

**UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO AGRÍCOLA DE FREDERICO WESTPHALEN
PÓS-GRADUAÇÃO EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO**

**MAPGEOCITY: UMA PROPOSTA DE SISTEMA PARA
AUXILIAR NA GESTÃO PÚBLICA MUNICIPAL**

MONOGRAFIA DE ESPECIALIZAÇÃO

NEIMIR RUGGINI

Frederico Westphalen, RS, Brasil

2013

MAPGEOCITY: UMA PROPOSTA DE SISTEMA PARA AUXILIAR NA GESTÃO PÚBLICA MUNICIPAL

por

Neimir Ruggini

Monografia apresentada ao Curso de Especialização em Gestão de Tecnologia da Informação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Especialista em Gestão de Tecnologia da Informação.**

Orientador: Prof. Dr. Joel da Silva

Frederico Westphalen, RS, Brasil

2013

**UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO AGRÍCOLA DE FREDERICO WESTPHALEN
PÓS-GRADUAÇÃO EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO**

A Comissão Examinadora, abaixo assinada,
aprova a Monografia de Especialização

**MAPGEOCITY: UMA PROPOSTA DE SISTEMA PARA AUXILIAR NA
GESTÃO PÚBLICA MUNICIPAL**

elaborada por
Neimir Ruggini

como requisito parcial para obtenção do grau de
Especialista em Gestão de Tecnologia da Informação

COMISSÃO EXAMINADORA:

Joel da Silva, Dr.
(Presidente/Orientador)

Edinara Filipiak de Cristo, Msc. (UFSM)

Catiane Priscila Barbosa Arenhardt Mazzutti, Msc. (UFSM)

Frederico Westphalen, 14 de dezembro de 2013.

DEDICATÓRIA

Dedico este trabalho à todas as pessoas que estiveram ao meu lado, me apoiaram e me incentivaram, direta ou indiretamente, para a continuidade e realização desta conquista.

AGRADECIMENTO

À todos que me apoiaram e contribuíram para a realização deste sonho, obrigado!

EPIGRAFE

"Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar aonde a maioria não chega, faça aquilo que a maioria não faz."

Bill Gates

"Para se ter sucesso, é necessário amar de verdade o que se faz. Caso contrário, levando em conta apenas o lado racional, você simplesmente desiste. É o que acontece com a maioria das pessoas."

Steve Jobs

RESUMO

Monografia de Especialização
Curso de Especialização em Gestão de Tecnologia da Informação
Universidade Federal de Santa Maria

MAPGEOCITY: UMA PROPOSTA DE SISTEMA PARA AUXILIAR NA GESTÃO PÚBLICA MUNICIPAL

Autor: Neimir Ruggini

Orientador: Joel da Silva

Data e Local da Defesa: Frederico Westphalen, 14 de dezembro de 2013.

Atualmente, devido à crescente evolução dos sistemas de informação geográfica, nota-se uma necessidade cada vez maior desta tecnologia fazer parte do dia a dia das pessoas, auxiliando e contribuindo na precisão e na tomada de decisão, agregando maior valor às informações. Dessa forma, pretende-se com este estudo, descrever uma proposta de implementação de uma extensão da solução Geoliberty, focando na gestão dos dados geográficos, afim de contribuir para o processo de tomada de decisão das informações geradas.

Para a realização desta pesquisa adotou-se somente tecnologias *open-source*, inclusive a elaboração deste material, prezando e contribuindo com a comunidade brasileira de software livre. O escopo do projeto foi determinado a partir da necessidade de aplicação das tecnologias SIG na gestão pública municipal. Dessa forma, este estudo visa a criação de um *framework* SIG, a partir da estrutura base dos dados geográficos definidos pelo GeoLiberty, para auxiliar na gestão das informações do cadastro de imóveis do município de Frederico Westphalen. A partir da aplicação de técnicas de georreferenciamento, torna-se possível definir a sua espacialização sobre os mapas, sobrepondo camadas de bases topográficas.

Palavras-Chave: Sistema de Informação Geográfica, GeoLiberty, Georreferenciamento, Gestão Municipal, Bases Topográficas.

ABSTRACT

Specialization Monograph
Specialization Course of the
Universidade Federal de Santa Maria

Author: Neimir Ruggini

Adviser: Joel da Silva

Date and place of the defense: Frederico Westphalen, december 14, 2013.

Currently, due to the increasing development of geographic information systems, one notes a growing need for this technology to be part of everyday people, helping and contributing precision and decision-making, adding more value to the information. Thus, the aim of this study was to describe a proposal to implement an extension of the solution Geoliberty, focusing on the management of spatial data, in order to contribute to the process of decision making the information generated .

For this research we adopted only open-source technologies , including the preparation of this material, valuing and contributing to the community Brazilian free software. The project scope has been determined from the necessity of application of GIS technologies in municipal public administration. Thus, this study aims to create a framework GIS, from the basic structure of spatial data defined GeoLiberty, to assist in the management of information cadastre of the city of Frederick. From the application of georeferencing techniques, it becomes possible to define their spatial distribution on the maps, overlapping layers of topographic bases.

LISTA DE FIGURAS

Figura 1 - Estrutura do Documento.....	18
Figura 2 - Visão geral da arquitetura SIG.....	21
Figura 3 - Representação geográfica (Fonte: Medeiros, 2010).....	22
Figura 4 - Formato de representação dos dados espaciais (Fonte: Medeiros, 2010).....	23
Figura 5 - Componentes de um SEAD.....	25
Figura 6 - Visão geral do modelo de alinhamento estratégico.....	27
Figura 7 - Visão da interface gráfica do Terra View.....	28
Figura 8 - Interface do Spring.....	29
Figura 9 - Vista Geral da ferramenta i3GEO.....	30
Figura 10 - Interface do gvSIG.....	31
Figura 11 - Interface do Quantum GIS.....	32
Figura 12 - Interface do MapInfo.....	34
Figura 13 - Exemplo de consulta na linguagem GeoMDQL.....	35
Figura 14 - Diagrama de classes do projeto GeoLiberty - Fonte: (VIANNA, 2012).....	36
Figura 15 - Fluxo de execução do código Python.....	38
Figura 16 - Fluxo de processo do Django - Fonte: (BRANDÃO, 2009).....	39
Figura 17 - Arquitetura de uma aplicação com o GeoDjango.....	40
Figura 18 - Tipos de dados espaciais do MySQL.....	41
Figura 19 - Exemplo de uso do OpenLayers.....	42
Figura 20 - Fluxograma da arquitetura proposta do framework MapGeoCity.....	46
Figura 21 - Visão geral do diagrama estrutural do framework.....	47
Figura 22 - Parâmetros de conexão com o Postgres.....	50
Figura 23 - Parâmetros de conexão com o MySQL.....	50
Figura 24 - Classe Imóvel derivando a classe Ponto do GeoLiberty.....	51
Figura 25 - Estrutura da tabela Imóvel.....	52
Figura 26 - Estrutura completa do framework na base de dados MySQL.....	52
Figura 27 - Classe Mpoly do projeto GeoLiberty.....	53
Figura 28 - Transformação da projeção espacial via módulo GDAL.....	53
Figura 29 - Transformação da projeção espacial via módulo GEOS.....	54
Figura 30 - Tratamento com os dados do PostGIS.....	54
Figura 31 - Tratamento com os dados do MySQL.....	54
Figura 32 - Amostragem do GeoLiberty em execução a partir da base MySQL.....	55
Figura 33 - Tela de apresentação do MapGeoCity.....	56
Figura 34 - Área administrativa do MapGeoCity.....	56
Figura 35 - Manutenção da tabela Tipo de Construção.....	57
Figura 36 - Área de pesquisa do framework.....	57
Figura 37 - Resultado do filtro de pesquisa.....	58
Figura 38 - Filtro dos Indicadores Estatísticos.....	59
Figura 39 - Imóveis por Tipo de Construção.....	59
Figura 40 - Amostragem dos indicadores estatísticos.....	60
Figura 41 - Camada do Bairro sobre o mapa da cidade.....	60
Figura 42 - Localização de Frederico Westphalen no mapa do Rio Grande do Sul e no Brasil.....	61

Figura 43 - Uso de SIG na administração municipal (Cavenaghi e Lima, 2013).....	62
Figura 44 - Base Cartográfica com as divisões do Zoneamento Urbano.....	63
Figura 45 - Base cartográfica do loteamento urbano após o tratamento efetuado.....	64
Figura 46 - Base cartográfica dos bairros.....	64
Figura 47 - Telas do cadastro de imóveis urbanos.....	65
Figura 48 - Ferramenta utilizada para padronizar os dados cadastrais de localização.....	66
Figura 49 - Lista de cidades que possuem a palavra "west"	67
Figura 50 - Filtro do cadastro de cidades do IBGE.....	67
Figura 51 - Associação das cidades origens com a destino.....	68
Figura 52 - Diversos Tipos de Logradouros.....	68
Figura 53 - Padronização do Tipo de Logradouro.....	68
Figura 54 - Relação de Tipos de Logradouros convertidos.....	68
Figura 55 - Assistente para geocodificação dos imóveis.....	69
Figura 56 - Recorte da Carta Topográfica.....	70
Figura 57 - Imóveis vinculados à quadra 50	70
Figura 58 - Distribuição espacial dos imóveis vinculados ao Bairro Centro.....	71

LISTA DE ABREVIATURAS E SIGLAS

API	APPLICATION PROGRAMMING INTERFACE
BI	BUSINESS INTELLIGENCE
BSD	BERKELEY SOFTWARE DISTRIBUTION
DRY	DON'T REPEAT YOURSELF
DSO	DYNAMIC SHARED OBJECTS
DXF	DRAWING EXCHANGE FORMAT
DWG	DATA WAREHOUSE GEOGRÁFICO
FTP	FILE TRANSFER PROTOCOL
GEOS	GEOMETRY ENGINE - OPEN SOURCE
GIS	GEOGRAPHIC INFORMATION SYSTEM
GEOTIFF	GEOCODED TAGGED IMAGE FILE FORMAT
GPL	GENERAL PUBLIC LICENCE
GPS	GLOBAL POSITIONING SYSTEM
HTTP	HYPertext TRANSFER PROTOCOL
IP	INTERNET PROTOCOL
IMG	IMAGE RASTER DATASET
JPEG	JOINT PHOTOGRAPHIC EXPERT GROUP
KML	KEYHOLE MARKUP LANGUAGE
MD5	MESSAGE-DIGEST algorithm 5
MTV	MODEL, TEMPLATE, VIEW
MVC	MODEL, VIEW, CONTROLLER
NCSA	NATIONAL CENTER OF SUPERCOMPUTING APPLICATIONS
OGC	OPEN GEOSPATIAL CONSORTIUM
OLAP	ONLINE ANALYTICAL PROCESSING
ORM	OBJECT ROLE MODELING
OSGEO	OPEN SOURCE GEOSPATIAL FOUNDATION
PDDI	PLANO DIRETOR DE DESENVOLVIMENTO INTEGRADO
SAD69	SOUTH AMERICAN DATUM 1969
SEAD	SISTEMAS ESPACIAIS DE APOIO À DECISÃO
SHP	SHAPEFILE

SIG	SISTEMA DE INFORMAÇÃO GEOGRÁFICA
SOLAP	SPATIAL ONLINE ANALYTICAL PROCESSING
SQL	STRUCTURED QUERY LANGUAGE
SSL	SECURE SOCKETS LAYER
URL	UNIFORM RESOURCE LOCATOR
UTM	UNIVERSAL TRANSVERSA DE MERCATOR
WFS	WEB FEATURE SERVICE
WMF	WEB MAP SERVICE
WKB	WELL-KNOW BINARY
WKT	WELL-KNOW TEXT

Sumário

CAPÍTULO 1 - INTRODUÇÃO.....	15
1.1 Contextualização.....	15
1.2 Motivação.....	16
1.3 Objetivos.....	17
1.3.1 Objetivo Geral.....	17
1.3.2 Objetivos Específicos.....	17
1.4 Estrutura do Documento.....	17
CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA.....	19
2.1 Literatura Relacionada.....	19
2.1.1 Geoprocessamento.....	19
2.1.2 Sistema de Informação Geográfica.....	20
2.1.3 Sistemas Espaciais de Apoio à Decisão (SEAD).....	24
2.1.4 Gestão de TI x Alinhamento Estratégico Organizacional.....	26
2.2 Trabalhos relacionados.....	28
2.2.1 TerraView	28
2.2.2 Spring.....	29
2.2.3 i3Geo.....	30
2.2.4 gvSIG.....	31
2.2.5 Quantum GIS.....	32
2.2.7 GeoMDQL.....	34
2.2.8 Comparativo dos trabalhos relacionados em relação ao projeto de pesquisa.....	35
2.2.9 GeoLiberty.....	36
2.3 Tecnologias.....	37
2.3.1 Python.....	37
2.3.2 Django.....	38
2.3.3 GeoDjango.....	40
2.3.4 MySQL.....	40
2.3.5 OpenLayers.....	42
2.3.6 Apache.....	43

CAPÍTULO 3 - DESENVOLVIMENTO DO FRAMEWORK MAPGEOCITY.....	46
3.1 Definição.....	46
3.2 Arquitetura do framework proposto.....	46
3.2 Modelo de Dados.....	47
3.3 Preparação do ambiente de trabalho.....	49
3.4 Etapas de migração do framework Geoliberty	49
3.4.1 Criação e configuração da Base de Dados MySQL.....	50
3.4.2 Criação do framework MapGeoCity.....	50
3.4.3 Exportação dos dados do Postgres para o MySQL.....	54
3.5 Módulo de Pesquisa das Informações.....	55
CAPÍTULO 4 - ESTUDO DE CASO.....	61
4.1 Caracterização Geral.....	61
4.2 Plano Diretor de Desenvolvimento Integrado.....	61
4.3 Sistema de Tributos.....	65
4.4 Resultados obtidos.....	69
CAPÍTULO 5 - CONCLUSÃO E TRABALHOS FUTUROS.....	72
REFERÊNCIAS BIBLIOGRÁFICAS.....	74

CAPÍTULO 1 - INTRODUÇÃO

1.1 Contextualização

Atualmente, devido à crescente evolução dos sistemas de informação geográfica, tem se observado uma relação mais harmônica dos usuários com essa tecnologia, pois antigamente ela era imprecisa, pouco organizada e de difícil acesso. O sucesso é tanto que, sem percebermos, estamos o tempo todo imersos nesse novo conceito de tecnologia, a qual agrega valor à informação das mais variadas formas. Tanto pelas redes sociais como pela popularização dos serviços baseados em localização embutidos nos dispositivos móveis.

Porém, a construção de uma solução baseada nos conceitos de web geográfica é desafiadora e é conhecido que um dos maiores desafios dessa solução é garantir a qualidade e agilidade na obtenção dessas informações.

Nessa perspectiva, o estudo deste tema torna-se relevante para a área de sistemas de informação geográfica, agregando conhecimento sobre a integração de diversas tecnologias. Pois, nota-se uma diversidade de sistemas, tanto gratuitos como pagos, na área de informação geográfica, porém, ainda existem diversas barreiras quanto à operacionalização destas, devido ao fato delas serem fechadas quanto à expansão de suas funcionalidades.

Sabe-se que, cada vez mais, é necessário mais agilidade e saber escolher as ferramentas certas no desenvolvimento de sistemas. E a adoção de *frameworks* certos pode contribuir em muito na tarefa repetitiva da codificação de sistemas. Atualmente, os bons *frameworks* seguem o padrão MVC, cujo objetivo é separar a lógica de aplicação, da interface do usuário e do fluxo da aplicação.

Uma das tecnologias que atualmente vem ganhando cada vez mais adeptos no desenvolvimento de Sistemas de Informação Geográfica é o *framework* Django, escrito na linguagem de programação Python, e sua extensão espacial GeoDjango. Este framework faz uso da arquitetura MVC para desenvolvimento de aplicações, estimulando os programadores a reaproveitar o código já feito, afim de evitar repetição.

Com isso, foi adotado neste estudo somente ferramentas de uso gratuito para que, futuramente, seja dada continuidade a este projeto sem gerar ônus para os pesquisadores.

Dessa forma, este trabalho descreve a proposta de implementação de uma extensão da

solução GeoLiberty nomeada MapGepCity, escrita em Python, a ser aplicada no processo de gerenciamento, manipulação e visualização de dados geográficos. E como estudo de caso, será adotado o processo de geoprocessamento do cadastro imobiliário do município de Frederico Westphalen/RS. Essa integração é um grande diferencial para o processo de tomada de decisão, facilitando o acesso às informações, eliminando duplicidades e garantindo maior integridade na entrada de dados.

1.2 Motivação

O propósito de desenvolvimento deste projeto foi embasado na necessidade de integração do Sistema de Gestão Municipal da prefeitura municipal de Frederico Westphalen com as ferramentas SIG, visando agilizar a análise de informações com maior rapidez e auxiliar no processo de tomada de decisão.

O sistema de Gestão Pública Municipal que a Prefeitura Municipal de Frederico Westphalen/RS utiliza faz parte de um dos produtos desenvolvidos pela empresa Digifred Informática Ltda (DIGIFRED, 2013), situada nesta mesma cidade. Porém, este sistema não possui recursos de georreferenciamento, dessa forma, partindo-se da definição do Plano Diretor de Desenvolvimento Integrado do Município de Frederico Westphalen (2010), regido pela Lei Municipal nº 3.620, de 01 de dezembro de 2010, onde o Artigo 187, § 2º menciona que “O Sistema de Informações Municipais deve, progressivamente, disponibilizar informações georreferenciadas, em meio digital, para as devidas cópias impressas”, foi proposto a integração dos dados cadastrais com as ferramentas de geoprocessamento a fim de melhorar o processo de gestão dessas informações.

Nessa perspectiva, a adoção das técnicas de geoprocessamento que as ferramentas SIG oferecem, possibilitam uma nova condição de armazenamento e manipulação de dados espaciais e socioeconômicos. Esses dados são de grande valia para o gestor municipal, pois com eles é possível acompanhar as necessidades no ambiente que estão inseridas. Dessa forma, esse estudo visa abstrair do usuário a complexidade que normalmente é agregada às atividades de mapeamento cadastral dos imóveis na cidade.

1.3 Objetivos

Os objetivos do projeto serão subdivididos em dois grupos: gerais e específicos. No primeiro, será abordado a intenção central do projeto e, no segundo, a definição das metas usadas para se alcançar o objetivo central.

1.3.1 Objetivo Geral

Implementar uma extensão da solução GeoLiberty, base para a construção de sistemas de informação geográfica, nomeada de MapGeoCity, usada para a integração e visualização de dados geográficos, tendo como estudo de caso a aplicação desta tecnologia integrada ao sistema de controle imobiliário do município de Frederico Westphalen/RS.

1.3.2 Objetivos Específicos

Para se alcançar o objetivo central do projeto foram definidos os seguintes objetivos específicos:

- Revisões bibliográficas e estudo de trabalhos correlacionados à área de pesquisa;
- Pesquisar as tecnologias de uso gratuito e de código aberto a serem adotadas no projeto a fim de contribuir com a comunidade brasileira de software livre com o estudo em questão;
- Aprender a linguagem de programação Python, o framework Django e sua extensão espacial GeoDjango e, também, a integração com o Banco de Dados MySQL;
- Contribuir com a evolução do projeto GeoLiberty;
- Pesquisar e aprofundar os estudos sobre a área de sistemas de informação geográfica;
- Construir uma ferramenta, a partir das tecnologias estudadas, para auxiliar no processo de tomada de decisão das informações georreferenciadas, integrando o processamento de dados analíticos com as coordenadas geográficas.

1.4 Estrutura do Documento

De acordo com a Figura 1, o presente trabalho está dividido em cinco capítulos, sendo o primeiro composto pela introdução, contextualização, motivação e a definição dos objetivos

do trabalho. No capítulo 2 é apresentada a revisão bibliográfica das tecnologias usadas na pesquisa deste trabalho, onde será feito um levantamento da literatura e trabalhos relacionados e as tecnologias envolvidas. No capítulo 3 será abordado a definição da arquitetura, modelo de integração de dados espaciais e a implementação do *framework* MagGeoCity. No capítulo 4 é apresentado o estudo de caso a ser aplicado este pesquisa. A conclusão, principais contribuições e os trabalhos futuros são apresentados no capítulo 5, sendo por fim apresentadas a referência bibliográfica usada como suporte teórico e justificativo para a realização deste trabalho.

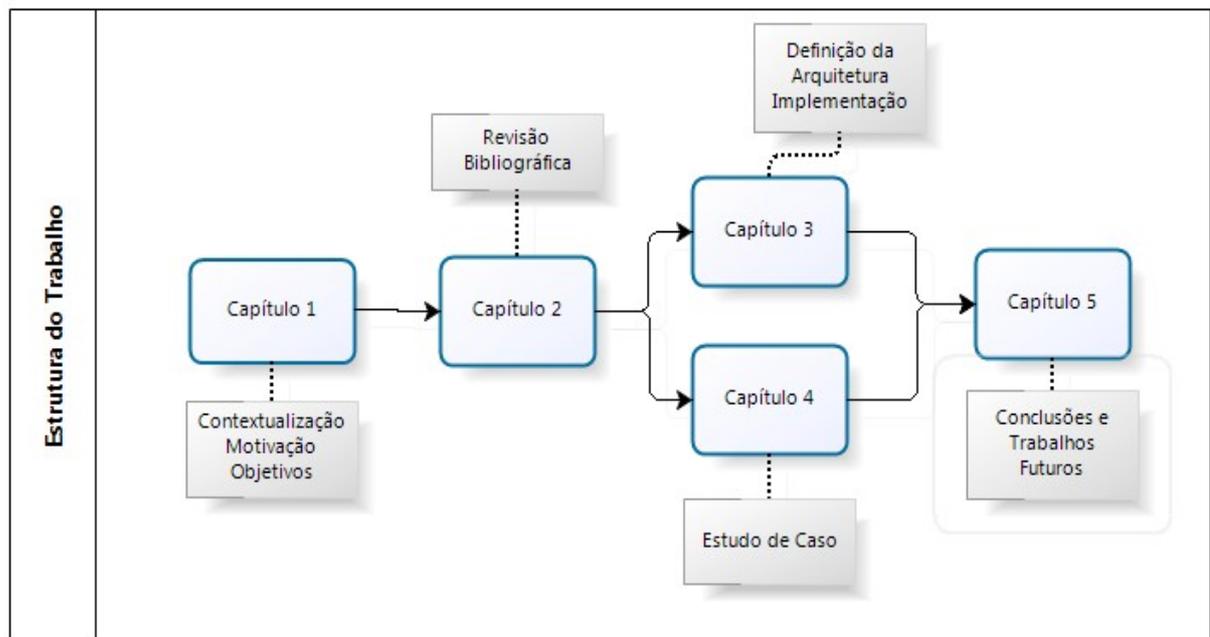


Figura 1 - Estrutura do Documento

CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA

2.1 Literatura Relacionada

2.1.1 Geoprocessamento

Segundo Câmara et. al., (2001) o termo Geoprocessamento teve origem no Brasil na década de 80, por pesquisadores da Universidade do Rio de Janeiro e denota o processo de realizar o tratamento de informação geográfica através de técnicas matemáticas e computacionais. O conjunto de ferramentas utilizadas nesse processo são chamadas de Sistemas de Informação Geográfica, as quais permitem realizar análises mais complexas, integrar fontes heterogêneas e construir bancos de dados georreferenciados.

Devido à sua crescente popularização através de ferramentas como o Google Earth e Google Maps, inclusive em dispositivos móveis, esse tipo de tecnologia vem a cada dia crescendo e sendo muito requisitada para auxiliar no processo de tomada de decisão de tarefas cotidianas.

Para a resolução de questões de ordem espacial é necessário fazer uso de ferramentas de geoprocessamento as quais se subdividem em quatro áreas: Cartografia Digital, Sensoriamento Remoto, Sistema Gerenciador de Banco de Dados e Análise Espacial. A partir dessa subdivisão, é possível desenvolver metodologias e técnicas para questões de ordem geográfica e espacial de forma rápida e objetiva.

Segundo Filho (2000), Cartografia Digital é um conjunto de ferramentas, incluindo programas e equipamentos, orientado para a conversão para o meio digital, armazenamento e visualização de dados espaciais. Um sistema de Cartografia Digital têm como ênfase a produção final de mapas.

Para Rosa (2003), o Sensoriamento Remoto pode ser definido como a forma de obter informações de um objeto ou alvo, sem que haja contato físico com o mesmo. As informações são obtidas utilizando-se a radiação eletromagnética gerada por fontes naturais como o Sol e a Terra, ou por fontes artificiais como por exemplo o radar.

De acordo com Elsmasri e Navathe (2006), um Sistema Gerenciador de Banco de Dados é uma coleção de programas que permite aos usuários criar e manter um banco de

dados, facilitando o processo de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações.

Câmara et. al., (2001) define que a ênfase da Análise Espacial é mensurar propriedades e relacionamentos, levando em conta a localização espacial do fenômeno em estudo de forma explícita, ou seja, incorporar o espaço à análise que se deseja fazer.

2.1.2 Sistema de Informação Geográfica

Conforme definição de Câmara et. al. (2001), os Sistemas de Informação Geográfica são sistemas automatizados usados para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente à informação e indispensável para analisá-la.

Dessa maneira, eles servem de instrumento para a organização do território, da gestão de recursos naturais, para o desenvolvimento sócio-econômico, pois permitem monitorar e avaliar em tempo real um número considerável de dados a partir de um referencial espacial, facilitando a simulação de ambientes e soluções alternativas num processo de tomada de decisão.

Para Cavenagui e Lima (2006), as ferramentas geográficas são meios confiáveis de se organizar as variáveis relevantes ao gerenciamento do município. Com a possibilidade de ser integrado aos demais sistemas da gestão municipal, possibilita maior acurácia no processo de arrecadação e gestão dos tributos municipais.

Segundo Câmara et. al. (2001), o Sistema de Informação Geográfica, cuja sigla é SIG, é aplicado em sistemas que realizam o tratamento computacional de dados geográficos e armazenam a geometria e os atributos dos dados que estão georreferenciados, ou seja, cuja localização na superfície terrestre seja representada numa projeção cartográfica. De modo geral, pode-se dizer que um SIG possui os seguintes componentes:

- Interface com o usuário;
- Entrada e integração de dados;
- Funções de processamento gráfico e de imagens;
- Visualização e plotagem;
- Armazenamento e recuperação de dados (organizados sob a forma de um banco de dados geográfico).

Conforme apresentado na Figura 2, a arquitetura de um Sistema de Informação Geográfica é composta por: Interface, Entrada e Integração dos Dados, Consulta e Análise Espacial, Gerência de Dados Espaciais, Visualização e Plotagem e um Banco de Dados Geográfico. Esses são os principais componentes de um SIG, onde cada sistema, em função de seus objetivos e necessidades, implementa estes componentes de forma distinta.

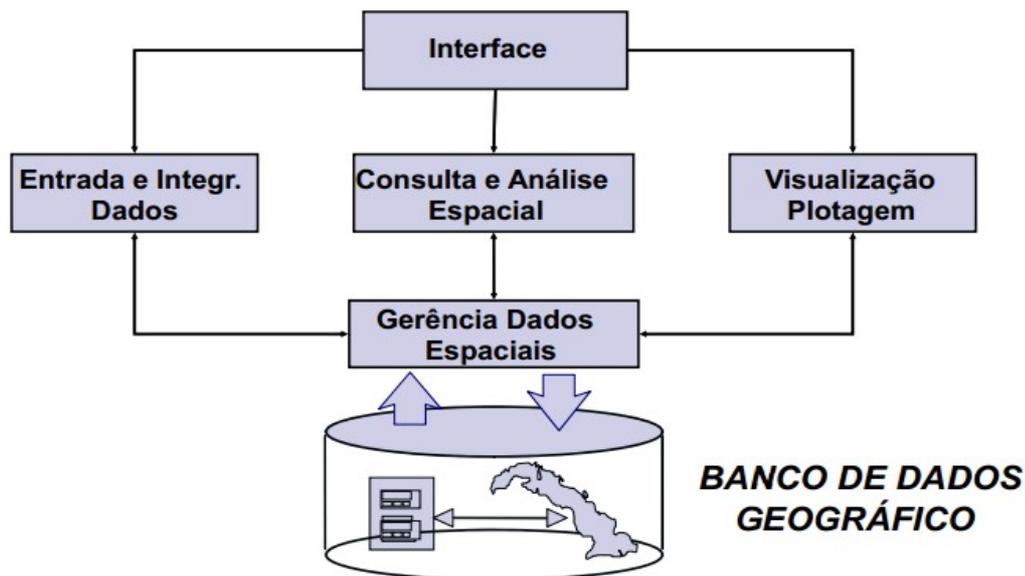


Figura 2 - Visão geral da arquitetura SIG

Atualmente, há uma série de softwares de SIG que possuem ótimos recursos para auxiliar no processo de tomada de decisão. Uns possuem mais características e recursos que outros, dependendo do foco de atuação. Podendo ser para planejamento urbano, sistema viário, análise de recursos naturais, monitoramento do tráfego, etc. Porém, há limitações quanto ao uso ou extensibilidade em muitos destes, especialmente os de código fechado.

Conforme observa-se na figura acima, o desenvolvimento de um SIG é construído de forma integrada e seus dados podem ser armazenados em SGBDs (Sistemas Gerenciadores de Banco de Dados) com características espaciais.

Segundo Silva (2004), as principais características de um Sistema de Informação Geográfica são:

- A capacidade de inserir e integrar, numa única base de dados, informações espaciais provenientes de dados cartográficos, dados censitários e cadastros urbano e rural, imagens de satélite, redes e modelos numéricos de terreno;
- Oferecer mecanismos para combinar as várias informações, através de algoritmos de

manipulação e análise, bem como para consultar, recuperar, visualizar e plotar o conteúdo da base de dados georreferenciados.

Dados georreferenciados podem ser representados pela natureza alfanumérica ou geográfica. Por natureza alfanumérica entende-se a representação sob a forma de textos (letras e números), os quais caracterizam os objetos geográficos. Já a representação geográfica, conforme vemos na Figura 3, é caracterizada por pontos, linhas e polígonos. Estes dados, também chamados de dados referenciados geograficamente, detalham e expõem fenômenos geográficos. O dado georreferenciado descreve a localização do fenômeno geográfico ligado a uma posição sobre ou sob a superfície da terra.

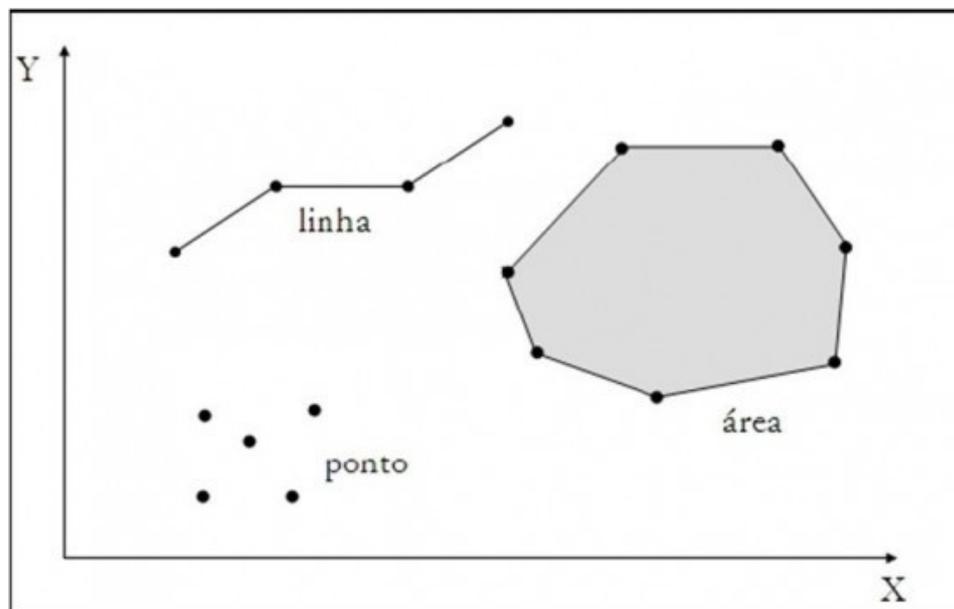


Figura 3 - Representação geográfica (Fonte: Medeiros, 2010)

Diante dessas três representações espaciais, é possível identificar qualquer informação geográfica. Com a representação do ponto é possível identificar localizações ou ocorrências no espaço, representada por um par de coordenadas espaciais ou em um par ordenado (x,y) . Uma linha representa qualquer tipo de feição onde ela é representada por um conjunto de pontos. Já com uma área ou polígono, é possível representar uma região do plano limitada por uma ou mais linhas poligonais conectadas.

O armazenamento e a manipulação de dados georreferenciados, exemplificados na Figura 4, poderá ser feito de duas formas: vetorial ou matricial. As estruturas matriciais, denominadas de *raster*, possuem seus valores associados a uma matriz de células, onde cada célula corresponde a uma coordenada de linha e coluna do mundo real. Já a estrutura vetorial

é representada por um par de coordenadas de latitude e longitude geográfica, sendo mais preciso do que a matricial.

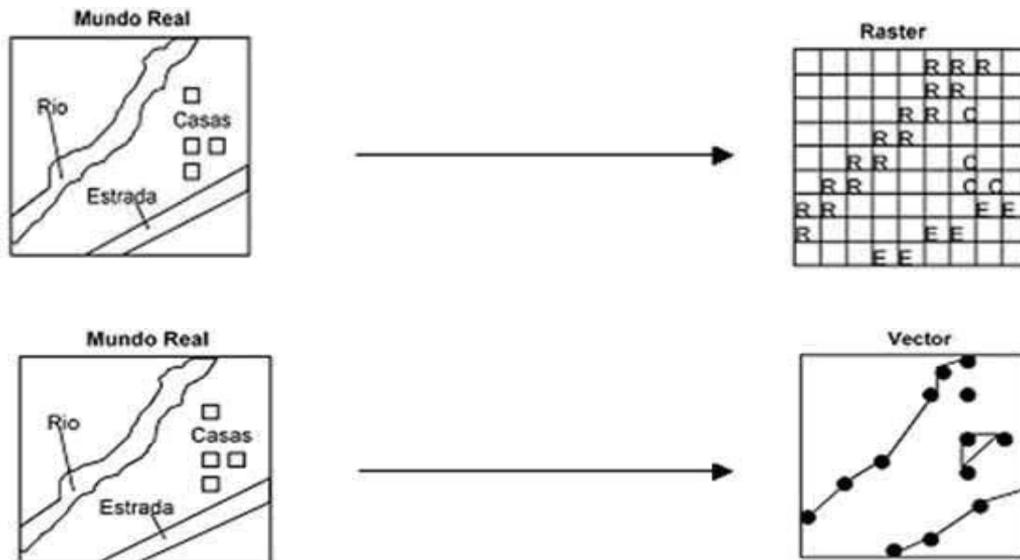


Figura 4 - Formato de representação dos dados espaciais (Fonte: Medeiros, 2010)

Segundo Pimenta et. al. (2012), mapas representam modelos simplificados da realidade. Mapas temáticos (temas ou *themes*) constituem representações cartográficas de informações sobre um determinado assunto ou tema específico. Nos SIGs, os dados geográficos são estruturados em várias camadas de informação, também denominadas de planos ou *layers*, cada um representando um tema diferente, que podem ser sobrepostos (*overlay*) em diferentes sequências, conforme o objetivo de cada trabalho. Nos Sistemas de Informações Geográficas, os dados são armazenados em dois tipos de arquivos: vetorial (*vector*) ou matricial (*raster*). No sistema vetorial, os dados são representados por pontos, linhas ou polígonos, conforme o tipo de informação representada: feições pontuais, lineares (ex.: estradas e rios) ou áreas fechadas (lagoas, áreas urbanas ou lavouras de milho), de acordo com a escala geográfica considerada. No sistema matricial ou *raster*, a área geográfica é representada por uma matriz subdividida em linhas e colunas, os *Pixels* ou células regulares. O formato dos arquivos também varia de acordo com o tipo de informação armazenada. Arquivos com extensões SHP (*Shapefiles*), DXF (*Drawing Exchange Format*), DGN (*Design*) e KML (*Keyhole Markup Language*) são exemplos de mapas digitais em formato vetorial. Arquivos com extensões GeoTIFF (*Geocoded Tagged Image File Format*), IMG (*Image Raster Dataset*) e JPEG (*Joint Photographic Experts Group*) são exemplos de arquivos digitais em formato matricial.

De acordo com Oliveira e Nogueira (2012), a construção de mapas temáticos é uma solução utilizada para aumentar e agilizar a disponibilização de informações para a sociedade e para o governo. Auxiliando, portanto, no processo de tomada de decisão com o objetivo de otimizar a aplicação de recursos.

Segundo Borrero (2013), "os SIG têm sido uma das mais revolucionárias e exitosas tecnologias introduzidas nas últimas décadas. Claramente têm transformado múltiplos processos técnicos e contribuindo na construção da Sociedade do Conhecimento em que vivemos. Os SIG e a forma como são aplicados para a produção, acesso e aplicação das bases de dados fundamentais e a geração de informação geoespacial constituem um dos aspectos críticos para o futuro dos Institutos Geográficos na América Latina e Caribe. Como os SIG estão ao alcance de todas as pessoas, os institutos geográficos deixaram de ser os produtores exclusivos da informação espacial. Hoje, distintas instituições, públicas e privadas, locais e internacionais, estão em igual ou melhor capacidade de produzir e tornar disponíveis dados espaciais fundamentais e suas aplicações, com maior velocidade e para um número crescente de aplicações, OpenStreetMaps ou Google Maps, por exemplo, podem dispor de mais e melhores mapas que muitos países com dificuldades para produzir a informação geoespacial que demanda o usuário do Século XXI".

2.1.3 Sistemas Espaciais de Apoio à Decisão (SEAD)

De acordo com Densham (1991), o aparecimento do conceito SEAD está relacionado com a necessidade de expansão das capacidades dos SIG para a resolução de problemas complexos de decisão espacial. Ele disponibiliza uma infraestrutura que integra os sistemas de gestão de base de dados com os modelos analíticos, os dispositivos gráficos de visualização, a possibilidade de produção de relatórios tabulares e a inclusão do conhecimento dos decisores.

O uso de técnicas de SIG possibilitam um entendimento melhor do problema, acelerando o processo de resolução, pois a análise espacial tem contribuído para subsidiar a tomada de decisão, que através de sua manipulação, gera-se novas informações. Dessa maneira, a análise espacial é um dos aspectos mais importantes para se criar a informação com recursos de geoprocessamento.

Problemas de decisão espacial normalmente são usados quando o problema em questão envolve dados geográficos. Para Malczewski (1997), os problemas de decisão

espacial caracterizam-se por permitirem a inclusão de um grande número de alternativas (quantitativa ou qualitativa) de decisão, cujas consequências ou resultados diferem no espaço.

Ainda segundo Malczewski (1997), as principais características dos problemas de decisão espacial incluem:

- um grande número de alternativas de decisão;
- os resultados ou consequências das alternativas de decisão são variáveis;
- cada alternativa é avaliada com base em vários critérios;
- alguns dos critérios podem ser qualitativos, enquanto outros podem ser quantitativos,
- normalmente há mais de um tomador de decisão (ou grupo de interesse) envolvidos no processo de tomada de decisão;
- os tomadores de decisão têm preferências diferentes no que diz respeito à importância relativa dos critérios de avaliação e as consequências de decisões;
- as decisões são muitas vezes cercadas pela incerteza.

Segundo Densham(1991), as características de um SEAD facilitam o processo de pesquisa de decisão, podendo ser caracterizado como iterativo, integrador e participativo. É iterativo porque gera uma série de decisões alternativas, beneficiando análises posteriores, dado que o ganho de informação conseguido neste processo servirá para ser, seguidamente, utilizado pelo decisor em novas avaliações. É participativo porque o decisor desempenha um papel ativo na definição do problema, ao efetuar a análise e ao avaliar os resultados. O benefício da participação resulta na integração dos conhecimentos do decisor sob a forma de dados quantitativos nos modelos e na qualidade da informação utilizada.

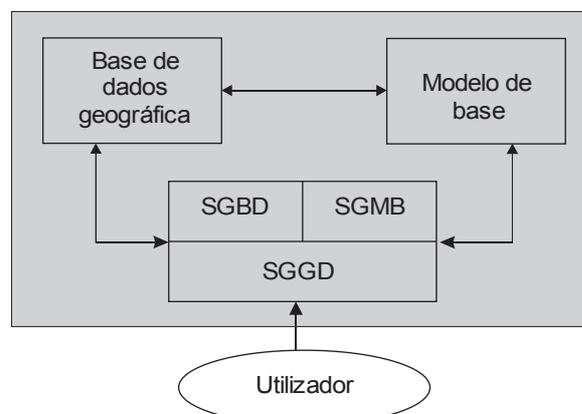


Figura 5 - Componentes de um SEAD

Conforme podemos observar na Figura 5, Malczewski (1997) define um SEAD com os seguintes componentes:

- um Sistema de Gestão de Base de Dados (SGBD), que contém as funções necessárias para gerir a base de dados geográfica, permitindo a leitura de dados de localização (coordenadas, locais, etc), de dados topológicos (pontos, linhas, polígonos e suas relações) e dados de atributos (geologia, elevação, etc). Deverá prever a possibilidade de obtenção de vistas a partir de questionamentos feitos ao sistema, e também efetuar a aquisição, o armazenamento, a manipulação e integração de dados lógicos.

- um Sistema de Gestão do Modelo de Base (SGMB), que contém as funcionalidades para gerir o modelo de base. Deverá permitir uma diversidade de análises, como, por exemplo, a busca por objetivos, otimização de percursos, simulação de cenários. Este modelo deve permitir a definição de níveis geográficos, como por exemplo, a seleção de cores e padrões para serem utilizadas em temas baseadas em atributos multivalorados.

- um Sistema de Geração e de Gestão de Diálogos (SGGD), responsável por gerar a interface entre o utilizador e o resto do sistema. Ela deverá ser amigável e intuitiva, facilitando a interação entre o utilizador e o sistema. Aconselha-se uma interface bem desenhada, com menus e botões intuitivos, configuração personalizada, fazer uso de mensagens de erro e de ajuda, para que o utilizador possa realizar suas tarefas com maior facilidade.

2.1.4 Gestão de TI x Alinhamento Estratégico Organizacional

Segundo Fagundes (2013), a complexidade das tecnologias de informação é diretamente proporcional à complexidade das organizações de TI, e para isso, é necessário a adoção de vários padrões, como, por exemplo, Cobit, ITIL, CMMI, PMBOK, desenvolvidos por organizações internacionais que fomentam a governança de TI e ajudam as organizações de TI a desenhar modelos de gestão.

Devido aos crescentes avanços tecnológicos que a área de TI tem obtido e devido à enorme massa de dados que os sistemas de informação tem gerado, torna-se praticamente impossível o gerenciamento dessas informações sem o uso adequado de ferramentas certas e sem a definição estratégica de um plano para obtenção desses dados para a tomada de decisão.

A norma NBR ISO/IEC 38500:2009, define Gestão de TI como “o sistema de controles e processos necessários à realizar um objetivo estratégico definidos pelo órgão que governa a organização”.

De acordo com Fernandes e Abreu (2012), o alinhamento estratégico é o processo de transformar a estratégia do negócio em estratégias e ações de TI que garantam que os objetivos de negócio sejam apoiados. Dessa forma, o seu método de aplicação é bidirecional, ou seja, da estratégia do negócio para a estratégia de TI e vice-versa, pois a TI pode potencializar estratégias de negócio que seriam impossíveis de serem implantadas sem o auxílio da TI.

O alinhamento estratégico pode ocorrer em vários momentos na vida da empresa, podendo ser um plano estratégico, plano de negócios ou similar. Com isso, o resultado da definição estratégica é desmembrado afim de que os setores da empresa possam tomar conhecimento dos seus objetivos.

Conforme podemos observar na Figura 6, proposto por Henderson e Venkatraman (1999), a relação entre a TI e os negócios mostra a importância estratégica que a TI possui dentro das organizações. Onde é tratado o sincronismo entre as ações e as estratégias a fim de se alcançar os objetivos propostos. Vemos na figura, que a estratégia de TI influencia e é influenciada pela estratégia de negócio e interage bidirecionalmente com a infraestrutura e os processos de TI e com a infraestrutura e os processos organizacionais. Essa ligação visa obter o máximo de compartilhamento de recursos, visto que o resultado dessa interação gera um forte impacto no momento em que se define a arquitetura de TI.

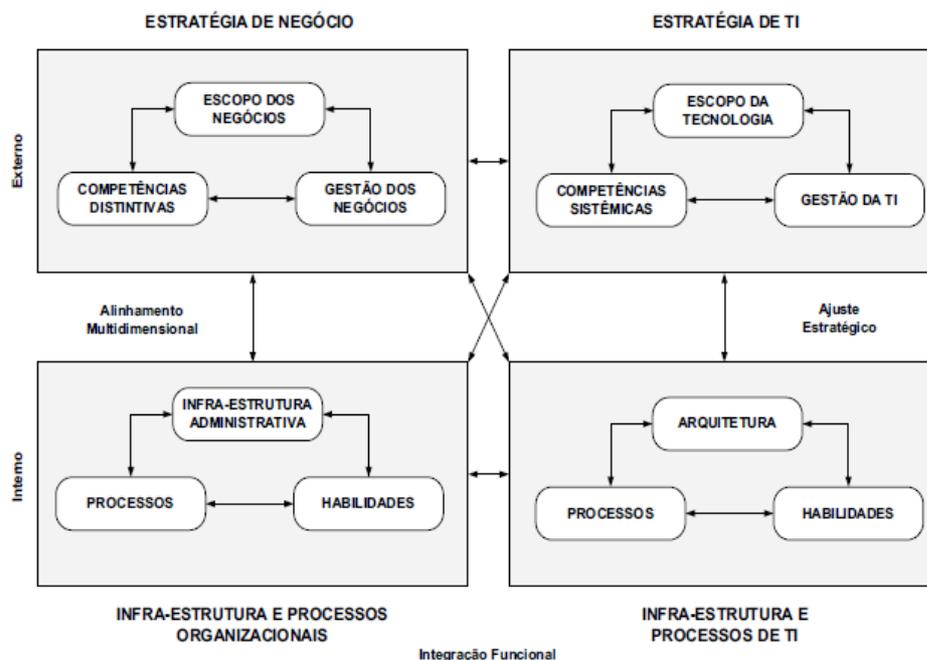


Figura 6 - Visão geral do modelo de alinhamento estratégico

Diante disso, a Digifred Informática (2013), empresa desenvolvedora de softwares para a gestão pública municipal, com sede no município de Frederico Westphalen/RS, possui um conjunto de softwares que auxiliam o gestor público no gerenciamento de informações municipais em áreas como Administração, Fazenda, Agricultura, Obras, Saúde e Educação. Possui aproximadamente 50 clientes entre prefeituras, câmaras, autarquias e consórcios, distribuídos na região noroeste do estado do Rio Grande do Sul e também na região de Valença/RJ. Porém, os sistemas de gestão que fazem parte do portfólio da empresa carecem de um módulo de gestão geográfica para tomada de decisão estratégica dessas informações, devido à falta de integração desses dados com ferramentas de geoprocessamento.

A Digifred, visando melhorar a gestão estratégica das informações geradas pelo seus produtos, busca ampliar a sua presença no mercado corporativo, através do alinhamento da TI com ferramentas de tomada de decisão, afim de gerar um diferencial competitivo, por isso ela colabora e apoia a ideia deste projeto.

2.2 Trabalhos relacionados

2.2.1 TerraView

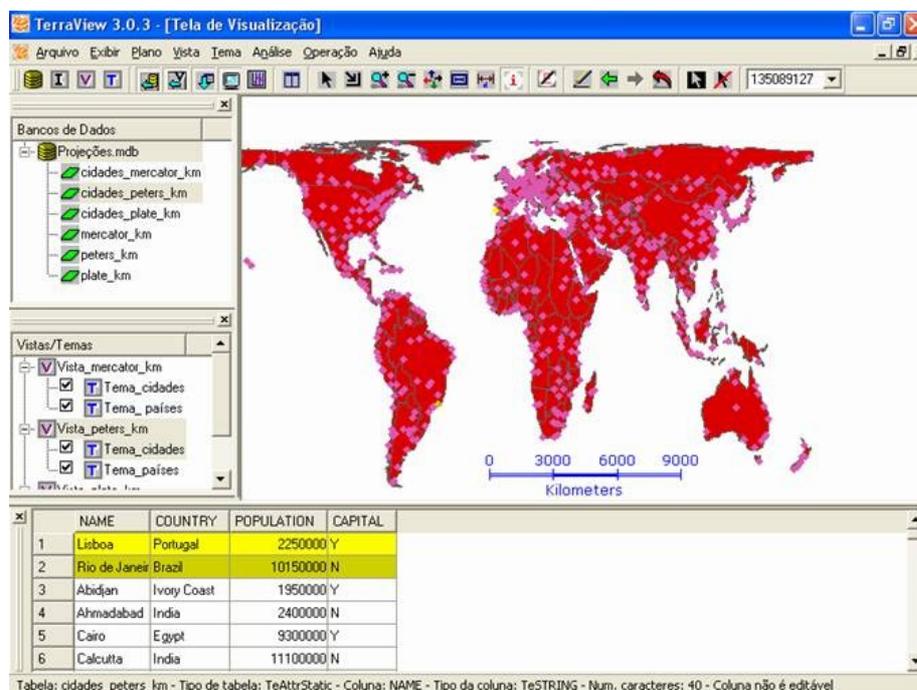


Figura 7 - Visão da interface gráfica do Terra View

O TerraView (2010) é um aplicativo construído a partir da biblioteca TerraLib, desenvolvido pelo INPE. Ele manipula dados vetoriais (pontos, linhas, polígonos) e matriciais (grades e imagens). Observa-se na Figura 7 uma amostragem da funcionalidade deste programa, o qual possui recursos de consulta de análise de dados geográficos, através de uma interface simples, facilitando a visualização de dados geográficos.

Ele foi construído a partir da biblioteca de geoprocessamento TerraLib para exploração e visualização de dados geográficos. É considerado um software livre sob os termos da Licença Pública Geral do GNU (GPL). Fazendo parte do pacote de softwares TerraView existe o programa TerraView Políticas Sociais, usado para análises e interpretações espaciais da realidade social das áreas urbanas. Essa ferramenta permite aos usuários realizarem diagnósticos e cálculos de diferentes questões sociais, econômicas e demográficas.

2.2.2 Spring

O Sistema de Processamento de Informações Georreferenciadas (SPRING) é um software para SIG desenvolvido pelo Instituto Nacional de Pesquisas Espaciais (INPE). Possui versões para as plataformas Windows e Unix, com capacidade para suportar grandes volumes de dados, em diferentes formatos vetoriais e matriciais.

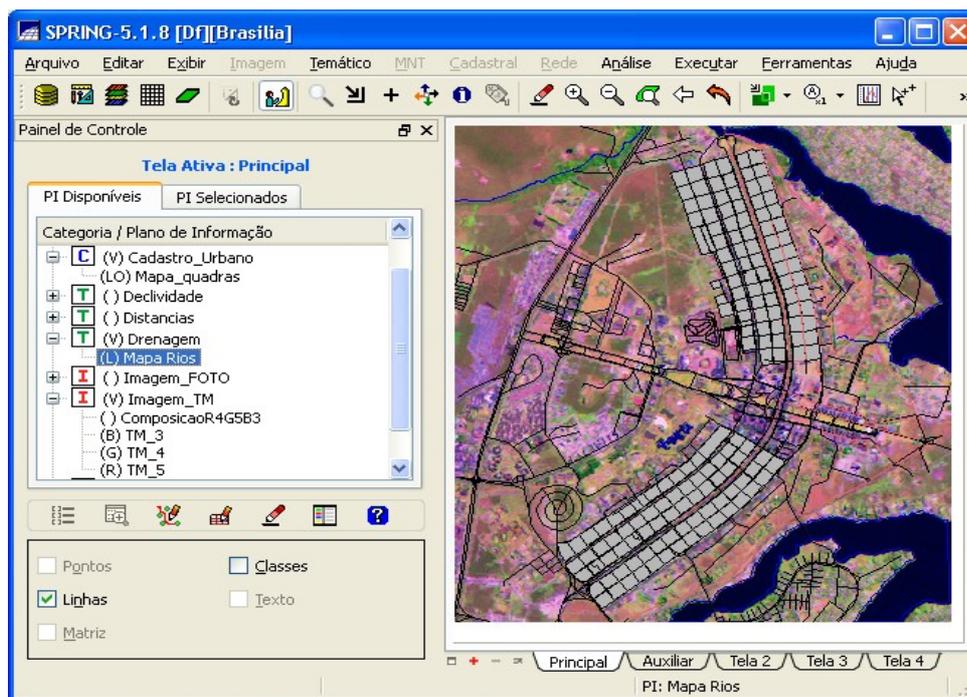


Figura 8 - Interface do Spring

Ele é composto pelos módulos Spring, Impima e Scarta. Na Figura 8, vemos o módulo Spring com sua diversidade de ferramentas para geoprocessamento. O módulo Impima é responsável pela conversão de imagens de satélite para o formato nativo do Spring. Já o módulo Scarta é responsável pela produção de produtos cartográficos, como mapas e cartas, para impressão em diversos formatos (MEDEIROS, 2012).

Contém uma infinidade de algoritmos e ferramentas para processamento digital de imagens, interpolação e análise espacial, modelagem numérica de terreno e interação com bancos de dados espacialmente georreferenciados. Outro diferencial dessa ferramenta é o fato dela ser baseada em um modelo de dados orientado à objetos com a combinação de janelas e menus que utilizam a Linguagem Espaço Geográfica baseada em Álgebra (LEGAL) (MEDEIROS, 2012).

Devido à necessidade de compatibilidade com as novas arquiteturas de 64 bits, para a plataforma Windows, apenas o MySQL é suportado, entretanto, para Linux, as opções são: MySQL, PostgreSQL e Oracle. Porém, para as versões de 32 bits, a compatibilidade é maior. Possui suporte aos novos formatos de dados geográficos como o DXF, DWG, KML e JPEG2000. E também é multiusuário, permitindo o acesso simultâneo ao mesmo banco de dados (MEDEIROS, 2012).

2.2.3 i3Geo

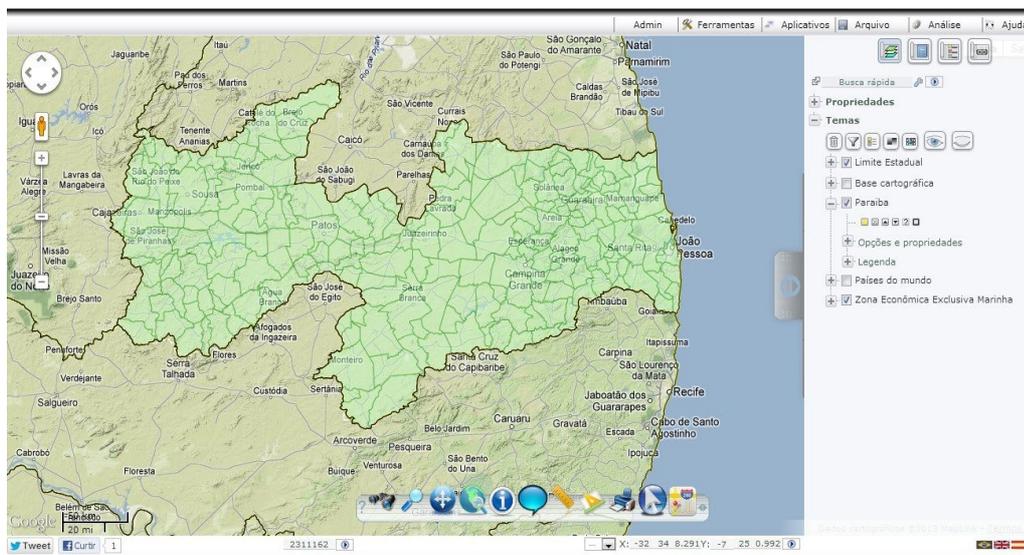


Figura 9 - Vista Geral da ferramenta i3GEO

A Interface Integrada para Internet de Ferramentas de Geoprocessamento (i3GEO) é um software para internet cuja base foi concebida através de software livres, dentre eles o MapServer. De propriedade do Ministério do Meio Ambiente (MMA). Seu principal objetivo é a disponibilização de dados geográficos para análise, compartilhamento e geração de mapas sob demanda por meio de uma interface interativa, repleta de ferramentas e controles de navegação e de customização da ferramenta (I3GEO, 2013), conforme podemos perceber na Figura 9.

Possui integração com diversos serviços, dentre eles a API do Google Maps, Google Earth, OpenLayers e outros mais, e, recentemente, foi anunciada a integração com o gvSIG (GVSIG, 2013), software livre para SIG. Com este aplicativo, é possível efetuar conexão remota a outras bases de dados geográficas via protocolo Web Map Service (WMS).

2.2.4 gvSIG

Segundo Medeiros (MEDEIROS, 2012), gvSIG é um software livre de origem europeia, desenvolvido em java, e é distribuído sob a licença GNU GPL. Possui diversas publicações sobre o seu uso na cartografia cadastral e gestão de estradas, no inventário florestal e no diagnóstico da distribuição dos telefone públicos.

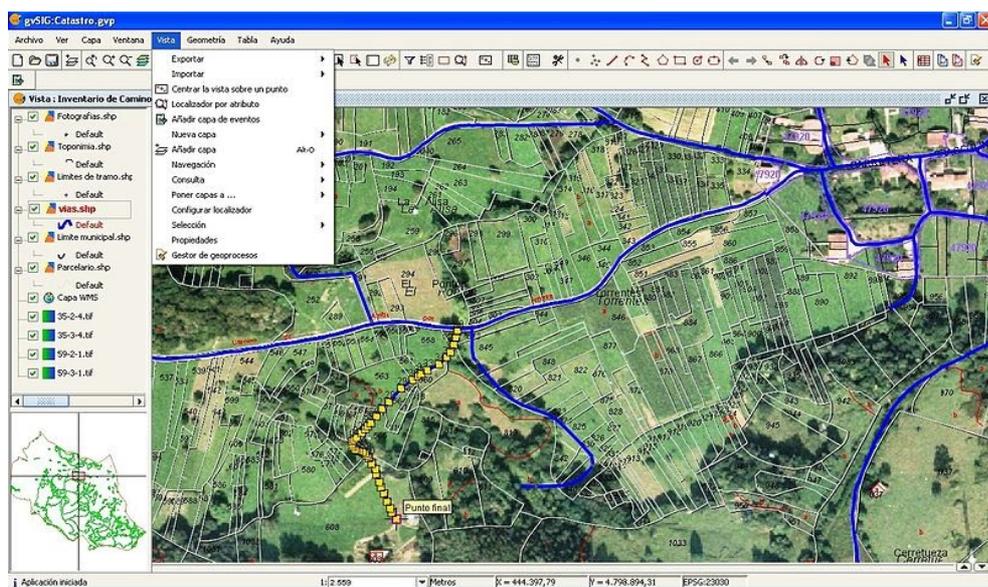


Figura 10 - Interface do gvSIG

Ele dispõe de diversas funções para aquisição, armazenamento, gerenciamento, manipulação, processamento, exibição e publicação de dados e informações geográficas. Sua

interface, apresentada na Figura 10, permite fácil manipulação de dados vetoriais e matriciais dos formatos mais comuns, inclusive suporte a padrões do OGC(WMS, WCS, WFS, KML) (GVSIG, 2013).

O gvSIG possui uma interface simples, dessa forma, usuários inexperientes na área de geoprocessamento terão facilidade na sua operacionalização.

2.2.5 Quantum GIS

Quantum GIS (QGIS, 2013) é um software *Open Source*, com interface simples e amigável, podendo ser executado pelos sistemas operacionais Windows, Apple OsX, Linux e Unix. Possui distribuição pela licença GNU Public Licence. É uma aplicação, demonstrada na Figura 11, que pode ser utilizada por todas as classes de utilizadores, dessa forma, possibilita que os mais inexperientes possam visualizar seus dados, os intermediários podem facilmente editar e criar informação num vasto número de formatos e os mais experientes podem realizar as suas análises espaciais utilizando o plugin GRASS (GRASS, 2013).

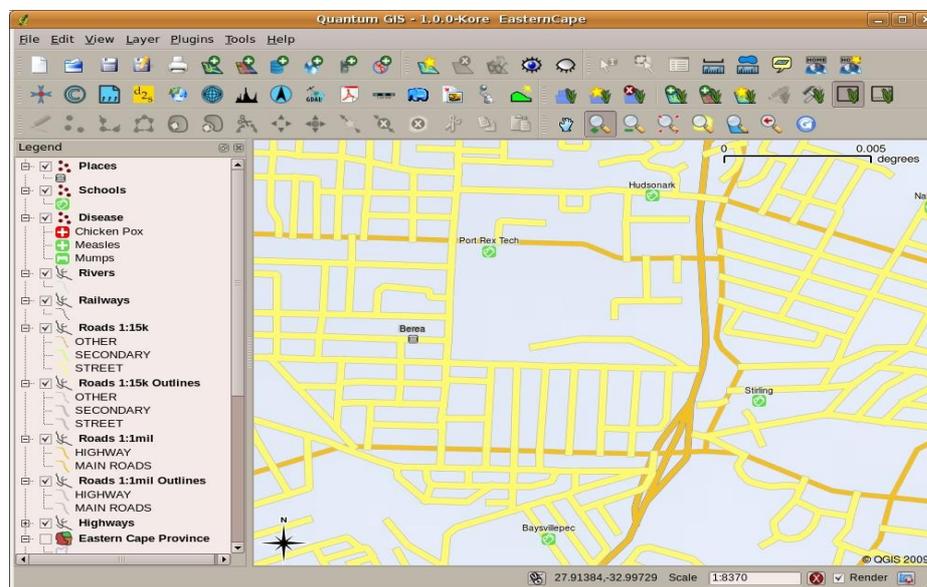


Figura 11 - Interface do Quantum GIS

Oferece diversos pontos positivos para seus utilizadores, porém, segundo Sherman (SHERMAN, 2008), este software não poderá ser considerado uma aplicação SIG em sua totalidade, pois esse recurso somente é oferecido com o uso do *plugin* GRASS.

O Quantum GIS é utilizado na visualização e manipulação de dados vetoriais(Shapefiles,

ESRI, PostGIS, MapINFO, SDTS, GML e a maioria dos formatos suportados pela biblioteca OGR) e matriciais(TIFF, ArcINFO, raster de GRASS, ERDAS, e a maioria dos formatos suportados pela biblioteca GDAL), onde se destaca o seu suporte de dados armazenados em entidades PostGIS (POSTGIS, 2013) recolhidos por GPS e vindos do software GRASS (QGIS, 2013) e SpacialLITE.

Como Geobrowser, o Quantum GIS possui as seguintes funcionalidades, segundo FAUNALIA(2013):

- Integração com o GRASS
- Integração com Python
- Integração com R (pacote de estatística)
- Extensibilidade através de *plugins*
- Ferramentas de digitalização
- Ferramentas de Geoprocessamento
- Painel de *overview*
- Marcadores espaciais
- Projeções *on de fly*
- Gestão de sistemas de coordenadas personalizadas
- Criar plugins personalizados com Python ou C++
- Criar aplicações SIG com Python, C++ com as bibliotecas de base de QGIS
- Criar layouts de impressão
- Download/Upload de dados de/para unidades de GPS
- Importar/Exportar shapefiles de/para PostGIS
- Exportar layers para MapServer
- Converter dados em formatos suportados pela biblioteca OGR
- Suporte para dados on-line, WFS e WMS

2.2.6 MapInfo

MapInfo é um aplicativo SIG de mapeamento e análise geográfica baseado na plataforma Windows voltado às áreas de marketing, vendas, logística, projetos e inteligência de negócio, permitindo visualizar facilmente a relação entre dados alfanuméricos e geográficos (MAPINFO, 2013).

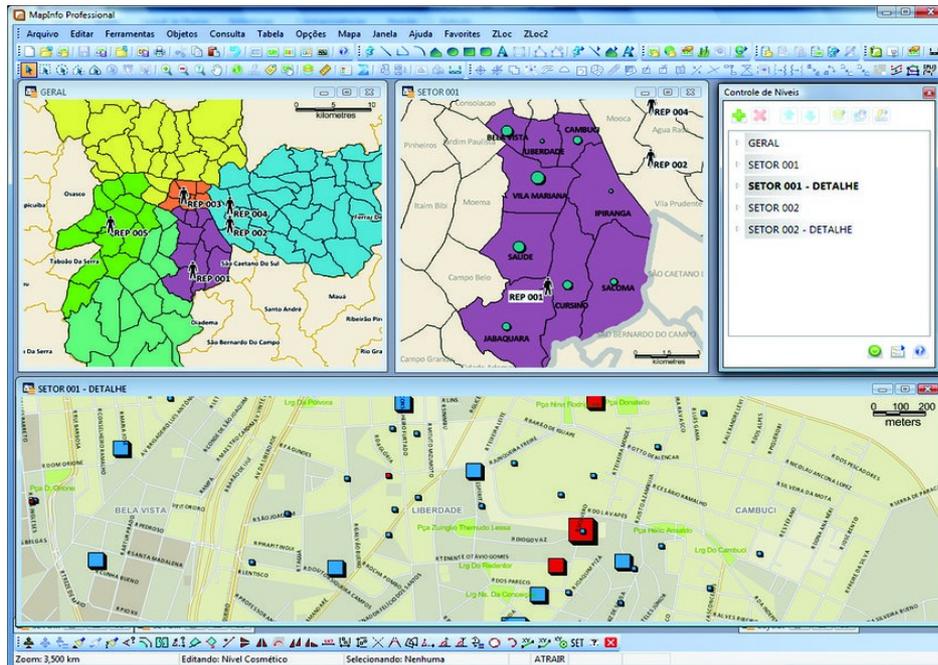


Figura 12 - Interface do MapInfo

Conforme observamos na Figura 12, a interface do MapInfo torna possível realizar análises sofisticadas e detalhadas, possibilitando a inclusão de localização geográfica em suas análises e tomadas de decisão, tendo como resultado a redução de custos, reflexo no aumento de receitas e melhoria nos serviços (MAPINFO, 2013).

2.2.7 GeoMDQL

GeoMDQL (*Geographic and Multidimensional Query Language*) é uma linguagem de consulta geográfica multidimensional específica para ambientes de DWG (*Data Warehouse Geográfico*) e SOLAP, construída a partir do projeto de defesa do doutorado de Joel da Silva na Universidade Federal de Pernambuco (SILVA, 2008).

Segundo o autor, devido às inúmeras propostas da literatura versando sobre a integração das funcionalidades e características pertinentes aos processamentos de dados analíticos (OLAP) e geográficos (SIG), este projeto de pesquisa visa integrar a sintaxe de consulta e de operadores multidimensionais e geográficos em uma única linguagem, afim de se prover um ambiente único, com capacidades de processamento geográfico e multidimensional, para dar suporte ao processo de tomada de decisões estratégicas (SILVA, 2008).

Uma das partes mais importantes desse processo de integração é a consulta de dados, a qual é demonstrada na Figura 13.

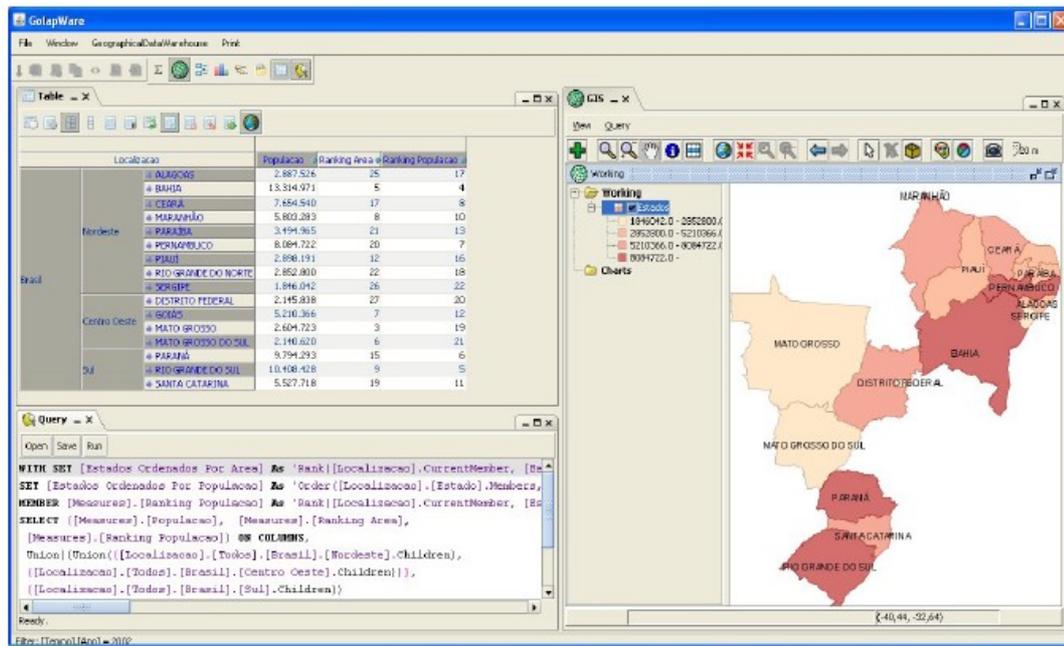


Figura 13 - Exemplo de consulta na linguagem GeoMDQL

2.2.8 Comparativo dos trabalhos relacionados em relação ao projeto de pesquisa

TerraView, Spring, i3Geo, gvSIG, Quantum GIS e MapInfo são ferramentas de SIG e apresentam excelentes capacidades, quer de edição, quer de análise espacial, devido à quantidade de recursos oferecidos por elas. São *open source* e possuem comunidades ativas, além de fóruns de discussão para auxiliar o processo de aprendizado de seus utilizadores.

A maioria possui versões para Windows e Linux e, também, compatibilidade com os principais sistemas gerenciadores de bases de dados *open source*, como PostgreSQL e MySQL. Possuem facilidade para customização conforme a necessidade do usuário.

Atualmente, de acordo com pesquisa feita por Medeiros (MEDEIROS, 2012), o Quantum GIS, possui o maior número de utilizadores. Porém, não existe integração com o módulo fiscal e tributário da empresa Digifred (DIGIFRED, 2013), que a Prefeitura Municipal de Frederico Westphalen utiliza. E, para se extrair estatísticas georreferenciadas para tomada de decisão com as ferramentas em questão, seria necessário realimentar as informações econômicas nessas ferramentas, gerando duplicidade de entrada de dados e desatualização dessas informações.

Dessa forma, o estudo em questão tem por objetivo a implementação de uma

ferramenta SIG *open source* com algumas das características apresentadas nas ferramentas anteriores, com a possibilidade de integração entre os bancos de dados Firebird (FIREBIRD, 2013) e MySQL e também com o pacote de softwares da empresa Digifred Informática Ltda., para que a atualização dos dados feita pelos setores fiscal e tributário da Prefeitura reflita em tempo real para a ferramenta de geoprocessamento. Facilitando a obtenção de informações para a tomada de decisão mais rápida e eficaz para a geração do conhecimento. A partir dessa integração, torna-se possível analisar e simular cenários que maximizam os resultados e aperfeiçoam o processo de gerenciamento dos dados do município.

2.2.9 GeoLiberty

GeoLiberty (VIANNA, 2012) é uma solução usada para auxiliar na construção de Sistemas de Informação Geográfica, a qual incorpora diversas funções básicas para a implementação de dados geográficos. Seu uso facilita a definição base de um projeto SIG, como é o caso deste, e também favorece a padronização das funcionalidades nele inseridas. Possui código aberto e extensível, dessa forma é possível adicionar funcionalidades através de novos recursos.

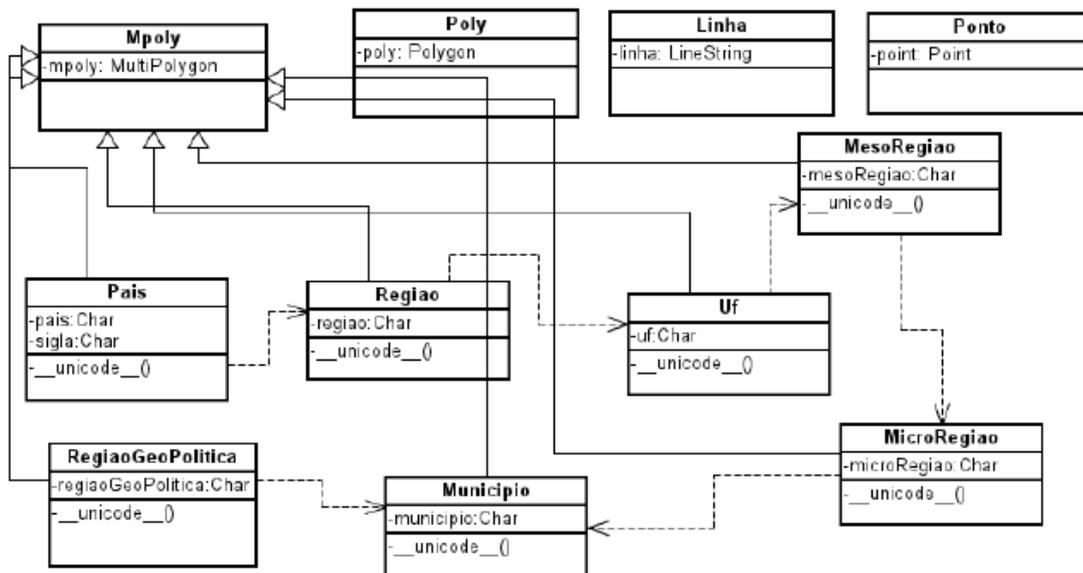


Figura 14 - Diagrama de classes do projeto GeoLiberty - Fonte: (VIANNA, 2012)

Vemos na Figura 14, o diagrama de classes do projeto GeoLiberty onde pode-se visualizar as dependências hierárquicas das classes base, onde são definidos os tipos de dados

espaciais, sendo eles: Mpoly, representada por um conjunto de polígonos; Poly, representando um polígono fechado; Linha, representada por uma sequência de pontos e Ponto, representando um ponto geográfico através de duas coordenadas geográficas, latitude e longitude.

Porém, o GeoLiberty foi projetado para integração com o banco de dados PostgreSQL e sua extensão espacial PostGIS. Dessa forma, o desafio deste estudo será a adaptação de suas funções para comunicar com o MySQL, ampliando e contribuindo com as possibilidades de conexões com os demais bancos de dados do mercado.

E, a partir do núcleo construído, serão implementadas novas funcionalidades a fim de se ampliar esse projeto e contribuir com a comunidade de software livre para a sua continuação.

2.3 Tecnologias

2.3.1 Python

Python (PYTHON, 2013) é uma linguagem de programação de altíssimo nível orientada a objeto, de tipagem dinâmica e forte, interpretada e interativa. Sua sintaxe é clara e concisa, favorecendo a legibilidade do código fonte, tornando a linguagem mais produtiva. Como dinâmica, entende-se que o tipo de uma variável é inferido pelo interpretador em tempo de execução.

É uma linguagem flexível, sendo possível adicionar recursos extras, como por exemplo, *frameworks* de terceiros. E sua escolha para fazer parte deste projeto ocorreu devido ao fato dela ser uma linguagem atual e de código aberto.

Foi criada em 1990 por Guido Van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda e tinha originalmente foco em usuários como físicos e engenheiros.

No ambiente Windows, as extensões dos arquivos são classificadas como ".py", ".pyw", ".pyc" e ".pyo". O código fonte é traduzido pelo Python em *bytecode*, ou seja, formato binário com instruções para o interpretador. Por ser multiplataforma, o *bytecode* pode ser distribuído e executado sem o código fonte.

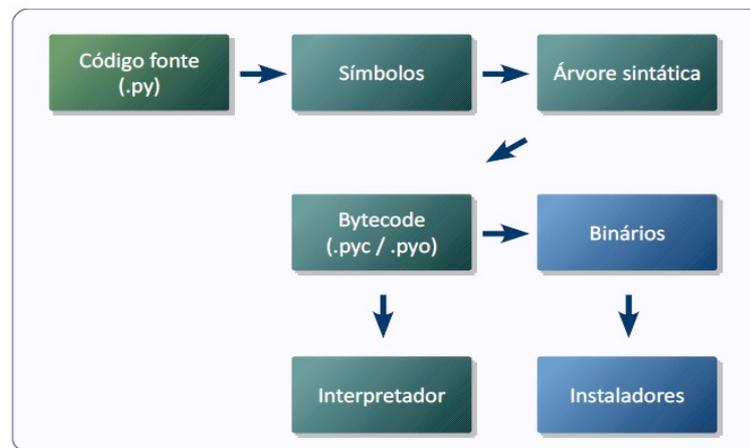


Figura 15 - Fluxo de execução do código Python

De forma geral, observamos na Figura 15, que o interpretador do Python compila o código fonte com extensão ".py" e gera o *bytecode* com extensão ".pyc", formato normal, ou ".pyo", formato otimizado. Dessa forma, numa segunda execução do código Python, o interpretador buscará automaticamente o *bytecode* gerado, economizando o tempo de carga, pois não será necessário recompilar, desde que os arquivos fontes não sejam alterados.

2.3.2 Django

Django (DJANGO, 2013) é um *framework* para desenvolvimento rápido para web, escrito em Python, que faz uso do padrão MTV (*model, template, view*), semelhante ao padrão MVC (*model, view, controller*), separando o desenvolvimento da aplicação em três camadas, o que favorece a legibilidade, organização e padronização do código fonte. Fazendo um comparativo entre o padrão MTV em relação ao MVC, temos o seguinte: *model* é o mesmo; *template* equivale ao *view* e *view* equivale ao *controller*. É um software livre e está disponível sobre a licença BSD e exige um versão do Python igual ou superior à 2.3.

Possui também a técnica de mapeamento objeto-relacional (ORM), facilitando a criação e manutenção de tabelas no banco de dados através da definição de suas classes, evitando assim fazer uso de comandos SQL para sua manipulação.

Django utiliza o princípio DRY (*Don't Repeat Yourself*), permitindo ao desenvolvedor reaproveitar ao máximo seu código, evitando repetição.

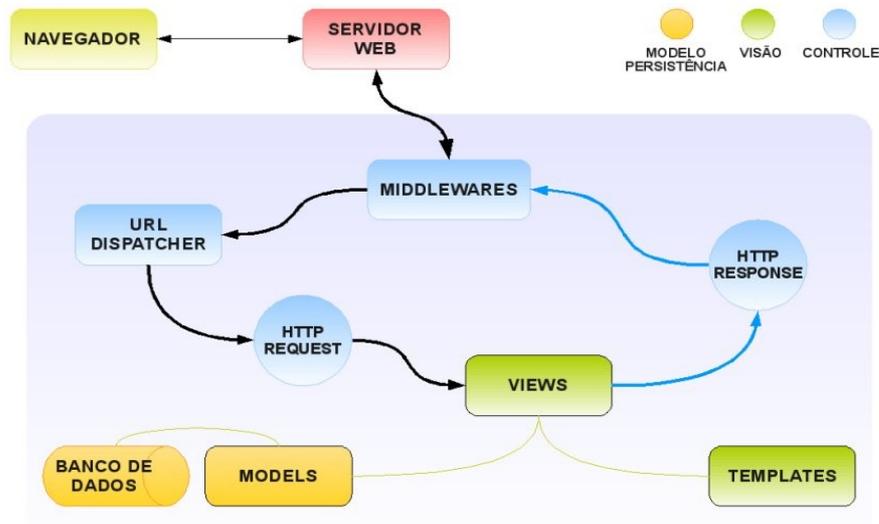


Figura 16 - Fluxo de processo do Django - Fonte: (BRANDÃO, 2009)

Na Figura 16, vemos o fluxo de processo das informações que o Django implementa e gerencia. A partir de uma requisição encaminhada pelo usuário, o servidor web encaminha esse pedido (*HttpRequest*), contendo diversas informações, entre elas, a *URL*, composta pelo protocolo, o domínio, porta, caminho e demais parâmetros.

Quando esse pedido chega até a função *view*, ela analisa e processa a solicitação e devolve o resultado (*HttpResponse*) de volta ao usuário, normalmente no formato de uma página web.

Na camada *Model* especificamos as definições dos dados, como devem ser armazenados e tratados. Em si, um *model* é uma classe Python que define uma tabela, seus atributos, valores padrões e a obrigatoriedade destes. Através da técnica de modelagem de dados ORM (*Object Role Modeling*) (ORM, 2013), evita-se manipular os dados através da criação de consultas SQL (SQL, 2013), pois essa tarefa é realizada a partir da instanciação da classe definida. Essa técnica facilita e agiliza a criação de tabelas, pois sua configuração torna-se independente do banco de dados a ser utilizado.

Na camada *View* são definidas as funções que recebem requisições e retornam respostas ao solicitante. Também podem ser utilizadas para a validação de formulários, envio de mensagens, autenticações, etc.

Já na camada *Controller*, ou *template*, temos o *handler*, os *middlewares* e *URL dispatcher*, ou seja, os agentes intermediadores que fazem parte da organização e distribuição dos dados.

2.3.3 GeoDjango

Geodjango (GEODJANGO, 2013) é um pacote de recursos usados na manipulação de dados espaciais com o *framework* Django. Também escrito em Python, oferece uma solução completa para manipulação de dados geoespaciais.



Figura 17 - Arquitetura de uma aplicação com o GeoDjango

Como podemos observar na Figura 17, são diversas as tecnologias que tem possibilidade de interagir com o GeoDjango. Para a construção do sistema de informação geográfica proposto neste projeto levou-se em consideração uma arquitetura de sistemas compatíveis com o ambiente do cliente adotado no estudo de caso, como por exemplo, a integração com o banco de dados MySQL. E o GeoDjango foi uma das escolhas deste projeto devido ao fato dele possuir integração nativa com esse banco de dados.

2.3.4 MySQL

MySQL (MYSQL, 2013) é um sistema de gerenciamento de banco de dados que utiliza a linguagem SQL como interface de comunicação. Ele está sob a licença *General Public Licence* (GPL) podendo ser, portanto, distribuído sem quaisquer custos com licença. Criada na Suécia na década de 80, atualmente encontra-se entre os bancos de dados mais populares do mundo e sua escolha neste projeto deve-se ao fato de que o órgão a ser aplicado o estudo de caso já o utiliza, aproveitando assim, toda sua estrutura de serviço já disponível.

A implementação das funções geométricas em MySQL não possui todas as funções necessárias para tratamento de dados espaciais como em outros bancos de dados (por

exemplo, PostgreSQL e Oracle *Spatial*), mas para os objetivos deste projeto, é suficiente as operações que ele oferece.

As extensões espaciais que o MySQL implementa seguem as especificações do OGC (*Open Gis Consortium*) (OGC, 2013), o qual conduz a padronização e integração de dados geográficos. Segundo o guia de referência do MySQL, a partir da versão 4.1, o MySQL introduziu extensões espaciais, composta por um conjunto de componentes diversificados disponíveis em uma biblioteca com recursos que oferecem análise, criação e armazenamento de dados geográficos(entidade, espaço ou localização definida). Antes da versão 5.0.16, esses recursos somente estavam disponíveis para as tabelas no formato MyISAM, e partir dessa versão, os formatos InnoDB, NDB, BDB e ARCHIVE também passaram a oferecer suporte a esses recursos espaciais (MYSQL, 2010)

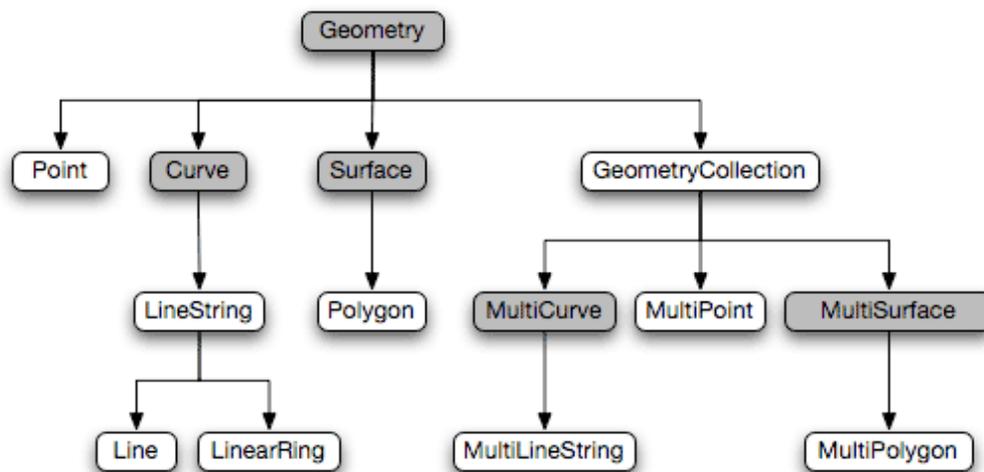


Figura 18 - Tipos de dados espaciais do MySQL

O padrão utilizado para o armazenamento de dados espaciais é baseado na hierarquia de objetos definido pelo Modelo Geométrico OpenGIS, tendo os seguintes tipos de geometria: Geometry, Point, LineString e Polygon, representados na Figura 18. Diferentemente dos demais, o Geometry permite armazenar qualquer tipo de geometria.

- Geometry: Tipo geométrico que permite armazenar pontos, linhas e polígonos.
- Point: Um ponto em um sistema de coordenadas; sem dimensão.
- LineString: Um ou mais segmentos lineares unindo dois pontos; unidimensional.
- Polygon: Um LineString fechado, onde o fim coincide com o início; bidimensional.
- MultiPoint: Um ou mais geometrias do tipo Point.

- MultiLineString: Uma ou mais geometrias do tipo LineString.
- MultiPolygon: Uma ou mais geometrias do tipo Polygon.
- GeometryCollection: Permite armazenar qualquer coleção de objetos de qualquer tipo. Porém, os demais tipos de coleções, como o MultiPoint, MultiLineString e MultiPolygon, só podem armazenar coleções de seus tipos específicos.

Ainda, segundo o Guia MySQL (MYSQL, 2010), há dois formatos de dados espaciais padrões para representar objetos geometry em consultas, são eles: Formato Well-Known Text(WKT) e Formato Well-Know Binary(WKB).

Exemplos de representação de dados geométricos no padrão WKT:

- Point: a representação das coordenadas não possui separação por vírgulas: POINT(15 20)
- LineString: uma representação com quatro pontos: LINESTRING(0 0, 10 10, 20 25, 50 60)
- Polygon: representação com dois anéis, um interior e outro exterior: POLYGON((0 0, 10 0, 10 10, 0 10, 00), 5 5, 7 5, 7 7, 5 7, 5 5))

2.3.5 OpenLayers



Figura 19 - Exemplo de uso do OpenLayers

OpenLayers (OPENLAYERS, 2013) é uma biblioteca Javascript de código aberto e de

uso livre para a criação de mapas interativos na *Web*. A partir de 2007 seu projeto faz parte da OSGeo (Open Source Geospatial Foundation) (OSGEO, 2013).

Conforme observamos na Figura 19, a biblioteca OpenLayers fornece uma API para auxiliar na criação de aplicações geográficas baseadas na web semelhantes ao Google Maps. As funcionalidades dessa ferramenta são vastas, possibilitando a customização de camadas, controles e eventos nos mapas. Segundo Hazzard (HAZZARD, 2011), uma camada é uma maneira de mostrar múltiplos níveis de informação de maneira independente uns dos outros. Os controles são responsáveis por permitir que o usuário interaja com o mapa, além de exibir informações adicionais sobre o estado atual do mapa. Já os eventos, podem ser definidos como alguma ação que está ocorrendo, como, por exemplo, um clique ou uma tecla pressionada.

Pode-se, com os recursos oferecidos pela ferramenta, combinar diversas camadas e ela também possui suporte com os padrões da OGC (OGC, 2013) como o *Web Map Service* (WMF), que provê imagens de mapas, e o *Web Feature Service* (WFS), que recupera ou altera descrições dos dados geográfico, bem como Google Maps, Bing Maps, OpenStreetMap, GeoServer, MapServer, GeoRSS e outros mais.

Hazzard (HAZZARD, 2011) destaca alguns pontos favoráveis da biblioteca OpenLayers em relação à API do Google Maps (GOOGLE MAPS API, 2013), sendo elas: Caso seja necessário, o controle do back-end é transferido integralmente para o desenvolvedor; Sobreposição de camadas para exibição de dados espaciais mais complexos, inclusive fazendo uso dos recursos do próprio Google Maps; Customização mais flexível, sem obstáculos das barreiras comerciais impostas pela solução Google.

2.3.6 Apache

O servidor Apache é um principais servidores web livres do mercado, disponibilizado em versões para Windows, Novell Netware, OS/2, Linux, Unix, etc, envolvendo tecnologias de transmissão via web, processamento de dados e execução de aplicativos distribuídos (APACHE, 2013).

Uma de suas principais características é a sua flexibilidade nos módulos utilizados nos programas, ou seja, ele possui a capacidade de se adaptar a diversos tipos de páginas da web. A sua portabilidade é praticamente absoluta, pois roda a maioria dos sistemas de hospedagem

de dados (DICAS DE HOSPEDAGEM, 2011).

Porém, para facilitar e agilizar a instalação e configuração dos softwares MySQL e Apache para a plataforma Windows, foi utilizado o pacote assistente de instalação WampServer (WAMP SERVER, 2013).

O Apache tem como base o servidor web NCSA 1.3 (*National Center of Supercomputing Applications*), que foi desenvolvido por Rob McCool. Quando Rob deixou o NCSA, o desenvolvimento foi interrompido, assim muitos desenvolvedores buscaram personalizar sua própria versão do NCSA ou adicionar mais características para atender as suas necessidades. Neste momento começa a história do Apache com *Brian Behlendorf* e *Cliff Skolnick* abrindo uma lista de discussão para interessados no desenvolvimento, conseguindo espaço em um servidor doado pela *HotWired* e trocando patches corrigindo problemas, adicionando recursos e discutindo ideias com outros desenvolvedores e hackers interessados neste projeto (SILVA, 2010).

Abaixo estão algumas características que fazem esse servidor web o preferido entre os administradores de sistemas (SILVA, 2010):

- Possui suporte a *scripts cgi* usando linguagens como *Perl, PHP, Shell Script, ASP, etc.*
- Suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/diretório acessado no servidor.
- Autenticação requerendo um nome de usuário e senha válidos para acesso a alguma página/sub-diretório/arquivo (suportando criptografia via Crypto e MD5).
- Negociação de conteúdo, permitindo a exibição da página Web no idioma requisitado pelo Cliente Navegador.
- Suporte a tipos mime.
- Personalização de logs.
- Mensagens de erro.
- Suporte a virtual hosting (é possível servir 2 ou mais páginas com endereços/ portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço).
- Suporte a IP virtual hosting.
- Suporte a name virtual hosting.
- Suporte a servidor Proxy ftp e http, com limite de acesso, caching (todas

flexivelmente configuráveis).

- Suporte a proxy e redirecionamentos baseados em URLs para endereços Internos.
- Suporte a criptografia via SSL, Certificados digitais
- Módulos DSO (Dynamic Shared Objects) permitem adicionar/remover funcionalidades e recursos sem necessidade de recompilação do programa.

CAPÍTULO 3 - DESENVOLVIMENTO DO FRAMEWORK MAPGEOCITY

3.1 Definição

Visando estender as funcionalidades do projeto GeoLiberty, projeto base, foi desenvolvido um estudo e implementado um *framework* direcionado à área de georreferenciamento de imóveis urbanos. Este *framework* foi nomeado de MapGeoCity, referindo-se ao mapeamento do georreferenciamento da cidade.

O MapGeoCity é um *framework*, escrito em Python, orientado a objetos, utilizado para abstrair a complexidade e padronizar o processo de construção de um SIG para ser utilizado no auxílio de informações georreferenciadas do cadastro de imóveis urbanos. Dessa forma, toda aplicação que fizer uso deste *framework* irá herdar toda a estrutura definida por esta tecnologia.

3.2 Arquitetura do framework proposto

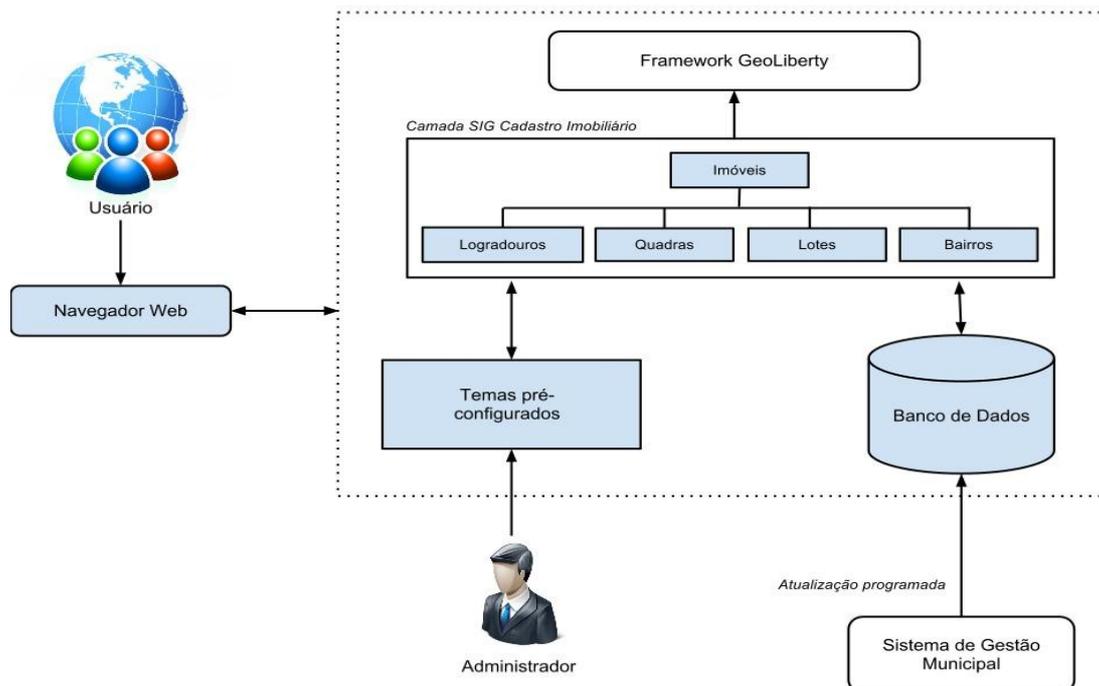


Figura 20 - Fluxograma da arquitetura proposta do framework MapGeoCity

Na Figura 20, é ilustrado o fluxograma das principais etapas para a construção do *framework* proposto neste estudo. O fluxo inicia-se através da disponibilização de uma página web, onde o usuário especifica o seu pedido. Por exemplo, ele necessita identificar os imóveis situados no centro da cidade com área de terreno superior a 1.000m². O *framework* trata esse pedido e filtra essas informações através da camada SIG implementada na sua estrutura de classes e comunica-se com a base de dados MySQL através do GeoDjango.

A estrutura da qual o *framework* MapGeoCity herda, provê toda a base para a representação geográfica do modelo de dados. A partir dessa estrutura foi implementado uma camada de classes específicas para abstrair a sistemática do cadastro de imóveis urbanos.

E, para integrar os dados como um todo, o Sistema de Gestão Municipal fornece informações em tempo real para a base de dados SIG, mantendo atualizada esta e possibilitando o acompanhamento visual no mapa de qualquer alteração efetuada no modelo de dados alfanumérico.

3.2 Modelo de Dados

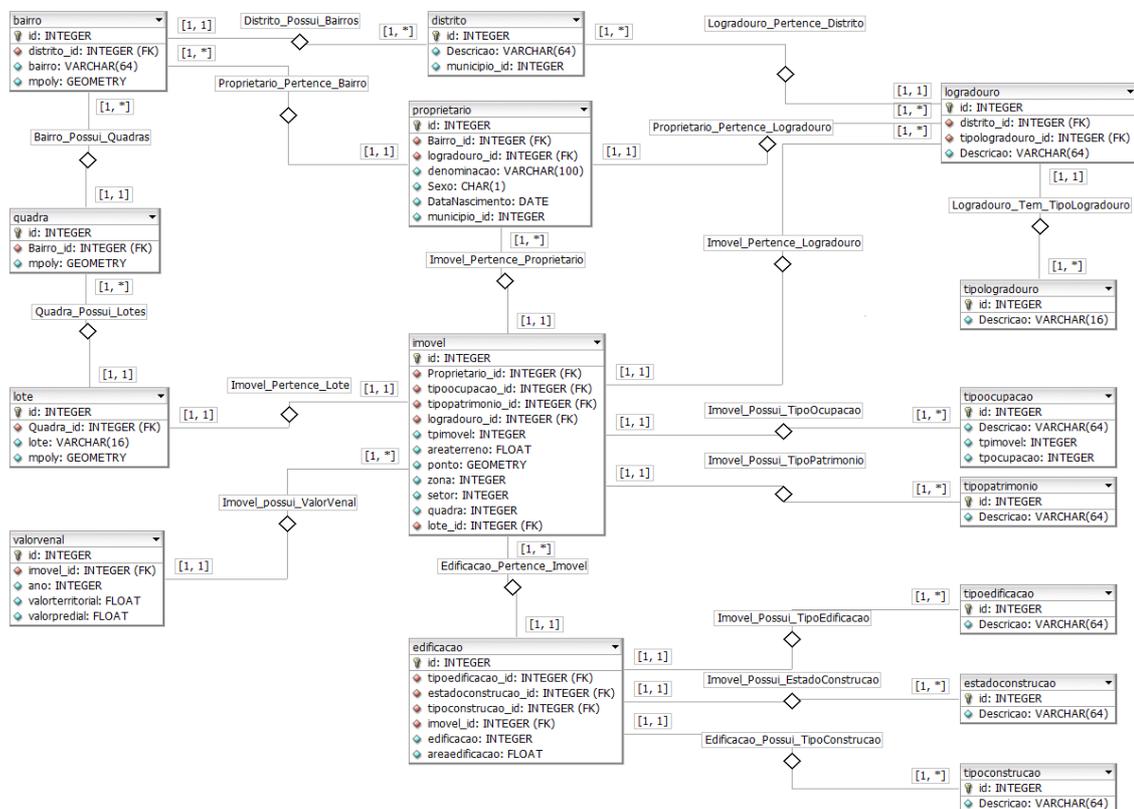


Figura 21 - Visão geral do diagrama estrutural do framework

Na Figura 21, temos a visão geral do diagrama estrutural do framework deste projeto, representando as características básicas das informações do setor imobiliário do município.

O diagrama é composto pelas seguintes classes:

- Distrito: Representa as divisões administrativas da Cidade. No caso do município de Frederico Westphalen, eles são representados por: Frederico Westphalen, Castelinho e Osvaldo Cruz.

- Bairro: Representa as unidades regionais da Cidade: Exemplo: Centro, Ipiranga, Itapagé, Fátima, Santo Antônio, etc

- Quadra: Representa o agrupamento de lotes

- Lote: Representa os lotes urbanos

- Imóvel: Representa os imóveis urbanos estabelecidos pelas leis municipais. Esta tabela é a base de referência para as pesquisas geográficas.

- Edificação: contendo as características dos imóveis prediais.

- Valor Venal: contendo os valores venais territoriais e prediais dos imóveis

- Proprietário: Representa o dono do Imóvel

- Estado Civil: Representa o Estado Civil do Proprietário. Exemplo: Solteiro, Casado, etc.

- Tipo de Logradouro: Representa os Tipo de Logradouro, sendo ele: Rua, Avenida, etc.

- Logradouro: Representa a descrição do Logradouro. Exemplo: Luiz Milani, Maurício Cardoso, etc. A definição completa do logradouro é definida pela junção do Tipo de Logradouro com a Descrição do Logradouro.

- Tipo de Ocupação: Representa o Tipo de Ocupação do Imóvel, sendo ele: Residencial, Comercial, Industrial, etc

- Tipo de Pavimento: Representa o Tipo de Pavimento do Logradouro do Imóvel: Paralelepípedo, Asfalto, Terra, Sem Calçamento, etc.

- Tipo de Edificação: Representa os Tipos de Edificação, sendo eles: Casa, Prédio, Apartamento, etc.

- Tipo de Construção: Representa os Tipos de Construção, sendo eles: Alvenaria, Mista, Madeira, etc.

- Estado da Construção: Representa os Estados da Construção, sendo eles: Bom, Muito Bom, etc.

3.3 Preparação do ambiente de trabalho

Para a implementação do *framework* MapGeoCity foi adotado as seguintes ferramentas e aplicativos.

Ambiente operacional: Windows 7, 64 bits

Ambiente de desenvolvimento:

- WampServer, contendo Apache 2.4.2, PHP e banco de dados MySQL 5.5.24.

Juntamente com o pacote é instalado, também, o PhpMyAdmin, utilizado para o gerenciamento do banco de dados MySQL.

- Python, versão 2.7.3

- MySQLdb, versão 1.2.4, módulo de suporte do MySQL no python

- Django, versão 1.5.2

- OsGeo4W, pacote de distribuição de ferramentas para tratamento de dados geoespaciais no ambiente windows

- Editor de código fonte python: Sublime Text 2

Ambiente do projeto base geoliberty:

- PostgreSQL, versão 9.1

- PostGIS 1.5.5, extensão espacial do Postgres

- Módulos de integração do python com a extensão espacial PostGIS:

 - argparse, versão 1.2.1

 - psycopg2, versão 2.5.1

 - wsgiref, versão 0.1.2

3.4 Etapas de migração do *framework* Geoliberty

Para atender o projeto de pesquisa em questão, foram feitos diversos ajustes no projeto base para que ele se adequasse aos requisitos definidos no escopo deste trabalho.

O primeiro deles foi migrar os dados do Geoliberty da base de dados PostgreSQL para o MySQL. Para isso foi necessário realizar as seguintes etapas:

3.4.1 Criação e configuração da Base de Dados MySQL

A criação do banco de dados no MySQL foi feita através da ferramenta phpMyAdmin. Após isso, foi configurado os parâmetros de conexão do projeto Geoliberty do banco Postgres para o MySQL especificando a tipo de tabela MyISAM, pois o tipo InnoDB não possui indexação espacial. Para isso foi necessário modificar o arquivo de configuração do projeto (settings.py), conforme é ilustrado nas Figura 22 e 23.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.contrib.gis.db.backends.postgis',
        'NAME': 'geo_liberty',
        'USER': 'postgres',
        'PASSWORD': 'admin',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Figura 22 - Parâmetros de conexão com o Postgres

```
DATABASES = {
    'default': {
        'ENGINE': 'django.contrib.gis.db.backends.mysql',
        'NAME': 'geoliberty',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': '127.0.0.1',
        'PORT': '3306',
        'OPTIONS': { 'init_command': 'SET storage_engine=MYISAM;' }
    }
}
```

Figura 23 - Parâmetros de conexão com o MySQL

3.4.2 Criação do framework MapGeoCity

Dentro da pasta do projeto, foi criado o módulo MapGeoCity, através do comando: "python manage.py startapp MapGeoCity". E, no arquivo models.py, desta aplicação, foi adicionado as classes do *framework* fazendo uso da extensão dos dados espaciais do projeto base do Geoliberty, de acordo com a Figura 24 a classe herdada foi a Ponto.

```

class Imovel(Ponto):

    TIPO_IMOVEL = (
        ('1', 'Territorial'),
        ('2', 'Predial')
    )

    proprietario = models.ForeignKey(Proprietario, verbose_name='Proprietário')
    municipio = models.ForeignKey(Municipio, default=4308508)
    distrito = models.ForeignKey(Distrito, verbose_name='Distrito')
    bairro = models.ForeignKey(Bairro, verbose_name='Bairro')
    logradouro = models.ForeignKey(Logradouro, verbose_name='Logradouro')
    nrologradouro = models.CharField('N° do Logradouro', max_length=8)
    tpimovel = models.IntegerField('Tipo de Imóvel', max_length=1, choices=TIPO_IMOVEL)
    tpocupacao = models.ForeignKey(TipoOcupacao, verbose_name='Tipo de Ocupação')
    tppatrimonio = models.ForeignKey(TipoPatrimonio, verbose_name='Tipo de Patrimônio')
    zona = models.IntegerField('Zona')
    setor = models.IntegerField('Setor')
    quadra = models.IntegerField('Quadra')
    lote = models.CharField('Lote', max_length=16)
    areaterreno = models.FloatField('Área do Terreno')

class Meta:
    verbose_name = 'Imovel'
    verbose_name_plural = 'Imóveis'

    def __unicode__(self):
        return str(self.id)

    def get_absolute_url(self):
        return 'imovel/%1/' % self.id

```

Figura 24 - Classe Imóvel derivando a classe Ponto do GeoLiberty

Após a definição de todas as classes necessárias do projeto, foi configurado a aplicação "mapgeocity" no arquivo "settings.py" do projeto, adicionando-a em INSTALLED_APPS. Isso se fez necessário para que o sincronizador do Django pudesse vincular a aplicação com o projeto.

Para que o Django criasse automaticamente o esquema do banco de dados a partir das classes definidas no models.py e também a API para acesso administrativo da aplicação, foi necessário executar o sincronizador na pasta do projeto, através do seguinte comando: "python manage.py syncdb". Com isso, toda a estrutura do banco de dados foi criada automaticamente, simplificando as tarefas e a organização do estudo.

Para comprovar a eficiência do ORM do Django, vemos na Figura 25 a estrutura da tabela "imovel" gerada pelo sincronizador.

#	Nome	Tipo de dados	Tamanho/It...	Padrão
1	id	INT	11	AUTO_INCREMENT
2	ponto	POINT		Nenhum padrão
3	proprietario_id	INT	11	Nenhum padrão
4	municipio_id	INT	11	Nenhum padrão
5	distrito_id	INT	11	Nenhum padrão
6	bairro_id	INT	11	Nenhum padrão
7	logradouro_id	INT	11	Nenhum padrão
8	nrologradouro	VARCHAR	8	Nenhum padrão
9	tpimovel	INT	11	Nenhum padrão
10	tpocupacao_id	INT	11	Nenhum padrão
11	tppatrimonio_id	INT	11	Nenhum padrão
12	zona	INT	11	Nenhum padrão
13	setor	INT	11	Nenhum padrão
14	quadra	INT	11	Nenhum padrão
15	lote	VARCHAR	16	Nenhum padrão
16	areaterreno	DOUBLE		Nenhum padrão

Figura 25 - Estrutura da tabela Imóvel

Na Figura 26, vemos a estrutura completa, no banco MySQL, de todas as tabelas do *framework* MapGeoCity e também as tabelas herdadas do projeto GeoLiberty, assim como, as tabelas administrativas do Django.

<input type="checkbox"/> auth_group	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	0 MyISAM utf8_general_ci	2 KB
<input type="checkbox"/> auth_group_permissions	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	0 MyISAM utf8_general_ci	1 KB
<input type="checkbox"/> auth_permission	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	87 MyISAM utf8_general_ci	10.5 KB
<input type="checkbox"/> auth_user	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	1 MyISAM utf8_general_ci	3.1 KB
<input type="checkbox"/> auth_user_groups	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	0 MyISAM utf8_general_ci	1 KB
<input type="checkbox"/> auth_user_user_permissions	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	0 MyISAM utf8_general_ci	1 KB
<input type="checkbox"/> django_admin_log	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	0 MyISAM utf8_general_ci	1 KB
<input type="checkbox"/> django_content_type	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	29 MyISAM utf8_general_ci	9.2 KB
<input type="checkbox"/> django_session	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	5 MyISAM utf8_general_ci	4.2 KB
<input type="checkbox"/> django_site	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	1 MyISAM utf8_general_ci	2 KB
<input type="checkbox"/> geoliberty_mesoregiao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	137 MyISAM utf8_general_ci	2.2 MB
<input type="checkbox"/> geoliberty_microregiao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	560 MyISAM utf8_general_ci	4 MB
<input type="checkbox"/> geoliberty_municipio	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	5,567 MyISAM utf8_general_ci	10.6 MB
<input type="checkbox"/> geoliberty_pais	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	1 MyISAM utf8_general_ci	585.6 KB
<input type="checkbox"/> geoliberty_regiao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	5 MyISAM utf8_general_ci	536.5 KB
<input type="checkbox"/> geoliberty_regiaogeopolitica	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	0 MyISAM utf8_general_ci	1 KB
<input type="checkbox"/> geoliberty_regiaogeopolitica_municipios	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	0 MyISAM utf8_general_ci	1 KB
<input type="checkbox"/> geoliberty_uf	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	27 MyISAM utf8_general_ci	336.5 KB
<input type="checkbox"/> mapgeocity_bairro	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	15 MyISAM utf8_general_ci	3.4 KB
<input type="checkbox"/> mapgeocity_distrito	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	1 MyISAM utf8_general_ci	3 KB
<input type="checkbox"/> mapgeocity_edificacao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	7,603 MyISAM utf8_general_ci	673 KB
<input type="checkbox"/> mapgeocity_estadoconstrucao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	6 MyISAM utf8_general_ci	2.1 KB
<input type="checkbox"/> mapgeocity_imovel	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	6,486 MyISAM utf8_general_ci	1.7 MB
<input type="checkbox"/> mapgeocity_logradouro	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	135 MyISAM utf8_general_ci	14 KB
<input type="checkbox"/> mapgeocity_lote	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	2,989 MyISAM utf8_general_ci	128.4 KB
<input type="checkbox"/> mapgeocity_proprietario	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	3,460 MyISAM utf8_general_ci	206 KB
<input type="checkbox"/> mapgeocity_quadra	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	326 MyISAM utf8_general_ci	17.4 KB
<input type="checkbox"/> mapgeocity_tipoconstrucao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	3 MyISAM utf8_general_ci	2.1 KB
<input type="checkbox"/> mapgeocity_tipoedificacao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	14 MyISAM utf8_general_ci	2.3 KB
<input type="checkbox"/> mapgeocity_tipologradouro	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	7 MyISAM utf8_general_ci	2.1 KB
<input type="checkbox"/> mapgeocity_tipoocupacao	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	16 MyISAM utf8_general_ci	2.5 KB
<input type="checkbox"/> mapgeocity_tipopatrimonio	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	6 MyISAM utf8_general_ci	2.1 KB
<input type="checkbox"/> mapgeocity_valorvenal	Visualizar	Estrutura	Procurar	Inserir	Limpar	Eliminar	62,395 MyISAM utf8_general_ci	3.1 MB

Figura 26 - Estrutura completa do framework na base de dados MySQL

Vamos na Figura 27, outra modificação que se fez necessária no projeto, devido ao processo de integração com a base de dados MySQL, foi adequar as classes base do projeto GeoLiberty, atribuindo-as o identificador de referencia espacial (srid = 4326).

```
class Mpoly(models.Model):

    mpoly = models.MultiPolygonField(srid=4326)
    objects = models.GeoManager()

    class Meta:
        abstract = True
```

Figura 27 - Classe Mpoly do projeto GeoLiberty

Também foi necessário adequar as *views* do projeto para adequação com a base de dados MySQL. Foi substituída a função de conversão do sistema de coordenadas espaciais, implementado pelo módulo "GDAL" de dados armazenados na base PostgreSQL, visualizado na Figura 28, pelo módulo "GEOS" de dados armazenados na base MySQL, ilustrado na Figura 29.

```
def Mapa_Regiao(request, id_regiao):

    obj = Regiao.objects.filter(id = id_regiao)
    ct = CoordTransform(
        SpatialReference('EPSG:4326'),
        SpatialReference('EPSG:900913'))

    nome = ''
    for feat in obj:
        nome = feat.get_name()

    dados = []
    for p in obj:
        aux = p.mpoly
        aux.transform(ct)
        dados.append(aux)

    return render_to_response('mapa.html',
        RequestContext(request, {'dados': dados,
                                'nome': nome}))
```

Figura 28 - Transformação da projeção espacial via módulo GDAL

```

def Mapa_Regiao(request,id_regiao):

    obj = Regiao.objects.filter(id = id_regiao)

    nome = ''
    for feat in obj:
        nome = feat.get_name()

    dados = []
    for p in obj:
        dados = GEOSGeometry( force_str(p.mpoly) )

    return render_to_response('mapa.html',
                               RequestContext(request,{'dados':dados,
                                                       'nome':nome}))

```

Figura 29 - Transformação da projeção espacial via módulo GEOS

Os *templates*, responsáveis por representar a camada final de exibição dos dados, também sofreram modificação. Foi necessário, também, forçar a transformação da projeção, para que a API do OpenLayers pudesse exibir corretamente as coordenadas espaciais, conforme podemos observar nas Figuras 30 e 31.

```

var geojson_format = new OpenLayers.Format.GeoJSON();

```

Figura 30 - Tratamento com os dados do PostGIS

```

var geojson_format = new OpenLayers.Format.GeoJSON({
    'internalProjection': new OpenLayers.Projection("EPSG:900913"),
    'externalProjection': new OpenLayers.Projection("EPSG:4326")
});

```

Figura 31 - Tratamento com os dados do MySQL

3.4.3 Exportação dos dados do Postgres para o MySQL

Para efetuar o processo de transferência dos dados entre as bases foi necessário executar as seguintes etapas:

- Exportação dos dados do Postgres: foi utilizado o comando "python manage.py dumpdata geoliberty > file.json". Esse comando deve ser executado dentro da pasta do projeto e com a configuração do settings.py apontando para a base do Postgres.

- Estender a cache de transferência de pacotes do servidor MySQL: através do WampServer, deve-se acessar o serviço MySQL e abrir o arquivo my.ini, substituir o valor da variável "max_allowed_packet" para 1060M. Isso se faz necessário, porque o volume de dados é grande e o servidor não suporta o processo de transferência pela configuração básica que ele possui.

- Importação dos dados para o MySQL: foi utilizado o comando "python manage.py

loaddata file.json". Esse comando deve ser executado dentro da pasta do projeto e com a configuração do settings.py apontando para a base do MySQL.

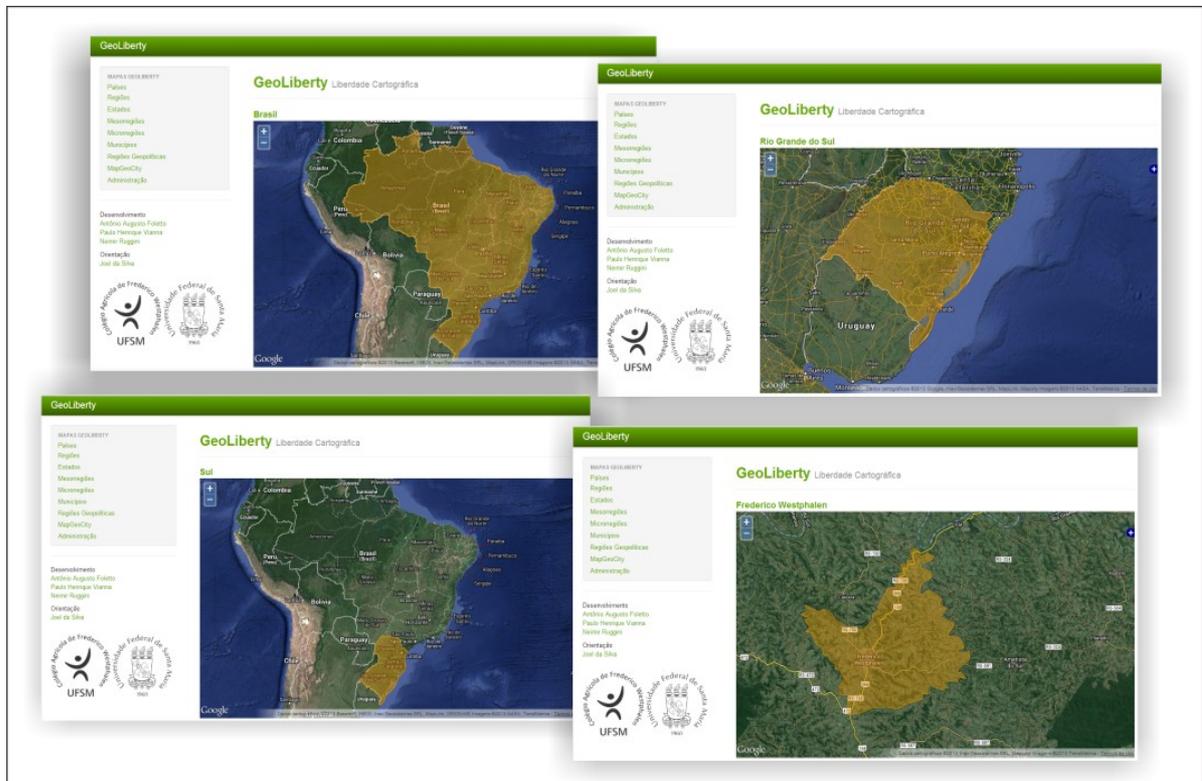


Figura 32 - Amostragem do GeoLiberty em execução a partir da base MySQL

Após o processo de transferência dos dados, podemos observar na Figura 32, algumas imagens do GeoLiberty acessando os dados da base MySQL. Dessa forma, cumpre-se o requisito inicial da proposta desta pesquisa que é a adequação do GeoLiberty com a base de dados MySQL.

3.5 Módulo de Pesquisa das Informações

Ao acessar o módulo MapGeoCity no menu do GeoLiberty, é apresentada a tela inicial do *framework*, demonstrada na Figura 33. Porém, o acesso a esta parte do projeto é controlada via autenticação de usuário e senha, disponibilizados pelo autor deste estudo.

Para o escopo deste estudo, foram definidas as seguintes funcionalidades a partir do menu do aplicativo: Início, Filtros de Pesquisa, Mapa dos Bairros, Administração e Logout.



Figura 33 - Tela de apresentação do MapGeoCity

- Início: cujo objetivo é mostrar o menu inicial do *framework* MapGeoCity, exibido acima, contendo um resumo dos objetivos deste projeto.
- Logout: encerra a sessão atual e volta para a área de login.
- Administração: esta área é restrita e seu acesso está vinculado com o supervisor do sistema. Por ser um dos principais recursos do Django, nela é possível efetuar a manutenção da aplicação através de um painel administrativo, conforme observado na Figura 34.



Figura 34 - Área administrativa do MapGeoCity

Nesta área é possível efetuar a manutenção das tabelas do *framework* de forma prática, sem a necessidade de criar os formulários para tal, pois eles são criados automaticamente pelo Django, conforme visualizado na Figura 35.



Figura 35 - Manutenção da tabela Tipo de Construção

- Filtros de Pesquisa: nessa área é possível filtrar diversas opções pra facilitar a conferência das informações a fim de direcionar o utilizador para uma amostragem mais específica dos dados a serem pesquisados, conforme podemos observar na Figura 36.

Figura 36 - Área de pesquisa do framework

Este item do menu detém as principais funcionalidades do *framework*, pois é a partir dele que é exibido o resultado da integração entre o Sistema de Gestão Municipal e as ferramentas SIG.

A partir dos critérios de pesquisa é possível determinar, por exemplo, se determinado imóvel encontra-se com a situação cadastral irregular ou determinar quais imóveis possuem área de terreno dentro de uma faixa de metragem, à uma certa distância da Prefeitura Municipal. A qual foi adotada como ponto de referência para este estudo, demonstrado na Figura 37.



Relação de Imóveis

(Total: 13 imóveis)
 Filtro: - Logradouro: Arthur Milani - Bairro: Centro - Área Mínima do Terreno: 70 m² - Área Máxima do Terreno: 300 m² - Raio de 500 metros de distância da Prefeitura Municipal

Figura 37 - Resultado do filtro de pesquisa

Na figura acima, foi marcada a opção "Exibir mapa do Bairro" na tela de pesquisa. Dessa forma, foi apresentado também no mapa, a camada do bairro Centro e sobrepondo a esta camada, estão os marcadores em vermelho que representam os imóveis enquadrados no filtro de pesquisa.

Cada marcador possui informações adicionais sobre as características dos imóveis, georreferenciados no mapa, fornecendo dados relevantes ao contexto da pesquisa.

Outra característica presente no *framework* é o cálculo da distância dos imóveis em relação à um ponto de referência. Para o tratamento deste cálculo foi utilizado a fórmula de Haversine (2013). Ela foi incorporada ao *framework* a fim de facilitar a pesquisa dos dados. Visto que os resultados de pesquisas feitas pela *engine* do MySQL não demonstraram uma precisão confiável.

Além da área de pesquisa, também foi disponibilizado no módulo do *framework*, visualizada na Figura 38, uma área para seleção dos indicadores estatísticos, pré-definidos pelo administrador do sistema.

Selecione o indicador desejado

Indicador

Gráfico dos Imóveis por Tipo de Construção

Exibir

Figura 38 - Filtro dos Indicadores Estatísticos

Esse filtro de pesquisa tem por objetivo exibir estatísticas das informações cadastrais dos imóveis georreferenciados. Como por exemplo, Gráfico dos Imóveis por Tipo de Construção, conforme é observado na Figura 39.

Classificação dos Imóveis por Tipo de Construção

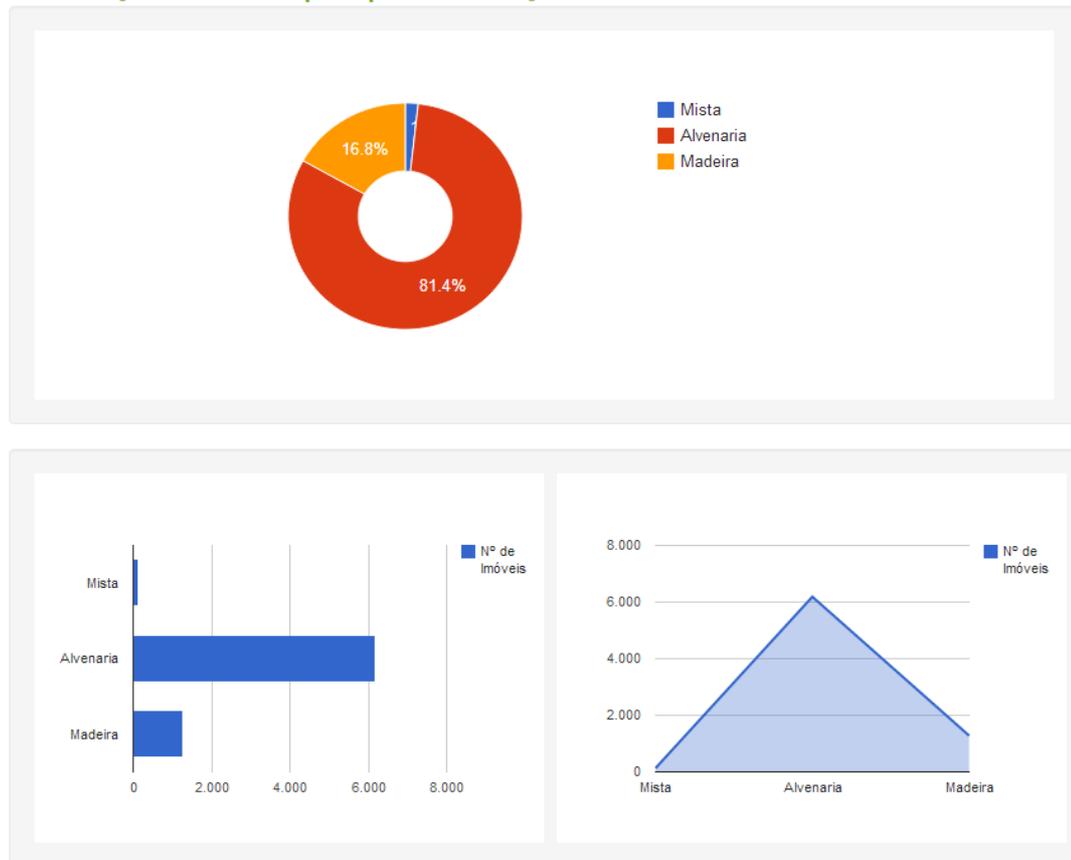


Figura 39 - Imóveis por Tipo de Construção

No gráfico, é possível visualizar o detalhamento do percentual calculado, apenas com a aproximação do ponteiro do mouse sobre a área indicada. Também é demonstrado, para a mesma pesquisa, três gráficos distintos.

Além do gráfico anterior, foi disponibilizado também os seguintes Indicadores: Gráfico dos Imóveis por Tipo de Ocupação, Gráfico dos Imóveis por Tipo de Patrimônio, Gráfico dos Imóveis por Estado da Construção, Gráfico dos Imóveis por Bairro, Gráfico dos Imóveis por Zona, Gráfico dos Imóveis por Setor e Gráfico dos Imóveis por Quadra, conforme Figura 40.

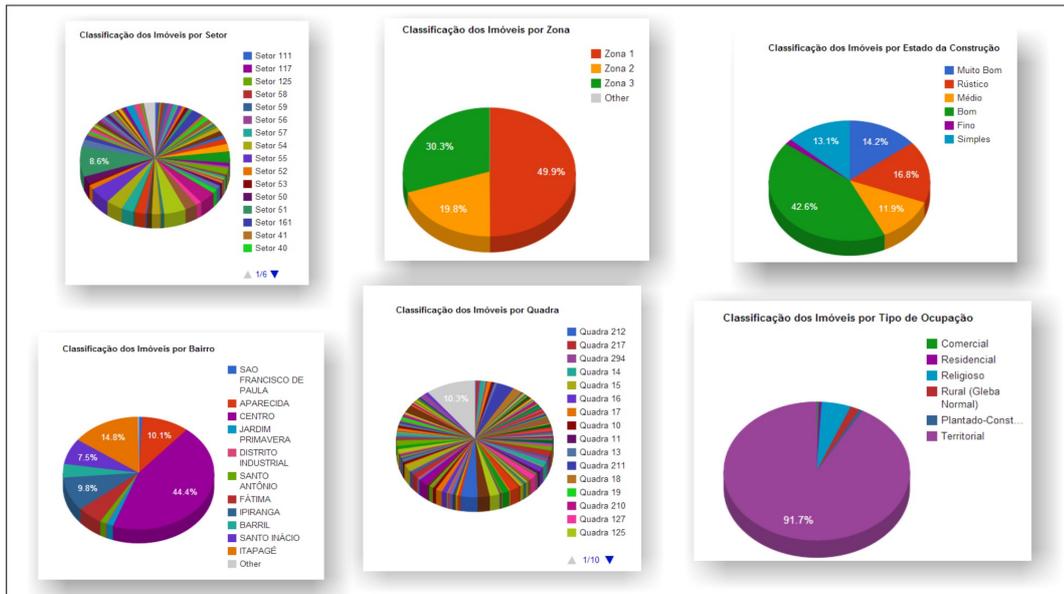


Figura 40 - Amostragem dos indicadores estatísticos

- Mapa dos Bairros: é uma área destina à visualização dos mapas de cada bairro da cidade. Primeiro é listado a relação de bairros e clicando sobre cada um é possível visualizar a sua localização espacial no mapa, conforme visualizado na Figura 41.

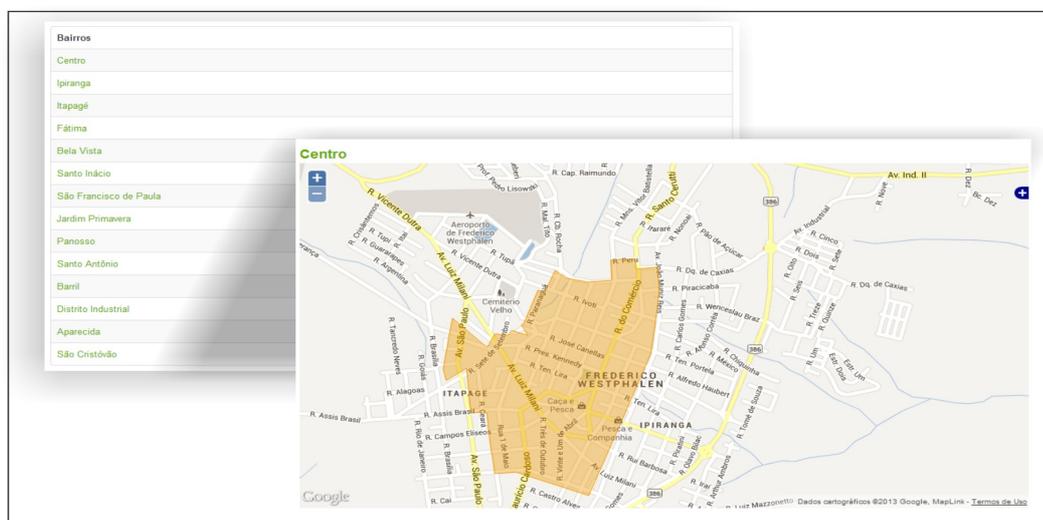


Figura 41 - Camada do Bairro sobre o mapa da cidade

CAPÍTULO 4 - ESTUDO DE CASO

4.1 Caracterização Geral

Frederico Westphalen é um município brasileiro do Estado do Rio Grande do Sul com população estimada em 30.251 habitantes, segundo dados do IBGE de 2013. Geograficamente, conforme observamos a Figura 42, o município está localizado a uma latitude 27°21'33" Sul e a uma longitude 53°23'40" a oeste. Tem uma densidade demográfica de 108,87 hab./km², altitude de 566 metros em relação ao nível do mar e clima subtropical úmido. Sua área é de aproximadamente 264,53 km², representando 0.099% do estado, 0.047% da região e 0.003% de todo o território brasileiro. Está localizado na mesorregião Noroeste Rio-Grandense à aproximadamente 434 km de distância da capital Porto Alegre (IBGE, 2013).

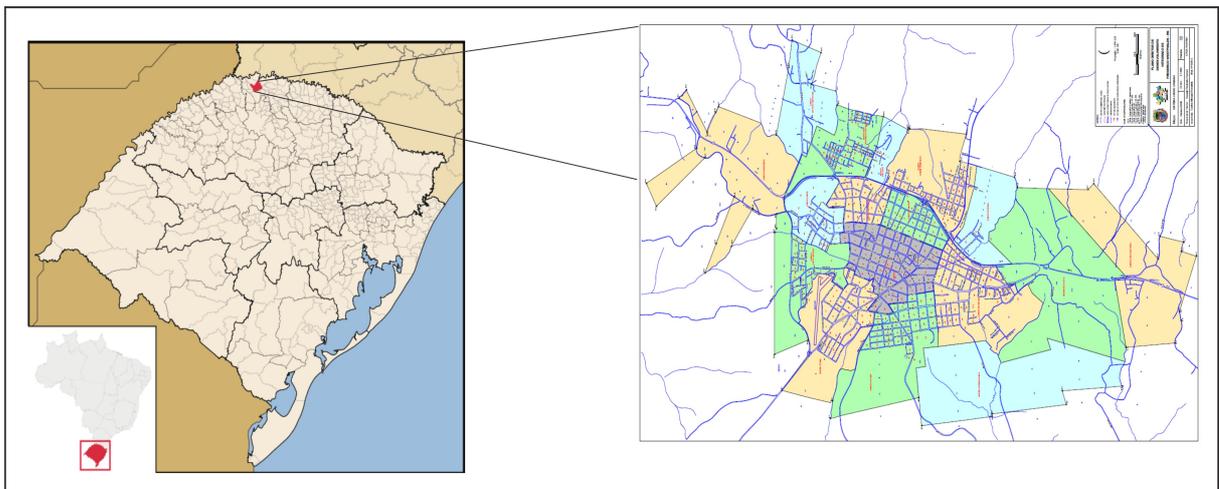


Figura 42 - Localização de Frederico Westphalen no mapa do Rio Grande do Sul e no Brasil

4.2 Plano Diretor de Desenvolvimento Integrado

Regido pela Lei Municipal nº 3.620/2010, de 01 de dezembro de 2010, o Plano Diretor de Desenvolvimento Integrado (2010) do município de Frederico Westphalen/RS, institui normas, princípios e diretrizes acerca do perímetro urbano, zoneamento, uso e ocupação do solo e do parcelamento do solo urbano e seu Artigo 187, § 2º menciona que "O Sistema de Informações Municipais deve, progressivamente, disponibilizar informações georreferenciadas, em meio digital, para as devidas cópias impressas".

O Artigo 46, institui que o ordenamento urbano divide-se em duas macrozonas, de modo a organizar o espaço urbano e rural:

I - Macrozona Urbana consolidada: formada pelo perímetro urbano da sede municipal e pelos bairros, respeitada a legislação que os criou: Centro, Santo Antônio, Itapagé, Fátima, Aparecida, Barril, Barrilense, Distrito Industrial, São Francisco de Paula, São José, Santo Inácio, Ipiranga, Jardim Primavera, Bela Vista, São Cristóvão, Jardim Paraíso e Panosso, onde se concentra a maior população urbana do Município;

II - Macrozona Urbana em Dinamização, formada pelos perímetros urbanos dos distritos de Osvaldo Cruz e Castelinho;

Cavenaghi e Lima (2013) afirmam que para a construção do Plano Diretor é fundamental conhecer a realidade de todo município, o que inclui a infraestrutura da cidade, o cadastro das áreas construídas, as redes de transporte, água e esgoto, os serviços públicos, os pontos turísticos, as áreas de preservação, dentre outras variáveis consideradas na gestão de uma Prefeitura. Em artigo publicado na versão online da revista InfoGeo, trazem exemplos de como uso de SIG e de imagens de satélite vai além de gerar os subsídios para a formação do Plano Diretor, auxiliar vários departamentos da Prefeitura a planejar suas ações e ainda disponibilizar estas informações via Internet, o que melhora e agiliza o atendimento ao cidadão.

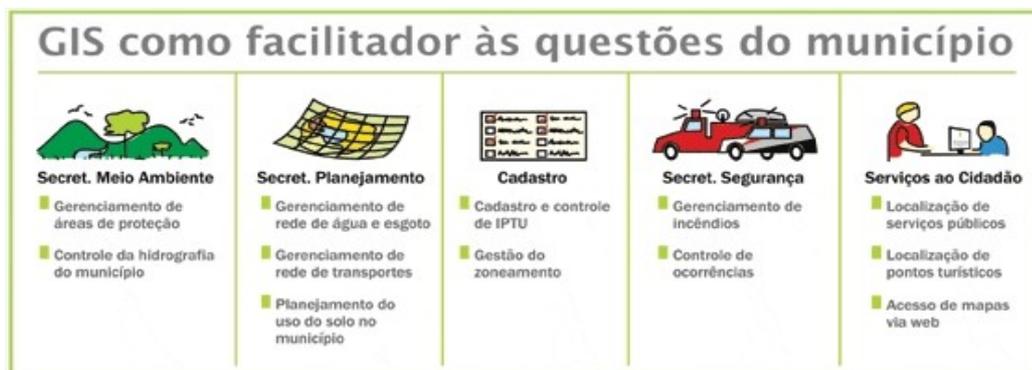


Figura 43 - Uso de SIG na administração municipal (Cavenaghi e Lima, 2013)

Vemos na Figura 43, que as possibilidades de uso dos Sistemas de Informação Geográfica são imensas. Podendo ser aplicadas em diversas áreas, como mecanismo de apoio à tomada de decisão em análises estratégicas, como por exemplo, coletar informações sobre a estimativa populacional por bairro através da contagem de unidades residenciais, ou, para identificar loteamentos clandestinos ou o mapeamento de loteamentos vazios.

Dessa forma, a sua utilização torna-se um diferencial no processo de automação do

planejamento urbano, pois com ela, torna-se mais ágil a identificação dos anseios imediatos à população, pois ela permite uma visão mais ampla da cidade, conduzindo à uma melhoria significativa na qualidade de serviços prestados à população.

Vemos na Figura 44, uma amostragem da base cartográfica do município de Frederico Westphalen definida pelo Plano Diretor. Porém, para a construção deste projeto foi adquirido através do departamento da Secretaria da Fazenda deste município, uma cópia do mapa urbano no formato DXF (*Drawing eXchange Format*), versão 2004. O sistema de coordenadas utilizado foi o SAD69 (South American Datum 1969), e a escala adotada foi 1:50.000.

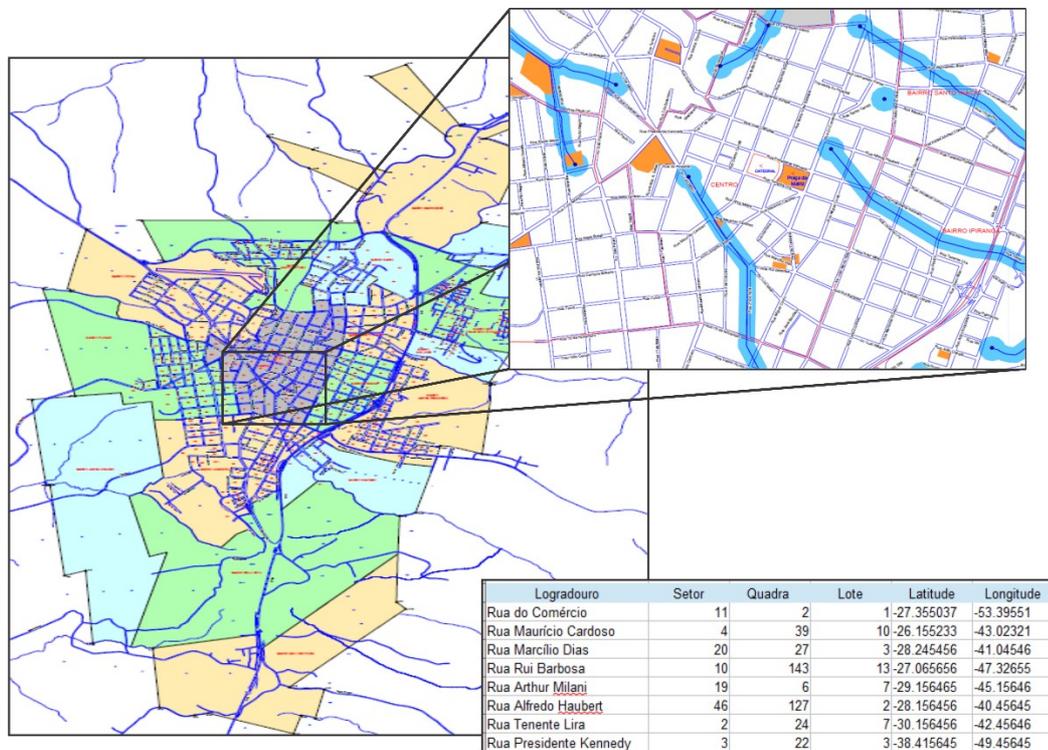


Figura 44 - Base Cartográfica com as divisões do Zoneamento Urbano

Porém, para a realização deste trabalho, foi necessário reestruturar a base cartográfica de lotes e quadras do município de Frederico Westphalen/RS, através do auxílio do software AutoCAD 2013 (AUTOCAD, 2013), pois havia informações em excesso no desenho dos mapas. Vemos na Figura 45, como era e como ficou o mapa das divisões do loteamento urbano.

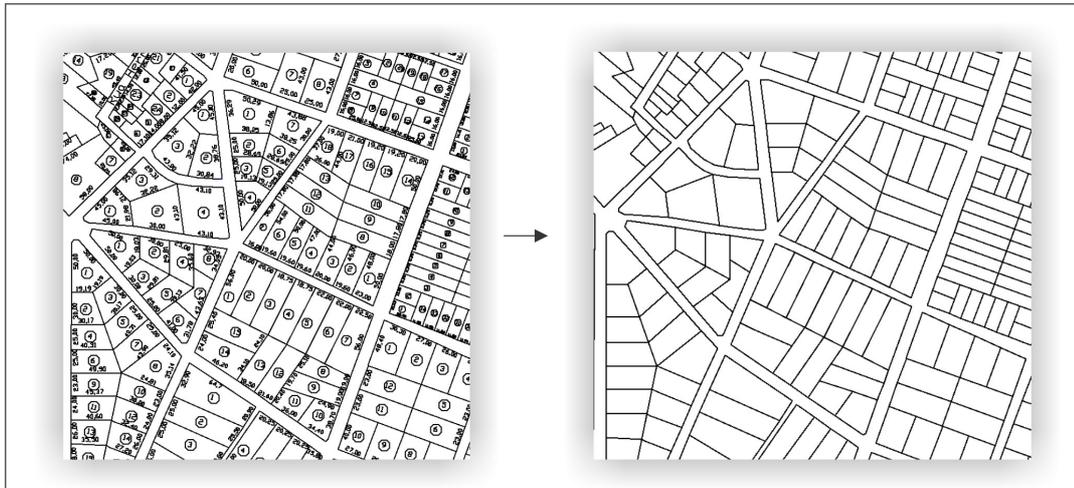


Figura 45 - Base cartográfica do loteamento urbano após o tratamento efetuado

Como objetivo deste trabalho é a integração de ferramentas SIG com o cadastro imobiliário do município, adotou-se como escopo principal o tratamento cartográfico da camada de bairros, visualizada na Figura 46, usada para este estudo. Já, as camadas de lotes e quadras necessitam de um tratamento individual de identificação e correção de sua geometria, para que seja possível vinculá-lo ao projeto em questão. Dessa forma, a continuidade deste projeto torna-se um grande diferencial para ampliar as possibilidades de controle e captação das informações geográficas sobre o estudo apresentado.

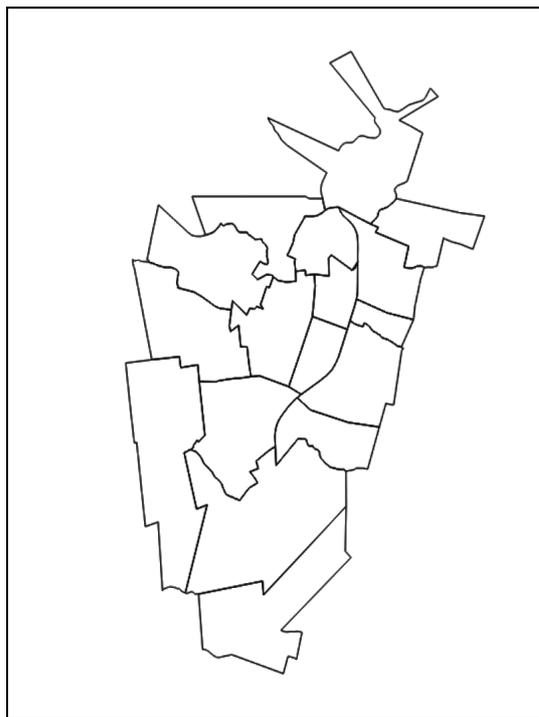


Figura 46 - Base cartográfica dos bairros

4.3 Sistema de Tributos

O Sistema de Tributos que atualmente a Prefeitura Municipal de Frederico Westphalen possui, tem por objetivo otimizar e facilitar a cobrança de impostos e taxas recolhidos pela administração pública municipal, afim de atender as normas do Código Tributário Municipal. Ou seja, capturar informações que caracterizam a área urbana, atualizar as informações de cada propriedade imobiliária, efetuar cálculos de IPTU, e demais operações que o setor exige.

Para o registro dos imóveis urbanos do município é utilizado o Sistema de Tributos, de propriedade da empresa Digifred Informática Ltda, visualizado na Figura 47.

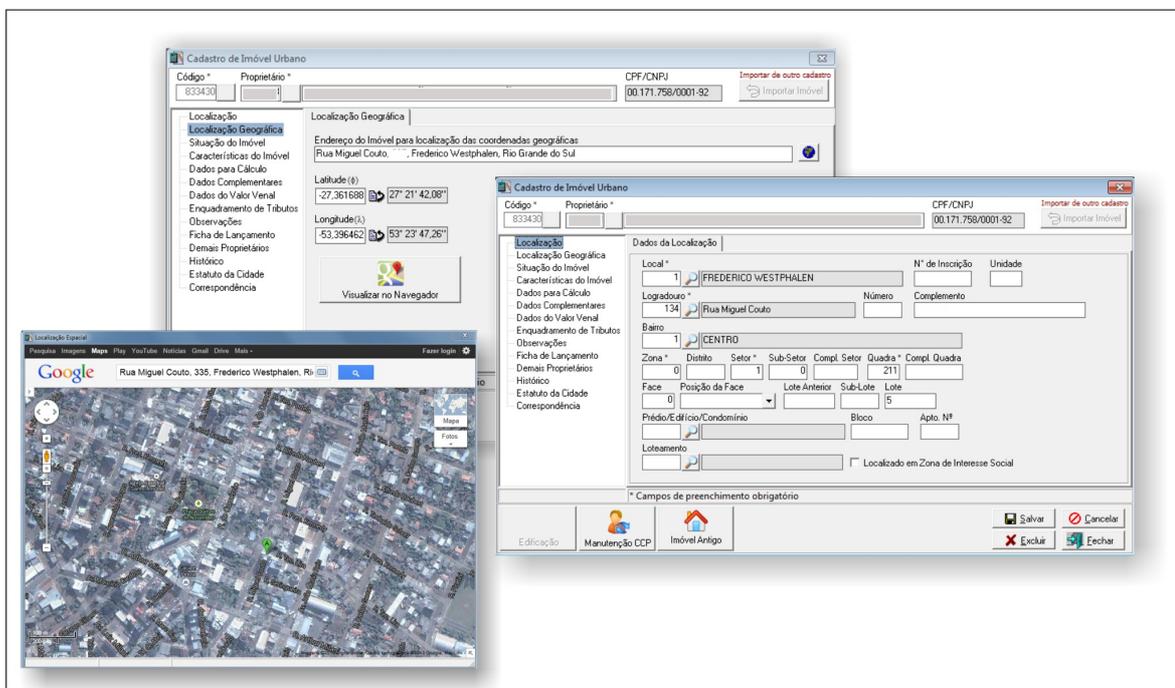


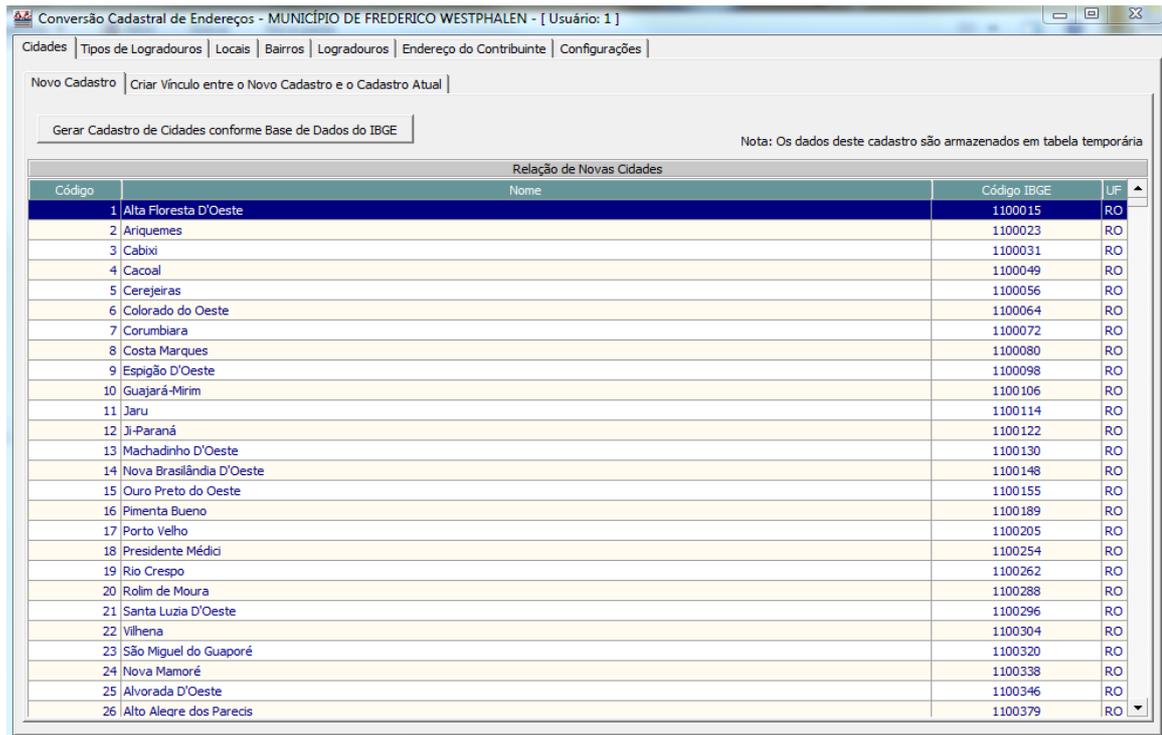
Figura 47 - Telas do cadastro de imóveis urbanos

Atualmente, o cadastro de imóveis urbanos somente possuía informações referente à sua localização física e dados para o cálculo tributário. Porém, devido à necessidade de referenciar o imóvel com as coordenadas geográficas, foi disponibilizado no próprio cadastro do imóvel essas informações, para que o usuário do sistema possa preenchê-las durante o seu registro ou para manutenção futura. No exemplo da Figura 47, temos uma ilustração do par de coordenadas identificadas para o imóvel que está sendo editado.

Para auxiliar no processo de identificação do par de coordenadas geográficas, foi disponibilizado, no próprio cadastro, uma tela para localização do imóvel através das ferramentas do Google Maps (GOOGLE MAPS API, 2013).

4.3.1 Tratamento dos Dados

Devido à necessidade de padronização nos dados alfanuméricos de localização a partir do nome dos logradouros, foi necessário desenvolver uma ferramenta, ilustrada na Figura 48, para a correção e tratamento dos dados afim de se padronizar.



Código	Nome	Código IBGE	UF
1	Alta Floresta D'Oeste	1100015	RO
2	Ariquemes	1100023	RO
3	Cabixi	1100031	RO
4	Cacoal	1100049	RO
5	Cerejeiras	1100056	RO
6	Colorado do Oeste	1100064	RO
7	Corumbiara	1100072	RO
8	Costa Marques	1100080	RO
9	Espigão D'Oeste	1100098	RO
10	Guajará-Mirim	1100106	RO
11	Jaru	1100114	RO
12	Ji-Paraná	1100122	RO
13	Machadinho D'Oeste	1100130	RO
14	Nova Brasilândia D'Oeste	1100148	RO
15	Ouro Preto do Oeste	1100155	RO
16	Pimenta Bueno	1100189	RO
17	Porto Velho	1100205	RO
18	Presidente Médici	1100254	RO
19	Rio Crespo	1100262	RO
20	Rolim de Moura	1100288	RO
21	Santa Luzia D'Oeste	1100296	RO
22	Vilhena	1100304	RO
23	São Miguel do Guaporé	1100320	RO
24	Nova Mamoré	1100338	RO
25	Alvorada D'Oeste	1100346	RO
26	Alto Alegre dos Parecis	1100379	RO

Figura 48 - Ferramenta utilizada para padronizar os dados cadastrais de localização

A necessidade de tal padronização deve-se ao fato de existirem diversas ocorrências da mesma informação na base de dados, o que gera confusão durante a pesquisa. Por exemplo, na Figura 49, vemos uma demonstração de como padronizar a situação cadastral da cidade de Frederico Westphalen. Foi utilizado como filtro a palavra "west" e o sistema encontrou 181 ocorrências desta palavra no cadastro de cidades.

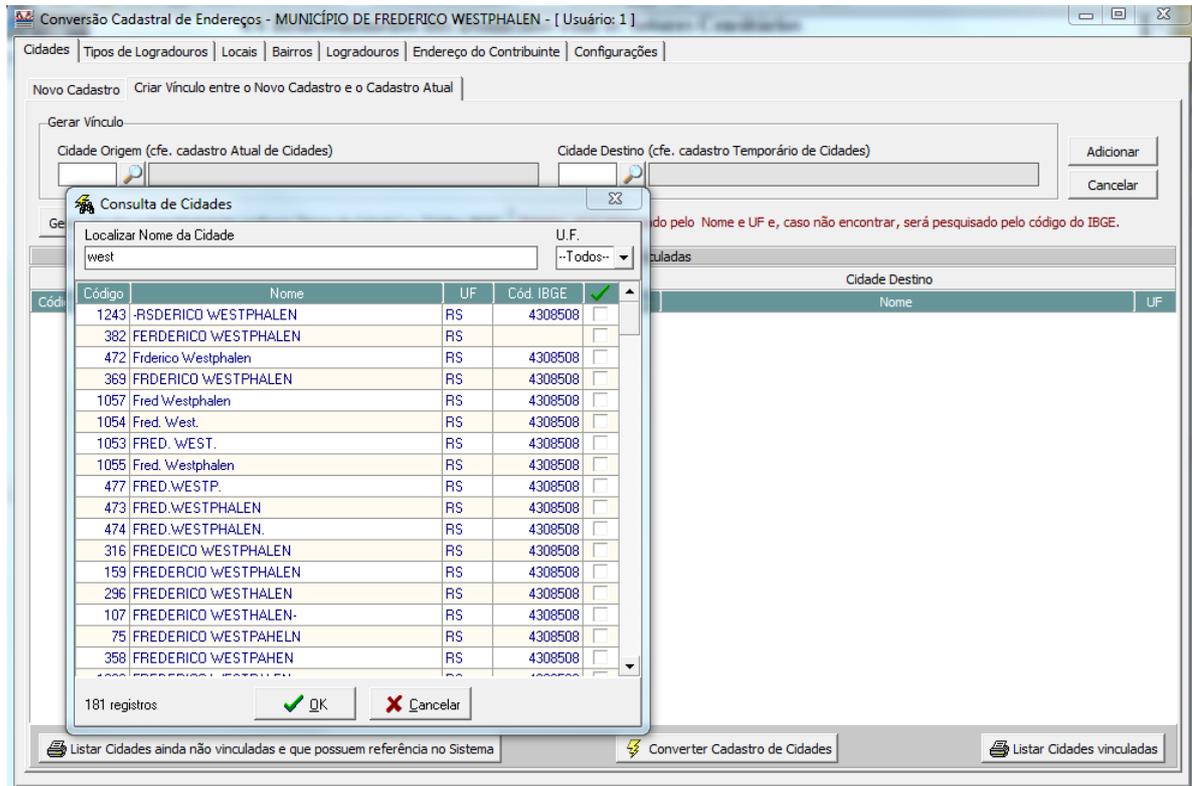


Figura 49 - Lista de cidades que possuem a palavra "west"

No caso acima, marca-se na última coluna da grade as linhas a serem substituídas pela cidade destino. Na Figura 50, vemos o filtro do cadastro de cidades definido pelo IBGE.

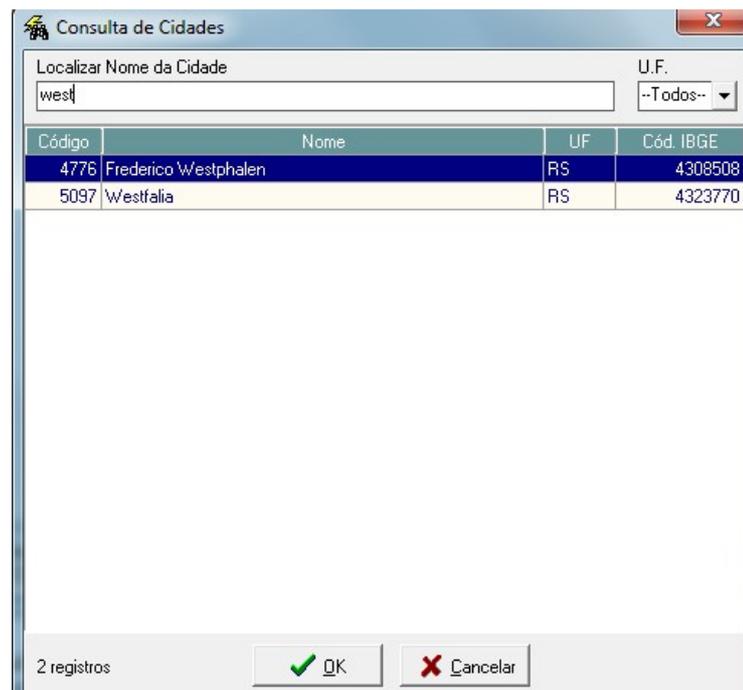


Figura 50 - Filtro do cadastro de cidades do IBGE

Dessa forma, vemos na Figura 51, a associação de todas as cidades origens com uma só destino, facilitando o processo de padronização cadastral.

Relação de Cidades vinculadas					
Cidade Origem			Cidade Destino		
Código	Nome	UF	Código	Nome	UF
159	FREDERICO WESTPHALEN	RS	4776	Frederico Westphalen	RS
296	FREDERICO WESTPHALEN	RS	4776	Frederico Westphalen	RS
369	FRDERICO WESTPHALEN	RS	4776	Frederico Westphalen	RS
382	FREDERICO WESTPHALEN	RS	4776	Frederico Westphalen	RS
473	FRED. WESTPHALEN	RS	4776	Frederico Westphalen	RS
1053	FRED. WEST.	RS	4776	Frederico Westphalen	RS
1243	RS DERICO WESTPHALEN	RS	4776	Frederico Westphalen	RS

Figura 51 - Associação das cidades origens com a destino

Outro exemplo, é o caso do cadastro de Tipos de Logradouro do sistema. Podemos observar na Figura 52, que existem três formas distintas do cadastro do tipo "Vila". Dessa forma, selecionamos os três itens, e associamos, na Figura 53, com o tipo padronizado Vila. Aplicamos a conversão e, na Figura 54, vemos a associação efetuada. Todos os cadastros do sistema serão atualizados com a aplicação desta conversão, facilitando assim a manutenção e padronizando a situação cadastral dos dados do sistema.

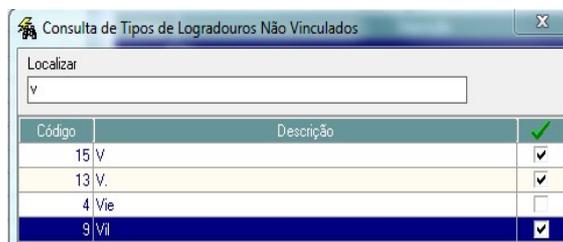


Figura 52 - Diversos Tipos de Logradouros

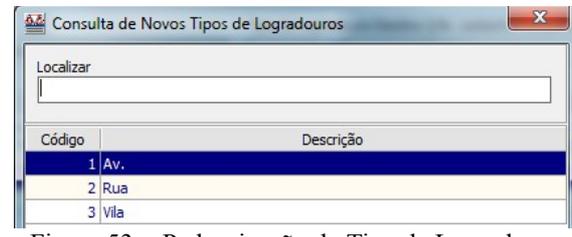


Figura 53 - Padronização do Tipo de Logradouro

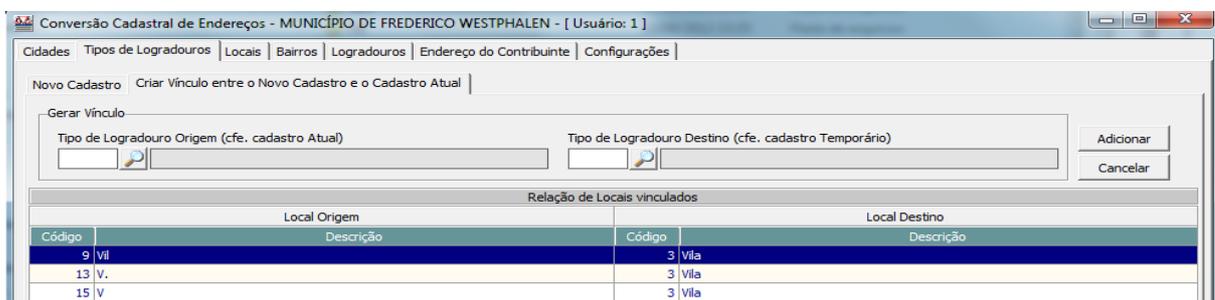


Figura 54 - Relação de Tipos de Logradouros convertidos

O processo de conversão desses dados é uma das etapas do processo de utilização de ferramentas SIG, pois para se obter qualidade na obtenção de informações é necessário que os dados estejam padronizados. Esse processo de limpeza e tratamento dos dados facilita a operacionalidade do sistema e garante maior precisão na coleta das informações.

Para auxiliar no processo de captação de informações para este estudo, foi desenvolvido também um assistente para geocodificar automaticamente o imóveis via API do GoogleMaps, a partir das informações do nome do logradouro e do nº do endereço do imóvel, conforme é observado na Figura 55.

Conversão Cadastral de Endereços - MUNICÍPIO DE FREDERICO WESTPHALEN - [Usuário: 1]

Cidades | Tipos de Logradouros | Locais | Bairros | Logradouros | Endereço do Contribuinte | Configurações | Geocodificação

Efetuar automaticamente a conversão do endereço dos imóveis em coordenadas geográficas de Latitude e Longitude.

Local
1 FREDERICO WESTPHALEN

Logradouro

Cidade (obrigatório) UF
4776 Frederico Westphalen RS

Localizar por
 Google Maps
 OpenStreetMap

Substituir Imóveis já geocodificados

Geocodificar

Observação: Somente farão parte da geocodificação, os imóveis que possuem N° de Endereço configurado.

Figura 55 - Assistente para geocodificação dos imóveis

4.4 Resultados obtidos

A partir da integração dos dados alfanuméricos do cadastro imobiliário do Sistema de Gestão Tributária do município com as ferramentas SIG, foi possível analisar e identificar algumas amostras da necessidade desta tecnologia estar inserida no contexto da gestão pública. A primeira amostra refere-se a um recorte da carta topográfica da área urbana do município de Frederico Westphalen, da quadra nº 50, situada entre os logradouros: Rua Tenente Lira, Rua Guerino Cerutti, Rua Aparício Borges, Rua Antônio Boscardin e Avenida Luiz Milani, conforme é ilustrado na Figura 56. Na figura, os números na cor vermelha representam a numeração das quadras definidas pelo Plano Diretor de Desenvolvimento Integrado (PDDI) instituído pela Lei Municipal nº 3.620/2010 de 01 de dezembro de 2010.

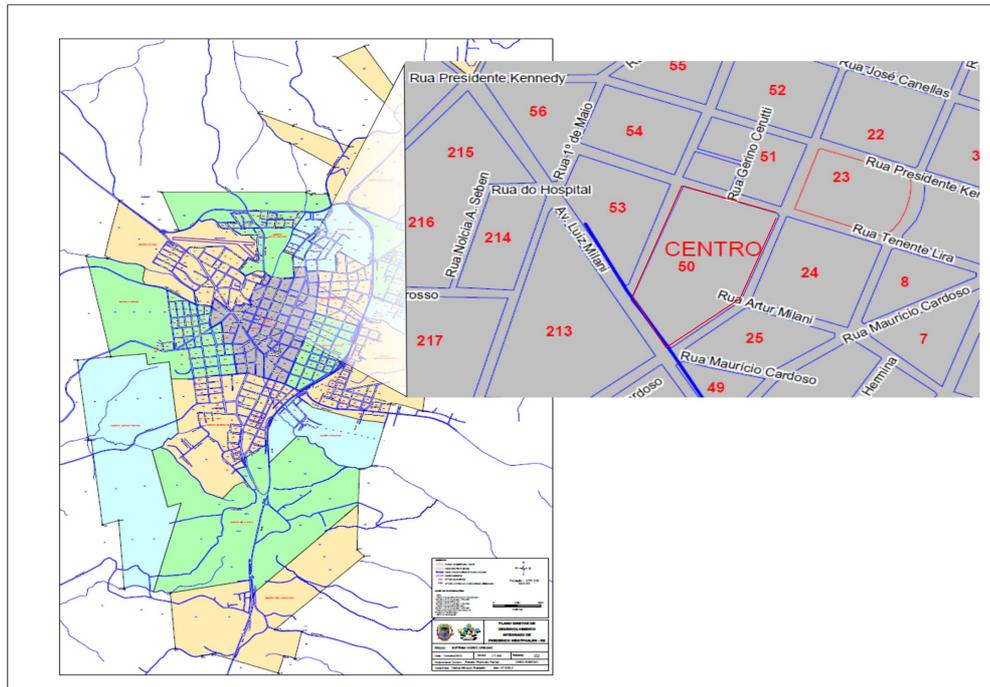


Figura 56 - Recorte da Carta Topográfica

Ao efetuar uma pesquisa dos imóveis identificados pela quadra nº 50, através do MapGeoCity, verifica-se uma dispersão geográfica desses imóveis no mapa, fora da área delimitada pela carta topográfica, destacado com um polígono de cor vermelha na Figura 57.

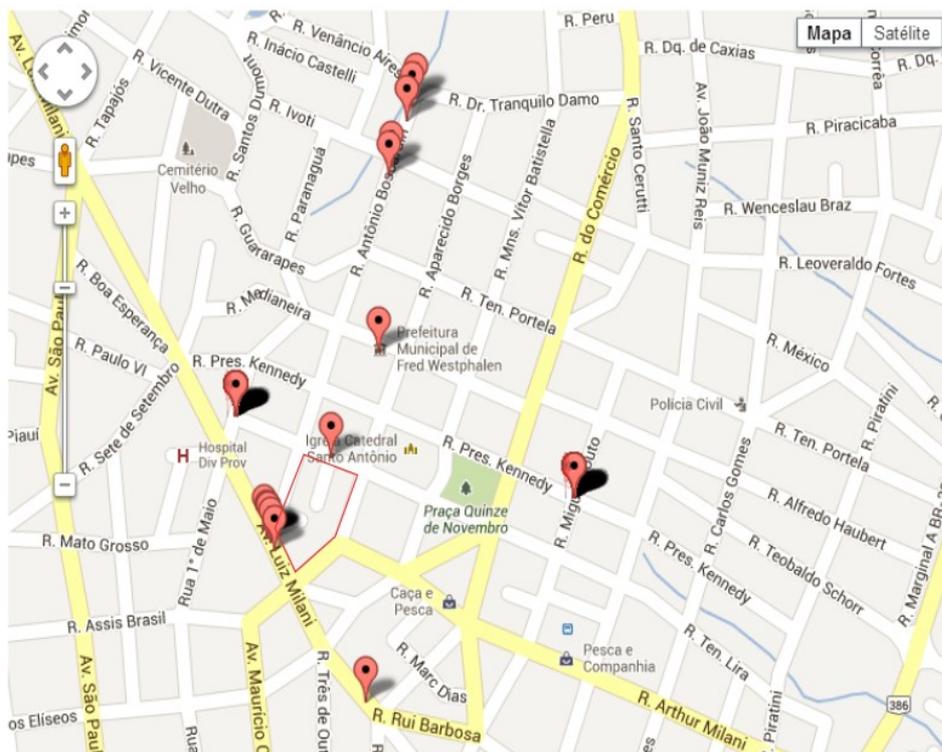


Figura 57 - Imóveis vinculados à quadra 50

Outro caso de uso, é a pesquisa dos imóveis vinculados ao Bairro Centro com área de terreno entre 50 e 70². Conforme podemos observar na Figura 58, o imóvel destacado localiza-se fora dos limites do bairro, demarcado pela camada na cor cinza.

Com isso, percebe-se a necessidade cada vez maior de ferramentas SIG na gestão pública, pois elas facilitam a identificação de problemas cadastrais nas bases de dados, auxiliando no processo de tomada de decisão e contribuindo com a integridade dos dados. Por mais simples que seja o tratamento de pesquisa efetuada pelo MapGeoCity, é possível extrair informações novas que não eram possíveis somente com o uso do Sistema de Gestão Tributária que o município possui. Com a integração dos dados cadastrais com as ferramentas SIG é possível ampliar o conhecimento sobre as informações do sistema, agregando mais valor aos dados.

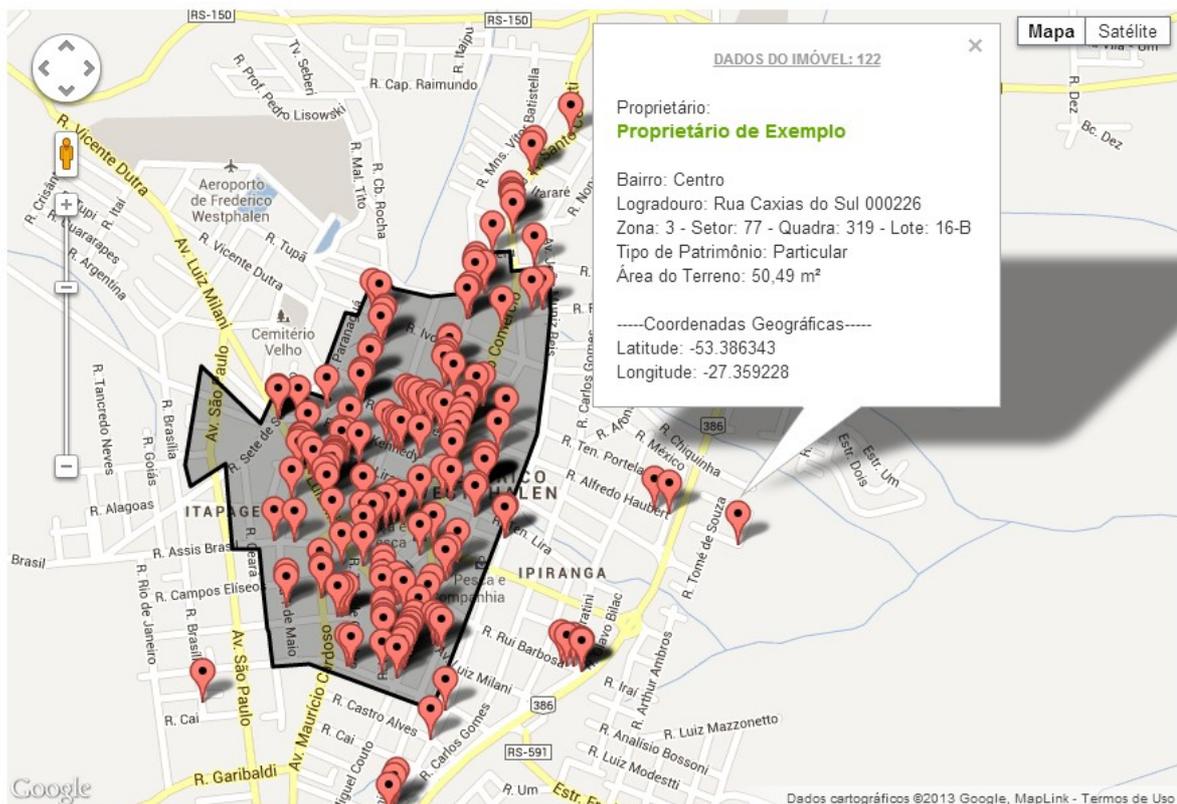


Figura 58 - Distribuição espacial dos imóveis vinculados ao Bairro Centro

CAPÍTULO 5 - CONCLUSÃO E TRABALHOS FUTUROS

Frente aos desafios propostos para este estudo, os objetivos contextualizados foram alcançados. Dentre eles, os principais foram o de portar o *framework* GeoLiberty do Postgres para o MySQL, construir um aplicativo para auxiliar no processo de tomada de decisão das informações georreferenciadas dos imóveis do Município de Frederico Westphalen e a integração com o Sistema de Gestão Tributária deste município.

Após o período de estudos para a elaboração deste trabalho foi possível determinar que as ferramentas aqui apresentadas tem um enorme potencial para a construção de Sistemas de Informação Geográfica. Para tanto, é necessário observar que, a curva de aprendizado das tecnologias envolvidas denota tempo e dedicação, a fim de se conseguir melhores resultados com o estudo em questão.

A implantação de um SIG municipal é complexa e deve ser planejada à longo prazo. Visto que, ela envolve várias etapas, desde a concepção de uma equipe técnica qualificada e permanente para cada área de pesquisa.

Um dos maiores desafios enfrentados nesse projeto foi a padronização e organização das ferramentas certas para se estabelecer um ambiente de trabalho. Contudo, na área de TI, a todo momento, nascem novas maneiras de se integrar as tecnologias e manter um sincronismo com elas é de fundamental importância para a saúde de um projeto.

Outro fator importante é sempre, quando possível, dispor de dados reais e atualizados, pois eles são de fundamental importância para diagnosticar resultados de qualidade no projeto. E, como o estudo em questão partiu-se dos dados disponíveis pelo Sistema de Gestão Municipal que a Prefeitura Municipal de Frederico Westphalen alimenta, facilitou a elaboração dos resultados.

Durante as fases de pesquisa deste estudo, observou-se que a extensão espacial do MySQL ainda possui muitas restrições quanto ao uso de operações de natureza espacial, assim como, ferramentas para facilitar o intercâmbio de dados com os diversos formatos de mapas digitais. Também foi possível diagnosticar resultados distintos aplicados sobre um caso específico entre o Postgres e o MySQL, dentre eles, a função de capturar a coordenada central de um determinado polígono.

Já, no PostgreSQL, fazendo uso de sua extensão espacial PostGIS, foi possível diagnosticar o vasto leque de opções que ele oferece, haja vista que a maioria das ferramentas

SIG possuem compatibilidade com essa tecnologia.

Entretanto, a conclusão desta pesquisa não significa o fim da jornada, pois com ela, foi possível alcançar apenas um degrau de tantos que este área oferece.

E como recomendação para trabalhos futuros aplicados sobre o contexto do tema em questão, são elencados alguns itens:

- Melhorar a *front-end* de pesquisa da aplicação, dando maiores funcionalidades e a capacidade de armazenar os filtros de pesquisa na forma de indicadores;
- Fazer integração com ferramentas de BI e OLAP, e também com dispositivos móveis, a fim de se ampliar a portabilidade da aplicação e estendendo a visão do negócio;
- Concluir o processo de identificação e georreferenciamento das cartas topográficas de quadras e lotes, a fim de se criar mais recursos para o aprimoramento da estatística dos imóveis;
- Ampliar a metodologia de estudo para outros municípios, e após projetar estatísticas comparativas entre os municípios, a fim de se obter resultados para melhorar a gestão das informações no município;

REFERÊNCIAS BIBLIOGRÁFICAS

- APACHE. Disponível em <<http://www.apache.org>>. Acesso em 07. jul. 2013.
- AUTOCAD 2013. Disponível em <<http://www.autodesk.com.br/>>. Acesso em 30. set. 2013.
- BORGES, Luiz Eduardo. Python para Desenvolvedores. 2ª ed. 2010. Disponível em <<http://ark4n.wordpress.com/python/>>. Acesso em: 03 mar. 2013.
- BORRERO, Santiago. Histórias Cruzadas: 15 anos conectando você às soluções geoespaciais. Revista MundoGeo. Ed. 72. (2013, mar. e abr.)
- BRANDAO, Marinho. Aprendendo Django no Planeta Terra. 1ª Edição. 2009.
- CÂMARA, G., MONTEIRO, A. M., DAVIS C. Introdução à Ciência da Geoinformação. São José dos Campos, INPE, 2001.2a. edição. Disponível em <http://www.dpi.inpe.br/gilberto/livro>. Acesso em 07 mar. 2013.
- CAVENAGHI, T. P.; LIMA, M. Plano Diretor: Como a Geotecnologia tem Facilitado a Gestão dos Municípios. REVISTA INFOGEO ONLINE, Curitiba-PR, Edição Especial: Cidades, nov 2006. Disponível em: <<http://mundogeo.com/blog/2000/01/01/plano-diretor-como-a-geotecnologia-tem-facilitado-a-gestao-dos-municipios>> Acesso em: 12 mar. 2013.
- DENSHAM, P. Spatial decision support systems. In: Maguire, D. J.; Goodchild, M. F.. Rhind, D. W. ed., Geographical Information Systems: principles and applications, New York, Longman, vol. 1, 1991.
- DICAS DE HOSPEDAGEM. (2011, nov. 25). Apache: conheça um pouco mais sobre o mais popular servidor web. Disponível em <<http://dicasdehospedagem.com/apache-conheca-um-pouco-mais-sobre-o-mais-popular-servidor-web/>>. Acesso em 07 jul. 2013.
- DIGIFRED Informática Ltda. Disponível em <<http://www.digifred.com.br>>. Acesso em 25 fev. 2013.
- DJANGO. Disponível em <<http://http://www.djangobrasil.org>> Acesso em: 07 mar. 2013.
em 22 fev. 2013.
- ELSMARI, RAMEZ e Shamkant NAVATHE. Sistemas de Banco de Dados. São Paulo: Pearson, 2006.
- FAGUNDES, Eduardo M. A Gestão eficiente de TI. Disponível em <http://www.efagundes.com/artigos/Gestao_eficiente_de_TI.htm>. Acesso em 21 abr. 2013.
- FILHO, Britaldo Silveira Soares. Cartografia Assistida por Computador: Conceitos e Métodos. Belo Horizonte: UFMG, 2000.
- FERNANDES, Aguinaldo A., ABREU, Vladimir F. Implantando a Governança de TI - da Estratégia à Gestão dos Processos e Serviços. 3ª. ed. Rio de Janeiro: Brasport, 2012.
- _____. NBR ISO/IEC 38500:2009: Governança corporativa de tecnologia da informação. Rio de Janeiro, Associação Brasileira de Normas Técnicas, 2009.
- FAUNALIA (2013). Infraestrutura de Dados Espaciais - Open Source. Disponível em <<http://www.faunalia.pt/downloads/brochuraide.pdf>>. Acesso em: 22 jun. 2013.

- FIREBIRD. Disponível em <<http://www.firebirdsql.org/>>. Acesso em 17 jun. 2013.
- HAZZARD, Erik. OpenLayers 2.10 Beginner's Guide. Packt Publishing. 2011.
- HAVERSINE (2013). Fórmula de Haversine. Disponível em <http://pt.wikipedia.org/wiki/F%C3%B3rmula_de_Haversine>. Acesso em: 10 nov. 2013.
- HENDERSON, J. C.; VENKATRAMAN N. Strategic Alignment. IBM Systems Journal, vol 38, nº 283, 1999.
- GEODJANGO. Disponível em <<http://geodjango.org>> Acesso em: 08 mar 2013.
- GOOGLE MAPS API – Disponível em <<https://developers.google.com/maps>> Acesso em: 20 mar. 2013.
- GRASS. Disponível em <<http://grass.osgeo.org/>>. Acesso em 18. jun. 2013.
- GVSIG. Disponível em <<http://www.gvsig.org/web>>. Acesso em: 17 jun. 2013.
- I3GEO. Disponível em <<http://i3geo.inde.gov.br>>. Acesso em: 10 jun. 2013.
- IBGE. Instituto Brasileiro de Geografia e Estatística – Disponível em <<http://www.ibge.gov.br>> Acesso em: 10 jun. 2013.
- ITGI (IT Governance Institute). CobiT® Quickstart, 2nd edition, Rolling Meadows, IL, 2007 (2007a).
- MALCZEWSKI, J. (1997), Spatial Decision Support Systems, NCGIA Core Curriculum in GIScience. Disponível em <<http://www.ncgia.ucsb.edu/giscc/units/u127/u127.html>> Acesso em 06 jun. 2013.
- MAPINFO. Disponível em <<http://www.geograph.com.br/#!mapinfo/c4nz>>. Acesso em 25 jun. 2013.
- MEDEIROS, Anderson. Dados Geográficos. 2010. Disponível em <<http://www.infoescola.com/geografia/dados-geograficos/>>. Acesso em 11 abr. 2013.
- MEDEIROS, Anderson M. L. SPRING - Tecnologia brasileira para SIG. Revista FOSSGIS. 4ª ed. (2012, jan. 04). Disponível em <<http://fossgisbrasil.com.br/download/>>. Acesso em 14 jun. 2013.
- MYSQL. Manual de Referência do MySQL 4.1. Disponível em <<http://ftp.nchu.edu.tw/MySQL/doc/refman/4.1/pt/>> Acesso em: 08 mar. 2013.
- MYSQLDB. Disponível em <<http://mysql-python.sourceforge.net/>> Acesso em: 20 mar. 2013.
- OGC STANDARDS – Disponível em <<http://www.opengeospatial.org>> Acesso em: 10 jun. 2013.
- OLIVEIRA, Felipe F., NOGUEIRA, Frederico G., Mapas Temáticos Dinâmicos e Dados Estatísticos. Revista FOSSGIS. 6ª ed. (2012, nov. 06). Disponível em <<http://fossgisbrasil.com.br/download/>>. Acesso em 05 jun. 2013.
- OPENLAYERS – Disponível em <<http://openlayers.org>> Acesso em: 20 mar. 2013.
- ORM – Object-relational Mapping – Disponível em <<http://www.orm.net>> Acesso em: 19 mar. 2013.
- OSGEO – Disponível em <<http://www.osgeo.org>>. Acesso em: 10 jun. 2013.

- PIMENTA, Fernando Martins ...[et al.]. Servidores de mapas : programação para disponibilizar dados geográficos multidisciplinares utilizando tecnologias livres - Brasília, DF : Embrapa, 2012.
- POSTGIS – Spatial Database Extension for PostgreSQL – Disponível em <<http://www.postgis.org>>. Acesso em: 22 jun. 2013.
- PLANO DIRETOR DE DESENVOLVIMENTO INTEGRADO DO MUNICÍPIO. (2010). Disponível em <<http://www.fredericowestphalen-rs.com.br/wp-content/uploads/Plano-Diretor.pdf>>. Acesso em 05 mar. 2013.
- PYTHON. Disponível em <<http://www.python.org.br/wiki>>. Acesso em 10 abr. 2013.
- QGIS (2013). Quantum GIS. Disponível em <<http://qgisbrasil.org>>. Acesso em: 17 jun. 2013.
- ROSA, Roberto. Introdução ao Sensoriamento Remoto. Uberlândia:UDUFU, 2003.
- RUGGINI, N.; SILVA, J. GeoLiberty como Solução para a Integração e Visualização de Dados Geográficos. In: Encontro Anual de Tecnologia da Informação, EATI, 2012, Frederico Westphalen, Anais... Frederico Westphalen, 2012.
- SHERMAN, G. E., 2008, Desktop GIS - Mapping the Planet with Open Source Tools (Pragmatic Bookshelf).
- SILVA, Evaldo de Oliveira, Introdução a Sistemas de Informações Geográficas. Disponível em <<http://www.devmedia.com.br/introducao-a-sistemas-de-informacoes-geograficas/1595>>. Acesso em: 13 mar. 2013.
- SILVA, Evaldo de Oliveira, Extensões Espaciais em MySQL, Revista SQLMagazine, ed. 14, 2004. ISSN 1677918 – 5. Leia mais em: Introdução a Sistemas de Informações Geográficas <http://www.devmedia.com.br/introducao-a-sistemas-de-informacoes-geograficas/1595#ixzz2Q72> OkGID.
- SILVA, Gleydson M. Guia Foca GNU/Linux - Capítulo 12 - Apache. (05 set. 2010). Disponível em <<http://www.guiafoca.org/cgs/guia/avancado/ch-s-apache.html>>. Acesso em 21 jul. 2013.
- SILVA, J. GEOMDQL: Uma Linguagem de Consulta Geográfica e Multidimensional. 2008. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2008.
- SQL – Structured Query Language – Disponível em <<http://www.w3schools.com/sql>> Acesso em: 19 mar. 2013.
- VIANNA, Paulo Henrique. GeoLiberty - Solução livre para construção de Sistemas de Informações Geográficas. 2012. Trabalho de Conclusão do Curso, CAFW/UFSM. 2012.
- TERRAVIEW 4.1.0. São José dos Campos, SP: INPE, 2010. Disponível em: <www.dpi.inpe.br/terraview>. Acesso em: 23 abr. 2013.
- WAMP SERVER. Disponível em <<http://www.wampserver.com/en/>>. Acesso em 21 jul. 2013.