



DANIEL PINHEIRO FRANCO

**ESTÁGIO EM EMPRESA PARA APLICAÇÃO
DE BANCO DE DADOS EM DADOS
ESPACIAIS NAS REPRESENTAÇÕES
VETORIAIS E MATRICIAIS**

LAVRAS – MG

2014

DANIEL PINHEIRO FRANCO

**ESTÁGIO EM EMPRESA PARA APLICAÇÃO DE BANCO DE DADOS
EM DADOS ESPACIAIS NAS REPRESENTAÇÕES VETORIAIS E
MATRICIAIS**

Relatório de Estágio de graduação apresentado
ao Departamento de Ciência da Computação da
Universidade Federal de Lavras para obtenção do
título de Bacharel em Ciência da Computação.

Orientador

Prof. Dr. Ricardo Terra

Coorientadora

Ma. Christiane Faleiro

Sidney

LAVRAS – MG

2014

DANIEL PINHEIRO FRANCO

**ESTÁGIO EM EMPRESA PARA APLICAÇÃO DE
BANCO DE DADOS EM DADOS ESPACIAIS NAS
REPRESENTAÇÕES VETORIAIS E MATRICIAIS**

Trabalho de Conclusão de Curso de
Graduação apresentado ao Colegiado do
Curso de Bacharelado em Ciência da
Computação, para obtenção do título de
Bacharel.

APROVADA em 25 de novembro de 2014.

Dr. Ramon Gomes Costa

Antonio Carlos de Sousa Couto Junior


Dr. Ricardo Terra Nunes Bueno Villela (Orientador)

Ma. Christiane Faleiro Sidney (Coorientador)

**LAVRAS-MG
Novembro/2014**

Dedico à minha família, em especial meus pais, Almir e Floramar.

AGRADECIMENTOS

Primeiramente, à Deus por todos os dons a mim concedidos.

Aos meus pais, por todo apoio em minha formação educacional.

Ao meu irmão, companheiro de apartamento em Lavras, que me ajudou nos momentos difíceis do começo do curso.

Aos meus amigos da faculdade, em especial à minha amiga Aline Reis, por sempre me fornecerem apoio e momentos de descontração.

Ao meu orientador Ricardo Terra e coorientadora Christiane Faleiro, pelas correções e sugestões dadas ao longo do desenvolvimento deste trabalho.

À todos os meus professores, por todo conhecimento transmitido.

Ao LEMAF, por proporcionar o desenvolvimento deste trabalho.

*"Não vos inquieteis, pois, pelo dia de amanhã;
porque o dia de amanhã cuidará de si mesmo.*

Basta a cada dia o seu mal."

(Mateus 6:34)

RESUMO

A representação de dados espaciais em sistemas de banco de dados tornou-se extremamente importante para a cartografia. Isso se deve à facilidade na atualização de mapas que são constantemente modificados em virtude de rearranjos espaciais ao longo do tempo. Por exemplo, informações ambientais de propriedades e posses rurais são geralmente modificadas por diversos fatores, como as práticas de desmatamento e queimada para o cultivo de plantações ou formação de pastos. Diante desse cenário, muitos projetos relacionados ao manejo e inventário florestal necessitam de banco de dados capazes de armazenar e manipular dados espaciais dinâmicos de forma eficiente. Tais dados podem ser representados na forma matricial (ou *raster*) ou vetorial. O Laboratório de Estudos e Projetos em Manejo Florestal – LEMAF, onde o estágio é realizado, conduz projetos em diversas áreas, como o desenvolvimento de Sistemas de Informação Geográfica, os quais utilizam de Banco de Dados Geográficos como componente essencial de sua arquitetura. Neste laboratório, foram realizadas as atividades de pesquisa bibliográfica em Banco de Dados Geográficos para total domínio das ferramentas utilizadas pela empresa, comparação de desempenho entre dados nas estruturas matriciais e vetoriais para cálculos de área de sobreposição realizados no Sistema Nacional de Cadastro Ambiental Rural, análise temporal de imagens de satélite utilizando Banco de Dados e administração e documentação do Banco de Dados SisBolsaVerde.

Palavras-Chave: Banco de Dados Geográficos, Sistemas de Informação Geográfica, LEMAF, PostGIS, Raster, SisBolsaVerde.

SUMÁRIO

1	Introdução	13
2	Descrição Geral: LEMAF	16
3	Descrição das Atividades	18
3.1	Pesquisa bibliográfica e principais ferramentas	18
3.2	Comparação de desempenho entre dados matriciais e vetoriais no SiCAR	19
3.3	Análise temporal de rasters em Banco de Dados	20
3.4	Participação no projeto SisBolsaVerde	21
4	Descrição dos Processos Técnicos	23
4.1	Pesquisa bibliográfica e principais ferramentas	23
4.1.1	Fundamentação Teórica	23
4.1.1.1	Representação de Dados Geográficos em Sistemas de Banco de Dados	23
4.1.1.2	Sistemas Gerenciadores de Banco de Dados Espaciais	26
4.1.2	Atividades Executadas	30
4.2	Comparação de desempenho entre dados matriciais e vetoriais no SiCAR	34
4.2.1	Fundamentação Teórica	34
4.2.2	Atividades Executadas	38
4.2.2.1	Metodologia e dados utilizados	38
4.2.2.2	Execução dos Testes	44
4.2.2.3	Resultados e Discussão	47
4.3	Análise temporal de rasters em Banco de Dados	49
4.3.1	Fundamentação Teórica	49
4.3.2	Atividades Executadas	50
4.4	Participação no projeto SisBolsaVerde	60
4.4.1	Fundamentação Teórica	60

4.4.1.1	Scrum	60
4.4.1.2	Modelagem de Banco de Dados	61
4.4.2	Atividades Executadas	62
5	Conclusão	69
5.1	Considerações Finais	69
5.2	Trabalhos Futuros	70
A	Tempos de execução das consultas para o cálculo de área de sobreposição	74
B	Padrões para Banco de Dados do LEMAF	75

LISTA DE FIGURAS

4.1	Representação matricial.	24
4.2	Representação vetorial.	25
4.3	Informações geográficas de uma região do mundo real dispostas em camadas.	26
4.4	Relacionamentos topológicos entre objetos espaciais.	26
4.5	Hierarquia dos tipos geométricos da especificação SFSQL.	29
4.6	Relacionamento entre a tabela com feição geométrica e as tabelas de metadados.	30
4.7	Dados espaciais da cidade de Nova Iorque.	33
4.8	<i>Bounding Boxes</i> cobrindo diferentes tipos de feições.	35
4.9	Árvore Quadtree.	36
4.10	Árvore R-tree.	37
4.11	Mapeamento da cobertura do solo do estado do Mato Grosso.	39
4.12	Remanescentes de vegetação nativa do estado do Mato Grosso.	40
4.13	Imagem ampliada das geometrias da tabela <i>vector_table</i>	42
4.14	Diferença entre as funções de conversão de raster para geometria.	43
4.15	Polígono representando uma única propriedade rural.	43
4.16	Polígonos representando 1.000 propriedades rurais.	44
4.17	Diferenças entre os resultados das consultas 2 (verde) e 3 (vermelho) nas bordas do polígono do <i>shapefile</i> 1.	49
4.18	Imagem RapidEye de 2011.	50
4.19	Imagem Rapideye de 2012.	51
4.20	NDVI da Imagem de 2011.	54
4.21	NDVI da Imagem de 2012.	54
4.22	Diferença dos NDVIs de 2011 e 2012.	55
4.23	Máscara de nuvens para a tabela <i>ndvi_diferenca</i>	56
4.24	<i>Shapefile</i> recortado nas regiões de nuvens.	56

4.25	<i>Shapefile</i> categorizado sobre o atributo <i>media_diferenca</i>	59
4.26	Padrões de processo <i>Scrum</i>	60
4.27	Modelagem parcial do esquema portal_seguranca do Banco de Dados SisBolsaVerde.	64
4.28	Planilha de atividades do Banco de Dados SisBolsaVerde.	65

LISTA DE TABELAS

4.1	Comparação pareada (<i>shapefile 1</i>).	47
4.2	Comparação pareada (<i>shapefile 2</i>).	47
A.1	Tempos de execução para o Shapefile #1	74
A.2	Tempos de execução para o Shapefile #2	74

LISTA DE SCRIPTS

4.1	Definição do objeto SDO_GEOMETRY no Oracle Spatial.	28
4.2	Exemplos de consultas realizadas sobre a base de dados de Nova Iorque.	34
4.3	Consultas para reclassificação binária da tabela <i>full_raster_table</i>	41
4.4	Consultas para adicionar <i>constraints</i> e o índice GiST na tabela <i>raster_table</i>	42
4.5	Consulta para vetorização da tabela raster.	42
4.6	Consultas dos testes realizados para avaliar o desempenho das estruturas matriciais e vetoriais.	45
4.7	Consultas para criação da tabela <i>ndvi_diferenca</i>	52
4.8	Funções utilizadas na álgebra de mapas para o cálculo do NDVI e do NDVI diferença.	53
4.9	Consulta para a criação da tabela <i>shape_diferenca</i>	57
4.10	Consulta utilizada para atribuir as médias dos valores de <i>pixels</i> aos polígonos.	58
4.11	Função para extrair todos os valores de <i>pixels</i> de um dado <i>raster</i>	59
4.12	Função PL/pgSQL para converter um WKT em uma geometria e calcular a área dessa geometria em hectares.	66
4.13	Função PL/pgSQL para determinar a projeção de um objeto geográfico no globo.	67
4.14	Função PL/pgSQL para preencher automaticamente a tabela <i>rel_area_municipio</i>	68

1 INTRODUÇÃO

O presente documento visa contemplar o plano de estágio desenvolvido no período de estágio supervisionado não obrigatório realizado no Laboratório de Estudos e Projetos em Manejo Florestal (LEMAF). Este documento especifica os conhecimentos utilizados em Banco de Dados Geográficos para o cumprimento das atividades propostas pela empresa na qual o estágio se realiza, tal como as metodologias empregadas e a descrição dessas atividades para o desenvolvimento de Sistemas de Informação Geográfica.

Os Sistemas de Informação Geográfica (SIGs) consistem de ferramentas computacionais para análise de fenômenos e objetos de natureza geográfica. As principais aplicações desses sistemas são o cadastro de entidades geográficas, a simulação de cenários e análises estratégicas para o auxílio na tomada de decisão (LUZ *et al.*, 2001).

Um dos componentes da arquitetura de um SIG é o Banco de Dados Geográficos, no qual as informações geográficas do mundo real são armazenadas em referência às suas características espaciais, não-espaciais e temporais de modo a realizar cálculos geométricos e análise espacial entre os dados georreferenciados.

Existem vários projetos ambientais financiados por instituições públicas ou privadas que visam o monitoramento ambiental do território brasileiro. A fácil visualização gráfica de informações geográficas obtidas em SIGs, unido ao uso das informações presentes em Banco de Dados, permitem que analistas identifiquem diversas características de regiões geográficas e possibilitam meios de monitoramento para proteção ambiental da região analisada.

O tema objeto do presente estágio é retratado nos diversos projetos conduzidos no Laboratório de Estudos e Projetos em Manejo Florestal – LEMAF, empresa na qual o estágio se realiza. Esses projetos, na maioria de caráter ambiental, utilizam-se de Banco de Dados espaciais a fim de obter uma integração entre os dados convencionais e os dados espaciais das aplicações geográficas.

Dentre os projetos citados acima, o Sistema Nacional de Cadastro Ambiental Rural (SiCAR) é conduzido no LEMAF atualmente, em parceria com o Ministério de Meio Ambiente, e constitui-se na mais importante ferramenta de implementação do novo Código Florestal – Lei 12.651/2012 – para a conservação do meio ambiente, com destaque na integração das informações ambientais de propriedades e posses rurais, como informações sobre o imóvel, os seus limites, áreas de reservas legais, áreas de proteção permanente, remanescentes de vegetação nativa, área rural consolidada, áreas de interesse social e de utilidade pública, compondo uma base de dados para controle, monitoramento, planejamento ambiental e econômico, e combate ao desmatamento.

Atualmente, a base de dados do SiCAR é constituída por dados alfanuméricos e geográficos na representação vetorial. Esses dados vetoriais são populados no Banco de dados através da importação de arquivos ESRI shapefiles (formato de arquivo vetorial), os quais são gerados a partir do geoprocessamento de imagens matriciais obtidas via satélite. Entretanto, a empresa pretende utilizar diretamente as imagens obtidas via satélite no Banco de Dados para obter um melhor desempenho e eliminar a necessidade da geração de *shapefiles*.

O SisBolsaVerde é outro projeto desenvolvido atualmente no LEMAF que visa o monitoramento do ecossistema brasileiro, porém com um outro objetivo, o de estimular famílias que vivem em propriedades rurais com situação de extrema pobreza a preservarem o meio ambiente e utilizarem os recursos da propriedade de modo sustentável. Esse estímulo se dá pelo benefício fornecido pelo programa social Bolsa Verde. O monitoramento geográfico dessas propriedades é feito de forma similar ao SiCAR, ou seja, a análise da propriedade obedece a Lei 12.651.

O objetivo geral do estágio em questão é aplicar os conhecimentos obtidos na área de Banco de Dados em Sistemas de Informação Geográfica de âmbito nacional, como o SiCAR e o SisBolsaVerde, e comparar o desempenho do uso de dados geográficos nas representações vetoriais e matriciais, a fim de otimizar o

processamento de consultas espaciais no Banco de Dados nos diversos projetos da empresa. Assim, o estágio contempla quatro objetivos específicos:

- Pesquisa bibliográfica em Banco de Dados Geográficos e estudo das principais ferramentas utilizadas pela empresa;
- Comparação entre dados do SiCAR nas representações matriciais e vetoriais para o cálculo de áreas de sobreposição de propriedades rurais com a cobertura do solo da região;
- Análise temporal da variação da vegetação de dois *rasters* utilizando Banco de Dados;
- Participação no projeto SisBolsaVerde, no qual foram contempladas atividades de modelagem, documentação e administração de Banco de Dados.

Este documento está dividido nas seguintes seções: A seção 2 contempla uma descrição geral do LEMAF. A seção 3 descreve as atividades exercidas pelo estagiário na empresa assim como o tempo gasto em cada atividade. A seção 4 descreve os processos técnicos utilizados para o cumprimento das atividades, relacionando a atividade pela teoria que a suporta. Por fim, a seção 5 apresenta as referências bibliográficas utilizadas para a escrita deste documento.

2 DESCRIÇÃO GERAL: LEMAF

Esta seção fornece uma descrição geral da empresa onde o estágio é realizado, incluindo informações sobre o histórico, atuação da empresa na área de computação, organização interna e produtos desenvolvidos.

O Laboratório de Estudos e Projetos em Manejo Florestal (LEMAF), fundado em Abril de 2004, é situado no Departamento de Ciências Florestais da Universidade Federal de Lavras (DCF/UFLA) e tem como objetivo efetuar pesquisa, ensino e extensão nas áreas de manejo e inventário florestal, sensoriamento remoto, geoprocessamento, sistemas de informação geográfica, etc.

Inicialmente, não existia um Setor de Tecnologia da Informação no LEMAF. Os primeiros profissionais desse setor somente davam suporte à infraestrutura computacional do local. Segundo Lasmar (2012), o Inventário Florestal do Estado de Minas Gerais, desenvolvido para a Secretaria de Estado de Meio Ambiente e Desenvolvimento Sustentável (SEMAD), foi o primeiro projeto com a participação de profissionais de TI. Devido ao sucesso do projeto, a SEMAD continuou sua parceria com o LEMAF, dando início ao desenvolvimento de sistemas de informação geográfica, o que permitiu a criação do setor de TI na empresa.

Atualmente, o LEMAF possui cerca de 85 funcionários, dos quais a maioria está presente no setor de Tecnologia da Informação, o qual é responsável pela maior parcela de projetos desenvolvidos no laboratório.

Os sistemas de software produzidos pela empresa contribuem para um melhor monitoramento e consequente proteção ambiental do território brasileiro, além de tornar acessível informações ambientais à população.

Apesar de a empresa não possuir um organograma (por ser uma instituição educacional), equipes das diversas áreas trabalham em conjunto para o desenvolvimento dos projetos, sob responsabilidade de professores. No ramo de TI, o diretor de Tecnologia da Informação é o profissional de maior responsabilidade que comanda as equipes dessa área, enquanto que os profissionais pertencentes às demais

— por exemplo, desenvolvimento, banco de dados, requisitos, etc. — se situam no mesmo nível hierárquico.

3 DESCRIÇÃO DAS ATIVIDADES

Esta seção objetiva descrever, em ordem cronológica, as atividades desenvolvidas no estágio, assim como informar o número de horas necessárias para a realização de cada tarefa. As principais atividades podem ser divididas em quatro principais: (i) *Pesquisa bibliográfica e principais ferramentas*, (ii) *Comparação de desempenho entre dados matriciais e vetoriais no SiCAR*, (iii) *Análise temporal de rasters em Banco de Dados*, (iv) *Participação no projeto SisBolsaVerde*.

3.1 Pesquisa bibliográfica e principais ferramentas

A primeira atividade do estágio se resume na imersão bibliográfica. Uma série de tutoriais relacionados ao Sistema Gerenciador de Banco de Dados PostgreSQL e sua extensão para dados espaciais, PostGIS, foram passados ao estagiário de modo que esse pudesse acompanhar as atividades exercidas pelos administradores de banco de dados da empresa.

Como os projetos da empresa geralmente consistem no desenvolvimento de Sistemas de Informação Geográfica, foi necessário entender como os dados geográficos são armazenados em um banco de dados PostGIS, além de aprender a manusear outras ferramentas SIGs utilizadas em projetos da empresa.

Essa atividade inicial foi cumprida nas primeiras semanas do mês de Agosto, sendo necessárias aproximadamente 54 horas para a total familiarização das ferramentas da empresa, além do conhecimento de termos geográficos para o entendimento dos projetos.

3.2 Comparação de desempenho entre dados matriciais e vetoriais no SiCAR

A segunda atividade do estágio, realizada nos meses de Agosto e Setembro, se concentra na comparação entre dados na estrutura vetorial e matricial para o cálculo de áreas no módulo de “análise geo” do SiCAR. Esse módulo do sistema permite analisar e calcular as áreas de inconsistências dos imóveis em relação à Lei 12.651 através das tabelas com feições geométricas contidas em seu esquema. O resultado desses cálculos definem, através de um filtro automático, o grupo (Verde, Amarelo ou Vermelho) em que o processo é classificado em relação ao grau de severidade de suas inconsistências. Além disso, esse módulo auxilia os técnicos do cadastro a verificarem a situação do imóvel pela exibição espacial da região, permitindo a identificação de inconsistências não detectadas pelo filtro automático e a regularização de outras que o técnico julgar necessário.

Entretanto, somente dados na representação vetorial são utilizados na modelagem atual do SiCAR. As geometrias referentes a região do país analisada são armazenadas no banco de dados através da importação de arquivos *shapefiles* gerados através do geoprocessamento de imagens de satélite.

Deste modo, a empresa propôs ao estagiário o uso da ferramenta PostGIS em sua versão 2.1, com suporte a dados *raster*, para o cálculo de inconsistências diretamente por essa estrutura. Uma das motivações para o uso dessa estrutura é evitar a geração de *shapefiles*¹ para inserção dos dados espaciais no Banco de Dados, o qual é um processo altamente custoso para imagens de satélite de grande resolução. Além disso, é importante mencionar que a utilização direta do *raster* poderia resultar em uma provável melhoria no desempenho do cálculo de área de sobreposição, visto que a estrutura utilizada atualmente é obtida através do *raster*.

¹No processo de geração de *shapefiles* podem ocorrer diversas inconsistências, tais como: erro de vetorização, suavização de bordas, geração de arquivos muito grandes devido ao grande número de vértices, etc.

Para isso, uma simulação foi realizada utilizando dados em ambas representações, vetoriais e matriciais, a fim de constatar qual delas obteria melhor desempenho para este tipo de cálculo. Para a comparação no uso de dados nas representações vetoriais e matriciais no projeto do SiCAR, foi exigido um estudo aprofundado das funções presentes no PostGIS de modo que ambas as abordagens obtivessem resultados coerentes em relação à área calculada. Dificuldades foram encontradas tanto em relação ao estudo da ferramenta quanto nos testes realizados, incorporando assim a carga horária de 87 horas para a realização dessa tarefa.

3.3 Análise temporal de rasters em Banco de Dados

As aplicações da ferramenta PostGIS para dados *raster* não se limitam somente à cálculos de área de intersecção: é possível também aplicar diversas funções para realizar análises estatísticas sobre o *raster* dentro do Banco de Dados. Deste modo, com o auxílio da equipe de geoprocessamento do LEMAF, foi possível realizar uma análise temporal sobre os *pixels* de alguns *rasters* de modo a reproduzir um novo *raster* a partir da análise das células do original. Especificamente, neste trabalho foram realizadas análises temporais sobre a vegetação de duas imagens de satélite de anos diferentes de uma mesma região.

A utilização de um Banco de Dados para a análise estatística e temporal de *rasters* é uma abordagem ainda não aplicada no laboratório e despertou o interesse dos profissionais da empresa. A propriedade ACID (Atomicidade, Consistência, Isolamento e Durabilidade), geralmente garantidas pelos SGBDs Relacionais, torna-se fundamental para as aplicações de geoprocessamento do laboratório, visto que muitas vezes plataformas como o ArcGIS são utilizadas, as quais acessam as imagens diretamente pelo sistema de arquivos do Sistema Operacional, ocorrendo problemas quando há acesso concorrente à um mesmo arquivo.

Outro fator importante é que uma aplicação SIG com acesso à Banco de Dados poderia realizar análises automáticas sobre *rasters* em *background*, por

meio do Banco de Dados, enquanto fornecesse meios amigáveis de visualização de dados na interface. Embora ainda não seja aplicado nos projetos atuais da empresa, essa ferramenta poderia ser utilizada para monitorar fenômenos que variam continuamente no tempo, como clima e vegetação, sendo que a variação do último pode ocorrer, por exemplo, devido à ações antropogênicas e mudanças sazonais.

É importante ressaltar que existem SGBDs não relacionais que também trabalham com dados *rasterizados*, como Rasdaman² e SciDB³. Neste trabalho, porém, somente a ferramenta PostGIS foi utilizada, visto que os projetos da empresa, em sua maioria, utilizam Banco de Dados Relacionais e a escolha do PostGIS se justifica por decisão arbitrária pela equipe do LEMAF. Ademais, essa extensão é mais apropriada para o tratamento de dados espaciais, uma vez que fornece operadores para dados tanto na estrutura vetorial quanto na estrutura matricial. Uma outra ferramenta do estado da prática que também realiza análises de *rasters* é o Google Earth Engine⁴, uma plataforma em nuvem que utiliza de processamento distribuído para obter um melhor desempenho na análise de imagens de satélite da base de dados da Google.

O desenvolvimento dessa atividade utilizou-se também da versão 2.1 do PostGIS, necessitando de aproximadamente 84 horas, nos meses de Setembro e Outubro, para a total aprendizagem das funções de processamento de *rasters* e para a execução dos testes de análise temporal.

3.4 Participação no projeto SisBolsaVerde

Durante o período de estágio, o LEMAF fechou um acordo com o Governo Federal para o desenvolvimento do projeto SisBolsaVerde, o qual objetiva registrar e monitorar informações das famílias beneficiárias do Programa Bolsa Verde.

²Disponível em <<http://www.rasdaman.org>>. Acesso em: 01 dez. 2014.

³Disponível em <<http://www.scidb.org>>. Acesso em: 01 dez. 2014.

⁴Disponível em <<https://earthengine.google.org>>. Acesso em: 01 dez. 2014.

De acordo com o Ministério do Meio Ambiente (2011), o Programa Bolsa Verde beneficia, a cada trimestre, famílias em situação de extrema pobreza que vivem em áreas de conservação ambiental. O benefício é concedido por dois anos, podendo ser renovado. A proposta do governo é aumentar a renda da população que vive em extrema pobreza na zona rural como forma de incentivo à conservação de ecossistemas e ao uso sustentável dos recursos naturais.

Esse sistema é de extrema importância para a efetivação do Programa, pois através dele será possível verificar, de forma automatizada, quais famílias beneficiárias estão de acordo com os pré-requisitos do Programa, o que atualmente é realizado manualmente pela análise dos documentos das famílias.

As propriedades rurais serão monitoradas através de imagens de satélite, assim como é verificado no SiCAR. O sistema verificará através de uma análise geográfica se a família está cumprindo o atual Código Florestal e analisará também os dados do beneficiário, decidindo quais famílias estão aptas à inserção ou renovação no Programa.

O projeto foi bastante interessante para o desenvolvimento deste relatório, pois com ele foi possível relatar a evolução de um Banco de Dados Geográfico desde o início de seu desenvolvimento. As metodologias empregadas pela empresa para a geração do Banco de Dados também estão documentadas na seção 4.4.2.

Dentre as atividades conduzidas no projeto SisBolsaVerde, consta-se a modelagem do Banco de Dados, a criação do Banco de Dados, a documentação do dicionário de dados e a manutenção de dados. A condução dessa atividade ocorreu em aproximadamente 129 horas nos meses de Setembro e Outubro.

4 DESCRIÇÃO DOS PROCESSOS TÉCNICOS

Esta seção apresenta a metodologia empregada para a condução das atividades do estágio e os passos para alcançar os objetivos previamente mencionados, além de apresentar as principais teorias que suportam cada atividade e as ferramentas de suporte utilizadas neste trabalho.

4.1 Pesquisa bibliográfica e principais ferramentas

4.1.1 Fundamentação Teórica

4.1.1.1 Representação de Dados Geográficos em Sistemas de Banco de Dados

Conforme Guting (1994), um banco de dados espacial ou geográfico é um sistema de banco de dados que oferece suporte a tipos de dados espaciais em seu modelo de dados e linguagem de consulta, além de prover tipos de dados espaciais em sua implementação, como indexação espacial e algoritmos para junção espacial.

Para Filho (1997), um objeto espacial representa uma entidade geográfica do mundo real em um banco de dados geográficos. Esses objetos são codificados utilizando um ou mais pares de coordenadas, o que permite a localização de uma determinada entidade geográfica. Os objetos espaciais são representados principalmente em duas estruturas de dados: estruturas matriciais e estruturas vetoriais.

De acordo com Câmara (2005), as estruturas matriciais ou *rasters* representam um elemento geográfico em células (*pixels*), onde cada célula possui um código referente a uma área específica do objeto. Esse tipo de representação supõe que o espaço é uma superfície plana e utiliza uma matriz de m linhas e n colunas para a representação das células, as quais são acessadas através de suas coordenadas. Como se trata de uma estrutura matricial, o tratamento dos dados depende da resolução do documento, a qual pode ser obtida pela relação entre o tamanho

da célula e a área por ela coberta no terreno. A representação matricial pode ser observada na Figura 4.1.

15	13	16	12	12	13
16	14	17	11	14	15
21	15	18	20	16	14
22	23	22	21	15	15
22	20	24	23	26	25
21	19	20	24	25	26

] tamanho da célula

Figura 4.1: Representação matricial.

Francisco (2004) define a estrutura vetorial como a representação dos vértices do elemento geográfico pelos seus pares de coordenadas. Como pode ser observado na Figura 4.2, esses elementos podem ser representados pelas seguintes feições geométricas dependendo da sua forma e escala cartográfica:

- Ponto: representado por um vértice que define a localização de um objeto espacial que não apresenta área ou comprimento. Por exemplo, dependendo da escala, cidades e hospitais podem ser representados por pontos em um banco de dados geográfico.
- Linha Poligonal ou arco: representado por no mínimo dois vértices conectados que expressam elementos com características como comprimento ou extensão linear. Estradas e rios são exemplos de entidades representadas por linhas.
- Polígono: representado por vários vértices conectados que indicam um objeto com área e perímetro. O último vértice deve ser igual ao primeiro para indicar polígonos fechados. Os polígonos são usados para representar municípios, estados, mapas, entre outros elementos.

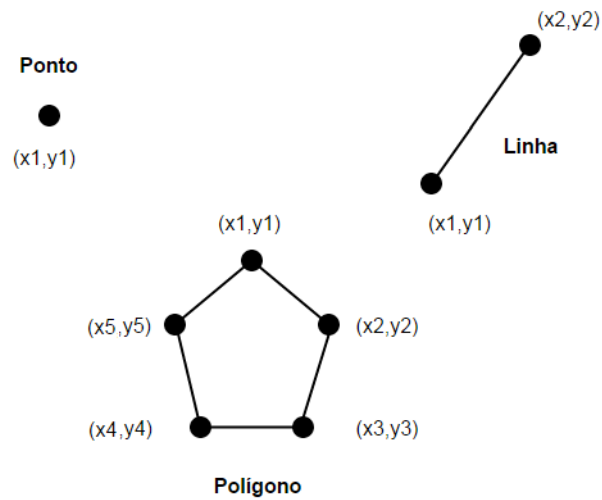


Figura 4.2: Representação vetorial.

Para a representação de um modelo do mundo real, como o da Figura 4.3, é necessário estruturar em camadas as informações geográficas da região analisada. Essas camadas podem ser estruturadas tanto na representação matricial como na representação vetorial e podem fornecer informações sobre diferentes características da região quando os dados são sobrepostos e georreferenciados em um mesmo sistema de referência espacial.

O termo topologia é utilizado para referenciar os relacionamentos espaciais entre objetos em um banco de dados geográficos. "A principal diferença entre os bancos de dados espaciais e os não-espaciais está associada ao fato de que entidades geográficas endereçadas podem ser afetadas por características de entidades vizinhas"(SIDNEY, 2010). Por exemplo, como demonstrado na Figura 4.4, uma entidade A pode conter uma entidade B, assim como A pode estar dentro de B.

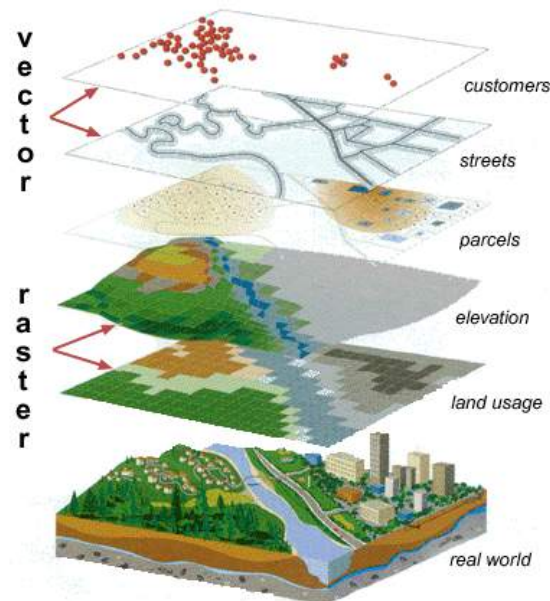


Figura 4.3: Informações geográficas de uma região do mundo real dispostas em camadas.
 Fonte: <http://cherokeecountyiowa.com/offices/gis/index.htm>

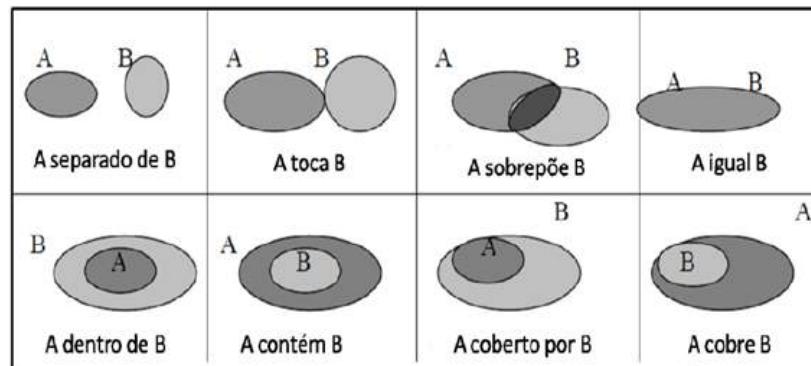


Figura 4.4: Relacionamentos topológicos entre objetos espaciais.
 Fonte: SIDNEY, 2010

4.1.1.2 Sistemas Gerenciadores de Banco de Dados Espaciais

Segundo Rigaux, Scholl e Voisard (2001), o atual crescimento de aplicações em diferentes domínios da computação, como os SIGs, permitiu o aumento do interesse em estender as capacidades de um SGBD tradicional. A estrutura de um SGBD Relacional torna-se interessante para essas aplicações devido as seguintes

características: manipulação de grande volume de dados, tratamento eficiente dos dados através do otimizador de consultas, estruturas de índices e algoritmos de junção, linguagem de consulta SQL extensível e de fácil utilização, segurança dos dados, controle de concorrência, etc.

Atualmente, os principais SGBDs espaciais são extensões de SGBDs populares produzidos por empresas como a Oracle, IBM e Microsoft. A Oracle possui uma extensão chamada Oracle Spatial desde a versão 10g, a IBM estendeu o seu SGBD e deu origem ao IBM DB2 Spatial Extender em sua versão 9.5, a Microsoft incluiu suporte a dados espaciais após o lançamento do SQL Server 2008 e o SGBD de código aberto PostgreSQL possui a extensão PostGIS que atualmente está na sua versão 2.1. Uma vez que o LEMAF utiliza as extensões Oracle Spatial e PostGIS, suas características são discutidas a seguir.

O Oracle Spatial, em sua mais recente versão (11g), oferece diversos recursos como suporte a diversos tipos de dados espaciais, índices espaciais do tipo R-Tree e Quadtree, diversos tipos de relacionamentos topológicos, geocodificação, operações de junções espaciais, criação de *webservices* espaciais, ferramentas de carregamento de dados espaciais, manipulação de imagens matriciais (*GeoRaster*), entre outras inúmeras funcionalidades (ORACLE, 2009).

Em tal SGBD, o esquema MDSYS controla os tipos de dados, pacotes, funções e metadados dos dados espaciais. Nesse esquema, é definido um tipo de objeto chamado SDO_GEOMETRY para a representação de dados espaciais, na qual o objeto espacial contém atributos como o código para indicar a dimensão e o tipo da geometria (SDO_GTYPE), o identificador do sistema de referência espacial da geometria (SDO_SRID), a sua localização (SDO_ORDINATES), entre outros. A definição do objeto SDO_GEOMETRY pode ser visualizada no Script 4.1. Também é possível definir novos tipos de dados espaciais através da linguagem estendida PL/SQL.

A extensão PostGIS é um software livre e gratuito que adiciona tipos dados de geometria, geográficos, *raster* e outros ao SGBD PostgreSQL, assim como funções, operadores e índices GiST¹. Essa extensão também fornece recursos de importação de dados *raster* e de ESRI *shapefiles*, geocodificação, relacionamentos topológicos, entre outros (POSTGRESQL, 2012).

```
CREATE TYPE sdo_geometry (
SDO_GTYPE NUMBER, /* tipo da geometria */
SDO_SRID NUMBER, /* sistema de referência espacial utilizado */
SDO_POINT SDO_POINT_TYPE, /* coordenadas caso o objeto seja um ponto */
SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY, /* informações adicionais do atributo
SDO_ORDINATES */
SDO_ORDINATES SDO_ORDINATE_ARRAY /* coordenadas do objeto */
);
```

Script 4.1: Definição do objeto SDO_GEOMETRY no Oracle Spatial.

O PostGIS é atualmente uma das extensões mais utilizadas do mercado por seguir os padrões da OGC (Open Geospatial Consortium), uma organização que define especificações (WMS, WFS, WCS, etc.) que facilitam o desenvolvimento de aplicações SIGs e, devido a isso, fornece integração com vários SIGs (QGIS, OpenJump, gvSig, etc.) e servidores de mapa (MapServer, GeoServer, QGIS Server, etc.) que também seguem os padrões da OGC. (POSTGRESQL, 2012).

A especificação SFSQL, proposta pela OGC e adotada pelo PostGIS, especifica um conjunto de tipos de geometrias vetoriais, operações topológicas e operações métricas. Nessa especificação, é introduzido o conceito de “tabela com feições” para representação de dados geográficos, na qual os atributos não espaciais são mapeados em colunas de tipos de dados disponíveis na SQL-92 e os atributos espaciais em colunas de tipos de dados geométricos adicionais para SQL.

A Figura 4.5 ilustra a hierarquia de dados espaciais, descritos abaixo, fornecidos por essa especificação.

¹*Generalized Search Tree*. Pacote de estruturas de dados utilizado no PostGIS como estrutura de acesso à dados espaciais.

- Básicos: `ST_Point`, `ST_LineString` e `ST_Polygon`;
- Abstratos: `ST_Curve`, `ST_Surface` e `ST_MultiSurface`;
- Coleções Homogêneas: `ST_MultiPoint`, `ST_MultiLineString` e `ST_MultiPolygon`;
- Coleções Heterogêneas: `ST_GeometryCollection` (atributos com mais de um tipo de geometria).

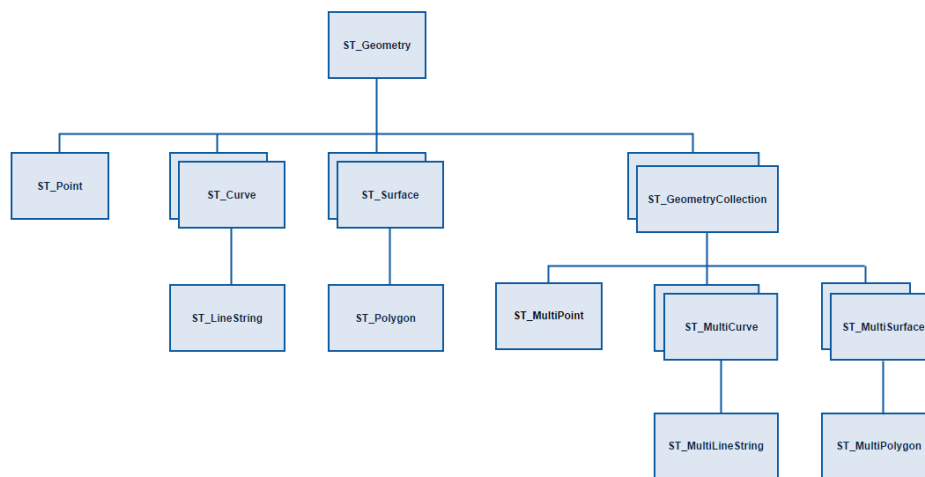


Figura 4.5: Hierarquia dos tipos geométricos da especificação SFSQL.

A SFSQL também propõe o esquema de metadados para representar conjuntos de geo-objetos em tabelas com atributos espaciais. Nesse esquema estão presentes as tabelas *spatial_ref_sys* e *geometry_columns*, onde a primeira armazena dados sobre cada sistema de coordenadas utilizado no banco de dados, já a segunda armazena as colunas geométricas das tabelas com feições. O relacionamento entre essas tabelas, apresentado na Figura 4.6, é realizado quando a função *AddGeometryColumn* é chamada. Essa função registra a tabela com feição na tabela de metadados *geometry_columns* e o SRID da feição, informado em um dos parâmetros da função *AddGeometryColumn*, é relacionado com um dos SRIDs reconhecidos pelo PostGIS na tabela *spatial_ref_sys*.

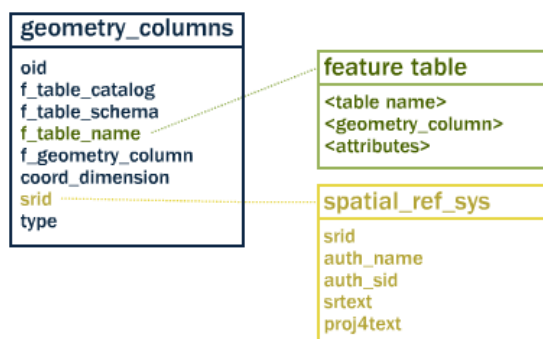


Figura 4.6: Relacionamento entre a tabela com feição geométrica e as tabelas de metadados.

Fonte: <http://workshops.boundlessgeo.com/postgis-intro/geometries.html>

4.1.2 Atividades Executadas

O LEMAF utiliza diversas ferramentas SIGs como suporte para o desenvolvimento de seus projetos. Com relação às ferramentas de Banco de Dados, a empresa utiliza Oracle e PostgreSQL como SGBDs, no qual a escolha de um sobre o outro é decidida juntamente com o cliente. O SGBD PostgreSQL e a sua extensão PostGIS são geralmente preferidos em projetos que tratam de dados geográficos por possuírem um maior número de recursos e um melhor suporte para esse tipo de aplicação, além de que os funcionários da empresa estão mais habituados no uso dessa ferramenta em função dos projetos previamente desenvolvidos.

Dentre as ferramentas utilizadas pela empresa, o presente trabalho utiliza o SGBD PostgreSQL e a sua extensão PostGIS para o gerenciamento de dados espaciais. Como forma de visualização desses dados, o software QGIS é utilizado através de seu complemento “Gerenciador BD”, o qual permite, ao se fazer uma conexão com o Banco de Dados, visualizar graficamente os dados contidos nas tabelas do PostGIS. A ferramenta TortoiseSVN é utilizada para o controle de versionamento dos documentos referentes ao Banco de Dados, tais como *scripts* da evolução do Banco de Dados, dicionário de dados, modelo de dados, etc.

Com relação ao aprendizado das ferramentas, uma série de tutoriais relacionados ao SGBD PostgreSQL e sua extensão para dados espaciais, PostGIS,

foram passados ao estagiário de modo que esse pudesse acompanhar as atividades exercidas pelos administradores de Banco de Dados da empresa.

No primeiro momento, uma revisão da linguagem SQL e das principais funções do SGBD PostgreSQL foi realizada através da apostila PostgreSQL Essencial do curso da Dextraining². Aproximadamente, 50 exercícios foram resolvidos, os quais compreendiam somente questões sobre Banco de Dados relacionais.

Em seguida, um curso de administração de Banco de Dados PostgreSQL foi feito, o que permitiu a compreensão da arquitetura, dos arquivos de configuração (`postgres.conf`, `pg_hba.conf`, entre outros), do gerenciamento dos servidores de banco de dados, tablespaces, usuários e permissões, catálogos de metadados, *backup* e restauração, monitoramento de atividades do Banco de Dados, entre outras particularidades essenciais do SGBD que qualquer administrador deve conhecer para um bom gerenciamento dos dados.

Posteriormente, tutoriais introdutórios sobre as funcionalidades presentes na extensão PostGIS foram estudados para um melhor entendimento da ferramenta. Nesses tutoriais, foram apresentadas as funções adicionais, os novos tipos de dados e os índices espaciais incorporados ao SGBD através de sua extensão para o tratamento de dados espaciais.

Um desses tutoriais, o workshop Boundless de Introdução ao PostGIS³, oferece uma base de dados geográfica na forma de quatro *shapefiles* referentes à cidade de Nova Iorque para serem importados no PostGIS. Os *shapefiles* possuem tanto atributos alfanuméricos para descrição textual dos dados quanto atributos espaciais para descrever as fronteiras da geometria do objeto espacial.

Neste estudo, os *shapefiles* foram importados através do utilitário *shp2pgsql* da extensão PostGIS com o valor de *Spatial Reference System Identifier* (SRID) igual a 26918. O SRID mapeia toda informação sobre a projeção

²Disponível em <<http://www.dextraining.com.br/cursos/postgresql-essencial/>>. Acesso em: 06 out. 2014.

³Disponível em <<http://workshops.boundlessgeo.com/postgis-intro/>>. Acesso em: 08 out. 2014.

do mapa em um único número, neste caso, o SRID 26918 define o sistema de projeção como *Universal Transverse Mercator (UTM) Zone 18 North* e permite expressar em metros os valores de coordenadas dos dados espaciais da cidade de Nova Iorque. Conforme pode ser observado na Figura 4.7, quatro *shapefiles* disponibilizados pelo workshop, *nyc_census_blocks*, *nyc_neighborhoods*, *nyc_streets* e *nyc_subway_stations* descrevem, respectivamente, os setores censitários da cidade, no qual o atributo geométrico são polígonos que representam geograficamente esses setores; os bairros de Nova Iorque, no qual o atributo espacial também é representado por polígonos; as ruas da cidade, no qual o atributo espacial é representado por linhas; as localizações dos metrô da cidade, no qual o atributo espacial é representado através de pontos.

Essa base de dados permitiu a aplicação dos conhecimentos obtidos durante a pesquisa bibliográfica e, através de consultas espaciais, foi possível extrair uma série de informações da cidade de Nova Iorque que não poderiam ser obtidas pelo suporte nativo dos SGBDs convencionais. No Script 4.2, são apresentadas algumas consultas espaciais realizadas sobre os *shapefiles*, importados em forma de tabelas no Banco de Dados.

Na consulta #1, a função *ST_Length* do PostGIS calcula, através do atributo geométrico (*geom*) da tabela *nyc_streets*, o comprimento das linhas que representam cada rua de Nova Iorque. Os valores de comprimento dessas ruas são então agrupados pela função de agregação *sum* e o resultado é dividido por 1.000 para fornecer a resposta em quilômetros. Na consulta #2, uma junção espacial é realizada sobre os atributos geométricos das tabelas *nyc_census_blocks* e *nyc_neighborhoods* onde essas geometrias se interseccionam (a função *ST_Intersects* retorna TRUE quando há interseção entre as geometrias e FALSE caso contrário). Deste modo, é possível obter informações geográficas tanto dos setores censitários como dos bairros, permitindo, por exemplo, o cálculo da densidade demográfica dos bairros *Upper West Side* e *Upper East Side*

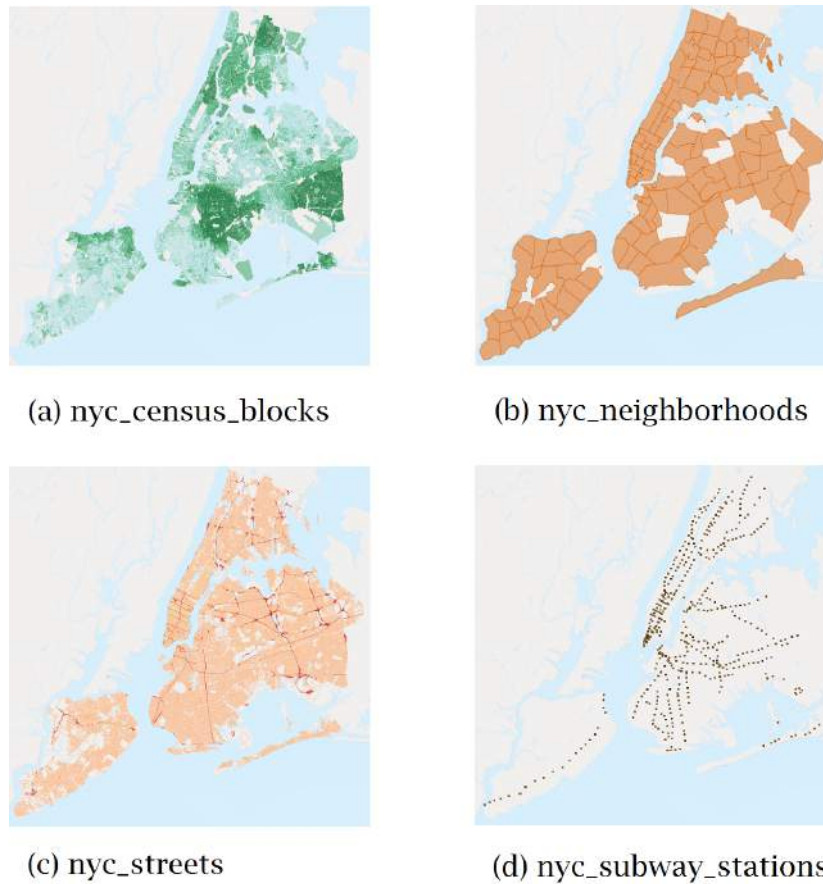


Figura 4.7: Dados espaciais da cidade de Nova Iorque.

Fonte: http://workshops.boundlessgeo.com/postgis-intro/about_data.html

em quilômetros quadrados. De forma similar, a consulta #3 realiza uma junção espacial sobre os atributos geométricos das tabelas *nyc_subway_stations* e *nyc_neighborhoods* para indicar as estações de metrô pertencentes a cada bairro de Nova Iorque (a função *ST_Within* retorna TRUE se a geometria do primeiro parâmetro está contida na geometria do segundo parâmetro e retorna FALSE caso contrário). Assim, é possível descobrir as estações de metrô do bairro Little Italy e as rotas de percurso realizadas pelo metrô.

É importante ressaltar que o desempenho das junções espaciais está intimamente ligado à utilização do índice espacial. Para essas consultas, o índice GiST foi criado diretamente pelo utilitário *shp2pgsql* através do parâmetro *-I*.

```

/* Consulta espacial #1 - retorna o comprimento, em quilômetros, de todas
   as ruas de Nova Iorque. */
SELECT sum(ST_Length(s.geom))/1000 AS comprimento
FROM nyc_streets AS s;

/* Consulta espacial #2 - retorna o nome e a densidade demográfica, em
   quilômetros quadrados, dos bairros Upper West Side e Upper East Side.
   */
SELECT n.name, sum(c.popn_total)/(ST_Area(n.geom) / 1000000) as densidade
FROM nyc_census_blocks AS c
INNER JOIN nyc_neighborhoods AS n
ON ST_Intersects(n.geom,c.geom)
WHERE n.name = 'Upper West Side'
OR n.name = 'Upper East Side'
GROUP BY n.name, n.geom;

/* Consulta espacial #3 - retorna o nome das estações de metrô do bairro
   Little Italy e as suas rotas de percurso. */
SELECT ss.name, ss.routes
FROM nyc_subway_stations AS ss
INNER JOIN nyc_neighborhoods AS n
ON ST_Within(ss.geom,n.geom)
WHERE N.name = 'Little Italy';

```

Script 4.2: Exemplos de consultas realizadas sobre a base de dados de Nova Iorque.

4.2 Comparação de desempenho entre dados matriciais e vetoriais no SiCAR

4.2.1 Fundamentação Teórica

Segundo Brinkhoff, Kriegel e Seeger (1993), a necessidade de criação de métodos de acesso mais eficientes para consultas espaciais surgiu por meio dos altos requisitos de desempenho e da grande complexidade envolvendo esse tipo de consulta em sistemas de banco de dados geográficos.

Como as consultas espaciais envolvem apenas parte do banco de dados, um plano de execução de consulta deve começar por uma fase de filtragem para

a identificação de objetos que satisfazem a declaração da consulta. Essa filtragem seleciona os possíveis candidatos que satisfazem a consulta e depende de estruturas de acesso como índices espaciais, os quais geralmente são definidos sobre aproximações das geometrias dos objetos, conhecidas como *bounding boxes*.

De acordo com workshop Boundless (2011), um *bounding box*, demonstrado na Figura 4.8, é o menor retângulo paralelo aos eixos de coordenadas capaz de conter uma dada feição. *Bounding boxes* são utilizados para responder a pergunta “a feição A está contida em B?”, que é computacionalmente custosa para polígonos, porém fácil de ser respondida para retângulos.

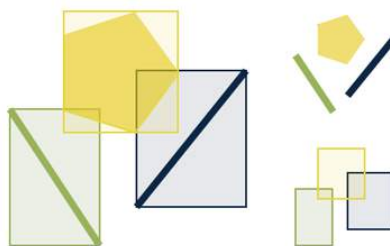


Figura 4.8: *Bounding Boxes* cobrindo diferentes tipos de feições.
Fonte: <http://workshops.boundlessgeo.com/postgis-intro/geometries.html>

Após a etapa de filtragem, algoritmos geométricos computacionalmente custosos são aplicados à geometria exata dos candidatos selecionados na etapa de filtragem para se obter o resultado correto da consulta.

Ao longo do tempo, diversos índices espaciais foram implementados com o objetivo de otimizar consultas espaciais que não podiam ser resolvidas em tempo hábil por índices utilizados em dados não-espaciais, por exemplo, o índice do tipo B-tree. São exemplos de índices espaciais: Quadtree, R-tree e K-Nearest Neighbors (KNN). Os principais SGBDs espaciais utilizam principalmente os índices R-tree e Quadtree.

A Quadtree (Figura 4.9) é uma estrutura de dados organizada em uma árvore quaternária, onde cada nó representa uma região quadrada do espaço. Essa região é subdividida em outras quatro partes a medida que a árvore caminha para

um nível inferior, repetindo o processo até que haja somente um ou nenhum objeto geográfico em todas as regiões quadráticas. O processamento de consultas por região utilizada nessa estrutura inicia-se a partir da raiz e expande somente os nós cujos quadrados estiverem em contato com um objeto geográfico, descartando todas as ramificações abaixo dos nós que não possuam essa configuração.

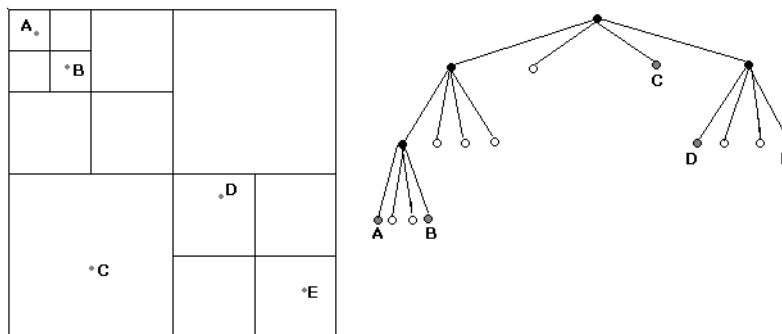


Figura 4.9: Árvore Quadtree.

Fonte: <https://www.cs.umd.edu/users/mount/Indep/Ransom>

A R-tree (Figura 4.10) é uma estrutura de dados hierárquica baseada na árvore B que também utiliza páginas com chaves de busca. Ao invés de indexarem valores numéricos, as chaves da R-tree armazenam os pontos extremos dos retângulos que cobrem o objeto espacial. Essa estrutura tenta encontrar o *bounding box* do objeto espacial através dos pontos máximos e mínimos do contorno do objeto, expressos pelas coordenadas de seus pontos inferior esquerdo e superior direito. Os nós internos possuem a delimitação dos retângulos que contém todos retângulos dos nós de níveis inferiores e a eficiência desse método depende da qualidade em que a R-tree atribui os *bounding boxes* aos nodos.

Uma comparação de desempenho entre esses dois índices foi realizada por Velicanu e Olaru (2010) na extensão Oracle Spatial, constatando que apesar do Quadtree obter melhor desempenho em consultas espaciais complexas, é aconselhável o uso de índices R-tree devido ao seu melhor desempenho em consultas espaciais básicas.

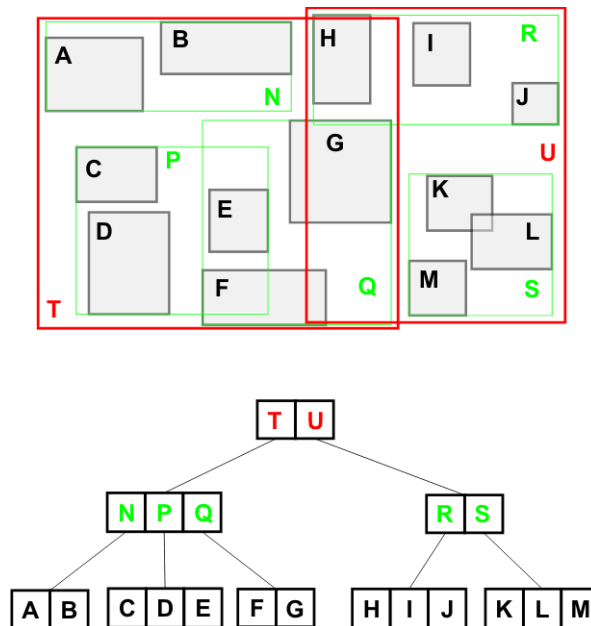


Figura 4.10: Árvore R-tree.

Fonte: http://cglab.ca/~cdillaba/comp5409_project/images/rtree.png

O PostGIS utiliza índices GiST (*Generalized Search Tree*) para indexação espacial. Os desenvolvedores definem GiST como uma estrutura de dados extensível que permite usuários desenvolverem índices sobre qualquer tipo de dado, suportando qualquer pesquisa sobre esses dados. Esse pacote unifica uma série de árvores de busca como R-trees, B+-trees, hB-trees, etc., em uma única estrutura de dados (HELLERSTEIN, 1999). No caso do PostGIS, o índice GiST é geralmente definido sobre uma coluna do tipo *geometry*.

As otimizações de consultas espaciais, porém, não envolvem somente o nível lógico do banco de dados. Casanova (2005) afirma que o subsistema de armazenamento deve implementar os métodos de armazenamento que administram a utilização de páginas físicas na memória secundária (local onde residem índices e dados espaciais) e deve tratar problemas de espaço para dados geográficos que ocupam mais de uma página, armazenando as páginas físicas referentes a um

mesmo objeto espacial em posições vizinhas na memória a fim de diminuir o custo de acesso e obter resultados mais rápidos.

4.2.2 Atividades Executadas

4.2.2.1 Metodologia e dados utilizados

A utilização de dados na estrutura matricial para o cálculo de áreas de sobreposição de imóveis, atualmente realizado no SiCAR por meio de dados vetoriais, é uma abordagem que ainda não havia sido explorada no LEMAF. Deste modo, para comprovar a eficiência dessa abordagem, foram realizados testes comparativos entre as representações matricial e vetorial para simular situações já tratadas atualmente pelo sistema.

Um dos critérios utilizados no SiCAR para avaliar a situação ambiental de uma propriedade rural refere-se ao uso da cobertura do solo da região onde o imóvel se situa. Por exemplo, o cadastrante poderia indicar áreas de vegetação nativa que o seu imóvel não possui na realidade, o que poderia indicar, dependendo do grau de tolerância estabelecido pela Lei 12.651, o uso irregular do solo. Nesse caso, a situação do cadastro desse imóvel se tornaria irregular e seria fornecido um prazo ao cadastrante para que esse regulamentasse o seu cadastro.

Atualmente, o arquivo *shapefile* utilizado pelo SiCAR para a análise espacial da cobertura do solo de imóveis localizados no estado do Mato Grosso foi gerado através do *raster* temático da Figura 4.11, o qual possui uma única banda de resolução de 49746x71819 *pixels* de 5 metros cada. Por se tratar de um *raster* temático, os valores de *pixels* da imagem representam diferentes classificações da cobertura do solo:

- Área Antropizada após 2008: valor de *pixel* igual à 16 (Desmatamento 2008 - 2011);

- Área Consolidada: valores de *pixels* iguais à 4 (Área Antrópica) e 18 (Área Urbana);
- Área em Regeneração: valor de *pixel* igual à 17 (Regeneração 2008 - 2011);
- Curso D'água: valor *pixel* igual à 13 (Água);
- Remanescente de Vegetação Nativa: valores de *pixels* iguais à 3 (Remanescente Nativo) e 10 (Área Natural).

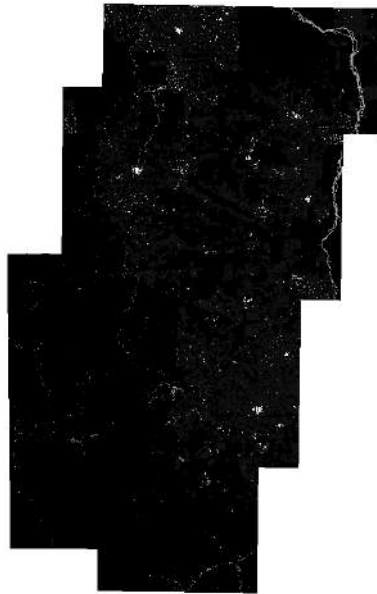


Figura 4.11: Mapeamento da cobertura do solo do estado do Mato Grosso.

A importação desse *raster* no Banco de Dados se deu através do utilitário *raster2pgsql* da ferramenta PostGIS em sua versão 2.1. Nesse utilitário, o parâmetro `-t` permite “recortar” o *raster* em blocos de menores, conhecidos como *tiles*⁴, e armazená-los como registros da coluna *raster* da tabela. A escolha do tamanho dos *tiles*, muitas vezes empírica e dependente do problema a ser tratado, refletem diretamente no tempo de execução de consultas *raster*, visto que permitem um melhor aproveitamento do índice espacial pela divisão do *raster* em blocos menores.

⁴É importante destacar que cada *tile* possui um identificador (*rid*) e que também é possível selecionar um recorte de uma região específica do *tile* através da função *ST_Clip*.

Nos testes realizados neste trabalho, a tabela *full_raster_table* foi criada a partir de *tiles* de 100x100 *pixels* do *raster* da Figura 4.11. Esse tamanho foi escolhido por ter obtido um melhor desempenho em consultas *raster* em comparação com outros tamanhos testados, como *tiles* de 50x50, 200x200 e 400x400 *pixels*. Por esse motivo, a comparação de desempenho entre as estruturas matriciais e vectoriais somente considerará os tempos de execução de consultas *raster* com *tiles* de 100x100 *pixels*.

Atualmente no SiCAR, as áreas de sobreposição são calculadas separadamente de acordo com a classificação da cobertura do solo. Por esse motivo, os testes foram limitados à classificação de Remanescentes de Vegetação Nativa, indicados pela cor verde na Figura 4.12, pois os cálculos realizados nas outras classificações são idênticos e somente variam as porcentagens de áreas sobrepostas permitidas por lei.



Figura 4.12: Remanescentes de vegetação nativa do estado do Mato Grosso.

Para essa limitação, a tabela *raster_table* foi criada a partir de uma reclassificação binária dos *pixels* da tabela *full_raster_table* através da função

ST_Reclass, como observado no *Script 4.3*. Foram utilizados como parâmetros dessa função: a coluna *raster* da tabela que seria reclassificada, o tipo composto *reclassarg* para definir expressões de reclassificação, o *pixeltype* da banda do *raster* gerado pela reclassificação e o valor de *NODATA* para *pixels* que representam nenhuma informação.

```
CREATE TABLE raster_table AS
SELECT rid,
       ST_Reclass(
         rast,'0-2:0, 3:1, 4-9:0, 10:1, 11-255:0', '1BB',0
       ) AS rast
FROM full_raster_table;
```

Script 4.3: Consultas para reclassificação binária da tabela *full_raster_table*.

A respeito das expressões utilizadas no parâmetro *reclassarg*, o caractere especial '-' define uma faixa de valores de *pixels*, já o caractere especial ':' separa os valores de *pixels* de antes e depois da reclassificação. No caso da tabela *full_raster_table*, os valores de *pixels* 3 e 10, os quais representam a classificação de remanescentes de vegetação nativa, foram reclassificados com o valor 1, já os demais valores foram reclassificados com o valor 0. Como os valores da reclassificação possuíam somente 2 valores (0 e 1), o *pixeltype* da banda do *raster* reclassificado foi definido como *1BB*, o qual utiliza somente 1 bit para representação de valores de *pixels*. O *raster* original possuía uma banda de *pixeltype* *8BUI*, o qual utiliza 8 bits sem sinal para representação de valores de *pixels* de 0 a 255. Em seguida, o valor *NODATA* foi definido como 0, a fim de considerar somente os *pixels* com valores iguais a 1, ou seja, *pixels* que representam remanescentes de vegetação nativa.

Após a reclassificação, como apresentado no *Script 4.4*, *constraints raster* foram adicionadas na tabela *raster_table* pela função *AddRasterConstraints*, responsável pelo registro de tabelas *rasters* na tabela de metadados *raster_columns*. O índice GiST foi definido sobre a mínima geometria convexa que cobre o *raster* pela função *ST_ConvexHull*.

```

SELECT AddRasterConstraints('public', 'raster_table', 'rast', TRUE, TRUE,
    TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, TRUE);

CREATE INDEX raster_table_rast_idx ON raster_table
USING GIST (ST_ConvexHull(rast));

```

Script 4.4: Consultas para adicionar *constraints* e o índice GiST na tabela *raster_table*.

Para garantir a mesma indexação espacial para os dados na estrutura vetorial, a tabela *raster_table* foi vetorizada pela consulta presente no *Script 4.5* para que cada *tile* se tornasse um polígono, como os da Figura 4.13. Deste modo, o índice GiST é definido sobre *bounding boxes* de mesmo tamanho para ambas as estruturas. As funções de conversão de *raster* para geometria no PostGIS diferem-se em relação ao tratamento de valores *NODATA* de *pixels* e do formato dos *tiles*, o que pode ser visualizado na Figura 4.14.

```

CREATE TABLE vector_table AS
SELECT rid as gid, ST_Polygon(rast) AS geom
FROM raster_table
WHERE ST_Polygon(rast) IS NOT NULL;

```

Script 4.5: Consulta para vetorização da tabela raster.

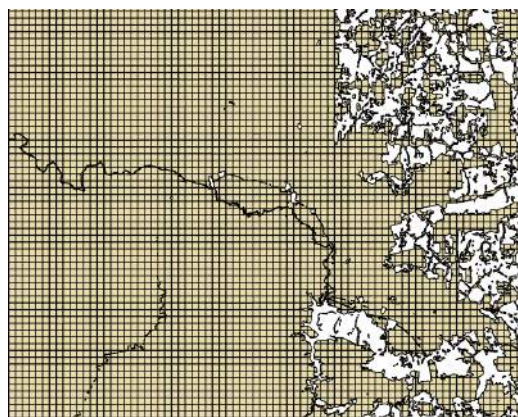


Figura 4.13: Imagem ampliada das geometrias da tabela *vector_table*.

Após a criação das tabelas vetoriais e matriciais, dois *shapefiles* foram importados no Banco de Dados a fim de simular o cálculo de sobreposição das propri-

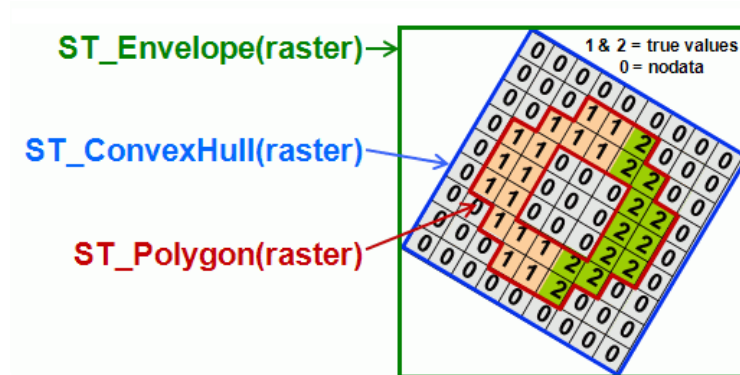


Figura 4.14: Diferença entre as funções de conversão de raster para geometria.
Fonte: <http://trac.osgeo.org/postgis/wiki/WKTRaster/SpecificationFinal01>.

idades rurais com regiões de remanescentes de vegetação nativa. O *shapefile* #1, apresentado na Figura 4.15, é um simples polígono que visa simular a geometria dos limites de uma única propriedade rural, a qual está situada em uma região onde há cursos de água e remanescentes de vegetação nativa. O *shapefile* #2, apresentado na Figura 4.16, foi gerado através da seleção de 1.000 polígonos aleatórios da tabela *vector_table* para o cálculo de área de diversos imóveis rurais.

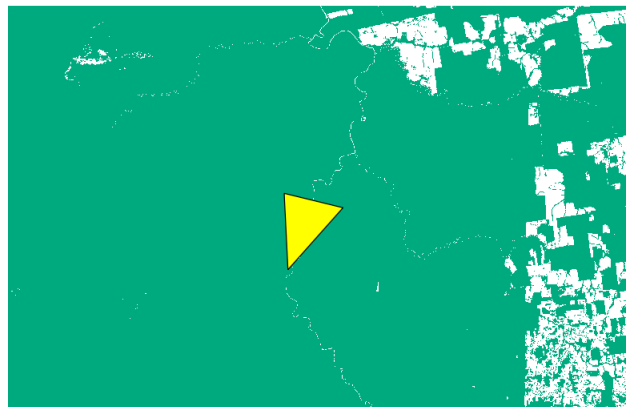


Figura 4.15: Polígono representando uma única propriedade rural.

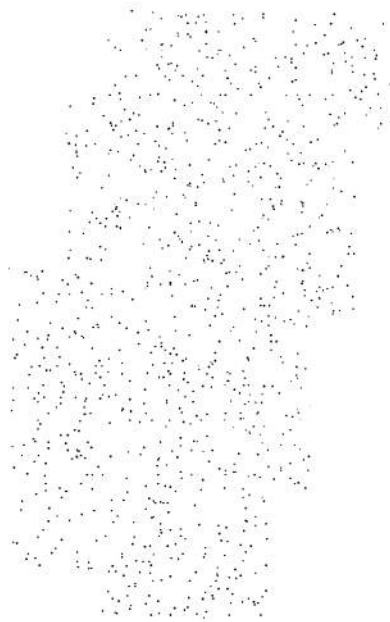


Figura 4.16: Polígonos representando 1.000 propriedades rurais.

4.2.2.2 Execução dos Testes

As quatro consultas utilizadas nos testes são apresentadas no *Script 4.6*, no qual as mesmas são realizadas sobre ambos os *shapefiles*, uma vez que a tabela *polygon_table* é utilizada para representar qualquer um dos *shapefiles* utilizados para os testes. Essas consultas visam o cálculo da área de sobreposição, porém utilizam diferentes estratégias por meio de funções do PostGIS fornecidas para os tipos de dados *raster* e *geometry*.

```

/* Consulta #1 - Interseção geometria/geometria */
SELECT sum(ST_Area(ST_Intersection(V.geom,V2.geom)))
FROM vector_table as V
     INNER JOIN polygon_table as V2
     ON (ST_Intersects(V.geom,V2.geom));

/* Consulta #2 - Interseção raster/geometria */
SELECT sum(ST_Area((intersection).geom))
FROM (
     SELECT ST_Intersection(rast, geom) AS intersection
     FROM raster_table
          INNER JOIN polygon_table
          ON ST_Intersects(rast, geom)
     ) AS intersection;

/* Consulta #3 - Vetorização do recorte da interseção raster/geometria */
SELECT sum(ST_Area(ST_Polygon(ST_Clip(rast, geom))))
FROM raster_table
     INNER JOIN polygon_table
     ON ST_Intersects(rast, geom);

/* Consulta #4 - Contagem de Pixels do recorte da interseção raster/
geometria */
SELECT (num_pixels * abs(scale_x) * abs(scale_y))
FROM (
     SELECT sum(ST_Count(ST_Clip(rast,geom))) AS num_pixels
     FROM raster_table
          INNER JOIN polygon_table
          ON ST_Intersects(rast, geom)
     ) AS intersection
CROSS JOIN
(
     SELECT ST_ScaleX(rast) as scale_x, ST_ScaleY(rast) as scale_y
     FROM raster_table
     LIMIT 1
     ) AS scales;

```

Script 4.6: Consultas dos testes realizados para avaliar o desempenho das estruturas matriciais e vetoriais.

A consulta #1 calcula a soma das áreas de interseção dos polígonos que interseccionam a tabela *vector_table* através do uso exclusivo de geometrias. A consulta #2 utiliza a mesma função da consulta #1, *ST_Intersection*, porém a utiliza em sua variante que recebe um *raster* e um *geometry* como parâmetros e retorna um *geomval*, um tipo de dado que representa um registro de pares geometria-valor. Nessa consulta, um *SELECT* aninhado teve de ser realizado para extrair somente as geometrias do registro *geomval* para que assim a área pudesse ser calculada. A consulta #3 utiliza uma abordagem um pouco diferente da consulta #2, no qual

primeiramente o *raster* é recortado pela função *ST_Clip* nas regiões onde há interseção com as geometrias da tabela *vector_table*, para que posteriormente esse recorte seja vetorizado pela função *ST_Polygon*, permitindo o cálculo da soma das áreas. A consulta #4 calcula a soma das áreas através da junção de dois *SELECTs* aninhados, no qual o primeiro conta, através da função *ST_Count*, o número de *pixels* do *raster* recortado e o segundo recupera as escalas X e Y dos *pixels* do *raster* através das funções *ST_ScaleX* e *ST_ScaleY*, respectivamente. O produto cartesiano é utilizado simplesmente para realizar uma junção entre as duas tuplas recuperadas. Após a obtenção dos dois resultados, a quantidade de *pixels* é multiplicada pelo tamanho desses nas dimensões X e Y de modo a obter a área sobreposta.

O *hardware* utilizado para a execução das consultas dos testes foi um Notebook DELL XPS L502X, com sistema operacional Microsoft Windows 7 Home Premium (64 bits), processador Intel Core i7-2670QM, 6GB de memória RAM e disco WD SATA 1TB.

É importante ressaltar que (i) o Servidor do Banco de Dados foi reiniciado a cada execução para liberar espaço de memória nos *buffers* e *caches* utilizados no PostgreSQL e (ii) a memória *cache* utilizada pelo Sistema Operacional também foi limpada antes de cada execução.

O processo experimental baseou-se em uma observação pareada dos tempos de execução, em segundos, das consultas utilizando dados vetoriais e matriciais. Deste modo, a consulta #1 foi avaliada em relação às consultas #2, #3 e #4 para os dois *shapefiles* de teste. A distribuição *t-Student* foi utilizada para calcular o intervalo de confiança para definir qual consulta obteve melhor desempenho. A fórmula dessa distribuição é transcrita a seguir:

$$IC = \bar{x} \pm t_{[1-\frac{\alpha}{2}; n-1]} \left(\frac{s}{\sqrt{n}} \right) \quad (4.1)$$

Inicialmente, o tamanho dos pares de amostra (n) foi definido como 9 e a diferença nos tempos de execução de cada cada par foi calculada a fim de obter a média aritmética (\bar{x}) e o desvio padrão (s) dos resultados. Um intervalo de confiança de 90% foi determinado para as amostras da distribuição *t-Student*, no qual $t_{[0.95;8]}$ é equivalente a 1,860. As Tabelas 4.1 e 4.2 reportam os resultados da comparação pareada, referentes às diferenças entre os tempos de execução das amostras. Para fins de conferência, os tempos de execução de cada amostra, a média aritmética e a área em metros quadrados estão reportados no Apêndice A.

Tabela 4.1: Comparação pareada (*shapefile 1*).

	Média (s)	Desvio Padrão (s)	Intervalo de Confiança
Consultas #1 e #2	-0.439	0.079	(-0.518, -0.420)
Consultas #1 e #3	-1.299	0.167	(-1.403, -1.195)
Consultas #1 e #4	-1.281	0.157	(-1.378, -1.183)

Tabela 4.2: Comparação pareada (*shapefile 2*).

	Média (s)	Desvio Padrão (s)	Intervalo de Confiança
Consultas #1 e #2	-86.059	2.854	(-87.828, -84.289)
Consultas #1 e #3	-5.825	2.608	(-7.442, -4.207)
Consultas #1 e #4	-4.337	1.664	(-5.368, -3.305)

4.2.2.3 Resultados e Discussão

Como observado nas Tabelas 4.1 e 4.2, os intervalos de confiança não incluem o valor 0 em todas as 6 observações desse experimento e o sinal negativo indica que a consulta #1 possui um melhor desempenho independente do *shapefile* utilizado.

A consulta #2 obteve um pior resultado em relação à consulta #1 porque a variante *raster* da função *ST_Intersection* primeiramente vetoriza o *raster* para depois utilizar a sua variante *geometry*, ou seja, o tempo para a conversão do *raster* em geometria é incluído no tempo de execução da consulta. Outro ponto a se

considerar é que essa conversão aumenta muito o tempo de execução quanto maior a área de interseção, como no *shapefile* #2 em relação ao *shapefile* #1.

A consulta #3 adota uma abordagem geralmente mais rápida e menos precisa do que a consulta #2, visto que a função *ST_Clip* é menos custosa por incluir no *raster* recortado somente os *pixels* que possuem mais de 50% de sua área interseccionando a geometria, desconsiderando a possibilidade da geometria interseccionar somente pedaços do *pixel*, o que pode acarretar no aumento ou na diminuição da área calculada devido aos *pixels* que fazem interseção com as fronteiras da geometria. Ao vetorizar o *raster* recortado, o resultado sempre será divisível pelo valor da área coberta por um *pixel*, ou seja, a precisão do cálculo é aumentada quanto maior é a resolução da imagem. As diferenças dos resultados das funções *ST_Intersection* e *ST_Clip* em conjunto com a função *ST_Polygon* podem ser visualizadas nas fronteiras do *shapefile* #1 na Figura 4.17. Apesar de ser mais rápida que a consulta #2 nos dados do *shapefile* #2, a consulta #3 obtém um pior desempenho em relação à consulta #1 independente do *shapefile* utilizado.

A consulta #4 produz o mesmo resultado que a consulta #3 para o cálculo de área de interseção por também utilizar a função *ST_Clip*, porém, ao invés de vetorizar o *raster* recortado, contabiliza os *pixels* do *raster* e os multiplica pela área coberta por eles. O desempenho da função *ST_Count* é diretamente proporcional ao número de elementos presentes na matriz de *pixels* do *raster* e deve verificar o valor de cada *pixel* para garantir que a interseção ocorrerá somente com *pixels* que não possuam valores *NODATA*. Assim como a consulta #3, apresentou uma precisão e um desempenho pior em relação à consulta #1.

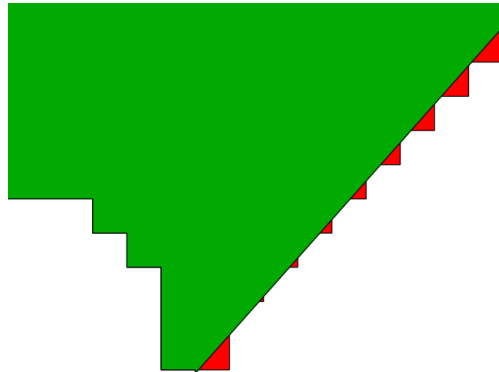


Figura 4.17: Diferenças entre os resultados das consultas 2 (verde) e 3 (vermelho) nas bordas do polígono do *shapefile* 1.

4.3 Análise temporal de rasters em Banco de Dados

4.3.1 Fundamentação Teórica

De acordo com ARAUJO e MELLO (2010), o uso de imagens de satélite tem demonstrado grande utilidade em levantamentos e pesquisas geológico-ambientais. Tal fato se deve à razão de bandas espectrais, que utiliza de operações matemáticas para obter sutis diferenças existentes no comportamento espectral de diferentes objetos, o que permite discriminar diversos alvos em uma mesma imagem.

Uma das inúmeras aplicações da razão de bandas se encontra no uso de índices de vegetação para monitoramento de coberturas vegetais. Dentre os índices de vegetação empregados, o *Normalized Difference Vegetation Index* (NDVI) é um dos indicadores mais sensíveis à quantidade de vegetação, cujos valores variam no intervalo de -1 a 1.

O cálculo do NDVI é baseado na razão entre a diferença das refletividades das bandas no infravermelho próximo e no vermelho pela soma dessas mesmas refletividades (JR *et al.*, 1974). Em suma, a razão de bandas ocorre pela seguinte fórmula:

$$NDVI = \frac{NIR - R}{NIR + R} \quad (4.2)$$

Onde NIR representa a energia refletida na região do infravermelho próximo e R a energia refletida na região do vermelho do espectro eletromagnético.

4.3.2 Atividades Executadas

A análise temporal realizada nesse trabalho considera duas imagens de 5.000x5.000 *pixels*, retiradas por meio de imagens de satélites RapidEye nos anos de 2011 e 2012, e visa identificar as regiões onde houveram aumento ou diminuição no índice de vegetação no ano de 2012 em relação ao ano de 2011. Através dessas imagens, encontradas nas Figuras 4.18 e 4.19, as tabelas *r_2011* e *r_2012* foram criadas no Banco de Dados com *tiles* de tamanho de 200x200 *pixels*, que é divisível por 5000x5000, o que evita o uso de *pixels* de valores NODATA em todos os *tiles*. Os índices espaciais, assim como as *constraints* para registro nas tabelas de metadados, foram criados sobre todas as tabelas descritas nesta seção pelos mesmos comandos presentes no *Script* 4.4.



Figura 4.18: Imagem RapidEye de 2011.



Figura 4.19: Imagem Rapideye de 2012.

Nessa análise, o NDVI foi utilizado para identificar as regiões de vegetação e de desmatamento em ambas as tabelas, nas quais os valores de *pixels* mais próximos de 1 indicam regiões com vegetação e os mais próximos de -1 regiões desmatadas. No *Script 4.7*, após a obtenção das imagens NDVI em ambos os anos, a diferença entre o NDVI de 2012 e de 2011 é computada para a criação da tabela *ndvi_diferenca*, onde os *pixels* com valores mais próximos de 2 indicam áreas que foram regeneradas e os *pixels* de valores mais próximos de -2 indicam áreas que foram desmatadas durante o intervalo de tempo em que as fotos de 2011 e 2012 foram retiradas.

```

CREATE TABLE ndvi_diferenca AS
SELECT r_2011.rid,
       ST_SetBandNoDataValue (
         ST_MapAlgebra (
           ST_SetBandNoDataValue (
             ST_MapAlgebra(r_2012.rast, 4, r_2012.rast, 3,
               'fn_NDVI(float[][][], integer[][], text[])'
               ::regprocedure, '32BF'::text), NULL
           ),
           1,
           ST_SetBandNoDataValue (
             ST_MapAlgebra(r_2011.rast, 4, r_2011.rast, 3,
               'fn_NDVI(float[][][], integer[][], text[])'
               ::regprocedure, '32BF'::text), NULL
           ),
           1,
           'fn_NDVI_diferenca(float[][][], integer[][],
             text[])'::regprocedure, '32BF'::text
         ), NULL
       ) AS rast
FROM r_2011
INNER JOIN r_2012
ON (r_2011.rid = r_2012.rid);

```

Script 4.7: Consultas para criação da tabela *ndvi_diferenca*.

No PostGIS, a função *ST_MapAlgebra* é responsável por aplicar uma função algébrica em cada *pixel* de um determinado *raster* e retornar o *raster* correspondente em uma única banda. Nesse caso, a função algébrica *fn_NDVI*, presente no *Script 4.8*, foi criada com a finalidade de realizar o cálculo do NDVI nas tabelas *r_2011* e *r_2012* ao utilizar os *pixels* das bandas 4 e 3 como parâmetro. Como o resultado do NDVI é geralmente um ponto flutuante, o *pixeltype* 32BF (32 bits com sinal) foi atribuído à banda do *raster* resultante. É importante destacar que a criação de uma nova tabela para cada processamento é opcional, visto que o resultado do processamento é recuperado pela cláusula *SELECT*.

```

CREATE OR REPLACE FUNCTION fn_NDVI(pixel_value float[][][], pos integer
  [][], VARIADIC userargs text[])
RETURNS FLOAT AS
$BODY$
BEGIN
  IF (pixel_value[1][1][1] + pixel_value[2][1][1]) = 0 THEN
    RETURN
      (pixel_value[1][1][1] - pixel_value[2][1][1])
      / 0.000001::float;
  ELSE
    RETURN
      (pixel_value[1][1][1] - pixel_value[2][1][1])
      / (pixel_value[1][1][1] + pixel_value[2][1][1])::float;
  END IF;
END;
$BODY$
LANGUAGE 'plpgsql' IMMUTABLE;

CREATE OR REPLACE FUNCTION fn_NDVI_diferenca(pixel_value float[][][], pos
  integer[][][], VARIADIC userargs text[])
RETURNS FLOAT AS
$BODY$
BEGIN
  RETURN (pixel_value[1][1][1] - pixel_value[2][1][1]);
END;
$BODY$
LANGUAGE 'plpgsql' IMMUTABLE;

```

Script 4.8: Funções utilizadas na álgebra de mapas para o cálculo do NDVI e do NDVI diferença.

Após realizar a álgebra de mapas, a função *ST_SetBandNoDataValue* é utilizada para garantir que a banda do *raster* gerado não possua *pixels* com valores *NODATA*. Isto é feito porque a função *ST_MapAlgebra* atribui automaticamente como *NODATA* o menor valor representado pelo *pixeltype* do *raster* original (POSTGIS, 2014). No caso das tabelas *r_2011* e *r_2012*, esse número corresponde ao menor número representado pelo *pixeltype* 8BUI, o valor 0, o que pode afetar os cálculos posteriores, visto que 0 é um valor aceitável na faixa de valores possíveis do NDVI, os quais compreendem os valores entre -1 até 1.

Por fim, a função algébrica *fn_NDVI_diferenca*, também presente no *Script 4.8*, é aplicada sobre os *rasters* gerados para calcular a subtração entre os *pixels* do NDVI de 2012 (Figura 4.21) e os *pixels* do NDVI de 2011 (Figura 4.20). No

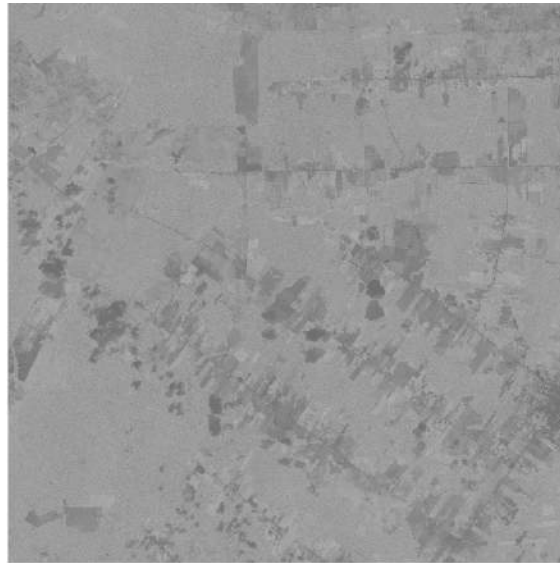


Figura 4.20: NDVI da Imagem de 2011.

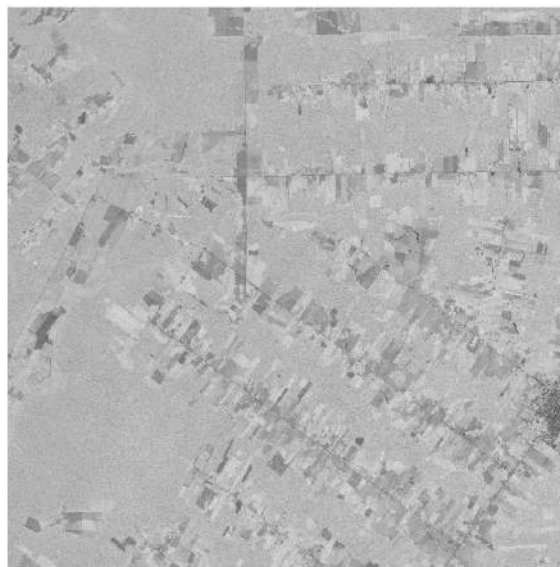


Figura 4.21: NDVI da Imagem de 2012.

final, a função `ST_SetBandNoDataValue` determina que a tabela *ndvi_diferenca* não possui *pixels* com valores *NODATA*.

Para analisar melhor as áreas vegetadas e desmatadas, foi gerado a tabela *shape* com vários polígonos criados a partir do *raster* da diferença dos NDVIs.

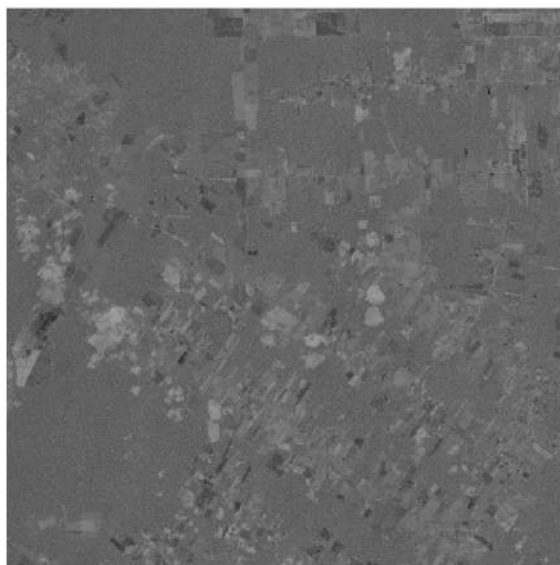


Figura 4.22: Diferença dos NDVIs de 2011 e 2012.

Através da interseção desses polígonos com a tabela *ndvi_diferenca*, é possível atribuir, a cada polígono, a média dos valores de *pixels* das regiões de interseção com o *raster*. Deste modo, é possível analisar a vegetação pelos valores dos polígonos, o que torna a análise mais viável, pois a detecção de *pixels* mais escuros ou mais claros a olho nu é mais difícil do que a análise por regiões.

Primeiramente, antes de calcular a média para cada polígono, os polígonos que faziam interseção com nuvens do *raster* de 2011 foram eliminados, pois a presença de nuvens influencia no cálculo do NDVI, visto que por elas e suas sombras cobrirem áreas da imagem, não seria possível afirmar a presença ou ausência de vegetação nessas regiões.

Para eliminar essas regiões, a máscara de nuvens da Figura 4.23 foi vetorizada onde os valores de *pixels* correspondiam a 2, o que indicava a presença de nuvens na imagem. Essa máscara, porém, não garante a eliminação de todas as nuvens da imagem. Após isso, a função *ST_Difference* foi aplicada para eliminar os polígonos da tabela *shape* que faziam interseção com as nuvens da tabela *nuvem*, gerando o *shapefile* da Figura 4.24.

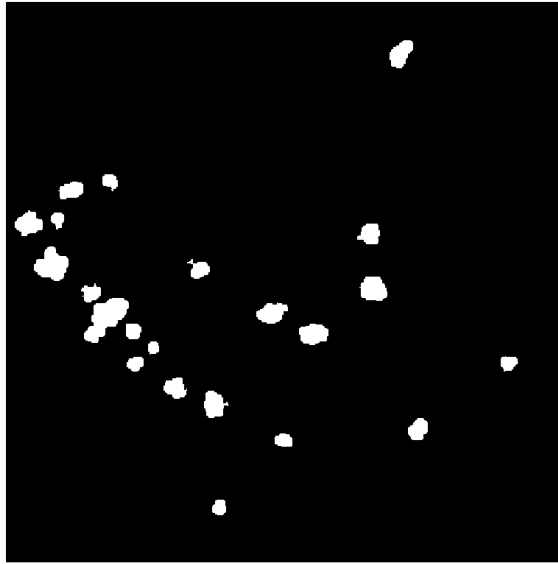


Figura 4.23: Máscara de nuvens para a tabela *ndvi_diferenca*.

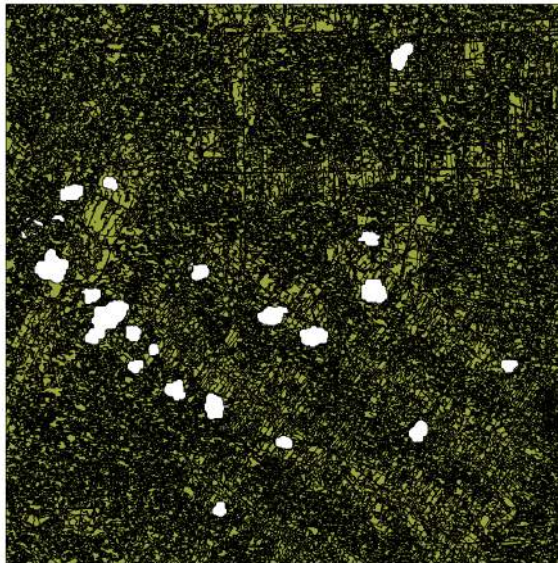


Figura 4.24: *Shapefile* recortado nas regiões de nuvens.

No *Script* 4.9 é apresentada uma otimização para a criação da tabela *shape_diferenca* ao se definir uma condição envolvendo o atributo *geom*, o que permite o uso do índice espacial. Isso foi possível quando as funções *ST_Difference* e *COALESCE* foram utilizadas em conjunto com a junção *LEFT OUTER JOIN*,

onde a função `COALESCE` era responsável em definir a geometria da tabela *shape_diferenca* como igual ao da tabela *shape* quando o resultado do `LEFT OUTER JOIN` fosse um valor nulo, ou seja, quando não havia interseção entre os polígonos e as nuvens. Em contrapartida, quando houvesse alguma interseção, a função *ST_Difference* se encarregaria de calcular a diferença entre as geometrias das duas tabelas. Para garantir a geração de valores únicos para os identificadores das feições da tabela *shape_diferenca*, a sequência temporária *geometry_intersection_seq* foi incrementada a cada resultado recuperado pela consulta.

```
CREATE TEMPORARY SEQUENCE geometry_intersection_seq
START WITH 1;

CREATE TABLE shape_diferenca AS
SELECT nextval('geometry_intersection') AS gid,
       COALESCE(ST_Difference(shape.geom, nuvem.geom), shape.geom)
       AS geom
FROM shape
LEFT OUTER JOIN nuvem
ON ST_Intersects(shape.geom, nuvem.geom);
```

Script 4.9: Consulta para a criação da tabela *shape_diferenca*.

Finalmente, no Script 4.10, o atributo *media_diferenca* foi adicionado à tabela *shape_diferenca* a fim de armazenar os valores das médias mencionados anteriormente. Como o *raster* e o *shapefile* possuem exatamente o mesmo tamanho, o *pixeltype* da banda da tabela *ndvi_diferenca* foi considerado como `NULL`, o que permitiu a otimização da consulta, visto que a verificação de *pixels* com valores *NODATA* não é feita.

```

ALTER TABLE shape_diferenca
ADD COLUMN media_diferenca DOUBLE PRECISION;

UPDATE shape_diferenca
SET media_diferenca = mean_values
FROM (
    SELECT gid, avg(pixel_values) AS mean_values FROM
    (
        SELECT gid, fn_valores_pixels(ST_Clip(R.rast,V.geom)) AS
        pixel_values
        FROM shape_diferenca as V
        INNER JOIN ndvi_diferenca as R
        ON ST_Intersects(R.rast, V.geom)
    ) AS clipped_raster
    WHERE pixel_values > -2 AND pixel_values < 2
    GROUP BY gid
    ) AS mean_values
WHERE mean_values.gid = shape_diferenca.gid;

```

Script 4.10: Consulta utilizada para atribuir as médias dos valores de *pixels* aos polígonos.

Nesta consulta, o *SELECT* mais interno é responsável por selecionar os identificadores das feições e todos os valores de *pixels* dos *rasters* recortados que fazem interseção com os polígonos. A função *fn_valores_pixels*, presente no Script 4.11, foi implementada para extrair todos os valores de *pixels* da matriz do *raster* passado por parâmetro. Em seguida, o *SELECT* mais externo calcula a média de cada polígono ao agrupar os valores de *pixels* entre -2 e 2. Após a execução da consulta, a tabela *shape_diferenca* foi exibida no software QGIS, como observado na Figura 4.25. Nesse SIG, uma faixa de valores foi definida para o atributo *media_diferenca*, no qual quanto maiores fossem os valores de *pixels*, mais roxo o polígono seria, indicando áreas que se regeneraram no ano de 2012 em relação ao ano de 2011, já as áreas mais claras indicam as áreas desmatadas.

```

CREATE OR REPLACE FUNCTION fn_valores_pixels(rast RASTER, band INTEGER
      DEFAULT 1)
RETURNS TABLE (
  pixel_value DOUBLE PRECISION
) AS
$BODY$

DECLARE
width INTEGER;
height INTEGER;

BEGIN
  SELECT ST_Width($1) INTO width LIMIT 1;
  SELECT ST_Height($1) INTO height LIMIT 1;
  RETURN QUERY
  SELECT ST_Value($1, $2, x, y) AS pixel_value
  FROM generate_series(1, width) AS x
       CROSS JOIN
       generate_series(1, height) As y;
END;
$BODY$
LANGUAGE plpgsql IMMUTABLE;

```

Script 4.11: Função para extrair todos os valores de *pixels* de um dado *raster*.

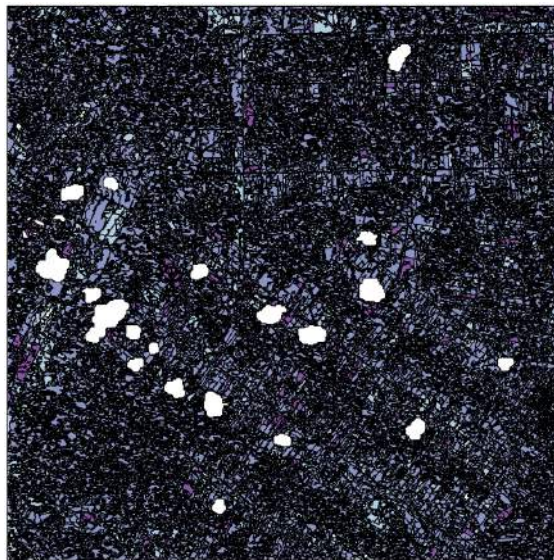


Figura 4.25: *Shapefile* categorizado sobre o atributo *media_diferenca*.

4.4 Participação no projeto SisBolsaVerde

4.4.1 Fundamentação Teórica

4.4.1.1 Scrum

Pressman (2011) conceitua *Scrum* como um método de desenvolvimento ágil que orienta atividades de desenvolvimento de software em um processo que incorpora requisitos, análise, projeto, evolução e entrega. O *Scrum* visa o gerenciamento de projetos de requisitos mutáveis e de baixo prazo de entrega através de um conjunto padrões de processos, visualizados na Figura 4.26, que priorizam o desenvolvimento das funcionalidades mais importantes, a divisão de tarefas entre as equipes e a comunicação frequente do cliente.

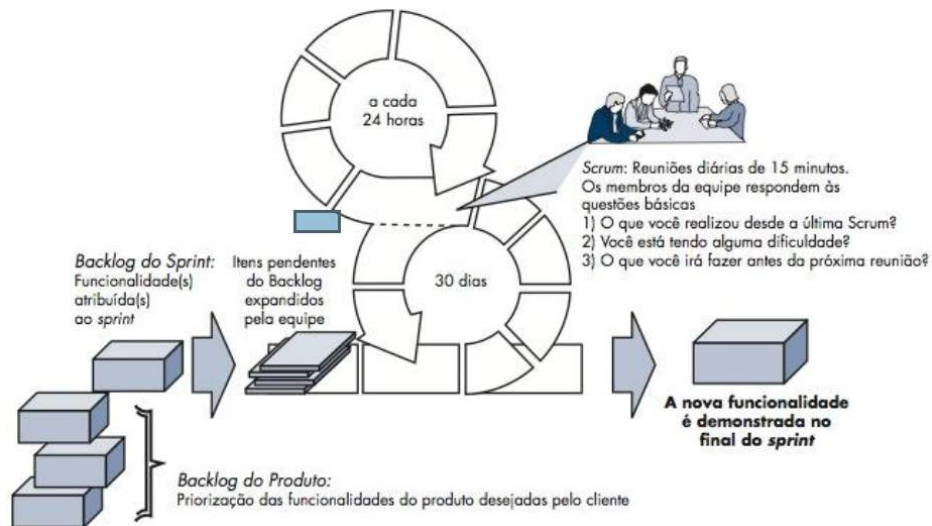


Figura 4.26: Padrões de processo *Scrum*.

Fonte: PRESSMAN, 2011

Nessa metodologia, toda atividade é realizada dentro de um padrão de processo chamado *Sprint*, no qual é definido um prazo de entrega, geralmente de 30 dias, para atingir os requisitos estabelecidos em uma lista conhecida como backlog, que registra as funcionalidades do sistema de principal interesse do cliente. No fi-

nal dessa *Sprint*, uma *demo* é apresentada para o cliente de modo que o *feedback* direto desse identifique novas necessidades para o projeto.

Reuniões diárias conhecidas como *Daily Scrum* são realizadas rapidamente pela equipe (*Scrum*) para discutir o desenvolvimento do projeto, as dificuldades enfrentadas e as novas etapas de desenvolvimento. O líder da equipe, o *Scrum Master*, conduz a reunião e analisa os pontos discutidos pela equipe para prever problemas potenciais o mais rápido possível.

4.4.1.2 Modelagem de Banco de Dados

Segundo Elmasri e Navathe (2010), um modelo de dados fornece os meios necessários para se obter uma abstração da estrutura de um banco de dados, uma vez que esta permite um melhor entendimento sobre os tipos, relacionamentos e restrições sobre os dados em um nível de detalhamento apropriado para diferentes tipos de usuários. Diversos tipos de modelos foram criados desde o surgimento dos primeiros Sistemas Gerenciadores de Banco de Dados (SGBDs). Esses modelos podem ser classificados em modelos de dados conceituais, modelos de dados lógicos e modelos de dados físicos.

Dentre os citados acima, os modelos de dados conceituais são apropriados para a compreensão da semântica dos dados sem a utilização de aspectos de implementação para sua construção. Diversos modelos são utilizados atualmente, tais como o Diagrama de Entidade-Relacionamento (DER), modelo IFO (*Is-a relationships, Functional relationships, complex Objects*) e modelos orientados a objetos, sendo o último comumente gerado a partir das padronizações da *Unified Modeling Language* (UML).

Em contrapartida, os modelos lógicos, como o modelo relacional, visam mostrar detalhes da representação das entidades de um Banco de Dados. Conforme Singh (2011), o modelo relacional representa uma estrutura lógica de um Banco de Dados originada pelos conceitos matemáticos de relação e teoria de conjun-

tos. Uma relação pode ser vista como uma tabela bidimensional de um conjunto de entidades relacionadas, onde as linhas representam uma entidade e as colunas representam características de uma entidade em particular. Nesse modelo, toda entidade é identificada por uma chave, o que permite o relacionamento entre diferentes entidades.

4.4.2 Atividades Executadas

No projeto SisBolsaVerde, assim como na maioria dos projetos do LEMAF, foi empregada a metodologia ágil *Scrum* para gestão e planejamento do projeto. O *Product Owner* (P.O.) mantém contato direto com o cliente através de breves reuniões, nas quais são definidas os requisitos principais do próximo *Sprint*, o qual deve ser entregue em um prazo combinado entre o P.O e o cliente.

Nessas reuniões, o P.O. leva consigo protótipos das telas para demonstrar e discutir as funcionalidades que devem ser implementadas e testadas para a entrega do *Sprint*, a fim de redefinir ou identificar novos requisitos do sistema. A prototipação de telas também pode ocorrer durante a reunião e possibilita um *feedback* direto do cliente em relação à usabilidade do software, além de fornecer uma melhor visualização das funcionalidades do sistema para os *stakeholders*.

O *Scrum Master* é responsável por estimar, em conjunto com as equipes, o tempo que será gasto para o desenvolvimento das funcionalidades definidas em cada *Sprint*, além de mediar as equipes para que essas utilizem as práticas de *Scrum*. Nesse projeto, a definição do prazo de entrega das *Sprints* é de geralmente 10 dias úteis, porém esse prazo é flexível e pode ser alterado de acordo com as necessidades do cliente, por exemplo, em alguns casos, é interessante para o cliente estender o prazo de entrega para verificar funcionalidades do sistema que não seriam possíveis de serem implementadas em somente 10 dias úteis.

Na fase inicial do projeto, uma reunião foi realizada entre alguns membros da equipe de desenvolvimento e banco de dados para definir a modelagem do

Banco de Dados do SisBolsaVerde. No presente relatório, para manter o sigilo do projeto, são apresentados somente os procedimentos empregados pela empresa em relação à modelagem, e não a modelagem em si.

Apesar de manipular dados espaciais, somente o modelo relacional foi adotado para a modelagem do Banco de Dados do SisBolsaVerde, uma vez que o projeto possui poucas feições geométricas (em sua maioria polígonos) e somente duas tabelas georreferenciadas, não foi necessário utilizar modelos robustos para dados geográficos, como o modelo OMT-G. No entanto, um modelo mais robusto pode vir a ser adotado caso o projeto alcance um maior nível de complexidade na representação de dados geográficos.

Nessa reunião, a modelagem se baseou nas informações adquiridas em outras reuniões, nos protótipos de interface e no dicionário de dados do Banco de Dados utilizado pelo Ministério do Meio Ambiente (MMA) anteriormente. Inicialmente, foram identificadas as entidades dos esquemas *portal_seguranca*, referente às entidades que constituem o portal de segurança da aplicação, e *bolsa_verde*, referente ao processo de notificação do Programa Bolsa Verde.

Como pode ser visualizado na Figura 4.27, no processo de modelagem, as tabelas eram projetadas a medida em que ambas as equipes identificavam as entidades de cada esquema. Com relação aos atributos, os campos obrigatórios (*NOT NULL*) eram marcados por um * e um prefixo foi adicionado ao nome do atributo de acordo com o seu tipo de dado. Os relacionamentos entre as entidades eram definidos conforme o incremento de uma nova entidade, no qual a cardinalidade entre elas não era especificada no desenho, mas nos atributos com chave estrangeira (para relacionamentos 1:1 e 1:N) ou em tabelas de relacionamento (para relacionamentos M:N).

Após essa modelagem inicial com a participação das equipes, o modelo de dados foi construído através da ferramenta SyBase Power Designer, a qual permite a geração automática do *script* de criação do Banco de Dados através do modelo,

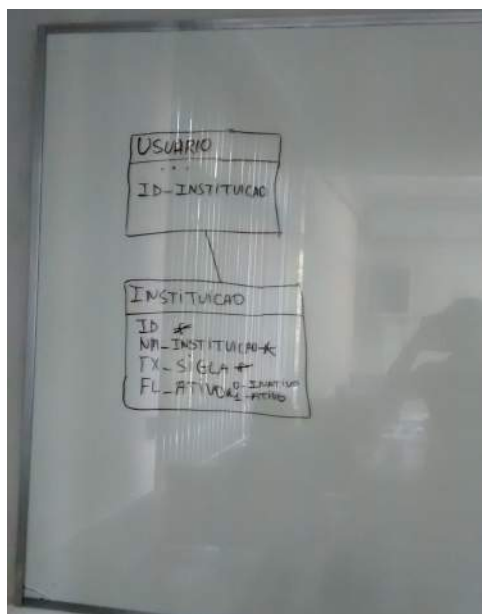


Figura 4.27: Modelagem parcial do esquema portal_seguranca do Banco de Dados SisBolsaVerde.

no qual é preciso apenas especificar qual o SGBD de destino. Na geração do modelo e do script, foram revisados: (i) os relacionamentos entre as entidades, (ii) os tipos de dados do SGBD PostgreSQL que seriam utilizados para representar cada atributo e (iii) se os prefixos utilizados nos nomes de cada campo seguiam a padronização utilizada pela equipe de Banco de Dados.

Depois da geração automática, foi solicitado a criação do dicionário de dados à equipe de Banco de Dados. O dicionário de dados visa especificar informações sobre os dados e facilita o desenvolvimento da aplicação, visto que esse permite que os desenvolvedores não participantes das reuniões entendam como o Banco de Dados está estruturado. Além disso, quando o projeto é entregue, o dicionário de dados, em conjunto com diagrama, permite que o cliente entenda as informações dos esquemas, tabelas e atributos sem que ele tenha participado do desenvolvimento do projeto. O Apêndice B contempla um documento de padrões elaborado pelo estagiário, em conjunto com a equipe de Banco de Dados do

LEMAF, para nomenclatura de tabelas, atributos, chaves primárias, chaves estrangeiras, comentários do dicionário de dados, entre outros.

Em seguida, o Banco de Dados foi parcialmente populado com dados fornecidos pelo MMA para que as equipes de desenvolvimento e teste pudessem desempenhar suas funções. À medida em que novas funcionalidades eram adicionadas ao sistema, ambas as equipes solicitavam diferentes operações, como a inserção de dados, criação de triggers, adição de novos esquemas, tabelas e colunas, alteração de tamanho de campos, etc.

Desde a criação do Banco de dados, todos os *scripts* criados foram numerados em ordem cronológica para manter o controle da evolução do Banco de Dados. A Figura 4.28 apresenta uma planilha de controle de atividades da equipe de Banco de Dados em relação ao projeto SisBolsaVerde. Dentre as colunas da planilha, a Coluna B se refere à execução do *script* no servidor local do LEMAF, Postgres2, para a simulação da aplicação. As colunas G, H, I possuíam a finalidade de manter a coerência, respectivamente, dos comentários, modelo e dicionário de dados a cada alteração realizada, de modo que a equipe acompanhasse pela planilha quais atividades foram executadas por outros membros e quais ainda deveriam ser finalizadas.

1	A	B	C	D	E	F	G	H	I
1	SCRIPT	POSTGRES2	COMENTÁRIO	DATA	SOLICITADO POR	EXECUTADO POR	COMENTARIO ESTRUTURA	MODELO	DIC. DADOS
2	1	X	Criação dos esquemas e tabelas do sisverde	9/19/2014	Equipe	Renan/Daniel	X	X	X
3	2	X	Correções nos esquemas, tabelas e sequências	9/19/2014	Equipe	Christiane	X	X	X
4	3	X	Carga de dados na tabela bolsa_verde.instrumento_gestao	9/19/2014	Equipe	Daniel	X	X	X
5	4	X	Comentários do banco SisVerde (portal_seguranc e Bolsa_Verde)	9/24/2014	Equipe	Renan/Daniel	X	X	X
6	5	X	Criação de triggers na tabela bolsa_verde.area	9/30/2014	Equipe	Renan/Daniel	X	X	X
7	6	X	Criação do esquema servicos	10/3/2014	Elias	Daniel	X	X	X
8	7	X	Criação das tabelas planilha_modelo, planilha_coluna e pl_planilha_modelo_coluna nos bancos sisverde e sisverde_cliente	10/3/2014	Elias	Renan	X	X	X
9	8	X	Criação da fk_s_id_orgao na tabela solicitacao no banco sisverde e sisverde_cliente	10/6/2014	Elias	Luciene	X	X	X
10	9	X	Criação da função utmzone e atualização da função f_gest_geometria_ijk para também calcular a área da geometria na tabela bolsa_verde.area	10/6/2014	Elias	Daniel	X	X	X
11	10	X	Comentários do banco SisVerde e Sisverde_Cliente (esquemas bolsa_verde e servicos)	10/7/2014	Equipe	Renan/Daniel	X	X	X
12	11	X	Alteração dos campos tx_nis da tabela bolsa_verde.responsavel_familia para NULL e col. familia_bolsa_verde da tabela bolsa_verde.familia para VARCHAR(40)	10/8/2014	Elias	Daniel	X	X	X
13	12	X	Alteração do campo tx_orgao_emissor da tabela bolsa_verde.responsavel_familia para VARCHAR(150)	10/10/2014	Elias	Daniel	X	X	X
14	13	X	Adição da coluna fl_ativo na tabela bolsa_verde.familia	10/14/2014	Elias	Christiane/Daniel	X	X	X
15	14	X	Remoção de restrição not null do campo id_municipio da tabela bolsa_verde.endereco	10/17/2014	Elias	Christiane	X	X	X

Figura 4.28: Planilha de atividades do Banco de Dados SisBolsaVerde.

Dentre os 14 *scripts* criados até o final do estágio, dois são referentes à criação de *triggers* para a automatização da inserção ou atualização de dados relacionados à coluna geométrica *the_geom* da tabela *area*, a qual armazena informações sobre as áreas onde residem as famílias beneficiárias do Programa Bolsa Verde. Essa coluna é atualizada por meio de dados na representação *Well Known Text* (WKT), uma padronização definida pela SFSQL para a representação textual de feições geométricas.

A função PL/pgSQL presente no *Script 4.12* é disparada por *triggers* após inserir ou atualizar a coluna não nula *tx_wkt* da tabela *area*, a qual é utilizada como entrada da função *ST_GeomFromText* para atualizar o campo geométrico nulo *the_geom* com um valor do tipo *geometry* somente nas linhas afetadas pela *trigger*. Após ser traduzida do WKT, a geometria é georreferenciada através da função *ST_SetSrid* pelo SRID padrão do Brasil (4674).

```
CREATE OR REPLACE FUNCTION fn_popula_dados_pelo_wkt ()
  RETURNS TRIGGER AS
$BODY$
BEGIN
  IF TG_OP = 'INSERT' OR TG_OP = 'UPDATE' THEN
    UPDATE bolsa_verde.area
      SET the_geom = ST_SetSrid (
          ST_GeomFromText (tx_wkt), 4674
        )
      WHERE id_area = new.id_area;

    UPDATE bolsa_verde.area
      SET nu_area_ha = ST_Area (
          ST_Transform (
            the_geom,
            fn_utmzone (ST_Centroid (the_geom))
          )
        ) / 10000
      WHERE id_area = new.id_area;
    END IF;
  RETURN new;
END;
$BODY$
LANGUAGE plpgsql VOLATILE;
```

Script 4.12: Função PL/pgSQL para converter um WKT em uma geometria e calcular a área dessa geometria em hectares.

Em seguida, a área em hectares da geometria é automaticamente calculada ao se dividir por 10.000 o valor retornado pela função *ST_Area*, a qual utiliza como parâmetro a geometria transformada pela função *ST_Transform*. Essa transformação ocorre antes do cálculo da área para garantir o uso da projeção correta, determinada pela localização espacial do objeto no globo. Para determinar essa projeção, foi aplicada a função *fn_utmzone*, presente no *Script 4.13*, sobre o centróide da geometria.

```
CREATE OR REPLACE FUNCTION fn_utmzone(geometry)
  RETURNS INTEGER AS
$BODY$
DECLARE
  geom GEOMETRY;
  geom_zone INTEGER;
  geom_pref INTEGER;
BEGIN
  geom := ST_Transform($1,4674);
  IF (ST_Y(geom))>0 THEN
    geom_pref := 32600;
  ELSE
    geom_pref := 32700;
  END IF;
  geom_zone := floor((ST_X(geom)+180)/6)+1;
  RETURN geom_zone + geom_pref;
END;
$BODY$
LANGUAGE plpgsql IMMUTABLE
```

Script 4.13: Função PL/pgSQL para determinar a projeção de um objeto geográfico no globo.

A criação de *triggers* não se limitou somente ao preenchimento de campos da tabela *area*. Uma *trigger* teve de ser criada para preencher tabelas de relacionamento entre tabelas com colunas georreferenciadas que possuíam relacionamentos topológicos entre si, por exemplo, a tabela *rel_area_municipio* deveria ser preenchida quando a geometria de um município fizesse interseção com a geometria da área de um beneficiário.

A função contida no *Script 4.14* é disparada por uma *trigger* depois que o atributo *the_geom* é inserido ou atualizado na tabela *area*. Nessa função, primeiramente são deletadas as entradas da tabela *rel_area_municipio* onde o novo

identificador é igual a um identificador já existente, ou seja, essa deleção somente ocorre quando há atualizações no campo *the_geom* da tabela *area*. Em seguida, é inserido na tabela *rel_area_municipio* o resultado da seleção dos identificadores dos municípios e áreas que se interseccionam.

```
CREATE OR REPLACE FUNCTION fn_insere_rel_area_municipio()
  RETURNS trigger AS
$BODY$
BEGIN
  DELETE FROM rel_area_municipio
  WHERE id_area = new.id_area;

  INSERT INTO rel_area_municipio(id_area, id_municipio)
  SELECT id_area, id_municipio
  FROM area AS a
  INNER JOIN municipio AS m
  ON ST_Intersects(a.the_geom, m.the_geom)
  WHERE id_area = new.id_area;

  RETURN new;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
```

Script 4.14: Função PL/pgSQL para preencher automaticamente a tabela *rel_area_municipio*.

5 CONCLUSÃO

5.1 Considerações Finais

O estágio realizado no LEMAF como Trabalho de Conclusão de Curso para obtenção do título de bacharel em Ciência da Computação proporcionou ao estagiário experiência profissional e uma melhor perspectiva do mercado de trabalho na área de Tecnologia da Informação.

Uma das características mais importantes do estágio foi a aplicação práticas dos conhecimentos adquiridos durante a vida acadêmica em projetos de grande porte, nos quais algumas técnicas tendem a se diferenciar em relação à teoria. Apesar de o setor de TI do LEMAF atribuir tarefas de diferentes áreas de conhecimento da computação para diferentes equipes, muitas vezes as habilidades de um bacharel são necessárias para a comunicação entre as equipes.

Uma oportunidade interessante fornecida pelo LEMAF é a possibilidade de conhecer aplicações computacionais de diferentes domínios, como os SIGs, os quais não são apresentados em matérias de cursos de graduação. A multidisciplinaridade fornece ao profissional novos caminhos em que seus conhecimentos podem ser aplicados.

A respeito das atividades exercidas, as apostilas e exercícios estudados no início do estágio contribuíram para a posterior execução das atividades da empresa. Por exemplo, a base de dados espacial de Nova Iorque contribui para a aprendizagem das funções básicas do PostGIS.

Apesar de ter sido comprovado que geometrias possuem um melhor desempenho e precisão em relação à dados matriciais para o cálculo de áreas, outras alternativas para dados *rasters* foram investigadas, como a análise temporal pela diferença de NDVIs, a qual permite monitorar as regiões que mais sofreram alterações na vegetação ao longo do tempo. Essas análises realizadas diretamente no Banco de Dados tornam-se interessantes para a automatização de operações

de processamento em imagens matriciais ou vetoriais através da linguagem de consulta SQL, porém deve-se ressaltar que o tamanho de um *raster* dentro de um Banco de Dados é superior do que o tamanho de um no sistema de arquivos devido à criação de *tiles* e índices espaciais.

Ademais, é importante mencionar que o estágio também permitiu o acompanhamento dos processos de engenharia de software empregados pela empresa e um pouco de experiência em administração de Banco de Dados através do projeto SisBolsaVerde.

Dentre as contribuições do estagiário ao LEMAF, além do cumprimento de atividades solicitadas, a redação desse relatório permitirá auxiliar estagiários ingressantes no estudo da área de Banco de Dados Geográficos. O manual de padronização de nomes de Banco de Dados, elaborado pelo estagiário, poderá ser utilizado como um guia aos funcionários ingressantes, evitando retrabalho para a equipe de Banco de Dados, nos casos em que o padrão não é seguido. Apesar do LEMAF ainda não possuir projetos que necessitem de análises estatísticas de *pixels*, os profissionais da empresa já estão cientes das funcionalidades da ferramenta PostGIS para dados *rasters* e as utilizarão em caso de novos projetos relacionados.

5.2 Trabalhos Futuros

Com os conhecimentos obtidos durante o desenvolvimento deste relatório, é possível considerar uma série de questões - relacionadas principalmente ao processamento de dados *raster* por meio de Banco de Dados – para uma futura continuação das atividades realizadas, tais como:

- Comparativo de desempenho no tratamento de dados espaciais nas representações vetoriais e matriciais entre a ferramenta PostGIS e outras extensões espaciais para SGBDs Relacionais, como o Oracle Spatial;

- Análise de *rasters* através de SGBDs nativos para o tratamento de dados nessa estrutura, como Rasdaman e SciDB, de modo a explorar uma abordagem diferente da utilização de Banco de Dados Relacionais;
- Aplicação de diferentes índices para análise de vegetação, assim como análise temporal de outros fenômenos naturais, como o clima;
- Estudo da implantação de novas padronizações e processos de engenharia de software nos projetos do LEMAF.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARAUJO, T. P. d.; MELLO, F. M. d. Processamento de imagens digitais-razão entre bandas. *Geociências (São Paulo)*, Geociências (São Paulo), v. 29, n. 1, p. 121–131, 2010.
- BOUNDLESS. *Boundless WorkShop - Introduction to PostGIS*. [S.l.], 2011. Disponível em: <<http://workshops.boundlessgeo.com/postgis-intro/introduction.html>>.
- BRINKHOFF, T.; KRIEGEL, H. peter; SEEGER, B. Efficient processing of spatial joins using r-tress. *SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data.*, p. 237–246, 1993.
- CÂMARA, G. *Modelos Conceituais de Dados para Sistemas de Informações Geográficas*. [S.l.: s.n.], 2005. 1-40 p.
- CASANOVA, M. A. *Processamento de consultas e gerência de transações*. [S.l.: s.n.], 2005. 219-262 p.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. 6. ed. [S.l.]: Pearson Education, 2010.
- FILHO, J. L. *Modelos Conceituais de Dados para Sistemas de Informações Geográficas*. [S.l.: s.n.], 1997.
- FRANCISCO, C. N. Sistemas de informação geográfica e geoprocessamento. 2004. Disponível em: <<http://www.professores.uff.br/cristiane/Estudodirigido/SIG.htm>>.
- GUTING, R. H. An introduction to spatial database systems. *The VLDB Journal — The International Journal on Very Large Data Bases - Spatial Database Systems*, v. 3, n. 4, p. 357–399, 1994. Disponível em: <<http://dl.acm.org/citation.cfm?id=615204.615206>>.
- HELLERSTEIN, J. M. *GiST research project website*. 1999. Disponível em: <<http://gist.cs.berkeley.edu/gist1.html>>.
- JR, J. R.; HAAS, R.; SCHELL, J.; DEERING, D. Monitoring vegetation systems in the great plains with erts. *NASA special publication*, v. 351, p. 309, 1974.
- LASMAR, E. B. de C. Administração de sistemas de informação no laboratório de projetos e estudos em manejo florestal. *UFLA*, 2012.
- LUZ, A. P.; FRANCÉS, A.; FERNANDES, J.; DILL, A. C. Aplicação de um SIG na seleção de locais para implantação de aterros sanitários em áreas vulneráveis à contaminação de aquíferos. *Seminário sobre a*

Hidroinformática em Portugal, Lisboa, LNEC, p. 1–14, nov. 2001. Disponível em: <<http://repositorio.lneg.pt/bitstream/10400.9/495/1/33624.pdf>>.

MINISTÉRIO DO MEIO AMBIENTE. *Site do Ministério do Meio Ambiente - Bolsa Verde*. [S.l.], 2011. Disponível em: <<http://www.mma.gov.br/desenvolvimento-rural/bolsa-verde>>.

ORACLE. *Oracle Spatial Developer's Guide 11g Release 1 (11.1)*. [S.l.], 2009. Disponível em: <http://docs.oracle.com/cd/B28359_01/appdev.111/b28400.pdf>.

POSTGIS. *PostGIS Ticket 2807*. 2014. Disponível em: <<http://trac.osgeo.org/postgis/ticket/2807>>.

POSTGRESQL. *PostGIS features*. [S.l.], 2012. Disponível em: <<http://postgis.net/features>>.

PRESSMAN, R. S. *Engenharia de software*. [S.l.]: McGraw Hill Brasil, 2011.

RIGAUX, P.; SCHOLL, M. O.; VOISARD, A. *Spatial Databases: With Application to GIS*. 1. ed. [S.l.: s.n.], 2001. 3-25 p.

SIDNEY, C. F. Aplicação de mineração de dados no banco de dados do zoneamento ecológico econômico de minas gerais. *UFLA*, 2010.

SINGH, S. K. *Database systems: Concepts, design and applications*. [S.l.]: Pearson Education India, 2011.

VELICANU, A.; OLARU, S. Optimizing spatial databases. *Database Systems Journal*, v. 1, p. 61–75, 2010.

A TEMPOS DE EXECUÇÃO DAS CONSULTAS PARA O CÁLCULO DE ÁREA DE SOBREPOSIÇÃO

As Tabelas A.1 e A.2 reportam os tempos de execução das consultas para o cálculo de área de sobreposição.

Tabela A.1: Tempos de execução para o Shapefile #1

Número da Amostra	Consulta #1 (s)	Consulta #2 (s)	Consulta #3 (s)	Consulta #4 (s)
1	0.296	0.848	1.719	1.781
2	0.315	0.815	1.530	1.803
3	0.347	0.807	1.677	1.615
4	0.272	0.860	1.859	1.693
5	0.418	0.811	1.796	1.712
6	0.409	0.798	1.578	1.395
7	0.469	0.811	1.503	1.684
8	0.341	0.795	1.781	1.589
9	0.294	0.839	1.409	1.416
Média (s)	0.351	0.820	1.650	1.632
Área (m²)	478102374.714964	478102374.714964	47811150	47811150

Tabela A.2: Tempos de execução para o Shapefile #2

Número da Amostra	Consulta #1 (s)	Consulta #2 (s)	Consulta #3 (s)	Consulta #4 (s)
1	15.754	102.159	18.639	16.128
2	13.364	101.225	20.627	18.311
3	10.792	93.807	17.854	16.641
4	9.604	98.676	20.380	15.230
5	11.935	102.838	17.754	18.177
6	13.016	95.969	18.383	16.627
7	15.576	98.589	16.501	19.441
8	12.636	100.395	18.755	16.391
9	13.575	97.123	19.781	18.336
Média	12.917	98.976	18.742	17.254
Área (m²)	197713250	197713250	201548925	201548925

B PADRÕES PARA BANCO DE DADOS DO LEMAF

Padrões Gerais:

- Procurar utilizar nomes que identifiquem o objeto que se deseja nomear.
- Utilizar *underscores* (_) como separador de nomes compostos.
- Evitar o uso de abreviações e utilizá-las somente nos casos onde o nome seja maior que a quantidade de caracteres permitida pelo SGBD.
- Nomear as entidades (tabelas) somente no singular.
- Nomear os atributos (colunas) com os prefixos referentes ao tipo de dado do atributo. Exemplo: atributo *nm_{nome_atributo}* utiliza o prefixo *nm* para indicar o tipo de dado VARCHAR do atributo.
- Nomear as tabelas que representam relacionamentos como *rel_{nome_primeira_tabela}_{nome_segunda_tabela}*.

Padrões para Objetos de Banco de Dados:

Objeto de Banco de Dados	Nome Segundo a Padronização
View	vw_{nome_view}
Sequence	{nome_tabela}_{nome_atributo}_seq
Index	{nome_tabela}_{nome_atributo}_idx
Function	fn_{nome_funcao}
Procedure	pr_{nome_procedimento}
Trigger	tr_{nome_trigger}
Primary Key	pk_{nome_chave_primaria}
Foreign Key	fk_{primeiras_letras_da_tabela}_ {nome_atributo_pk_tabela_estrangeira}

Padrões para Atributos:

Prefixo	Tipo de Dado	Tamanho	Exemplo de nome
id	SERIAL		id_municipio
	INTEGER		id_estado
nm	VARCHAR	100	nm_municipio
cod	VARCHAR	Variável	cod_uf
tx	TEXT		tx_url
	VARCHAR	Variável	tx_sigla
dt	DATE		dt_nascimento
	TIMESTAMP WITHOUT TIME ZONE		dt_cadastro
fl	BOOLEAN		fl_ativo
nu	DOUBLE PRECISION		nu_latidade
	NUMERIC	13,4	nu_area
tp	INTEGER		tp_arquivo

Padrões para comentários no Dicionário de Dados:

- Esquemas e entidades são referenciadas por letras maiúsculas para distinguir o nome do objeto a ser representado de sua entidade no Banco de Dados.
- Comentários de esquemas são precedidos por: “*O esquema {nome_esquema} contém todas as tabelas referentes à(ao)...*”
- Comentários de entidades são precedidos de: “*Entidade responsável por armazenar...*”
- Tabelas de relacionamento possuem o seguinte formato de comentário: “*Entidade que representa o relacionamento entre as entidades*”

{nome_primeira_entidade} e {nome_segunda_entidade}.”, sendo possível adicionar comentários complementares ao final.

- Atributos de chave primária possuem o seguinte formato de comentário: *“Identificador único da entidade {nome da entidade}.”*
- Atributos de chave estrangeira possuem o seguinte formato de comentário: *“Identificador único da entidade {nome_entidade} que realiza o relacionamento entre as entidades {nome_entidade} e {nome_entidade_relacionada}.”*, sendo possível adicionar comentários complementares ao final.
- Comentários de atributos booleanos são precedidos de: *“Flag que identifica se...”*.
- Atributos com restrição de valores devem ter todos os seus valores exemplificados após a descrição do atributo. Para tal, os possíveis valores devem ser descritos entre parênteses e separados por ponto e vírgula. Exemplo: *Tipo do arquivo (1- Planilha de Importação; 2- Planilha de Exportação).*
- Atributos nominais com valores específicos à entidade devem ser exemplificados ao final do comentário, a fim de facilitar o entendimento do leitor. Por exemplo, para o atributo *cod_estado*, um possível comentário seria: *“Código do estado. Exemplo: SP, RJ, MG.”*.
- Atributos geométricos devem ser precedidos pelo comentário *“Coluna georreferenciada”* e devem especificar, entre parênteses, os possíveis tipos geométricos que a coluna pode assumir.