



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**TECNOLOGIAS E PADRÕES ABERTOS PARA O DOMÍNIO GEOGRÁFICO  
NA WEB: UM ESTUDO EM ECOTURISMO**

Francisco Luís Pompéia Gioielli

Dissertação de Mestrado em Computação Aplicada, orientada pelo  
Dr. Antônio Miguel Vieira Monteiro

INPE  
São José dos Campos  
**2006**

*"Aceita o conselho dos outros, mas nunca  
desistas da tua própria opinião"*

William Shakespeare

A meus pais e à minha esposa,  
pela colaboração e carinho...

## AGRADECIMENTOS

Primeiramente, agradeço a meus pais, Carlos e Margarida, por todo esforço e carinho dedicados à minha educação. Em especial ao meu pai pela lembrança de que é sempre possível e necessário evoluir. E à minha mãe por me ensinar a sonhar sem deixar de ser feliz com a realidade.

À minha querida esposa Sabrina, com quem convivo com muito amor há nove anos, que sempre me apóia em minhas decisões e foi a principal revisora desta dissertação, seguida da minha mãe.

Ao meu orientador Dr. Antônio Miguel Vieira Monteiro, que soube aproveitar e ampliar minhas capacidades, e foi fundamental no processo de elaboração e correção deste trabalho.

Ao primo e amigo Dr. Reinaldo Roberto Rosa, que me apresentou o INPE, me incentivou a ingressar no curso e, durante este, como coordenador, fez um excelente trabalho, sempre dando conselhos e orientações precisas a mim e a meus colegas.

Aos primos Pedro e Carla, que durante o curso, por diversas vezes, me acolheram com muita atenção e amizade em seu lar em São José dos Campos, e que em nossas longas conversas me ensinaram muito sobre pesquisa, física e ciência em geral.

À Nilde, que, como minha analista por muitos e muitos anos, me deu a ferramenta do auto-conhecimento, fundamental para a superação de nossos limites.

À Geograph, empresa onde trabalho há mais de sete anos, que consentiu com minha ausência por diversas vezes durante esta jornada.

À MSc. Alda Monteiro Barbosa, pelo seu belo trabalho feito em Capitólio, que motivou a existência do meu, e por ter sido atenciosa e prestativa em todos os momentos em que precisei.

Ao Dr. Gilberto Câmara, por suas aulas memoráveis de Introdução ao Geoprocessamento, sempre repletas de bom conteúdo, e por suas mensagens contra a mediocridade na produção científica.

Aos colegas Warley, Joanito, Giani, Daniela, Íris, Adriana, Andréia, Adair, Sérgio, Danilo, Magda, Bruno, Dawilmar, dentre outros, pelo companheirismo e pelos momentos divertidos que passamos juntos durante as aulas no INPE.

Ao Dr. José Luiz Antunes de Oliveira e Sousa, que, como meu orientador de iniciação científica durante meu curso de graduação, despertou meu interesse pela pesquisa acadêmica, e por ter me recomendado no processo de ingresso neste curso.

Ao Dr. Paulo Sérgio Franco Barbosa, coordenador do meu curso de graduação, que também me recomendou no processo de ingresso neste curso.

Ao colega Gilberto Ribeiro, que me ensinou muito sobre computação, geoprocessamento e, em especial, sobre a TerraLib.

Por fim, agradeço ao povo brasileiro, que, ao financiar o INPE, me ajudou a concretizar este trabalho.

## RESUMO

Este trabalho apresenta contribuições para os bancos de dados espaciais compartilhados, para a biblioteca de software livre TerraLib, e para os sistemas de informação ao ecoturista. A presente dissertação apresenta uma solução de arquitetura para sistemas de informação no domínio geográfico apoiados na web. A arquitetura em questão é baseada no conceito de agentes, modelados como serviços para a Web - Web Services – e atuando sobre bancos de dados espaciais compartilhados, sob condições de alta interatividade. A solução proposta se baseia em protocolos e padrões abertos e no uso extensivo de software livre. Uma implementação desta arquitetura foi desenvolvida utilizando a TerraLib, o sistema de banco de dados MySQL, a plataforma de web services Apache Axis, e o servidor web Apache HTTP Server. O protótipo de um sistema de informação ao ecoturista foi implementado para demonstrar a validade dos conceitos e da arquitetura proposta. Este protótipo apresenta as características da arquitetura aplicadas em um sistema de apoio ao ecoturista para a região de Capitólio-MG, fazendo uso de um banco de dados espaciais da referida região, disponível a partir de um trabalho anterior, e demonstra a capacidade desta proposta em acomodar padrões, especificações e contextos já previstos e ainda não previstos nas especificações atuais. Dentro deste conceito de extensibilidade, é proposto um serviço de georreferenciamento qualitativo, e é discutida a aderência desta solução aos padrões e especificações do Open Geospatial Consortium.

# **TECHNOLOGIES AND OPEN STANDARDS FOR GEOGRAPHIC DOMAIN ON THE WEB: A STUDY IN ECOTOURISM**

## **ABSTRACT**

This work represents contributions for the shared spatial databases, for the free software library TerraLib, and for information systems to support ecotourists. This work presents an architecture solution for Web based information systems in the geographic domain. This architecture is based on the concept of agents, modeled as Web Services, acting over shared spatial databases, under conditions of high interactivity. The proposed solution is based on open protocols and standards and wide use of free software. An implementation of this architecture was developed using the TerraLib, the Apache Axis Web Services framework, the Apache HTTP Web Server and the MySQL database. A prototype of an information system to support ecotourists was developed as a proof-of-concept for this architecture. This prototype presents the features of the proposed architecture applied on a system to support ecotourists in the area of Capitolio – MG, using a previously created spatial database of the referred region, which proves that this proposal can accommodate standards, specifications and contexts, determined and still undetermined. Under this concept of extensibility, an innovative service of qualitative georeferencing is proposed, and its adherence to the standards and specifications of the Open Geospatial Consortium.

## SUMÁRIO

	Pág.
<b>LISTA DE FIGURAS</b>	<b>10</b>
<b>LISTA DE TABELAS</b>	<b>12</b>
<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>13</b>
1.1 Motivação.....	13
1.2 Objetivo.....	16
1.3 Organização da Dissertação.....	17
<b>CAPÍTULO 2 - O CONCEITO DE UMA BASE VIVA.....</b>	<b>18</b>
2.1 Definição.....	18
2.2 Extensões de uso da Base Viva .....	20
2.3 Agentes como meios de disseminação da Base Viva.....	23
2.3.1 Interação agente – base de informações.....	24
2.3.2 Interação agente – usuário.....	26
2.3.3 Agentes implementados como web services.....	27
2.5 Comentários finais.....	28
<b>CAPÍTULO 3 - TERRA WEBSERVICES.....</b>	<b>30</b>
3.1 A Tecnologia de Web Services.....	30
3.2 Web Services para o domínio geográfico.....	32
3.3 Arquitetura do Terra WebServices.....	33
3.3.1 TerraLib.....	34
3.3.2 Apache Axis.....	35
3.3.3 Apache HTTP Server.....	36
3.3.4 SGBD.....	36
3.4 Contexto tecnológico.....	36
3.5 Propostas de serviços para o Terra WebServices.....	38
3.6 Aderência do Terra WebServices às especificações do OGC.....	42
3.7 Escalabilidade do Terra WebServices.....	45
3.8 Comentários finais.....	46

<b>CAPÍTULO 4 - GEOAGENTE: UM AGENTE PARA O GEORREFERENCIAMENTO QUALITATIVO.....</b>	<b>48</b>
4.1 Processos de georreferenciamento.....	48
4.2 Georreferenciamento qualitativo.....	50
4.3 Interpretação de descrições informais com elementos oriundos de raciocínio espacial.....	52
4.4 Raciocínio espacial.....	53
4.5 Proposta de agente para o georreferenciamento qualitativo.....	55
4.6 O GeoAgente como um web service.....	58
4.7 Comentários finais.....	59
<b>CAPÍTULO 5 - UM SISTEMA PARA CAPITÓLIO/MG.....</b>	<b>60</b>
5.1 Motivação.....	60
5.2 Domínio do problema.....	60
5.3 Arquitetura.....	62
5.4 Metodologia de desenvolvimento.....	64
5.4.1 Terra WebServices.....	64
5.4.2 Capitólio Web.....	64
5.4.3 GeoAgente.....	64
5.5 Interação usuário – sistema Capitólio Web – Terra WebServices.....	65
5.6 Comentários finais.....	70
<b>CAPÍTULO 6 - CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>72</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>74</b>
<b>ANEXO I - INTERATIVIDADE DE SIG PARA WEB.....</b>	<b>79</b>
<b>ANEXO II - MÉTODOS DO PROTÓTIPO DO TERRA WEBSERVICES.....</b>	<b>97</b>
<b>ANEXO III - EXEMPLO DE UTILIZAÇÃO DO GEOAGENTE.....</b>	<b>90</b>
<b>ANEXO IV - REQUISITOS PARA UM SISTEMA DE INFORMAÇÕES AO     TURISTA.....</b>	<b>92</b>

## LISTA DE FIGURAS

	Pág.
2.1 ESQUEMA ILUSTRATIVO DO CONCEITO DA SEPARAÇÃO DAS INFORMAÇÕES NA BASE VIVA.....	20
2.2 A BASE VIVA E SEUS POTENCIAIS USUÁRIOS.....	22
2.3 ARQUITETURA BÁSICA DE UM AGENTE PARA SISTEMAS COMPUTACIONAIS.....	23
2.4 O AGENTE COMO INTERFACE ENTRE OS USUÁRIOS E A BASE VIVA.....	24
2.5 COMPOSIÇÃO DE AGENTES.....	26
2.6 ENTIDADES ENVOLVIDAS NA BASE VIVA.....	29
3.1 ARQUITETURA DA TECNOLOGIA DE WEB SERVICES.....	31
3.2 ARQUITETURA DO TERRA WEBSERVICES.....	34
3.3 CONTEXTO TECNOLÓGICO DO TERRA WEBSERVICES.....	37
3.4 PARÂMETROS DO MÉTODO GETCAPABILITES.....	43
3.5 PARÂMETROS DO MÉTODO GETMAP RELATIVO AO SERVIÇO WMS DO OGC.....	44
3.6 COMPOSIÇÃO DE SERVIÇOS BÁSICOS PARA A CONSTRUÇÃO DO SERVIÇO WMS ESPECIFICADO PELO OGC.....	45
3.7 ESCALABILIDADE DOS TERRA WEBSERVICES.....	46
4.1 ARQUITETURA DOS AGENTES EM CONTEXTOS DE GEORREFERENCIAMENTO.....	49
4.2 ARQUITETURA DO AGENTE PARA GEORREFERENCIAMENTO QUALITATIVO.....	51
4.3 RELAÇÕES ESPACIAIS BIDIMENSIONAIS.....	53
4.4 POSIÇÕES RELATIVAS A UM CAMINHO PERCORRIDO ENTRE OS PONTOS A E B.....	54
4.5 INTERAÇÕES USUÁRIO – GEOAGENTE.....	55
4.6 ESQUEMA PARA AUXILIAR O USUÁRIO NA IDENTIFICAÇÃO DA POSIÇÃO RELATIVA ENTRE A TRILHA E LOCAL ENCONTRADO.....	57

4.7	RESULTADO DE UM GRUPO DE ITERAÇÕES EXECUTADAS PELO GEOAGENTE NO PROCESSO DO GEORREFERENCIAMENTO QUALITATIVO.....	58
4.8	O GEOAGENTE COMO UM CLIENTE DO TERRA WEBSERVICES.....	59
5.1	IMAGEM SINTÉTICA DA REGIÃO DO MUNICÍPIO DE CAPITÓLIO E MUNICÍPIOS VIZINHOS.....	61
5.2	COMPONENTES, PROTOCOLOS E PADRÕES UTILIZADOS.....	63
5.3	ARQUITETURA DO PROTÓTIPO PARA CAPITÓLIO.....	62
5.4	PLATAFORMA DE UML UTILIZADA PARA O GEOAGENTE.....	65
5.5	TELA INICIAL DO PROTÓTIPO, ONDE DEVEM SER INSERIDAS AS INFORMAÇÕES PARA A CONEXÃO À BASE DE DADOS.....	66
5.6	SEGUNDA TELA DO PROTÓTIPO, ONDE SÃO APRESENTADAS AS OPÇÕES DE LAYERS DISPONÍVEIS NA BASE DE DADOS CONECTADA.....	67
5.7	TELA DE MAPA OBTIDA QUANDO EXISTEM LAYERS HABILITADOS NAS OPÇÕES.....	67
5.8	TELA DE MAPA OBTIDA APÓS OPERAÇÃO DE VÔO.....	68
5.9	POPUP COM INFORMAÇÕES DO OBJETO CLICADO.....	69
5.10	TELA DE ABERTURA DO PROGRAMA GOOGLE EARTH.....	70
5.11	PROGRAMA GOOGLE EARTH EXIBINDO INFORMAÇÕES EXPORTADAS DA BASE DE CAPITÓLIO .....	70
A.1	RECURSOS INTERATIVOS DISPONÍVEIS EM ALGUNS SITES.....	84
A.2	ÁREA NO INÍCIO DO PROCESSO DE ITERAÇÕES.....	90
A.3	ÁREA APÓS O PRIMEIRO CONJUNTO DE ITERAÇÕES.....	91
A.4	SOBREPOSIÇÃO ENTRE AS ÁREAS INICIAL E APÓS O PRIMEIRO GRUPO DE ITERAÇÕES.....	91
A.5	CASOS DE USO.....	96
A.6	DIAGRAMA DE CLASSES.....	97

## LISTA DE TABELAS

	Pág.
4.1 PROPOSTA DE WEB SERVICES E SEUS MÉTODOS PARA O TERRA WEBSERVICES.....	39

## CAPÍTULO 1

### INTRODUÇÃO

#### 1.1 Motivação

A cada dia que passa, um maior número de organizações públicas e privadas tem se beneficiado, ou ao menos vislumbrado os benefícios, do geoprocessamento. As atividades de construção e manutenção de uma base de informações geográficas, pré-requisito para o geoprocessamento, geralmente envolvem altos custos. No entanto, caso esta base seja bem planejada e administrada, é passível de ser utilizada para diversas finalidades, extrapolando os limites de uso estabelecidos pelos interesses particulares desta ou daquela organização (Davis Jr. e Alves, 2005). Desse esforço, resultam bases de dados espaciais com grande volume de informação com vínculo geográfico, e qualidade cartográfica conhecida.

Quando o acesso a essas bases é possibilitado de modo amplo e aberto, um universo de novos contextos de uso para estas informações aparece, e traz consigo um conjunto de velhos e novos desafios. Considere-se um exemplo: o caso do **ecoturismo**. De acordo com Embratur (1994), o ecoturismo é “um segmento da atividade turística que utiliza, de forma sustentável, o patrimônio natural e cultural, incentiva sua conservação e busca a formação de uma consciência ambientalista através da interpretação do meio ambiente, promovendo o bem estar das populações envolvidas”.

Talvez o principal diferencial do ecoturismo, em relação às outras modalidades de turismo, seja a possibilidade do viajante interagir de forma intensa com o meio-ambiente, extrapolando os limites da observação. Esta forma de interação permite que o ecoturista descubra, freqüentemente, novas atrações ou formas diferentes de se relacionar com as já conhecidas, o que transforma este viajante em uma potencial fonte de informações sobre o local visitado, que se soma às outras disponíveis.

Uma plataforma tecnológica que permita a interação, através da troca de informações, entre os diversos agentes - futuros viajantes, viajantes com experiência no local, a comunidade do lugar, o poder público em esfera local ou regional e o setor privado - pode se transformar em uma importante ferramenta para fomentar e melhorar a prática do ecoturismo, sobretudo no Brasil, que é um dos países com maior potencial neste segmento (Barbosa, 2003). Isto porque informações abrangentes, atualizadas e dispostas adequadamente permitem e motivam o ecoturista a planejar sua viagem (Almer, *et al.*, 2004). Portanto, quanto maior for a intensidade da interação entre os agentes envolvidos e a base de informações, maior será sua abrangência, atualização e utilização e, conseqüentemente, maior será sua eficiência. Por outro lado, como mapas e outras informações sobre o espaço são itens indispensáveis para os viajantes (Oliveira e Decanini, 2002), a base de informações deve ser capaz de lidar com informações espaciais desta natureza.

Neste contexto, onde a interação intensa com a base de informações é necessária e fundamental, quatro grandes desafios aparecem: Como facilitar o processo de disponibilização das bases espaciais? Como manter íntegras as bases disponibilizadas para acesso? Como permitir o compartilhamento das novas informações produzidas entre todos os agentes no contexto do problema? Como possibilitar um crescimento nas funcionalidades que permitem interação com a base estendendo os usos do sistema?

Para responder a estas questões é preciso encontrar uma plataforma tecnológica que possua grande capacidade de comunicação, por onde possam trafegar dados espaciais com eficiência sob protocolos padrões e abertos e, ao mesmo tempo, seja flexível para poder ser acoplada com facilidade aos sistemas e processos que demandam ou geram dados e informações, existentes ou em desenvolvimento nas organizações.

Dentre os atuais paradigmas de sistemas de informação, a plataforma *Web* parece ser a que mais se aproxima destes requisitos. Isto porque a *Web* confere aos sistemas de informação que a utilizam como plataforma de comunicação, os benefícios da flexibilidade, maior acessibilidade e menores riscos de obsolescência e isolamento (Anderson, Moreno-Sanchez, 2003). Os principais motivos para isso são o fato de que a

Web está cada vez mais difundida geográfica e socialmente, além de ser compatível com a maioria dos dispositivos computacionais existentes e em desenvolvimento, participantes de algum processo remoto de troca de informações. Porém, a web é apenas a plataforma de comunicação. Sobre ela devem estar apoiados sistemas que atendam às demandas dos usuários com eficiência e que possuam recursos para manter as bases de informações espaciais íntegras e atualizadas.

Considerando as questões levantadas, esta dissertação parte da hipótese de que os desafios levantados pelos novos contextos de uso da informação geográfica, disponibilizada em situações onde a alta interatividade entre os agentes e a base espacial é requisito fundamental, têm uma resposta de base tecnológica: (a) a utilização da Web como plataforma de comunicação, o que estabelece o uso de seus protocolos e padrões para o tráfego dos dados e informações; e (b) uma arquitetura de serviços baseada no uso de bibliotecas e/ou componentes abertos, no estado-da-arte em recursos de geoprocessamento, sobre a qual possa ser desenvolvida a funcionalidade necessária ao contexto da aplicação, ampliando as possibilidades da participação ativa dos diversos agentes envolvidos na produção da base de informações espaciais, sejam eles usuários, dispositivos ou outros sistemas.

Partindo da hipótese formulada e das premissas estabelecidas, esta dissertação apresenta o conceito de uma Base Viva, que estabelece a possibilidade de interação direta entre agentes, usuários e dispositivos, com a base de dados espaciais disponibilizada ao acesso público. Este conceito permite a criação e disponibilização de novas informações, porém observando a integridade e a consistência das informações da base consolidada. Para viabilizá-lo, é proposta uma arquitetura computacional, baseada em uma plataforma de serviços na Web. Uma implementação desta arquitetura foi desenvolvida com base na biblioteca de componentes geográficos TerraLib<sup>1</sup>, no sistema gerenciador de banco de dados MySQL<sup>2</sup>, no servidor web Apache HTTP Server<sup>3</sup> e na plataforma de *web services* Apache Axis<sup>4</sup>. Um serviço geográfico na web, denominado

---

1 <http://www.terralib.org>

2 <http://www.mysql.com>

3 <http://httpd.apache.org>

4 <http://ws.apache.org/axis>

*georreferenciamento qualitativo*, foi proposto, especificado e implementado sob a arquitetura apresentada. Por fim, o protótipo de um aplicativo, com base na temática do ecoturismo para a região de Capitólio, MG, foi desenvolvido e implementado nesta arquitetura como prova dos conceitos desenvolvidos nesta dissertação, apresentando as principais características de interatividade e extensibilidade necessárias a esse e outros domínios de aplicação.

## **1.2 Objetivo**

Esta dissertação teve como objetivo geral explorar os recursos de comunicação e padronização da plataforma Web e o uso da arquitetura de serviços nela baseada – *Web Services* – no contexto de uso relacionado a bases de informações espaciais. Com isso, uma solução baseada em padrões e softwares abertos, que amplia as possibilidades de interação para bases de dados geográficas, é apresentada. Para desenvolvedores de aplicativos geográficos, esta dissertação apresenta e implementa uma proposta de arquitetura que permite a redução dos custos e do tempo de implementação de uma solução, através de técnicas de incorporação de funcionalidades de bibliotecas e/ou componentes geográficos de terceiros e o uso extensivo de padrões abertos. Esta arquitetura permite a aplicação do uso dos Sistemas de Informação Geográfica (SIG) em contextos onde a participação ativa sobre bases de informações, tornadas públicas, é requisito essencial.

Para atender a este objetivo geral, os seguintes objetivos específicos foram estabelecidos:

(a) O conceito de Base Viva foi desenvolvido. Uma arquitetura computacional, baseada em uma arquitetura de serviços na web, que dá suporte computacional ao conceito apresentado, é proposta;

(b) Uma implementação para a arquitetura computacional proposta foi desenvolvida com base na biblioteca TerraLib: a plataforma Terra WebServices. Através desta implementação foi possível explorar os benefícios dos sistemas baseados na web, para o caso de dados geográficos, e ainda simplificar o uso dos recursos da biblioteca

TerraLib, por programadores de aplicação, em sistemas existentes ou em desenvolvimento;

(c) Uma revisão dos recursos interativos sobre bases de dados espaciais na Web e SIG para Web foi realizada. Uma categorização para os recursos analisados foi proposta e apresentada, gerando um sistema de classificação baseado em recursos básicos, avançados e inovadores presentes nos sistemas em operação pesquisados;

(d) Um novo recurso de interação, o georreferenciamento qualitativo, foi proposto, especificado e implementado como um serviço geográfico, na arquitetura apresentada;

(e) Um protótipo de sistema, com a temática do ecoturismo, para a região de Capitólio, MG, foi desenvolvido para demonstrar as possibilidades de uso da arquitetura. Para esse estudo de caso foi utilizada a plataforma Terra WebServices. O protótipo implementa o conceito de Base Viva e possibilita a interação dos ecoturistas e seus dispositivos com um banco de dados geográfico da área visitada, comentando e acrescentando dados e informações e utilizando os recursos dos serviços disponíveis através da arquitetura. Neste caso, a ênfase foi dada ao serviço de georreferenciamento qualitativo, como um inovador recurso de interatividade.

### **1.3 Organização da Dissertação**

Esta dissertação está organizada em 6 capítulos. No Capítulo 2 é apresentado o conceito de Base Viva. O Capítulo 3 apresenta a proposta de arquitetura de serviços geográficos para web. O Capítulo 4 é destinado à apresentação da proposta de um serviço novo, o georreferenciamento qualitativo. No Capítulo 5, mostra-se o protótipo de um sistema para informação ao ecoturista, desenvolvido com os dados de Capitólio/MG, fazendo uso do Terra WebServices e implementando o serviço de georreferenciamento qualitativo. Por fim, no Capítulo 6, apresentamos as conclusões e possibilidades de trabalhos futuros.

## CAPÍTULO 2

### O CONCEITO DE UMA BASE VIVA

#### 2.1 Definição

Para auxiliar na compreensão do conceito a ser apresentado, é preciso imaginar um cenário, que mostre um contexto de uso para um sistema de informação que necessita de altas taxas de atualização e de alta capacidade de interação remota e distribuída. Um praticante de *trekking*, que é a atividade física da caminhada acompanhada de alguma motivação, como, por exemplo, o relaxamento, a reflexão ou o convívio com a natureza, será o usuário no cenário descrito a seguir.

Este usuário planeja escolher um lugar para praticar *trekking* e outras atividades de turismo de aventura<sup>5</sup>. Para isso, utiliza a internet como meio facilitador para a sua busca e encontra referências e informações. Ele encontrou um site com informações sobre uma localidade para a sua prática, que também dispunha de um banco de dados com acesso disponível a diversos mapas. Os mapas das trilhas também estavam lá. O usuário encontrou ali uma trilha que lhe despertou enorme interesse. Ele decide, então, percorrer esta trilha. Ele viaja e leva consigo os mapas que capturou no site da internet, afinal, sabe que estes mapas são ferramentas simples e importantes para orientação em ambientes desconhecidos (Gartner *et al.*, 2004). Ele chega, se instala e vai percorrer a trilha. A primeira surpresa acontece, a trilha não estava exatamente onde o mapa que ele encontrara na consulta à internet dizia estar. Mais adiante, um rio indicado no mapa que ele trazia não foi localizado na trilha, mas ele encontrou uma cachoeira, em outra parte da trilha, que não constava do seu mapa de orientação. Os mapas de trilhas, por nem sempre poderem contar com informações corretas do ponto de vista cartográfico,

---

<sup>5</sup> Segmento do mercado turístico que promove a prática de atividades de aventura e esporte recreacional, em ambientes naturais e espaços urbanos ao ar livre, que envolvam emoções e riscos controlados, exigindo o uso de técnicas e equipamentos específicos, a adoção de procedimentos para garantir a segurança pessoal e de terceiros e o respeito ao patrimônio ambiental e sócio-cultural (Embratur, 1994)

exploram extensamente as capacidades cognitivas de seus leitores, tais como o senso de distância e orientação (Frank, 1998), fazendo uso de referências a elementos do ambiente, tais como árvores, rios, depressões e elevações. O usuário gostaria então de compartilhar com os responsáveis pela base de dados esses problemas e acrescentar à base as novidades que encontrou, para ajudar a melhorar as informações disponíveis. A base, no entanto, já continha várias informações importantes e corretas, por exemplo, aquelas relativas a geomorfologia e a geologia da região, que este usuário não poderia alterar. A questão passou a ser então como possibilitar ao usuário, interagir com esta base de informações disponibilizada, melhorando a qualidade dos dados e das informações da mesma, mas sem alterar aquilo que na base já tinha qualidade.

É neste cenário que será apresentado um conceito simples, para uma base de informações de interesse público, com conteúdo geográfico, que disponha de mecanismos para consulta, inserção e edição de informações pelos seus múltiplos usuários, com possibilidade de acesso remoto via *web*.

Esta base de dados geográficos possibilita a inserção contínua de novos dados geográficos e informações associadas a eles, e dissemina estas inserções na forma de contribuições ao conteúdo da base. Estas contribuições podem possuir ou não identificação da fonte provedora da informação. É importante estabelecer que esta inserção de novos elementos é feita sobre uma base de conteúdos já estabelecida para o domínio da aplicação, e que tem qualidade de posicionamento e qualidade cartográfica adequadas ao seu contexto de uso. Por exemplo, na situação de uso do cenário proposto, esta seria uma base geográfica com a descrição da geologia, da geomorfologia, dos solos, da drenagem e das estradas catalogadas na região visitada, entre outros aspectos. Para um contexto urbano, esta base teria setores censitários, divisões administrativas (como distritos, por exemplo), as ruas catalogadas, praças e equipamentos públicos, entre outros. A estas bases constituídas denominamos **Base Consolidada**. Para o conjunto de usuários, entretanto, não há diferença visível entre o conteúdo da Base Consolidada e as novas inserções. Para estes, a base é observada como em um processo de constante atualização, sem distinção entre novas inserções e dados consolidados. Uma base de dados geográfica com estas características é que denominamos **Base Viva**.

O termo busca enfatizar a natureza dinâmica do conteúdo da base, característica de seu uso em contextos que requisitam alta interatividade entre os usuários e a base de informação, que é, enfim, o elo de comunicação entre os diversos usuários conectados. A Figura 2.1 auxilia a compreender esquematicamente a separação entre as atualizações dos usuários, localizadas em uma camada separada das camadas com as informações relativas a Base Consolidada.

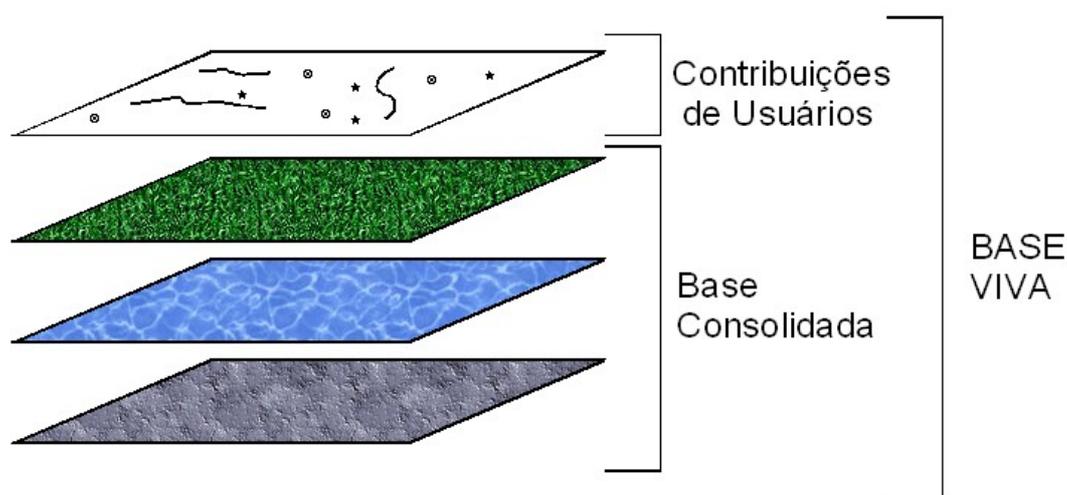


FIGURA 2.1 – Esquema ilustrativo do conceito da separação das informações na Base Viva

## 2.2 Extensões de uso da Base Viva

Estendendo o cenário do sistema de informações ao ecoturista, é possível imaginar que seria pertinente que a base contivesse informações indicando a previsão do tempo. Estas informações podem ajudar os usuários a decidir, dentre as alternativas de passeio, a mais adequada para um determinado dia. Por exemplo, em um dia para o qual está sendo prevista forte chuva, deve-se evitar as áreas nas proximidades de cursos de rios que apresentam risco de tromba d'água. Já em um dia de pouca nebulosidade, deve-se evitar caminhadas longas, em trechos abertos, dando-se preferência às caminhadas em áreas de mata fechada, onde as temperaturas são mais amenas e a exposição ao sol é menor, reduzindo, assim, o risco de desidratação e insolação. No Brasil, essas informações climáticas já estão disponíveis na internet, dentre outros meios de

comunicação, por iniciativa do Centro de Previsão do Tempo e Estudos Climáticos – CPTEC – ligado ao Instituto Nacional de Pesquisas Espaciais – INPE. Portanto, seria interessante que a Base Viva, neste contexto, também fosse alimentada permanentemente, de alguma forma, por esta fonte de informações, já disponível em um sistema *web*.

Em um contexto semelhante, de um rio onde se pratica *rafting*, outra idéia aparece. O *rafting* é um esporte coletivo caracterizado pela descida de um trecho de rio ou curso d'água, onde ocorrem corredeiras, por uma equipe de remadores em um bote. Por envolver um risco maior de acidentes em relação a outros esportes, como o *trekking*, o *rafting* é considerado um esporte radical, dentre as modalidades de turismo de aventura. Neste esporte é fundamental conhecer o nível das águas do rio, na ocasião em que se deseja praticá-lo. Com base nesta informação e em outras como o traçado e relevo do trecho, e, preferencialmente, com o auxílio de um guia com experiência no local, é possível determinar as alternativas viáveis e seguras de percurso no trecho, e, posteriormente, escolher dentre estas alternativas a que mais se adequa ao nível de risco e “emoção” pretendido pela equipe.

Dentre as alternativas de dispositivos para a medição do nível das águas, estão os sensores eletrônicos, que, quando utilizados, são estrategicamente posicionados ao longo do trecho monitorado, e transmitem suas leituras para uma base de monitoração. Neste contexto, seria bastante interessante que a Base Viva também fosse alimentada, de forma permanente, com as informações provenientes destes sensores, pois assim seria possível fazer o planejamento da descida antes de ir até o local, evitando-se que, no caso da descida estar impossibilitada, os remadores se desloquem à toa.

Baseando-se nestes exemplos – previsão do tempo e nível do curso d'água –, é visível que as atividades de inserção e edição de informações na Base Viva não devem se limitar aos usuários, mais caracterizados por pessoas, mas sim abranger outras possibilidades, como dispositivos e outros sistemas, os quais, em outros contextos, poderiam também buscar informações na Base Viva, e que, desta forma, se comportam também como usuários do sistema. A Figura 2.2 ilustra esta ampliação no conjunto de

possíveis usuários da Base Viva.

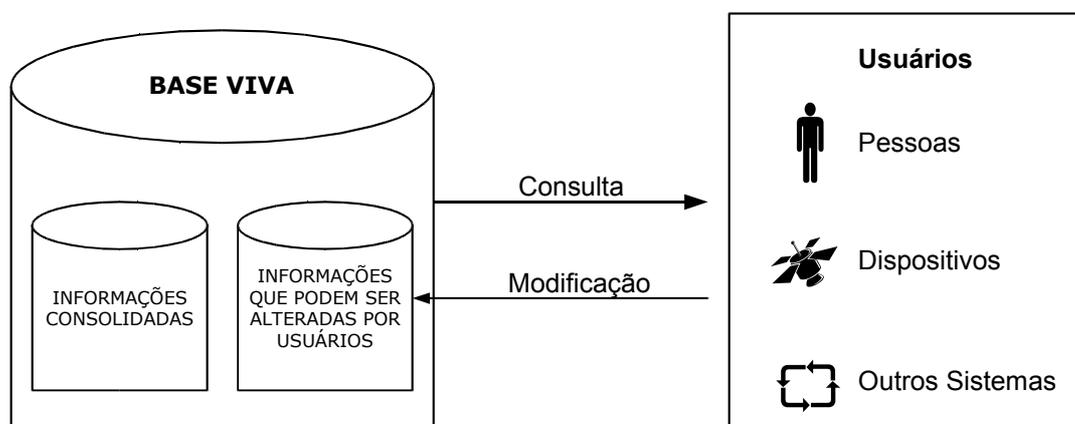


FIGURA 2.2 – A Base Viva e seus potenciais Usuários

Este conceito (de que podem existir diversos tipos de usuários para a Base Viva) traz consigo novas demandas para os recursos de interatividade.

Para projetar um sistema de informação eficiente, mesmo com tantas possibilidades de interação, foram definidas duas alternativas. (1) estabelecer todos os requisitos possíveis e imagináveis para compor um sistema completo e sofisticado, para cujo desenvolvimento, seria necessário consumir muito tempo e recursos; ou (2) projetar uma arquitetura com alguns elementos básicos, na qual seja simples implementar novos recursos, conforme as necessidades apareçam.

Esta dissertação parte da segunda alternativa. Ou seja, projetar uma arquitetura com alguns elementos básicos, que possa ser estendida com facilidade, de acordo com as novas necessidades, através de novos e independentes elementos que ampliam as possibilidades do sistema, sem interferir nas já estabelecidas.

A chave desta arquitetura são estes elementos independentes, que fazem a interface entre os usuários e a base de informações, e que possuem funcionalidades e mecanismos simples, projetados para atender a um número reduzido de necessidades específicas. Nestas condições, considerando-se os atuais paradigmas dos sistemas de informação, estes elementos podem ser chamados de **agentes**, dotados de alguns recursos de

comunicação e alguma inteligência, cuja definição é apresentada a seguir.

### 2.3 Agentes como meios de disseminação da Base Viva

Na literatura existem diversas definições sobre o que é um agente. Para Russell (2003), um agente é uma entidade que pode ser percebida pelo seu ambiente e agir sobre ele, ou ainda uma entidade que apresente algumas características da inteligência humana, funcionando contínua e autonomamente em um ambiente. Para Bradshaw (1997), agente pode ser definido por um processo autônomo e direcionado a metas. Em outra definição, de Gilbert *et al.* (1995), mais próxima do paradigma computacional, "*Intelligent agents are software entities that carry out some set of operations on behalf of an user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires*"<sup>6</sup>.

No conceito geral, a função dos agentes é mapear uma sequência de percepções em ações, como podemos ver na Figura 2.3, que mostra a arquitetura básica de agentes em sistemas computacionais.

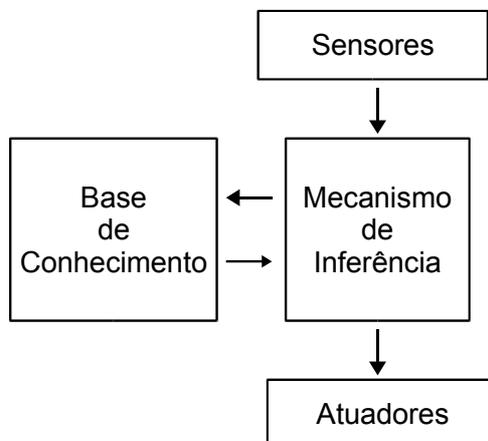


FIGURA 2.3 – Arquitetura básica de um agente para sistemas computacionais.

As setas significam o fluxo de informações. Adaptado de Davidsson (1992).

Dentro do contexto da Base Viva, os agentes devem ser módulos independentes de software, com maior ou menor autonomia e inteligência, desenvolvidos para atuar como

<sup>6</sup> Agentes inteligentes são entidades de software que possuem um conjunto de operações em benefício de um usuário ou outro programa, possuindo algum grau de independência ou autonomia e que, de alguma forma, empregam algum conhecimento ou a representação dos desejos e anseios dos usuários.

interface entre os usuários e a base de informações, como apresentado na Figura 2.4.

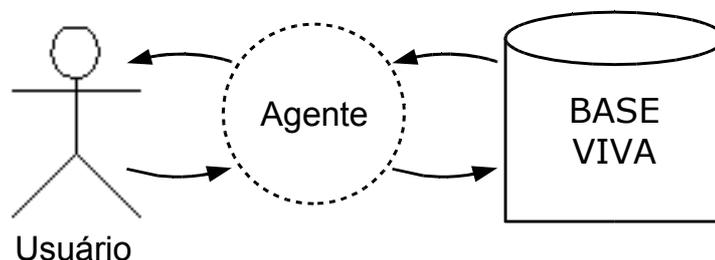


FIGURA 2.4 – O agente como interface entre os usuários e a Base Viva

Para tanto, os agentes, que são o elo de ligação entre os usuários e a base de informações, devem possuir duas capacidades básicas: (1) a capacidade de atuar de forma ativa e/ou passiva sobre a base de informações, o que, na prática, significa a habilidade para fazer operações de consulta, inserção ou modificação nas informações da base; (2) possuir uma interface por onde seja possível estabelecer comunicação com os usuários.

Como já estabelecido anteriormente, no contexto da Base Viva, em que o agente é um software, é necessário definir o conjunto básico de mecanismos que, se implementado neste módulo independente de software, confira ao mesmo a capacidade de se comunicar com os usuários e atuar sobre a base de informações. Para cumprir este objetivo, esta discussão será iniciada pela definição da solução para atuação sobre as informações, seguida da definição dos mecanismos de interação com os usuários.

### 2.3.1 Interação agente – base de informações

Para gerenciar uma base de informações como a Base Viva, que possui conteúdo bastante dinâmico e, ao mesmo tempo, uma elevada taxa de interações ativas e passivas, e, muitas vezes, executadas de forma simultânea, é preciso um mecanismo robusto, que possua recursos para lidar com estas questões, com pouco, ou, preferencialmente, nenhum risco de perda de informações ou de degradação da performance e estabilidade do sistema.

Dentre os atuais paradigmas de sistemas de informação, o mecanismo que se apresenta

como solução para estas questões é o Sistema Gerenciador de Bancos de Dados (SGDB). Isto porque este tipo de sistema se propõe a gerenciar grandes volumes de informação, com eficiência performática, possuindo recursos para sua modelagem lógica, esquemas de segurança contra falhas e acesso indevido e a possibilidade de se fazer acessos simultâneos e de forma remota (Pressman, 2002). Em relação ao acesso remoto, este recurso é importante para conferir maior autonomia aos agentes, uma vez que os mesmos, poderão estar em estruturas independentes, do ponto de vista lógico e físico, da base de informações.

Em meio às várias alternativas de SGBDs disponíveis, estão o Oracle<sup>7</sup>, o MS SQL Server<sup>8</sup>, o MySQL<sup>9</sup> e o PostgreSQL<sup>10</sup>. Em particular, no contexto da Base Viva, é necessário que, de algum modo, o SGBD comporte informações espaciais, tais como os mapas das trilhas, que foram mencionados no início deste Capítulo. Alguns SGBDs, como o Oracle, em seu módulo Spatial, e o Postgres, em seu módulo PostGIS (Ferreira, *et. al.*,2002), já possuem esta habilidade de lidar com informações espaciais. Para os demais, é possível complementá-los com uma camada adicional de software que, entre outras coisas, possa conferir esta capacidade ao sistema. Um exemplo desta camada de software é a TerraLib, mencionada com mais detalhes no Capítulo 3.

Neste cenário, com o uso do SGBD na Base Viva, os agentes, para serem capazes de atuar sobre a base de informações, devem ser complementados por uma pequena camada de software, denominada *driver*, ou, em português, pseudocontrolador (Pressman, 2002), que implementa os protocolos de comunicação com o SGBD. Assim, para executar uma ação sobre a base de informações, seja ela consulta, inserção, alteração ou remoção de informações, o agente deverá executar um dos métodos disponibilizados pelo *driver*.

Outra alternativa possível, e que será explorada nos capítulos seguintes, é que como os agentes podem se comunicar com outros agentes, podemos utilizar um conjunto reduzido de agentes que carregam estes *drivers*, e, portanto podem atuar diretamente

---

7 <http://www.oracle.com/database>

8 <http://www.microsoft.com/sql>

9 <http://www.mysql.com>

10 <http://www.postgresql.org>

sobre as informações com o auxílio do SGBD. Os demais agentes, quando necessitarem de interação com as informações, vão se comportar como usuários dos primeiros, solicitando a estes as operações sobre a base de informações. Os agentes que acessam diretamente o SGBD devem, preferencialmente, disponibilizar funções genéricas através de suas interfaces para, assim, ampliar as possibilidades de interação com a base de informações pelos demais agentes. A principal vantagem desta abordagem é que, exceto pelos agentes que fazem uso dos drivers para acesso ao SGBD, todos os demais independem do sistema escolhido, e, assim, podem atuar até mesmo simultaneamente em mais de um SGBD. A Figura 2.5 ilustra esta situação, onde é possível observar que o agente que se comunica com o usuário, agente “C”, se comunica com o Agente “A”, o qual efetiva as operações sobre a base de informações.

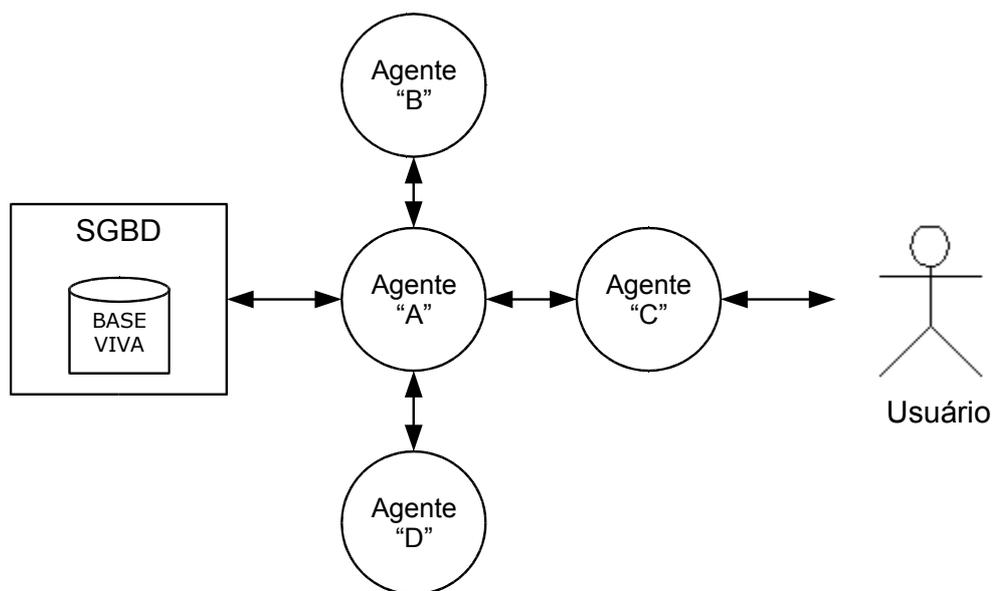


FIGURA 2.5 – Composição de Agentes

Agora que já foi definido o mecanismo de gerenciamento das informações, bem como a maneira dos agentes atuarem sobre este mecanismo, é necessário definir como se dará a comunicação entre os agentes e os usuários da Base Viva.

### 2.3.2 Interação agente – usuário

Uma das premissas estabelecidas para a Base Viva é a alta interatividade e, como foi visto anteriormente, os usuários que iniciam as interações podem não somente ser

peessoas, como também outros sistemas ou dispositivos. Portanto, os meios de comunicação da plataforma onde a Base Viva estará inserida não devem somente suportar um grande volume de transações, mas também uma grande quantidade de tipos de interação.

Entre o que há de mais recente em sistemas de informação, a plataforma *web* parece ser a que mais se aproxima destes requisitos. Isto porque a *web* confere aos sistemas de informação que a utilizam como plataforma de comunicação, os benefícios da flexibilidade, maior acessibilidade e menores riscos de obsolescência e isolamento (Anderson, Moreno-Sanchez, 2003). Os principais motivos para isso são o fato de que a Web está cada vez mais difundida geográfica e socialmente, além de ser compatível com a maioria dos dispositivos computacionais existentes e em desenvolvimento, participantes de algum processo remoto de troca de informações. Porém, a Web é apenas a plataforma de comunicação através da qual os agentes se comunicarão com os usuários. Ainda é necessário definir como os agentes serão vistos e localizados pelos usuários.

Ao refletir sobre esta arquitetura que está sendo proposta, pode-se concluir que é possível dois elementos de software (agentes) trocarem mensagens de forma remota, seja para executar uma inserção ou consulta na base de dados, ou para qualquer outro tipo de finalidade. A troca de mensagens entre elementos de software, feita remotamente, é a característica essencial dos sistemas distribuídos (Pressman, 2002). Portanto se faz necessário encontrar, dentre as tecnologias para sistemas distribuídos, uma que mais se adeque às demandas da Base Viva.

### **2.3.3 Agentes implementados como *Web Services***

Para que a Base Viva cumpra suas finalidades de possibilitar a interação do maior número possível de usuários, é preciso que a plataforma em que ela esteja inserida use padrões e protocolos abertos, para permitir a construção de ferramentas de acesso à base sem a necessidade de utilização de um software específico deste ou daquele fabricante. Além disso, é desejável que seja uma tecnologia já difundida e bem aceita, para reduzir os riscos de isolamento. Dentre as tecnologias disponíveis para sistemas distribuídos -

DCOM, Corba, RMI e Web Services (Gunzer, 2002) -, a tecnologia de *web services* é a que parece mais adequada. Isto porque, como será visto no Capítulo 3, os *web services* são exatamente aqueles que implementam somente padrões e protocolos abertos, e que estão ganhando destaque no mercado em função dos produtos da maioria dos fabricantes de plataformas e linguagens de programação já estarem adequados a este esquema de funcionamento.

Portanto, seguindo esta linha de raciocínio, os agentes atrelados à Base Viva deveriam ser disponibilizados através de serviços, ou *web services*, os quais devem disponibilizar mecanismos para atuação sobre um base de informações com conteúdo geográfico. Em relação à disseminação de informações geográficas através de *web services*, percebe-se que esta parece ser uma forte tendência observada por diversos autores, como Hagel e Brown (2002), Parsons (2003), Sonnet (2004) e Davis Jr. e Alves, (2005).

No próximo capítulo, uma proposta de implementação para esta arquitetura, o Terra WebServices, é apresentada. Não são discutidos detalhes relativos ao modelo de dados para a Base Viva. Por mais bem elaborado que seja, é natural que o modelo de dados se altere em função das demandas que podem surgir com a utilização do sistema e/ou da dinâmica das relações entre as entidades do mundo real representadas pelo modelo (Chen, *et al.*, 1999). No contexto desta dissertação não se discutem técnicas de modelagem e alternativas de modelagem dinâmica para bases de dados geográficas. [Breu e Ding (2004), Chen *et al.* (1999).]

## **2.5 Comentários finais**

Resumidamente, de acordo com as premissas estabelecidas, é possível concluir que, para suportar as condições de alta interatividade sobre a base de informações a partir de um múltiplo e diverso grupo de usuários, o mais indicado para a Base Viva é possuir um mecanismo de gerenciamento da base de dados, SGBD, com recursos de armazenamento e recuperação de informações espaciais. As atuações sobre este mecanismo devem ser executadas por agentes, no formato de *web services*, os quais devem atender aos requisitos e solicitações dos usuários. A base de informações deve possuir uma separação interna entre as informações fornecidas e alteradas pelos

usuários e as informações consolidadas. Esta separação das informações pode ou não ser visível para os usuários. A Figura 2.6 ilustra o cenário construído a partir destas conclusões.

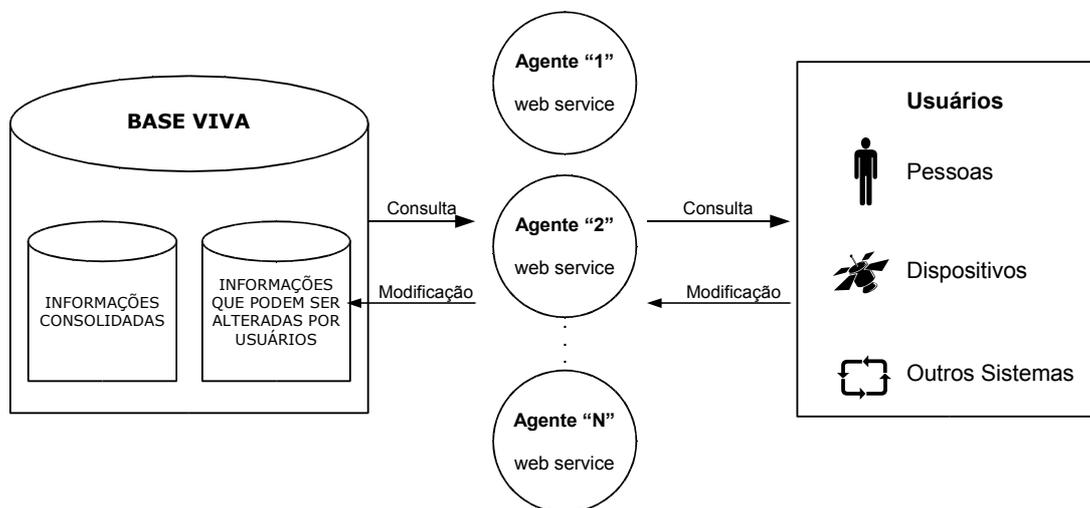


FIGURA 2.6 – Entidades envolvidas na Base Viva

## CAPÍTULO 3

### TERRA WEBSERVICES

#### 3.1 A tecnologia de *web services*

Nos últimos anos, uma nova forma de utilização da internet, chamada de *web services*, tem ganhado muitos adeptos e está se tornando o foco das políticas de Tecnologia da Informação (TI) de um grande número de organizações, independentemente de seu porte e ramo de atividade. Esta nova abordagem possui uma característica curiosa. Diferentemente dos *web sites* tradicionais, projetados para as pessoas interagirem com informações, os *web services* conectam sistemas com outros sistemas. E fazem isso através de ligações livres, o que permite que as conexões sejam estabelecidas entre diferentes sistemas, sem a necessidade de adaptações. Como resultado, as conexões podem ser estabelecidas não importando quais são as plataformas e linguagens de programação utilizadas nos sistemas envolvidos, o que torna muito mais fácil compartilhar recursos tecnológicos para uma enorme gama de usuários e organizações (Hagel e Brown, 2002).

A base tecnológica dos *web services* são diversas especificações do *World Wide Web Consortium* (W3C), às quais a grande maioria dos fabricantes de plataformas e linguagens de programação já adequaram seus produtos. Em especial, por serem os líderes de mercado em plataformas de sistemas de última geração, as plataformas Java EE<sup>11</sup> (Java Platform, Enterprise Edition), da Sun Microsystems<sup>12</sup>, e Microsoft .NET<sup>13</sup>, da Microsoft Corporation<sup>14</sup>, ao recomendarem fortemente a utilização de *web services*, estão direcionando os sistemas corporativos para a utilização extensa desta tecnologia.

Como opções de tecnologias de sistemas distribuídos, além dos *web services*, podemos

---

11 <http://java.sun.com/javaee>

12 <http://www.sun.com>

13 <http://www.microsoft.com/net>

14 <http://www.microsoft.com>

citar também o OMG Corba<sup>15</sup>, Java RMI<sup>16</sup> e Microsoft DCOM<sup>17</sup>. Em relação ao RMI e ao DCOM, tecnologias da Sun e Microsoft, respectivamente, específicas para as plataformas destes fabricantes, os *web services* tem a vantagem de poderem ser utilizados praticamente em qualquer plataforma. Já em relação ao Corba, que é uma abrangente e complexa especificação da OMG<sup>18</sup> (Object Management Group), embora os *web services* utilizem vários conceitos desta, são muito mais simples, principalmente por utilizarem protocolos e padrões abertos e já largamente adotados – os protocolos TCP/IP e HTTP e o padrão XML – que já estavam estabelecidos nas plataformas, sendo bastante conhecidos pelos programadores e utilizados em grande parte dos sistemas. Assim, os conhecimentos a serem adquiridos pelos programadores e as implementações a serem executadas nas plataformas para se adequarem à tecnologia de *web services* são muito menores se comparados ao Corba.

A Figura 3.1 mostra o desenho da arquitetura da tecnologia de *web services*, onde são exibidos os padrões e protocolos envolvidos.

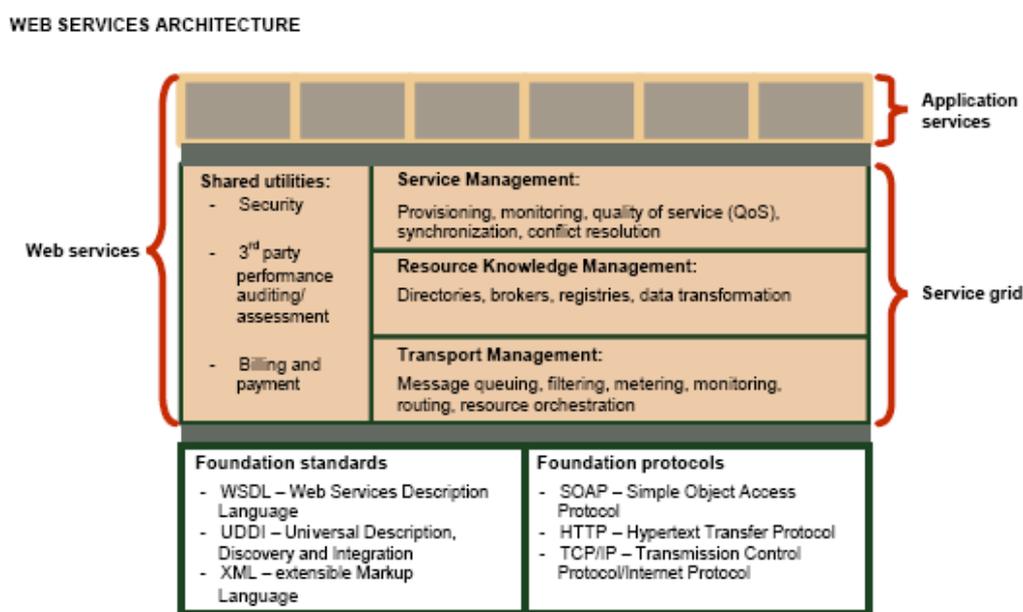


FIGURA 3.1 – Arquitetura da tecnologia de Web Services (Hagel e Brown, 2002)

15 <http://www.omg.org/corba/>

16 <http://java.sun.com/products/jdk/rmi/>

17 <http://www.microsoft.com/com>

18 <http://www.omg.org/>

Tendo em vista que este trabalho tem por objetivo apresentar uma arquitetura de sistema para o domínio geográfico, torna-se necessário saber quais padrões já estão estabelecidos para este domínio de aplicação, o que será discutido no tópico a seguir.

### 3.2 Web Services para o domínio geográfico

Ao mencionar sistemas em ambiente distribuído e, sobretudo, multi-plataforma, existe uma palavra que sintetiza as demandas para os componentes destes sistema: **interoperabilidade**. Interoperabilidade é a capacidade de dois sistemas computacionais, de tipos diferentes, operarem colaborativamente através de algum meio de comunicação.

A interoperabilidade de dados do domínio geográfico, ou dados espaciais, é de tamanha importância que, em 1994, foi criado o Open Geospatial Consortium<sup>19</sup> (OGC) – antigamente conhecido por Open GIS Consortium - um organismo internacional, sem fins lucrativos, formado por representantes de governos, universidades e empresas, ligados de forma direta ou indireta ao geoprocessamento, com o objetivo de criar especificações para o intercâmbio de dados espaciais.

De acordo com Câmara *et al.* (1999) e Lima *et al.* (2002), algumas das especificações do OGC, do ponto de vista semântico, não se adequam totalmente à realidade dos softwares de geoprocessamento. Apesar disto uma parte considerável destas especificações já foram implementadas pelos principais fabricantes de softwares de geoprocessamento<sup>20</sup>. Dentre elas estão a *Simple Feature Specification*, que define o modelo dos objetos espaciais como pontos, linhas, regiões e multi-pontos; o GML, ou *Geography Markup Language*, que é uma especificação que define padrões para transporte de dados espaciais sob o padrão XML; o *Web Map Service*, que define um *web service* de geração de mapas em formato de imagens; o *Web Coverage Service* que define um *web service* para o provimento de informações espaciais de geocampos; e o *Web Feature Service* que define um *web service* para a consulta e edição de dados espaciais através da troca de dados no padrão GML.

---

19 <http://www.opengeospatial.org>

20 <http://www.opengeospatial.org/resources/?page=products>

Contudo, a criação do OGC e de muitas de suas especificações precederam algumas das especificações do W3C, que se tornaram padrão de mercado, sobretudo no que diz respeito aos *web services*. Sendo assim, várias especificações do OGC poderiam ser consideradas fora do padrão de mercado. Por exemplo, ao invés de usar o UDDI, o OGC propõe o uso de um serviço de catálogo. Além disso, o OGC propõe uma interface de conexão específica, sem a utilização de descritores para os serviços (Davis Jr. e Alves, 2005).

Para transpor estes obstáculos, em relação aos *web services*, o OGC criou o OWS - *OpenGis Web Services Initiative* - uma iniciativa visando conduzir as especificações pertinentes do OGC em direção às especificações do W3C. Os primeiros trabalhos desta comissão foram publicados em 2004 (Sonnet, 2004), definindo o escopo das mudanças, e em novembro de 2005, boa parte destes trabalhos foram concluídos (Whiteside, 2005).

Ao projetar um sistema com a pretensão de ser adequado aos padrões de mercado em sistemas de informação geográfica, não se deve desconsiderar os trabalhos do OGC (Kralidis, 2004). Tendo em vista esta constatação, posteriormente à arquitetura proposta para o Terra WebServices, apresentada a seguir, será feita uma breve explanação sobre as possibilidades implantação pelo Terra WebServices dos *web services* especificados pelo OGC.

### **3.3 Arquitetura do Terra WebServices**

O Terra WebServices foi concebido como uma solução tecnológica para um sistema de alta interatividade sobre uma base de informações com conteúdo espacial, cujos requisitos básicos, estabelecidos no Capítulo 2, são: ser um sistema de agentes distribuídos implementado através de *web services*, nos padrões W3C; o repositório de dados deve ser gerenciado por SGBD; o modelo de dados deve prever uma separação lógica interna entre dados consolidados e informações passíveis de alteração pelos usuários; e deve suportar informações espaço-temporais.

Como elementos principais da arquitetura proposta, estão a biblioteca TerraLib, a

plataforma de *web services* Apache Axis e o servidor *web* Apache HTTP Server, e um SGBD, descritos a seguir na Figura 3.2.

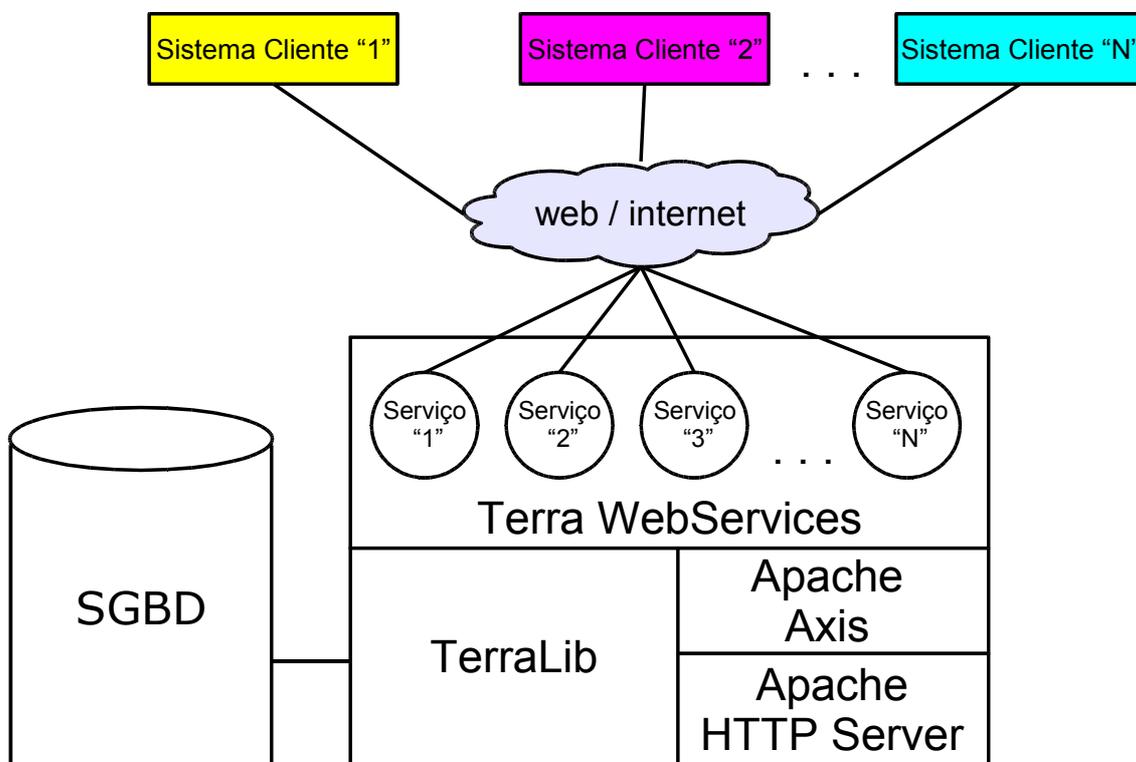


FIGURA 3.2 – Arquitetura do Terra WebServices

### 3.3.1 TerraLib

A biblioteca TerraLib<sup>21</sup> é um projeto de software livre, com base científica, que permite o trabalho colaborativo entre a comunidade de desenvolvimento de aplicações geográficas (Vinhas e Ferreira, 2005). A TerraLib foi concebida para ser uma biblioteca de funcionalidades geográficas, inovadora em relação aos produtos encontrados no mercado, pelo fato de conter estruturas e funcionalidades no estado-da-arte em recursos de geoprocessamento e por estar em permanente expansão, já que vem sendo ampliada por provas de conceito de diversas pesquisas nesta área.

A TerraLib é uma biblioteca de classes escrita em C++ (Vinhas e Ferreira, 2005) que, dentre as linguagens de alto nível, é a de melhor performance para a maioria das aplicações, razão pela qual é a linguagem grande parte dos *softwares desktop*<sup>22</sup> foram

21 <http://www.terralib.org>

22 Programa de computador projetado para execução em modo *stand alone* ou cliente/servidor

concebidos. Contudo, por ser relativamente complexa e pouco adaptada ao ambiente *web*, apenas uma pequena parte dos desenvolvedores de aplicativos está habituada a ela.

Deste modo, surge um desafio: como disponibilizar as funcionalidades desta biblioteca para outras linguagens. Algumas iniciativas foram feitas objetivando a criação de projetos de software para a integração da TerraLib com outras linguagens de programação, como, por exemplo o TerraPHP<sup>23</sup>, que integra a TerraLib com o PHP; e o Terra SDK que disponibiliza a TerraLib no formato de OCX, o que a torna facilmente adaptável aos sistemas desenvolvidos para Windows compatíveis em Visual Basic, Delphi e outras linguagens de programação. Estas abordagens, de fato, elevam a interoperabilidade da TerraLib, porém ainda esbarram nos desafios de compilação por plataforma e nas limitações da linguagem para a qual se fez a ponte.

Face às deficiências encontradas nos sistemas existentes, surge mais uma motivação para a disposição das funcionalidades da TerraLib através do Terra WebServices.

### 3.3.2 Apache Axis

A Apache Software Foundation<sup>24</sup> é provavelmente a organização de gerenciamento de projetos de software livre de maior renome internacional. Vários projetos de software geridos pela Apache são padrões em distribuições do Linux, e até mesmo são utilizados como base para projetos de software comercial.

Dentre as centenas de projetos geridos pela Apache, o Axis<sup>25</sup> é a plataforma mais completa para a publicação de *web services*. O Axis conta com as versões Java e C++, sendo que, em ambas as versões, é necessário a utilização associada a um servidor HTTP. Na versão C++, o mais indicado é o Apache HTTP Server, e na versão Java é possível utilizar o Apache Coyote HTTP, que acompanha o servidor de aplicações java Apache Tomcat<sup>26</sup>, sobre o qual o Axis Java pode ser executado.

O principal objetivo da plataforma Axis é permitir a simples e rápida disponibilização

---

23 [http://www.dpi.inpe.br/terralib/docs/v310/terraphp\\_tutorial\\_portugues.html](http://www.dpi.inpe.br/terralib/docs/v310/terraphp_tutorial_portugues.html)

24 <http://www.apache.org>

25 <http://ws.apache.org/axis>

26 <http://tomcat.apache.org>

de funcionalidades de uma biblioteca, seja em Java, ou em C++, como a TerraLib, através de *web services*. Na prática, o Axis é que implementa o protocolo SOAP e, portanto, é o responsável pela formatação e interpretação das mensagens trocadas com os sistemas cliente.

### **3.3.3 Apache HTTP Server**

O Apache HTTP Server é o principal, e talvez o mais antigo, projeto de software aberto gerido pela Apache Foundation. É utilizado pela maioria dos sistemas *web* baseados em servidores Linux e Unix, e por parte dos sistemas baseados em servidores Windows.

### **3.3.4 SGBD**

O SGBD, ou Sistema Gerenciador de Banco de Dados, cuja definição mais detalhada pode ser encontrada no item 2.3.1 do Capítulo 2, é o responsável pelo gerenciamento dos dados. Pelo fato da comunicação com o SGBD ser feita através da TerraLib, a plataforma Terra WebServices é compatível com todos os SGBDs com os quais a TerraLib estiver preparada para se conectar. Dentre os SGBD livres e de código aberto com os quais a TerraLib se conecta podemos citar o MySQL e o PostgreSQL.

## **3.4 Contexto tecnológico**

Nesta proposta, os quatro elementos centrais da arquitetura do Terra WebServices – TerraLib, ApacheAxis, Apache HTTP Server e os módulos de *web services* - são desenvolvidos em C++. Isto, que pode ser visto como uma restrição, na realidade, permite simplificar o desenvolvimento, através de uso de uma única plataforma de desenvolvimento que produz objetos executáveis bastante otimizados e potencialmente estáveis. Mesmo com esta imposição, é possível ter bastante liberdade de escolha para os demais elementos, como o sistema operacional, o SGBD e os sistemas cliente, como pode ser visto na Figura 3.3.

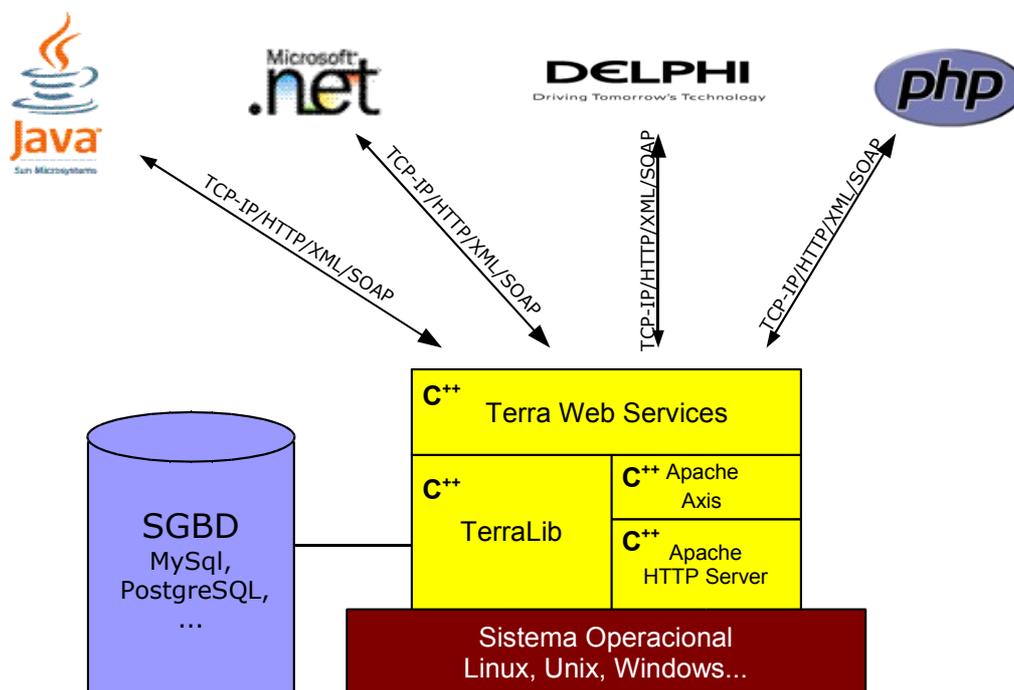


FIGURA 3.3 – Contexto tecnológico do Terra WebServices

A principal característica que permite a alta interatividade desta solução é a grande quantidade de sistemas cliente compatíveis. Além dos inúmeros sistemas *desktop* ou *web* desenvolvidos em Java, .Net, Delphi, Php e em outras linguagens compatíveis com *web services*, vários dispositivos móveis, como PDAs<sup>27</sup> e celulares, que atualmente estão sendo produzidos e comercializados em escala industrial, já executam Java e .NET, ampliando mais ainda a gama de possíveis usuários para o Terra WebServices.

Não existem limites, dentro do contexto de SIG, para os *web services* que podem ser disponibilizados através desta arquitetura. O importante é a sólida base tecnológica para construção destes serviços: a biblioteca TerraLib com recursos no estado-da-arte em geoprocessamento; diversos opções de SGBDs livres e comerciais; e a plataforma Apache Axis que, por ser gerida pela Apache Foundation, provavelmente estará sempre à frente das implementações de novas versões das especificações da W3C para *web services*. Com estes recursos podem ser desenvolvidos *web services* muito simples, como os que serão propostos a seguir, outros mais elaborados, como os *web services* especificados pelo iniciativa OWS do OGC, e outros inovadores como o que será

<sup>27</sup> Personal Digital Assistant – Pequeno computador portátil

proposto mais adiante, no Capítulo 5.

### **3.5 Propostas de serviços para o Terra WebServices**

Embora existam propostas sobre quais são as operações básicas dos SIGs, por exemplo relativas aos operadores espaciais descritos por Queiroz (2003), não existe um padrão para a implementação destes conceitos em bibliotecas que dão suporte para a construção de SIG, como a TerraLib, que pudesse ser seguido no Terra WebServices. Portanto, durante a elaboração deste trabalho, foi necessário fazer uma reflexão sobre quais funções básicas o Terra WebServices poderia prover.

O primeiro passo foi identificar as categorias de funções utilizadas pelos SIG de acordo com suas interfaces e objetivos. Por exemplo, funções de persistência, que são reponsáveis pelo armazenamento e recuperação de informações, e funções de operações espaciais. Após esta categorização, apresentada a seguir, foram elencadas algumas das funções de cada uma das categorias.

**a) Armazenamento e Recuperação:** os sistemas de informação, em geral, possuem algum repositório de dados e, portanto, possuem mecanismos para armazenar, alterar e recuperar as informações do repositório.

**b) Metadados:** alguns SIG, principalmente aqueles que operam em conjunto a SGBD, possuem um serviço de metadados, que tem por objetivo fornecer informações sobre os dados armazenados em seu repositório, como, por exemplo, os nomes e as estruturas das tabelas existentes.

**c) Renderização:** provavelmente o elemento mais característico dos SIGs é a apresentação de mapas. Como os dados podem estar armazenados em formato vetorial, é necessário um processo para transformá-los em imagens. Este processo é conhecido pelo nome de renderização.

**d) Consultas Espaciais:** as consultas espaciais talvez sejam o elemento mais elementar dos SIG. Através delas é que se pode obter os apontadores necessários para a recuperação das informações, baseada em critérios espaciais, tais como relações

topológicas.

**e) Operações sobre dados espaciais:** estas funcionalidades, muito comuns em SIGs onde são geradas informações espaciais, destinam-se à criação de objetos espaciais a partir de operações sobre outros objetos espaciais, tais como a geração de *buffer*.

A Tabela 4.1 apresenta as funções identificadas dentro desta categorização. Como sugestão para a implementação, cada categoria seria um *web service* e as funções listadas seriam os métodos deste *web service*.

<b>Grupo de Recursos</b>	<b>Método</b>	<b>Descrição</b>
Metadados	getLayers	Obtenção da identificação das tabelas de conteúdo geográfico disponíveis na base
	getAttributeTables	Obtenção da identificação das tabelas de atributos associados à uma tabela com conteúdo geográfico.
	getTableFields	Obtenção da identificação dos campos de informação disponíveis em uma tabela de atributos
Consultas Espaciais	searchWithinBox	Obtenção da identificação dos objetos de uma tabela cuja geometria está, total ou parcialmente, contida dentro de uma área retangular definida por um box
	searchAtPoint	Obtenção da identificação dos objetos de uma tabela cuja geometria intercepta a geometria de um ponto
	searchWithinDistance	Obtenção da identificação dos objetos de uma tabela cuja geometria está, total ou parcialmente, contida dentro de uma área circular definida por uma coordenada geográfica e uma distância
	searchWithinFeature	Obtenção da identificação dos objetos de uma tabela cuja geometria está, total ou parcialmente, contida dentro da área definida por um objeto geográfico
Armazenamento e Recuperação	getQueryResult	Obtenção da identificação dos objetos de uma tabela filtrados por um critério baseado em seus atributos

	getObjectAttributes	Obtenção dos atributos associados a um objeto
	getWKT	Obtenção da geometria de um objeto no formato <i>Well Known Text</i> (OGC)
	insertObject	Inserção de um objeto em uma tabela com conteúdo geográfico
	deleteObject	Remoção de um objeto
	updateGeometry	Alteração da geometria de um objeto
	updateAttribute	Alteração de um atributo
Operações sobre dados espaciais	getBox	Obtenção da geometria do mínimo box que contenha todos os objetos de um layer ou mais layers
	getBoxWidth	Obtenção da largura de um box
	getBoxHeight	Obtenção da altura de um box
	processBox	Obtenção de um box
	getBuffer	Obtenção da geometria determinada pela operação de buffer em outra geometria
Renderização	getMapImage	Obtenção da imagem do mapa de uma área definida por um box, obtida através da renderização dos objetos de layers também determinados
	getClickCoords	Obtenção da coordenada geográfica associada a um ponto de uma imagem renderizada

Tabela 4.1 – Proposta de web services e seus métodos para o Terra WebServices

Os métodos apresentados, que são relativamente simples, também poderiam ser combinados para se obter funcionalidades mais sofisticadas, como os dois exemplos a seguir:

**Exemplo 1:** A renderização de um mapa contendo todos os objetos de todos os *layers* da base poderia ser executada em três etapas:

- 1) Método *getLayers* para obter a identificação de todos os *layers* da base;
- 2) Método *getBox* para obter o mínimo box contendo os *layers* encontrados no método anterior;

- 3) Método `getMapImage` para obter a imagem renderizada da área do box e contendo os *layers*, obtidos pelos métodos anteriores.

**Exemplo 2:** Obtenção das informações dos objetos cuja geometria intercepta um ponto do mapa do Exemplo 1, definido pelo clique do usuário na imagem:

- 1) Método `getClickCoords` para obter a coordenada geográfica do clique do usuário no mapa;
- 2) Método `searchAtPoint` executado para cada um dos *layers* utilizados na renderização, para a obtenção da identificação dos objetos;
- 3) Método `getAttributeTables` para cada *layer* onde algum objeto foi obtido pelo método anterior;
- 4) Método `getTableFields` executado para cada tabela de atributos obtida pelo passo anterior;
- 5) Método `getObjectAttributes` executado para cada objeto encontrado e cada tabela obtida pelos passos 2 e 3, solicitando os atributos obtidos pelo passo 4;

Com base nestes exemplos é possível concluir que é viável a construção de sistemas com funcionalidades bastante sofisticadas, operando sobre os Terra WebServices. Neste ponto cabe discutir sobre a aderência do Terra WebServices às especificações do OGC.

### **3.6 Aderência do Terra WebServices às especificações do OGC**

Como já constatado anteriormente, a arquitetura Terra WebServices contém base tecnológica suficiente para comportar serviços em inúmeros contextos, dentre eles os *web services* nas especificações do OGC. Uma vez implementado um serviço de acordo com uma destas especificações, o sistema poderá ser considerado aderente àquela especificação.

Para implementar um *web service* nestas condições, deve-se consultar os documentos da

especificação, para saber quais são os métodos necessários e opcionais do serviço, e quais os padrões de invocação e resposta para estes métodos. A seguir, é apresentado um exemplo deste processo de consulta para um dos serviços OGC, o WMS.

O WMS ou *Web Map Service* é um *web service* que tem por objetivo retornar mapas em formato de imagens. Para obter a imagem, o sistema cliente precisa invocar um dos métodos do serviço, cujo nome é GetMap, passando parâmetros como a região geográfica do mapa (*box*), a projeção cartográfica, os *layers* que devem ser plotados e seus estilos de representação, e o formato desejado para a imagem resultante. Porém, este cliente deve ser informado anteriormente sobre quais as opções disponíveis no serviço, tais como os sistemas de projeção, *layers* e formatos de imagem. Para tanto, existe um outro método, de nome GetCapabilities que, quando invocado, fornece um documento no formato XML contendo estas informações.

A Figura 3.4 e 3.5, obtidas das mais recentes especificações do OGC, contém os parâmetros do métodos GetCapabilities e GetMap, respectivamente.

Name <sup>a</sup>	Definition	Data type and value	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., "WMS", "WFS")	One (mandatory)
request	Operation name	Character String type, not empty Value is operation name (e.g., "GetCapabilities")	One (mandatory)
Accept Versions	Prioritized sequence of one or more specification versions accepted by client, with preferred versions listed first	Sequence of Character String type, each not empty Value is list of x.y.z "version" values	Zero or one (optional) When omitted, return latest supported version (see Subclause 7.3.2)
Sections	Unordered list of zero or more names of requested sections in complete service metadata document <sup>b</sup>	Sequence of Character String type, each not empty Value is list of section names Allowed section names are specified by each Implementation Specification	Zero or one (optional) When omitted or not supported by server, return complete service metadata document
update Sequence	Service metadata document version, value is "increased" whenever any change is made in complete service metadata document	Character String type, not empty Values are selected by each server, and are always opaque to clients	Zero or one (optional) When omitted or not supported by server, return latest service metadata document
Accept Formats	Prioritized sequence of zero or more response formats desired by client, with preferred formats listed first	Sequence of Character String type, each not empty Value is list of format identifiers Identifiers are MIME types of formats useful for service metadata documents	Zero or one (optional) When omitted or not supported by server, return service metadata document using MIME type "text/xml"
<p><sup>a</sup> Although some values listed in the "Name" column appear to contain spaces, they shall not contain spaces.</p> <p><sup>b</sup> The "Sections" parameter specifies which XML elements within a service metadata document shall be returned, within an (usually abbreviated) "Capabilities" element. The allowed section name values shall be specified by each Implementation Specification, as specified in Subclause 7.3.3.</p>			

FIGURA 3.4 – Parâmetros do Método GetCapabilities. Adaptado de Whiteside (2005).

O método apresentado acima retorna as informações necessárias para a invocação dos serviços WMS, WFS e WCS [(Buehler, 2003), (Lieberman, 2003),(Mabrouk, 2004)]. O retorno dele é um documento XML, cujo padrão pode ser encontrado na especificação *OpenGIS® Web Services Common Specification*, editada por Whiteside (2005).

Request Parameter	Mandatory/Optional	Description
VERSION=1.3.0	M	Request version.
REQUEST=GetMap	M	Request name.
LAYERS=layer_list	M	Comma-separated list of one or more map layers.
STYLES=style_list	M	Comma-separated list of one rendering style per requested layer.
CRS=namespace:identifier	M	Coordinate reference system.
BBOX=minx,miny,maxx,maxy	M	Bounding box corners (lower left, upper right) in CRS units.
WIDTH=output_width	M	Width in pixels of map picture.
HEIGHT=output_height	M	Height in pixels of map picture.
FORMAT=output_format	M	Output format of map.
TRANSPARENT=TRUE FALSE	O	Background transparency of map (default=FALSE).
BGCOLOR=color_value	O	Hexadecimal red-green-blue color value for the background color (default=0xFFFFFF).
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=XML).
TIME=time	O	Time value of layer desired.
ELEVATION=elevation	O	Elevation of layer desired.
Other sample dimension(s)	O	Value of other dimensions as appropriate.

FIGURA 3.5 – Parâmetros do método GetMap relativo ao serviço WMS do OGC.

Adaptado de Beaujardiere (2004).

O retorno do método é uma sequência de bytes, cujo padrão de codificação deve seguir o mesmo padrão de codificação do tipo da imagem. Ex: png, gif ou jpeg.

Valendo-se da abordagem de combinação de funcionalidades básicas para compor outras mais complexas, é possível construir os serviços nos padrões OGC baseados nas funcionalidades definidas anteriormente. No caso do serviço WMS, os métodos GetCapabilities e GetMap poderiam ser obtidos pela combinação dos serviços Metadados e Renderização propostos. O serviço de Metadados, através do método getLayers, forneceria as opções de níveis de informação disponíveis, e o serviço Renderização forneceria a imagem do mapa, através do método getMapImage. As informações sobre os sistemas de coordenadas e datum disponíveis poderiam estar embutidas no *web service*, ou até mesmo serem fornecidas por uma extensão do serviço Metadados. Já os formatos de imagens renderizadas poderiam também estar embutidos

no *web service*, ou então serem obtidos por uma extensão do *web service* de renderização. A Figura 3.6 ilustra este cenário para a construção do serviço WMS.

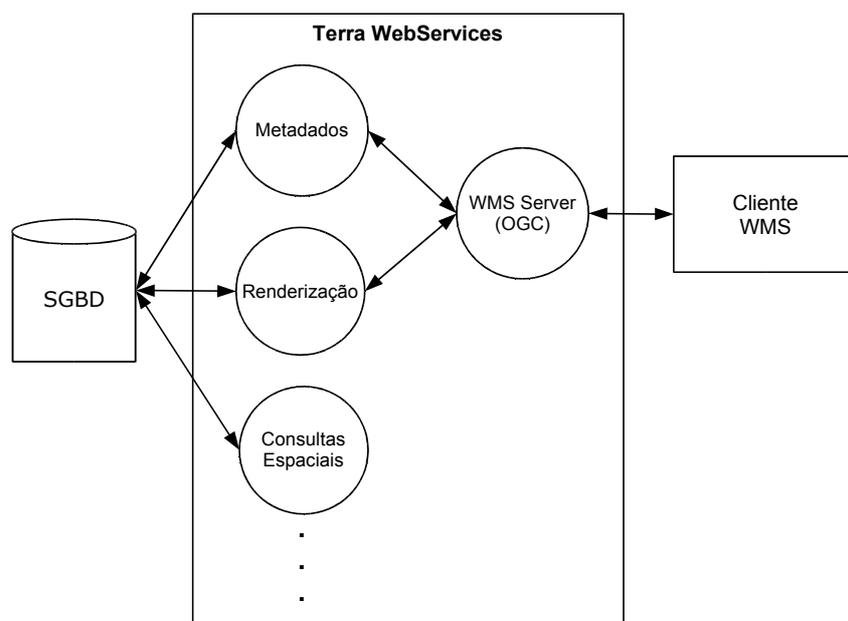


FIGURA 3.6 – Composição de serviços básicos para a construção do serviço WMS especificado pelo OGC

Conforme visto anteriormente, o Terra WebServices pode ser utilizado por diversos tipos de usuários, em contextos bastante diversificados, os quais podem ser obtidos pela extensão do Terra WebServices, e que, portanto, não necessitam ser previstos. Desta forma, essa arquitetura de serviços *web* tem os recursos necessários para construção de sistemas de alta interatividade e adequados às especificações de software predominantes no mercado. Falta apenas saber como, em um cenário de adoção desta tecnologia, este sistema poderia ser escalável, ou seja, ampliado conforme as necessidades de uso.

### 3.7 Escalabilidade do Terra WebServices

É crescente a preocupação com o desenvolvimento de sistemas com alta escalabilidade. Isto porque sistemas com esta característica podem ser mais facilmente estendidos, com o objetivo de melhorar sua performance e/ou estabilidade.

Neste quesito as tecnologias adotadas pelo Terra WebServices demonstram mais uma vantagem. Ao separar os sistemas em camadas, a arquitetura de *web services* confere maior escalabilidade, pois é possível ampliar a quantidade de elementos, *hardware* e *software*, da camada onde as questões de performance ou estabilidade são os limitadores do sistema. A Figura 3.7 mostra as possibilidades de escalabilidade do Terra WebServices.

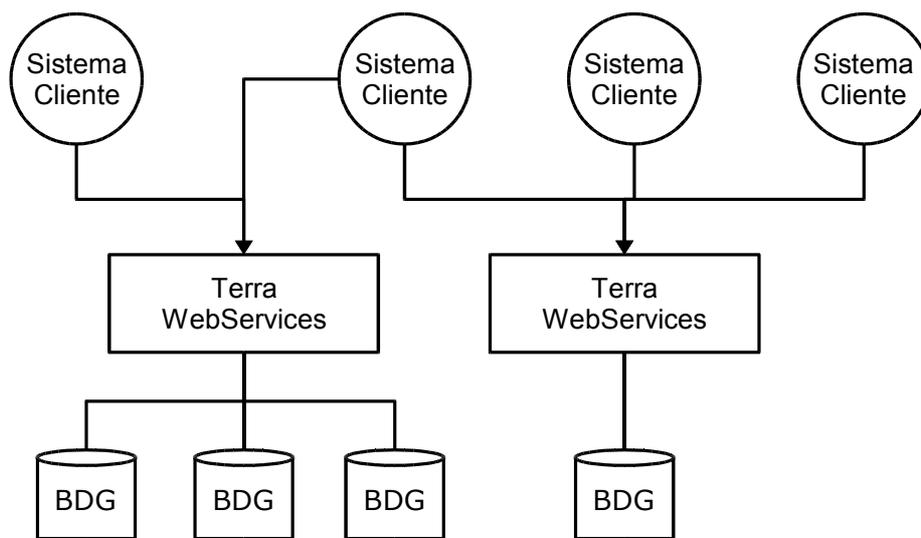


FIGURA 3.7 – Escalabilidade dos Terra WebServices

### 3.8 Comentários finais

Como mostra a Figura 3.5, uma instância do Terra WebServices pode suportar diversos clientes de forma simultânea, o que é muito importante em um cenário de múltiplos usuários. Além disso, esta mesma instância pode se conectar, também de forma simultânea, a mais de um SGBD, que podem até mesmo ser diferentes, o que é importante em contextos em que é necessário acessar mais de uma base de dados, ou quando, para evitar quedas do sistema por falhas no SGBD, este deve operar redundantemente em dois repositórios. Por outro lado, um cliente também pode se conectar a mais de uma instância do Terra WebServices, pelos mesmos motivos de prevenção de falhas, ou nos contextos em que isso é desejado. Por fim, é possível executar as 3 camadas, SGBD, Terra WebServices e o Cliente Terra WebServices, em diferentes computadores, ou em apenas um.

Outro aspecto importante do Terra WebServices é que um único servidor pode conviver

com várias versões do Terra WebServices, bastando para isso, por exemplo, indicar a versão nas URLs<sup>28</sup> dos serviços. Outra vantagem importante é que as alterações na TerraLib e/ou no SGBD podem ter impacto zero para os sistemas cliente.

Agora que já é conhecida a proposta de arquitetura e implementação do Terra WebServices, será apresentado, no próximo capítulo, a proposta de um serviço, estendendo seus serviços básicos, com o objetivo de implementar uma solução de georreferenciamento qualitativo.

---

<sup>28</sup> Uniform Resource Locator ou Localizador Uniforme de Recursos, o esquema utilizado na *web* para localizar uma determinada página ou arquivo

## CAPÍTULO 4

### GEOAGENTE: UM AGENTE PARA O GEORREFERENCIAMENTO QUALITATIVO

#### 4.1 Processos de georreferenciamento

A produção de informações com vínculo espacial envolve a realização do processo de georreferenciamento ou referenciamento geográfico, necessário para atribuição deste vínculo. Este processo pode se dar de diversas maneiras, como ocorre, por exemplo, dentro dos aparelhos de GPS<sup>29</sup>. Nestes equipamentos, as leituras de sensores que captam ondas eletromagnéticas emitidas por um conjunto de satélites em órbita terrestre passam por um mecanismo de inferência, que as transforma nos valores das coordenadas de longitude e latitude. Outro exemplo comum nas atividades de geoprocessamento urbano é a geocodificação, ou a transformação de endereços do modo como são fornecidos pelas pessoas ou armazenados em sistemas de informação, em coordenadas geográficas (Davis Jr., 1998). Um terceiro exemplo ocorre quando uma posição é apontada em um mapa geográfico. Neste caso, pela medida das distâncias entre a referida posição e as referências do mapa, tais como os limites (bordas) ou linhas de grade, é possível calcular as coordenadas do ponto através do processo de interpolação.

Em todas as situações, o georreferenciamento parte de uma posição geográfica conhecida. No caso do GPS, parte-se das posições dos satélites na órbita terrestre no instante da leitura, obtidas por um processo interno baseado na leitura de um relógio de alta precisão e nas informações precisas sobre as órbitas dos satélites, armazenadas pelo equipamento. Já o processo de geocodificação parte de alguma referência geográfica conhecida, como, por exemplo, as coordenadas da rua do endereço. No caso do apontamento no mapa, são utilizadas as informações das coordenadas dos limites do mapa e/ou de referências internas.

---

<sup>29</sup> *Global Positioning System* ou Sistema de Posicionamento Global

Com base na descrição geral do processo de georreferenciamento, complementada pelos exemplos descritos, é possível concluir que este sempre envolve uma **entrada de dados** e um **mecanismo de inferência**, que, apoiado por uma **base de informações**, apresenta um **resultado** na forma de uma referência geográfica. Este é um processo análogo ao mecanismo dos agentes, o que significa dizer que o georreferenciamento é passível de ser representado e executado por agentes. A Figura 4.1 mostra um esquema geral para um agente, e de como podemos definir o processo de georreferenciamento com base nos elementos deste esquema.

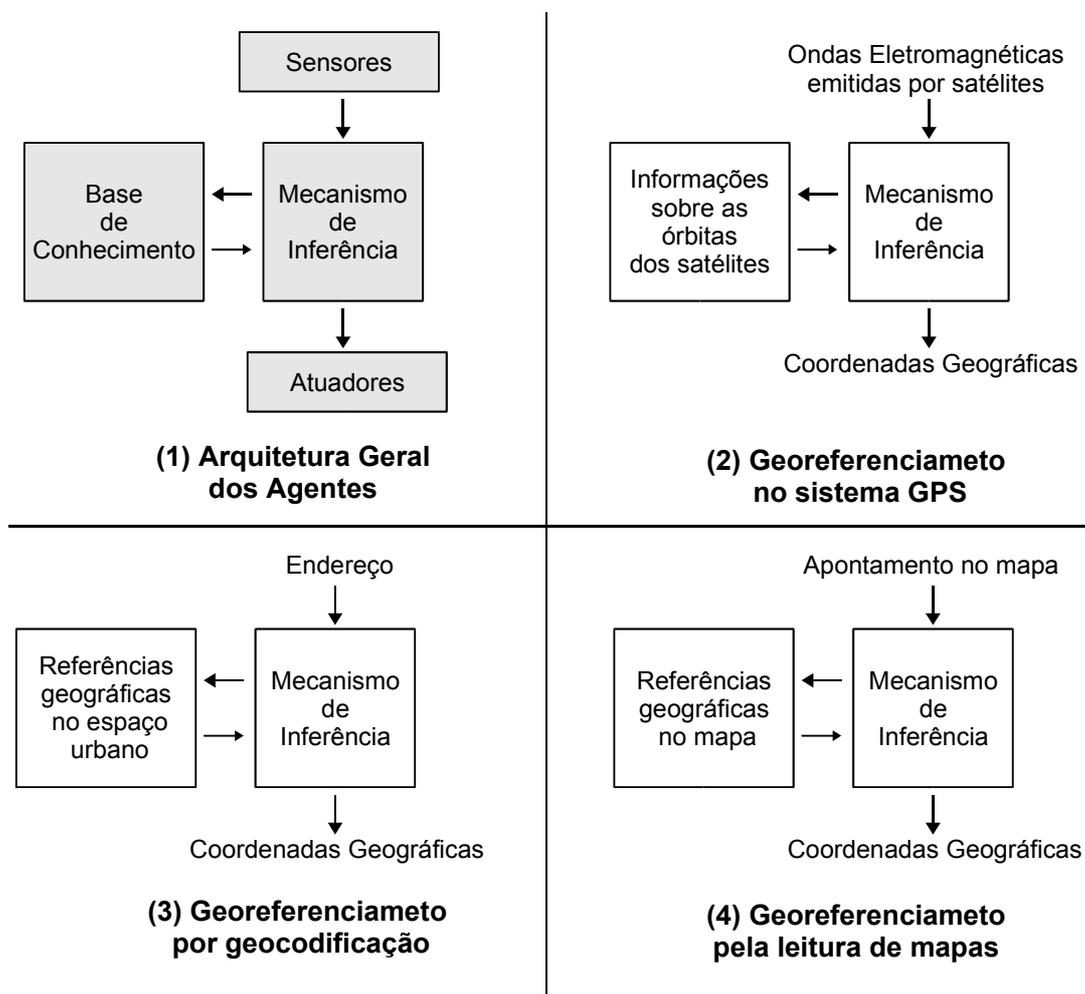


FIGURA 4.1 – Arquitetura dos agentes em contextos de georreferenciamento

Agora que já está estabelecido que o processo de georreferenciamento pode ser modelado por agentes é necessário entender o caso particular do georreferenciamento qualitativo, para que depois seja possível sugerir um mecanismo de inferência para este

caso, objetivo deste capítulo.

## **4.2 Georreferenciamento qualitativo**

A seguir são descritos dois cenários para auxiliar na compreensão do conceito de georreferenciamento qualitativo. Em um primeiro cenário, um biólogo, praticante de *trekking*, ao percorrer uma trilha localizada em um parque nacional, se depara com um espécime raro de uma planta. Ele sabe de colega que está catalogando a ocorrência destes espécimes no território brasileiro, e, portanto, ao retornar para casa, resolve contar sua descoberta a ele. Este fica tão interessado que pede uma descrição de todos os passos percorridos até encontrar o espécime em questão. O biólogo, então, conta tudo que se lembra: em que trilha estava, quanto tempo andou, por quais pontos da trilha passou antes e depois de encontrar o espécime, e ainda algumas outras referências que ele pôde observar no local onde o espécime se encontrava. Após isso, o colega providencia todos os mapas disponíveis daquele parque e, baseando-se no relato ouvido, procura pela localização do espécime, o mais precisamente possível, para catalogá-lo em sua pesquisa. Em um segundo cenário, uma moça está na cidade de São Paulo, se deslocando de carro até a casa de seu namorado quando, por um descuido, erra o caminho. Logo em seguida ela consegue voltar ao seu trajeto habitual, mas não antes de avistar um restaurante de comida vietnamita que lhe desperta interesse. Ela fala sobre o restaurante a seu namorado que também se interessa por conhecê-lo. Eles então decidem almoçar neste restaurante, só que ela não gravou o nome do lugar e tão pouco se recordava exatamente em que rua ele ficava. Sua única lembrança era que este ficava ao lado de uma locadora de vídeo de uma marca famosa. Assim, o rapaz teve a idéia de entrar no *site* da locadora para procurar por suas lojas na região onde ela tinha passado. Por sorte, a rede só tinha uma loja naquele bairro. Desta forma ele anota o endereço da locadora que, por ser ao lado do restaurante, é tudo o que eles precisam para ir até lá e apreciar as exóticas iguarias daquela cozinha oriental.

Estes dois cenários apresentados, um no ambiente rural e outro no urbano, mostram exemplos de processos de georreferenciamento qualitativo, que é a transformação de uma descrição informal em informação espacial (Yao, 1999). Como foi postulado

anteriormente, para qualquer processo de georreferenciamento é necessário uma base de referência com conteúdo espacial. No primeiro cenário esta base é o conjunto de mapas do parque e, no segundo, os endereços das locadoras de vídeo. Com a descrição informal, a qual normalmente descreve a localização através de uso de referências (Borges *et al.*, 2003), a base de referência, habilidades para entender as referências apresentadas, que foram obtidas a partir das percepções cognitivas, ou raciocínio espacial, de seus informantes, e, alguma criatividade e inteligência para pesquisar na base de referência, os atores destes cenários - o pesquisador de espécimes raros e o fã da culinária oriental - foram capazes de obter a informação espacial desejada.

Um mecanismo computacional que seja capaz de executar o processo automático de georreferenciamento qualitativo precisa ter duas habilidades: possuir capacidade de processar a descrição que contém os elementos obtidos a partir das percepções cognitivas dos seres humanos, e ser dotado de alguma inteligência para pesquisar no repositório existente. Um agente com estas características é representado na Figura 4.2.

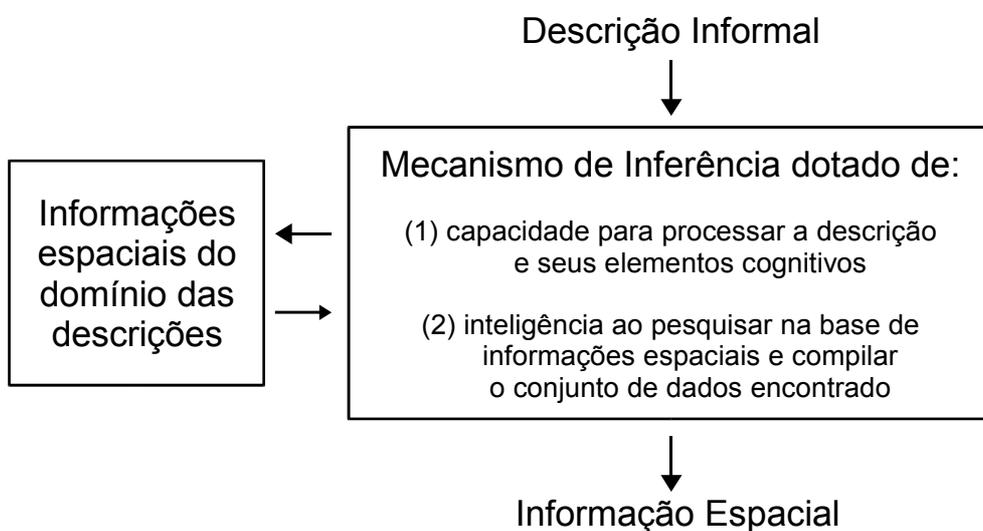


FIGURA 4.2 – Arquitetura do agente para georreferenciamento qualitativo

Considerando os elementos apresentados na Figura 4.2, é possível tirar algumas conclusões preliminares quanto à viabilização computacional deste agente: a descrição informal depende do usuário; a base de dados depende do contexto da aplicação; a informação espacial é o resultado esperado; e, portanto, a princípio, só se pode atuar no mecanismo de inferência. Em relação ao desenvolvimento deste mecanismo de

inferência, é necessário conhecer sobre a interpretação computacional de descrições informais, contendo elementos oriundos do raciocínio espacial.

### **4.3 Interpretação de descrições informais com elementos oriundos de raciocínio espacial**

De acordo com Friendschuh e Egenhofer (1997), a forma como as pessoas concebem o espaço é importante de ser considerada ao se projetar Sistemas de Informação Geográfica, pois quanto mais afinado o sistema estiver com esta forma de raciocínio, mais fácil ele será entendido pelos usuários. Esta visão reforça a idéia que foi discutida anteriormente, de que, no caso de um sistema que se propõe a implementar o georreferenciamento qualitativo, este conhecimento é essencial.

Porém, no caso de descrições informais, existe um desafio: como levar em consideração as particularidades da linguagem ou idioma? De acordo com Becker (2004) diferentes linguagens fazem com que as pessoas percebam mais certas relações espaciais do que outras. Por exemplo, em relação às expressões de orientação, o português, o inglês e o alemão possuem tanto termos alocêntricos/absolutos (direções cardeais: norte, sul, leste, oeste) como termos egocêntricos (frente/trás, esquerda, direita). Já o dialeto Maia – *Tzeltal* – possuía apenas direções absolutas: subida=sul, descida=norte e transverso. O dialeto dos aborígenes australianos – *Guugu-Yimithirr* - possui apenas direções cardeais: norte, sul, leste e oeste (Becker, 2004). Esta heterogeneidade torna o georreferenciamento qualitativo mais complexo e, quando desenvolvido, limitado as linguagens para às quais foi projetado.

Assim, dada a complexidade deste tema e aproveitando-se dos recursos gráficos e habituais dos atuais sistemas de informação foi proposta uma abordagem que combina perguntas simples e diretas, eventualmente amparadas por figuras, organizadas de forma lógica, - um tipo de *wizard*<sup>30</sup> - para que desta forma ao mesmo tempo que auxilie o usuário a passar somente as informações relevantes em suas descrições, simplifique o processamento das informações pelo sistema. Na prática, isso significa atuar no mecanismo de captação do agente, fazendo com que ele trabalhe de forma mais

---

30 Parte do programa que assiste ao usuário executar uma função da melhor forma

eficiente.

Agora é necessário ver um pouco mais detalhadamente quais são os principais elementos do raciocínio espacial.

#### 4.4 Raciocínio espacial

Além da área de pesquisa em SIG, existe uma outra área também muito interessada no raciocínio espacial: a robótica [Ligozat (2002), Mark *et al.* (1997), Mark *et al.* (1989) e Zimmermann e Freksa (1996)]. Isto se deve ao interesse de se construir robôs com as mesmas habilidades de entendimento do espaço que os seres humanos, o que, conseqüentemente, ampliaria a capacidade de locomoção e de atuação no espaço desses seres artificiais. Portanto, essa área de pesquisa é uma fonte valiosa de informações sobre o assunto, como os elementos tratados a seguir. A Figura 4.3, adaptada de Ligozat (2002), apresenta os elementos do raciocínio espacial que mais parecem se relacionar à descrição informal do espaço:

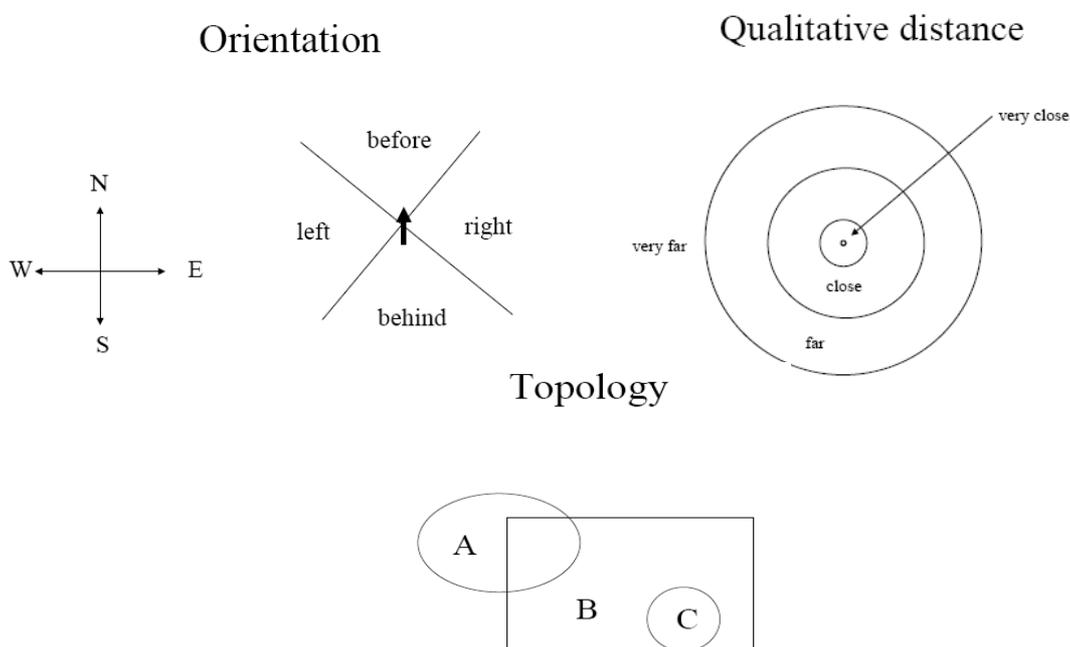


FIGURA 4.3 – Relações espaciais bidimensionais. Adaptado de Ligozat (2002).

**a) Orientação:** O senso de orientação pode ser aloccêntrico/absoluto – norte, sul, leste, oeste – ou egocêntrico – frente/trás, esquerda, direita. Para interpretar este último tipo

de descrição de orientação, é necessário antes identificar a orientação base. Como exemplo de orientação absoluta, pode-se dizer “Osasco fica a oeste de São Paulo”; e de orientação egocêntrica “o INPE fica à direita da Rodovia Presidente Dutra, quando no sentido São Paulo – Rio de Janeiro”.

**b) Distância qualitativa (relações *fuzzy*):** O senso de distância, embora possua algumas graduações – como, por exemplo, muito perto, perto, longe e muito longe – varia com as grandezas das entidades envolvidas na representação. Tome como exemplo, a distância entre as cidades de São Paulo e Rio de Janeiro, que é de aproximadamente 460 km. Em relação à dimensão total do Brasil, pode-se considerar que São Paulo fica perto do Rio de Janeiro. Já a distância do INPE ao centro de São José dos Campos, menor do que 5 km, pode ser considerada grande se comparada às dimensões da cidade. Portanto, concluiu-se que, para interpretar a distâncias qualitativas, é necessário antes identificar as grandezas envolvidas.

**c) Topologia:** Os relacionamentos topológicos são comuns nas descrições de espaço. Por exemplo, “o Obelisco fica ao lado do Parque do Ibirapuera”; “o município de São José dos Campos está contido na região do Vale do Paraíba”. A Figura 4.4 apresenta um esquema gráfico de 15 relações topológicas relativas entre uma linha e um ponto.

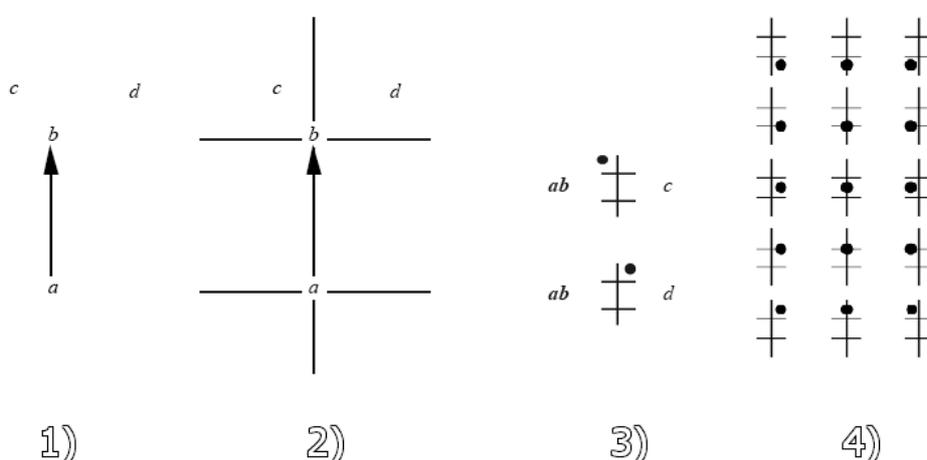


FIGURA 4.4 – Posições relativas a um caminho percorrido entre os pontos a e b. **1)** Imagine uma pessoa andando do ponto a para o ponto b. Durante o caminho ele observa o ponto c a frente e a esquerda do ponto b, e o ponto d. **2)** Introduzindo-se duas linhas

ortogonais em relação ao eixo  $ab$ , uma passando por  $a$  e outra por  $b$ , e outra linha através de  $a$  e  $b$  surge uma grade com 15 possibilidades de posicionamento como mostrado em 4). 3) As posições de  $c$  e  $d$  podem ser descritas como uma das 15 possibilidades. Adaptado de Zimmermann e Freksa (1996).

Estes elementos da descrição informal de espaço serão explorados na proposta de agente para o georreferenciamento, apresentada a seguir. Sabe-se que existem outros modelos, como o *9-intersection* (Egenhofer e Shariff, 1998) mais formal e utilizado pelos SIGs para definição das operações entre geometrias.

#### 4.5 Proposta de agente para o georreferenciamento qualitativo

O mecanismo proposto para o GeoAgente, como já mencionado anteriormente, é um *wizard* com perguntas para extrair de forma clara e objetiva os elementos do raciocínio espacial relacionado ao local a ser georreferenciado. A metodologia é partir do espaço abrangendo toda a região do domínio até que, através de um conjunto de iterações, essa área seja reduzida a uma área menor, mais próxima da realidade do local procurado. Cada iteração é composta de uma pergunta feita pelo GeoAgente e uma resposta fornecida pelo usuário. Depois de diversas iterações, esse espaço, que inicialmente abrangia toda a região do domínio, deve ser reduzido até chegar à melhor resposta possível, a qual é expressa pelo objeto de uma região, como ilustra a Figura 4.5.

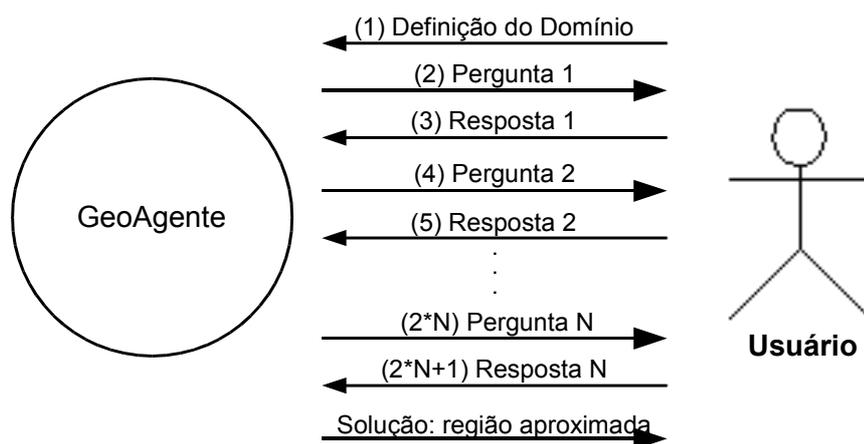


FIGURA 4.5 – Interações Usuário – GeoAgente

As iterações podem ser agrupadas de forma seqüencial, em um processo de três etapas: (1a etapa) localização de uma referência espacial na base; (2a etapa) estabelecimento de

um relacionamento espacial entre a referência obtida na 1ª etapa e o local a ser georreferenciado; (3ª etapa) definição de uma relação quantitativa aproximada.

Em alguns casos, é interessante fazer uso de esquemas gráficos para ajudar o usuário a compreender mais facilmente qual das alternativas mais se adequa a sua situação.

Como exemplo, cabe lembrar o caso do biólogo descrito no início deste capítulo. Sabendo-se que o usuário percorria uma trilha quando encontrou o espécime, algumas das iterações possíveis seriam compostas das seguintes perguntas:

1) Em qual das trilhas do parque você estava?

- a) Trilha do Sol
- b) Trilha do Macaco
- c) Caminho das Pedras
- d) Trilha do Lago Azul
- e) Outra
- f) Não sei

2) De acordo com o esquema gráfico abaixo – Figura 4.6 -, considerando os pontos A e B, respectivamente, como início e fim da trilha, em qual das posições, numeradas de 1 a 15, que mais se parece com o local onde você encontrou o espécime?

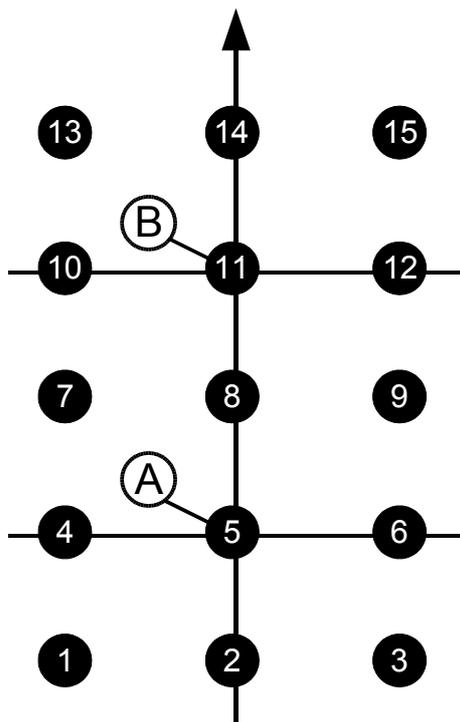


FIGURA 4.6 – Esquema para auxiliar o usuário na identificação da posição relativa entre a trilha e local encontrado

3) A qual distância da trilha se encontrava

- a) a até 100 metros
- b) entre 100 e 500 metros
- c) entre 500 metros e 1 km
- d) entre 1 km e 2 km
- e) entre 2 km e 5 km
- f) mais de 5 km

Neste exemplo, caso o usuário respondesse às três perguntas com as alternativas A, 9 e C, seria possível determinar que o local procurado se encontra em uma região delimitada por um *buffer* de 1000 metros, recortado por outro *buffer* de 500 metros, desenhados ao redor de uma seção da Trilha do Sol, entre 35 e 65%<sup>31</sup> de seu comprimento, e à direita da trilha em relação ao seu sentido habitual. A Figura 4.7 ajuda

<sup>31</sup> Valores estabelecidos arbitrariamente considerando-se uma margem de 30% de erro, para mais ou para menos, de acordo com o senso do usuário acerca da distância correspondente à metade do comprimento da trilha.

a compreender esta operação.

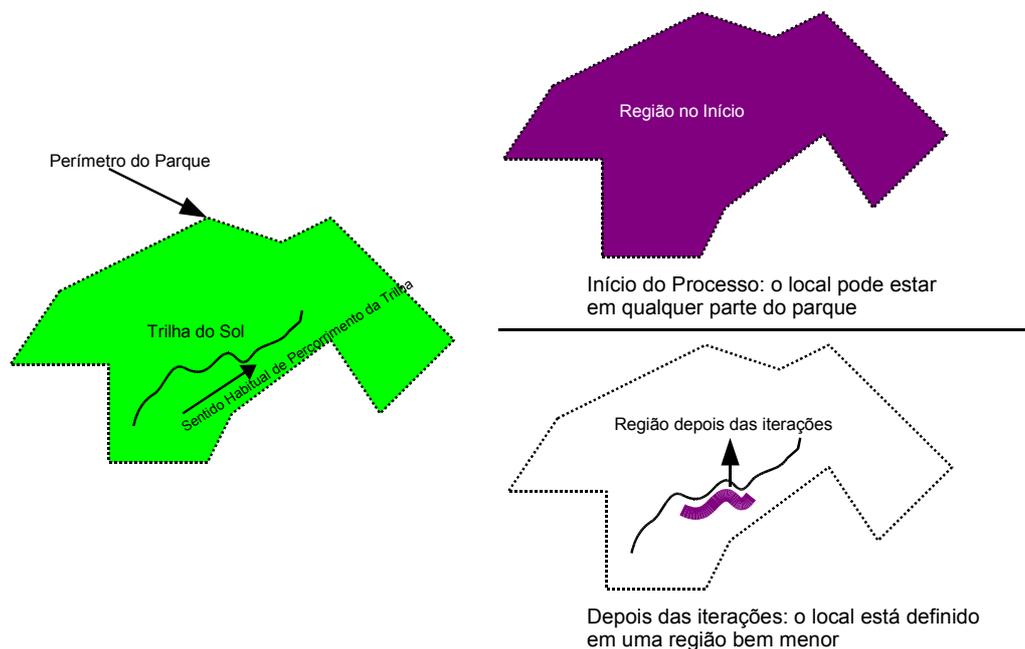


FIGURA 4.7 – Resultado de um grupo de iterações executadas pelo GeoAgente no processo do georreferenciamento qualitativo

Para refinar mais, as iterações seguintes vão buscar outros elementos no espaço do domínio, dentro ou próximas da área pré-estabelecida, para tentar encontrar outro relacionamento espacial com o local procurado. Por exemplo, podem ser apresentados pontos de referência nas proximidades da região já refinada e, caso o usuário se lembre de um ou mais destes, deve-se então passar para a segunda etapa - estabelecimento do relacionamento espacial - seguida da etapa quantitativa.

#### 4.6 O GeoAgente como um *web service*

Como já foi discutido no Capítulo 3, os agentes podem ser implementados através de *web services*. No caso do GeoAgente, este poderia se apoiar em outros *web services*, os quais teriam acesso à base de informações, seja para as consultas necessárias para a definição das perguntas, como para executar as operações espaciais ou para armazenar o objeto resultante do processo de georreferenciamento qualitativo. A Figura 4.8 ilustra um cenário deste tipo, como o que foi concebido para o protótipo, onde o *web service* GeoAgente está apoiado em *web services* do Terra WebServices.

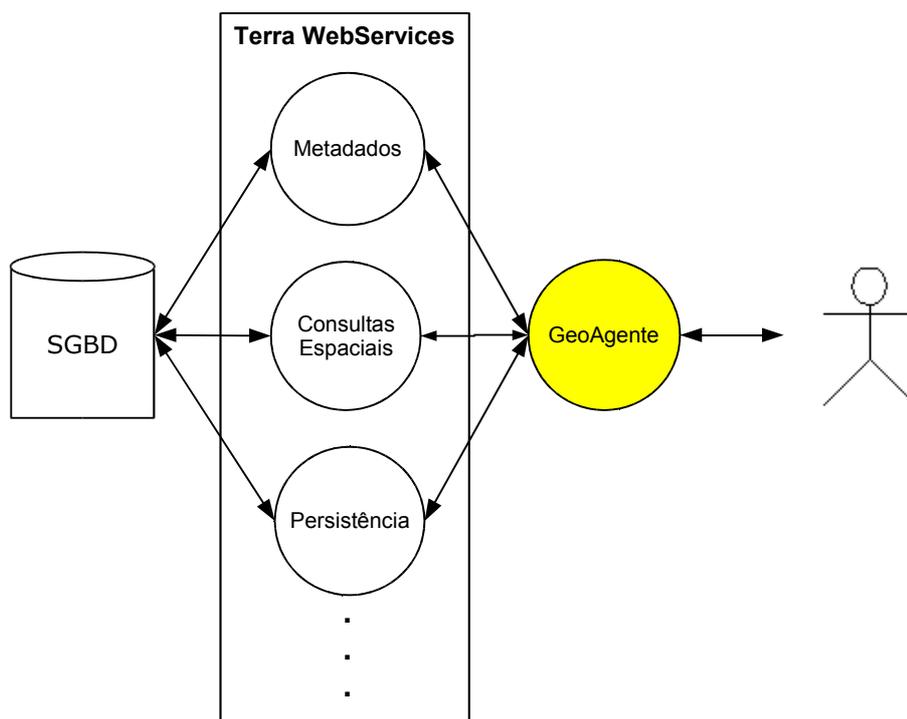


FIGURA 4.8 – O GeoAgente como um cliente do Terra WebServices

O uso desta abordagem – um *web service* como cliente de outros – é uma solução interessante, pois reduz o escopo de atuação deste agente, permitindo ao mesmo que “se concentre” nas atividades para as quais foi concebido, neste caso, definir as perguntas a serem feitas ao usuário e interpretar as respostas obtidas.

#### 4.7 Comentários finais

O GeoAgente é a implementação de um serviço baseado na arquitetura de web services proposta no Capítulo 3, concebido para ser uma solução de georreferenciamento qualitativo. Parece ser viável tanto para o ambiente urbano, o qual normalmente contém muitas referências geográficas, quanto para o ambiente rural, que possui menos referências deste tipo. No próximo capítulo, será apresentado o projeto de um sistema de apoio ao turismo, dentro do contexto do ecoturismo no município de Capitólio, MG, no qual o GeoAgente está inserido.

## **CAPÍTULO 5**

### **UM SISTEMA PARA CAPITÓLIO**

Este capítulo apresenta o projeto e a implementação de um protótipo para um contexto de uso no domínio do ecoturismo, concebido para a região de Capitólio – MG. Este protótipo é usado como modelo demonstrativo e prova de conceito das proposições desta dissertação.

#### **5.1 Motivação**

A região do município de Capitólio – MG possui um grande potencial para o ecoturismo (Barbosa, 2003). Apesar da região ser muito visitada em algumas épocas do ano, em especial no feriado de Carnaval, devido ao evento “Carnapitólio”, a região não possui turismo consolidado nas outras épocas do ano. Imagina-se que o ecoturismo possa preencher as lacunas deste turismo “festivo”, tornando a atividade turística mais consistente, e, portanto potencialmente mais benéfica para a comunidade local.

Dentre as diversas ações para incentivar o ecoturismo na região, imagina-se que a disponibilização na Web - para um número ilimitado de usuários - de um sistema de informações ao turista, contendo informações atualizadas relacionadas às belezas naturais, atividades de ecoturismo, e turismo de aventura, possa contribuir para o aumento da procura pela região por quem busca a prática destas modalidades de turismo.

#### **5.2 Domínio do problema**

Para a concepção do sistema, foi utilizada uma base espacial consolidada, fruto de uma extensa pesquisa desenvolvida por Alda Monteiro Barbosa na região do município de Capitólio – MG e municípios vizinhos – São João Batista do Glória e São José da Barra -, cujos resultados foram apresentados no trabalho intitulado “Utilização de geoprocessamento e sensoriamento remoto como subsídio para o planejamento em

ecoturismo na região do Médio Rio Grande, Minas Gerais” (Barbosa, 2003). Naquele estudo, a pesquisadora colheu diversos tipos de dados, tais como geologia, geomorfologia, informações socio-econômicas e equipamentos turísticos disponíveis na região.



FIGURA 5.1 – Imagem sintética da região do município de Capitólio e municípios vizinhos. Fonte: Barbosa (2003)

No escopo desta pesquisa<sup>32</sup>, os três principais requisitos para o sistema de informação ao ecoturista são:

1. Alta interatividade sobre a base de informações: a alta interatividade, tanto para consulta como para inserção de informações, é uma das condições principais do sistema, uma vez que é através destas interações que a base de informações será melhorada e mantida atualizada;
2. Uso da internet: o sistema deve ser disponibilizado através da internet, no intuito de atingir um grupo maior e mais diversificado de usuários;
3. Inovação: é necessário que o sistema apresente recursos inovadores em relação aos sistemas de informação geográfica existentes. Como principais propostas de inovação para o usuário, estão os recursos de georreferenciamento qualitativo e a exportação de dados para um formato compatível com o software Google Earth<sup>33</sup>. A inovação em recursos de arquitetura se dará por meio do

32 No Anexo IV está um processo de engenharia de requisitos baseado nas motivações dos turistas, como contribuição para o desenvolvimento de uma solução mais abrangente.

33 <http://earth.google.com>

desenvolvimento de sistemas baseados em *web services*.

### 5.3 Arquitetura

O Sistema para Capitólio envolve uma série de tecnologias. Listadas a seguir:

- TerraLib: biblioteca C++
- MySQL : sistema SGBD
- Apache Axis: biblioteca C++
- Apache HTTP Server: servidor *web*
- Plataforma J2SE: Máquina Java (*Java Virtual Machine*)
- Apache Tomcat: Servidor de aplicações web em Java
- Protótipo do Terra WebServices: classes C++
- Protótipo do GeoAgente: classes Java
- Protótipo para Capitólio Web: páginas JSP (*Java Server Pages*)

A Figura 5.2 demonstra o uso destas tecnologias, e a Figura 5.3 apresenta a arquitetura do protótipo para Capitólio.

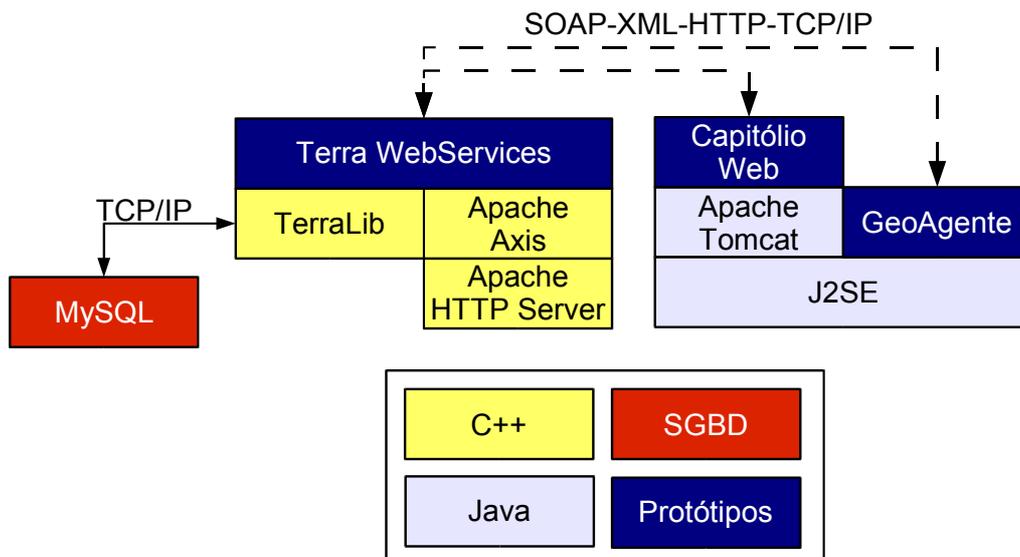


FIGURA 5.2 – Componentes, protocolos e padrões utilizados

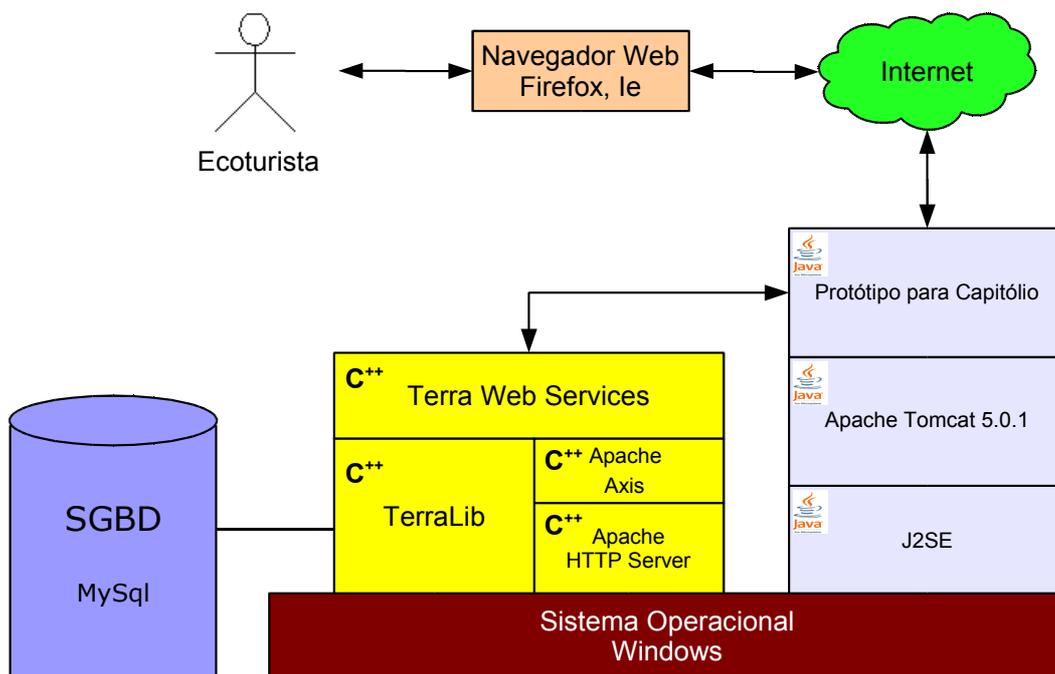


FIGURA 5.3 – Arquitetura do Protótipo para Capitólio

## **5.4 Metodologia de desenvolvimento**

### **5.4.1 Terra WebServices**

O protótipo do Terra WebServices foi desenvolvido pelo modelo de prototipagem<sup>34</sup>. Para o desenvolvimento de cada método foi utilizada a seguinte sequência de eventos:

- 1) Desenvolvimento de método no Terra WebServices
- 2) Adaptação do WSDL (*Web Service Definition Language*)
- 3) Publicação
- 4) WSDL2Java
- 5) Teste do novo método através do Java

### **5.4.2 Capitólio Web**

O protótipo do Capitólio Web foi construído em associação com o Terra WebServices. A cada novo serviço disponibilizado, uma nova funcionalidade era adicionada ao sistema. Não houve preocupação com o *layout*, mas apenas com as funcionalidades do sistema, como prova de conceito da arquitetura desenvolvida.

### **5.4.3 GeoAgente**

O GeoAgente foi concebido dentro da ferramenta de UML<sup>35</sup> Poseidon for UML<sup>36</sup>, a qual é uma ferramenta de desenvolvimento em alto nível. A Figura 5.4 exibe uma tela deste software.

---

34 De acordo com Pressman (2002), o paradigma da prototipagem envolve um ciclo de três etapas - requisitos, desenvolvimento e testes – executado diversas vezes, sendo que a cada ciclo se adicionam mais requisitos. Esta abordagem permite apresentar resultados rapidamente, reduzindo os riscos de um desenvolvimento equivocado, porém, em geral, eleva a quantidade de retrabalho.

35 Unified Modeling Language

36 <http://www.gentleware.com>

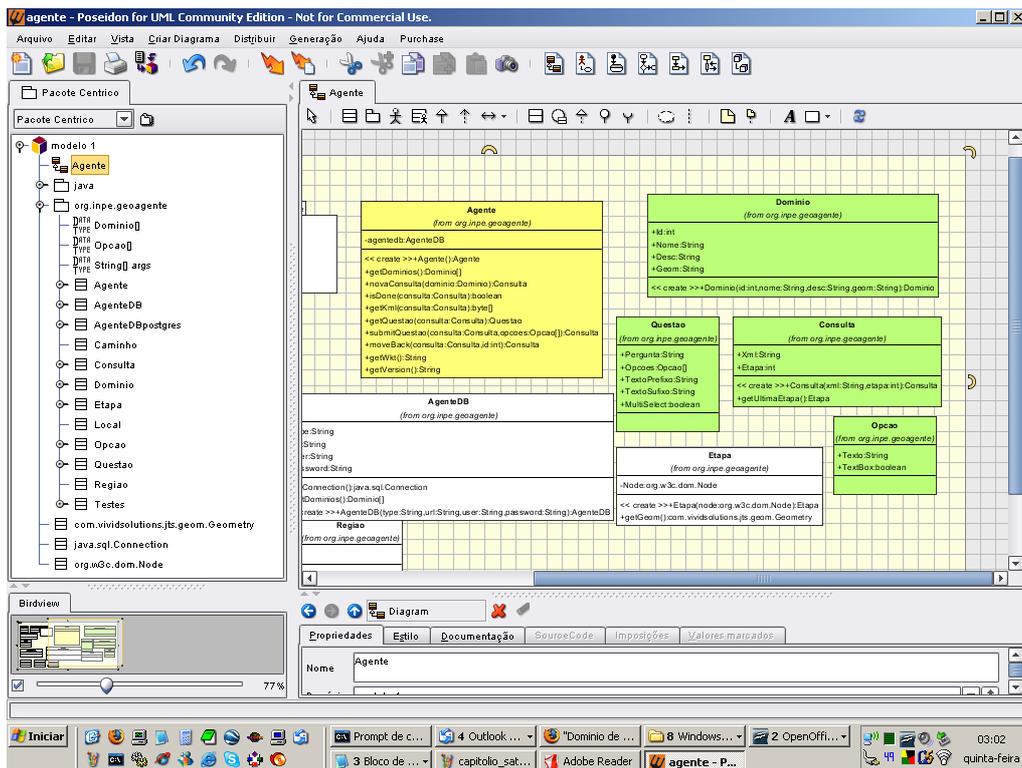


FIGURA 5.4 – Plataforma de UML utilizada para o GeoAgente

### 5.5 Interação usuário – sistema Capitólio Web – Terra WebServices

A interação dos usuários com o sistema é feita através do uso de um software navegador de internet, onde o usuário digita o endereço do site e então é direcionado para a tela de abertura do sistema. Nessa primeira tela é apresentado um campo texto e um botão. No campo texto, o usuário deve digitar os parâmetros de conexão com a base de dados: IP<sup>37</sup> ou nome do servidor de banco de dados, o tipo de sistema de banco de dados<sup>38</sup>, a porta de conexão, o login do usuário e respectiva senha. A Figura 5.5 mostra um exemplo destas informações. Depois de inserir estes dados, ao clicar no botão “connect”, caso as informações de conexão com o banco estiverem corretas, o usuário é direcionado para a tela de seleção de *layers*. O primeiro acesso ao Terra WebServices é feito neste momento, quando o sistema invoca o método `getLayers`, passando para ele os dados sobre a conexão – todos os métodos que necessitam acessar alguma informação do

37 Internet Protocol

38 A TerraLib diversos sistemas de banco de dados tais como o MySQL, Postgres, Oracle, MS Sql Server, MS Access, dentre outros. Porém, nesta primeira versão do Terra WebServices, foi utilizado apenas o *driver* do MySQL.

banco de dados devem receber este parâmetro com as informações para conexão – o qual retorna um vetor de *strings* com os nomes dos *layers*, que são então apresentados como opções para o usuário.

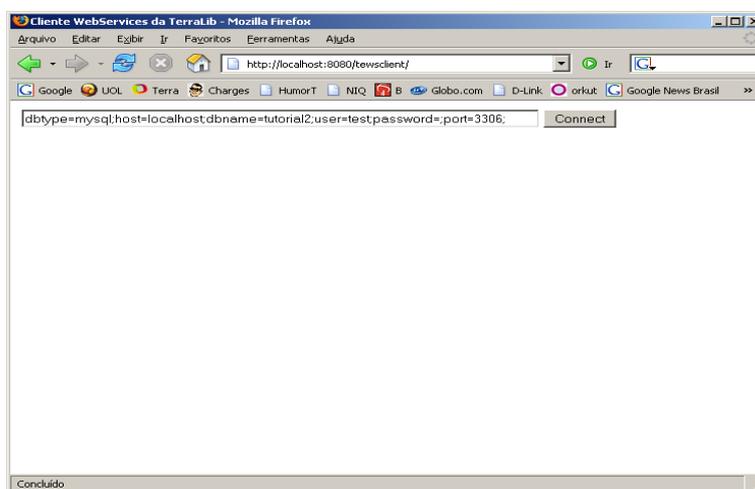


FIGURA 5.5 – Tela inicial do protótipo, onde devem ser inseridas as informações para a conexão à base de dados

Na tela de seleção de *layers* são exibidos todos os *layers* disponíveis na base conectada. O usuário deve então escolher as opções que quer visualizar (ao menos uma) e clicar sobre o botão “Atualizar” como mostra a Figura 5.6. Após este clique o usuário é direcionado para a tela de mapa interativo. Para mostrar o mapa, o sistema invoca dois métodos do Terra WebServices: *getBox* e *getMapImage*. Para o primeiro, *getBox*, além dos parâmetros de conexão foi passado um vetor de *strings* com os nomes dos *layers* selecionados pelo usuário. O método, então, retorna um *string* contendo as informações do mínimo *box* que abrange todos os objetos de todos os *layers* passados por parâmetro. De posse do *box*, o sistema invoca o método *getMapImage*, que recebe como parâmetros as informações de conexão, o vetor de *layers* cuja ordem determina a sequência de renderização, e o *box* da área desejada. O Terra WebServices então retorna para o sistema o *link* para a imagem renderizada.

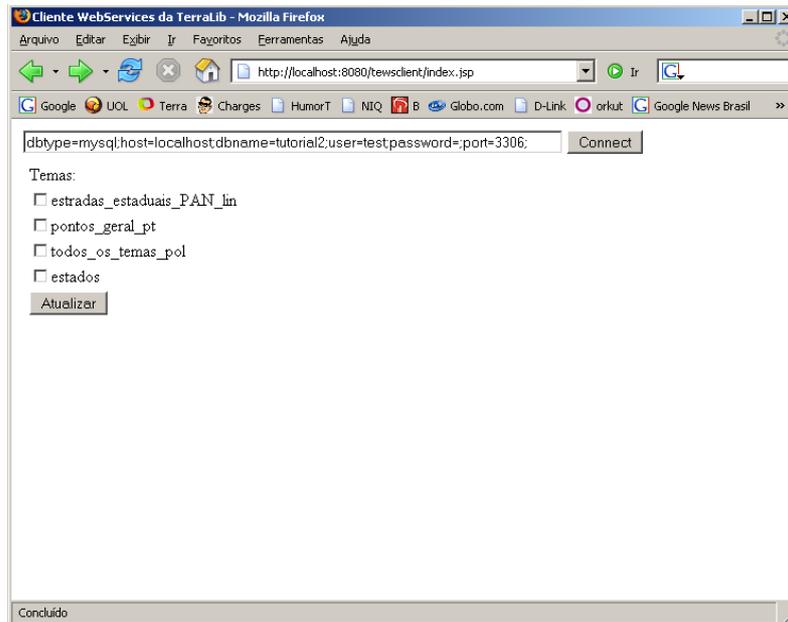


FIGURA 5.6 – Segunda tela do protótipo, onde são apresentadas as opções de *layers* disponíveis na base de dados conectada.

Na tela de mapa interativo, como mostra a Figura 5.7, o usuário pode fazer diversas operações sobre o mapa, tais como o vôo.

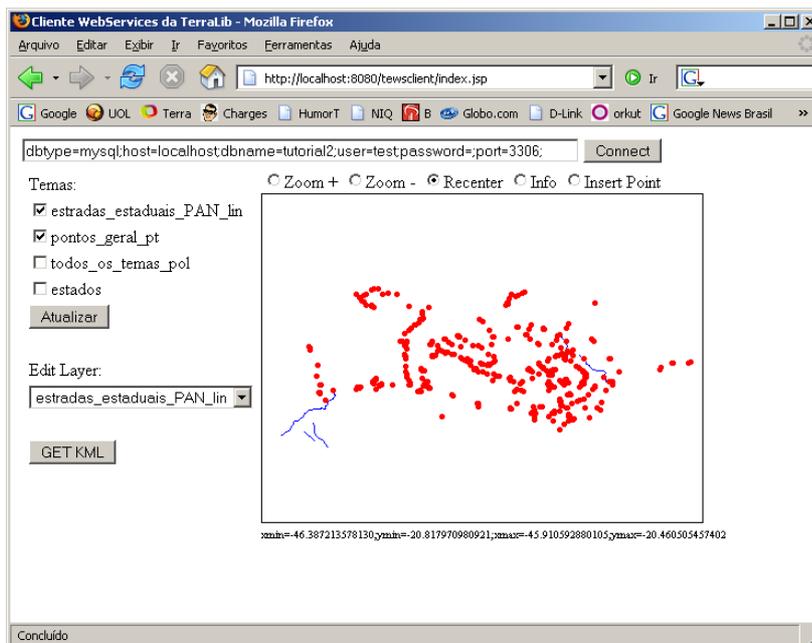


FIGURA 5.7 – Tela de mapa obtida quando existem *layers* habilitados nas opções.

Ao selecionar uma das opções, “Zoom -”, “Zoom +” ou “Recenter”, e clicar sobre o mapa, este irá se mover, centralizando no ponto clicado e, conforme a opção selecionada, a escala será dividida ou multiplicada por 2, exceto na operação “Recenter”, onde a escala é mantida. A Figura 5.8 mostra o mapa inicial após uma operação de vôo. Esta operação significa a modificação do *box* já conhecido, cujas coordenadas foram alteradas conforme a operação. Um método auxiliar, `processBox`, para o cálculo da alteração, também foi implementado no Terra WebServices. Este recebe como parâmetros o *box* atual, as coordenadas do clique e o tipo de operação selecionada (“Zoom +”, “Zoom -” ou “Recenter”).

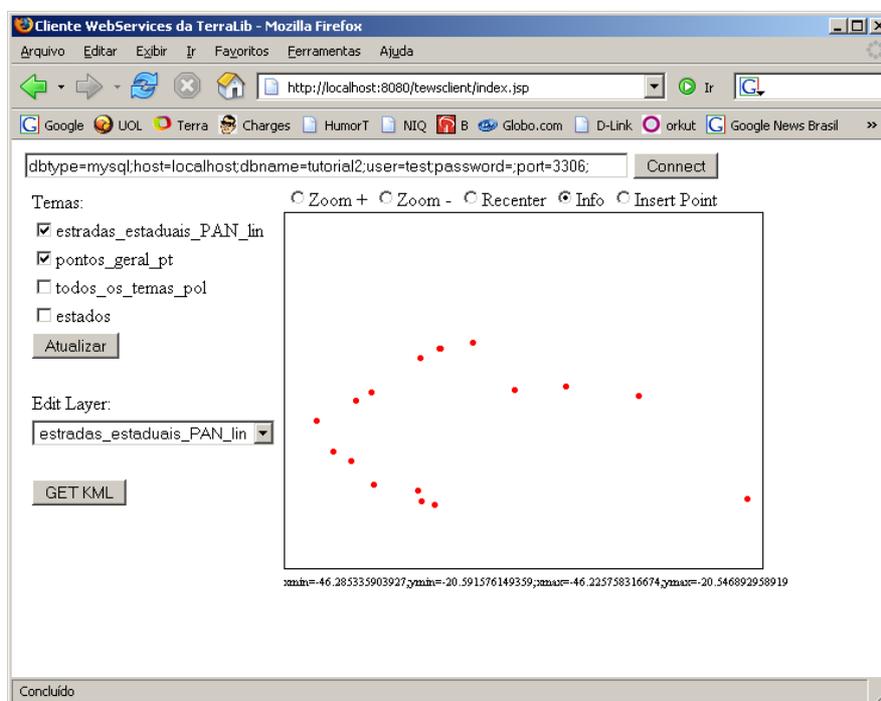


FIGURA 5.8 – Tela de mapa obtida após operação de vôo

As demais operações ocorrem de forma análoga. A operação de obter as informações dos objetos é executada da seguinte forma: ao selecionar a opção “Info” e clicar sobre um dos elementos do mapa, um *popup* será aberto contendo as informações do objeto clicado. Caso no ponto clicado exista mais de um objeto o *popup* vai conter as informações de todos eles. Caso não encontre nenhum objeto, será apresentada a mensagem “nenhum objeto foi encontrado no ponto clicado”. Para efetuar esta

operação, são invocados os métodos `getClickCoords`, para obter as coordenadas do click; o método `searchWithinDistance`, para procurar nos *layers* os objetos na coordenada encontrada; o método `getAttributeTables`, para obter as tabelas de atributos relacionadas ao *layer* onde foram obtidos resultados; o método `getTableFields`, para obter o nome dos campos das tabelas de atributos; e o método `getObjectAttributes`, para finalmente obter os valores dos objetos. A Figura 5.9 mostra o popup com as informações obtidas por este processo.



FIGURA 5.9 – *Popup* com informações do objeto clicado

A qualquer momento, o usuário pode alterar as opções de *layers* visíveis, bastando clicar sobre o botão “Atualizar”. O usuário também pode, a qualquer momento, alterar os dados de conexão com a base de dados. Neste caso, será direcionado à tela com as opções de *layers* da nova conexão.

O usuário pode também clicar sobre o botão “GETKML” para fazer o *download* do arquivo KML<sup>39</sup>, que é um arquivo XML contendo dados geográficos no formato utilizado pelo sistema Google Earth. Quando o usuário abrir, pelo Google Earth, o arquivo KML que contém todos os elementos visíveis no mapa do protótipo Capitólio, estes elementos serão exibidos na tela deste software, sobre seu mapa base, conforme mostram as Figuras 5.10 e 5.11. Para obter o KML, o sistema busca todos os objetos dos *layers* visíveis, passa para um arquivo XML, e, através de uma transformação, obtém o KML.

<sup>39</sup> [http://www.keyhole.com/kml/kml\\_doc.html](http://www.keyhole.com/kml/kml_doc.html)

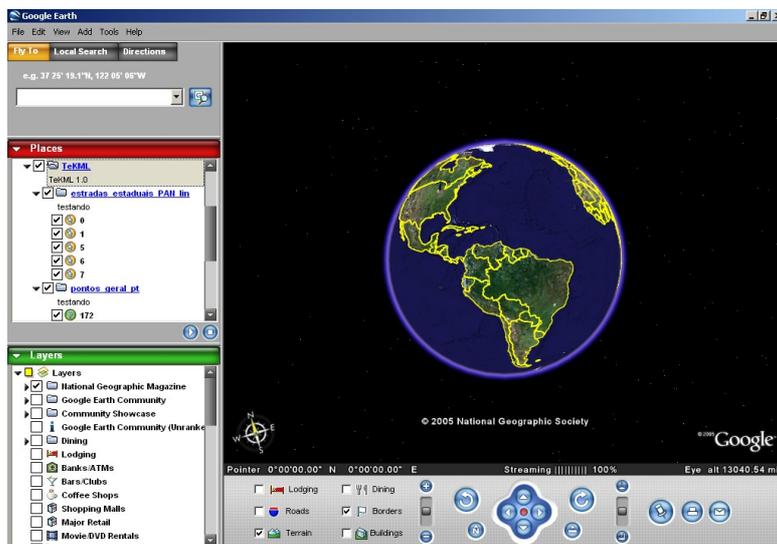


FIGURA 5.10 – Tela de abertura do Programa Google Earth

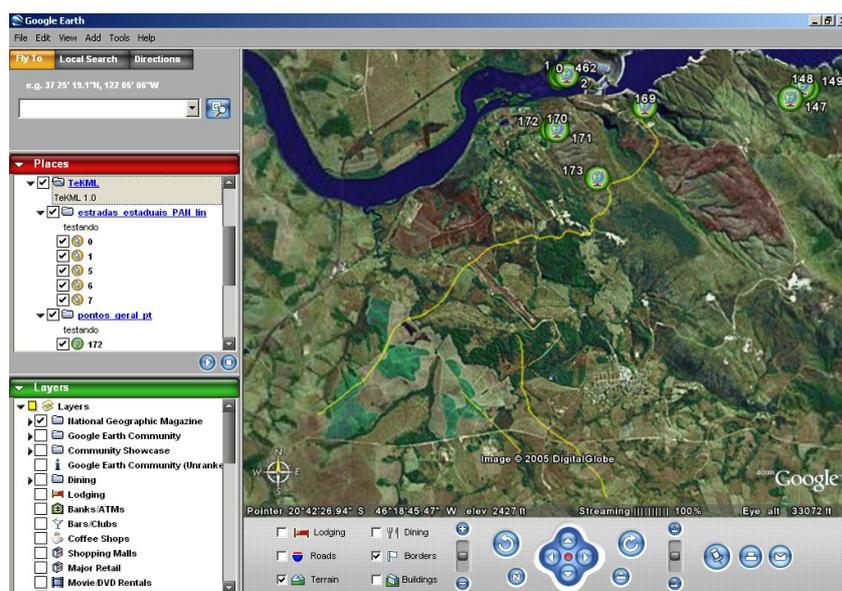


FIGURA 5.11 – Programa Google Earth exibindo informações exportadas da base de Capitólio

## 5.6 Comentários finais

Neste capítulo foi apresentado o processo de desenvolvimento do principal protótipo deste trabalho, o sistema para Capitólio. Este protótipo está fundamentado nos resultados da pesquisa desenvolvida.

Os métodos do Terra WebServices, outro protótipo, são apresentados no Anexo II, e um exemplo de uso do protótipo do GeoAgente é apresentado no Anexo III.

O Anexo I contém uma revisão dos recursos interativos presentes nos atuais SIGs para *web*, parte dos quais poderia ser implementada em uma versão “empacotada” do sistema para Capitólio.

## CAPÍTULO 6

### CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho representa contribuições para as bases de dados espaciais compartilhadas, para a TerraLib, e para os sistemas de informação ao turista.

Em relação a TerraLib, apresenta uma técnica de compartilhamento e distribuição de recursos, os chamados *web services*, que estão bastante em voga e provavelmente continuarão a ser um dos pilares da evolução da *web* para os próximos anos. Uma das principais vantagens desta técnica concebida sobre a TerraLib é que esta permite que desenvolvedores que não conhecem C++, linguagem na qual a TerraLib foi desenvolvida, façam uso de suas funções, e, assim, possam criar sistemas baseados nesta biblioteca, caracterizada por seus recursos no estado-da-arte em geoprocessamento.

Por meio do conceito de Base Viva desenvolvido, são apresentados os principais elementos envolvidos em bases de dados espaciais compartilhadas, sob condições de alta interatividade. Para viabilizar a existência destes repositórios, nestas condições, é proposto um modelo de arquitetura baseada em SGBD e agentes inteligentes sob o formato de *web services*. Estes agentes inteligentes podem acomodar padrões e especificações como as do OGC relativas a *web services*, bem como outros padrões, especificações e contextos de uso não previstos no momento. Dentro deste contexto de acomodação de novos serviços, também é proposto um inovador serviço de georreferenciamento qualitativo. Uma das possíveis soluções tecnológicas para esta arquitetura, baseada na TerraLib e em outros softwares livres, é apresentada.

Já para os sistemas de informação ao turista, este trabalho contribui com um estudo dos recursos necessários para um sistema com esta finalidade, que em parte envolve os sistemas de informações geográficas. Para estes, foi apresentada uma proposta de sua viabilização computacional através da arquitetura de *web services* desenvolvida.

Se duas palavras pudessem sintetizar os objetivos propostos para este estudo, dentro do domínio de sistemas de informação geográfica, seriam elas: inovação e interação. Pelo trabalho apresentado, acredita-se que estes objetivos foram alcançados.

Como trabalhos futuros, é esperada a criação de um projeto dos Terra WebServices que possa contar com diversos colaboradores, o qual seria uma contribuição ao projeto TerraLib, nos moldes das comunidades de software livre. Em relação ao sistema de suporte ao turista, acreditamos que, baseado no estudo de requisitos apresentado nesta dissertação, seja possível criar soluções e o seu “empacotamento”, com o uso de padrões e protocolos abertos e software livre, para as diversas áreas com potencial de exploração do turismo como atividade econômica e sustentável no Brasil.

## REFERÊNCIAS BIBLIOGRÁFICAS

Almer, A.; Stelzl, H.; Schnabel, T. *Multimedia for e-communities and eco-tourism*. **Geo Informatics – Magazine for Geo-IT Professionals**, v. 7, June 2004.

Anderson, G.; Moreno-Sanchez, R. *Building web-based spatial information solutions around open specifications and open source software*. **Transactions in GIS**, v. 7, n. 4, p. 447-466, March 2003.

Andrade, J. R. L. **Uma contribuição à análise econômica da demanda por turismo**. 2002. Tese (Doutorado em Turismo) - Escola de Comunicações e Artes (ECA), Universidade de São Paulo, 2002.

Barbosa, A. M. **Utilização de geoprocessamento e sensoriamento remoto como subsídio para o planejamento em ecoturismo na região do Médio Rio Grande, Minas Gerais**. 2003. 226 p. Dissertação (Mestrado em Sensoriamento Remoto) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2003.

Beaujardiere, J. L. (ed.). *Web Map Service. OGC Implementation*. Open Geospatial Consortium Inc., ref. number OGC 04-024, 08/02/2004, version 1.3, 85 p. Disponível em: <[http://portal.opengeospatial.org/files/?artifact\\_id=5316](http://portal.opengeospatial.org/files/?artifact_id=5316)>. Acesso em: 18/02/2006.

Becker, M. *Language and spatial relations*. Lecture Notes of Linguistics 145 - Language and Mind, University of North Carolina at Chapel Hill, April 2004. Disponível em: <<http://www.unc.edu/~mbecker/lectnotes/Ling145/linspace.pdf>> . Acesso em 12/02/2006.

Borges, K. A. V.; Assis, G. T.; Oliveira, W. A. **Georreferenciamento de relações espaciais presentes na web: um estudo de caso**. Universidade Federal de Minas Gerais, Setembro de 2003.

Bradshaw, J. M. *An introduction to software agents*. In: Bradshaw, J. M. (ed.). **Software agents**. Cambridge: The MIT Press, 1997. Cap. 1, p. 3-46.

Breu, M.; Ding Y. *Modeling the world: databases and ontologies*. **Corporate Ontology Grid (COG)**, January 2004. Disponível em: <<http://www.cogproject.org/publications/Modelling.pdf>>. Acesso em: 12/02/2006.

Buehler, K. (ed.). *OpenGIS® reference model*. **OGC Reference Model**. Open Geospatial Consortium Inc., ref. n. OGC 03-040, 04/03/2003, version 0.1.2, 100 p. Disponível em: <<http://www.opengeospatial.org/docs/03-040.pdf>>. Acesso em: 18/02/2006.

Câmara, G.; Thomé, R.; Freitas, U., Monteiro, A. M. V. *Interoperability in practice: problems in semantic conversion from current technology to OpenGIS*. In: International

Conference on Interoperable GIS, 2., 1999, Zurich, Switzerland. **Proceeding...** Berlin: Springer-Verlag GmbH, 1999, Lecture Notes in Computer Science, v. 1580.

Chen, P. P.; Thalheim, B.; Wong, L. Y. *Future directions of conceptual modeling*. In: Chen, P.P. *et al.* (ed.). **Conceptual modeling: current issues and future directions**. Berlin: Springer-Verlag GmbH, 1999, Lecture Notes in Computer Science, v. 1565, p. 294-308 (ISBN 3-540-65926-9).

Davidsson, P. **Concept acquisition by autonomous agents: cognitive modeling versus the engineering approach**. Lund University Cognitive Studies 12 (ISSN 1101-8453). Lund University: Sweden, 1992.

Davis Jr., C. A. Endereços em GIS urbano. **InfoGeo**, v. 1, n. 4, p. 44-46, nov. 1998. Disponível em: <[http://www.pbh.gov.br/prodabel/cde/publicacoes/1998/davis2\\_1998.pdf](http://www.pbh.gov.br/prodabel/cde/publicacoes/1998/davis2_1998.pdf)>. Acesso em: 12/02/2006.

Davis Jr.; C. A.; Alves, L. L. *Local spatial data infrastructures based on a service-oriented architecture*. In: Simpósio Brasileiro de Geoinformática (GeoInfo), 7., 2005, Campos do Jordão. **Anais...** São José dos Campos: INPE, 2005. Disponível em: <<http://www.geoinfo.info/geoinfo2005/papers/p71.pdf>>. Acesso em: 12/02/2006.

Egenhofer, M. J.; Shariff, A. R. B M. *Metric Details for Natural-Language Spatial Relations*. **ACM Transactions on Information Systems**, v. 16, n. 4, p. 295-321, 1998.

EMBRATUR – Instituto Brasileiro de Turismo. **Diretrizes para uma política nacional de ecoturismo**. Brasília – DF, 1994.

Ferreira, K. R.; Queiroz, G. R.; Paiva, J. A.; Souza, R. C. M.; Câmara, G. Arquitetura de software para construção de bancos de dados geográficos com SGBD objeto-relacionais. In: Simpósio Brasileiro de Banco de Dados (SBBD), 17., 2002, Gramado. **Anais...** Porto Alegre: SBC, 2002.

Frank, A. U. *Formal models for cognition – taxonomy of spatial location description and frames of reference*. In: C. Freksa *et al* (ed.). **Spatial cognition: An interdisciplinary approach to representing and processing spatial knowledge**. Berlin: Springer-Verlag GmbH, 1998, Lecture Notes in Computer Science, v. 1404 (Lecture Notes in Artificial Intelligence), p. 293-312 (ISBN 3-540-64603-5).

Freundschuh, S.; Egenhofer, M. J. *Human conceptions of spaces: implications for GIS*. **Transactions in GIS**, v. 2, n. 4, p. 361-375, 1997. Disponível em: <<http://www.spatial.maine.edu/~max/RJ31.html>>. Acesso em: 12/02/2006.

Gartner, G; Frank, A.; Retscher, G. *Pedestrian navigation system in mixed indoor/outdoor environment – the NAVIO Project*. In: CORP 2004 and Geomultimedia04 Symposium, 2004, Vienna, Austria. **Proceedings...** Vienna: CORP, 2004. p. 165-171. Disponível em:

<[http://corp.mmp.kosnet.com/CORP\\_CD\\_2004/archiv/papers/CORP2004\\_GARTNER\\_FRANK\\_RETSCHER.PDF](http://corp.mmp.kosnet.com/CORP_CD_2004/archiv/papers/CORP2004_GARTNER_FRANK_RETSCHER.PDF)>. Acesso em 12/02/2006.

Gilbert, D.; Aparicio, M.; Atkinson, B.; Brady, S.; Ciccarino, J.; Grosf, B.; O'Connor, P.; Osisek, D.; Pritko, S.; Spagna, R.; and Wilson, L. **IBM Intelligent Agent Strategy**, 1995, IBM Corporation.

Gunzer, H. *Introduction to Web Services*. March 2002. Disponível em: <<http://bdn.borland.com/article/images/28818/webservices.pdf>>. Acesso em: 18/02/2006.

Hagel, J.; Brown, J. S. *Break on through to the other side: a missing link in redefining the enterprise*. **Manyworlds.com**, December 2002. Disponível em: <<http://www.manyworlds.com/index2.aspx?from=/exploreCO.aspx&coid=CO418033466072>>. Acesso em 12/02/2006.

Hendry, F. *Best practices for web mapping design*. In: Open Source GIS Conference & Second Annual MapServer Users Meeting (OSGIS 2004 & MUM2), 2004, Ottawa, Canada. **Proceeding...** Canada: Carleton University, 2004. Disponível em: <<http://www.omsug.ca/osgis2004/proceedings.html>>. Acesso em: 17/02/2006.

Houaiss, A. **Dicionário Houaiss de língua portuguesa**. 1. ed. Rio de Janeiro: Objetiva, 2004.

Kralidis, T.; Pulsifer, P.; van den Eijnden, B.; McKenna, J. *OGC in open source GIS products*. In: Open Source GIS Conference & Second Annual MapServer Users Meeting (OSGIS 2004 & MUM2), 2004, Ottawa, Canada. **Proceeding...** Canada: Carleton University, 2004. Disponível em: <<http://www.omsug.ca/osgis2004/proceedings.html>>. Acesso em: 17/02/2006.

Lieberman, J. (ed.). *OpenGIS® web services architecture*. **OGC Discussion Paper**. Open Geospatial Consortium Inc., ref. n. OGC 03-025, 18/01/2003, version 0.3, 58 p. Disponível em: <<http://www.opengeospatial.org/docs/03-025.pdf>>. Acesso em: 18/02/2006.

Ligozat, G. *Qualitative spatial and temporal reasoning*. Bombay, India, 2002. Invited talk at Artificial Intelligence: Theory and Practice (KBCS-2002 Conference), December 2002.

Lima, P.; Câmara, G.; Queiroz, G. R. GeoBR: intercâmbio sintático e semântico de dados espaciais. In: Simpósio Brasileiro de Geoinformática (GeoInfo 2002), 4., 2002, Caxambu. **Anais...** Belo Horizonte: SBC/INPE/PRODABEL, 2002. Disponível em: <<http://www.geoinfo.info/geoinfo2002/papers/Lima.pdf>>. Acesso em: 12/02/2006.

Lopes, L. T.; Audy, J. L. N. Em busca de um modelo de referência para engenharia de requisitos em ambientes de desenvolvimento distribuído de software. In: Workshop em

Engenharia de Requisitos (WER), 6., 2003, Piracicaba. **Anais...** Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), 2003. Disponível em: <<http://wer.inf.puc-rio.br/wer03>>. Acesso em: 12/02/2006.

Mabrouk, M. (ed.). *OpenGIS® location services (OpenLS): core services. OGC Implementation Specification*. Open Geospatial Consortium Inc., ref. number OGC 03-006r3, 16/01/2004, version 1.0, 165 p. Disponível em: <[portal.opengis.org/files/?artifact\\_id=3418](http://portal.opengis.org/files/?artifact_id=3418)>. Acesso em: 18/02/2006.

Mark, D. M.; Egenhofer, M. J.; Hornsby, K. (ed.). *Formal models of commonsense geographic worlds*. Report on the Specialist Meeting of Research Initiative 21, Technical Report 97-2. Santa Barbara: University of California – National Center for Geographic Information and Analysis (NCGIA), 1997. Disponível em: <<http://www.geog.buffalo.edu/ncgia/i21/i21report.html>>. Acesso em: 18/02/2006.

Mark, D. M.; Gould, M. D.; Friendschuh S. M.; Egenhofer, M. J.; Kuhn, W.; McGranaghan, M.; Svorou, S. (ed.). *Working bibliography on languages of spatial relations*. Technical Report 89-10, 1. ed., 28 p. Santa Barbara: University of California – National Center for Geographic Information and Analysis (NCGIA), 1989. Disponível em: <<http://www.spatial.maine.edu/~max/NCGIA89-10.pdf>>. Acesso em: 18/02/2006.

Oliveira, K. C. L.; Decanini, M. M. S. Projeto e produção cartográfica do guia turístico eletrônico das represas paulistas para internet. **Revista Brasileira de Cartografia**, n. 54, dez. 2002. Disponível em: <[http://www2.prudente.unesp.br/rbc/\\_2002/54\\_05.htm](http://www2.prudente.unesp.br/rbc/_2002/54_05.htm)>. Acesso em: 12/02/2006.

Parsons, E. *The role of web services for spatial data delivery*. In: Geospatial Information & Technology Association Annual Conference (GITA), 26., 2003, San Antonio, USA. Disponível em: <[http://www.gita.org/resources/downloads/web\\_services.pdf](http://www.gita.org/resources/downloads/web_services.pdf)>. Acesso em 18/02/2006.

Perez, A. L. F., Fogaça, E. T., Pozzebon, E. **Uma ferramenta de auxílio ao turismo da Serra Catarinense**. In: Congresso Brasileiro de Pesquisas Ambientais e Saúde (CBPAS 2003), 3., 2003, Santos. Disponível em: <<http://www.das.ufsc.br/~eliane/artigos/perez03a.pdf>>. Acesso em: 18/02/2006

Pressman, R. **Engenharia de Software**. 5. ed. Rio de Janeiro: McGraw-Hill, 2002.

Queiroz, G. R. **Algoritmos geométricos para bancos de dados geográficos: da teoria à prática na TerraLib**. 2003. 143 p. Dissertação (Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2003.

Rizzi, P. Visualização cartográfica aplicada ao turismo: uma proposta metodológica. In: Congresso Brasileiro de Cartografia, 21., 2003, Belo Horizonte. **Anais...** Rio de Janeiro: Sociedade Brasileira de Cartografia, Geodésia, Fotogrametria e Sensoriamento Remoto, 2003. Disponível em: <[http://www.cartografia.org.br/xxi\\_cbc/284-C60.pdf](http://www.cartografia.org.br/xxi_cbc/284-C60.pdf)>. Acesso em: 12/02/2006.

Russel, S.; Norvig, P. *Artificial intelligence: a modern approach*. 2. ed. New Jersey: Prentice Hall, 2003. 1132 p.

Sonnet, J. (ed.). *OWS 2 common architecture: WSDL SOAP UDDI. OGC Discussion Paper*. Open Geospatial Consortium Inc., 20/08/2004, Version 1.0.0, 76 p. (OGC 04-060rl). Disponível em: <[https://portal.opengeospatial.org/files/?artifact\\_id=8348](https://portal.opengeospatial.org/files/?artifact_id=8348)>. Acesso em: 14/02/2006.

Vinhas, L.; Ferreira, K. R. Descrição da TerraLib. In: Casanova, M. A.; Câmara, G.; Davis Jr., C. A.; Vinhas, L.; Queiroz, G. B. (ed.). **Bancos de dados geográficos**. Curitiba: Editora MundoGEO, 2005, 1. ed., v. 1, p. 397-440.

Whiteside, A. (ed.). *OpenGIS® web services common specification. OGC Implementation Specification*. Open Geospatial Consortium Inc., ref. number OGC 05-008, 22/11/2005, version 1.0.0, 106 p. Disponível em: <[https://portal.opengeospatial.org/files/?artifact\\_id=8798](https://portal.opengeospatial.org/files/?artifact_id=8798)>. Acesso em: 18/02/2006.

Yao, X. *Qualitative geo-referencing for web-based GIS*. In: The International Conference on Geoinformatics and Socioinformatics (Geoinformatics'99), 1999, Ann Arbor, USA. **Proceedings...** Ann Arbor: University of Michigan, 1999. Disponível em: <[http://www.umich.edu/~iinet/chinadata/geoim99/Proceedings/Yao\\_xiaobai.PDF](http://www.umich.edu/~iinet/chinadata/geoim99/Proceedings/Yao_xiaobai.PDF)>. Acesso em: 18/02/2006.

Zimmermann, K.; Freksa, C. *Qualitative spatial reasoning using orientation, distance, and path knowledge*. **Applied Intelligence**, v. 6, n. 1, p. 49-58, January 1996. Disponível em: <<http://www.cosy.informatik.uni-bremen.de/staff/freksa/publications/QualiSpa96ZiFr.pdf>>. Acesso em: 12/02/2006.

## ANEXO I

### INTERATIVIDADE DE SIG PARA WEB

Desde um simples mapa de papel que pode ser observado, dobrado e carregado, até um poderoso SIG em web, todos os SIGs possuem algum nível de interatividade com o usuário. Embora, dependendo da finalidade, os SIG possam ter apenas interface alfanumérica com os usuários, a característica que mais se sobressai é a apresentação gráfica de dados geográficos, mais conhecida como **mapa**.

Os SIGs para web podem ser divididos em estáticos e dinâmicos. Estático significa que as informações são passadas do servidor para a máquina do usuário de uma só vez, em forma de imagens estáticas; e dinâmico significa que as informações do mapa podem ser descarregadas de acordo com as solicitações do usuário. Dentre os SIGs dinâmicos podemos classificá-los em “apenas vistos” e “interativos”. Apenas vistos significa que o usuário não tem participação na construção da visualização do mapa, e interativos, como o próprio nome diz, mostra uma situação de interação do usuário com a forma de visualização (Rizzi, 2003).

#### **Recursos interativos**

Dentre os SIGs interativos, o recurso mais comum que podemos encontrar na web é a possibilidade de o usuário mudar a escala e centro do mapa apresentado na tela de forma a obter informações mais detalhadas sobre parte do mapa ou mais abrangentes sobre uma área maior do mapa. Em um nível um pouco mais interativo, o SIG permite que o usuário clique sobre um determinado elemento do mapa para que, em seguida, sejam exibidas informações detalhadas sobre o mesmo. Outro recurso deste tipo permite ao usuário controlar quais camadas de informação devem ser exibidas no mapa.

Nos SIGs de maior interatividade existentes na web, além dos recursos citados anteriormente, também é possível localizar e obter informações de elementos baseados

em seus atributos ou selecionar elementos de uma área do mapa através da determinação da mesma por desenho. Um SIG com este nível de interatividade, da cidade de Dublin, localizada no estado de Ohio, nos EUA, pode ser acessado pelo link <http://www.dublin.oh.us/business/gis/intro/>. Neste SIG é possível interagir com um mapa bastante detalhado da cidade, com diversas camadas das quais os usuários podem controlar a visualização, possuindo ainda alguns recursos de consulta.

Outro recurso de interatividade, pouco comum nos SIGs para web, é a possibilidade de o usuário inserir ou modificar informações (Hendry, 2004). Um dos motivos para isto é que a finalidade da maioria dos SIGs para web é apenas consulta. Talvez outro motivo seja que os usuários nem sempre podem ser considerados uma fonte confiável de informações. Mesmo assim, o site <http://www.confluence.org>, um sucesso mundial, é caracterizado por possuir informações fornecidas exclusivamente pelos usuários. Neste SIG, usuários de todo o mundo podem registrar informações e fotos sobre determinadas coordenadas da superfície do globo terrestre, baseadas no sistema longitude/latitude (pontos geográficos), de forma a receber o mérito por ter sido o primeiro a registrá-las para o ponto.

De maneira geral, os recursos interativos existentes atualmente nos SIGs na web são: vôo, exibição de atributos, impressão, controle dos níveis de informação, localização de endereço, seleção por atributos, seleção espacial e edição da base de dados, descritos abaixo.

## **Recursos comuns**

### **a) Vôo**

Talvez o recurso mais básico de um SIG interativo seja o controle sobre a escala e posição do centro do mapa. Com este recurso os usuários podem centralizar o mapa em determinada região e aumentar sua escala, obtendo assim um maior nível de detalhamento ou, com menor escala, observar uma região com menos detalhamento porém mais abrangente. Este recurso amplia a possibilidade de compreensão e análise do mapa.

### **b) Exibição de atributos**

Provavelmente a característica mais fundamental de qualquer SIG é a possibilidade de vincular atributos aos dados de localização. Dentre as diversas utilidades dos atributos, estes podem ser usados para adicionar textos (rótulos) aos elementos geográficos ou para modificar suas representações gráficas através de temáticos, ampliando o entendimento ou a profundidade de informações apresentadas no mapa. O potencial analítico de um SIG é proporcional à quantidade de atributos associados aos dados geográficos. Através do recurso de exibição de atributos, o usuário, ao identificar um elemento no mapa, o que normalmente é feito através de um clique sobre o mesmo, pode obter uma lista dos atributos associados ao elemento.

### **c) Localização de endereço**

Provavelmente os primeiros SIGs para web a se tornarem populares na internet foram os guias de ruas, onde o usuário digita um endereço para obter o mapa de sua localização. Recurso bastante útil, sobretudo nos grandes centros urbanos, para quem precisa se deslocar para um endereço do qual desconhece a localização.

### **d) Seleção por atributos**

A seleção por atributos permite obter uma listagem ou a representação gráfica de parte dos elementos de um nível de informação cujos atributos atendam a determinadas condições impostas pelo usuário. Por exemplo, em um nível de informações com as cidades brasileiras e seus atributos demográficos, obter a lista das cidades com população acima de 1 milhão de habitantes; ou em um nível de informação com as ruas de uma cidade com o atributo de nome do bairro, obter o mapa das ruas de determinado bairro.

## **Recursos avançados**

### **e) Seleção espacial**

Diferente da seleção por atributos, a seleção espacial permite consultar os elementos de um nível de informações que atendam a determinadas restrições ou relacionamentos espaciais. As possíveis restrições ou relacionamentos espaciais são métricas, direcionais ou topológicas (Queiroz, 2003). As restrições métricas se referem ao objeto geográfico e permitem, por exemplo, selecionar os distritos de uma cidade baseados nos valores de suas áreas ou perímetros. Os relacionamentos direcionais permitem selecionar elementos baseados em relacionamentos direcionais com outros elementos. Um exemplo de seleção espacial por relacionamento direcional é a seleção das cidades localizadas a Norte de Brasília. Os relacionamentos topológicos permitem selecionar elementos que atendam a determinadas condições topológicas em relação a outros elementos. Por exemplo, selecionar os estados que fazem divisa com o Estado de São Paulo ou as ruas que cruzam limites de determinado bairro.

#### **f) Edição da base de dados**

Os recursos de edição da base de dados permitem aos usuários inserir, modificar ou excluir informações na base de dados do SIG. Este recurso pode ser bastante útil em SIGs cujas informações necessitam ser atualizadas por diversas pessoas de forma simultânea.

#### **g) Roteamento**

O roteamento, disponível em alguns sites, permite ao usuário encontrar o caminho mais curto ou mais rápido entre dois ou mais pontos. Para respostas precisas, além de algoritmos sofisticados geralmente apoiados em grandes recursos computacionais, é necessário uma base de dados bastante rica em informações sobre a malha viária, com informações tais como mãos de direção, conversões permitidas, velocidade média por horário, etc. Em alguns casos o site chega a informar o preço aproximado de uma corrida de táxi pelo trajeto.

### **Recursos inovadores**

#### **h) Visualização 3D**

A presença de SIGs na web com visualização tridimensional ainda é incipiente. Este recurso permite ao usuário obter percepção bastante melhorada da ocorrência da terceira dimensão pelo espaço. A terceira dimensão pode ser tanto o dado altimétrico, formando uma representação do relevo, quanto outras informações temáticas.

#### **i) Georreferenciamento qualitativo**

Os processos de georreferenciamento normalmente presentes na web e nos SIGs em geral são processos determinísticos. O georreferenciamento qualitativo, como discutido no Capítulo 4, pode auxiliar os usuários nas situações onde o quantitativo não é possível e portanto, dada a sua incoerência, é um recurso inovador.

A Figura A.1 contém uma tabela com o resultado de uma pesquisa de alguns sites e seus recursos interativos de acordo com as definições e classificação feitas anteriormente.

	Vôo	Exibe Atributos de Objetos	Impressão	Seleção dos Níveis de Informação	Localização/Seleção por Atributos	Roteamento	Seleção Espacial	Edição da Base de Dados	Visualização 3D	Georeferenciamento Qualitativo
<a href="http://www.dublin.oh.us/business/gis/intro/">http://www.dublin.oh.us/business/gis/intro/</a>	☑	☑	☑	☑	☑	☒	☑	☒	☒	☒
<a href="http://www.confluence.org">http://www.confluence.org</a>	☑	☑	☑	☑	☑	☒	☒	☑	☒	☒
<a href="http://www.eatlas.sp.gov.br">http://www.eatlas.sp.gov.br</a>	☑	☑	☑	☑	☑	☒	☒	☒	☒	☒
<a href="http://mapas.terra.com.br">http://mapas.terra.com.br</a>	☑	☒	☑	☒	☑	☒	☒	☒	☒	☒
<a href="http://www.swissgeo.ch">http://www.swissgeo.ch</a>	☑	☑	☑	☑	☑	☒	☒	☒	☒	☒
<a href="http://www.apontador.com.br">http://www.apontador.com.br</a>	☑	☑	☑	☑	☑	☑	☒	☒	☒	☒
<a href="http://thayatal.geo-konzept.de/z/regeo">http://thayatal.geo-konzept.de/z/regeo</a>	☑	☑	☑	☑	☑	☒	☒	☒	☑	☒
<a href="http://maps.google.com/">http://maps.google.com/</a>	☑	☑	☑	☑	☑	☑	☒	☒	☒	☒
<a href="http://earth.google.com/">http://earth.google.com/</a>	☑	☑	☑	☑	☑	☑	☒	☒	☑	☒
	Comuns				Avançados				Inovadores	

☑ = Possui o Recurso

☒ = Não Possui o Recurso

FIGURA A.1 - Recursos interativos disponíveis em alguns sites

Alguns dos recursos interativos apresentados podem ser potencializados através do uso de uma recente tecnologia denominada Ajax.

## Ajax

Ajax ou “Asynchronous JavaScript And XML” é uma nova técnica que combina HTML, JavaScript e XML para criar aplicações web de alta interatividade, eficiência e usabilidade que está despontando como uma forte tendência para as aplicações web. Um exemplo bastante difundido de SIG utilizando Ajax é o site Google Maps<sup>40</sup>.

O princípio do Ajax é descarregar, de forma assíncrona, os dados para a máquina cliente de forma a se adiantar em relação a algumas solicitações dos usuários, de forma que no momento em que o usuário solicitar a informação ele já foi descarregada e portanto aparece praticamente de forma instantânea.

<sup>40</sup> <http://maps.google.com/>

Para atingir estes objetivos o Ajax se vale de recursos dos navegadores que, em sua maioria, são compatíveis com JavaScript e permitem a navegação em diversos frames de forma simultânea. Assim enquanto o usuário interage com o frame principal da aplicação, outros frames secundários estão descarregando informações para serem posteriormente utilizadas pelo principal. Desta forma são reduzidos os recarregamentos totais da página que ao necessitarem de tempo para se processar fazem o usuário perder tempo de navegação ativa. No Google Maps, por exemplo, além do mapa que o usuário está vendo na tela são descarregados mapas ao redor do mapa da tela de forma que em caso da utilização do recurso de vôo o usuário já tem descarregadas ao menos uma parte das informações desejadas. E no caso de pontos de interesse os mesmos são descarregados e plotados por cima do mapa em outra camada do navegador, sem a necessidade de recarregar as imagens com os pontos plotados.

No Ajax todo o tráfego de informações é feito por XML, o que torna os sistemas desenvolvidos em Ajax compatíveis com os web services, como os que foram explorados no Capítulo 3, sem a necessidade de uma ou mais camadas intermediárias.

Além do Google Maps que utiliza Ajax, o Google<sup>41</sup> disponibilizou outro SIG bastante inovador em recursos interativos, o Google Earth.

### **Google Earth**

O Sistema Google Earth<sup>42</sup>, lançado pelo Google em 2005, e que em pouco tempo se tornou um sucesso mundial, apresenta diversas inovações do ponto de vista da interatividade: Pela primeira vez foi possível que qualquer usuário de internet tivesse acesso ao acervo de imagens de satélite e fotografias aéreas de toda a extensão do globo terrestre, e com velocidade de navegação surpreendentemente rápida.

Outro aspecto importante do Google Earth é que é possível integrar dados próprios com a base do Google. Isto pode ocorrer apenas na máquina do usuário ou o usuário pode descarregar e visualizar informações produzidas por terceiros.

---

41 <http://www.google.com>

42 <http://earth.google.com>

Como limitantes, o Google Earth, que possui uma versão gratuita para uso não comercial, no lado cliente é baseado em um software desenvolvido pelo Google, até agora de código fechado, que tem versões para um número limitado de sistemas operacionais. Mesmo com estes limitantes o Google Earth já criou um novo paradigma de aplicativos de SIG para web.

## ANEXO II

### MÉTODOS DO PROTÓTIPO DO TERRA WEBSERVICES

#### 1) getLayers

Parâmetros:

- String com os dados para conexão à base de dados

Ex: "dbtype=mysql;host=localhost;dbname=tutorial2;user=test;password=;port=3306;"

Retorno: array de String com os nomes dos layers disponíveis na base

#### 2) getBox

Parâmetros:

- String com os dados para conexão à base de dados

- Array de String com os nomes dos layers

- String com os dados de tamanho da imagem do mapa

Ex: "imagewidth=400;imageheight=400" (medidas em pixels)

Retorno: String com a representação do Box definido pelo conjunto de layers e ajustado

para o tamanho da imagem de mapa

Ex: "xmin=-46.656760442662;ymin=-21.997607208534;xmax=-44.747065846205;ymax=-20.565336261191;" \*

\* O sistema de coordenadas do protótipo é Longitude/Latitude na projeção WGS84

#### 3) getMapImage

Parâmetros:

- String com os dados para conexão à base de dados

- Array de String com os layers a serem plotados

- String com os dados de tamanho da imagem do mapa

- String com a representação do Box

Retorno: String com o link para a imagem

#### 4) getMapWidth

Parâmetros:

- String com a representação do Box

Retorno: String com o valor da largura do box em metros

#### 5) getClickCoords

Parâmetros:

- String com as coordenadas (em pixels) do clique

Ex: "clickx=30;clicky=110;"

- String com a representação do Box

- String com os dados de tamanho da imagem do mapa

Retorno: String com a coordenada (em Long/Lat WGS84) do click

Ex: "x=-46.656760442662;y=-21.997607208534;"

#### 6) processBox

Parâmetros:

- String com a representação do Box
- String com a operação sobre o Box

Ex1: "west" ; Ex2: "doublescale" ; Ex3: "recenter"

- String com as coordenadas (em pixels) do clique

Retorno: String com a representação do Box processado

### **7) searchWithinBox**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer a ser pesquisado

Ex: "estados"

- String com a representação do Box

Retorno: Array de String com os IDs dos objetos encontrados

### **8) searchWithinDistance**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer a ser pesquisado
- String com os parâmetros da busca

Ex: "x=-46.656760442662;y=-21.997607208534;r=200" (raio em metros)

Retorno: Array de String com os IDs dos objetos encontrados

### **9) getAttributeTables**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer

Retorno: Array de String com os nomes das tabelas de atributo do layer

### **10) getTableFields**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer a ser pesquisado
- String com o nome da tabela de atributos

Retorno: Array de String com os nomes do fields da tabela

### **11) getObjectAttributes**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome da tabela de atributos
- String com o ID do objeto

Retorno: Array de String com os valores dos atributos do objeto

## **12) getWKT**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer
- String com o ID do objeto

Retorno: String com a representação Well Known Text (WKT) (OGC) da geometria do objeto

## **13) insertObject**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer
- String com a representação Well Known Text (WKT) (OGC)

Retorno: String com o ID do objeto inserido

## **14) deleteObject**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer
- String com o ID do objeto

Retorno: inteiro 1 se executado com sucesso 0 se ocorreu algum erro

## **15) updateGeometry**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome do layer
- String com o ID do objeto
- String com a representação Well Known Text (WKT) (OGC)

Retorno: inteiro 1 se executado com sucesso 0 se ocorreu algum erro

## **16) updateAttribute**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o nome da tabela de atributos
- String com o ID do objeto
- String com o nome do field a ser atualizado
- String com o novo valor

Retorno: inteiro 1 se executado com sucesso 0 se ocorreu algum erro

## **17) getQueryResult**

Parâmetros:

- String com os dados para conexão à base de dados
- String com o a query

Ex: "select nome from estados"

- String com o nome do campo a ser retornado no array

Ex: "nome"

Retorno: Array de String com as valores resultantes da query

## ANEXO III

### EXEMPLO DE UTILIZAÇÃO DO GEOAGENTE

As Figuras A.1, A.2 e A.3 demonstram o filtro espacial realizado pelo processamento do seguinte grupo de iterações:

- 1) Definição do Ponto
- 2) Definição da unidade de medida para a distância
- 3) Definição de um dos intervalos de distância



FIGURA A.2 – Área no início do processo de iterações



FIGURA A.3 – Área após o primeiro conjunto de iterações

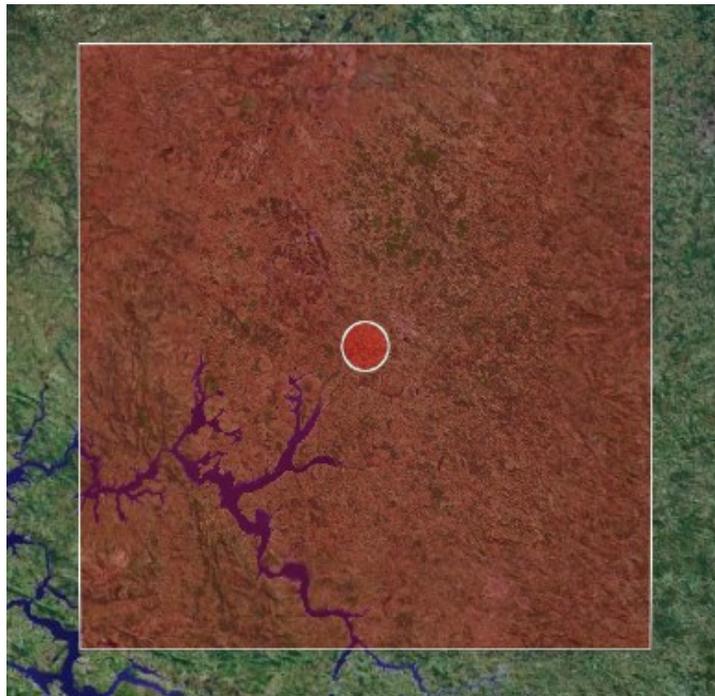


FIGURA A.4 – Sobreposição entre as áreas inicial e após o primeiro grupo de iterações

## ANEXO IV

### REQUISITOS PARA UM SISTEMA DE INFORMAÇÕES AO TURISTA

#### 1. Sistema de Informações ao Turista

O sucesso do desenvolvimento de um software é medido principalmente pela sua eficiência ao realizar a tarefa para o qual foi proposto. Para que o sucesso seja atingido é fundamental identificar e documentar as necessidades e propósitos do software (Lopes e Audy, 2003). Este processo de identificação e documentação se inicia pela busca de informações sobre a base de usuários prevista. Um passo seguinte é definir os processos dos usuários que o sistema pretende viabilizar ou automatizar. Portanto para que se desenvolva um sistema para o suporte ao turista deve-se entender o turista e os processos de decisão que o mesmo deve tomar (Perez, *et al.*, 2003).

#### 2. O Turista

De acordo com as definições da WTO (*World Tourism Organization*) o viajante é qualquer pessoa em uma viagem entre duas ou mais localidades e se divide em duas categorias: o visitante e outros viajantes. O visitante é definido como qualquer pessoa viajando para um local diferente de seu local habitual por menos de 12 meses e cujo motivo principal não seja o exercício de atividade remunerada. Ainda segundo a WTO o visitante se divide em dois grupos: os excursionistas que permanecem apenas durante o dia e os turistas que pernoitam ao menos uma noite no local.

#### 3. As motivações dos Turistas

De acordo com o modelo de motivação apresentado no trabalho de Andrade (2002) e desenvolvido por McIntosh, Goeldner & Ritchie existem quatro categorias de motivação para o turismo:

- Motivações físicas: considera as atividades que reduzem as tensões, como

relaxamento, esportes, saúde, etc;

- Motivações culturais: considera o desejo de conhecimento de outras culturas;
- Motivações interpessoais: inclui visita a parentes e amigos, desejo de conhecer outras pessoas;
- Motivação baseada no Status: desejo de reconhecimento por parte de outras pessoas, ou estímulo ao ego.

Baseado nestas motivações é possível deduzir que seria interessante que um sistema de informações ao turista disponibilizá-se as seguintes informações:

- Atividades físicas disponíveis, acompanhadas do nível de intensidade, duração e risco;
- Atividades culturais, tais como festas, museus e culinária;
- O perfil dos visitantes em geral;
- Se possível deveria fornecer algum tipo de ferramenta que permita ao visitante mostrar sua experiência a outras pessoas.

Segundo a WTO, os objetivos dos turistas ao iniciar uma viagem podem ser divididos segundo as seguintes categorias:

- Descanso, recreação e férias
- Visitar amigos e parentes
- Viagem de negócios ou relacionado a objetivos profissionais
- Tratamento de saúde
- Religião / Peregrinação
- Outros

Portanto seria interessante que o sistema pudesse classificar as atrações conforme o objetivo da viagem. Por exemplo, se o objetivo é religioso o sistema deveria mostrar as principais atrações relacionadas com este tema, sejam locais ou festas relacionados a religião. Se o objetivo é recreação deveria mostrar as atividades mais relacionadas a esportes ou festas, por exemplo.

#### **4. O processo de tomada de decisão**

De acordo com o modelo apresentado por Andrade (2002) citando Mathielson & Wall, o processo de tomada de decisões por parte do turista ocorre em cinco fases principais:

1. necessidade de sentir o desejo de viajar;
2. informação e avaliação dos possíveis produtos a serem consumidos;
3. decisões de viagem, onde ocorre efetivamente a seleção do destino;
4. preparativos para a viagem;
5. satisfação e avaliação da viagem que, certamente, influenciarão nas próximas decisões de viagem.

Assim um sistema de informações ao turismo em determinada região deve ser útil a partir do item 2, ou seja, informando o potencial visitante sobre as características, sobretudo as atrações, da região. Também deve conter grande parte das informações necessárias para que o turista possa preparar sua viagem, tais como telefones de hotéis e operadoras de turismo.

#### **5. Análise Orientada à Objetos**

##### **5.1 Requisitos**

Compilando as informações apresentadas nos itens anteriores, os seguintes requisitos foram estabelecidos para o sistema:

- a) Possuir recursos para armazenar informações sobre as características da região;

- b) Possuir recursos para consulta a estas informações;
- c) Possuir recursos para a edição destas informações;
- d) Possuir interfaces intuitivas que permitam que mesmo um usuário que nunca acessou o sistema possa interagir sem dificuldades;
- e) As informações que o sistema deve conter devem ser sobre as atrações, acomodação, alimentação e meios de transporte;
- f) Conter roteiros sugeridos e informações sobre pacotes turísticos;
- g) Possuir recursos que permitam ao turista contar sobre sua experiência para outros turistas;
- h) Possuir recursos administrativos para que um grupo de usuários com tal incumbência critique as informações fornecidas pelos usuários;
- i) Deve pensar em 4 diferentes tipos de usuários, ou, atores:
  1. O que busca informações gerais sobre o local porém ainda não decidiu se vai viajar para lá;
  2. O que já decidiu viajar e busca informações para planejar sua viagem;
  3. O que está viajando ou já viajou e quer emitir suas impressões e descobertas;
  4. O administrativo que tem a incumbência de criticar as informações fornecidas pelos turistas.
- j) As informações do local com algum vínculo geográfico devem possuir uma interface gráfica de representação: mapas interativos.

## 5.2 Casos de Uso

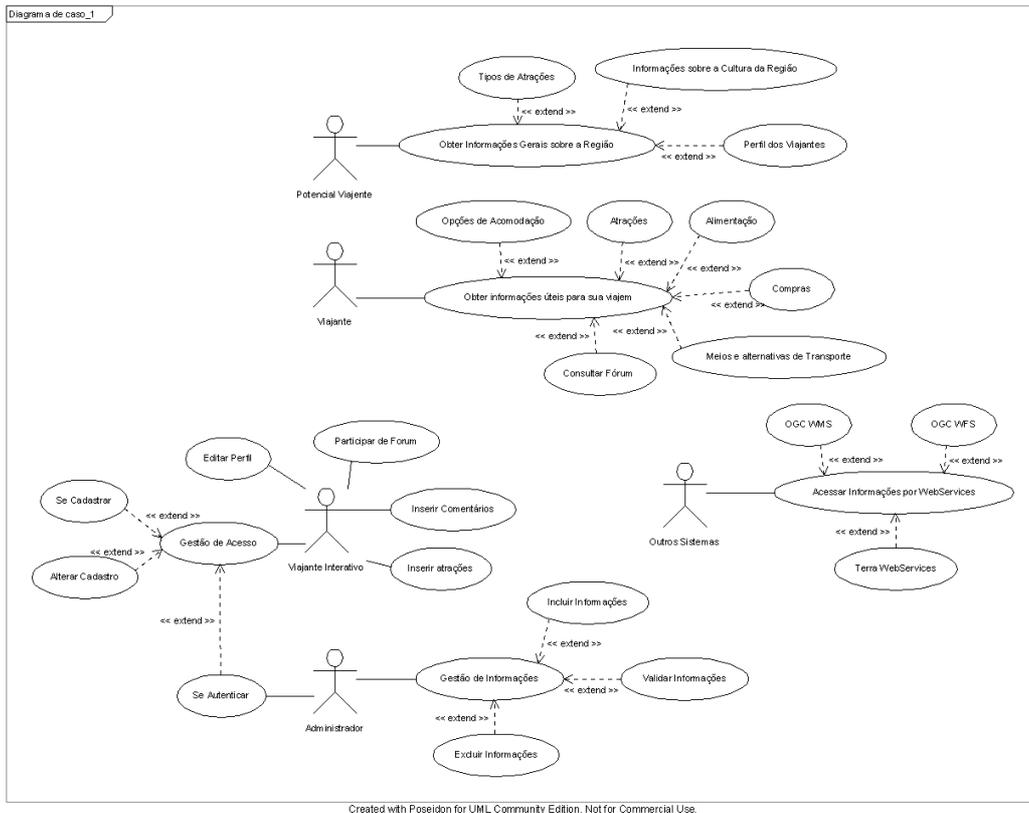


FIGURA A.5 – Casos de Uso

### 5.3 Diagrama de Classes

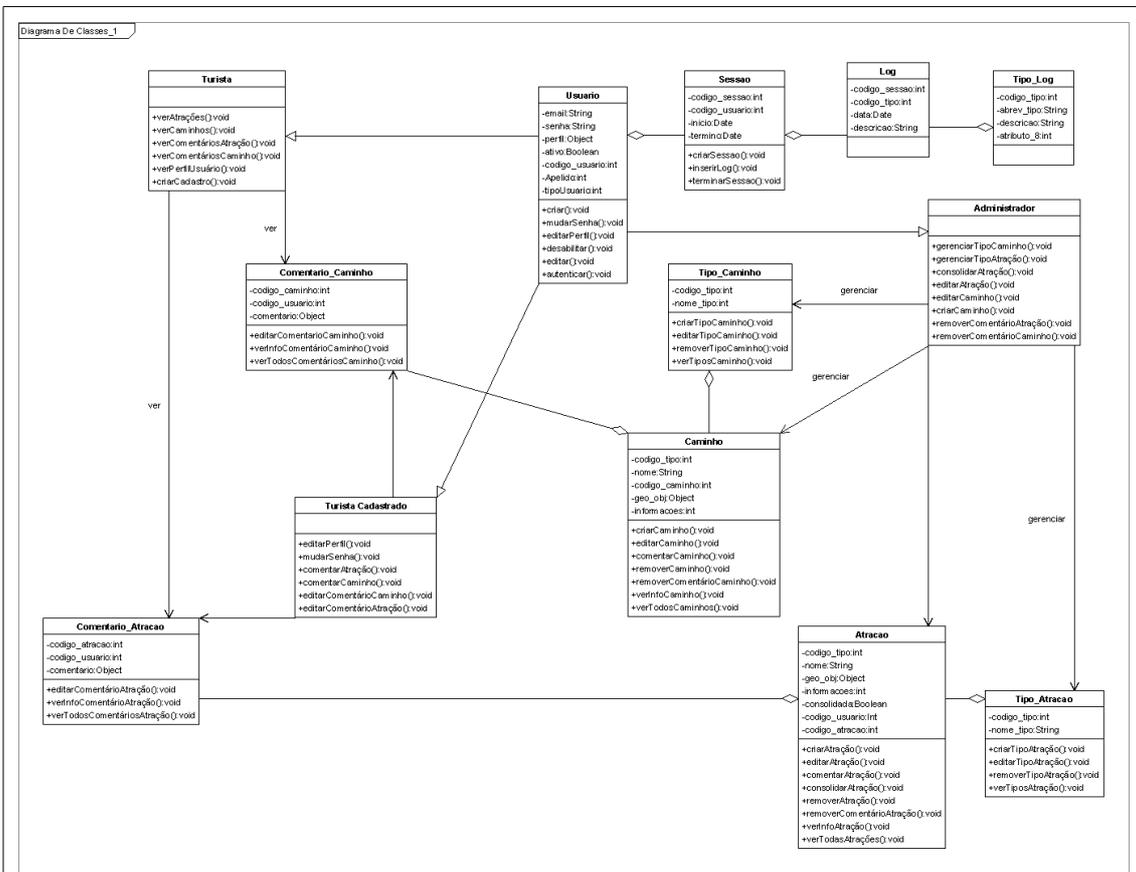


FIGURA A.6 – Diagrama de Classes