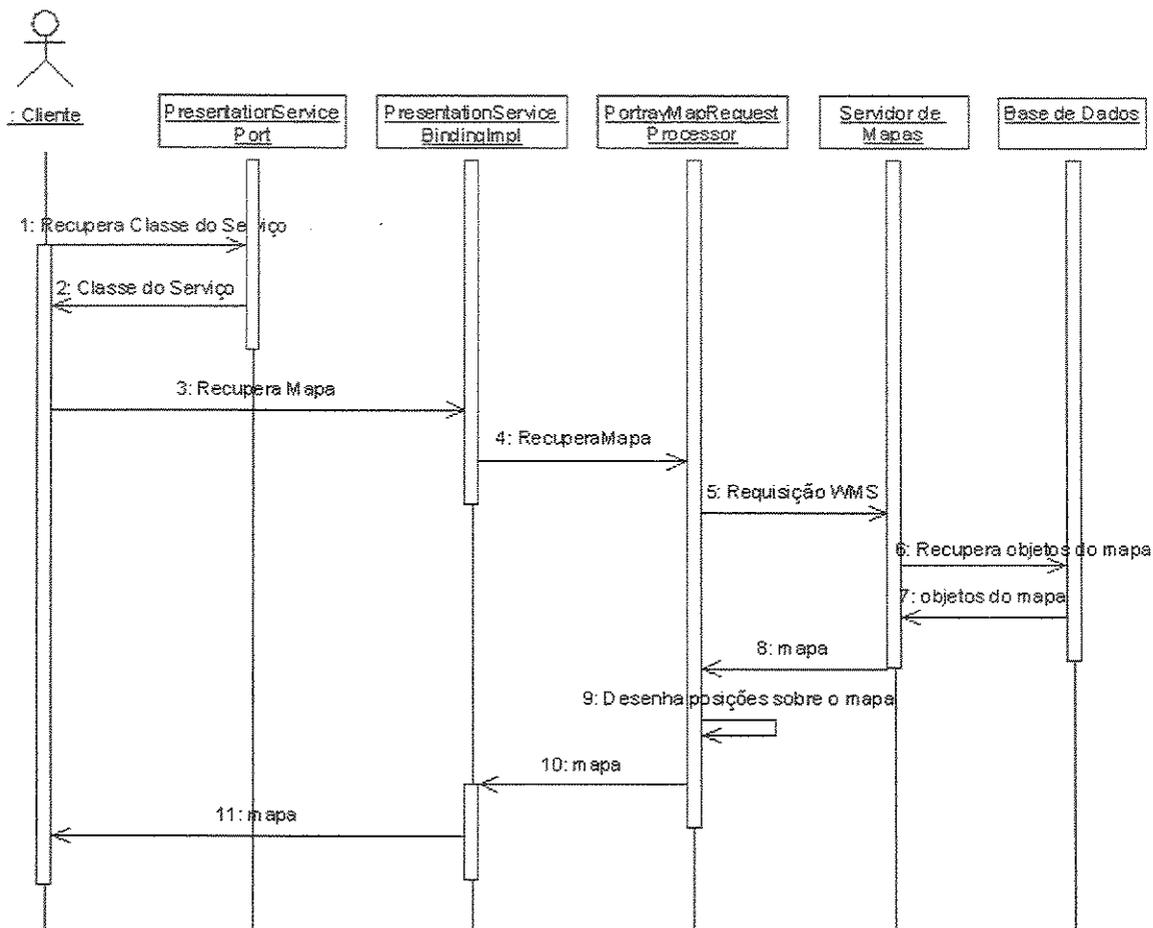


Figura 4.8: Diagrama de Sequência.

4.2.3 Publicação do Serviço de Apresentação de Mapas

A publicação de um serviço via *Web Services* requer a definição do documento WSDL que contém a descrição do serviço a ser oferecido e como acessá-lo. Nesta seção é apresentado o documento WSDL do Serviço de Apresentação de Mapas.

O elemento raiz do documento WSDL é o <definitions>. Ele contém os atributos que definem os *namespaces* utilizados no documento, além dos outros elementos que descrevem um serviço: *types*, *message*, *portType*, *binding* e *service*.

```
<wsdl:definitions
  targetNamespace='http://www.cpqd.com.br/sam/wsdl/gis/
    presentationService'
  xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:tns='http://www.cpqd.com.br/sam/wsdl/gis/presentationService'
  xmlns:xls='http://www.opengis.net/xls' >
```

O elemento <types> define os tipos de dados utilizados na interação entre consumidores e provedores de serviços. No Serviço de Apresentação de Mapas, foram utilizados os tipos definidos na especificação OpenLS:

```
<wsdl:types>
  <xsd:schema elementFormDefault='qualified'>
    <xsd:import namespace='http://www.opengis.net/xls' schemaLocation
      ='PresentationService.xsd'/>
  </xsd:schema>
</wsdl:types>
```

O elemento <message> define os tipos de dados de uma operação. Cada mensagem contém um ou mais <parts>, que podem ser comparados aos parâmetros de uma função em uma linguagem de programação tradicional [W3S99a]. O Serviço de Apresentação de Mapas possui duas mensagens, uma para a requisição do serviço e outra para a resposta.

```
<wsdl:message name='PortrayMapRequest'>
  <wsdl:part name='PortrayMapRequest' type='xls:PortrayMapRequestType'
  />
</wsdl:message>

<wsdl:message name='PortrayMapResponse'>
```

```

    <wsdl:part name='PortrayMapResponse' type='xls:PortrayMapResponseType'
    </wsdl:message>
  
```

O elemento considerado mais importante de um documento WSDL é o `<portType>`. Ele define o serviço, as operações que podem ser executadas e as mensagens envolvidas. O Serviço de Apresentação de Mapas oferece apenas a operação `getPresentation`. Esta operação retorna no elemento `PortrayMapResponse` um mapa gerado de acordo com as informações fornecidas no `PortrayMapRequest`.

```

<wsdl:portType name='PresentationServicePort'>
  <wsdl:operation name='getPresentation'>
    <wsdl:input message='tns:PortrayMapRequest' />
    <wsdl:output message='tns:PortrayMapResponse' />
  </wsdl:operation>
</wsdl:portType>
  
```

O elemento `<binding>` define o formato da mensagem e detalhes do protocolo para cada `<portType>`. Este elemento possui os atributos `name` e `type`. O primeiro atributo define o nome da ligação e o segundo o `<portType>` para a qual a ligação está sendo definida, neste caso `PresentationServicePort`.

```

<wsdl:binding name='PresentationServiceBinding' type='
  tns:PresentationServicePort'>
  <soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/
  http' />
  <wsdl:operation name='getPresentation'>
    <soap:operation soapAction='PresentationService\#getPresentation' />
    <wsdl:input>
      <soap:body encodingStyle='http://schemas.xmlsoap.org/soap/
      encoding/' namespace='http://www.cpqd.com.br/sam/wsd/gis/
      presentationService' use='encoded' />
    </wsdl:input>
    <wsdl:output>
      <soap:body encodingStyle='http://schemas.xmlsoap.org/soap/
      encoding/' namespace='http://www.cpqd.com.br/sam/wsd/gis/
      presentationService' use='encoded' />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
  
```

O elemento `<soap:binding>` contido no trecho acima também possui dois atributos: *style* e *transport*. O atributo *style* controla a organização da chamada da operação dentro do elemento *Body* da mensagem SOAP, podendo assumir os valores “*rpc*” ou “*document*”. No caso da opção “*rpc*”, o corpo da mensagem deve seguir as regras estabelecidas na especificação SOAP, a qual define que o elemento *Body* deve conter apenas um elemento após o nome da operação, e todos os parâmetros devem ser representados como sub-elementos deste elemento principal. Já na opção “*document*”, o corpo da mensagem SOAP é definido pelo XML *Schema* [W3C00] contido no elemento `<types>` do documento WSDL. No Serviço de Apresentação de Mapas foi utilizada a opção “*rpc*”.

O atributo *transport* define o protocolo SOAP a ser utilizado. Neste caso é utilizado o protocolo HTTP.

O elemento `<operation>` define cada uma das operações oferecidas por um `<portType>`. Para cada operação é definida uma ação SOAP. Além disso, o atributo *use* define como os parâmetros de entrada e saída são codificados na mensagem, podendo assumir um dos valores “*encoded*” ou “*literal*”. O valor de *use* igual a “*literal*” significa que as definições de tipos seguem um XML *Schema*. Na opção “*encoded*” os dados são representados em XML de acordo com a especificação SOAP. Neste caso, as regras para se codificar e interpretar o corpo de uma mensagem SOAP são definidas em uma URL especificada no atributo *encodingStyle*. Esta foi a opção adotada no Serviço de Apresentação de Mapas.

O último elemento do documento WSDL do Serviço de Apresentação de Mapas, `<service>`, contém a localização do serviço na rede.

```

<wsdl:service name='PresentationService'>
  <wsdl:port name='PresentationServicePort'
    binding='tns:PresentationServiceBinding'>
    <soap:address location='http://localhost:8080/axis/servlet/
      AxisServlet' />
  </wsdl:port>
</wsdl:service>

```

4.2.4 Implementação

Após a definição do documento WSDL do Serviço de Apresentação de Mapas, foi utilizada a ferramenta *WSDL2Java* do Axis para se gerar as classes dos pacotes *PresentationWS* e *OpenGIS* apresentados na seção 4.2.2. Segue a sintaxe do comando *WSDL2Java* utilizado:

```

java org.apache.axis.wsdl.WSDL2Java --output src --server-side --skeletonDeploy true
PresentationWS.wsdl

```

Tendo como base o documento `PresentationWS.wsdl`, este comando gera os arquivos dentro do diretório `src` especificado na opção “`-output`” São geradas duas estruturas de diretórios, uma correspondente ao *namespace* definido no atributo *targetNamespace* do documento WSDL e outra correspondente ao *namespace* definido no elemento `<types>`. A primeira estrutura contém as classes do pacote *PresentationWS* e a segunda, as classes do pacote *OpenGIS*.

A opção “`-server-side -skeletonDeploy true`” determina que também seja gerada a classe *skeleton* correspondente ao lado servidor do *Web Service*. Caso contrário, será gerada apenas a classe *stub* do lado do cliente, o *ServiceLocator* e a classe *template* de implementação do Serviço.

Além da geração das classes Java, a ferramenta *WSDL2Java* gera os arquivos *deploy.wsdd* e *undeploy.wsdd* para cada serviço definido no documento WSDL. Estes arquivos são utilizados para publicação do serviço no servidor Axis. O apêndice C possui o conteúdo do arquivo *depploy.wsdd* gerado.

O comando a seguir pode ser utilizado para publicação do serviço. Ele copia o conteúdo do arquivo *deploy.wsdd* para um local específico no arquivo *server-config.wsdd* localizado no servidor Axis.

```
java org.apache.axis.client.AdminClient <deploy.wsdd/undeploy.wsdd>
```

Vale destacar que o processo de geração das classes do pacote *OpenGIS* a partir do XML *Schema PresentationService.xsd*, apresentou problema em uma das classes geradas. Na classe *net.opengis.www.gml.DoubleList* não foi gerado um construtor que receba um *string* como parâmetro, no entanto este construtor é invocado pela classe *net.opengis.www.gml.DirectPositionType*.

No esquema do OpenLS, o elemento *doubleList* está definido da seguinte forma:

```
<simpleType name='doubleList'>
  <list item Type='double' />
</simpleType>
```

O *WSDL2Java* não possui suporte para o elemento primitivo *list* do XSD. De acordo com W3Schools [W3S99b] este elemento representa uma lista de um determinado tipo (*string*, *integer*, *double*, etc) na forma de uma *string* única, sendo que o separador dos itens da lista é um espaço em branco.

A solução encontrada foi a alteração do esquema do *doubleList* de um *list* de *double* para uma *string* simples.

```

<simpleType name='doubleList'>
  <restriction base='string' />
</simpleType>

```

A função exercida pelo elemento primitivo *list* fica subentendida, os clientes do Serviço de Apresentação de Mapas que processarem um *doubleList* devem tratar seu valor como uma *string* de *double* separados por espaços em branco. Após esta alteração, as classes foram geradas novamente e a compilação destas foi bem sucedida.

4.2.5 Execução do Protótipo

A execução do protótipo do projeto SAM é detalhado nesta seção. A partir de um *Browser*, o usuário acessa a página do projeto, na qual ele deve informar o “Usuário” e “Senha” para fins de autenticação (Figura 4.9).

Na página seguinte (Figura 4.10) o usuário pode escolher o agente que deseja visualizar no mapa. Ao se confirmar a operação, a aplicação principal acessa o Serviço de *Gateway* para recuperar a posição do agente e em seguida aciona o Serviço de Apresentação de Mapas a fim de recuperar o mapa com a localização do dispositivo.

O seguinte trecho de código ilustra como a aplicação cliente acessa o Serviço de Apresentação de Mapas via *Web Services*:

```

/*
  Cliente constrói o objeto PortrayMapRequest com os parâmetros da requisição.
  A classe FactoryOpenLSPresentation facilita a criação das classes OpenLS que
  compõem a requisição.
  Largura = 208, altura = 250, formato = image/png
  Bounding box = (-47.0574124449709,-22.883660046013777),(-47.05448761775622,
  -22.880949971677772)
*/
OutputType[] output = new OutputType[1];
output[0] = FactoryOpenLSPresentation.CreateOutput(208, 250, ‘image/png’,
-47.0574124449709,-22.883660046013777, -47.05448761775622 ,
-22.880949971677772);

/* Define a posição dispositivo a ser desenhado sobre o mapa */
OverlayType[] overlay = new OverlayType[1];
overlay[1] = FactoryOpenLSPresentation.CreateOverlay(‘-47.05595001684953
-22.8823050160925’);

/* Lista de layers a serem carregados no mapa */

```

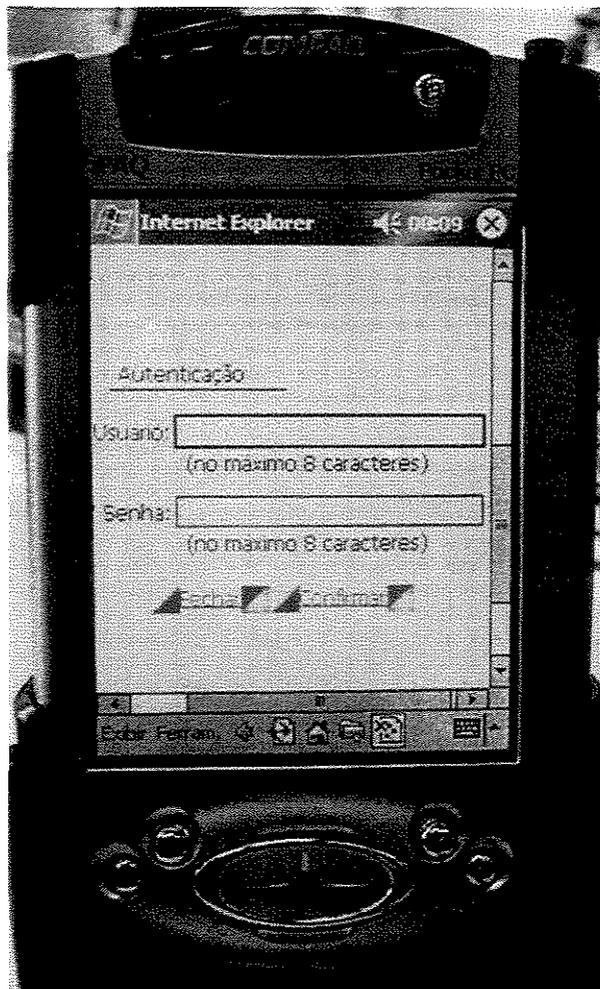


Figura 4.9: Página Inicial do Projeto SAM.

```

LayerType basemap = FactoryOpenLSPresentation.CreateBasemap('municipios
    alinhamento trecho\_logradouro estados\_contorno');

/* Recupera a classe de implementação do Serviço de Apresentação de Mapas */
URL urlDoServico = new URL('http://localhost:8080/axis/servlet/AxisServlet');
PresentationService service = new PresentationServiceLocator();
PresentationServicePort port = service.getPresentationServicePort(urlDoServico)
    ;

/* Executa o Serviço de Apresentação de Mapas passando o PortrayMapRequest */
PortrayMapResponseType gRest = port.getPresentation(portrayMapReq);

```

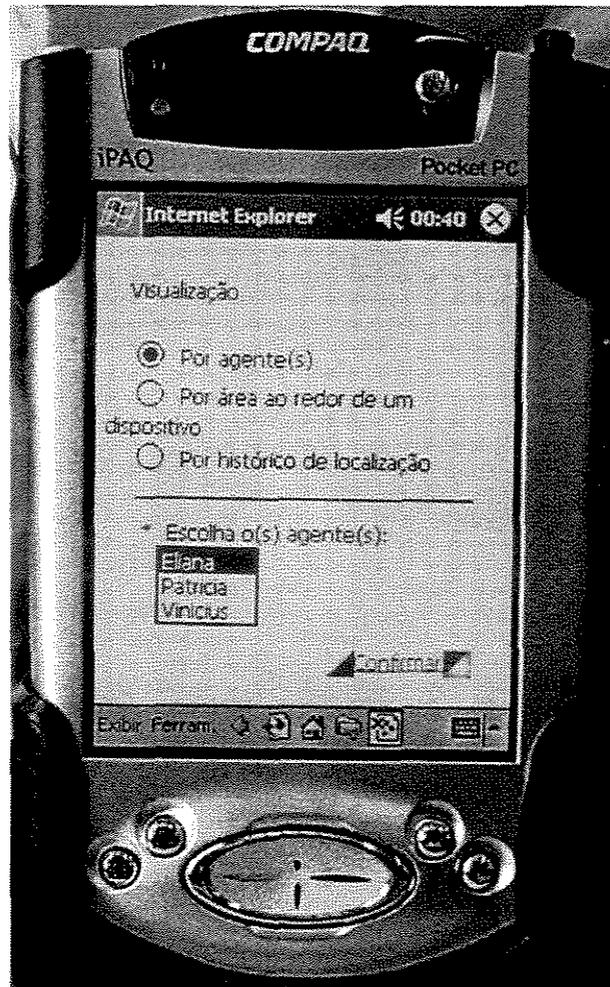


Figura 4.10: Escolha do Agente para Visualização.

A mensagem SOAP enviada para o Serviço de Apresentação de Mapas a partir da chamada do serviço acima está apresentada no apêndice D.

O mapa retornado pelo Serviço de Apresentação de Mapas é apresentado para o usuário conforme ilustrado na Figura 4.11.

4.3 Considerações Finais

As Arquiteturas Orientadas a Serviços visam resolver o problema de interoperabilidade não solucionado pelas arquiteturas distribuídas como CORBA, *Java Remote Method Invocation* (JRMI) e DCOM. Os *Web Services* são um caso específico de serviços que utilizam

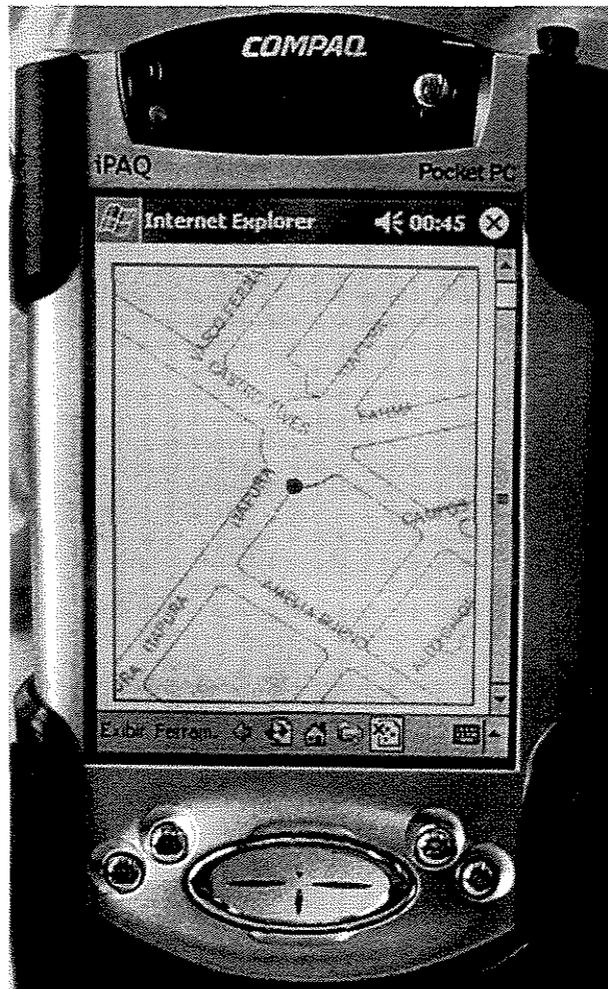


Figura 4.11: Mapa com a Localização do Agente.

a Internet para se comunicarem e interagirem através de padrões *web*.

A adoção de uma arquitetura orientada a serviços baseada em *Web Services*, combinada com o uso das interfaces abertas definidas na especificação OpenLS, são uma forma de garantir a interoperabilidade e ubiqüidade das aplicações de LBS.

Durante o desenvolvimento do protótipo do Serviço de Apresentação de Mapas para validação desta arquitetura, foi constatado que as interfaces definidas na especificação OpenLS ainda possuem alguns problemas quando acessadas via *Web Services*. Alguns tipos de dados definidos na especificação não são tratados por todas as implementações do SOAP, como é o caso do tipo *list* não contemplado pelo Axis da Apache.

Apesar dos problemas encontrados, que puderam ser resolvidos com alguma solução

de contorno, concluímos que a adoção de uma arquitetura aberta para desenvolvimento de Serviços Baseados em Localização utilizando *Web Services* é totalmente viável e promissora, devendo ser considerada na implementação de aplicações de LBS.

No caso específico do projeto SAM, a utilização da tecnologia *Web Services* e a adoção da especificação OpenLS facilitaram muito a integração dos diferentes sistemas. Uma vez definida a interface de comunicação entre eles, estes puderam ser integrados de forma simples, rápida e sem grande esforço de desenvolvimento.

Apesar da análise de requisitos não funcionais, como o desempenho, não ser escopo deste protótipo, nos testes realizados, o tempo de resposta de uma requisição do mapa de localização de um determinado agente, a partir de um computador *desktop*, foi de menos de 4 segundos, o que é bastante razoável. São necessárias análises mais específicas quando a requisição ocorre a partir de um dispositivo móvel. Neste caso, alguns aspectos inerentes à comunicação sem fio devem ser considerados, como a latência da rede e largura de banda.

Capítulo 5

Análise de Produtos do Mercado

Serviços Baseados em Localização representam uma nova fonte de receitas para empresas operadoras de rede sem fio, sendo uma importante forma de diferenciá-las das demais empresas e mantê-las competitivas.

Uma pesquisa do Frost & Sullivan [SUL03] sobre o Mercado Norte Americano de Serviços Baseados em Localização, revela que em 2006:

- Haverá aproximadamente 20 milhões de clientes de LBS, o que representa cerca de 11% de todos os clientes móveis.
- O mercado de LBS terá um faturamento de mais de \$850 milhões.
- Aproximadamente um terço de todos os clientes móveis estarão dispostos a pagar por algum tipo de Serviço Baseado em Localização.

Vislumbrando as demandas do crescente mercado sem fio, várias empresas da área de tecnologia e geoprocessamento desenvolveram aplicações de LBS.

Neste capítulo são analisados alguns produtos de mercado na área de Serviços Baseados em Localização. São apresentadas as principais características de cada produto e uma análise de cada um quanto ao uso de padrões abertos.

5.1 Autodesk[©]

O *Autodesk LocationLogic* é o componente central da solução de LBS provida pela Autodesk[©]. É uma plataforma para desenvolvimento de Serviços Baseados em Localização que provê processamento de grande volume de transações, integração com infra-estrutura de rede existente e um ambiente para desenvolvimento de novas aplicações [Aut03].

5.1.1 Funcionalidades

O *LocationLogic* possui funcionalidades para geração de mapas, definição de rotas, consultas por proximidade, alertas baseados na localização, dentre outras.

A plataforma oferece também mecanismos para personalização e gerenciamento de usuários através dos seguintes recursos:

- Gerenciamento de usuários e senhas.
- Autenticação de usuário, controle de acesso por perfil e privilégios de segurança.
- Definição de localizações favoritas.
- Definição de rotas preferenciais.
- Armazenamento de histórico de localização do usuário.

A arquitetura do *LocationLogic* consiste de um repositório de dados, interfaces para acesso a componentes externos e um conjunto de serviços baseados em localização, os quais estão descritos a seguir.

Localização de Dispositivos

O *LocationLogic* possui um subsistema para determinação da posição de dispositivos e proximidade entre dois terminais, trabalhando com tecnologias de posicionamento tanto baseadas em dispositivos como baseadas na rede. A solução possui também adaptadores genéricos que permitem a integração com sistemas de localização de terceiros.

Em outubro de 2004, a Autodesk[©] juntamente com a empresa Agilent Technologies Inc.[©] anunciou um acordo para prover a primeira solução integrada de localização de dispositivos para a indústria de telecomunicações [Dir04]. Esta solução utiliza uma tecnologia patenteada de monitoramento da rede de forma não-intrusiva, a qual elimina a necessidade de consulta à rede para determinar a localização de dispositivos. Desta forma, operadores de rede sem fio poderão utilizar o *Autodesk LocationLogic* juntamente com o sistema de localização *acceSS7* da Agilent[©] para prover Serviços Baseados em Localização para uma grande massa de usuários, sem impactar as redes já existentes.

Consultas Espaciais

Esta funcionalidade permite a recuperação de informações, tais como pontos de interesse, através de consultas espaciais relativas a uma região, rota ou localização do usuário. O sistema oferece suporte para os seguintes tipos de consultas espaciais:

- Busca de objetos que estejam a uma determinada distância ao longo de uma linha reta, podendo ser uma estrada ou avenida.
- Busca de objetos próximos a uma determinada rota.
- Busca de objetos dentro de um determinado raio.
- Busca de objetos que estejam a uma certa distância de outro objeto.
- Busca de objetos dentro de uma região.

Geocodificação

O Serviço de Geocodificação do *LocationLogic* oferece recursos para exibição de informações de endereço associadas a uma posição do mapa e consultas de objetos associados a um endereço.

Existe uma interface única para as funcionalidades de geocodificação e geocodificação reversa, sendo que estes serviços possuem suporte tanto para um endereço simples como para interseção de ruas.

A arquitetura *LocationLogic* possui um *adapter* que permite que as aplicações desenvolvidas nesta plataforma utilizem pacotes de geocodificação de terceiros.

Alertas Baseados em Localização

Este serviço permite a notificação de eventos sensíveis à localização disparados a partir da relação de proximidade com outros usuários e pontos de interesse. Também permite a visualização de informação associada à localização do usuário, como por exemplo, informação de tráfego ao longo de uma rota.

Geração de Mapas

Este serviço produz mapas contendo pontos de interesse, rotas e informações de navegação ao longo de uma rota. Os mapas também podem exibir a localização de um usuário.

As aplicações que utilizam este serviço podem definir o formato de saída do mapa, tamanho e resolução. Elas possuem ainda acesso a controles de navegação no mapa, tais como *zoom* e *pan*. Os formatos oferecidos pelo *LocationLogic* são BMP, WBMP e PNG.

Definição de Rotas e Navegação

Esta funcionalidade gera rotas otimizadas entre dois pontos e informações de navegação ao longo da rota. Podem ser definidos parâmetros de otimização da rota a fim de minimizar o tempo ou a distância do percurso, levando-se em consideração o modo de transporte. Os

seguintes critérios de custo podem ser considerados na definição da rota: menor distância, menor tempo, evitar túneis ou pontes, evitar tráfego. Os modos de transporte possíveis são: carro, à pé ou bicicleta.

Rotas também podem ser calculadas com base em um determinado horário, levando-se em consideração condições como restrições de conversão, fechamento de ruas e informações de tráfego.

Gerenciamento de Conteúdo

O *LocationLogic* possui uma base de dados integrada com informações de pontos de interesse, redes de ruas e estradas e mapas urbanos. O esquema da base de dados também possui suporte para conteúdo dinâmico, como informações atualizadas de tráfego, notícias e previsão do tempo.

A Autodesk[®] possui parceria com empresas que fornecem mapas digitais para soluções de LBS. A empresa NavTech[®] permite que provedores de serviço licenciem os seus mapas diretamente com a Autodesk[®] para integração na solução *LocationLogic*.

Os mapas digitais da NavTech[®], além de possuírem a geometria das estradas e ruas, possuem informações associadas a cada quilômetro da rede, como restrições de velocidade e configurações da pista. Os mapas também possuem pontos de interesse, possibilitando o envio de instruções de navegação passo a passo em uma rota entre dois pontos [NAV04].

5.1.2 Uso de Padrões

A solução *Autodesk LocationLogic* é desenvolvida em uma arquitetura J2EE, utilizando tecnologias baseadas em padrões abertos como Java e XML.

A Autodesk[®] é um dos membros do OGC e foi co-autora da especificação *Open Location Services*. Ela também é membro associado do OMA e tem participação ativa no fórum WAP.

A participação da Autodesk[®] em organizações que tem como objetivo a definição de padrões abertos é um bom indicativo de que suas soluções estejam de acordo com estes padrões.

5.2 LocatioNet[®]

Os principais produtos da empresa LocatioNet[®] são o LBS *Middleware*, o GIS *Engine* e um conjunto de aplicações de LBS [Loc04]. O *middleware* integra conteúdo, aplicações, tecnologias de posicionamento e tecnologias GIS. Ele inclui uma API aberta e ferramentas de desenvolvimento que oferecem flexibilidade na utilização de recursos da plataforma, permitindo o desenvolvimento de aplicações customizadas.

O GIS *Engine* provê recursos para geração de mapas nos formatos *raster* e vetoriais, possui funcionalidades de geocodificação e geocodificação reversa e definição de rotas.

5.2.1 Funcionalidades

As aplicações de LBS do *LocatioNet*® são voltadas tanto para o mercado consumidor como para o mercado corporativo, oferecendo soluções na área de gerenciamento de frotas e força de trabalho, visualização de mapas, definição de rotas, busca por pontos de interesse e eventos. Estas funcionalidades estão descritas a seguir.

Gerenciamento de Frota e Força de Trabalho

Esta aplicação permite o controle da força de trabalho móvel através da atribuição de tarefas, envio de ordens de trabalho e comunicação direta com o pessoal em campo através do aparelho celular. Trabalhadores podem receber as instruções de trabalho e atualizações via SMS, podem visualizar sua posição corrente em um mapa, localizar pontos de interesse nas proximidades e enviar relatórios de andamento do trabalho, controlando sua privacidade através de WAP. A aplicação também permite a definição de rotas, fornecendo instruções do percurso entre a posição do usuário e um determinado ponto de interesse.

Visualização de Mapas

A aplicação *MyMap* permite que usuários visualizem sua posição em um mapa, procurem e calculem a melhor rota até pontos de interesse e se comuniquem com amigos enviando mapas detalhados e instruções de navegação. *MyMap* possui suporte para *Java 2 Micro Edition (J2ME)*, *Binary Runtime Environment for Wireless (BREW)* e WAP.

Um mecanismo de busca permite a localização de endereços, lugares ou qualquer outro ponto de interesse através do nome, endereço ou um ponto de referência, provendo um mapa com a localização do lugar desejado e instruções de navegação até o local.

Rastreamento de Crianças

Pais ou responsáveis podem utilizar um navegador Web para visualizar a localização das crianças no mapa e definir áreas de controle e alertas. O desvio de uma área segura ou a entrada em áreas de risco disparam o envio de alertas na forma de texto SMS ou *Multimedia Messaging Service (MMS)* diretamente para o dispositivo móvel.

A aplicação também permite a configuração de alertas periódicos na forma de mensagens SMS ou MMS enviadas pelo dispositivo, permitindo a atualização da localização da criança periodicamente.

Comércio e Entretenimento

Esta aplicação permite que usuários procurem eventos de entretenimento ou promoções nas suas proximidades e também que comerciantes enviem propaganda de seus produtos para usuários na vizinhança, de acordo com preferências dos usuários.

InfoCommerce também oferece mapas de navegação a fim de facilitar o percurso do usuário até a loja desejada ou o local do evento procurado. O usuário pode personalizar a aplicação definindo suas categorias e destinos favoritos.

Pontos de Interesse e Eventos

POI Finder é uma aplicação clássica de páginas amarelas, facilitando a busca por pontos de interesse nas proximidades tais como caixas eletrônicos, hotéis, estacionamentos, restaurantes, teatros e outros. O *Event Finder* funciona de maneira análoga, mas para localização de eventos como festas, concertos e *shows* que ocorram em um determinado local, data e horário. Juntamente com a informação do ponto de interesse ou evento localizado, são fornecidas instruções de navegação da posição corrente do usuário até o local de interesse.

Os serviços são sempre baseados em uma localização específica, a partir da qual a busca é realizada. A localização pode ser a posição corrente do usuário, recebida da plataforma *LocatioNet*, localização fornecida manualmente pelo usuário, localização de pontos de referência definidos previamente pelo usuário ou um ponto de interesse localizado em uma busca anterior.

5.2.2 Uso de Padrões

LocatioNet® provê interoperabilidade entre as soluções de LBS através do uso de padrões abertos. Para atingir este objetivo, *LocatioNet*® é membro de vários fóruns da indústria e organizações internacionais para definição de padrões, incluindo *Israel Mobile Association* (IMA), OGC, OMA e *Wireless Location Industry Association* (WLIA).

Apesar de não haver na documentação uma referência explícita sobre os padrões utilizados, acredita-se que a participação da empresa em órgãos como OGC e OMA indicam um comprometimento no uso de padrões abertos.

5.3 ESRI®

A solução para dispositivos móveis da ESRI®, o *ArcPad* [ESR04a], envolve a realização de tarefas mais simples e que não exigem muitas ferramentas de processamento geográfico. Além da aplicação *ArcPad*, a ESRI® possui a solução *ArcWeb Services* [ESR04b] que

contém um conjunto de serviços que podem ser acessados via *Web Services* e utilizados para construção de aplicações customizadas.

A solução *ArcWeb Services* é baseada no padrão SOAP, o que significa que ela pode ser integrada a qualquer aplicação Web ou *desktop*. Na solução *ArcWeb Services*, o armazenamento, manutenção e atualização das bases de dados de conteúdo são feitos pela própria ESRI®, estando disponíveis no esquema 24/7.

5.3.1 Funcionalidades

A solução *ArcPad* para dispositivos móveis permite a visualização de mapas vetoriais e *raster*, criação e edição de dados geográficos, navegação nos mapas através de operações de *zoom*, *pan* e centralização na posição corrente do usuário obtida através de GPS, realização de consultas para localização e identificação de objetos, medidas de distâncias e áreas e navegação através de GPS.

A ESRI® também possui um conjunto de aplicações que utilizam a plataforma *ArcWeb Services*. Estas aplicações estão descritas a seguir.

Visualização de Mapas

Esta aplicação permite a visualização de uma grande variedade de mapas de várias regiões do mundo, desde mapas tradicionais de ruas até imagens de satélite, além de permitir a customização do estilo de exibição. A aplicação pode ser modificada para exibir apenas alguns tipos de mapas para uma determinada área de interesse.

A aplicação também possui serviço de geocodificação para identificação e visualização da localização de endereços, lugares ou pontos de referência.

Definição de Rotas

A aplicação permite a geração de mapas com rotas e instruções de navegação entre duas ou mais localizações nos Estados Unidos, Canadá e Europa. Nos Estados Unidos, também é possível visualizar informações de meteorologia e tráfego em tempo real ao longo da rota escolhida.

Os mapas podem exibir ao longo da rota pontos de interesse definidos pelo usuário e apresentar as instruções de navegação em diferentes línguas.

Gerenciamento de Frotas

A aplicação de gerenciamento de frotas permite que usuários visualizem a posição corrente de veículos ao longo da rota definida. Alertas são emitidos quando os veículos saem dos limites geográficos aos quais foram associados.

O propósito dos limites é criar uma cerca eletrônica ao redor de objetos em movimento para fins de monitoramento. Os objetos podem ser veículos de entrega, equipamentos de construção, pessoal em campo, veículos de patrulhamento ou qualquer outro objeto que possa se movimentar. Os limites podem ser definidos na forma de círculos simples ao redor de um endereço ou áreas complexas como regiões ao longo de uma rota ou áreas administrativas.

Pontos de Interesse

A aplicação *Real Estate* permite que usuários procurem casas para compra ou locação que atendam a determinados critérios de busca. A busca pode ser feita especificando-se áreas de interesse, como código postal ou nomes de cidades e características da casa, como preço, número de quartos, banheiros, etc. Usuários podem visualizar informações detalhadas sobre a casa, incluindo mapas da sua localização. Também é possível gerar relatórios da vizinhança, com dados como densidade demográfica, pontos de interesse, riscos de enchentes, assim como visualizar imagens aéreas do local.

Um outro módulo da aplicação permite que usuários autorizados publiquem suas casas para consultas por outros usuários.

Esta aplicação foi customizada para busca de casas, porém ela pode ser alterada para atender consultas por eventos e locais de entretenimento, lojas comerciais, pontos turísticos ou qualquer outro ponto de interesse.

5.3.2 Uso de Padrões

A ESRI[®] atuou como um dos principais editores da especificação *Open Location Services* do OGC. A solução *ArcWeb Services* possui suporte para esta especificação, além de suporte ao padrão SOAP.

A ESRI[®] também vem trabalhando no suporte às especificações WMS, WFS e GML do OGC, contribuindo para a implementação de soluções GIS baseadas em padrões abertos que garantam a interoperabilidade entre as aplicações.

5.4 Herecast

Herecast é uma plataforma de código aberto para desenvolvimento de Serviços Baseados em Localização para dispositivos com suporte Wi-Fi [Her04]. O código fonte está disponível no repositório do *SourceForge* sob a licença *GNU General Public License (GPL)*.

Um serviço bem simples desta solução é a identificação da localização do usuário. Serviços mais avançados podem utilizar a informação de localização para prover mapas

com a posição corrente do usuário, publicar a localização para amigos, deixar mensagens para pessoas que passarem pelo mesmo local, conversar com pessoas que estejam na mesma área e brincar com jogos baseados em localização.

A idéia básica do *Herecast* é utilizar os pontos de acesso da rede Wi-Fi como pontos de referência para identificar uma determinada localização. Informações como endereço, nomes de prédios e números de salas podem ser associados a cada ponto de acesso da rede. Por exemplo, o ponto de acesso “A” pode ser reconhecido como a Sala 110, do Instituto de Computação da Unicamp, em Campinas.

Qualquer pessoa pode incluir na base de dados do *Herecast* informações associadas a qualquer ponto de acesso do mundo, tornando esta informação disponível para todas as outras pessoas.

Um ponto forte da infraestrutura *Herecast* é que todos os cálculos de localização são feitos no próprio dispositivo. Isto permite um total controle do usuário sobre as informações que são divulgadas para as outras pessoas, preservando assim a sua privacidade.

5.4.1 Funcionalidades

A seguir estão descritos alguns exemplos de aplicações desenvolvidas na plataforma *Herecast*.

Localização

O serviço básico do *Herecast* é a localização do usuário. A posição corrente do usuário é exibida continuamente no menu principal da aplicação. Informações mais detalhadas da localização podem ser obtidas com base nos dados referentes ao ponto de acesso ao qual o terminal móvel está associado.

Quadro de Mensagens

Este serviço permite que o usuário envie mensagens que podem ser lidas por pessoas na mesma sala ou prédio.

O usuário também pode configurar o sistema para receber alertas quando entrar em um prédio que tenha novas mensagens.

Visualização de Mapas

Os mapas visualizados no *Herecast* são imagens *raster* às quais são associados pontos de referência. Os mapas podem ser criados com o nível de detalhamento desejado e enviados para o servidor de forma bem simples.

Busca por Amigos

Quando o usuário se inscreve no serviço de busca por amigos, seu dispositivo automaticamente publica sua localização em uma página Web à medida que ele se movimenta de um lugar para o outro.

Busca por Pontos de Acesso Wi-Fi

Este serviço exibe a lista de todos os pontos de acesso detectados na área, juntamente com a sua localização e intensidade do sinal. Também podem ser visualizadas informações adicionais como os canais utilizados por cada ponto de acesso.

Caso seja detectado algum ponto de acesso sem informação associada, o usuário pode associar informações de localização como Cidade, Província e Área. A Área pode ser utilizada para identificar uma subdivisão dentro da cidade, podendo ser uma universidade, bairro, distrito, prédio, etc.

5.4.2 Uso de Padrões

A documentação do *Herecast* disponível não faz referência ao uso de padrões abertos para acesso aos Serviços Baseados em Localização. Foi identificado somente o uso de XML para acesso à base de dados *Herecast*. Porém, vale ressaltar a importância deste projeto pelo fato de possuir código aberto.

5.5 Tabela de Comparação dos Produtos

A Tabela 5.1 apresenta um resumo da comparação dos produtos de mercado descritos neste capítulo. Além destes produtos, a tabela inclui o projeto SAM, do qual faz parte o Serviço de Apresentação de Mapas apresentado no Capítulo 4.

	Autodesk	LocatioNet	ESRI ¹	ESRI ²	Herecast	SAM
Busca por Pontos de Interesse	Sim	Sim	Não	Sim	Não	Não
Localização de Dispositivos	Sim	Sim	Sim	Sim	Sim	Sim
Geocodificação	Sim	Sim	Não	Sim	Não	Sim
Apresentação de Mapas	Sim	Sim	Sim	Sim	Sim	Sim
Rotas	Sim	Sim	Não	Sim	Não	Não
Utilização de Padrões Abertos	Sim	Sim	Sim	Sim	Não	Sim

Tabela 5.1: Tabela de Comparação dos Produtos.

5.6 Considerações Finais

Neste capítulo foram analisadas aplicações de LBS de três grandes empresas da área de geoprocessamento, Autodesk[®], LocatioNet[®] e ESRI[®] e um projeto de código aberto, Herecast[®].

Nesta análise pode-se observar a importância dada ao uso de padrões abertos. Praticamente todas as empresas analisadas fazem parte de alguma organização para definição

¹Solução *ArcPad*

²Solução *ArcWeb Services*

de padrões e todas se preocupam em desenvolver aplicações que tenham um alto grau de interoperabilidade.

Vale destacar a importância do projeto Herecast[©] que, apesar de não indicar o uso de padrões abertos na sua solução, é um projeto de código aberto, indicando que esta pode ser uma possível tendência na área de Serviços Baseados em Localização.

Capítulo 6

Conclusão

6.1 Considerações Finais

Esta dissertação apresentou um estudo sobre um importante tipo de aplicação que surge com a evolução das redes de comunicação sem fio e o crescente uso de dispositivos móveis como celulares, PDAs e *notebooks*, os Serviços Baseados em Localização. Estas aplicações representam uma evolução em relação às aplicações GIS tradicionais, pois utilizam, além de informações geográficas, dados de localização para prover informações mais precisas e úteis para o usuário.

Os Serviços Baseados em Localização têm como principais características a ubiquidade e interoperabilidade, uma vez que devem estar disponíveis em qualquer dispositivo, em qualquer lugar, a qualquer hora. Além disso, devem executar em diferentes plataformas e linguagens de programação e, freqüentemente, devem se comunicar com sistemas legados.

O uso de padrões abertos é uma forma de garantir a interoperabilidade entre as aplicações. Pensando nisso, este trabalho apresentou um estudo sobre a especificação *Open Location Services* do OGC, a qual define um conjunto de interfaces para o desenvolvimento de Serviços Baseados em Localização.

A tecnologia *Web Services* também vem sendo amplamente difundida como uma solução revolucionária para os problemas de integração entre os sistemas de computação. A arquitetura proposta neste trabalho combina a utilização da tecnologia *Web Services* com o uso de padrões abertos para o desenvolvimento de Serviços Baseados em Localização.

A fim de validar a arquitetura proposta, foi desenvolvido um protótipo do Serviço de Apresentação de Mapas definido na especificação OpenLS. A combinação da tecnologia *Web Services* com a utilização de padrões abertos foi um grande desafio neste trabalho, uma vez que a especificação OpenLS 1.0 ainda não está totalmente preparada para esta tecnologia. Mas, apesar dos problemas encontrados, que puderam ser resolvidos com al-

guma solução de contorno, podemos concluir que o desenvolvimento de Serviços Baseados em Localização utilizando a tecnologia *Web Services* é totalmente viável.

Esta dissertação também apresentou um estudo sobre outras tecnologias envolvidas no desenvolvimento de aplicações de LBS, como os bancos de dados espaciais e tecnologias de posicionamento, como Cell-ID, GPS e Wi-Fi. A escolha da melhor tecnologia a ser adotada depende muito do tipo de aplicação a ser desenvolvida. Devem ser consideradas questões como volume de dados, complexidade das operações a serem executadas, número de acessos, além de aspectos como precisão da localização e tempo de resposta, para escolha da tecnologia de posicionamento.

6.2 Trabalhos Futuros

Esta dissertação se limitou ao desenvolvimento do Serviço de Apresentação de Mapas que integra a especificação OpenLS. Este trabalho deixa possibilidades para a construção de um *Framework* que contemple os demais serviços especificados. O *Framework* deve ser flexível o suficiente para oferecer suporte a diferentes tecnologias de posicionamento, servidores de mapas e bancos de dados espaciais.

Um segundo aspecto a ser explorado é a utilização de outras implementações do SOAP na comunicação via *Web Services*, já que este trabalho utilizou apenas o Axis. Neste ponto, também podem ser avaliadas outras configurações do SOAP quanto ao envio das mensagens. No protótipo desenvolvido, foi utilizada a opção *RPC/Encoding* para formatação da mensagem, porém o tipo *Document/Literal*, em teoria, permite um menor acoplamento entre as aplicações e um maior grau de interoperabilidade.

Bibliografia

- [AB95a] MySQL AB. MaxDB by MySQL, 1995.
- [AB95b] MySQL AB. MySQL Documentation, 1995.
- [Amo04] S. Amorim. A Tecnologia Web Services e sua Aplicação num Sistema de Gerência de Telecomunicações. Dissertação de Mestrado, Instituto de Computação, Universidade Estadual de Campinas, 2004.
- [ANS92] American National Standards Institute ANSI. Database Language SQL - Entry Level, X3.135-1992, 1992.
- [ANS99] American National Standards Institute ANSI. Database Language SQL - Core, X3.135-1999, 1999.
- [Apa99] Apache Software Foundation. Apache Tomcat, 1999.
- [Apa00] Apache Software Foundation. Apache AXIS, 2000.
- [Aut03] Autodesk. Autodesk Location Services: Solution Overview, 2003.
- [Bie04] C. Biever. Wi-Fi finds the way when GPS can't. *New Scientist*, 28 June 2004.
- [Bos03] A. Bosworth. Interview: A Conversation with Adam Bosworth. *ACM Queue*, 1(1), Mar. 2003.
- [CDK⁺02] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, e S. Weerawarana. Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, 6(2):86–93, Mar./Apr. 2002.
- [Del03] Delphi Group. The Value of Standards, June 2003.
- [Dir04] Directions Magazine. Autodesk and Agilent Technologies to Enable Telecommunication Industry's Most Scalable Mobile Location Services Solution, Oct. 2004.

- [ESR04a] ESRI. ArcPad - Mobile GIS Software for Field Mapping Applications, 2004.
- [ESR04b] ESRI. ArcWeb Services - On-Demand GIS Data and Capabilities Via the Web, 2004.
- [ETS96] European Telecommunication Standards Institute ETSI. GTS GSM 01.02: General description of a GSM Public Land Mobile Network (PLMN). Relatório Técnico, European Telecommunication Standards Institute, May 1996.
- [FCC04] Federal Communications Commission FCC. 911 Services, 2004.
- [FT04] N. Faggion e A. Trocheris. Location-Based Services Strengthen the Strategic Position of Mobile Operators. Relatório Técnico, Alcatel Telecommunications, Jan. 2004.
- [Her04] Herecast. Herecast - An Open Infrastructure for Location-Based Services using Wi-Fi, 2004.
- [Hug01] Hughes Technologies Pty Ltd. MiniSQL, 2001.
- [IBM01] IBM. Web Services Conceptual Architecture (WSCA 1.0). Relatório Técnico, IBM, May 2001.
- [Kol04] K. Kolodziej. Indoor Location Technology Opens New Worlds. *GeoWorld Magazine*, June 2004.
- [KR03] S. Kleijnen e S. Raju. An Open Web Services Architecture. *ACM Queue*, 1(1), Mar. 2003.
- [Lea04] N. Leavitt. Are Web Services Finally Ready to Deleiver? *Computer Magazine - IEEE Computer Society*, 37(11):14-18, Nov. 2004.
- [Loc04] LocatioNet Systems Ltd. LBS Applications, 2004.
- [MATH03] J. Martin, A. Arsanjani, P. Tarr, e B. Hailpern. Web Services: Promises and Compromises. *ACM Queue*, 1(1):48-58, Mar. 2003.
- [MDB03] Z. Maamar, A. Dogac, e W. Binder. Guest editor's note: special issue on mobile commerce. *ACM SIGecom Exchanges*, 3(4):1, Winter 2003.
- [Mic96] Microsoft Corporation. DCOM Technical Overview, Nov. 1996.
- [NAV04] NAVTEQ. Navteq, 2004.
- [NEN] National Emergency Number Association NENA. Wireless 9-1-1 Overview.

- [OGC94] Open Geospatial Consortium OGC. Vision & Mission, 1994.
- [OGC99] Open Geospatial Consortium OGC. Simple Features Specification For SQL. Versão 1.1.0, May 1999.
- [OGC02] Open Geospatial Consortium OGC. Web Feature Service Implementation Specification. Versão 1.0.0, Sep. 2002.
- [OGC03a] Open Geospatial Consortium OGC. Geography Markup Language (GML) Implementation Specification. Versão 3.0.0, Jan. 2003.
- [OGC03b] Open Geospatial Consortium OGC. Web Coverage Service (WCS). Versão 1.0.0, Aug. 2003.
- [OGC04a] Open Geospatial Consortium OGC. OpenGIS Location Services (OpenLS): Core Services [Parts 1-5]. Versão 1.0.0, Jan. 2004.
- [OGC04b] Open Geospatial Consortium OGC. Web Map Service. Versão 1.3.0, Aug. 2004.
- [OMA02] Open Mobile Alliance OMA. TS 101: Mobile Location Protocol. Version 3.0.0, Jun. 2002.
- [OMG04] Object Management Group OMG. Common Object Request Broker Architecture: Core Specification. Version 3.0.3, Mar. 2004.
- [Ope04] Openwave Systems Inc. Location Studio 2.1: MLP 3.0.0 Developer's Guide, 2004.
- [Ora03] Oracle Corporation. Oracle Spatial and Oracle Locator: An Oracle Technical White Paper. Relatório Técnico, Oracle Corporation, Dec. 2003.
- [Pai04] F. Paiva. Onde está você? *Revista Teletime*, 7(67), June 2004.
- [Pap03] M. P. Papazoglou. Service-Oriented Computing: Concepts, characteristics and Directions. Em *Fourth International Conference on Web Information Systems Engineering (WISE'03)*, páginas 3–12, Roma, Italy, Dec. 2003. IEEE.
- [Pos04] PostgreSQL Global Development Group. PostgreSQL Documentation, 2004.
- [Ref04] Refrations Research Inc. PostGIS Manual, 2004.
- [Sco97] J. Scourias. Overview of the Global System for Mobile Communications. Relatório Técnico, University of Waterloo, Ontario, Canada, Oct. 1997.

- [Sem97] J. G. Sempere. An overview of the GSM system. Relatório Técnico, University of Strathclyde, Scotland, 1997.
- [Sna03] Snaptrack. Location Technologies for GSM, GPRS and UMTS Networks. Relatório Técnico, SnapTrack, Campbell, 2003.
- [Spi03] J. Spinney. A Brief History of LBS and How OpenLS Fits Into the New Value Chain. *Directions Magazine*, July 2003.
- [SSY+04] G. C. Silva, I. L. Silva, J. S. Yamamoto, P. M. Pereira, e V. J. Latorre. Serviços e Aplicações Móveis: Definição de Arquitetura e Tecnologias. Relatório Técnico, Fundação CPqD, Campinas, SP, 2004.
- [SUL03] FROST & SULLIVAN. North American Location-based Service Markets, Mar. 2003.
- [Sun94a] Sun Microsystems. Java servlet technology, 1994.
- [Sun94b] Sun Microsystems. Java Technology, 1994.
- [Sun94c] Sun Microsystems. JavaServer Pages Technology, 1994.
- [Sun02] Sun Microsystems. Java RMI over IIOP, 2002.
- [Tru04] TruePosition. An Examination of U-TDOA and Other Wireless Location Technologies: Their Evolution and Their Impact on Today's Wireless Market. Relatório Técnico, TruePosition, 2004.
- [Uni96] University of Minnesota. MapServer. 1996.
- [W3C96] W3C. Extensible Markup Language (XML), 1996.
- [W3C00] W3C. XML Schema, 2000.
- [W3C03] W3C. SOAP Version 1.2 Part 1: Messaging Framework, 2003.
- [W3C04] W3C. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, 2004.
- [W3S99a] W3Schools. WSDL Tutorial, 1999.
- [W3S99b] W3Schools. XML Schema Tutorial, 1999.
- [Wik04a] Wikipedia: The free encyclopedia. General Packet Radio Service, 2004.
- [Wik04b] Wikipedia: The free encyclopedia. Global Positioning System, 2004.

Apêndice A

Lista de Siglas

ADT	<i>Abstract Data Types</i>
A-GPS	<i>Assisted-Global Positioning System</i>
AOI	<i>Area of Interest</i>
API	<i>Application Program Interface</i>
BREW	<i>Binary Runtime Environment for Wireless</i>
CDMA	<i>Code Division Multiple Access</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CPqD	Centro de Pesquisa e Desenvolvimento em Telecomunicações
DCOM	<i>Distributed Component Object Model</i>
E-OTD	<i>Enhanced Observed Time Difference</i>
FCC	<i>Federal Communications Commission</i>
FUNTTTEL	Fundo para o Desenvolvimento Tecnológico das Telecomunicações
GIS	<i>Geographic Information System</i>
GML	<i>Geography Markup Language</i>
GMLC	<i>Gateway Mobile Location Center</i>
GPL	<i>GNU General Public License</i>
GPRS	<i>General Packet Radio Service</i>

GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IMA	<i>Israel Mobile Association</i>
J2ME	<i>Java 2 Micro Edition</i>
JRMI	<i>Java Remote Method Invocation</i>
LBS	<i>Location-Based Service</i>
LMU	<i>Location Measurement Units</i>
MIME	<i>Multipurpose Internet Mail Extensions</i>
MLP	<i>Mobile Location Protocol</i>
MMS	<i>Multimedia Messaging Service</i>
MPC	<i>Mobile Positioning Center</i>
NAICS	<i>North American Industry Classification System</i>
NENA	<i>National Emergency Number Association</i>
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>
OGC	<i>Open Geospatial Consortium</i>
OMA	<i>Open Mobile Alliance</i>
OpenLS	<i>Open Location Services</i>
PDA	<i>Personal Digital Assistant</i>
POI	<i>Point of Interest</i>
RPC	<i>Remote Procedure Call</i>
SAM	Serviços e Aplicações Móveis
SGBD	Sistema de Gerenciamento de Banco de Dados
SIC	<i>Standard Industrial Classification</i>

SIG	Sistema de Informação Geográfica
SIM	<i>Subscriber Identity Module</i>
SMLC	<i>Serving Mobile Location Center</i>
SMS	<i>Short Message Service</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SOC	<i>Service-Oriented Computing</i>
TDMA	<i>Time Division Multiple Access</i>
UDDI	<i>Universal Description, Discovery, and Integration</i>
UMTS	<i>Universal Mobile Telecommunications Systems</i>
U-TDOA	<i>Uplink Time Difference of Arrival</i>
W3C	<i>World Wide Web Consortium</i>
WAP	<i>Wireless Application Protocol</i>
WCS	<i>Web Coverage Service</i>
WFS	<i>Web Feature Service</i>
WGS	<i>World Geodetic System</i>
Wi-Fi	<i>Wireless Fidelity</i>
WLIA	<i>Wireless Location Industry Association</i>
WMS	<i>Web Map Service</i>
WSDL	<i>Web Services Description Language</i>
WS-I	<i>Web Services Interoperability</i>
XML	<i>Extensible Markup Language</i>
XSD	<i>XML Schema Definition</i>

Apêndice B

Exemplos de Requisição e Resposta dos Serviços OpenLS

A seguir são apresentados exemplos de requisição e resposta dos serviços OpenLS descritos no Capítulo 3.

B.1 Serviço de Diretório

“Onde é o Hospital mais próximo do Estádio Moisés Lucarelli?”

```
<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/xls
..\..\docs\OpenGIS\OpenLS\03-006r3\schema\DirectoryService.xsd" version="1">
<RequestHeader/>
<Request methodName="getDirectory" maximumResponses="10" requestID="1" version="1">
<DirectoryRequest sortCriteria="Distance" sortDirection="Ascending">
  <POILocation>
    <Nearest nearestCriterion="Shortest">
      <POI ID="1">
        <POIAttributeList>
          <POIInfoList>
            <POIInfo name="POI Name" value="Estádio Moisés Lucarelli"/>
          </POIInfoList>
        </POIAttributeList>
      </POI>
    </Nearest>
  </POILocation>
  <POIProperties directoryType="Yellow Pages">
```

```

    <POIProperty name="NAICS_type" value="Hospital"/>
  </POIProperties>
</DirectoryRequest>
</Request>
</XLS>

```

A resposta para esta requisição é a seguinte:

```

<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/xls
..\..\docs\OpenGIS\OpenLS\03-006r3\schema\DirectoryService.xsd" version="1">
  <ResponseHeader/>
  <Response requestID="1" version="1" numberOfResponses="1">
    <DirectoryResponse>
      <POIContext>
        <POI ID="1" phoneNumber="1932725501" POIName="Hospital Municipal Dr
Mário Gatti">
          <Address countryCode="BR">
            <StreetAddress>
              <Building number="340"/>
              <Street typePrefix="Av">Faria Lima</Street>
            </StreetAddress>
            <Place type="CountrySubdivision">São Paulo</Place>
            <Place type="Municipality">Campinas</Place>
            <Place type="MunicipalitySubdivision">Parque Itália</Place>
            <PostalCode>13036220</PostalCode>
          </Address>
        </POI>
      </POIContext>
    </DirectoryResponse>
  </Response>
</XLS>

```

B.2 Serviço de *Gateway*

Segue um exemplo para uma requisição de localização de um dispositivo móvel:

```

<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/xls
..\..\docs\OpenGIS\OpenLS\03-006r3\schema\GatewayService.xsd" version="1">

```

```

<RequestHeader/>
<Request maximumResponses="1" requestID="1" version="1" methodName="GetPosition">
  <SLIR>
    <InputGatewayParameters locationType="CURRENT_OR_LAST" priority="HIGH"
      requestedSrsName="WGS84">
      <RequestedQoP responseReq="Low_Delay" responseTimer="40">
        <HorizontalAcc>
          <Distance value="5"/>
        </HorizontalAcc>
        <VerticalAcc>
          <Distance value="5"/>
        </VerticalAcc>
      </RequestedQoP>
    <InputMSIDs>
      <InputMSInformation msIDType="MSISDN" encryption="ASC" msIDValue="
        +12066741000"/>
    </InputMSIDs>
  </InputGatewayParameters>
</SLIR>
</Request>
</XLS>

```

Segue uma resposta válida para a requisição acima:

```

<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\GatewayService.xsd" version="1">
  <ResponseHeader/>
  <Response numberOfResponses="1" requestID="1" version="1">
    <SLIA>
      <OutputGatewayParameters locationType="CURRENT_OR_LAST" priority="HIGH"
        requestedSrsName="WGS84">
        <RequestedQoP responseReq="Low_Delay" responseTimer="40">
          <HorizontalAcc>
            <Distance value="5"/>
          </HorizontalAcc>
          <VerticalAcc>
            <Distance value="5"/>
          </VerticalAcc>
        </RequestedQoP>
      <OutputMSIDs>
        <OutputMSInformation encryption="ASC" msIDType="MSISDN" msIDValue="
          +12066741000">

```

```

        <Position>
          <gml:Point>
            <gml:pos>23.50 -52.56</gml:pos>
          </gml:Point>
        </Position>
      </OutputMSInformation>
    </OutputMSIDs>
  </OutputGatewayParameters>
</SLIA>
</Response>
</XLS>

```

B.3 Geocodificação/Geocodificação Reversa

B.3.1 Serviço de Geocodificação

Segue um exemplo para uma requisição de geocodificação do endereço R. Conselheiro Paula Sousa, 1080 Cidade Universitária, Campinas - SP.

```

<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\LocationUtilityService.xsd" version="1">
  <RequestHeader/>
  <Request maximumResponses="1" methodName="getGeocodedAddress" requestID="1" version
    ="1">
    <GeocodeRequest>
      <Address countryCode="BR">
        <StreetAddress>
          <Building number="1080"/>
          <Street typePrefix="R">Conselheiro Paula Sousa</Street>
        </StreetAddress>
        <Place type="CountrySubdivision">São Paulo</Place>
        <Place type="Municipality">Campinas</Place>
        <Place type="MunicipalitySubdivision">Cidade UniversitÁria</Place>
      </Address>
    </GeocodeRequest>
  </Request>
</XLS>

```

A resposta para esta requisição é a seguinte:

```
<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\LocationUtilityService.xsd" version="1">
  <ResponseHeader/>
  <Response numberOfResponses="1" requestID="1" version="1">
    <GeocodeResponse>
      <GeocodeResponseList numberOfGeocodedAddresses="1">
        <GeocodedAddress>
          <gml:Point>
            <gml:pos>48.568595 -125.256858</gml:pos>
          </gml:Point>
          <Address countryCode="BR">
            <StreetAddress>
              <Building number="1080"/>
              <Street typePrefix="R">Conselheiro Paula Sousa</Street>
            </StreetAddress>
            <Place type="CountrySubdivision">São Paulo</Place>
            <Place type="Municipality">Campinas</Place>
            <Place type="MunicipalitySubdivision">Cidade Universitária</
              Place>
            <PostalCode>13083080</PostalCode>
          </Address>
          <GeocodeMatchCode accuracy="0.9"/>
        </GeocodedAddress>
      </GeocodeResponseList>
    </GeocodeResponse>
  </Response>
</XLS>
```

B.3.2 Serviço de Geocodificação Reversa

Uma pessoa deseja saber onde ela está em função da sua posição corrente. A aplicação deve definir a posição corrente do usuário através do Serviço de *Gateway* e posteriormente fazer uma requisição ao Serviço de Geocodificação Reversa para apresentar o endereço para o usuário:

```
<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
```

```

www.opengis.net/xls...\docs\OpenGIS\OpenLS\03-006r3\schema\
LocationUtilityService.xsd" version="1">
<RequestHeader/>
<Request maximumResponses="1" methodName="getReverseGeocodedAddress" requestID="1"
  version="1">
  <ReverseGeocodeRequest>
    <Position>
      <gml:Point>
        <!-- Posicao obtida do Servico de Gateway -->
        <gml:pos>45.784567 -123.563950</gml:pos>
      </gml:Point>
    </Position>
  </ReverseGeocodeRequest>
</Request>
</XLS>

```

O endereço encontrado na coordenada fornecida é retornado da seguinte forma:

```

<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls...\docs\OpenGIS\OpenLS\03-006r3\schema\
  LocationUtilityService.xsd" version="1">
<ResponseHeader/>
<Response numberOfResponses="1" requestID="1" version="1">
  <ReverseGeocodeResponse>
    <ReverseGeocodedLocation>
      <gml:Point>
        <gml:pos>45.784567 -123.563950</gml:pos>
      </gml:Point>
      <Address countryCode="BR">
        <StreetAddress>
          <Building number="935" buildingName="Poupa Tempo"/>
          <Street typePrefix="Av">Francisco Glicerio</Street>
        </StreetAddress>
        <Place type="CountrySubdivision">São Paulo</Place>
        <Place type="Municipality">Campinas</Place>
        <Place type="MunicipalitySubdivision">Centro</Place>
      </Address>
    </ReverseGeocodedLocation>
  </ReverseGeocodeResponse>
</Response>
</XLS>

```

B.4 Serviço de Apresentação de Mapas

O exemplo seguinte apresenta a requisição de um mapa para o Serviço de Apresentação de Mapas. Deseja-se obter um mapa do centro da cidade de Campinas, centralizado na Prefeitura Municipal, cuja localização pode ser obtida através do Serviço de Diretório.

Primeiramente é feita uma requisição para se recuperar os *layers*, estilos, formatos e Sistemas de Coordenadas oferecidos:

```
<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\PresentationService.xsd" version="1">
  <RequestHeader/>
  <Request maximumResponses="1" methodName="getCapabilites" requestID="1" version="1"
    >
    <GetPortrayMapCapabilitiesRequest/>
  </Request>
</XLS>
```

Uma resposta válida para a requisição `getCapabilites` é a seguinte:

```
<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\PresentationService.xsd" version="1">
  <ResponseHeader/>
  <Response numberOfResponses="1" requestID="1" version="1">
    <GetPortrayMapCapabilitiesResponse>
      <AvailableSRS>
        <SRS>WGS84</SRS>
      </AvailableSRS>
      <AvailableLayers>
        <Layer>BLK_BDY</Layer>
        <Layer>CENTER_LINE</Layer>
        <Layer>LOT</Layer>
        <Layer>STREET</Layer>
        <Layer>URBANISTIC</Layer>
      </AvailableLayers>
      <AvailableFormats>
        <Format>image/jpg</Format>
        <Format>image/png</Format>
```

```

    </AvailableFormats>
    <AvailableStyles>
      <Style>BLK_BDY</Style>
      <Style>CENTER_LINE</Style>
      <Style>LOT</Style>
      <Style>STREET</Style>
      <Style>URBANISTIC</Style>
    </AvailableStyles>
  </GetPortrayMapCapabilitiesResponse>
</Response>
</XLS>

```

A partir das informações de formato, sistema de coordenadas, layers e estilos disponíveis, pode ser montada a requisição do mapa com as informações desejadas:

```

<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\PresentationService.xsd" version="1">
  <RequestHeader/>
  <Request maximumResponses="1" methodName="getPresentation" requestID="1" version="1"
  ">
    <PortrayMapRequest>
      <Output content="URL" format="image/png" height="400" width="600"
        transparent="false">
        <CenterContext SRS="WGS84">
          <CenterPoint>
            <!-- Coordenada da Prefeitura Municipal de Campinas obtida do
              Servico de Directorio -->
            <gml:pos>46.893845 -124.639842</gml:pos>
          </CenterPoint>
          <DisplayScale>5000</DisplayScale>
          <DPI>120</DPI>
        </CenterContext>
      </Output>
      <Basemap filter="Include">
        <Layer name="BLK_BDY"/>
        <Layer name="CENTER_LINE"/>
        <Layer name="LOT"/>
        <Layer name="STREET"/>
        <Layer name="URBANIST"/>
      </Basemap>
    </PortrayMapRequest>
  </Request>

```

```
</XLS>
```

A resposta para esta requisição é uma URL na qual o mapa pode ser acessado.

```
<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\PresentationService.xsd" version="1">
  <ResponseHeader/>
  <Response numberOfResponses="1" requestID="1" version="1">
    <PortrayMapResponse>
      <Map>
        <Content format="image/png" height="400" width="600">
          <URL>http://www.maps.com/lbs/maps/385940.png</URL>
        </Content>
        <CenterContext SRS="WGS84">
          <CenterPoint>
            <gml:pos>46.893845 -124.639842</gml:pos>
          </CenterPoint>
          <DisplayScale>5000</DisplayScale>
          <DPI>120</DPI>
        </CenterContext>
      </Map>
    </PortrayMapResponse>
  </Response>
</XLS>
```

B.5 Serviço de Rotas

O usuário deseja visualizar um mapa e obter as instruções da rota da Av. Paulista, 1200 até o Parque do Ibirapuera. Uma requisição válida para este caso de uso é a seguinte:

```
<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schema\RouteService.xsd" version="1">
  <RequestHeader/>
  <Request maximumResponses="1" methodName="getRoute" requestID="1" version="1">
    <DetermineRouteRequest distanceUnit="M" provideRouteHandle="true">
      <RoutePlan useRealTimeTraffic="true">
```

```

<RoutePreference>Shortest</RoutePreference>
<WayPointList>
  <StartPoint>
    <Address countryCode="BR">
      <StreetAddress>
        <Building number="1200"/>
        <Street typePrefix="Av">Paulista</Street>
      </StreetAddress>
      <Place type="CountrySubdivision">São Paulo</Place>
      <Place type="Municipality">São Paulo</Place>
    </Address>
  </StartPoint>
  <EndPoint>
    <POI ID="1" POIName="Parque Ibirapuera"/>
  </EndPoint>
</WayPointList>
</RoutePlan>
<RouteInstructionsRequest format="text/plain"/>
<RouteMapRequest>
  <Output format="image/jpg" height="400" width="600"/>
</RouteMapRequest>
</DetermineRouteRequest>
</Request>
</XLS>

```

Um trecho da resposta para a requisição acima:

```

<?xml version="1.0" encoding="UTF-8"?>
<XLS xmlns="http://www.opengis.net/xls" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
  www.opengis.net/xls
  ..\..\docs\OpenGIS\OpenLS\03-006r3\schemas\RouteService.xsd" version="1">
  <ResponseHeader/>
  <Response numberOfResponses="1" requestID="1" version="1">
    <DetermineRouteResponse>
      <RouteHandle routeID="1"/>
      <RouteSummary>
        <TotalTime>P20M</TotalTime>
        <TotalDistance value="2740"/>
        <BoundingBox>
          <gml:pos>-23.57 -46.63</gml:pos>
          <gml:pos>-24.61 -46.74</gml:pos>
        </BoundingBox>
      </RouteSummary>
      <RouteGeometry>

```

```
<gml:LineString>
  <gml:pos>-23.61 -46.49</gml:pos>
  <gml:pos>-23.60 -46.50</gml:pos>
  <!-- Seguem todos os pontos que determinam a geometria da rota-->
  <gml:pos>-23.59 -46.63</gml:pos>
</gml:LineString>
</RouteGeometry>
<RouteInstructionsList>
  <RouteInstruction duration="P0M">
    <Instruction>Inicie o percurso em: Av Paulista 1200</Instruction>
    <distance value="0"/>
  </RouteInstruction>
  <!-- Seguem todas as intrucoes para percorrer a rota do ponto inicial
    ate o ponto final -->
  <RouteInstruction duration="P0.5M">
    <Instruction>Vire ã esquerda na Av Pedro Alvares Cabral e percorra
      150 metros</Instruction>
    <distance value="150"/>
  </RouteInstruction>
  <RouteInstruction duration="P15M">
    <Instruction>Ponto de destino: Av Pedro Alvares Cabral</Instruction>
    <distance value="0"/>
  </RouteInstruction>
</RouteInstructionsList>
<RouteMap>
  <Content format="image/png" height="400" width="600">
    <URL>http://www.maps.com/lbs/maps/516478.png</URL>
  </Content>
  <BoundingBox>
    <gml:pos>-23.57 -46.63</gml:pos>
    <gml:pos>-24.61 -46.74</gml:pos>
  </BoundingBox>
</RouteMap>
</DetermineRouteResponse>
</Response>
</XLS>
```

Apêndice C

Arquivo *deploy.wsdd* do Serviço de Apresentação de Mapas

Segue trecho do arquivo *deploy.wsdd* do Serviço de Apresentação de Mapas implementado no protótipo apresentado na seção 4.2:

```
<deployment
  xmlns="http://xml.apache.org/axis/wsdd/"
  xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
  <!-- Services from PresentationService WSDL service -->
  <service name="PresentationServicePort" provider="java:RPC" style="rpc" use="encoded
">
    <parameter name="wsdlTargetNamespace" value="http://www.cpqd.com.br/sam/wsdl/gis/
presentationService"/>
    <parameter name="wsdlServiceElement" value="PresentationService"/>
    <parameter name="wsdlServicePort" value="PresentationServicePort"/>
    <parameter name="className" value="br.com.cpqd.www.sam.wsdl.gis.
presentationService.PresentationServiceBindingSkeleton"/>
    <parameter name="wsdlPortType" value="PresentationServicePort"/>
    <parameter name="allowedMethods" value="*/>

    <typeMapping
      xmlns:ns="http://www.opengis.net/gml"
      qname="ns:CompassPointEnumeration"
      type="java:net.opengis.www.gml.CompassPointEnumeration"
      serializer="org.apache.axis.encoding.ser.EnumSerializerFactory"
      deserializer="org.apache.axis.encoding.ser.EnumDeserializerFactory"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    />
    <typeMapping
      xmlns:ns="http://www.opengis.net/xls"
```

```
    qname="ns:AvailableFormatsType"
    type="java:net.opengis.www.xls.AvailableFormatsType"
    serializer="org.apache.axis.encoding.ser.BeanSerializerFactory"
    deserializer="org.apache.axis.encoding.ser.BeanDeserializerFactory"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  />
  ...
</service>
</deployment>
```

Apêndice D

Mensagem SOAP enviada para o Serviço de Apresentação de Mapas

Segue mensagem SOAP enviada para o Serviço de Apresentação de Mapas para a requisição do mapa de localização de um agente:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd=
  "http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
  <soapenv:Body>
    <ns1:getPresentation soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/
      encoding/" xmlns:ns1="http://www.cpqd.com.br/sam/wsd/gis/presentationService">
      <PortrayMapRequest href="#id0"/>
    </ns1:getPresentation>
    <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.
      org/soap/encoding/" xsi:type="ns2:PortrayMapRequestType" xmlns:soapenc="http://
      schemas.xmlsoap.org/soap/encoding/" xmlns:ns2="http://www.opengis.net/xls">
      <Output href="#id1"/>
      <Basemap href="#id2"/>
      <Overlay href="#id3"/>
    </multiRef>
    <multiRef id="id2" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.
      org/soap/encoding/" xsi:type="ns3:LayerType" xmlns:ns3="http://www.opengis.net/
      xls" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    <multiRef id="id3" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.
      org/soap/encoding/" xsi:type="ns4:OverlayType" xmlns:ns4="http://www.opengis.
      net/xls" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
      <POI xsi:type="ns4:PointOfInterestType" xsi:nil="true"/>
      <RouteGeometry xsi:type="ns4:RouteGeometryType" xsi:nil="true"/>
    </multiRef>
  </Body>
</Envelope>
```

```

<Position href="#id4"/>
<Map xsi:type="ns4:MapType" xsi:nil="true"/>
<Style xsi:type="ns4:StyleType" xsi:nil="true"/>
</multiRef>
<multiRef id="id1" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns5:OutputType" format="image/png" height="250" transparent="false" width="208" xmlns:ns5="http://www.opengis.net/xls" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <BoundingBox href="#id5"/>
  <CenterContext xsi:type="ns5:CenterContextType" xsi:nil="true"/>
</multiRef>
<multiRef id="id5" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns6:EnvelopeType" xmlns:ns6="http://www.opengis.net/gml" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <pos href="#id6"/>
  <pos href="#id7"/>
  <pos xsi:type="ns6:DirectPositionType" xsi:nil="true"/>
  <pos xsi:type="ns6:DirectPositionType" xsi:nil="true"/>
</multiRef>
<multiRef id="id4" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns7:PositionType" xmlns:ns7="http://www.opengis.net/xls" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <Point href="#id8"/>
  <Ellipse xsi:type="ns7:EllipseType" xsi:nil="true"/>
  <CircleByCenterPoint xsi:type="ns8:CircleByCenterPointType" xsi:nil="true" xmlns:ns8="http://www.opengis.net/gml"/>
  <CircularArc xsi:type="ns7:CircularArcType" xsi:nil="true"/>
  <Polygon xsi:type="ns9:PolygonType" xsi:nil="true" xmlns:ns9="http://www.opengis.net/gml"/>
  <MultiPolygon xsi:type="ns10:MultiPolygonType" xsi:nil="true" xmlns:ns10="http://www.opengis.net/gml"/>
  <QoP xsi:type="ns7:QualityOfPositionType" xsi:nil="true"/>
  <Time xsi:type="ns7:TimeType" xsi:nil="true"/>
  <Speed xsi:type="ns7:SpeedType" xsi:nil="true"/>
  <Direction xsi:type="ns7:AngleType" xsi:nil="true"/>
</multiRef>
<multiRef id="id7" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns11:DirectPositionType" xmlns:ns11="http://www.opengis.net/gml" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  -47.05448761775622 -22.880949971677772</multiRef>
<multiRef id="id8" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns12:PointType" xmlns:ns12="http://www.opengis.net/gml" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <pos href="#id9"/>
</multiRef>
<multiRef id="id6" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

```

```
    org/soap/encoding/" xsi:type="ns13:DirectPositionType" xmlns:ns13="http://www.
    opengis.net/gml" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    -47.0574124449709 -22.883660046013777</multiRef>
  <multiRef id="id9" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.
    org/soap/encoding/" xsi:type="ns14:DirectPositionType" xmlns:ns14="http://www.
    opengis.net/gml" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    -47.05595001684953 -22.8823050160925</multiRef>
</soapenv:Body>
</soapenv:Envelope>
```