UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Presidente Prudente

Bruno César Vani

Concepção e Implementação de um Sistema de Controle de Arborização Urbana através da Integração de Softwares Livres e de Código Aberto

Presidente Prudente, 15 de Dezembro de 2011

Bruno César Vani

Concepção e Implementação de um Sistema de Controle de Arborização Urbana através da Integração de Softwares Livres e de Código Aberto

Trabalho de conclusão de curso apresentada ao Departamento de Matemática, Estatística e Computação (DMEC), da Universidade Estadual Paulista "Júlio de Mesquita Filho", sob o título **"Construção de um Sistema de Controle de Arborização Urbana através da Integração de Ferramentas Livres"**.

Orientador: Prof. Dr. Milton Hirokazu Shimabukuro

Presidente Prudente

15 de Dezembro de 2011

Termo de Aprovação

Aluno Bruno César Vani

Concepção e Implementação de um Sistema de Controle de Arborização Urbana através da Integração de Softwares Livres e de Código Aberto

Monografia sob o título **"Concepção e Implementação de um Sistema de Controle de Arborização Urbana através da Integração de Softwares Livres e de Código Aberto"**, defendida por Bruno César Vani e aprovada em 15 de Dezembro de 2011, em Presidente Prudente, Estado de São Paulo, pela banca examinadora constituída pelos doutores:

Prof. Dr. Milton Hirokazu Shimabukuro
FCT Unesp

Prof. Dr. Marco Antonio Piteri
FCT Unesp

Prof. Dr. Edilson Ferreira Flores
FCT Unesp

Presidente Prudente, 15 de dezembro de 2011

Agradecimentos

Agradeço primeiramente a Deus pela oportunidade de concluir este curso, superando todas as dificuldades.

Agradeço ao Professor Dr. Milton Hirokazu Shimabukuro e Me. Ítalo Tsuchya, pelo apoio e incentivo nos projetos que me possibilitaram grande aprendizado extracurricular.

Agradeço também aos amigos Vinicius Akira Suyama e Dallan Augusto Toledo Reis, pela parceria nos trabalhos em equipe ao longo do curso.

Por fim, agradeço à minha família e namorada pela paciência e compreensão.

Resumo

Ferramentas computacionais adequadas permitem construir aplicações capazes de vincular informações à sua localização física, bem como representá-la em um esquema visual e interativo, atingindo eficientemente o poder de comunicação visual. Isso faz com que o usuário sintetize informações de maneira simples e eficiente. A estas aplicações podemos atrelar a definição de Sistema de Informação Geográfica (SIG). Os SIG's abrangem diversos conceitos e disciplinas, com a finalidade principal de coletar, armazenar, visualizar e processar dados espaciais, obtendo assim as informações necessárias para tomada de decisões.

Em meio a este contexto, apresentamos neste trabalho a Concepção e a Implementação de um Sistema de Controle de Arborização Urbana através da Integração de Softwares Livres e de Código Aberto. Esta concepção surgiu a partir da necessidade de um Projeto Ambiental desenvolvido pela Casa da Agricultura do município de Regente Feijó, que tem como objetivos principais a catalogação e gerenciamento da arborização urbana do município.

Por abordar esta diversidade de conceitos, o desafio na construção deste sistema está na integração das plataformas que estão envolvidas em todas as suas etapas: coleta e armazenamento de dados, inclusão de mapas e demais informações espaciais, operações sobre as informações armazenadas, obtenção de resultados e visualização gráfica dos mesmos.

Após a implementação, foi possível propiciar ao usuário do sistema um aumento na capacidade de percepção na análise das informações, bem como facilitar o processo de tomada de decisões.

palavras-chave: Integração de *Software*. SIG. Servidor *Web* de Mapas.

Abstract

Suitable computacional tools allow to build applications that can link information to its physical location, and represent them into visual and interactive schemes, effectively reaching the power of visual comunication. This leads the user to synthesize information in a simple and efficient way. These applications are linked to the definition of Geographic Information System (GIS). GIS are comprised by many concepts and tools, which have the main purpose of collecting, storing, viewing and processing spatial data, obtaining the information needed for decision making.

Within this context, this paper presents the Conception and Implementation of a Control System for Urban Forestry through Integration of Free and Open Source Software. This conception arose from the need of an Environmental Project developed by the Agriculture's House of the city of Regente Feijó, which has as main objectives cataloging and management of urban afforestation of the municipality.

Due to this diversity of concepts, the challenge in building this system is the integration of platforms that are involved in all stages: collecting and storage of data, including maps and other spatial information, operations on the stored information, obtaining results and graphical visualization of the same.

After implementation, it was possible to provide for the system users an improvement in the capacity of perception in the information analysis and facilitate the process of decision making.

keywords: Software Integration. GIS. Web Map Service.

Sumário

1 Introdução p. 12

1.1 Visão Geral . . . . . p. 12

1.2 Objetivos do Trabalho . . . . . p. 13

1.3 Organização do Trabalho . . . . . p. 13

2 Conceitos e Ferramentas Computacionais p. 14

2.1 Informações Espaciais . . . . . p. 14

2.2 Dados Espaciais . . . . . p. 15

2.3 Geoprocessamento e Sistema de Informação Geográfica . . . . . p. 16

2.4 Banco de Dados (BD) e Banco de Dados Geográfico . . . . . p. 18

2.4.1 Sistema Gerenciador de Banco de Dados (SGBD) . . . . . p. 18

2.4.2 PostgreSQL e PostGIS . . . . . p. 18

2.5 Arquitetura Cliente-Servidor . . . . . p. 19

2.5.1 Cliente e Servidor Web . . . . . p. 19

2.5.2 Servidor Web Apache . . . . . p. 20

2.6 Servidor de Mapas . . . . . p. 20

2.7 *Web mapping* . . . . . p. 21

2.8 Software Livre e de Código Aberto . . . . . p. 21

2.9 MapServer . . . . . p. 22

2.9.1 MapServer CGI . . . . . p. 23
2.9.2 MapServer MapScript . . . . . p. 24
2.9.3 Mapfile . . . . . p. 24
2.10 Shapefile . . . . . p. 26
2.11 p.mapper . . . . . p. 26
2.12 PHP . . . . . p. 27
2.13 Javascript e jQuery . . . . . p. 27
2.14 PHPlot . . . . . p. 27

3 Concepção do Projeto p. 28
3.1 Obtenção e Análise de Requisitos . . . . . p. 28
3.2 Modelagem . . . . . p. 32

4 Integração das Ferramentas p. 33
4.1 Estrutura da Aplicação . . . . . p. 33
4.2 Banco de Dados e MapServer . . . . . p. 33
4.2.1 Criando um banco de dados espacial com PostgreSQL/PostGIS p. 34
4.2.2 Garantindo a Interoperabilidade com o MapServer . . . . . p. 35
4.2.3 Inserção de dados de natureza espacial . . . . . p. 36
4.2.4 Consulta a dados de natureza espacial . . . . . p. 37
4.3 Interface de Navegação e Edição - PHP/mapscript . . . . . p. 37
4.3.1 Mapa dinâmico . . . . . p. 37
4.3.2 Propriedade “EXTENT” de um mapa . . . . . p. 38
4.3.3 Zoom . . . . . p. 41
4.3.3.1 Zoom pontual . . . . . p. 42
4.3.3.2 Zoom por rolagem de mouse . . . . . p. 42
4.3.3.3 Zoom por retângulo . . . . . p. 43

4.3.4 Pan . . . . . p. 44
4.3.5 Opção novo ponto de referência . . . . . p. 46
4.3.6 Opção de Edição de Ponto de Referência . . . . . p. 46
4.3.7 Opção Nova Árvore . . . . . p. 49
4.3.8 Ajuste de Árvore . . . . . p. 50
4.3.9 Listar Árvore . . . . . p. 50
4.3.10 Opção Atualizar/Limpar . . . . . p. 51
4.3.11 Opção *Home* . . . . . p. 51
4.3.12 Interface de seleção de Layers e Legendas . . . . . p. 51
4.3.13 Mapa de Referência e Barra de Escala . . . . . p. 51
4.4 Interface para Dados Descritivos . . . . . p. 52
4.4.1 Gerenciamento de Espécies e Catálogos de Origem . . . . . p. 53
4.4.2 Exibição das Vistorias . . . . . p. 53
4.4.3 *Upload* de imagens . . . . . p. 54
4.4.4 Relatórios e Gráficos . . . . . p. 55
4.4.4.1 Gráfico de Distribuição de Espécies e de Sanidade . . . p. 56
4.4.4.2 Médias dos parâmetros específicos . . . . . p. 57
4.4.4.3 Estimativa de Área Verde . . . . . p. 57
4.4.4.4 Relação de Substituições . . . . . p. 57
4.5 Experimentos e Retorno dos Usuários . . . . . p. 58

5 Conclusão p. 59

5.1 Considerações Finais . . . . . p. 59
5.2 Futuros Trabalhos . . . . . p. 60

Referências p. 61

Lista de Figuras

1 Representação das estruturas matricial (a) e vetorial (b) . . . . . p. 16

2 Arquitetura de Sistema de Informação Geográfica. Fonte: (CAMARA et al., 1996). . . . . p. 17

3 Arquitetura Cliente-Servidor. Fonte: (KUROSE; ROSS, 2006). . . . . p. 19

4 Arquitetura de uma aplicação MapServer. Fonte: (MAPSERVER, 2010). p. 23

5 Visualizador p.mapper. Fonte:(P.MAPPER, 2011). . . . . p. 26

6 Diagrama de classes do Sistema . . . . . p. 32

7 Arquitetura da aplicação MapServer para este contexto. Adaptada de (MAPSERVER, 2010). . . . . p. 34

8 Interface de navegação e edição implementada . . . . . p. 38

9 Demonstração da propriedade extent . . . . . p. 39

10 Comportamento da funcionalidade *pan* numa posição clicada . . . . . p. 44

11 Comportamento da funcionalidade *pan* na coordenada de destino . . . p. 45

12 Operações de deslocamento para satisfazer a funcionalidade pan . . . . p. 46

13 Opção de novo ponto de referência . . . . . p. 47

14 Opção de ajuste de ponto de referência . . . . . p. 48

15 Opção de nova árvore com ajuste manual de coordenadas . . . . . p. 50

16 Interface de seleção de *layers* . . . . . p. 52

17 Mapa de Referência e Barra de Escala . . . . . p. 52

18 Gerenciador de Espécies . . . . . p. 53

19 Interface de criação de espécies - seletor de cores . . . . . p. 53

20 Interface de vistorias de uma árvore . . . . . p. 54

21 Interface de seleção para relatórios e gráficos . . . . . p. 55

22 Interface de relatórios e gráficos . . . . . p. 55

Lista de Listagens

1 Exemplo de Mapfile . . . . . p. 24
2 Instrução SQL para criação de um banco de dados espacial . . . . . p. 34
3 Instrução SQL para criação de uma tabela com dado espacial . . . . . p. 35
4 Instrução SQL para criação de uma tabela com dado espacial . . . . . p. 36
5 Instrução SQL para criação de uma tabela com dado espacial . . . . . p. 37
6 Execução do utilitário para obtenção da extensão de um mapa . . . . . p. 39
7 Função para conversão de coordenadas de imagem em coordenadas geográficas . . . . . p. 40
8 Função atualizar extent . . . . . p. 40
9 Bloco de execução de *zoom* pontual . . . . . p. 42
10 Javascript imgAreaSelect . . . . . p. 43
11 Consultas para seleção de ponto . . . . . p. 48
12 Uso da biblioteca WideImage . . . . . p. 54
13 Consulta para distribuição de espécies . . . . . p. 56
14 Consulta para distribuição de sanidade . . . . . p. 56
15 Consulta para obtenção de médias . . . . . p. 57
16 Consulta estimativa de área verde . . . . . p. 57
17 Consulta para relação de substituições . . . . . p. 58

Capítulo 1

Introdução

1.1 Visão Geral

A manipulação de informações espaciais está presente em parte das aplicações envolvendo banco de dados em geral. Como o volume de informações desse tipo cresce a todo tempo, o oferecimento de mecanismos complementares na manipulação de dados auxilia nos processos de tomada de decisão.

Ferramentas computacionais adequadas permitem construir aplicações capazes de vincular a informação à sua localização física, bem como representá-la em um esquema visual e interativo, atingindo eficientemente o poder de comunicação visual. Isso faz com que o usuário sintetize informações de maneira simples e eficiente.

A estas aplicações pode-se atrelar a definição de Sistema de Informação Geográfica (SIG). Os Sistemas de Informações Geográficas abrangem diversos conceitos e disciplinas, com a finalidade principal de coletar, armazenar, visualizar e processar dados espaciais, obtendo assim as informações necessárias para tomada de decisões. Além de trabalhar com estruturas capazes de armazenar as feições geométricas (polígonos, por exemplo) e localizações dos dados, um SIG deve também ser capaz de trabalhar com os atributos convencionais (alfanuméricos) que completam a descrição das informações, propiciando uma integração harmônica entre ambos.

1.2 Objetivos do Trabalho

Em meio a este contexto de informação *versus* localização, utilizamos a idéia de construir um sistema de controle de arborização urbana, que surgiu a partir da necessidade de um Projeto Ambiental desenvolvido pela Casa da Agricultura do município de Regente Feijó/SP. Um grande trabalho de campo foi realizado, tendo como resultado um catálogo com informações específicas, de localização e fotos das árvores do perímetro urbano da cidade.

Com base neste catálogo, o foce deste trabalho é conceber e implementar um sistema de controle, representando as árvores visualmente no mapa da cidade. As funcionalidades requeridas pelo mesmo foram elaboradas a partir da necessidade do Projeto.

Por abordar esta diversidade de conceitos, o desafio na construção deste sistema está na integração das plataformas que estão envolvidas em todas as suas etapas: armazenamento e manipulação dos dados, inclusão de mapas e demais informações espaciais, obtenção de resultados e visualização gráfica dos mesmos. Todas estas etapas aqui resumidas fazem partes do problema a ser solucionado como proposta deste trabalho de conclusão de curso.

1.3 Organização do Trabalho

O Capítulo 2 contém a revisão teórica e prática, no qual são definidos conceitos e ferramentas importantes para a implementação do sistema proposto.

O Capítulo 3 destina-se à definição do projeto: obtenção e análise dos requisitos e modelagem.

O Capítulo 4 exibe as soluções encontradas para a resolução do problema proposto.

Por fim, no Capítulo 5, uma conclusão é apresentada, destacando-se os resultados obtidos e a perspectiva de trabalhos futuros.

Capítulo 2

Conceitos e Ferramentas Computacionais

Neste Capítulo, os conceitos e ferramentas computacionais para a solução do problema proposto como tema deste Trabalho de Conclusão de Curso são apresentados. São explanados alguns conceitos específicos para se compreender o uso de banco de dados espaciais. Também são definidas algumas ferramentas utilizadas na manipulação deste tipo de dado, bem como alguns conceitos acerca de arquiteturas de aplicações que os utilizam.

2.1 Informações Espaciais

Inicialmente, cabe ressaltar que os termos informações geoespaciais e informações geográficas também são utilizados. Conforme trechos publicados pela Comunicação Social do Instituto Brasileiro de Geografia e Estatística (IBGE) em seu site oficial, obtem-se uma boa definição para o conceito e uso de informações espaciais:

> Informações geoespaciais são, por exemplo, os dados cartográficos e topográficos que representam o território, as imagens de satélites, ortofotos, as malhas que representam a infraestrutura de transportes, a localização e descrições que representam as áreas protegidas, a descrição e representação dos imóveis urbanos e rurais, assim como os distintos usos do solo. Também são consideradas informações geoespaciais as de natureza estatística que descrevem aspectos demográficos, bem como a distribuição da população ou suas variáveis sócio-econômicas. [...] A valorização da informação geoespacial decorreu da crescente preocupação, em nível global, com a conservação do meio ambiente e das demandas sociais e econômicas por uma melhor compreensão da realidade territorial. No início dos anos 90, a Agenda 21, documento final da Conferência das Nações Unidas para o Meio Ambiente e Desenvolvimento - realizada no Rio de Janeiro - em sua Seção IV, Capítulo 40, intitulado "Informação para a Tomada de Decisão", enfatizou a necessidade de se incrementar as atividades de aquisição, avaliação e análise de dados utilizando novas tecnologias tais como: Sistema de Informações Geográficas (SIG), Sensoriamento Remoto (SR) e Sistema

> de Posicionamento Global. Informações geoespaciais são necessárias em quase toda atividade de planejamento e administração. Mapas e dados associados a localizações são usados no cotidiano para planejamento e gestão de recursos, oferta de serviços nos setores público e privado; e também na elaboração de políticas públicas. É por esta razão que o uso do Geoprocessamento e da Geotecnologia torna-se essencial para gestão, análise, tomada de decisão em uma imensa gama de atividades. Os dados digitais são o pré-requisito para o manejo de grandes volumes de dados, em diversas formas, e para o uso de técnicas de visualização que permitam o entendimento do passado e do presente, e a projeção do futuro. Estando em formato digital, os dados espaciais podem ser armazenados, manipulados, integrados, editados, gerar novas informações por processamento, e distribuídos entre pesquisadores e cidadãos.(IBGE, 2010).

Como se percebe, a disseminação das informações espaciais reflete diretamente em variadas áreas de interesse geral da população, expandindo o acesso à informação.

2.2 Dados Espaciais

Dados espaciais, também chamados de geoespaciais ou geográficos, são utilizados para representação de informações espaciais. Este tipo de dado pode ser dividido em dois componentes, conforme definem Rigaux, Scholl e Voisard (2002): um componente descritivo (para informações tais como nome e população de uma cidade, dentre outros) e um componente espacial (localização em certo espaço geográfico e sua forma geométrica).

Pode-se citar duas classes de representação para este tipo de informação, conforme descrevem Camara, Davis e Monteiro (2004):

- Estrutura Matricial: também conhecida como *raster*, este tipo de estrutura utiliza uma matriz de duas dimensões para representação de uma informação espacial. Cada célula (ou *pixel*, para o caso da tela) desta matriz representa uma coordenada da informação representada, logo, o elemento da célula mantém um valor capaz de identificar a informação mapeada naquele ponto. Estes valores que as células armazenam podem ser, por exemplo, números inteiros limitados em um intervalo (como exemplo, valores entre 0 e 255 para representação de imagens em 8 bits). Pode-se destacar como vantagens deste tipo de estrutura a sua simplicidade e facilidade para implementação de operações com os dados nela armazenados, uma vez que todas as células possuem o mesmo tipo e tamanho. No entanto, este tipo de estrutura pode causar perda de precisão, uma vez que para a representação de estruturas contínuas (uma curva, por exemplo), é necessário fragmentá-la em *pixels*,

e a perda de qualidade pode ser significante. Como alternativa, pode-se aumentar a quantidade de células. Neste caso, as perdas serão menores, contudo, mais espaço de armazenamento será utilizado, resultando-se em mais custo de processamento;

- Estrutura Vetorial: este tipo de estrutura utiliza basicamente pontos, linhas e polígonos para representação. Os pontos representam entidades por um par de coordenadas. As linhas são formadas por segmentos originados por pelo menos dois pontos. Já os polígonos, são formados por linhas que delimitam uma área. Como vantagens da representação vetorial, são destacadas a maior precisão e a facilidade para representação de objetos de naturezas variadas e irregulares, tais como a delimitação de uma cidade ou de um rio. Cabe ressaltar que a implementação de operações sobre este tipo de dado são complexas.

A seguir, a imagem de um mesmo polígono criado utilizando-se as estruturas matricial - Figura 1 a) - e vetorial - Figura 1 b) -, evidenciando-se as características citadas:

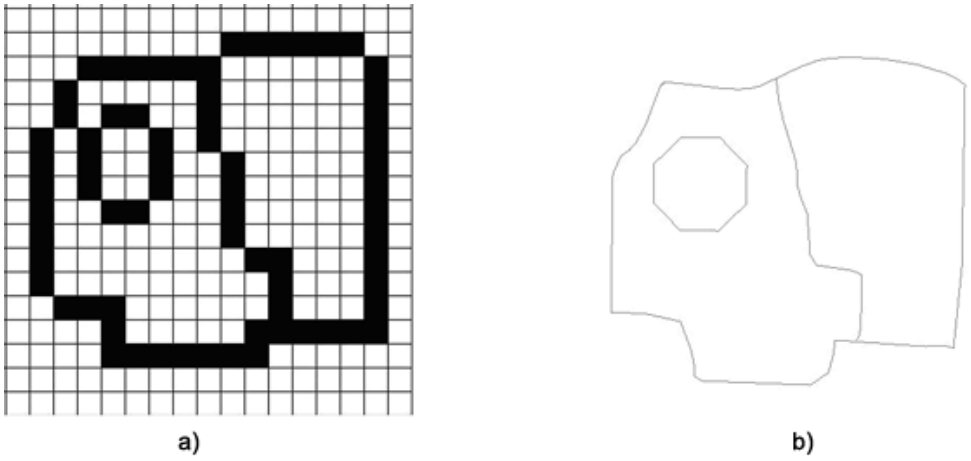

Figura 1: Representação das estruturas matricial (a) e vetorial (b)

2.3 Geoprocessamento e Sistema de Informação Geográfica

Camara, Davis e Monteiro (2004) definem que "o termo *Geoprocessamento* denota a disciplina do conhecimento que utiliza técnicas matemáticas e computacionais para o tratamento da informação geográfica[...]".

Um Sistema de Informação Geográfica abrange diversos conceitos, ferramentas, disciplinas e tecnologias. Sua característica principal é o trabalho com dados espaciais. Pode

ser utilizado em diversas áreas, tais como: Geografia, Sensoriamento Remoto, Planejamento Urbano, Segurança Pública e Engenharia de Tráfego. Esta abordagem interdisciplinar de um SIG possibilita o surgimento de novas tecnologias e aplicações, possibilitando ao usuário obter maneiras mais aprofundadas de interagir com as informações armazenadas, além de destacar uma importante característica inerente ao ser humano: a comunicação visual.

Os componentes da estrutura básica de um SIG propostos por Camara et al. (1996) são mostrados na Figura 2.

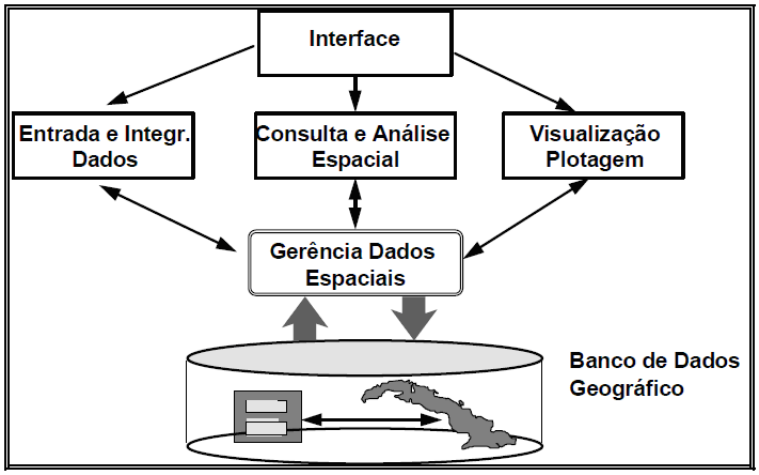

Figura 2: Arquitetura de Sistema de Informação Geográfica. Fonte: (CAMARA et al., 1996).

Na camada base (Banco de Dados Geográfico), há o repositório dos dados espaciais, cujas definições serão detalhadas adiante. As operações sobre estes dados são de responsabilidade da camada superior (Gerência de Dados Espaciais), através do Sistema Gerenciador de Banco de Dados (SGBD) utilizado. Logo acima, aparecem as operações realizadas no SIG, que envolvem a entrada e edição dos dados, realização de consultas e análises, bem como os esquemas de visualização dos mesmos. Por fim, no topo da estrutura, é apresentada a interface na qual as operações são realizadas pelo usuário. Atualmente, há diversas plataformas livres que possibilitam a criação de um SIG. Algumas delas são inclusive voltadas a aplicações *web*, que garantem maior portabilidade através do uso de um navegador de Internet.

2.4 Banco de Dados (BD) e Banco de Dados Geográfico

Em Date (2000), pode-se abstrair que um banco de dados é um conjunto de dados persistentes que servem de base para uma aplicação. O termo "persistente" define que estes dados são mantidos em certo repositório para estarem à disposição do usuário da aplicação a qualquer tempo. A capacidade de armazenamento e manipulação de dados geográficos é o componente adicional que caracteriza um **banco de dados geográfico**.

2.4.1 Sistema Gerenciador de Banco de Dados (SGBD)

Date (2000) também define SGBD como um sistema que permite ao usuário armazenar, consultar e alterar dados no Banco de Dados. A arquitetura básica de um SGBD envolve basicamente quatro componentes:

1. Dados: informações representadas do mundo real;
2. Hardware: dispositivos de armazenamento permanente onde ficarão armazenados os dados;
3. Software: elo entre os dados e o hardware, e realização das operações sobre os dados que estão armazenados no dispositivo de armazenamento permanente;
4. Usuários: desenvolvedores e demais utilizadores da aplicação final.

2.4.2 PostgreSQL e PostGIS

O PostgreSQL é um SGBD de código-fonte aberto muito utilizado no meio acadêmico e científico. Seu projeto foi lançado há mais de quinze anos e novas versões vêm sendo incorporadas, propiciando suporte a novas tecnologias e melhorias de desempenho. Seu projeto colaborativo permite que usuários de todo o mundo contribuam para seu desenvolvimento e suporte.

O PostGIS é um projeto que "adiciona suporte a dados espaciais ao PostgreSQL, possibilitando então utilizá-lo como um Banco de Dados Geográfico em um Sistema de Informações Geográficas", como destacado no site oficial (http://postgis.refractions.net/ - tradução nossa). Também de natureza livre, este módulo extra do PostgreSQL apresenta diversas funcionalidades e é aberto para contribuições dos usuários.

2.5 Arquitetura Cliente-Servidor

Considerando um ambiente de rede de computadores, Kurose e Ross (2006) definem que neste tipo de arquitetura há uma aplicação servidora que deve estar sempre em funcionamento. Ela será responsável por receber requisições de aplicações cliente (que não necessitam se manter em funcionamento o tempo todo) e enviar as respostas a estas requisições. Cabe ressaltar que cliente e servidor funcionam em sistemas finais diferentes.

No contexto da *web*, há uma aplicação cliente, à qual é denominada cliente *web*, e uma aplicação servidora, à qual é denominada servidor *web*. A maneira pela qual estas aplicações se comunicam está definida no Protocolo de Transferência de Hipertexto (HTTP), que define a estrutura das mensagens que são trocadas pelos programas cliente e servidor. A troca de mensagens é a maneira com que estes programas se comunicam, num ciclo de requisições e respostas. Na Figura 3, a representação desta estrutura proposta por Kurose e Ross (2006) é apresentada, onde um servidor recebe requisições de dois clientes:

Figura 3: Arquitetura Cliente-Servidor. Fonte: (KUROSE; ROSS, 2006).

2.5.1 Cliente e Servidor Web

Kurose e Ross (2006) também definem que o cliente *web* é o responsável por enviar ao servidor as mensagens de requisição HTTP. Um navegador de internet, também conhecido como *browser*, é um programa que, além de exibir conteúdos de uma página *web*, implementa o lado cliente *web*, ou seja, envia requisições HTTP a um servidor *web*. Pode-se citar como exemplos de *browsers* o "Mozilla Firefox" e o "Internet Explorer". Já um servidor *web* é o responsável por receber as requisições HTTP do cliente *web* e enviar as mensagens de resposta HTTP. Como exemplo de servidores *web*, pode-se citar o

"Apache' e o "Microsoft Internet Information Server". No contexto da *web*, o objeto final destas requisições e respostas são arquivos (comumente, páginas HTML, imagens, dentre outros).

2.5.2 Servidor Web Apache

O "Apache HTTP Server Project", conforme denominação oficial, é um "projeto de software desenvolvido através de esforço colaborativo que visa a criação de um servidor *web* robusto, de grau comercial, com mais recursos e de código-fonte aberto" (APACHE, 2011). Este projeto é gerenciado em conjunto por um grupo de voluntários espalhados pelo mundo, e faz parte da "Apache Software Fundation". Os usuários comuns também contribuem para o desenvolvimento do projeto, através do envio de sugestões, código ou documentação.

2.6 Servidor de Mapas

Conforme Peng e Tsou (2003), um servidor de mapas é o componente capaz de gerar mapas através da realização de consultas espaciais baseadas em requisições do usuário. Algumas características podem ser citadas, tais como:

- Geração de mapas através de requisições do usuário;
- Realização de consultas básicas acerca do conteúdo do mapa e de atributos de características espaciais;
- Comunicação com outros programas e serviços, tais como SGBD's ou outros servidores de mapa;
- Extração de dados que estejam armazenados em um Banco de Dados para geração de mapas;
- Interface gráfica e simbologia que permitam aos usuários um resultado satisfatório;
- O mapa produzido, normalmente, tem uma das formas que seguem:

1. Uma simples imagem (em formato GIF, JPEG, dentre outros);

2. Uma composição gráfica mais complexa, envolvendo estilos, legendas, camadas, símbolos, dentre outros componentes;
3. Um componente em formato vetorial ou raster, de forma que possa ser manipulado pelo cliente, com funcionalidades extras, como consultas, zoom, demarcações, dentre outras.

2.7 *Web mapping*

O conceito de *web mapping* está relacionado à publicação de mapas através da Internet. Peng e Tsou (2003) apresentam um breve histórico:

1. Publicação de Mapas Estáticos (arquivo de imagem em formato "jpg", "png", dentre outros);
2. *Web mapping* estático — Interatividade básica através de formulários HTML (imagem e HTML);
3. *Web mapping* dinâmico — DHTML (HTML dinâmico) — adição de recursos que incrementam o dinamismo do mapa.

2.8 Software Livre e de Código Aberto

No site oficial da *Free Software Foundation* (Fundação para o *Software* Livre - http://www.fsf.org/ - Acesso em 15/05/2011) é definido como livre o *software* que pode ser compartilhado, copiado, adaptado, estudado e usado livremente e sem restrições.

Já em OSI (2011), são apresentadas as características de um *Software* de código-aberto, que não estão associadas somente à disponibilização do código-fonte:

1. Livre distribuição;
2. Código-fonte incluso;
3. Modificações e trabalhos derivados são permitidos;
4. A integridade do Autor deve ser preservada;
5. Sem restrições de uso a pessoas ou grupos;

6. Sem restrições de uso a campos de trabalho;
7. Distribuição da licensa a todos que utilizem;
8. Sub-componentes do *software* sob a mesma licença;
9. Licença sem restrições a outros *softwares*;
10. Licença de tecnologia neutra - sem embasamento em tecnologias ou estilos de interfaces individuais.

Além das facilidades relacionadas ao uso e distribuição dos *Softwares* Livres, optou-se neste trabalho pela escolha de ferramentas que pertençam a este contexto para propiciar o desenvolvimento e compartilhamento de novas tecnologias e linhas de pesquisa.

2.9 MapServer

O MapServer é um projeto de servidor de mapas de código-fonte aberto que tem como objetivo principal a criação dinâmica de mapas para serem visualizados através de um navegador de Internet. Abaixo, são descritas algumas de suas características principais, conforme descreve MapServer (2010):

- Visualização e consultas a dados matriciais, vetoriais e descritivos;
- Interface com SGBD's, tais como o PostgreSQL e outras fontes de dados (tais como *web services* e *shapefiles*);
- Suporte a vários sistemas operacionais (Windows, Linux, Mac OS X, etc);
- Suporte a várias linguagens de programação (PHP, Python, Perl, Ruby, Java, .NET);
- Alta qualidade de processamento e saídas personalizáveis.

A Figura 4, adaptada de MapServer (2010), ilustra a arquitetura básica de uma aplicação MapServer, com os diversos tipos de tecnologias que podem ser incluídas.

Na parte superior (entrada), o esquema indica que os dados para uma aplicação podem ser provenientes de repositórios de dados (arquivos com dados matriciais e vetoriais, que podem ser mantidos por um SGBD) ou de *web services*, que são tecnologias utilizadas para comunicação entre sistemas de diferentes plataformas através da *web*.

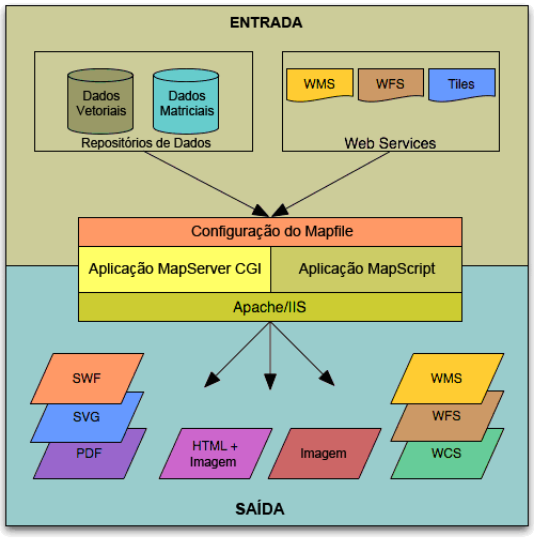

Figura 4: Arquitetura de uma aplicação MapServer. Fonte: (MAPSERVER, 2010).

Na camada central da arquitetura aparece a configuração do mapfile - arquivo texto de configuração que serve de base para a aplicação MapServer - a aplicação MapServer implementada em modo CGI ou modo MapScript, cujas definições serão abordadas adiante, rodando em um servidor *web*, como o "Apache" ou o "IIS (*Microsoft Internet Information Service*)".

Por fim, na camada inferior, aparece a saída, que pode ser uma imagem (várias extensões de arquivo são suportadas), uma composição de página HTML e imagem ou até mesmo interfaces WMS, WFS e WCS (*Web Map Services*, *Web Feature Services* e *Web Coverage Server*, respectivamente). As definições e padrões para desenvolvimento destas interfaces são definidos pelo OGC (*Open Geospatial Consortium*), consórcio internacional que define padrões para criação de objetos espaciais e serviços baseados em localização. As definições podem ser encontradas no site da entidade - http://www.opengeospatial.org (Acesso em 10/09/2011).

2.9.1 MapServer CGI

Numa aplicação em modo CGI (*Common Gateway Interface*), um programa executável fica à disposição no servidor *web* da aplicação, e recebe parâmetros para a geração de uma saída.

Logo, no MapServer modo CGI, o programa executável do MapServer (mapserv.exe) deve ficar à disposição no servidor *web* da aplicação, e será responsável pela criação dos

mapas e dados de saída. A aplicação requisita a criação de um mapa, para tanto, envia os parâmetros necessários para geração do mesmo através de um endereço pelo URL (*Uniform Resource Locator* - em português, localizador universal de recurso). Em um URL, o endereço de um recurso em certo servidor é informado pelo lado cliente de uma aplicação, bem como os parâmetros para a execução deste recurso. Por fim, o programa executável recebe os parâmetros e gera a respectiva saída.

2.9.2 MapServer MapScript

No modo MapScript (que funciona sem intervenção do modo CGI), o MapServer é utilizado em conjunto com outras linguagens de programação, tais como PHP, Perl, Python, Ruby, Tcl, Java e .NET, através da inclusão de módulos e bibliotecas. Neste modo, é possível explorar recursos adicionais da linguagem de programação utilizada, sendo possível se obter aplicações mais dinâmicas e complexas.

2.9.3 Mapfile

O mapfile é um arquivo texto de configuração (deve ter a extensão ".map") que serve de base para a aplicação MapServer. Este arquivo descreve todos os atributos que serão dispostos no mapa, tais como layers, cores, legendas, formato do arquivo de saída, dentre outros, bem como onde estão os dados que servirão de base para criação deste mapa. Este arquivo define os atributos de um mapa estruturados como objetos, o que mantém a implementação organizada e facilmente personalizável.

Na Listagem 1, um exemplo de mapfile é apresentado.

Listagem 1: Exemplo de Mapfile

```
1  MAP #início do objeto mapa
2
3          #opções do mapa e do arquivo de saída
4          EXTENT -73.99 -33.75 -28.83 5.27
5          SIZE 600 400
6          IMAGECOLOR 200 200 230
7          SHAPEPATH "../shapefiles"
8
9          LAYER
10                 NAME "brasil"
11                 TYPE polygon
12                 STATUS default
```

```
13                  DATA "55BR2500gc_SIR"
14
15                  CLASS
16                          NAME "Limite nacional"
17                          STYLE
18                                  COLOR 250 250 190
19                                  OUTLINECOLOR 0 0 0
20                          END
21                  END
22          END
23
24  END #final do objeto mapa
```

No mapfile da Listagem 1, pode-se observar os aspectos do respectivo mapa de saída. Inicialmente, a definição do objeto principal "MAP", delimitado ao final por "END". Em seu escopo, são definidos parâmetros específicos, como a extensão do mapa em coordenadas, tamanho da imagem de saída, cor de fundo e caminho dos dados ("EXTENT", "SIZE", "IMAGECOLOR" e "SHAPEPATH", respectivamente). Dentro do objeto principal ocorre a definição de um "LAYER". Em um mapa, pode-se ter a representação de várias informações. Por exemplo, consideremos uma cidade: pode-se ter num mapa que a represente, além da delimitação geográfica, as ruas, as quadras, as casas, os pontos turísticos, os hospitais, etc. Como há várias categorias de informações, pode-se dividi-las em camadas, também chamadas de *layers*. Esta divisão é utilizada no MapServer, e apresenta várias vantagens, dentre as quais pode-se citar a possibilidade de programação estruturada (códigos-fonte mais legíveis), reuso de informações e aplicações com representações visuais bem definidas, onde o usuário pode optar por visualizar os layers individualmente ou agrupados a seu critério. No *layer* da Listagem 1, são definidos parâmetros específicos, tais como nome, tipo de dado, visibilidade e arquivo de origem ("NAME", "TYPE", "STATUS" e "DATA", respectivamente). Por fim, em um *layer* deve-se definir uma ou mais classes de representações temáticas, onde são definidos estilos de representação temática. No exemplo já citado, é definida uma classe de representação (objeto "CLASS"), onde é definido um estilo composto por cor de fundo e cor de linha ("COLOR" e "OUTLINECOLOR").

2.10 Shapefile

Regulamentado pela “Environmental Systems Research Institute, Inc.” (ESRI), o shapefile é um tipo de arquivo bastante utilizado para Sistemas de Informação Geográfica. Embora pertença a uma corporação privada, sua especificação, está disponível gratuitamente, e permite que outras ferramentas o implementem. Este tipo de arquivo representa as estruturas espaciais através de um conjunto de dados em vários formatos, cujos principais possuem a extensão “.shp”, “.dbf” e “.shx”. Estes arquivos indicam seus atributos não-espaciais e suas geometrias, conforme definido em ESRI (1998).

2.11 p.mapper

O p.mapper é uma ferramenta para visualização de aplicações MapServer baseado em php/mapscript, com recursos javascript. Possui diversos *plugins* configuráveis, dentre eles, para realização de consultas de atributos descritivos. Sua configuração com o MapServer é realizada através de um arquivo XML (*Extensible Markup Language*) - uma linguagem de demarcação destinada principalmente a compartilhamento de informações. É possível editar a interface de acordo com os *layers* utilizados na apliação desejada, basta efetuar as configurações de acordo com o mapfile utilizado. De natureza livre, pode ser baixado gratuitamente em seu site oficial (http://www.pmapper.net/ - Acesso em 12/11/2011). Na Figura 5, uma demonstração de interface do p.mapper é apresentada.

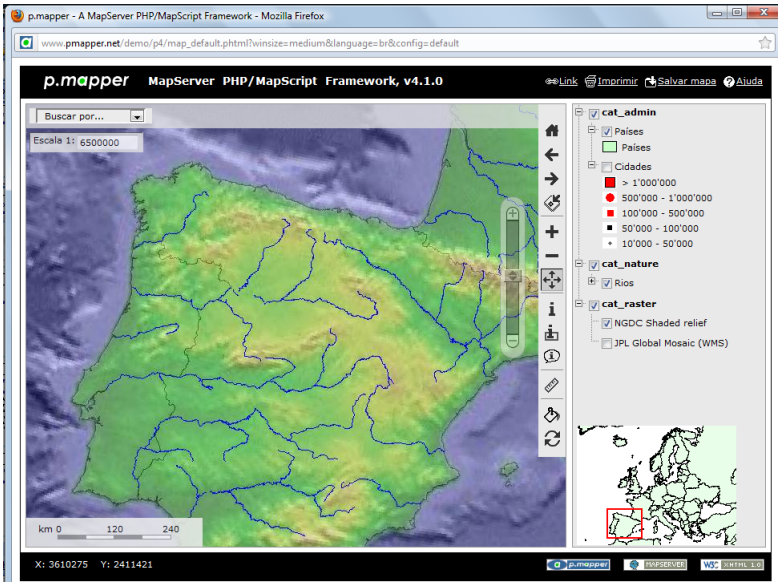

Figura 5: Visualizador p.mapper. Fonte:(P.MAPPER, 2011).

2.12 PHP

PHP é uma linguagem de *scripting* de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a *web* e pode ser mesclada dentro do código HTML (PHP, 2011). Sua sintaxe se assemelha à de outras linguagens de programação, tais como C e Java. Esta linguagem é interpretada no lado servidor da aplicação (*server-side*), e, com seus recursos, é possível criar páginas *web* através de técnicas e recursos avançados de programação, obtendo resultados rapidamente.

2.13 Javascript e jQuery

Javascript é uma linguagem de *scripting* utilizada para incrementar aplicações *web*. Assim como em PHP, seu código pode ser embutido em páginas HTML. A principal diferença está no modo de funcionamento: enquanto PHP é interpretada no lado servidor, Javascript é executada no lado cliente (*client-side*), sendo então interpretada por navegadores de internet. Cabe observar que Javascript e Java são duas linguagens de programação distintas.

Com seu uso, é possível a obtenção de características mais dinâmicas às páginas, tais como: ações por movimentos do mouse, navegação entre janelas personalizadas, efeitos visuais, dentre outras opções. A biblioteca jQuery é construída sobre a linguagem Javascript, e adiciona ainda mais recursos de dinamismo no lado cliente de uma aplicação *web*.

2.14 PHPlot

PHPlot é uma biblioteca gráfica que permite a criação de gráficos dinâmicos para a linguagem PHP. Ela oferece a criação dos principais tipos de gráficos, tais como gráficos de linha, barras, pontos e pizza. Seu uso, resumidamente, consiste na inclusão dos pacotes necessários, configuração de parâmetros para a geração do gráfico, tais como tipo de gráfico, cores, tamanho, dentre outros, e definição dos dados de base para a criação do mesmo.

Capítulo 3

Concepção do Projeto

Antes da concepção do projeto, foi evidenciado junto à equipe do projeto ambiental que a construção de um sistema de controle de arborização faz parte do propósito deste Trabalho de Conclusão de Curso, sendo um subproduto do mesmo, e que o foco principal do trabalho é a realização de pesquisa e integração acerca de tecnologias livres existentes de modo a alcançar os objetivos propostos neste Capítulo.

3.1 Obtenção e Análise de Requisitos

A definição de requisitos para este sistema foi definida em conjunto com os responsáveis pelo projeto ambiental. Foram realizadas duas reuniões. Na primeira reunião, foi realizada uma entrevista com a Engenheira Agrônoma responsável pelo projeto, onde foram respondidas as seguintes questões:

1. Qual o objetivo do projeto?
   O objetivo é saber a situação real da arborização na cidade, quais espécies estão presentes, quais árvores e espécies serão substituídas e/ou introduzidas (o objetivo é ter o mínimo 10 espécies no município) e quais locais carecem de mais arborização. Uma melhor arborização pode influenciar no clima (já que a cidade é muito quente) e é também uma questão de saúde pública. A ideia é futuramente centralizar também as solicitações de corte e poda, no entanto, ainda não há estrutura para tal finalidade.

2. Como vem sendo feita a catalogação das árvores?
   Um mapa da cidade foi setorizado em quadras identificadas por números. Uma equipe de campo percorre estas quadras, obtendo as informações sobre as árvores e

tirando uma foto de cada uma.

3. Quais as informações das árvores estão sendo coletadas?

   - Coordenadas - obtida com GPS (coordenadas em quilômetros, projeção: UTM);
   - Nome popular da espécie da árvore;
   - Nome científico da espécie da árvore;
   - DAP - diâmetro na altura do peito;
   - Fuste - altura do tronco da árvore até as primeiras ramificações;
   - Projeção de copa - área de sombra da árvore, tendo como parâmetro de medição o horário do meio dia;
   - Sanidade - estado de saúda da árvore. Existem equipamentos capazes de "escanear" a árvore para verificar a integridade do seu interior. No entanto, a equipe ainda não dispõe desta tecnologia, e a sanidade foi definida visualmente;
   - Foto - uma foto para cada árvore, tirada em alta resolução (cinco *megapixels*).

4. Quais árvores são objeto desta catalogação? Calçadas, canteiros centrais, praças públicas, quintais?
   Somente as árvores das calçadas das quadras e das praças e áreas públicas livres. Para as árvores dos canteiros centrais, houve um início de trabalho semelhante, mas não houve conclusão. A ideia é que futuramente, estas árvores também sejam objeto de catálogo.

5. Possíveis novas ruas e/ou áreas/quadras serão adicionadas no mapa da cidade?
   Por enquanto não, pois o projeto servirá de base para elaboração de um novo plano de plantio na cidade, e nesta etapa de trabalho, o mapa existente é suficiente para tal fim.

6. Quais as características do mapa e do GPS utilizados?
   Foi utilizado um mapa construído em *software* de desenho assistido por computador (*computer aided design* - CAD), em escala 1:5000 e formato "DWG". O GPS utilizado possui baixa precisão.

7. O projeto tem ciência da falta de precisão do mapa e do GPS utilizados?
   Sim, temos a ciência de que os recursos não são adequados, no entanto, a expectativa é de que estes dados, ainda que em baixas condições de precisão, sejam suficientes para o alcance dos objetivos de forma geral.

8. O sistema será acessível à população?
   Inicialmente, a ideia é que o sistema seja utilizado pela equipe do projeto. Futuramente, a ideia é que a população possa visualizar estas árvores no mapa da cidade através da Internet, e também realizar os pedidos de poda e corte através do sistema. No entanto, ainda não temos estrutura suficiente para tal (espaço, equipamentos e funcionários).

9. Serão realizadas novas "vistorias" nas árvores? Existirá uma equipe para tal função?
   Atualmente está sendo feita uma única vistoria em cada árvore, através do percurso inicial da equipe de campo pelas quadras da cidade. Com o apoio do sistema, o objetivo é que novas vistorias em árvores já catalogadas sejam realizadas, principalmente em árvores mais antigas e que estão em sanidade ruim, inclusive, tirando novas fotos das mesmas.

10. E quando uma árvore for arrancada?
    Quando uma árvore é arrancada, a responsabilidade do morador ou responsável é de se plantar pelo menos uma nova espécie. Na impossibilidade de se plantar no mesmo lugar, o plantio pode ser feito em outro local indicado pelo mesmo.

11. Existe a ideia de armazenar outras informações além das árvores? Ex: focos de dengue, acidentes, etc.?
    Por enquanto, somente a arborização. Existe a expectativa de trabalhos futuros para mapeamento das nascentes do município.

12. Existe um esquema de visualização desejado? A ideia é que as árvores sejam representadas por pontos no mapa, e as espécies sejam identificadas por cores. Desta forma, ao simplesmente visualizar o mapa, a equipe poderá identificar alguns dos aspectos desejados, por exemplo:

    - Quadras ou áreas com poucas árvores;
    - Quadras ou áreas com pouca diversidade de espécies;
    - Ausência de certas espécies em áreas potenciais.

13. Quais tipos de relatórios e/ou gráficos são desejados?
    É desejável que seja possível obter, a princípio:

    - Distribuição de espécies;
    - Médias sobre os dados específicos das árvores;

- Somatório da projeção de copa, para estimar a área verde;
- Relação de substituições pendentes.

A partir desta entrevista, foi possível extrair um aspecto importante. Devido à imprecisão do mapa e também do GPS, a localização das árvores será distorcida, logo, serão necessários mecanismos de correção manual. Com base nesta entrevista e na imprecisão do mapa e das coordenadas, foram definidos os seguintes requisitos:

1. O sistema deve manter as informações coletadas sobre as árvores em um banco de dados;
2. O sistema deve representar as árvores como pontos sobre o mapa da cidade, identificando-as por pontos de referências (como quadras e praças), sendo que as espécies devem ser diferenciadas por cores;
3. O sistema deve permitir a correção manual de posicionamento das árvores, devido a possíveis distorções causadas pela falta de precisão do GPS e do mapa;
4. O sistema deve armazenar as informações obtidas nas vistorias das árvores, mantendo os parâmetros específicos (DAP, Fuste, Projeção de Copa e Sanidade) e a foto da mesma em um banco de dados;
5. O sistema deve manter as informações de uma árvore arrancada, bem como a definição da(s) árvore(s) substituta(s);
6. O sistema deve identificar quais as árvores pertencem a um certo catálogo, com a finalidade de quantificar o plantio de árvores realizado em certo período de tempo ou categoria de origem, sendo possível obter os seguintes parâmetros:
   - Distribuição de espécies;
   - Médias sobre os parâmetros específicos das árvores;
   - Somatório da projeção de copa, para estimar a área verde;
   - Relação de substituições de árvores pendentes.

Estes requisitos foram apresentados e discutidos numa segunda reunião, na qual foram aprovados.

3.2 Modelagem

Após a definição dos requisitos, o projeto foi definido e modelado e um diagrama de classes - mostrado na Figura 6 - foi elaborado.

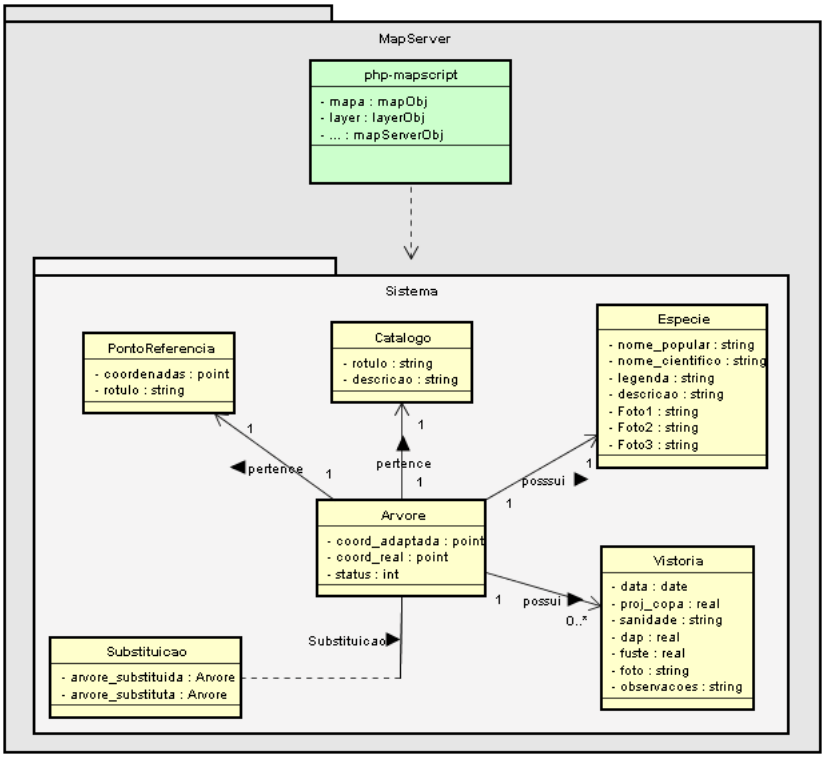

Figura 6: Diagrama de classes do Sistema

O MapServer e suas funcionalidades foram resumidos na classe "php-mapscript". Isto se deve ao fato de que a ferramenta pode ser embutida em uma linguagem de programação (no caso, PHP), desta forma, pode-se explorar os seus recursos e funcionalidades, em conjunto com os recursos da linguagem de programação escolhida.

Ao centro, a classe "Arvore", tendo como atributos sua localização real (obtida pelo GPS), sua coordenada ajustada (em caso de distorção) e o *status*, para identificar se trata-se de árvore viva ou arrancada/substituída, conforme relação com a classe "Substituição". A classe "PontoReferência" representa as quadras e praças as quais as árvores estão vinculadas. Da mesma forma, há vínculos com a espécie e catálogo de origem da árvore - classes "Espécie" e "Catalogo" - todas relações de cardinalidade 1 para 1. Por fim, a classe "Vistoria" foi criada para representar os dados das árvores coletados pela equipe de campo. Cabe observar que várias vistorias podem ser realizadas (cardinalidade 1 para muitos).

Capítulo 4

Integração das Ferramentas

Neste Capítulo, será mostrada a integração das ferramentas utilizadas para a solução do problema proposto. Algumas peculiaridades podem ser observadas, principalmente na seção referente à integração do banco de dados com o MapServer.

4.1 Estrutura da Aplicação

Após a definição do projeto, passamos então ao principal propósito deste trabalho, a integração das ferramentas utilizadas. Simplificando a arquitetura de uma aplicação MapServer conforme descrito na Seção 2.9 (p. 22) para o contexto deste trabalho, obtemos o esquema da Figura 7.

Na camada de entrada, temos o mapa da cidade em formato "shapefile", além dos novos dados adicionados para representação das árvores, através do banco de dados PostgreSQL e sua extensão espacial PostGIS. Em seguida, na camada central, temos a configuração e ambiente de execução: configuração do mapfile que define a saída gráfica do mapa, e a aplicação construída com o MapServer modo mapscript rodando sobre o servidor *web* Apache. Por fim, na saída, temos a interface DHTML (mapa e editor de dados espaciais e descritivos).

4.2 Banco de Dados e MapServer

A interoperabilidade entre o banco de dados PostgreSQL/PostGIS e o MapServer depende de alguns procedimentos específicos, conforme destacados em PostGIS (2011). Estes procedimentos serão detalhados nas subseções a seguir.

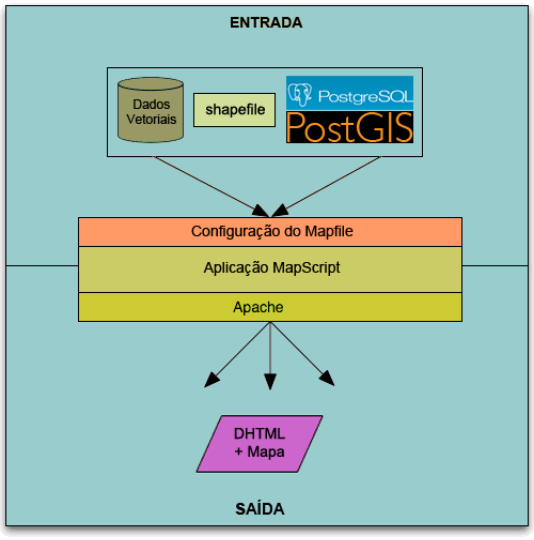

Figura 7: Arquitetura da aplicação MapServer para este contexto. Adaptada de (MAPSERVER, 2010).

4.2.1 Criando um banco de dados espacial com PostgreSQL/-PostGIS

Com o PostgreSQL e sua extensão espacial PostGIS, é possível armazenar e manipular tanto os componentes descritivos e espaciais da informação. É necessária a criação de uma instância de banco de dados PostgreSQL e habilitação de sua extensão espacial PostGIS, obtendo um banco de dados *"spatially enabled"*. Dentre as várias opções a seguir, pode-se, primeiramente, criar um banco de dados convencional. Em seguida, basta associá-lo ao PostGIS e às suas funções pré-definidas através do uso do "template_postgis" . Este template funciona como um modelo que pode ser associado a uma instância de banco de dados no ato da criação do mesma, com um simples comando visto na Listagem 2.

Listagem 2: Instrução SQL para criação de um banco de dados espacial

```
1 create database exemplo_db template = template_postgis
```

Este comando cria o banco de dados "exemplo_db" e já o habilita a trabalhar com dados espaciais (*spatially enabled*), associando-o ao "template_postgis". Ao habilitar o PostGIS em uma instância de banco de dados, duas tabelas são criadas automaticamente: "spatial_ref_sys" e "geometry_columns".

A tabela "spatial_ref_sys" armazena identificadores e descrições adicionais dos Sistemas de Referência Espaciais que são utilizados nos bancos de dados espaciais (utiliza-se a sigla SRS, do inglês *Spatial Reference System*). São quase quatro mil sistemas de re-

ferência espaciais definidos pela instituição “European Petroleon Survey Group” (EPSG), que os mantém e os publica.

Já a tabela “geometry_columns” mantém registros de tabelas com colunas espaciais. Estes registros são utilizados por outras aplicações que acessam o banco de dados (como o MapServer).

Estas tabelas são denominadas “tabelas de metadados” e são criadas de acordo com as especificações definidas pelo OGC. São essenciais para a consistência dos dados e interoperabilidade.

4.2.2 Garantindo a Interoperabilidade com o MapServer

Para criação de uma tabela espacial destinada a receber consultas de aplicações MapServer, é necessário que se utilize a função do PostGIS “AddGeometryColumn” para especificar colunas espaciais. A justificativa é que esta função adiciona automaticamente registros com parâmetros específicos na tabela de metadados “geometry_columns”, que é utilizada por outras ferramentas que utilizam o PostGIS, dentre elas, o MapServer, e também por funções do próprio PostGIS. Caso o usuário não utilize esta função, a comunicação entre o MapServer e o banco de dados PostgreSql/PostGIS pode ser prejudicada, resultando em erros na aplicação final. É importante também que se utilize a primeira coluna da tabela para armazenar o identificador de cada registro, pois algumas funções de consulta espacial do MapServer retornam somente a primeira coluna como resultado.

Na Listagem 3, um exemplo de criação de tabela.

Listagem 3: Instrução SQL para criação de uma tabela com dado espacial

```
1 create table vias (
2         ID serial not null PRIMARY KEY,
3         nome varchar(25),
4         tipo varchar(25)
5 );
6 select AddGeometryColumn('vias', 'vias_geom', -1, 'LINESTRING', 2);
```

O comando acima cria a tabela vias com as colunas especificadas (ID, nome e tipo), em seguida, adiciona a esta tabela uma coluna espacial denominada ’vias_geom’. Os parâmetros da função “AddGeometryColumn” são:

- SCHEMA_NAME (opcional) - nome do esquema do banco (padrão: public);

- TABLE_NAME - nome da tabela em que será inserida coluna do tipo geometry;
- COLUMN_NAME - nome da coluna que receberá os dados tipo geometry;
- SRID - identificador do SRS para a tabela;
- TYPE - tipo de dado (POLYGON, POINT, LINESTRING, etc.);
- DIMENSION - dimensão da coluna (2, 3 ou 4 dimensões são suportadas).

4.2.3 Inserção de dados de natureza espacial

Para inserção de dados em grande quantidade num banco de dados espacial no PostgreSQL/PostGIS, pode-se utilizar o utilitário para carga de shapefiles "shp2pgsql". O utilitário converte os *shapefiles* em comandos SQL adequados para a inserção dos dados no banco, excluindo-se a necessidade de inserção manual de cada registro. Ele possui várias configurações de execução, pelas quais é possível optar pela criação de novas tabelas para os dados ou inserção dos dados em tabelas existentes. O programa é executado via linha de comando, e as opções são escolhidas através de "flags", que são conjuntos de caracteres que identificam cada opção. Contudo, neste trabalho, devido à imprecisão de coordenadas já citada anteriormente, a inserção dos dados através do utilitário de carga não foi pertinente. Desta forma, houve a necessidade de se proceder à inserção manual de cada registro.

Para a inserção de dados em tabela espacial, deve-se utilizar comandos SQL INSERT (assim como nas tabelas de atributos não espaciais), e, o único diferencial está na utilização da função do PostGIS "St_GeomFromText". Esta função é utilizada para se criar objetos de tipos espaciais aceitos pelo PostGIS à partir da representação OGC WKT (WKT vem do inglês "*well-known-text*" - descrição textual). Um exemplo é mostrado na Listagem 4.

Listagem 4: Instrução SQL para criação de uma tabela com dado espacial

```
1 insert into ponto_onibus (ID, bairro, the_geom) values (10, 'Centro',
    St_GeomFromText('POINT(7543533 467721)',-1) );
```

A instrução acima adiciona um registro na tabela "ponto_onibus" com os valores ID='15', bairro='Centro' e um objeto ponto aceito pelo PostGIS com coordenadas (7543533, 467721).

4.2.4 Consulta a dados de natureza espacial

Se, para adicionar um dado de natureza espacial no banco de dados à partir de uma representação textual é necessário convertê-lo em um objeto espacial, para consultá-lo, deve-se fazer o inverso, ou seja, converter o objeto espacial em representação textual. Para esta ação utiliza-se a função "St_asText". Na Listagem 5, um exemplo é mostrado.

Listagem 5: Instrução SQL para criação de uma tabela com dado espacial

```
1 select ID, bairro, St_asText(the_geom) from ponto_onibus
```

A definição dos parâmetros destas funções pode ser consultada na documentação do PostGIS.

4.3 Interface de Navegação e Edição - PHP/mapscript

O p.mapper, conforme descrito na Seção 2.11 (p. 26), é um poderoso visualizador que permite, inclusive, a realização de consultas aos dados descritivos das informações espaciais. No entanto, para manipulação destes dados (inserção, exclusão ou atualização), não há suporte nativo nesta ferramenta (de fato, sua natureza sugere apenas a visualização). Logo, no contexto de nosso trabalho, considerando-se também a necessidade de se realizar operações de ajustes de coordenadas, optou-se então pelo desenvolvimento de uma nova interface que abrangesse navegação e edição. Desta forma, foi necessário integrar funcionalidades de navegação pelo mapa, tais como *zoom* e *pan*, e também de manipulação de dados descritivos e espaciais.

4.3.1 Mapa dinâmico

Conforme citado na Seção 2.7 (p. 21), o *web mapping* dinâmico está relacionado à interatividade através de recursos DHTML. E conforme pode-se observar na Seção 2.5 (p. 19), a estrutura de uma aplicação baseia-se no formato de requisições e respostas.

Desta forma, para propiciar um mapa dinâmico, seguindo estes preceitos, a solução encontrada para envio de requisições e respostas foi baseada em formulários HTML. No formulário, o mapa aparece como componente HTML de entrada de imagem ("image"). Campos para os dados descritivos também foram inseridos. A solução desenvolvida foi baseada em requisições através de ações do mouse sobre o mapa ou na submissão do

formulário por algum botão definido. Quando uma destas ações ocorre, os parâmetros são enviados ao MapServer (requisição) para que seja retornado um novo mapa (resposta), obtendo o aspecto de dinamismo esperado.

A Figura 8 exibe a interface construída: à esquerda, a barra de ferramentas cujas opções serão definidas neste Capítulo. Ao centro, o mapa navegável com suporte à adição de dados espaciais (pontos de referência e árvores). À direita, a interface de seleção de layers e o mapa em miniatura para referência de visualização. Abaixo, a interface de exibição e edição de atributos descritivos.

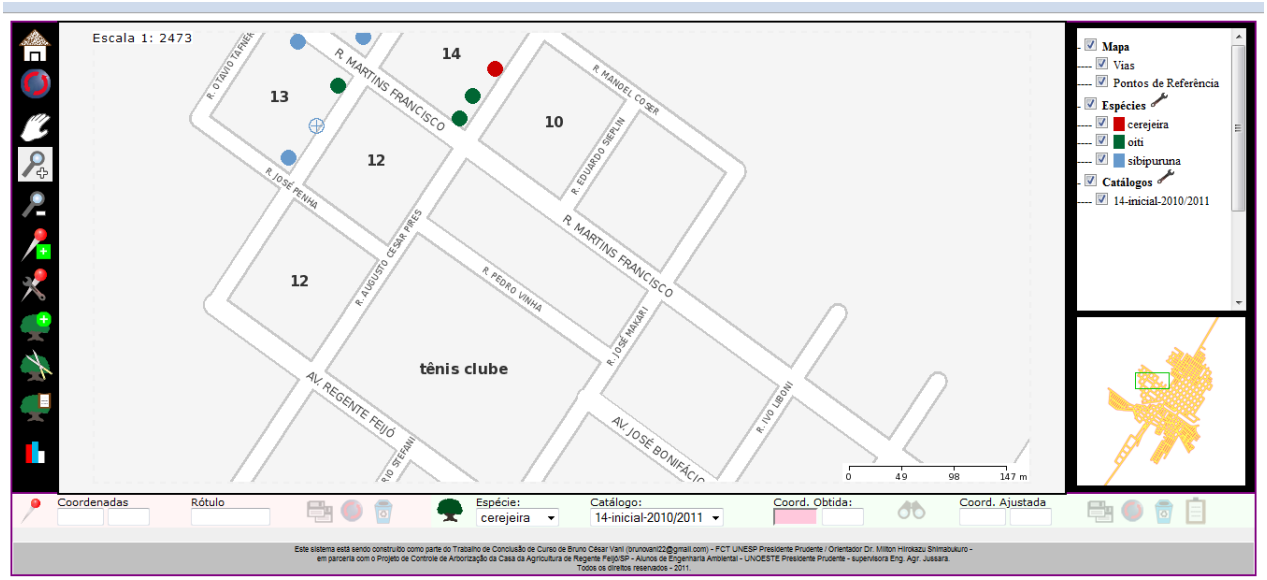

Figura 8: Interface de navegação e edição implementada

4.3.2 Propriedade "EXTENT" de um mapa

Em um mapfile, deve-se definir a propriedade "EXTENT" (extensão), conforme pode ser visto na Listagem 1 (p. 24). Esta propriedade indica ao MapServer qual a extensão geográfica do mapa que será projetado, tomando como base a unidade de medida em que o mapa se encontra (por exemplo, kilômetros ou graus decimais) e também o sistema de referência espacial utilizado. A extensão é definida por um retângulo que envolve toda a área do mapa, e é delimitado pelos pontos [min x, min y] e [max x, max y]. Pode-se verificar na Figura 9 como seria a delimitação para o polígono em azul:

Concluindo, o valor para o parâmetro "EXTENT" que seja suficiente para exibir todo o mapa é definido pelo menor retângulo envolvente de um polígono. Se modificarmos as

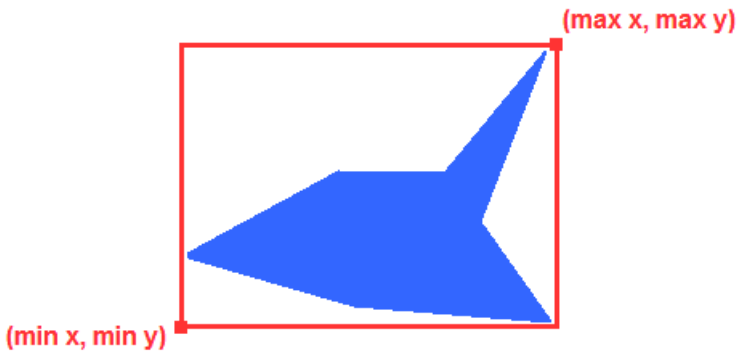

Figura 9: Demonstração da propriedade extent

medidas de “EXTENT”, a imagem pode ser cortada ou aproximada (retângulo menor que a imagem) ou receber um aspecto de distanciamento (retângulo maior que a imagem). Cabe notar que, em um mapa com muitas camadas, esta propriedade deve ser definida de acordo com a extensão da maior delas, para não causar prejuízo de informação.

Pode-se obter a extensão de um mapa armazenado no banco de dados PostGIS aplicando a função “ST_Extent”, que retornará o menor retângulo delimitador de um polígono. Também é possível obter a propriedade a partir de *shapefiles* com o utilitário “ogrinfo”, sem a necessidade de carregá-los no banco de dados para obtenção de tal parâmetro. Este utilitário é provido pela biblioteca GDAL, e mais detalhes podem ser consultados em seu site http://www.gdal.org/ - Acesso em 12/11/2011. Na Listagem 6, o resultado da execução do utilitário “ogrinfo” para obtenção da extensão do mapa da cidade utilizado.

Listagem 6: Execução do utilitário para obtenção da extensão de um mapa

```
1  C:\ms4w\tools\gdal-ogr>ogrinfo -so c:/ms4w/apps/regente/mapfiles/shape/mapa
       .shp mapa
2  INFO: Open of `c:/ms4w/apps/regente/mapfiles/shape/mapa.shp'
3        using driver `ESRI Shapefile' successful.
4
5  Layer name: mapa
6  Geometry: Line String
7  Feature Count: 262
8  Extent: (466852.755516, 7540631.913132) - (469686.649299, 7544841.671822)
9  Layer SRS WKT:
10 (unknown)
11 NOME: String (100.0)
12 BAIRRO: String (50.0)
13 relevancia: Integer (4.0)
```

```
14 id: Integer (4.0)
```

Concluindo, o parâmetro "EXTENT" é essencial para operações de navegação em um mapa. Estas operações foram criadas através de ações do mouse sobre o mesmo. Ao clicar numa posição do mapa, as coordenadas da posição (em pixels) são enviadas através do formulário HTML. Muitas das funcionalidades do MapServer requerem que estas coordenadas (em pixels) sejam convertidas na mesma unidade de medida do mapa. Esta conversão é feita com base numa regra de três simples, tomando-se como base o tamanho da imagem (em pixels) e o valor "EXTENT" do mapa atual. A função "click2map" - mostrada na Listagem 7 - adaptada à partir da sugestão disponível na documentação do MapServer, efetua esta conversão.

Listagem 7: Função para conversão de coordenadas de imagem em coordenadas geográficas

```
1  function click2map ($map, $click_fig) {
2      $ex = ms_newRectObj();
3      $ex = $map->extent;
4
5          $x_pct = ($click_fig->x / $map->width);
6          $y_pct = 1 - ($click_fig->y / $map->height);
7
8          $x_map = $ex->minx + ( ($ex->maxx - $ex->minx) * $x_pct);
9          $y_map = $ex->miny + ( ($ex->maxy - $ex->miny) * $y_pct);
10
11     $point_geo = ms_newPointObj();
12     $point_geo->setXY( (int)$x_map, (int)$y_map);
13
14         return $point_geo;
15 }
```

Na ocorrência de aplicações de *zoom* ou *pan* sobre o mapa, um novo valor "EXTENT" é calculado com base no mesmo parâmetro do mapa em tela. Desta forma, para cada atualização do formulário, o referido parâmetro deve ser armazendo para garantir a integridade destes cálculos. Para tanto, foi utilizado um campo de formulário oculto ("hidden") para armazenar tal valor, e, na ocorrência de operações de deslocamento sobre o mapa, uma função foi utilizada para obter o parâmetro do formulário e atualizar no novo mapa. Esta função é definida na Listagem 8.

Listagem 8: Função atualizar extent

```
1  function atualizarExtent ($mapa, $novo_extent){
```

```
2
3       #separa valor num vetor
4       $aux = explode(" ",$novo_extent);
5
6       #atualiza o extent do mapa atual
7       $mapa->setextent($aux[0],$aux[1],$aux[2],$aux[3]);
8
9       return $mapa;
10
11  }
```

4.3.3 Zoom

A implementação das funcionalidades *zoom in* e *zoom out* foi baseada na função do MapServer "zoomPoint". Esta função realiza uma operação que calcula uma nova extensão para o mapa, resultando em um aspecto de aproximação (*zoom in*), distanciamento (*zoom out*), ou ainda, a recentralização. São parâmetros desta função:

- Fator de *zoom*: valor 1 resulta em centralização (utilizado em operações de funcionalidade pan), valores positivos e maiores que um resultarão em operação *zoom in*, e negativos resultarão em *zoom out*;
- Posição do clique: coordenadas da imagem (em pixels) que receberam o clique;
- Largura e Altura: medidas da imagem do mapa em pixels;
- Extensão: parâmetro "extent" do mapa atual.

É conveniente em aplicações com mapas fixar limites para aproximação ou distanciamento, para evitar visualizações exageradas e processamento desnecessário. Para efetuar esta limitação, foram fixados valores mínimos e máximos de escala, desta forma, ao aplicar uma operação de *zoom*, deve-se verificar se os limites fixados foram ultrapassados. Caso isso ocorra, a operação não deve ser realizada. A escala atual de um mapa pode ser obtida pelo parâmetro "SCALEDENOM".

Foram definidos três tipos de *zoom*, os quais são definidos a seguir.

4.3.3.1 Zoom pontual

O *zoom* pontual foi implementado na forma original da função "zoomPoint". Ao clicar sobre o mapa, é aplicado *zoom* sobre aquela posição. A Listagem 9 mostra o trecho de código referente à aplicação do *zoom* pontual.

Listagem 9: Bloco de execução de *zoom* pontual

```
1  case 'zoom_in':{                    //aplica zoom fator 2
2                  $map->zoompoint(2, $click_figura, $map->width, $map->height
                       , $map->extent); //aplica zoom
3                  if($map->scaledenom < MAX_SCALE){   //caso ultrapasse a
                       escala limite
4                      $map=atualizarExtent($map, $_REQUEST['extent']);  //
                           desfaz operacao, retornando ao ultimo valor extent
5                      $warnings[]="Limite de zoom máximo atingido.";
6                  }
7                  include_once 'redesenha_selecao';    // mantem eventuais
                       componentes selecionados
8          }break;
9
10 case 'zoom_out':{                   //aplica zoom fator -2
11                 $map->zoompoint(-2,$click_figura, $map->width, $map->height
                       , $map->extent);
12                 if($map->scaledenom > MIN_SCALE){
13                         $map=atualizarExtent($map, $_REQUEST['extent']);
14                         $warnings[]="Limite de zoom mínimo atingido.";
15                 }
16                 include_once 'reposiciona_mapa.php';
17         }break;
```

Após a execução de operações de *zoom*, é desejável que eventuais seleções (de árvores ou pontos) feitas pelo usuário sejam mantidas. Esta propriedade foi implementada de maneira simples, obedecendo-se a seguinte lógica: caso houvesse uma seleção ativa na interface anterior, desenha-se a mesma seleção no novo mapa.

4.3.3.2 Zoom por rolagem de mouse

Em aplicações com mapas populares, tais como Google Maps, e também no software CAD utilizado pela equipe do projeto, as operações de *zoom* também são executadas pela rolagem de mouse. Evidenciou-se, em testes com os usuários, a necessidade de implementação desta funcionalidade. A solução encontrada foi baseada na seguinte lógica:

- Obter a direção da rolagem do mouse para determinar se a operação será de *zoom in* ou *zoom out*;
- Obter as coordenadas da imagem no ato da rolagem do mouse, para executar a operação de *zoom* naquela posição.

Para implementação desta funcionalidade, foi necessário utilizar recursos Javascript - definidos na Seção 2.13. Foi utilizada a biblioteca jQuery "Mousewheel", que atribui um valor delta à rolagem do mouse, onde a direção da rolagem é diferenciada por delta positivo ou delta negativo. Em seguida, basta se seguir o procedimento do *zoom* pontual de acordo com o resultado do valor delta.

4.3.3.3 Zoom por retângulo

Foi verificada também a necessidade de se implementar uma operação de *zoom* por seleção de área, através do recurso de clicar, arrastar e soltar do mouse, delimitando um retângulo. De fato, esta operação agiliza a navegação pelo mapa.

O MapServer possui uma função para realização de zoom por retângulo - "zoomRectangle". A função é similar à "zoomPoint", e a única diferença deve se ao fato de que, ao invés de se passar como parâmetro a posição clicada, deve-se passar como parâmetro a delimitação do retângulo. Este retângulo deve ser definido pelas coordenadas limites [min x, min y] e [max x, max y], em *pixels*.

Logo, para implementação desta funcionalidade, também foi necessário utilizar Javascript para habilitar os recursos de arrastar e soltar do mouse, que não são nativos aos navegadores *web* atuais. Foi utilizada a bibliteca jQuery "imgAreaSelect", que, dentre outras funcionalidades, permite a seleção de uma área em um componente de imagem definido em uma página *web*. A Listagem 10 demonstra o uso da biblioteca.

Listagem 10: Javascript imgAreaSelect

```
1 $('input#mapa').imgAreaSelect({
2         autoHide: true,
3         fadeSpeed: 300,
4
5         onSelectStart: function(input, selection){
6             inix=selection.x1;
7             iniy=selection.y1;
8         },
9         onSelectEnd: function (input, selection) {
```

```
10              if (selection.x1 == inix) finalx = selection.x2;
11              else finalx=selection.x1;
12
13              if(selection.y1 == iniy) finaly = selection.y2;
14              else finaly = selection.y1;
15
16              get("rect_zoom").value=inix+' '+ iniy+ ' '+finalx + ' ' +
                    finaly;
17              //get("warnings").innerHTML=inix+' '+ iniy+ ' '+finalx + ' ' +
                    finaly;
18              submit();
19          } //on select end
20      });
```

4.3.4 Pan

Esta operação foi a mais complexa de ser implementada, já que não há função "nativa" ao MapServer para realização desta funcionalidade (também chamada de ferramenta "mão"). Em solução proposta em MapServer (2010), a funcionalidade de recentralização é proposta, baseada na função "zoomPoint" com fator 1, conforme descrito na Seção 4.3.3.

Primeiramente, observemos na Figura 10 a funcionalidade aplicada a um ponto clicado P1.

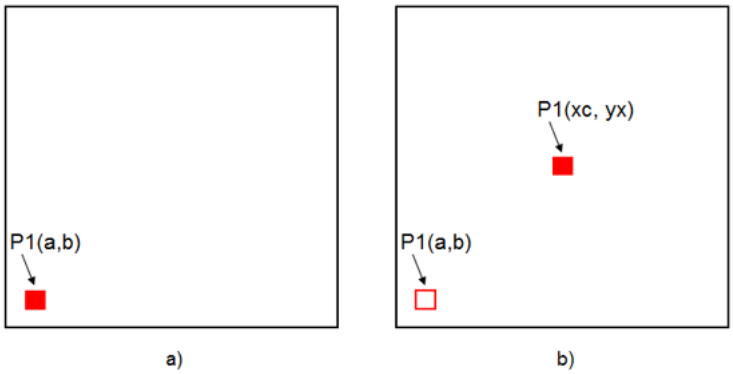

Figura 10: Comportamento da funcionalidade *pan* numa posição clicada

Pode-se observar o aspecto de recentralização: ao clicar numa posição P1 - Figura 10 a), a mesma é deslocada para o centro da imagem - Figura 10 b). Neste cenário, todo o restante de um mapa também seria deslocado.

Este aspecto não é o esperado para a funcionalidade *pan*. Em outras ferramentas

semelhantes, a funcionalidade é ativada através dos recursos de arrastar e soltar do mouse sobre a imagem (assim como na funcionalidade de zoom por retângulo), no qual espera-se que o usuário clique numa posição de origem e arraste até uma posição de destino.

Logo, a biblioteca jQuery "imgAreaSelect" também pode ser utilizada para capturar as posições de origem e de destino. Aplicando-se a função "zoomPoint" com fator 1 na coordenada de destino, a solução desejada para a funcionalidade *pan* não é satisfeita, conforme pode-se observar na Figura 11, onde P1 é origem e P2 é destino.

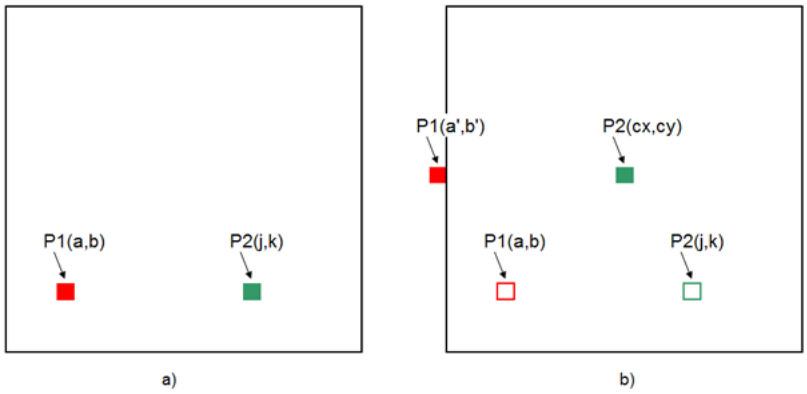

Figura 11: Comportamento da funcionalidade *pan* na coordenada de destino

Conclui-se que para o sucesso da operação, pelo menos duas operações de deslocamento através da função "zoomPoint" com fator 1 são necessários. A solução foi encontrada, e é exibida na Figura 12, onde a sequência de imagens ilustra as seguintes operações:

- Obter o ponto inicial P1 (onde o mouse foi clicado) e o ponto final P2 (onde o mouse foi soltado) - Figura 12 a);
- Aplicar a funcionalidade *pan* no ponto inicial - Figura 12 b);
- Obter o ponto P2A com coordenadas reversas de P2 - Figura 12 c);
- Aplicar a funcionalidade pan neste ponto - Figura 12 d).

Desta forma, a função pan obtém o resultado esperado, de se deslocar uma posição que foi clicada pelo mouse (origem) até a posição de onde o mouse foi soltado (destino).

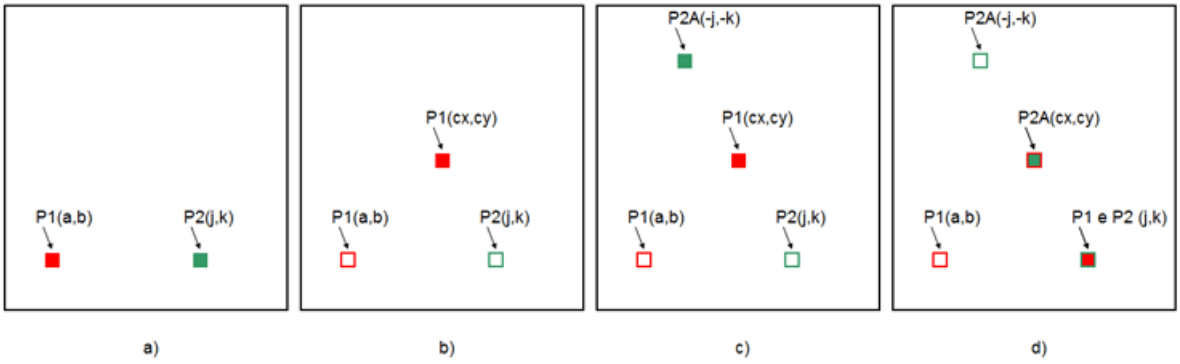

Figura 12: Operações de deslocamento para satisfazer a funcionalidade pan

4.3.5 Opção novo ponto de referência

Para inserção de um novo ponto de referência (como quadras ou praças), a seguinte lógica foi idealizada:

1. Clicar na posição desejada, suas coordenadas numéricas serão exibidas em campos de texto e um novo ponto virtual com a legenda “new” aparecerá no mapa;
2. Definir um rótulo identificador para o ponto (por exemplo, número da quadra ou nome da praça), através de um campo de texto;
3. Salvar o ponto no banco de dados através de um botão “salvar”.

Para exibir o ponto clicado como um novo elemento no mapa, deve-se instanciar este ponto como um novo objeto do MapServer tipo “point” e aplicar a função “draw”, que recebe como parâmetros:

- Mapa: o objeto mapa de destino;
- Layer: o objeto layer em que o ponto será renderizado;
- Classe: o objeto classe de estilo para representação do ponto;
- Texto: representação textual para anotar o ponto.

Na Figura 13, o detalhe da interface para criação de um novo ponto de referência.

4.3.6 Opção de Edição de Ponto de Referência

Para a edição de um ponto de referência existente, a seguinte lógica foi idealizada:

Figura 13: Opção de novo ponto de referência

1. Clicar sobre o ponto desejado para selecioná-lo, um círculo de seleção deve aparecer;
2. Clicar em uma nova posição, caso se trate de ajuste de coordenadas, e/ou editar componentes descritivos no campo de texto;
3. Atualizar os dados, através de um botão “atualizar”.

Para selecionar um ponto de referência existente para posteriormente editá-lo ou excluí-lo, foi utilizada como forma de seleção o clique do mouse sobre o mesmo. Para tanto, foi utilizada a função do MapServer “queryByPoint”.

Esta função recebe como parâmetros:

- Objeto do tipo ponto: o ponto clicado em coordenadas de mapa;
- Modo: resultado singular ou resultado múltiplo - em nosso caso, o modo singular foi utilizado, pois espera-se apenas um ponto por posição;
- Tolerância: especificação em certa unidade de medida (por exemplo, pixels ou metros) para retorno de resultados adjacentes.

Por exemplo, utilizando-se o modo de resultados múltiplos e uma tolerância de dez metros, o MapServer retornará todos os resultados sobre o layer consultado que estejam a uma distância de até dez metros do ponto clicado.

Através da consulta realizada com a função “queryByPoint” é possível obter somente a primeira coluna de uma tabela espacial, que, comumente, é utilizada para a armazenar o

identificador (ID) dos dados. Desta forma, é necessária a realização de uma nova consulta para obtenção dos demais dados do componente desejado. A Listagem 11 mostra os procedimentos para as consultas.

Listagem 11: Consultas para seleção de ponto

```
1  @$layer_consulta->queryByPoint($ponto, MS_SINGLE, 10); # o @ previne
       mensagens de resultado nao encontrado
2  $numResults = $layer_consulta->getNumResults(); # numero de resultados
3
4  # caso encontre um resultado
5  if ($numResults) {
6          $query_result = $layer_consulta->getResult(0);
7          $id = $query_result->shapeindex; # obtem a primeira coluna da
               tabela - identificador
8
9          #faz novaa consulta no banco para obter demais atributos
10         $conexao = new Conexao();
11         $conexao->conecta();
12
13         $sql = "select rotulo_ponto, st_astext(the_geom) as ponto from
               pontoreferencia where id_ponto={$id}";
14
15         ...
```

Na Figura 14, o detalhe para a interface na operação de ajuste de ponto, com definição de novas coordenadas e novo rótulo para o ponto de referência “Quadra1” selecionado:

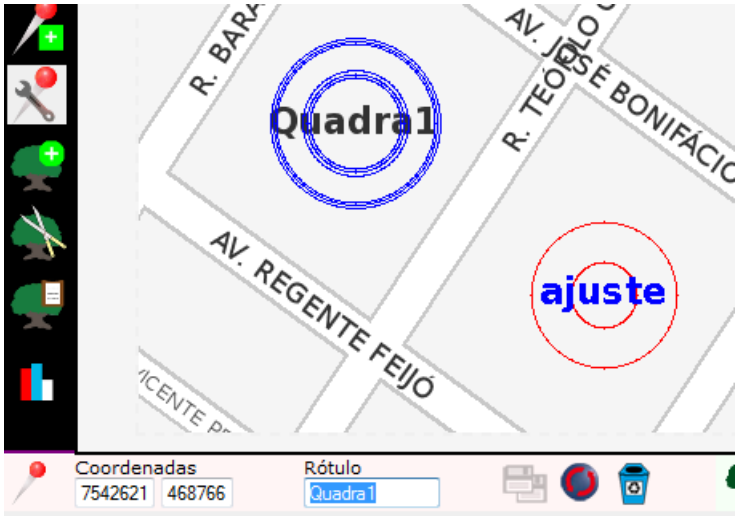

Figura 14: Opção de ajuste de ponto de referência

4.3.7 Opção Nova Árvore

Para inserção de uma nova árvore no banco de dados, a seguinte lógica foi idealizada:

1. Selecionar um ponto de referência;
2. Selecionar em uma lista de texto a espécie da árvore e a qual catálogo de origem ela pertencerá;
3. Digitar as coordenadas obtidas pelo GPS em campos de texto e pressionar um botão de "visualização prévia" para obter a posição no mapa. O ponto será desenhado na cor da espécie selecionada com um indicativo de coordenada GPS (G);
4. Caso haja distorsão de localização, pode-se clicar na posição correta para fazer um ajuste manual. Um novo ponto também será desenhado na cor da espécie selecionada e com um indicativo de coordenada ajustada (A);
5. Gravar os dados no banco através de um botão "salvar";
6. Adicionar em nova janela (aberta automaticamente) os dados da vistoria e a foto.

Conforme a modelagem utilizada para o sistema, uma árvore deve estar vinculada a um ponto de referência. Logo, para criação de uma nova árvore, deve-se primeiramente selecionar o ponto de referência desejado. Esta seleção é feita da mesma forma realizada na subseção anterior, através do clique do mouse sobre o ponto e da função "queryByPoint".

Após a seleção de um ponto de referência, deve-se também selecionar em uma lista qual a espécie da árvore a ser inserida, e também a qual catálogo de origem a árvore pertencerá.

Por fim, devido ao problema de distorsão de localização, o ajuste manual de localização pode ser feito pelo clique na coordenada desejada. Ambas as coordenadas serão mantidas no banco de dados, e a coordenada ajustada será utilizada para a visualização permanente da árvore.

Cabe ressaltar que também é possível que se introduza diretamente um ponto através da coordenada manual através do clique na posição desejada, sem a necessidade de digitação de coordenadas visualização prévia.

A Figura 15 ilustra o cenário de adição de nova árvore com ajuste de coordenadas.

Em seguida, a interface de adição de vistorias é exibida em uma nova janela. Esta interface será definida na Seção 4.4.

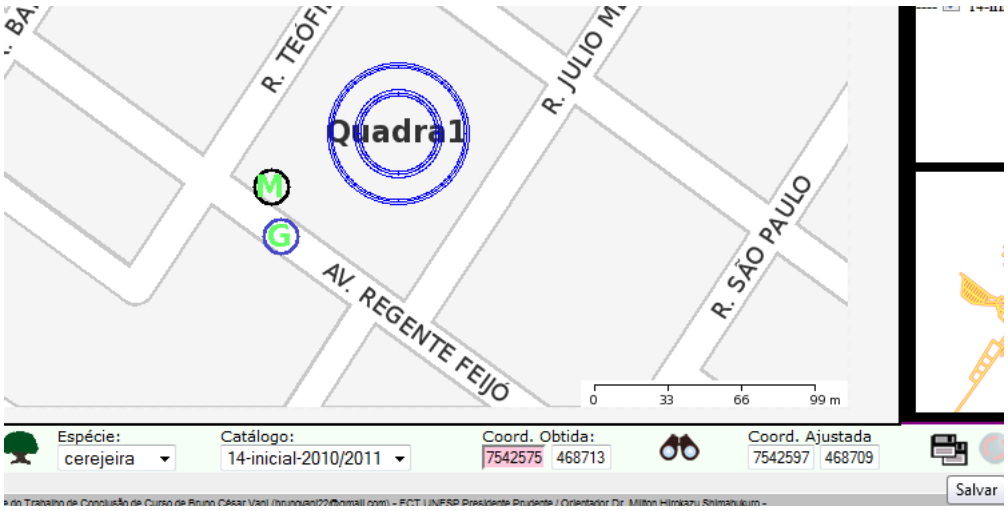

Figura 15: Opção de nova árvore com ajuste manual de coordenadas

4.3.8 Ajuste de Árvore

A estrutura desta funcionalidade é a mesma utilizada na funcionalidade de ajuste de ponto descrita na Subseção 4.3.6. Sua sequência lógica é a seguinte:

- Clicar na árvore desejada para selecioná-la. Uma indicação aparecerá no mapa indicando a seleção;
- Clicar na posição desejada para ajuste do componente espacial, ou alterar os atributos descritivos nos campos de seleção;
- Salvar as alterações através de um botão "atualizar".

4.3.9 Listar Árvore

Esta ação é a mesma que ocorre ao término da opção de salvar árvore: uma nova janela é exibida para informações sobre as vistorias das árvores. Esta funcionalidade é baseada na seguinte lógica:

1. Clicar na árvore desejada para selecioná-la;
2. Visualizar as informações de vistoria da árvore (ou adicionar mais vistorias) em uma nova janela.

Caso uma árvore já esteja selecionada, a mesma ação também pode ser executada pelo botão "listar", localizado no canto superior direito da interface de edição.

4.3.10 Opção Atualizar/Limpar

É conveniente termos, em aplicações baseadas em mapas, uma opção que retire seleção de objetos atualmente selecionados. Em nosso caso, que retire eventuais seleções de árvores ou quadras. Desta forma, um botão foi adicionado à barra de ferramentas com esta finalidade, e, ao acionado, retira as seleções existentes, mantendo-se o mapa na mesma posição.

4.3.11 Opção *Home*

Também comum em ferramentas de edição, esta opção faz com que o mapa retorne à sua extensão inicial. Foi utilizado um *link* para a página inicial, o que reduziu a complexidade da implementação.

4.3.12 Interface de seleção de Layers e Legendas

Baseando-se na interface do p.mapper, foi idealizada para o sistema uma interface de seleção de *layers* junto com a interface de legenda. O usuário pode optar por visualizar:

- Mapa: vias e/ou quadras;
- Árvores:

  de todas as espécies ou de espécies específicas;

  de todos os catálogos ou de catálogos específicos.

Como as árvores são representadas por cores sólidas, a legenda para espécies foi construída manualmente através da exibição de um pequeno quadro à frente do nome das mesmas, utilizando-se a mesma cor representada no mapa que está mantida no banco de dados.

A Figura 16 detalha a interface de seleção de layers com as legendas das espécies.

4.3.13 Mapa de Referência e Barra de Escala

Um mapa de referência é utilizado para indicar, em uma miniatura, a região do mapa que está sendo visualizada. Com o MapServer, esta opção é implementada através do objeto “REFERENCE”. Uma imagem da miniatura do mapa principal deve ser utilizada,

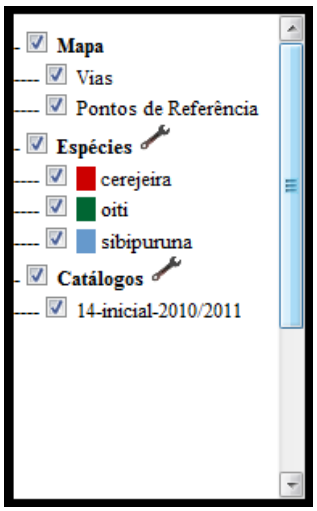

Figura 16: Interface de seleção de *layers*

e, através da extensão do mapa principal, um retângulo indicador é sobreposto à imagem de referência, indicando a região visualizada. A barra de escala, definida pelo objeto do MapServer "SCALEBAR" cria uma régua com referência em unidades de medida definidas, para tornar possível uma relação entre as medidas da imagem atual e o contexto no mundo real.

A Figura 17 exibe o detalhe do mapa de referência e da barra de escala.

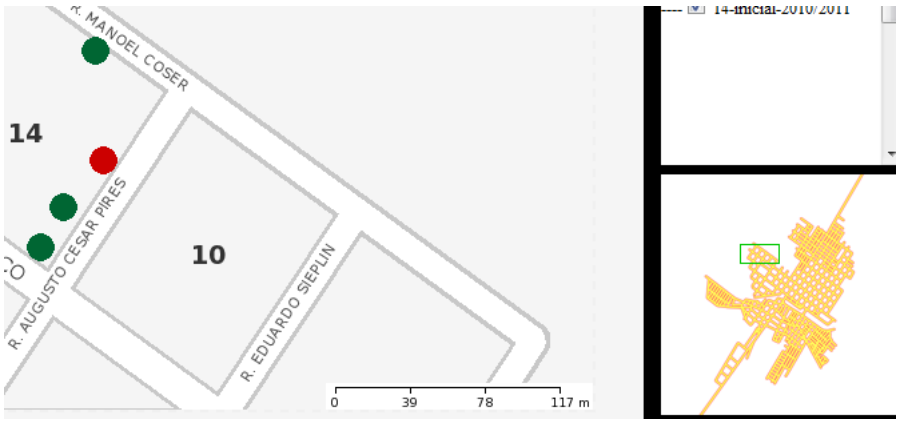

Figura 17: Mapa de Referência e Barra de Escala

4.4 Interface para Dados Descritivos

Para a interface gráfica de manipulação dos demais dados descritivos, alguns detalhes importantes são listados nas subseções abaixo.

4.4.1 Gerenciamento de Espécies e Catálogos de Origem

Através do botão "configurações", identificado por uma "chave inglesa" junto ao seletor de *layers* no lado esquerdo da interface de mapa, é possível abrir uma nova página para o gerenciamento de catálogos de origem ou de espécies. A interface padrão definida para estas páginas foi a exibição de uma listagem contendo a opção de adicionar novos itens e editar ou excluir itens já existentes, conforme a Figura 18.

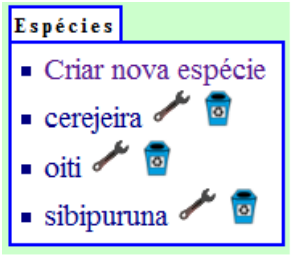

Figura 18: Gerenciador de Espécies

Cabe ressaltar que uma espécie ou catálogo de origem não pode ser excluída caso esteja sendo referenciada por alguma árvore. Esta restrição de integridade foi definida no esquema de bando de dados para evitar registros com valores nulos.

Na definição de uma nova espécie, o usuário também define a cor representativa através da seleção em uma lista de cores pré-definidas, conforme a Figura 19.

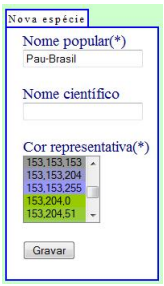

Figura 19: Interface de criação de espécies - seletor de cores

4.4.2 Exibição das Vistorias

A interface para exibição das vistorias de uma árvore foi construída baseada em tabelas. Inicialmente, uma descrição da árvore é exibida. Logo abaixo, uma caixa de opção para adicionar novas vistorias, e, por fim, a listagem das vistorias já cadastradas. A foto aparece como miniatura que é ampliada com a passagem do mouse sobre a mesma. Esta interface pode ser vista na Figura 20.

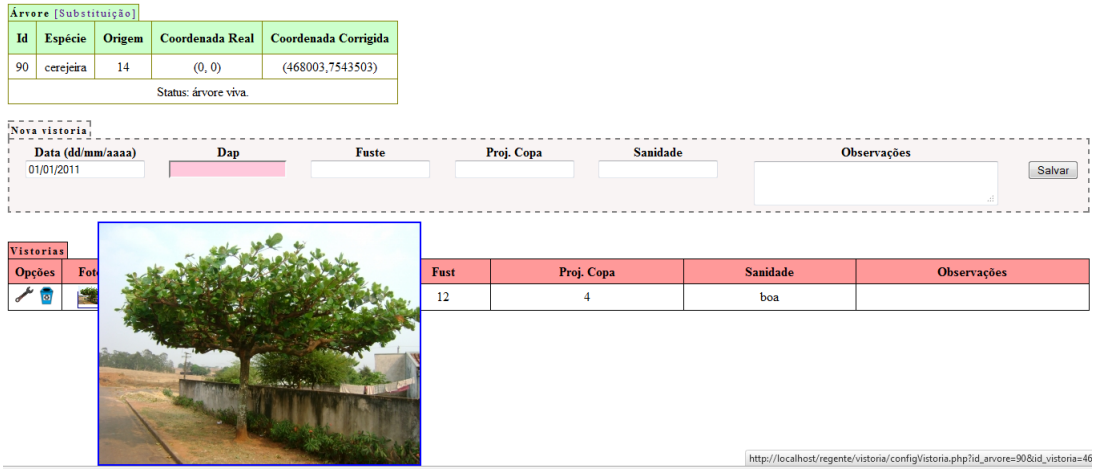

Figura 20: Interface de vistorias de uma árvore

4.4.3 *Upload* de imagens

Conforme destacado na Seção 3.1 (p. 28), as fotografias das árvores já catalogadas estão em alta resolução. Como o objetivo é publicar estas fotos em um sistema *web*, com a possibilidade, inclusive, de futuramente permitir o acesso à população através da internet, arquivos muito grandes acarretarão em lentidão para exibição, além de requererem mais espaço de aramazenamento. Neste contexto, a solução utilizada foi realizar uma redução no tamanho das imagens logo após o *upload* através da biblioteca PHP "WideImage". Com esta biblioteca é possível definir novos tamanhos para as imagens, mantendo-se as proporções do arquivo. A Listagem 12 mostra o uso da biblioteca.

Listagem 12: Uso da biblioteca WideImage

```
1  //caso a imagem seja maior que o tamanho sugerido
2          if(isset($resize)){
3                  // Chama o arquivo com a classe WideImage
4                  require('../../classes/wideimage/WideImage.php');
5
6                  // Carrega a imagem a ser manipulada
7                  $image = WideImage::load($caminho_imagem);
8
9                  // Redimensiona a imagem
10                 if($resize=='retrato')
11                         $image = $image->resize(600, 400);
12                 else
13                         $image = $image->resize(400, 600);
14
15
```

```
16                  // Salva a imagem em um arquivo (novo ou não)
17                  $image->saveToFile($caminho_imagem);
18          }
```

4.4.4 Relatórios e Gráficos

Para a geração dos relatórios e gráficos definidos na Seção 3.1 (p. 28), como não há setorização do mapa da cidade em bairros, foi introduzida uma interface de seleção de quadras. Logo, o usuário pode optar por visualizar os relatórios e gráficos tendo como base as árvores de toda a cidade, ou apenas a área delimitada por um conjunto de quadras. A Figura 21 mostra a interface de seleção de quadras, e a Figura 22 exibe os resultados.

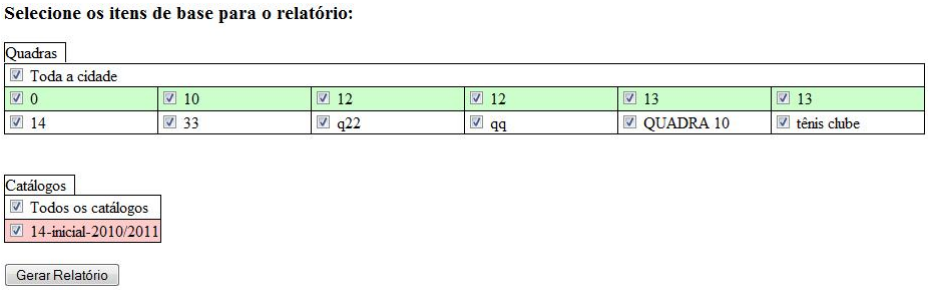

Figura 21: Interface de seleção para relatórios e gráficos

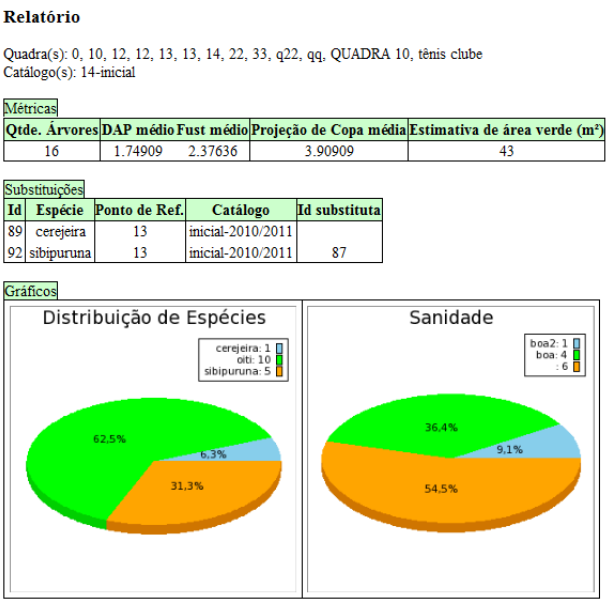

Figura 22: Interface de relatórios e gráficos

Para todos os gráficos ou relatórios, a condição de pertinência às quadras escolhidas e

aos catálogos selecionados é verificada. Esta condição é verificada através de uma cláusula SQL "IN", que verifica se os resultados satisfazem os valores esperados de um conjunto. Os parâmetros desta cláusula foram armazenados em uma variável PHP denominada "in_clause", conforme valores marcados na interface de seleção exibida na Figura 21.

Para a construção dos gráficos, foi utilizada a bibliteca PHPlot, descrita na Seção 2.14 (p. 27). Uma função para criação de gráficos de pizza foi criada, que recebe como parâmetros principais os dados em pares (valor, quantidade).

4.4.4.1 Gráfico de Distribuição de Espécies e de Sanidade

Os valores para criação do gráfico de distribuição de espécie foram obtidos atravbés da consulta exibida na Listagem 13. Como base para a consulta, apenas as árvores arrancadas foram desconsideradas (status = 0).

Listagem 13: Consulta para distribuição de espécies

```
1 $sql = "select nome_popular, count(nome_popular) from arvore
2                 where status != 0 {$in_clause} group by nome_popular ";
3
4 $rs= pg_query($sql); #executa consulta
```

Ao somarmos o total das espécies obtidas na consulta da Listagem 13, também é possível obter a quantidade de árvores existentes.

Já para o gráfico de sanidade, como este é um parâmetro pertencente às vistorias, foi necessário realizar uma consulta com junção das duas tabelas (árvore e vistoria), tomando como base a sanidade da árvore obtida na última vistoria. A seguinte consulta foi realizada:

Listagem 14: Consulta para distribuição de sanidade

```
1 $sql = "select lower(vistoria.sanidade), count(*)
2                 from arvore full join vistoria on arvore.id_arvore =
                      vistoria.id_arvore
3                 where vistoria.id_vistoria in (select max(id_vistoria)
4                 from vistoria group by id_arvore)
5                 and arvore.status != 0 $in_clause group by lower(vistoria.
                      sanidade)";
6
7 $rs= pg_query($sql); #executa consulta
```

4.4.4.2 Médias dos parâmetros específicos

Utilizando-se a mesma lógica da consulta para o gráfico de sanidade (de se obter o parâmetro referente à última vistoria), a consulta constante da Listagem 15 foi realizada para obteção de valores médios de DAP, Fuste e Projeção de Copa.

Listagem 15: Consulta para obtenção de médias

```
1 $sql = "select  avg(vistoria.dap) as dap,
2                 avg(vistoria.fust) as fust,
3                 avg(vistoria.proj_copa) as proj_copa
4                 from arvore full join vistoria on arvore.id_arvore =
                      vistoria.id_arvore
5                 where vistoria.id_vistoria in (select max(id_vistoria) from
                       vistoria group by id_arvore)
6                 and arvore.status != 0 $in_clause";
7
8 $rs= pg_query($sql); #executa consulta
```

4.4.4.3 Estimativa de Área Verde

Para a estimativa de área verde, o método utilizado foi o somatório dos valores de projeção de copa, também mantendo-se a lógica de se obter o parâmetro referente à última vistoria de uma árvore. A Listagem 16 exibe esta consulta.

Listagem 16: Consulta estimativa de área verde

```
1 $sql = "select  sum(vistoria.proj_copa)::real as total
2                 from arvore full join vistoria on arvore.id_arvore =
                      vistoria.id_arvore
3                 where vistoria.id_vistoria in (select max(id_vistoria)
4                 from vistoria group by id_arvore)
5                 and arvore.status != 0 $in_clause";
6
7 $rs= pg_query($sql); #executa consulta
```

4.4.4.4 Relação de Substituições

Esta relação visa identificar os casos de árvores arrancadas que ainda não foram substituídas pelos responsáveis, e também os casos de substituições já concluídas. Esta consulta realiza uma pesquisa das árvores arrancadas (status = 0) e que não contém

registros na tabela de substituição como substitutas. A consulta que satisfaz a proposta está definida na Listagem 17.

Listagem 17: Consulta para relação de substituições

```
$sql = "select arvore.id_arvore, arvore.nome_popular,
                pontoreferencia.rotulo_ponto,
                origem.rotulo_origem, substituicao.id_substituta
                from pontoreferencia, origem, arvore left join substituicao
                on arvore.id_arvore=substituicao.id_substituida
                where
                arvore.id_ponto=pontoreferencia.id_ponto and
                arvore.id_origem=origem.id_origem and
                arvore.status = 0  $in_clause order by substituicao.
                   id_substituta desc";

$rs= pg_query($sql); #executa consulta
```

4.5 Experimentos e Retorno dos Usuários

Para realização de testes com os usuários, ao término da implementação do sistema, o mesmo foi instalado provisoriamente em computador do Projeto Ambiental. Foram realizados testes de uso do sistema junto à equipe do Projeto Ambiental, onde os usuários utilizaram o sistema e demonstraram satisfação com o desempenho do mesmo. As seguintes características foram as mais evidenciadas:

- Interface gráfica de simples utilização e visual agradável: as operações disponíveis no sistema são de fácil aprendizado e não requerem muito tempo de treinamento. A interface gráfica do mesmo causou boa impressão;
- Facilidade para obtenção de estimativa de área verde: antes da implementação do sistema, era necessário somar os parâmetros oriundos de centenas de planilhas de dados, o que era inconveniente e necessitava de muito tempo. Com o sistema, este parâmetro é obtido rapidamente e em poucos passos.

Os usuários também mencionaram a possibilidade de expandir o sistema para incluir dados referentes à conservação de nascentes do município, conforme levantamento em andamento.

Capítulo 5

Conclusão

5.1 Considerações Finais

O objetivo de se construir um Sistema de Controle de Arborização através da Integração de Ferramentas Livres existentes foi alcalçado. São muitas as tecnologias que podem ser incorporadas em um SIG via *web*, no entanto, percebe-se, atualmente, que falta cultura de uso em aplicações deste porte, o que faz com que esta tecnologia se dissemine lentamente. A integração de ferramentas permite a construção de aplicações completas e que facilitam o processo de tomada de decisão do usuário final.

Foi possível abordar vários conceitos acerca de aplicações com dados espaciais, em especial as que utilizam o MapServer e o PostgreSQL/PostGIS. Ambas as ferramentas possuem inúmeras funcionalidades que superam os limites deste estudo, no entanto, com o que foi aqui apresentado, é possível ter uma base sólida para a construção de aplicações bem estruturadas. Foi possível adquirir também a habilidade de se introduzir esquemas visuais com dados espaciais em bancos de dados somente com atributos descritivos, criando visualizações temáticas a partir dos dados existentes.

É possível ir além, explorando ainda mais os recursos do MapServer e também de outras ferramentas que o utilizam, tais como frameworks de visualização como o "p.mapper", e também do PostGIS, realizando operações topológicas sobre mapas.

Finalizando, com a realização deste trabalho de conclusão de curso foi possível aplicar e solidificar conteúdos vistos em disciplinas do curso de graduação em Bacharelado em Ciência da Computação, além de ampliar habilidades práticas e de realizar um Programa de Formação Complementar sobre o uso de informações espaciais.

5.2 Futuros Trabalhos

Novos trabalhos com dados espaciais podem ser desenvolvidos a partir da base construída neste trabalho de conclusão de curso. Ainda no município de Regente Feijó, em outras vertentes de monitoramento de recursos ambientias, há a oportunidade de expadir e solidificar um SIG que contemple o monitoramento de nascentes, reservas municipais, dentre outros recursos.

Inúmeras outras áreas de aplicação também podem ser exploradas, tais como monitoramento de doenças infecto-contagiosas, de indicadores sociais, de acidentes de trânsito, dentre outras, fazendo com que a organização e visualização de informações em grande quantidade permita melhorias no processo de tomada de decisões.

Referências

APACHE. *Apache HTTP Server Project*. 2011. Disponível em: <http://httpd.apache-.org/ABOUT_APACHE.html>. Acesso em: 03/05/2011.

CAMARA, G.; DAVIS, C.; MONTEIRO, M. V. *Introdução à Ciência da Geoinformação*. São José dos Campos: [s.n.], 2004. Disponível em: <http://www.dpi.inpe.br/gilberto-/livros.html>. Acesso em: 03/05/2011.

CAMARA, G. et al. *Anatomia de Sistemas de Informação Geográfica*. 1996. Disponível em: <http://www.dpi.inpe.br/gilberto/livros.html>. Acesso em: 03/05/2011.

DATE, C. J. *Introdução a Sistemas de Bancos de Dados*: Tradução vanderberg dantas de souza. Rio de Janeiro: Campos, 2000.

ESRI. *ESRI Shapefile Technical Description - An ESRI White Paper - July 1998*. 1998. Disponível em: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. Acesso em: 21/04/2011.

IBGE. *Sala de Imprensa :: Portal na Internet integrará informações geoespaciais do país*. 2010. Disponível em: <http://www.ibge.com.br/home/presidencia% -/noticias/noticia_visualiza.php?id_noticia=1587>. Acesso em: 03/05/2011.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a Internet: uma abordagem top-down.*: Tradução arlete smille marques. 3. ed. [S.l.]: Pearson Addison Wesley, 2006.

MAPSERVER. *MapServer Documentation - Release 5.6.6*. 2010. Disponível em: <http://mapserver.org/MapServer-56.pdf>. Acesso em: 10/04/2011.

OSI. *The Open Source Definition*. 2011. Disponível em: <http://www.opensource.org-/docs/osd>. Acesso em: 15/11/2011.

PENG, Z.-R.; TSOU, M.-H. *Internet GIS*: distributed geographic information services for the internet and wireless networks. John Wiley: Hoboken, 2003.

PHP. *Manual do PHP*. 2011. Disponível em: <http://php.net/manual/pt_BR/index-.php>. Acesso em: 03/05/2011.

P.MAPPER. *p.mapper — A MapServer PHP/MapScript Framework*. 2011. Disponível em: <http://www.pmapper.net/demo/p42/map.phtml?language=br&config=default>. Acesso em: 01/12/2011.

POSTGIS. *PostGIS 1.5.2 Manual*. 2011. Disponível em: <http://www.postgis.org-/documentation/manual-1.5/>. Acesso em: 03/05/2011.

RIGAUX, P.; SCHOLL, M.; VOISARD, A. *Spatial Databases*: with application to gis. San Francisco: Elsevier, 2002.