

**VINÍCIUS GARCIA SPERANDIO**

**GERAÇÃO AUTOMÁTICA DE BANCO DE DADOS PARA GESTÃO MUNICIPAL A  
PARTIR DE INFORMAÇÃO GEOGRÁFICA VOLUNTÁRIA EXPORTADA DA  
PLATAFORMA OPENSTREETMAP**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

Orientador: Jugurta Lisboa Filho

**VIÇOSA – MINAS GERAIS  
2019**

**Ficha catalográfica preparada pela Biblioteca Central da Universidade  
Federal de Viçosa - Câmpus Viçosa**

T

S749g  
2019 Sperandio, Vinícius Garcia, 1995-  
Geração automática de banco de dados para a gestão municipal a partir de informação geográfica voluntária exportada da plataforma openstreetmap / Vinícius Garcia Sperandio. – Viçosa, MG, 2019.  
51f. : il. ; 29 cm.

Inclui apêndices.

Orientador: Jugurta Lisboa Filho.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.44-45.

1. Banco de dados geográficos. 2. Administração municipal. 3. Sistemas de informação geográfica. I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. II. Título.


CDD 22 ed. 621.367


VINÍCIUS GARCIA SPERANDIO

**GERAÇÃO AUTOMÁTICA DE BANCO DE DADOS PARA GESTÃO  
MUNICIPAL A PARTIR DE INFORMAÇÃO GEOGRÁFICA  
VOLUNTÁRIA EXPORTADA DA PLATAFORMA  
OPENSTREETMAP**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 17 de outubro de 2019.

  
\_\_\_\_\_  
Vinicius Garcia Sperandio  
(Autor)

  
\_\_\_\_\_  
Jugurta Lisboa Filho  
(Orientador)

## AGRADECIMENTOS

Agradeço ao meu professor e orientador Jugurta Lisboa Filho por todos os ensinamentos transmitidos como professor e pessoa. Obrigado por acreditar que juntos seríamos capazes de desenvolver e finalizar este trabalho.

À minha família que sempre se fizeram presentes. Em especial meus pais Valécia e Vanderlei e minha irmã Victória, que sempre compreenderam minhas dificuldades durante o período de mestrado e me apoiaram na realização desse sonho.

À minha namorada, parceira e amiga Larissa, por todo apoio, compreensão, amor e incentivo, não só para a conclusão deste trabalho, mas em tudo.

Aos meus sogros Laércio e Madalena e cunhada Laura, pelo carinho, conselhos e me acolherem sempre que preciso.

Aos meus parceiros Lucas Barros, Rômulo Gonçalves, Vanessa Alves e Yago Caiaffa por me aguentarem desde a graduação.

Aos meus amigos Hector Baranda, Fabio Reinoso, Fernando Passe, Victor Dias, Roney Grillo por estarem sempre comigo, tanto nos bons e maus momentos, são pessoas que tenho um grande carinho e levarei para o resto da vida.

À turma da pós-graduação em Ciência da Computação, pelo companheirismo e amizade.

Ao meu orientador na graduação, Sergio Murilo Stempliuc, por tornar possível meu ingresso no mestrado e pelo conhecimento, conselhos e experiências passados a mim.

A todos que trabalham comigo na DinniSoluções (José Alberto, Allefy, Tião, Léo Jaime, Guttardo, Icaro, Gelmer, Dudu, Gustavo, Rodrigo, Bryan, Maria, Adyla e Larrisa) pelo carinho, em especial à Tia Conceição e ao nosso chefe e mais novo amigo Gardiego Luiz.

À Universidade Federal de Viçosa pelo aprendizado e infraestrutura disponibilizada para a conclusão do meu mestrado. Agradeço a todos do Departamento de Informática e à CAPES, pela oportunidade, pela bolsa e auxílios concedidos para a realização desse trabalho.

A todos que de alguma forma me apoiaram.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

## RESUMO

SPERANDIO, Vinícius Garcia, M.Sc., Universidade Federal de Viçosa, outubro de 2019. **Geração automática de banco de dados para gestão municipal a partir de informação geográfica voluntária exportada da plataforma OpenStreetMap.** Orientador: Jugurta Lisboa Filho.

As restrições existentes no orçamento de pequenos e médios municípios, para o financiamento de projetos cartográficos provocou uma redução na produção de informação geográfica oficial. Entretanto, a facilidade de acesso a aparelhos eletrônicos moveis com geolocalizador, fez com que o mapeamento realizado através de Informação Geográfica Voluntária (VGI) ultrapassasse os métodos convencionais disponíveis, dada à simplicidade e baixo custo, sendo uma excelente alternativa para que os municípios tenham acesso a informações geográficas para auxílio nas tomadas de decisões. O presente trabalho apresenta um método e a ferramenta OSM2Diagram para extração de dados da plataforma OpenStreetMap visando a construção de uma base de dados geográfica para a gestão municipal de municípios de pequeno e médio porte, além do desenvolvimento do WebMap OSMCityView. A ferramenta OSM2Diagram possui a capacidade de produzir de modo automatizado o esquema conceitual da área exportada, além dos arquivos em formato *Shapefile* para visualização e análise em um software de Sistema de Informação Geográfica (SIG), em um sistema WebGIS ou serem armazenados em um Sistema Gerenciador de Banco de Dados Geográficos com capacidade para dados espaciais, como o sistema PostGIS. Um segundo produto desenvolvido nesta pesquisa foi um sistema WebMap, chamado OSMCityView, o qual permite ao gestor, ou qualquer cidadão, consultar informações geográficas sobre seu município a partir de um navegador Web. O potencial de uso do método e a funcionalidade dos sistemas desenvolvidos foram testados por meio do desenvolvimento de um estudo de caso tendo como alvo a cidade de Ubá, localizada na Zona da Mata do estado de Minas Gerais. Além de gerar um banco de dados geográfico estruturado no PostGIS, o estudo de caso mostrou exemplos de análises espaciais potenciais, realizadas por meio do SIG QGIS, que podem auxiliar o gestor na tomada de decisão.

Palavras-chave: Banco de dados geográficos. OpenStreetMap. Informação Geográfica Voluntária. Gestão municipal.

## ABSTRACT

SPERANDIO, Vinícius Garcia, M.Sc., Universidade Federal de Viçosa, October, 2019. **Automatic generation of database for city management from Volunteered Geographic Information exported from the OpenStreetMap platform.** Advisor: Jugurta Lisboa Filho.

The present budgetary constraints in small and medium towns for the financing of cartographic projects has led to a decrease in the production of official geographic information. However, the easy access to electronic devices (such as mobile phones) with embedded geolocators has made mapping through Volunteered Geographic Information (VGI) widespread, thus surpassing conventional available methods. Given its simplicity and low cost, VGI is an excellent alternative for cities to have access to geographic information useful in decision-making. This work presents a method and the OSM2Diagram tool, used for the extraction of data from the OpenStreetMap platform, with the goal of building a geographic database for the management of small and medium towns. The OSM2Diagram tool is able to produce, in an automated manner, the conceptual scheme of the exported area, as well as Shapefile files for viewing and analysis in a Geographic Information System (GIS), in a WebGIS system, or for storing in a Geographic Database Management System with capacity for spatial data, such as the PostGIS system. The second product of this research was a WebMap system, called OSMCityView, which allows the manager, or any citizen, to access geographic information about their city from a Web browser. The potential use of the method and the functionality of the developed systems were tested in a case study targeting the town of Ubá, located in the region known as Zona da Mata, in the state of Minas Gerais. Besides generating a geographic database structured in PostGIS, the case study showed examples of potential spatial analyses, performed with QGIS, which may assist the manager in decision-making.

Keywords: Geographic Database. OpenStreetMap. Voluntary Geographic Information. City management.

## LISTA DE FIGURAS

Figura 1 - Relevo de uma região fictícia .....	17
Figura 2 - Regiões de Planejamento de Minas Gerais.....	17
Figura 3 - Estrutura do framework UML-GeoFrame .....	20
Figura 4 - Estereótipos de framework UML-GeoFrame .....	20
Figura 5 - Utilização de temas no modelo UML-GeoFrame.....	21
Figura 6 - Modelagem conceitual utilizando o modelo UML-GeoFrame.....	22
Figura 7 - Diagrama de temas dos objetos disponíveis na plataforma OSM .....	25
Figura 8 - Trecho do diagrama de classes de alguns temas.....	26
Figura 9 - Área para estudo do arquivo OSM-XML .....	27
Figura 10 – Fluxograma do modulo gerador do esquema conceitual.....	30
Figura 11 - Fluxograma do modulo gerador de Shapefile.....	31
Figura 12 - Exemplo do arquivo GeoJson de objetos mapeados como <i>fast-food</i> em Viçosa...	32
Figura 13 - Perímetro urbano da cidade Ubá-MG.....	34
Figura 14 - Trecho do Relatório das classes mapeadas com falta de informações .....	35
Figura 15 - Trecho da modelagem conceitual da cidade Ubá .....	35
Figura 16 - Tela do sistema WebMap OSMCityView .....	37
Figura 17 – Perímetro de Ubá no QGIS .....	38
Figura 18 - Centro da cidade de Ubá no QGIS.....	39
Figura 19 – Análise de APP sobre o Rio Ubá .....	40
Figura 20 - Centro da cidade de Ubá gerada pela ferramenta OSMQuick.....	41

## LISTA DE CÓDIGOS

Código 1 - Exemplo da Tag Node sem atributos.....	27
Código 2 - Exemplo da Tag Node com atributos .....	28
Código 3 - Exemplo da Tag Way .....	28
Código 4 - Exemplo da Tag Relation .....	28
Código 5 - SQL APP Rio Ubá.....	40

## **LISTA DE ABREVIATURAS E SIGLAS**

**APP** – Área de Preservação Permanente

**OSM** – OpenStreetMap

**SIG** – Sistema de Informação Geográfica

**SGBD** – Sistema Gerenciador de Banco de Dados

**UBS** – Unidade Básica de Saúde

**UGC** – Conteúdo Gerado por Usuários

**VGI** – Informação Geográfica Voluntária

## SUMÁRIO

1	INTRODUÇÃO.....	10
1.1	Motivação .....	10
1.2	Objetivos.....	11
1.3	Estrutura da dissertação .....	12
2	REFERENCIAL TEÓRICO.....	13
2.1	Web 2.0.....	13
2.2	Informação Geográfica Voluntária .....	13
2.3	OpenStreetMap .....	14
2.4	Representação de dados espaciais .....	16
2.5	Modelos conceituais para banco de dados geográficos .....	18
2.6	Tipos de software para tomada de decisão usando informação espacial.....	22
3	MÉTODO PARA GERAÇÃO DE BASE DE DADOS GEOGRÁFICA.....	24
3.1	Modelagem conceitual da plataforma OpenStreetMap .....	24
3.2	Geração do esquema conceitual relativo a uma área selecionada .....	26
3.3	Geração de arquivos Shapefile .....	30
4	OSM2DIAGRAM – IMPLEMENTAÇÃO DO MÉTODO.....	33
4.1	Geração do esquema conceitual .....	34
4.2	Geração de arquivos Shapefile .....	36
4.3	Visualização de arquivos Shapefile usando um WebMap.....	36
4.4	Análise espacial utilizando o QGIS.....	37
4.5	Comparando o OSM2Diagram com a ferramenta OSMQuick .....	41
5	CONCLUSÕES .....	42
	REFERENCIAS .....	44
	APÊNDICE A - MANUAL DE INSTALAÇÃO E USO DA FERRAMENTA OSM2DIAGRAM .....	46
	APÊNDICE B - MANUAL DE CONFIGURAÇÃO DO OSMCITYVIEW.....	50

# 1 INTRODUÇÃO

A popularização da Web 2.0 favoreceu o desenvolvimento de aplicações que permitem aos usuários compartilharem informações espaciais por meio da Internet. Com isto, os usuários deixaram de ser apenas consumidores, e passaram a desempenhar o papel de colaboradores e produtores de informação (BUDHATHOKI, 2007).

Desde então, plataformas responsáveis por permitir que usuários sejam mais que simples consumidores, pois permitem que os usuários atuem como sensores e produtores de dados de forma voluntária (GOODCHILD, 2007). Neste contexto, a Informação Geográfica Voluntária (VGI) tem sido a principal fonte de dados em plataformas como OpenStreetMap<sup>1</sup> e Wikimapia<sup>2</sup>. Esses sistemas permitem que os usuários realizem o mapeamento de qualquer parte do globo terrestre, de modo livre, rápido e intuitivo (HAKLAY e WEBER, 2008).

Assim, as geotecnologias vêm ganhando espaço no mercado, por mostrarem eficiência como ferramentas de auxílio à tomada de decisão em diversas áreas. Em um município, por exemplo, elas podem auxiliar seus gestores e os empreendedores municipais que necessitam de um conhecimento do território onde pretendem atuar (ALMEIDA e ANDRADE, 2015).

Este trabalho aborda uma alternativa para que gestores municipais de cidades de pequeno e médio porte possam produzir uma base de dados municipal geográfica. A base da alternativa é a utilização de VGI, a qual possibilita que, com a participação do cidadão voluntário, o município seja capaz de manter uma base cartográfica atualizada, de forma a auxiliar o gestor na tomada de decisões.

## 1.1 Motivação

Segundo Goodchild (2007), nas últimas décadas ocorreu uma redução considerável na produção de informação geográfica oficial em todo o mundo, principalmente pelo alto custo para geração destas informações e na diminuição do financiamento para estes serviços cartográficos. Sendo assim, a informação geográfica fornecida de forma voluntária se transforma em uma grande possibilidade de solução para minimizar este problema.

No Brasil, os pequenos municípios, em especial, sofrem com a dificuldade de recursos financeiros, além disso, existe escassez de mão-de-obra qualificada para produção e manutenção de informações geográficas. Essas informações são de suma importância para o planejamento e tomada de decisões por parte dos dirigentes locais, pois podem auxiliar no

---

<sup>1</sup> <https://www.openstreetmap.org/>

<sup>2</sup> <http://wikimapia.org>

acompanhamento da evolução econômica e social de uma cidade, respeitando a questão ambiental. Portanto, um município precisa ter acesso a informações sobre seu território para possibilitar uma gestão com mais qualidade e eficiência (MIRANDA, 2010).

Normalmente os dados geográficos são armazenados em um Bancos de Dados Espaciais e manipulados por SIG, para que os gestores possam realizar análises espaciais que apoiem a tomada de decisões. No entanto, é importante que a modelagem dos dados seja acurada, pois é neste momento que os elementos do mundo real são transformados em elementos de um banco de dados (ELMASRI e NAVATHE 2011). Conseqüentemente, uma boa modelagem reduz o número de manutenções corretivas, contribui para uma melhor compreensão dos dados armazenados e seus relacionamentos. Dessa forma é possível prevenir gastos futuros, uma vez que dentre as atividades da engenharia de software, a manutenção é a que mais gera custos e um alto volume de esforço.

A plataforma OpenStreetMap (OSM) (OPENSTREETMAP, 2019) tem como finalidade coletar e disponibilizar dados geográficos de forma colaborativa e gratuita, a partir de conhecimento local de seus usuários. Com isto, torna-se uma alternativa interessante para o mapeamento municipal, onde a própria população pode contribuir, mesmo sem formação em cartografia. No entanto, para que os dados possam ser utilizados num processo de tomada de decisão, não basta fazer uma exportação direta dos dados para um banco de dados geográficos, é necessário que esses dados sejam estruturados adequadamente para serem utilizados em análises espaciais por meio de um software de SIG.

O QGIS (QGIS Development, 2019), por exemplo, é um software livre capaz de ler o arquivo exportado pelo OSM, permite a visualização e manipulação dos dados, contudo as tabelas de atributos são geradas de acordo com os tipos geográficos de cada elemento, dessa maneira os elementos do tipo Ponto, por exemplo, são adicionados numa mesma tabela, causando problemas de normalização de dados. Um banco de dados mal estruturado dificulta a compreensão dos dados adquiridos.

## **1.2 Objetivos**

O objetivo geral desta pesquisa é automatizar a geração de uma base de dados geográfica, propondo um método que utilize os dados obtidos do sistema de informação geográfica voluntária OpenStreetMap, a fim de auxiliar na gestão municipal de pequenos e médios municípios.

Especificadamente, pretende-se:

- Estudar a estrutura do arquivo de metadados exportado pela plataforma OpenStreetMap;
- Processar o arquivo de metadados exportado pelo OpenStreetMap para gerar o esquema conceitual dos dados e obter os dados no formato de arquivo Shapefile;
- Propor e implementar um método que automatize o processo de geração da base de dados geográficos para uma região selecionada no OpenStreetMap;
- Implementar um software visualizador de dados que facilite a consulta à base de dados gerada por usuários não especialistas no uso de software SIG;
- Desenvolver um estudo de caso que envolva uma aplicação referente à gestão de municípios a fim de verificar a eficácia do método proposto.

### **1.3 Estrutura da Dissertação**

Os próximos capítulos desta dissertação estão organizados como segue: O Capítulo 2 apresenta a fundamentação teórica, onde são discutidos o OpenStreetMap, informações geográficas voluntárias, representação e modelagem de dados espaciais e outros; O Capítulo 3 apresenta o método de geração automática de banco de dados geográficos a partir de dados exportados do OSM; O Capítulo 4 descreve a ferramenta OSM2Diagram, a qual implementa o método proposto e mostra sua aplicabilidade por meio de um estudo de caso utilizando dados da cidade de Ubá - MG, Brasil e; o Capítulo 5 apresenta as conclusões gerais e possíveis trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos da área de Banco de Dados Geográficos que são utilizados neste trabalho. A seção 2.1 faz uma breve caracterização da Web 2.0. A seção 2.2 expõe os conceitos sobre VGI e algumas de suas aplicações. A seção 2.3 caracteriza a plataforma *OpenStreetMap*. A seção 2.4 apresenta os tipos de representação de dados espaciais utilizados durante a interpretação dos fenômenos geográficos. A seção 2.5 define modelagem conceitual de dados geográficos com destaque para o modelo UML-GeoFrame. A seção 2.6 contextualiza alguns tipos de *software* utilizados para tomada de decisão com base em informação espacial, como SIG, WebMap e WebGIS.

### 2.1 Web 2.0

Antes da massiva popularização da Internet que vivemos hoje, os sites eram apresentados em um formato similar à de um livro, os usuários o acessavam, analisavam, observam o que era de seu interesse e fechavam. Entretanto com o passar do tempo a interação entre o usuário e a Internet foi estruturada, dinamizada e novas formas de interação surgiram. Dessa evolução surgiu um novo paradigma de utilização, em que os usuários de forma voluntária são capazes de contribuir com suas informações. Essas mudanças levaram a Internet a passar por uma revolução, assim o termo Web 2.0 foi usado pela primeira vez em 2004, pelo chefe executivo da O'Reilly Media, Tim O'Reilly, em seu artigo "Whats is Web 2.0", tornando-se bastante popular desde então (O'REILLY, 2005).

A nova Web tem como objetivo fornecer uma navegação que proporcione ao internauta uma imersão mais satisfatória, com uma navegação criativa que permite o compartilhamento de informação e a colaboração entre eles. Sendo assim, os serviços de relacionamentos sociais, páginas com vídeos, gifs, wikis, blogs e outros começam a dominar a rede mundial de computadores com uma participação massiva do internauta na geração de conteúdo dinâmico, seja ela por meio de um comentário, avaliação ou uma personalização.

Além disso é possível utilizar sistemas de *software* como, por exemplo, editores de foto e texto, proporcionando um maior conforto ao usuário por evitar a instalação de um programa ou possuir uma máquina com configuração específica.

### 2.2 Informação Geográfica Voluntária

A popularização da Web 2.0 despertou um interesse nas pessoas por serviços que permitem a produção de conteúdo na Internet e poder compartilhá-lo, com isso surgiu o

fenômeno chamado “Conteúdo Gerado por Usuários” (UGC) (KRUMM; DAVIES; NARAYANASWAMI, 2008). Por exemplo, a Wikipédia é uma enciclopédia livre virtual que é mantida por dados voluntários de pessoas do mundo todo. Então, partindo do princípio do conceito de Wiki, o site possui um grande acervo de conteúdo sobre temas diversos e todos os dias centenas de colaboradores editam e criam uma vasta quantidade de novos artigos.

Diferentes sites começaram a surgir compartilhando da ideia da colaboração do usuário na criação e manutenção de seu acervo, que por sua vez pode ser constituído de dados geográficos. Como aparelhos eletrônicos equipados com GPS se tornaram padrão de fábrica, torna-se fácil o compartilhamento e a coleta de VGI.

Segundo Goodchild (2007), quando um UGC possui dados sobre latitude e longitude, este UGC se caracteriza como uma VGI. Logo é possível adquirir conhecimento sobre detalhes de uma região desconhecida como hidrografia ou a rede de transporte de uma cidade. Entretanto, o conteúdo gerado pode vir de usuários que não possuem um conhecimento em cartografia ou SIG, mesmo assim a VGI se mostrou adequada para mapeamento simples, em regiões bem conhecidas pelos colaboradores, como ruas ou edificações (GOODCHILD, 2007).

Pode ser citado como exemplo de sistema VGI a plataforma OpenStreetMap (OSM), descrita na próxima seção, que é um site onde usuários voluntários podem mapear qualquer lugar do planeta Terra, a partir de imagens de satélites de alta resolução. A plataforma OSM permite que os mapas possam ser exportados e utilizados em ferramentas SIG. Diversas cidades europeias utilizam essa ferramenta como fonte de geração de mapas comerciais (DA SILVA e DAVIES, 2008).

Atualmente, VGI é utilizada em diversos sistemas de rotas e tráfego terrestre como, por exemplo, o sistema de navegação Waze<sup>3</sup>. O sistema verifica constantemente o posicionamento do dispositivo móvel do motorista, a fim de calcular a velocidade do automóvel e, através dela, informar como está a condição do trânsito, além de oferecer alertas sobre acidentes, bloqueios policiais, alagamentos e outros.

### **2.3 OpenStreetMap**

A plataforma OpenStreetMap (OSM), lançada em 9 de agosto de 2004, é um projeto colaborativo que tem como objetivo permitir que usuários possam fazer o mapeamento de qualquer região, bem como de qualquer país, de forma livre e voluntária (OPENSTREETMAP, 2019). A plataforma disponibiliza seus dados com base na *Open Database License* (ODbL,

---

<sup>3</sup> <https://www.waze.com/>

2018), logo torna possível exportar e inseri-los em SIG e em Sistemas Gerenciadores de Banco de Dados (SGBD) com suporte a dados espaciais, permitindo assim a sua ampla utilização.

Na maioria dos sistemas colaborativos o usuário tem a liberdade de criar conteúdo, adicionar algumas marcações *tags* que tenham relação com o conteúdo e compartilhar com outros usuários. A plataforma OSM possui um sistema de marcação para os elementos em mapeamento, onde características importantes para o entendimento destes podem ser inseridas como, por exemplo, informar que o item mapeado é um hospital que possui 5 andares e um heliporto (BALLATORE e MOONEY, 2015). O *Map Feature*<sup>4</sup> disponibilizado pela plataforma OSM, descreve e ilustra cada *tag* disponível, de modo a aumentar a probabilidade de que os dados contribuídos possam ser disponibilizados de forma correta por ferramentas de visualização de mapas que utilizam a plataforma.

Os usuários conseguem exportar os dados de qualquer área selecionada na plataforma. Os dados são exportados no formato de um arquivo OSM-XML, que possui 3 tipos de objetos fundamentais: *node*, *way* e *relation* (MOONEY e CORCOCAN, 2012). O tipo *node* representa um ponto, definido apenas com um par de coordenadas, que é utilizado para representar objetos pontuais, como uma parada de ônibus, o local de um semáforo ou um monumento. Os objetos do tipo *way* são utilizados para representar estruturas lineares (polilinha), como ruas, estradas, cursos d'água ou regiões fechadas (polígono), como edificações e limites. Os objetos do tipo *relation* representam o relacionamento entre os elementos anteriores, podendo ser uma restrição, por exemplo, informando locais onde o acesso de veículos é restrito, ou informar multipolígonos, por exemplo, para indicar que um conjunto de edificações faz parte de um mesmo condomínio. Além dos elementos e suas características, um arquivo OSM-XML armazena a linha do tempo de atualizações que cada elemento obteve, com data e usuário responsável pelas alterações.

As aplicações envolvendo VGI foram capazes de mapear 29% do território inglês em apenas 4 anos, mas o grande destaque se dá à qualidade das informações, que são cerca de 80% similar com bases cartográficas de agências nacionais (HAKLAY e WEBER, 2008).

De acordo com Kitchin (2014), existe um grande volume de dados produzidos por meio de censos nacionais e registros governamentais sobre os municípios e seus cidadãos. No entanto, esses dados são baseados em levantamentos por amostragens e normalmente não há continuidade desses levantamentos, outro problema a ser elencado é a existência de restrições ao acesso dos dados. Como resultado, esses volumes de dados precisam ser complementados

---

<sup>4</sup> [https://wiki.openstreetmap.org/wiki/Map\\_Features](https://wiki.openstreetmap.org/wiki/Map_Features)

pelo que pode ser chamado de estudos *Small Data*. Aplicação de questionários, estudos de caso e entrevistas são usados para capturar detalhes pontuais sobre problemas relativos ao município.

Muito do que se conhece sobre as cidades atualmente, foi obtido a partir de estudos caracterizados por escassez de dados. Por outro lado, a plataforma OpenStreetMap busca oferecer um entendimento de controle urbano mais amplo e às vezes em tempo real, sendo caracterizada como um projeto de Big Data, por conter característica como: grande volume de dados e a velocidade com que esses dados são gerados (MILLER, 2010).

Segundo Neis e Zipf (2012), a comunidade de usuários do OSM crescia cerca de 10% por mês, no início de 2008 eram 20.000 membros, entretanto a marca de 200.000 foi batida em janeiro de 2010 e em 2012 existiam mais de 500.000 contribuintes, dos quais mais de 16.000 eram membros altamente ativos. Portanto os dados são revistos e corrigidos por dezenas de usuários, especialmente em áreas urbanas e em 2018 a plataforma alcançou a marca de 1 milhão de contribuintes.

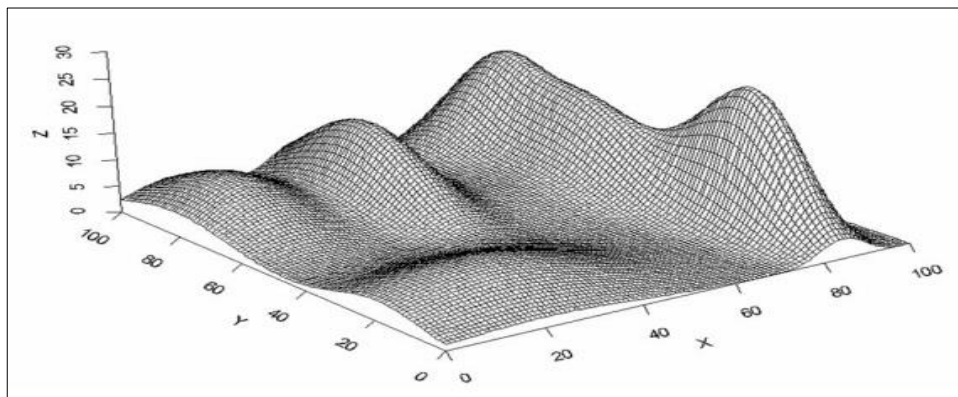
## **2.4 Representação de Dados Espaciais**

Em um banco de dados geográfico, além de conter dados sobre fenômenos espaciais, é comum que existam objetos convencionais. Uma edificação é considerada um objeto geográfico caso seja armazenada a sua localização (latitude e longitude), mas ao armazenar no banco de dados um objeto com as informações de cada morador, tem-se um objeto convencional, por não possuir característica espaciais (LISBOA FILHO; IOCHPE, 1999).

Segundo Longley et al. (2013), dado geográfico pode ser classificado de 2 formas: na visão de campos contínuos e na visão de objetos discretos.

No modelo de campo, o mundo real é visto como um plano contínuo, onde as entidades geográficas variam entre si, a qual geralmente está relacionada a fenômenos físicos, naturais ou biológicos, como o tipo de solo, topografia, temperatura e teor de minerais. Geralmente em sua representação são utilizadas polilinhas, isolinhas ou interpolação de triângulos, a Figura 1 retrata a representação do relevo de uma região fictícia.

Figura 1 - Relevo de uma região fictícia



Fonte: CÂMARA (2005)

Já o modelo de objeto é a representação de entidades individualizáveis do mundo real, onde cada entidade possui uma posição e características próprias, como lotes, edifícios e postes de uma rede elétrica. Como ilustrado na Figura 2, onde estão representados os limites territoriais de cada região de planejamento do estado de Minas Gerais, cada região é considerada um objeto discreto. As formas geométricas primitivas do tipo Ponto, Linha e Polígono são usadas para representação de dados do tipo objeto discreto.

Figura 2 - Regiões de Planejamento de Minas Gerais



Fonte: GOVERNO DE MINAS GERAIS (2019)

Entretanto a escolha das primitivas vai depender da escala utilizada, pois em uma escala grande um edifício pode ser representado por um Polígono, já em uma escala pequena ele pode ser representado por um Ponto. Logo a primitiva Ponto tende a representar objetos como ponto de ônibus, semáforo, hidrante entre outros. As primitivas Linhas tendem a representações de fenômenos lineares como ruas ou linhas de transmissão. Já os Polígonos são utilizados para representar fenômenos geográficos como lagos e áreas de preservação ambiental.

As representações de objetos geográficos apresentadas nessa seção são conceitos abstratos referentes à maneira de como a realidade pode ser interpretada e representada pelo usuário.

## **2.5 Modelos Conceituais para Banco de Dados Geográficos**

Um projeto de banco de dados geográficos necessita de modelos específicos para sua modelagem devido à complexidade dos dados geográficos, como visto na seção anterior, por não utilizarem apenas dados alfanuméricos em sua representação e podendo o mesmo fenômeno geográfico ser representado de várias formas diferentes, dependendo de como o usuário interpreta a realidade. Na área de banco de dados geográficos, modelos conceituais específicos vêm sendo desenvolvidos para melhor satisfazer as aplicações de SIG (STEMPLIUC, 2008).

Portanto, um modelo conceitual de banco de dados geográfico tem como objetivo fornecer construtores que sejam capazes de representar as necessidades dos usuários. Anteriormente, quando não existiam modelos para dados geográficos, os usuários eram obrigados a converter o modo como a realidade era percebida em estruturas internas dos SIG, gerando uma dificuldade em produzir bons esquemas conceituais (BORGES, DAVIS e LAENDER, 2005). Atualmente os modelos conceituais vêm sendo refinados para atender as particularidades encontradas na modelagem de aplicações SIG, como: permitir a especificação de restrições de integridade espaciais, temporais e semânticas; a representação de relacionamento entre entidades geográficas, além da abstração de fenômenos geográficos em relação as suas dimensões espaciais e temporais.

Dentre os modelos específicos para modelagem espacial, destacam-se aqueles que satisfazem a maioria dos requisitos geográficos, como, múltiplas representações e construtores de redes. Entretanto, ao satisfazer a maioria dos requisitos, tais modelos podem deixar os esquemas complexos e de difícil utilização. Por isso, é importante destacar o equilíbrio entre a abrangência e a facilidade de seu uso (FRIISCHRISTENSEN, TRYFONA e JENSEN, 2001). Atualmente não existe um modelo conceitual padrão, mas alguns modelos conceituais são bastante conhecidos, entre eles pode-se citar o modelo GeoOOA (KÖSTERS; PAGEL; SIX, 1997), o modelo OMT-G (BORGES; DAVIS; LAENDER, 2001), o modelo MADS (PARENT;

SPACCAPIETRA; ZIMÁNUI, 2008) e o modelo UML-GeoFrame (LISBOA FILHO; IOCHPE, 2008), o qual foi utilizado neste trabalho e está descrito com mais detalhes na seção seguinte.

### 2.5.1 Modelo UML-GeoFrame

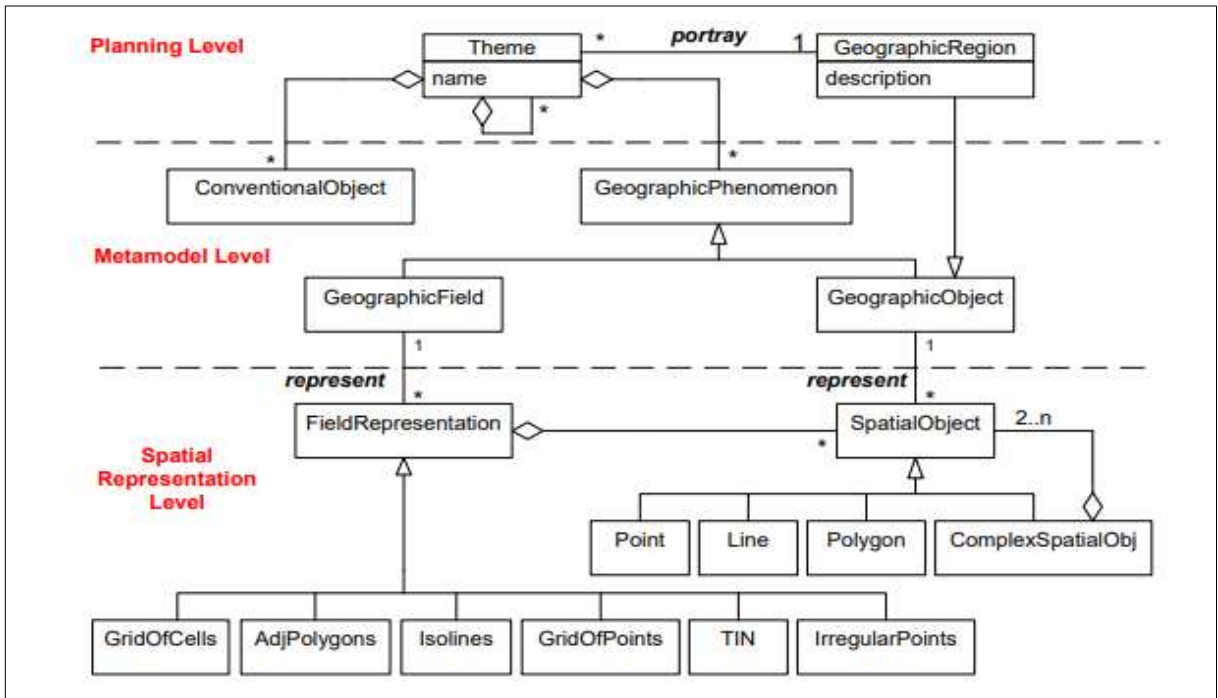
O modelo UML-GeoFrame (LISBOA FILHO e IOCHPE, 2008) segue o paradigma de orientação a objetos, em que suas classes são organizadas hierarquicamente com base no *framework* conceitual Geoframe (LISBOA FILHO e IOCHPE, 1999), o qual disponibiliza os construtores fundamentais para a modelagem de um banco de dados geográficos e propõem uma linguagem de fácil entendimento por conta de sua combinação com a UML.

A hierarquia de classes do Geoframe está dividida em 3 níveis de abstração: *Planning*, *Metamodel* e *Spatial Representation*, como ilustra a Figura 3. O nível mais alto de abstração, o de Planejamento, é onde se encontra a base de toda aplicação geográfica, onde cada região geográfica de interesse na aplicação pode conter vários temas em uma hierárquica de subtemas como, por exemplo, hidrografia, relevo e vias. Neste nível, não existe a preocupação de como os objetos devem ser representados.

O nível intermediário, de metamodelo, contém a classe de objetos convencionais, objetos que não possuem característica geográfica e a classe de fenômeno geográfico que generaliza os objetos na visão de campo e na visão de objeto. Neste momento, as classes começam a refletir o modo como a realidade é interpretada.

O terceiro nível é onde se encontra as demais classes responsáveis por representar os fenômenos geográficos. A classe *SpatialObject* generaliza as classes capazes de representar os fenômenos geográficos na visão de objetos e a classe *FieldRepresentation* generaliza as classes de representação de fenômenos na visão de campo.

Figura 3 - Estrutura do framework UML-GeoFrame



Fonte: Lisboa Filho e Iochpe (2008)

Lisboa Filho e Iochpe (2008) recomendam a utilização de cinco passos para utilizar o modelo UML-GeoFrame na modelagem de bancos de dados geográficos. Por meio desses passos é percebido o modo de como os elementos do GeoFrame são interligados aos construtores do digrama de classes da UML. A Figura 4 ilustra os elementos que podem ser utilizados na modelagem e seus respectivos estereótipos.

Figura 4 - Estereótipos de framework UML-GeoFrame

<i>Fenômeno geográfico e Objeto convencional</i>	<i>Componente espacial de objetos geográficos</i>	<i>Componente espacial de campos geográficos</i>
<ul style="list-style-type: none"> <li> Objeto geográfico</li> <li> Campo geográfico</li> <li> Objeto não geográfico</li> </ul>	<ul style="list-style-type: none"> <li> Ponto</li> <li> Linha</li> <li> Polígono</li> <li> Obj. espacial complexo</li> </ul>	<ul style="list-style-type: none"> <li> Pontos irregulares</li> <li> Grade de pontos</li> <li> Polígonos adjacentes</li> <li> Isolinhas</li> <li> Grade de células</li> <li> TIN</li> </ul>
<p>&lt;&lt;função&gt;&gt; <i>função categórica</i></p>		

Fonte: Lisboa Filho et al. (2004)

Aplicado no nível de planejamento, o primeiro passo é identificar as regiões geográficas consideradas relevantes para a aplicação SIG. Nesse momento, são identificados também os

temas relacionados com as regiões escolhidas e modelados utilizando os construtores da UML. Dessa maneira, inicia-se a utilização de uma abordagem Top-Down e o particionamento do problema em problema menores, facilitando o processo de modelagem e entendimento para os envolvidos no projeto.

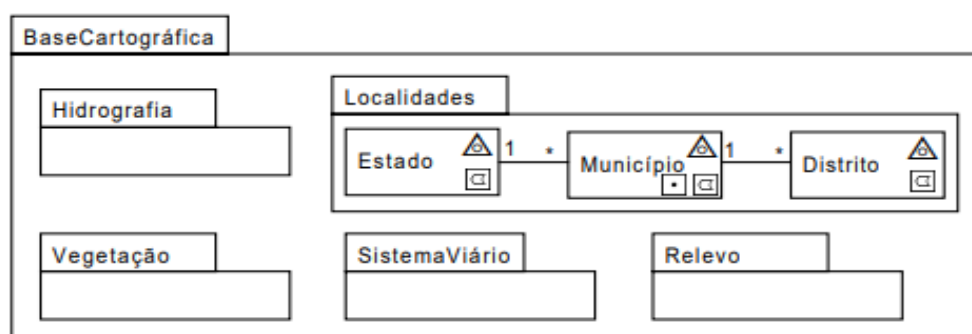
No segundo passo, o qual é aplicado no nível de metamodelo, são escolhidos os objetos essenciais do projeto de cada tema definido no passo anterior e classificados em convencional e fenômeno geográfico na visão de campo ou de objeto. As associações entre as classes são feitas nesse passo, sejam elas pertencentes ao mesmo tema ou não, mas sem considerar as restrições espaciais.

O terceiro passo está relacionado com a representação espacial, onde são definidos os estereótipos de cada objeto, a fim de especificar a forma de representar cada fenômeno geográfico.

O quarto passo é responsável por identificar os relacionamentos entre as classes identificadas, classificando os relacionamentos em semânticos ou espaciais.

A Figura 5 ilustra um exemplo simples de esquema conceitual UML-GeoFrame após os 4 primeiros passos, onde tem-se a região de interesse e seus respectivos temas utilizando os construtores da UML. O Tema Localidades possui as classes Estado e Distrito, classificadas como polígono, e a classe Município utilizando da múltipla representação, sendo classificada como ponto e polígono. Estas três classes são representadas na visão de objeto e apresentam seus respectivos relacionamentos.

Figura 5 - Utilização de temas no modelo UML-GeoFrame



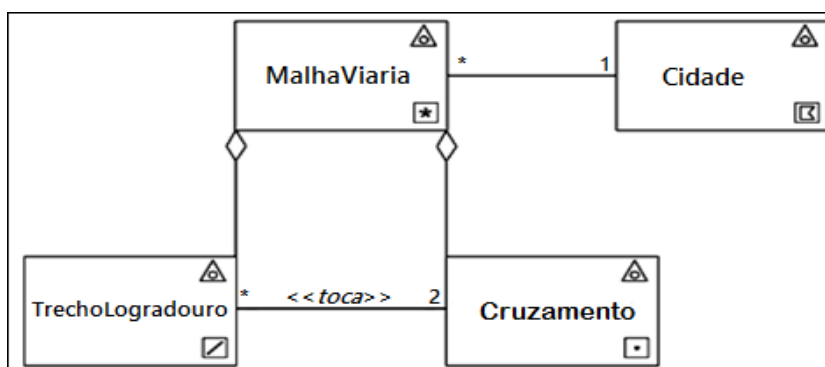
Fonte: Lisboa Filho e Iochpe (1999)

No quinto e último passo, são modelados os aspectos temporais relacionados as classes convencionais ou geográficas, caso existam. O objetivo da temporalidade é expressar as mudanças ocorridas ao longo do tempo nos dados armazenados.

A Figura 5 representa um esquema conceitual que retrata uma malha viária, onde é

possível observar a utilização de construtores espaciais básicos do UML-GeoFrame e um relacionamento do tipo espacial “<<toca>>” entre *TrechoLogradouro* e *Cruzamento*, este informa que um *TrechoLogradouro* toca em 2 ou mais *Cruzamento* e que N *Cruzamento* tocam em um *TrechoLogradouro*.

Figura 6 - Modelagem conceitual utilizando o modelo UML-GeoFrame



Fonte: Lisboa Filho et al. (2000)

## 2.6 Tipos de software para tomada de decisão usando informação espacial

A análise de dados espaciais normalmente é realizada utilizando-se sistemas de *software* com capacidade para processar dados estruturados com base em geometrias envolvendo coordenadas espaciais. Exemplos desses tipos de sistemas incluem SIG desktop, WebGIS e WebMaps.

O SIG desktop reúne um conjunto de ferramentas desenvolvidas para representar e manipular grandes volumes de dados espaciais armazenados ou não em um banco de dados geográfico. Este tipo de *software* tem sido utilizado por vários setores que tratam de questões ambientais e urbanas para planejamentos e fiscalizações, por proporcionar uma avaliação integrada de muitas variáveis e gerando informações que são usadas nos processos de tomada de decisão.

O *software* QGIS é um exemplo de SIG distribuído com base na licença GPL (*GNU General Public License*) e com uma interface gráfica amigável. Possui extensões para conexão com SGBDs e permite a realização de consultas SQL no formato padrão ou em uma linguagem de alto nível disponibilizada pelo QGIS. Caso seja alterado algum dado através do *software*, o mesmo pode ser persistido na base de dados se for da vontade do usuário. Portanto, é possível gerenciar o banco de dados pelo QGIS, aumentando a abstração com o usuário final.

Um sistema WebGIS é basicamente um SIG rodando na Web, o qual permite utilizar

ferramentas de geoprocessamento para analisar os dados cartográficos, além de possibilitar o uso de ferramentas básicas de interação como movimentação, zoom, definição das camadas que serão apresentadas, consultas a atributos da base de dados, entre outras (HEDA & CHIKURDE, 2016). Já um sistema WebMap é um visualizador de mapas por meio de navegadores Web, o qual possibilita ao usuário a visualização e a interação com o mapa por meio de ferramentas básicas como, zoom, definição de camadas e outras semelhantes ao do WebGIS (JOHANSSON et al., 2009).

Sistemas WebGIS e WebMap normalmente são desenvolvidos com base em uma arquitetura cliente-servidor<sup>5</sup>. Esses servidores de dados espaciais são responsáveis pela publicação de mapas interativos na Web. Para a criação desse tipo de aplicação, entretanto, é preciso programar a interface e as ferramentas que ficarão disponíveis. Existem diversos frameworks que auxiliam no desenvolvimento de sistemas WebGIS ou WebMap como, por exemplo, MapServer<sup>6</sup>, Geoserver<sup>7</sup>, Geonode<sup>8</sup> e Leaflet (LEAFLET, 2019). Este trabalho utiliza apenas a biblioteca Leaflet, portanto apenas este software será detalhado.

A Leaflet é uma biblioteca de código aberto desenvolvida em *javascript* projetada para criar mapas interativos como WebGIS e WebMap. A biblioteca disponibiliza diversas ferramentas de geoprocessamento como cálculo de distância entre pontos, interseção entre camadas, cálculo de área, cálculo de buffer e outros, além das ferramentas de interação básica (LEAFLET, 2019).

---

<sup>5</sup> A máquina titulada cliente faz solicitações de conteúdo a um servidor e o servidor tem como responsabilidade prover ou não esse conteúdo ao cliente.

<sup>6</sup> <https://mapserver.org/>

<sup>7</sup> <http://geoserver.org/>

<sup>8</sup> <http://geonode.org/>

### 3 MÉTODO PARA GERAÇÃO DE BASE DE DADOS GEOGRÁFICA

Como este trabalho tem como objetivo permitir que cidades de pequeno e médio porte, tenham acesso a dados geográficos que os ajudem a tomar decisões, a utilização de VGI e SIG é fundamental para manipulação desses dados. Portanto, este capítulo apresenta as etapas do método proposto para realizar a extração dos dados da plataforma OpenStreetMap, gerando uma base de dados geográficos, que pode ser utilizada em sistemas de software SIG. O método consiste em três etapas:

1. Elaboração do esquema conceitual completo da plataforma OSM;
2. Geração de esquemas conceituais a partir do processamento dos dados exportados da plataforma OSM, tendo como base o esquema OSM da etapa 1;
3. Geração de arquivos no formato Shapefile.

Para a primeira etapa, foi necessário realizar uma revisão da literatura para descobrir todo o potencial da plataforma OSM em representar fenômenos geográficos e propor um esquema conceitual geral que sirva de base para os esquemas conceituais que forem gerados utilizando a ferramenta proposta. Essa primeira etapa é detalhada na seção 3.1.

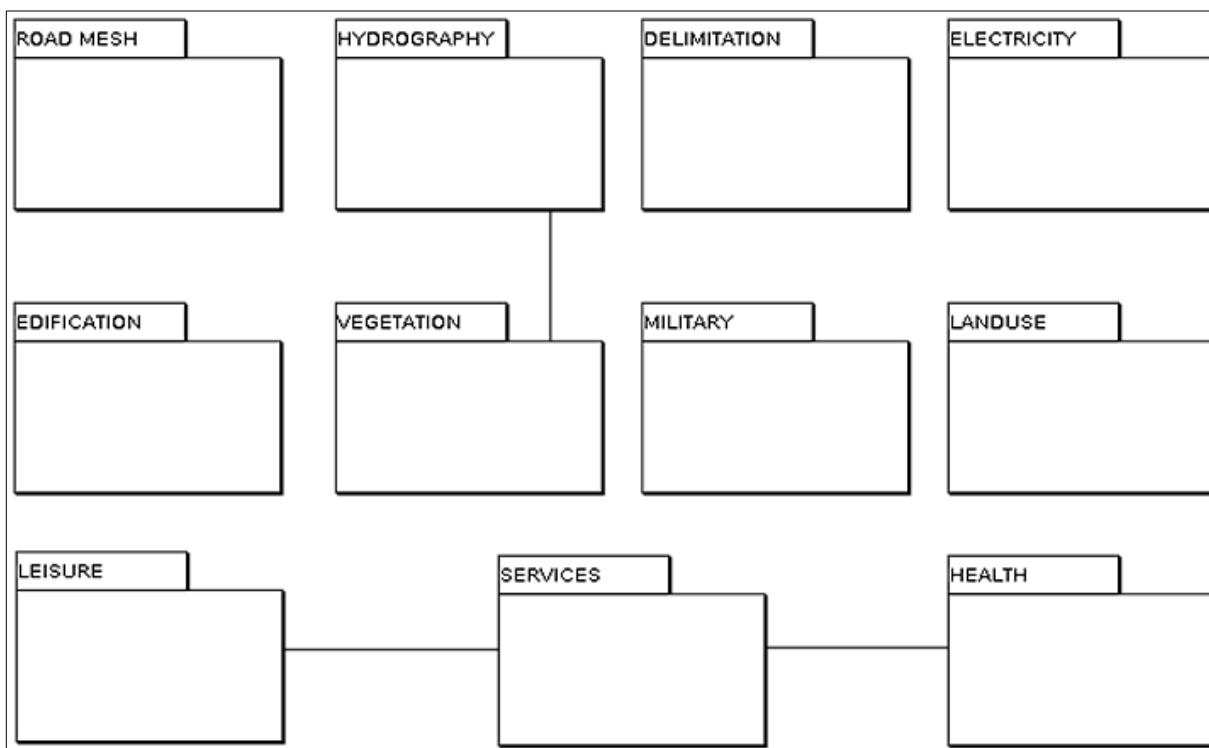
A partir dos dados exportados da plataforma OSM, a segunda etapa (detalhada na seção 3.2) realiza um processo de engenharia reversa e transforma os metadados (*tags*) presentes no arquivo OSM-XML em esquemas conceituais, além de gerar um relatório de dados mapeados com informações incompletas.

Por fim, a última seção descreve o processo de transformação dos dados em arquivos Shapefile, feito com o auxílio da biblioteca GDAL (*Geospatial Data Abstraction Library*) e, assim, permitindo a utilização dos dados em SIG.

#### 3.1 Modelagem Conceitual da Plataforma OpenStreetMap

A primeira etapa do método consiste na modelagem conceitual completa de todas as classes de objetos que podem ser mapeados no OSM. A modelagem conceitual foi realizada utilizando o método proposto no modelo UML-GeoFrame, descrito na seção 2.4.4. A Figura 7 ilustra o diagrama de temas resultante. Como é utilizada uma abordagem *Top-Down*, o propósito é abstrair o conteúdo interno de cada tema, facilitando o entendimento do minimundo. Ao todo foram identificados 11 temas e seus respectivos relacionamentos.

Figura 7 - Diagrama de temas dos objetos disponíveis na plataforma OSM

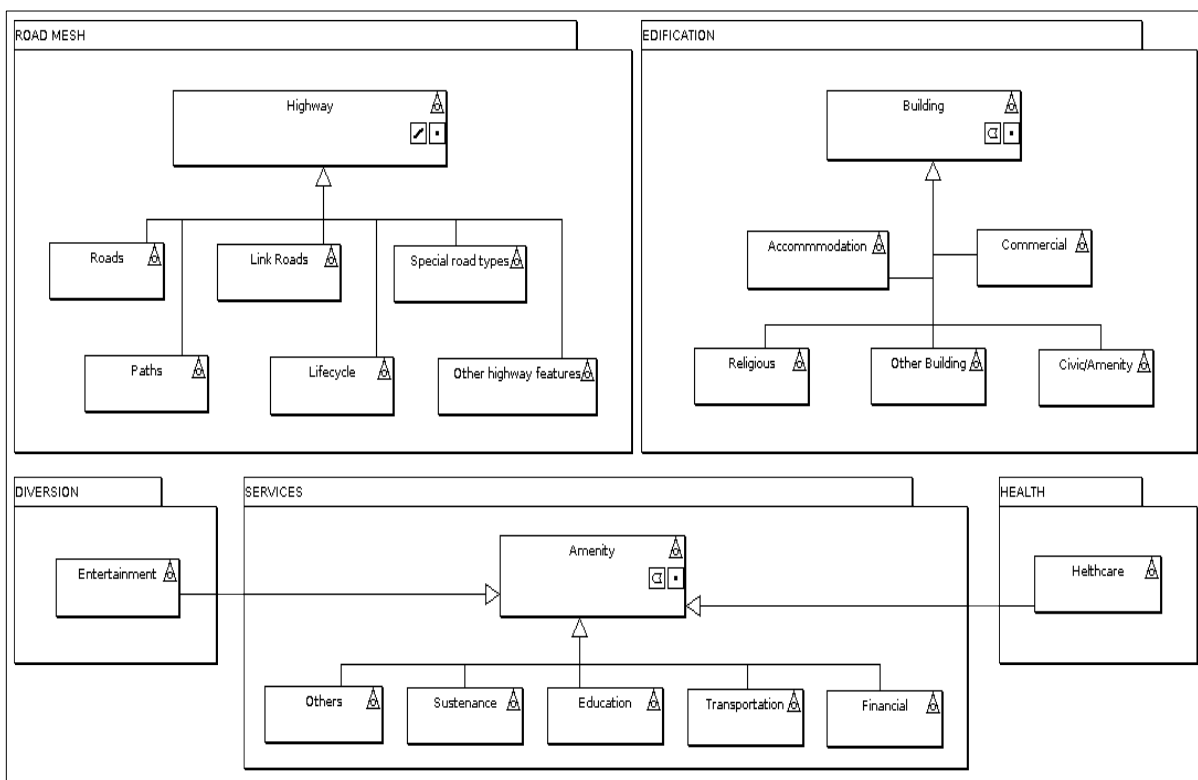


A plataforma OSM disponibiliza uma documentação *online*. No site OSM Wiki<sup>9</sup> é possível acessar as informações sobre todas as *tags* disponíveis para representar fenômenos geográficos e os respectivos estereótipos aceitos por cada *tag*. Além de um livro de regras com sugestões e orientações de como atribuir uma característica a um objeto durante uma contribuição.

Dessa maneira, com o auxílio dessa documentação, para cada tema foi feita a modelagem das classes de objetos possíveis de serem mapeados na plataforma OSM, de acordo com o seu grau de afinidade. A Figura 8 ilustra um fragmento desse esquema onde é possível observar classes de objetos do mundo real separadas por temas. Além disso, exemplos de como ocorrem os relacionamentos entre temas podem ser vistos entre os temas “*Diversion*” e “*Services*” ou entre “*Services*” e “*Health*”. Importante ressaltar que cada classe possui muitas subclasses, que por limitação de espaço não é possível exibir todas elas. Por exemplo, a classe “*Transportation*”, que é subclasse da classe “*Amenity*”, possui ainda diversas subclasses (ex.: “*taxi*”, “*parking*”, “*fuel*”, “*car\_wash*”) que não foram incluídas na modelagem.

<sup>9</sup> <http://wiki.openstreetmap.org/wiki>

Figura 8 - Trecho do diagrama de classes de alguns temas



### 3.2 Geração do Esquema Conceitual relativo a uma Área Seleccionada

Nesta etapa de geração do esquema conceitual é realizado o processo de engenharia reversa, feito a partir do arquivo de metadados OSM-XML exportado da plataforma OSM, o qual corresponde à uma determinada área mapeada.

Inicialmente, com base na documentação disponível da plataforma OSM e no trabalho de Pruvost e Mooney (2017), foi realizado um estudo no arquivo de dados exportados, com a finalidade de entender a estruturação dos dados, complexidade, flexibilidade e seus relacionamento, como agrupamentos lógicos responsáveis por representar relações geográficas do mundo real.

Com a finalidade de explorar e mostrar a estrutura dos metadados no arquivo exportado pela plataforma OSM, seleccionou-se uma pequena área geográfica (Figura 9). Esta área corresponde a uma pequena parte do campus da Universidade Federal de Viçosa (UFV), localizado na cidade de Viçosa, Minas Gerais. O exemplo contém os elementos necessários para exemplificar a estrutura do arquivo OSM-XML, como, fragmentos de algumas vias (representadas como linha), alguns departamentos (representados como polígonos) e relacionamentos do tipo multipolígonos.

Figura 9 - Área para estudo do arquivo OSM-XML



Ao exportar os dados da área selecionada na Plataforma OpenStreetMap, o arquivo OSM-XML vem ordenado primeiramente por todas as *tags node*, seguido das *tags way* e por fim as *tags relation*. Todos os elementos possuem um identificador (id) que pode ser utilizado em relacionamentos ou dependências.

A *tag node* contém apenas um par de coordenadas, podendo ser associado ao tipo geométrico Ponto. Como toda forma geométrica em sua essência é formada por pontos, essa *tag* é utilizada também na construção de objetos do tipo Polígono (cada vértice sendo representada por uma *tag node*) e do tipo Linha (sendo uma sequência de alguns *tag node*, ligados por segmentos de reta). Por exemplo, no Código 1 a *tag node* refere-se ao par de coordenadas (latitude e longitude) do ponto mais à direita do polígono relativo ao “Departamento de Geografia” da Figura 9 e contém também informações sobre data e usuário voluntário que a criou.

Código 1 - Exemplo da *tag node* sem atributos

```
<node id="2799312298" visible="true" version="2" changeset="21773219" timestamp="2014-04-18T17:34:49Z" user="Roger Ricardo Pinto" uid="2037766" lat="-20.7620055" lon="-42.8644693"/>
```

Entretanto, pode ocorrer de uma *tag node* receber um atributo, como mostra o exemplo do Código 2. Neste caso, além de possuir um par de coordenadas, a *tag* possui 2 atributos, sendo o primeiro “*building*” com o valor “*true*” e o segundo atributo “*name*” com o valor de “Departamento de Geografia”, sempre seguindo o padrão do par <chave K, valor V>.

## Código 2 - Exemplo de *tag node* com atributos

```
<node id="3069059233" visible="true" version="2" changeset="49928688" timestamp="2017-06-29T19:49:36Z"
user="Valério Castro" uid="6237447" lat="-20.7610520" lon="-42.8648950">
  <tag k="building" v="yes" />
  <tag k="name" v="Departamento de Engenharia de Produção, Mecânica e Elétrica" />
</node>
```

A *tag way* possui estrutura similar à da *tag node* exemplificada anteriormente, a diferença está na inclusão de *tags* “nd” que fazem referência a *tags node* sem atributos. O Código 3 ilustra o elemento “Departamento de Geografia” e seus atributos, a primeira *tag* “nd” faz uma referência ao código de número “2799312298” que é o “id” do *node* descrito no Código 1. O elemento possui o último par de coordenadas igual ao primeiro para delimitar um polígono, por esse motivo a primeira *tag* “nd” é igual a última. Portanto, as vias representadas na Figura 9 mantém o padrão de estrutura da *tag way*, mas com a diferença do último par de coordenadas não ser igual ao primeiro por representar uma linha.

## Código 3 - Exemplo de *tag way*

```
<way id="275292897" visible="true" version="3" changeset="24152926" timestamp="2014-07-15T04:05:00Z"
user="Artur Vieira" uid="2180959">
  <nd ref="2799312298" />
  <nd ref="2799312299" />
  <nd ref="2799312300" />
  <nd ref="2799315601" />
  <nd ref="2799312298" />
  <tag k="building" v="yes" />
  <tag k="name" v="Departamento De Geografia" />
</way>
```

Diferente das duas *tags* apresentadas anteriormente, a *tag relation* (Código 4) não contém *tags* “nd”. Ela utiliza a *tag* “member”, que é semelhante às *tags* “nd”, pois elas referenciam *tags* do tipo “node” ou “way” através da propriedade “type” e passando o identificador (ref). O restante da estrutura refere-se aos seus atributos e ao final é informado o tipo do relacionamento. No Código 4, o relacionamento do tipo “multipolygon” é usado para registrar que os “Departamentos de Engenharia de Produção, Mecânica e Elétrica” correspondem a dois polígonos (ver Figura 9).

## Código 4 - Exemplo da *tag relation*

```
<relation id="4026400" visible="true" version="1" changeset="25336789" timestamp="2014-09-09T20:34:20Z"
user="Artur Vieira" uid="2180959">
  <member type="way" ref="302657450" role="outer" />
  <member type="way" ref="302657451" role="outer" />
  <tag k="building" v="yes" />
  <tag k="name" v="Departamentos de Engenharia de Produção, Mecânica e Elétrica" />
  <tag k="type" v="multipolygon" />
</relation>
```

A plataforma OSM ainda não disponibiliza muitos tipos de relacionamentos, no entanto o site OSM Wiki traz informações de quais estão disponíveis, quais estão sendo votados pela comunidade e quais estão em produção.

A Figura 10 apresenta o fluxograma simplificado do algoritmo de engenharia reversa proposto para geração do esquema conceitual, tendo como entrada o arquivo de metadados OSM-XML. O processo é iniciado com a leitura do arquivo OSM-XML, de onde são extraídas as informações de interesse. Neste passo, a biblioteca *BeautifulSoup* (BEAUTIFUL SOAP, 2018) é utilizada para facilitar a manipulação do arquivo XML e o *parser* é realizado pela biblioteca LXML (LXML, 2019).

Em seguida, os dados obtidos são testados a fim de separar os objetos mapeados que possuem informações incompletas dos demais. Durante a separação é verificado se o objeto possui um nome, caso não tenha, o objeto é armazenado em uma lista onde se encontram os objetos mapeados sem nome (esses arquivos também serão convertidos em Shapefile). Posteriormente, essa lista é salva em um arquivo texto no formato de relatório, contendo o estereótipo geográfico, as coordenadas e o respectivo retângulo envolvente dos objetos incompletos.

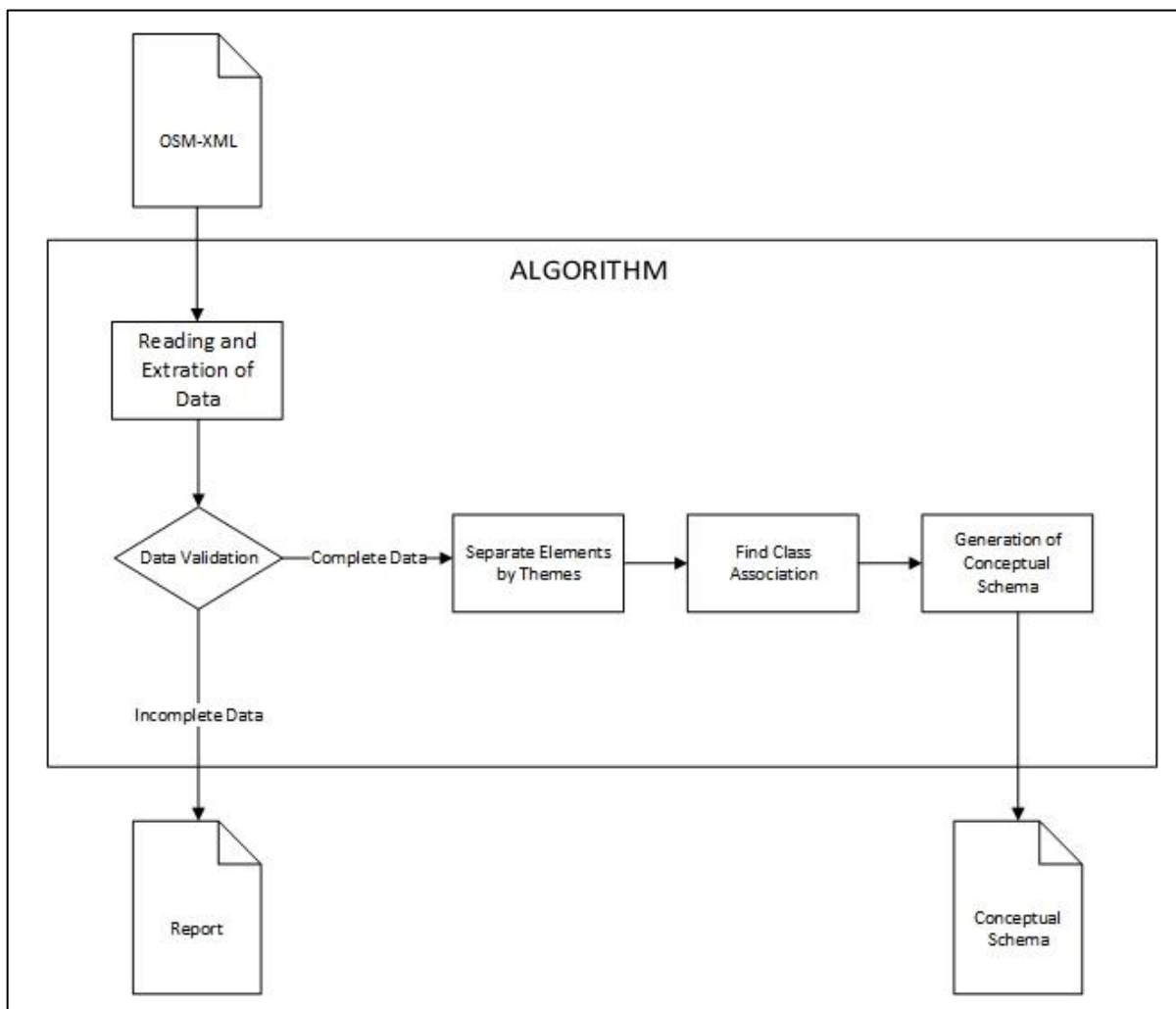
Os objetos válidos são separados em listas de acordo com os temas modelados (Figura 7), sendo que o arquivo OSM-XML disponibiliza a informação sobre qual superclasse cada objeto pertence. Por exemplo, o trecho do arquivo OSM-XML que descreve o objeto “*Hospital*” contém uma *tag* informando que ele pertence à superclasse “*Amenity*”, desse modo, o objeto “*Hospital*” será incluído na lista que contém os objetos do tema “*Services*”.

Embora o arquivo OSM-XML disponibilize um conjunto de informações (*tags*) sobre cada objeto, ainda assim não é suficiente para que seja possível classificar a subclasse exata dos objetos. Portanto é necessário fazer uma busca dentre as subclasses da superclasse informada a fim de encontrar a classe exata do objeto. Por exemplo, para o objeto “*Taxi*” o arquivo OSM-XML informa que sua superclasse é “*Amenity*”, desse modo é preciso pesquisar nas subclasses de “*Amenity*” (Figura 8) com a finalidade de encontrar a que melhor agrega o objeto “*Taxi*”. Dessa forma é possível criar as especializações de “*Taxi*” para “*Transportation*” e para “*Amenity*”.

Durante esta fase de busca de classes e relacionamentos, é gerado um arquivo que contém o *layout* do esquema conceitual, como cor, fonte, ícones e outros. Além dos objetos em forma de classes e suas conectividades. Ao final, esse arquivo é processado pelo *software* Graphviz (Graphviz, 2019), que é capaz de transformar um esquema conceitual textual em um esquema

gráfico, gerando assim o diagrama de classes UML.

Figura 10 – Fluxograma do módulo gerador do esquema conceitual



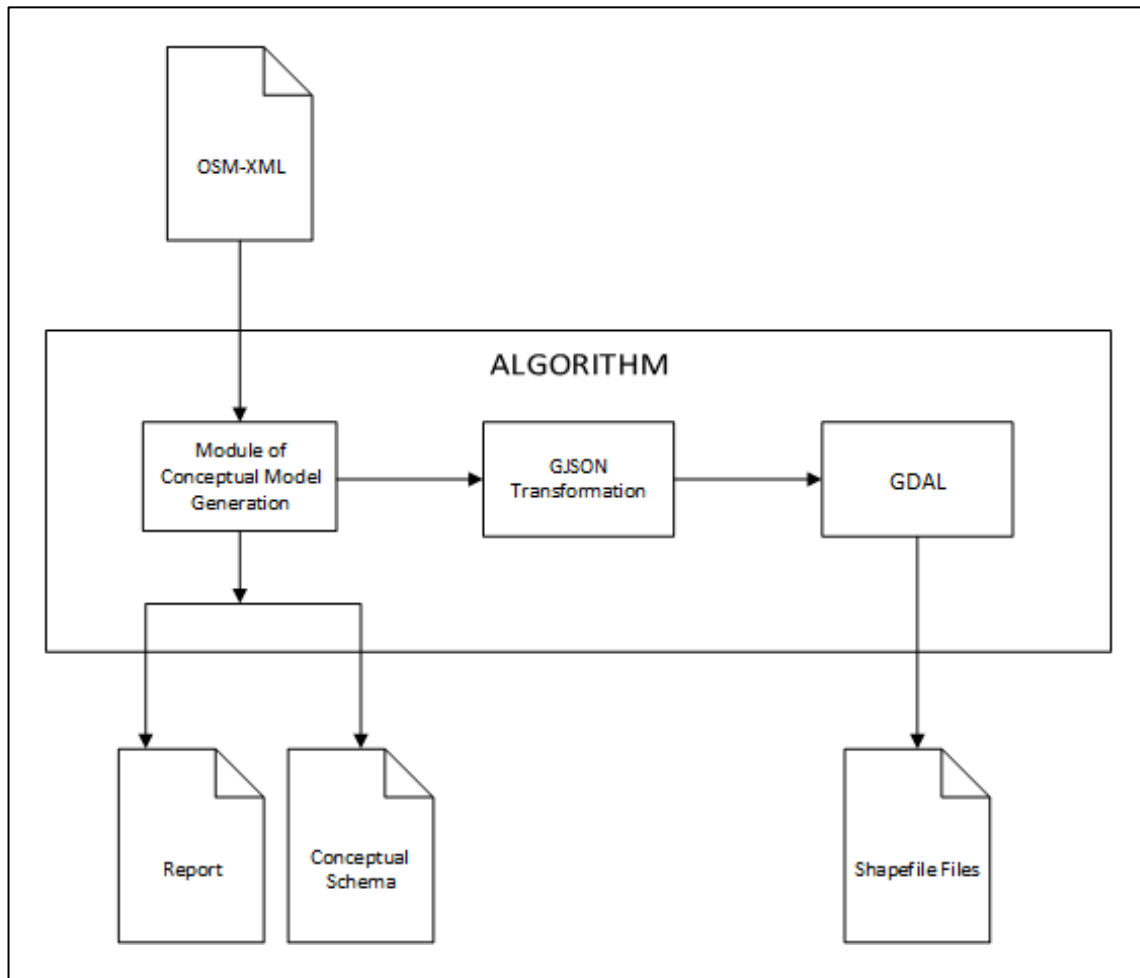
### 3.3 Geração de Arquivos Shapefile

Por fim, o terceiro módulo desenvolvido é responsável por fazer a geração dos arquivos Shapefile, cujo fluxograma pode ser visualizado na Figura 11. A primeira parte corresponde à execução do fluxograma anterior, o qual tem como entrada o arquivo OSM-XML, em sequência a realização dos processamentos essenciais nos metadados para que seja possível fazer a geração do esquema conceitual e do relatório com os objetos parcialmente mapeados. Portanto, este segundo fluxograma tem como entrada, os dados processados no módulo apresentado anteriormente.

Ao término da execução do módulo gerador de esquema conceitual, os dados se encontram armazenadas em uma estrutura de dados denominada dicionário, indexadas pelo

seu ID e ordenados de acordo com suas classes para facilitar os próximos passos. No primeiro momento é necessário estruturar os dados no formato GeoJson (BUTLER, 2019), este formato tem como base o JSON, suas especificações estão definidas no RFC 7946 e é um padrão utilizado para codificar uma variedade de estruturas de dados geográficos, como, pontos, multipontos, polilinhas e outras propriedades.

Figura 11 - Fluxograma do modulo gerador de Shapefile



No momento da estruturação, os arquivos do tipo GeoJson são criados para cada classe individualmente contendo todos suas entidades, entidades essas que representam os objetos do mundo real como, por exemplo, se um hospital aparece no mapa é criado o arquivo Hospital.geojson, que contém todos os hospitais da área exportada. Dado que o dicionário de objetos está ordenado pelo sua classe, a criação dos arquivos GeoJson e a inserção dos objetos no arquivo correto é feito através da verificação da classe do próximo objeto no dicionário, caso a classe dele seja diferente, o arquivo atual é finalizado e então é criado um novo arquivo com o nome da nova classe.

Como estão sendo criados arquivos, é importante ter atenção no tratamento de caracteres especiais inseridos pelo usuário durante o mapeamento, então é feito um tratamento minucioso de remoção e substituição desses caracteres.

A Figura 12 representa um trecho de um arquivo GeoJson retirado do arquivo OSM-XML exportado da cidade de Viçosa-MG, onde estão contidos os dados de todos os estabelecimentos considerados como “fast-food” no momento do mapeamento voluntário. Nessa imagem estão representados 3 estabelecimentos, sendo o primeiro mapeado como ponto, o segundo como polígono e o terceiro como ponto, seguidos de suas coordenadas. Em todo documento criado é mantida uma ordem em sua estruturação, começando pelo nome do objeto mapeado, seguido por seus atributos (caso tenha) e, por fim, as informações referentes a sua geometria.

Figura 12 - Exemplo do arquivo GeoJson de objetos mapeados como *fast-food* em Viçosa

```
{
  "type": "FeatureCollection",
  "features":
  [
    {
      "name": "Bar do DCE",
      "type": "Feature",
      "geometry": { "type": "Point", "coordinates": [ -42.8682665, -20.7621346 ] }
    },
    {
      "name": "Barzinho DCE",
      "type": "Feature",
      "geometry": { "type": "Polygon", "coordinates": [ [ [ -42.8684033, -20.7621808 ],
      [ -42.8682956, -20.7622624 ],
      [ -42.8680883, -20.7620230 ],
      [ -42.8681959, -20.7619415 ],
      [ -42.8684033, -20.7621808 ] ] ] }
    },
    {
      "name": "Burguer Max",
      "type": "Feature",
      "geometry": { "type": "Point", "coordinates": [ -42.8756805, -20.7569405 ] }
    }
  ]
}
```

Com todos os arquivos GeoJson criados e salvos em um diretório, a última etapa consiste na geração dos arquivos *Shapefile* e para isto é utilizada a biblioteca GDAL (WARMERDAM, 2008). Esta biblioteca possui ferramentas para tradução de formatos de dados geoespaciais raster e vetoriais liberada sob uma licença *Open Source* de estilo X/MIT<sup>10</sup> pela *Open Source Geospatial Foundation* (OSGeo).

<sup>10</sup> É uma licença permissiva utilizada tanto em software livre quanto em software proprietário.

#### 4 OSM2DIAGRAM – IMPLEMENTAÇÃO DO MÉTODO

Este capítulo descreve a ferramenta OSM2Diagram<sup>11</sup>, desenvolvida para mostrar a viabilidade do método de criação de base de dados geográficos para ser utilizada para auxiliar na tomada de decisão em cidades de pequeno e médio porte.

O esquema conceitual da plataforma OSM foi elaborado conforme descrito na seção 3.1 e deverá ser reproduzido novamente somente quando a plataforma receber atualizações, como o acréscimo de novos elementos para mapeamento.

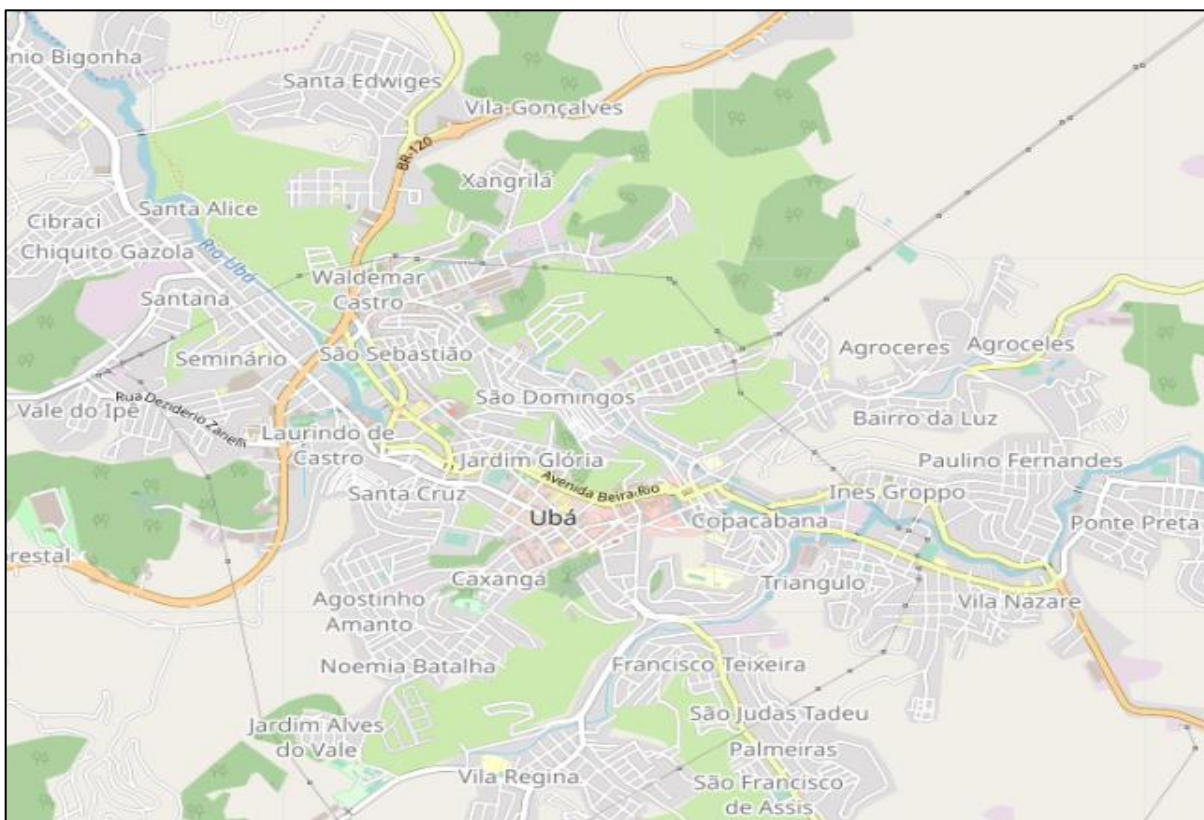
Os testes foram realizados com dados da cidade de Ubá-MG, a qual segundo o IBGE (2010) possui uma população aproximada de 100 mil habitantes e densidade demográfica de 249,16hab/km<sup>2</sup>. Utilizou-se um computador configurado com um processador Intel(R) core(TM) i5-8250U de 1.6GHz, 8GB de memória RAM, placa de vídeo Nvidia Geforce 930mx e Windows 10 Home de 64 bits. O tempo total de execução da ferramenta foi de 36.842 milissegundos e identificou 5.404 feições geográficas.

Inicialmente pensou-se em comparar o sistema com outras ferramentas disponibilizadas de modo online que são capazes de fazer a geração de arquivos Shapefile partindo de um arquivo OSM-XML como, por exemplo, o site <https://mygeodata.cloud/converter/osm-to-shp>, mas todas examinadas possuem restrição de tamanho de arquivo. Portanto, cidades com um certo número de objetos mapeados precisariam pagar para poder fazer a geração, esse número não é muito grande, arquivos OSM-XML com apenas 1 MB já não são convertidos de modo gratuito. A Figura 13 apresenta a parte central da área exportada da plataforma VGI OSM para a cidade de Ubá, o arquivo OSM que representa essa figura possui aproximadamente 9 MB.

---

<sup>11</sup> <http://www.dpi.ufv.br/projetos/OSM2diagram/>

Figura 13 - Perímetro urbano da cidade Ubá-MG



#### 4.1 Geração do Esquema Conceitual

A ferramenta OSM2Diagram segue o fluxograma apresentado na Figura 10 para realizar a geração do esquema conceitual, portanto a ferramenta tem como entrada um arquivo OSM-XML exportado pela plataforma OSM.

Após a ferramenta processar o arquivo e realizar as associações necessárias para definir cada objeto, o relatório e o esquema conceitual estão disponíveis para visualização. O relatório gerado para a cidade de Ubá, com as classes que foram identificadas com falta de informações<sup>12</sup> contém muitos elementos, produzindo uma listagem de quase 60 mil linhas. Este grande volume de elementos gráficos mapeados indica um problema de falta de conhecimento dos usuários, que demonstram importância apenas na parte gráfica do mapa e um descuido com a semântica desses elementos mapeados. Como exemplo, a Figura 14 apresenta um objeto do mundo real que foi mapeado como polígono, seguido de suas coordenadas e o respectivo retângulo envolvente que está contido no relatório. Porém, não se sabe qual objeto do mundo real este

<sup>12</sup> Os objetos que não possuem um nome são considerados como objetos com falta de informações, pois não possibilitam sua classificação.

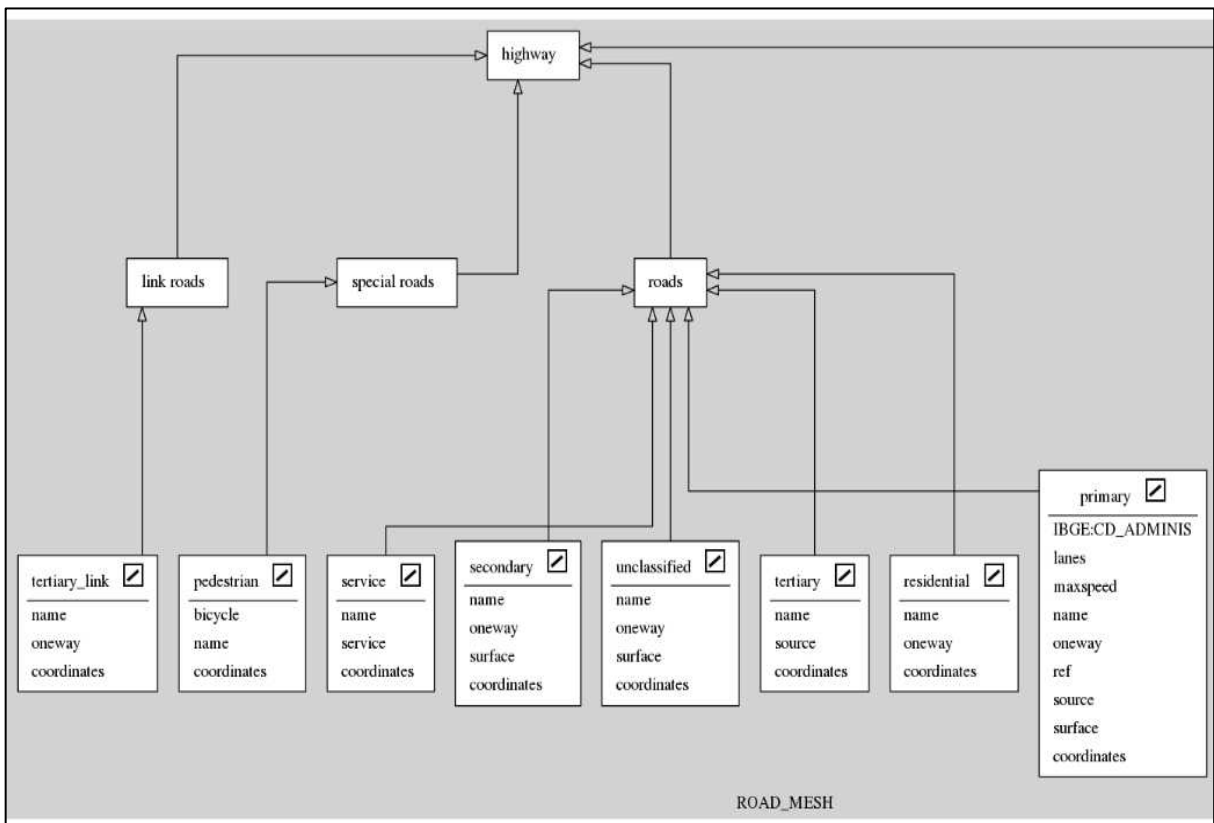
polígono representa.

Figura 14 - Trecho do Relatório das classes mapeadas com falta de informações

```
Polygon
-21.1098940, -42.9486764
-21.1099515, -42.9486716
-21.1099467, -42.9486047
-21.1098892, -42.9486095
-21.1098940, -42.9486764
.....
|                -42.9486047 |
|-21.1099515      -21.1098892 |
|                -42.9486764 |
|.....
```

O esquema conceitual gerado para a cidade de Ubá consiste em 183 classes e, por limitação de espaço, a Figura 15 apresenta apenas um pequeno trecho do esquema. A Figura 15 ilustra algumas classes pertencentes ao pacote *ROAD\_MESH* encontradas durante o processamento do arquivo OSM-XML. Todas as classes ilustradas possuem atributos próprios de acordo com o mapeamento voluntário realizado no OSM e estão sendo representadas espacialmente como Linha.

Figura 15 - Trecho da modelagem conceitual da cidade Ubá



## 4.2 Geração de Arquivos Shapefile

Por fim, a geração dos arquivos Shapefile ocorre conforme o fluxograma apresentado na Figura 11, o qual aproveita os dados que foram processados anteriormente na etapa de geração do esquema conceitual para gerar os arquivos GeoJSON e como último processo da ferramenta OSM2Diagram, são transformados em Shapefile.

Os arquivos Shapefile são criados individualmente para cada classe, então o arquivo Shapefile *residential* faz referência a classe *residential* que se encontra representada no trecho do esquema conceitual apresentado na Figura 15 e nele estarão contidas todas as ruas mapeadas como vias residenciais. No total foram gerados 183 arquivos Shapefile que compreendem todas 5404 feições encontradas.

Para manusear os arquivos Shapefile em busca de análises por meio de ferramentas de geoprocessamento, é preciso utilizar um software SIG desktop ou um WebGIS. Entretanto, para interação mais simples e visualização pode ser utilizado um WebMap. Para possibilitar a consulta e o uso dos dados espaciais por gestores municipais, que não tenham treinamento em sistemas de geoprocessamento, foi desenvolvido um WebMap denominado OSMCityView, descrito na seção 4.3. Usuários e gestores mais experientes podem fazer uso dos arquivos Shapefile por meio de um software de SIG, o que é exemplificado na seção 4.4.

## 4.3 Visualização de Arquivos Shapefile usando um WebMap

O OSMCityView<sup>13</sup> é um WebMap desenvolvido com base no *framework* Leaflet e permite a visualização, interação e análise dos dados cartográficos. Sua interface foi adaptada levando em consideração que nem todo usuário possui um conhecimento prévio sobre cartografia, sendo assim, possui uma interface intuitiva e amigável. Por fim, estão disponibilizadas as funcionalidades básicas de interação como mudança de escala, acesso à informação sobre atributos, deslocamento e o cálculo de distância entre pontos.

Por se tratar de uma aplicação cliente-servidor, o visualizador implementa as funções do *framework* Leaflet no lado cliente. Já do lado servidor, são utilizadas funções desenvolvidas na linguagem PHP para localizar os arquivos GeoJSON e enviá-los para a máquina cliente, permitindo assim a interação do usuário com os dados.

A Figura 16 ilustra uma tela do sistema OSMCityView, onde os arquivos GeoJSON são

---

<sup>13</sup> <http://www.dpi.ufv.br/projetos/OSM2diagram/>

apresentados em formato de camadas (*layers*), para uma visualização rápida e básica do arquivo exportado da plataforma OSM. Na Figura 16 pode-se observar um trecho central da cidade de Ubá, tendo em destaque ao centro um trecho do Rio Ubá, pois seu respectivo *layer* (*water*) está selecionado na janela de camadas. A janela à direita exibe os *layers* disponíveis e à esquerda estão listadas as ferramentas básicas de interação do usuário.

Figura 16 - Tela do sistema WebMap OSMCityView



#### 4.4 Análise Espacial utilizando o QGIS

A informação geográfica é constantemente utilizada para tomadas de decisões quando se trata de problemas geoespaciais. Silva et al. (2018) demonstram a importância do uso de VGI para gestores municipais por meio de análises espaciais utilizando o QGIS. Exemplos de análises espaciais citadas por estes autores incluem: o cálculo da quilometragem total de vias de um município para uma melhor distribuição dos pontos de ônibus; interseções entre buffers para escolher o local da construção de uma garagem que atenda duas regiões; e a análise da melhor rota entre uma Unidade Básica de Saúde (UBS) até o hospital.

Para realizar as análises no QGIS, os arquivos Shapefile são armazenados e acessados

em/de um SGBD com suporte a dados espaciais, por exemplo, o PostgreSQL (PostgreSQL, 2018). O SGBD PostgreSQL possui código aberto e é otimizado para aplicações que envolvem processamento de grande volume de dados, além de conter o módulo geográfico PostGIS que permite a manipulação de dados com coordenadas geográficas. Apresenta ainda uma ferramenta de importação de arquivos Shapefile chamada *PostGis Shapefile Import/Export Manager*, que está disponível na instalação do PostgreSQL.

Ao inserir os arquivos Shapefile no SBDG espacial PostGIS utilizando sua ferramenta de importação, cada arquivo é transformado em uma tabela, logo o número de tabelas criadas no banco de dados é igual ao número de arquivos Shapefile importados. Após a importação dos dados para o SGBD é preciso realizar a comunicação entre o SGBD e o SIG. O QGIS disponibiliza essa comunicação e por meio dela é possível carregar os dados armazenados no banco de dados para manipulação no QGIS.

A Figura 17 mostra o resultado de uma consulta feita no QGIS, a dados armazenados no PostGis. Tal imagem pode ser comparada com a Figura 13 (do *OpenStreetMap*), sendo possível observar que praticamente todos os elementos contidos na Figura 13 estão sendo mostrados em forma de *layers* no QGIS (Figura 17). A Figura 18 mostra o centro da cidade e permite perceber a riqueza de detalhes que pode ser adquirida a partir da plataforma OSM, tendo o rio Ubá representado pela coloração azul ao centro.

Figura 17 – Perímetro de Ubá no QGIS

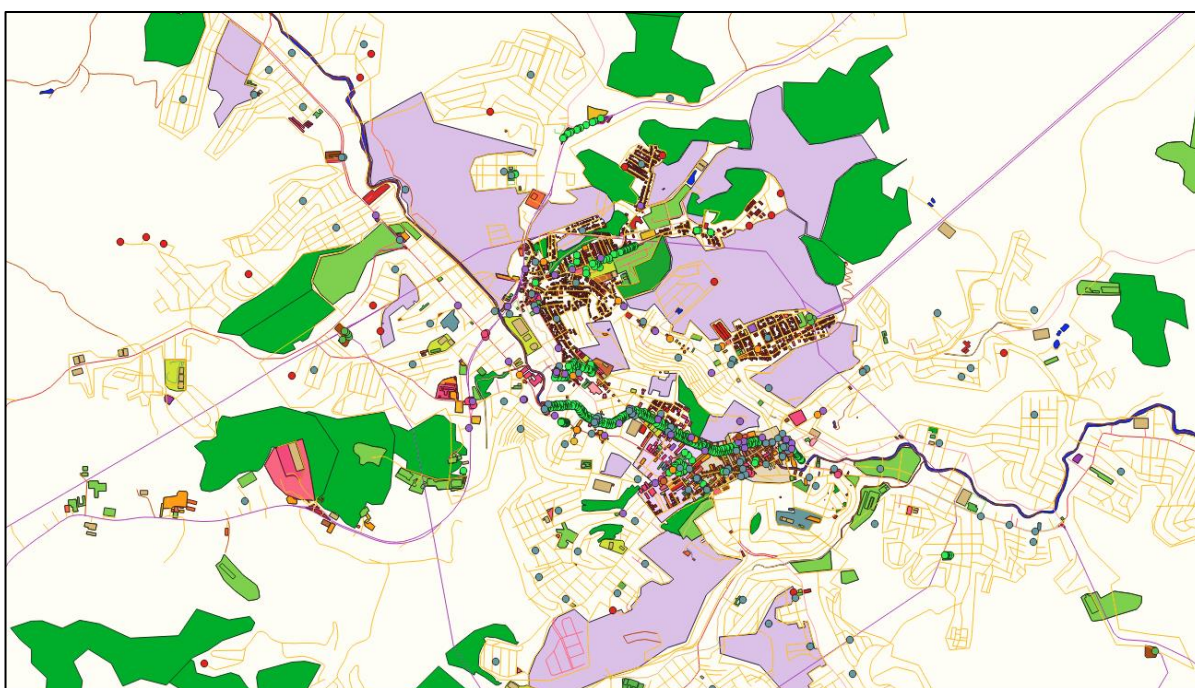
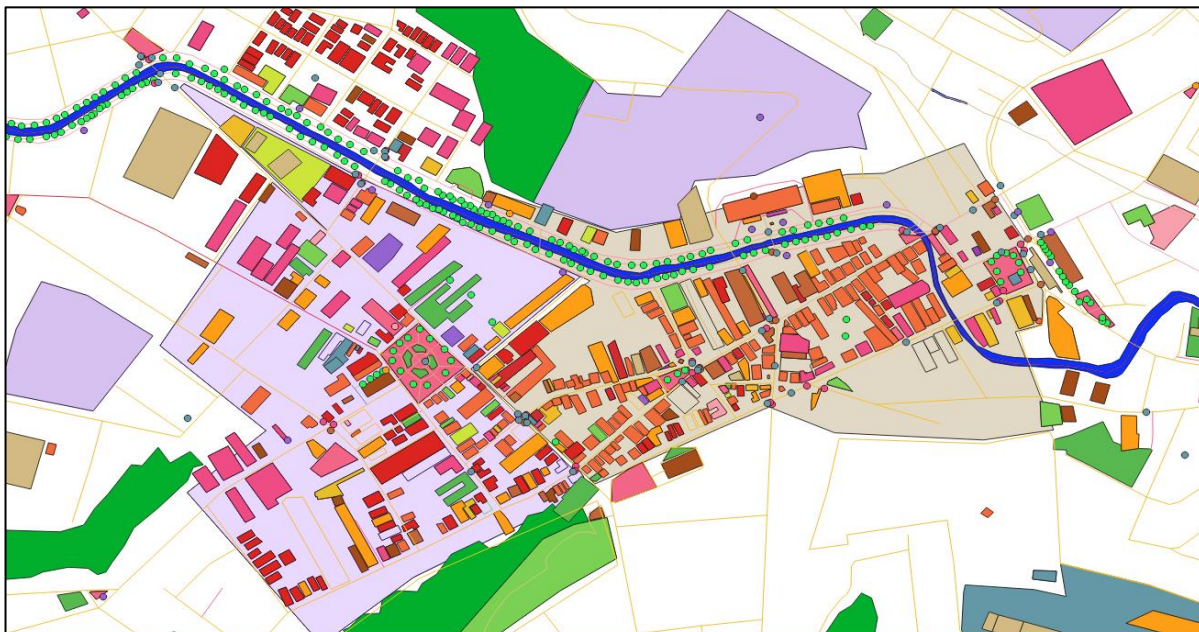


Figura 18 - Centro da cidade de Ubá no QGIS



A cidade de Ubá possui um histórico de enchentes, que normalmente causam estragos aos arredores do Rio Ubá, localizado no centro da cidade e ilustrado na Figura 18. Desse modo, tomando como base o código florestal, a Figura 19 mostra a região da Área de Preservação Permanente (APP), utilizada em rios de largura máxima de 10 metros. Essa região é representada pelo *buffer* com largura de 30m de cor rosa e deveria estar coberta por vegetação e não ter construções civis, como pode-se observar na Figura 19. Esse é um problema atual de muitas cidades, onde as construções em áreas de APP não recebem as devidas fiscalizações.

Figura 19 – Análise de APP sobre o Rio Ubá



O comando SQL correspondente à consulta anterior pode ser visto no Código 5. Este comando utiliza a função de criação de *buffer* do PostGIS (*st\_buffer*) que recebe como parâmetros a geometria do objeto base e o tamanho do *buffer* (30), gerando como resultado um novo polígono (exibido na cor rosa). Os dados utilizados nesta consulta são recuperados da tabela *Water*, que corresponde ao *layer* de mesmo nome.

É interessante destacar que o QGIS tem uma ferramenta de *buffer* que pode ser aplicada diretamente ao Shapefile, a qual possui uma interface mais amigável que a de um código SQL.

Código 5 - SQL APP Rio Ubá

```
select st_buffer(W.geom, 30)
from water W
where W.name='Rio_Ubá'
```

Portanto, é possível obter informações relevantes com consultas SQL simples e, caso necessário, o QGIS também permite realizar consultas complexas no banco de dados, com opção de que o código SQL fique encapsulado.

#### 4.5 Comparando o OSM2Diagram com a ferramenta OSMQuick

O software QGIS permite a instalação de *plugins* desenvolvidos e criados pela comunidade de usuários do software. Dentre esses *plugins*, existem alguns capazes de trabalhar com os dados provenientes da plataforma OSM.

O *plugin* OSMQuick foi desenvolvido para processar dados obtidos da plataforma OSM e se destaca pela quantidade de downloads, sendo esse o software mais instalado pela comunidade. Ele recebe como entrada um arquivo exportado pela plataforma OSM e após processar o arquivo, os *layers* são carregados no QGIS. Entretanto, apenas 4 *layers* são disponibilizados, sendo eles: points, lines, multipolygons, multilinestrings. A Figura 20 ilustra a mesma área utilizada para analisar a APP sobre o Rio Ubá (Figura 19) na seção anterior e ao comparar as duas imagens é possível perceber o grau de diferença entre elas, por conta da falta de detalhes.

Figura 20 - Centro da cidade de Ubá gerada pela ferramenta OSMQuick



A falta de detalhes se dá por conta da generalização de todas as *tags* do OSM em apenas 4 *layers*, o que dificulta o uso do mapa e afeta diretamente nas consultas SQL, deixando mais complexas ou até mesmo impossibilitando a realização de algumas análises.

## 5 CONCLUSÕES

Este trabalho mostrou que é possível a partir de dados colaborativos disponíveis de forma gratuita em uma plataforma online chamada OpenStreetMap, obter o esquema conceitual e os arquivos Shapefile que representam as áreas a serem exploradas. Tudo isso com o uso de uma ferramenta simples e intuitiva denominada OSM2Diagram. Além disso, foi desenvolvido um WebMap que permite visualizar/explorar dados geográficos das áreas de interesse utilizando um navegador web.

Este trabalho abordou a automatização de todo o processo de geração de banco de dados geográficos e a sua importância utilizando VGI, tendo como alvo principal municípios de pequeno e médio porte. Gerar, manipular e manter dados geográficos é uma das principais dificuldades enfrentadas por municípios de pequeno e médio porte. Manter em funcionamento sistemas e equipes especializadas em manipulação de dados cartográficos exige empregar uma gama de recursos que por muitas vezes o município de pequeno porte não possui ou não terá condições de perpetuar por muito tempo.

As barreiras encontradas por municípios de pequeno e médio porte para investir em projetos cartográficos, na maioria das vezes, estão relacionadas ao alto custo necessário para administrar esses projetos, pois tanto o processo de coletar dados georreferenciados como a confecção de mapas a partir desses dados devem ser realizados por profissionais com uma formação específica em conjunto de técnicas e de ferramentas de alto custo.

Uma alternativa simples para auxiliar na diminuição de custos e de tempo no processo de geração de uma base de dados capaz de gerar informações para apoio e tomada de decisões, além da geração de mapas, se dá com a combinação de plataformas VGI e ferramentas como o OSM2Diagram, capazes de gerar e manipular informações geográficas.

A plataforma OpenStreetMap obtendo dados de forma compartilhada com uma comunidade estimulada a manter essas informações e o software OSM2Diagram em conjunto com o OSMCityView para extrair essas informações, constituem um ambiente com um conjunto mínimo de recursos para análise de dados geográficos com custo relativamente baixo, já que não exige uma equipe especializada em cartografia e pouquíssimos recursos de hardware e software.

Contudo, as plataformas VGI estão surgindo para diminuir esses empecilhos relacionados aos altos valores para a coleta e a confecção de um mapa com os objetos que caracterizem um município. Mesmo que algumas cidades não possuam muitos objetos mapeados na plataforma OpenStreetMap é possível, ainda assim, gerar uma base com informações interessantes e

relevantes aos gestores municipais. Progressivamente e com pouco tempo, um município com o auxílio da população, consegue gerar uma base consistente o bastante para a tomada de decisões. Mesmo que ainda precisem de um profissional para utilizar um SIG, não haverá custo com a coleta, nem com a elaboração da base municipal geográfica pois, como demonstrado na seção anterior, uma pessoa com pouco conhecimento cartográfico e com o auxílio de software livre consegue construir uma base municipal de grande escala de qualquer lugar que esteja mapeado no OpenStreetMap.

Utilizar dados geográficos e recursos tecnológicos, tais como o software OSM2Diagram e dispositivos conectados à Internet, para tomada de decisão é algo que vai além da obtenção de informações e mapas, mas significa ter um olhar mais profundo sobre os elementos que caracterizam as cidades. Permitindo evitar o uso desnecessário de recursos para correção de problemas que podem ser previstos, além da eficácia, já que as ações podem ser tomadas de forma rápida ao observar soluções para problemas ainda não explorados. O que gera benefícios para a população que se sente mais a par das ações tomadas pelos gestores municipais.

É importante destacar que a duração do processo de geração de uma base geográfica utilizando o OSM2Diagram é relativamente baixa e praticamente automática. Portanto, o uso de um sistema pouco dependente da ação humana evita a propagação de erros no produto final e mantém a base concisa. Essas características puderam ser visualizadas na demonstração da ferramenta no Capítulo 4, que mostrou que com um método simples e poucos passos é possível obter uma base geográfica pronta para ser utilizada em um SIG. Além disso, a utilização de consultas SQL simples fornece informações importantes para o município que possibilitam a resolução de problemas que afetam diretamente a vida dos cidadãos e a economia de recursos financeiros destinados ao município.

Diante dos resultados expostos, é possível concluir que todos os objetivos enumerados na Seção 1.2 foram alcançados com êxito por este trabalho.

A proposta deste trabalho dá início à transformação de uma cidade em uma cidade digital, base para uma *smart city*, por utilizar de ferramentas tecnológicas para melhorar as decisões e antever problemas. Os trabalhos futuros devem seguir na direção de viabilizar a interoperabilidade entre modelos conceituais de banco de dados geográficos como, OMT-G, MADS ou outros. Além de buscar uma maneira de permitir que o usuário realize o mapeamento geográfico voluntário diretamente na ferramenta OSMCityView de modo que as contribuições voluntárias sejam salvas diretamente na base de dados da plataforma OpenStreetMap.

## REFERENCIAS

ALMEIDA, F.; ANDRADE, M. A integração entre BIM e GIS como ferramenta de gestão urbana. In: ENCONTRO BRASILEIRO DE TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO, 7., 2015, Recife. **Anais...** Porto Alegre: ANTAC, 2015.

BALLATORE, A.; MOONEY, P. Conceptualizing the geographic world: the dimensions of negotiation in crowdsourced cartography. **International Journal of Geographical Information Science**, v.12, n. 29, 2015.

BEAUTIFUL SOAP. **Beautiful Soap 4.4.0 Documentation**. Disponível em: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. Acesso em 10 de maio de 2019.

BORGES, K. A. V.; DAVIS, C. A.; LAENDER, A. H. F. OMT-G an object-oriented data model for geographic applications. **GeoInformatica**, v. 5, n. 3, p. 221-260, 2001.

BORGES, K. A. V., DAVIS, C. A., LAENDER, A. H. F. Modelagem Conceitual de Dados Geográficos. In: Casanova, M. A., Câmara, G., Davis Jr., C. A., Vinhas, L., Queiroz, G. R. (Eds.) **Bancos de Dados Geográficos**. Curitiba (PR): EspaçoGeo, p. 93-146, 2005.

BUDHATHOKI, N. R. Reconceptualization of user is essential to expand the voluntary creation and supply of spatial information. In: WORKSHOP ON VOLUNTEERED GEOGRAPHIC INFORMATION. Santa Barbara. **Proceedings...** California, 2007.--

BUTLER, H.; DALY, M.; DOYLE, A.; GILLIERS, S.; SCHAUB, T.; SCHMIDT, C. The GeoJSON Format Specification. **Internet Engineering Task Force (IETF)**. v. 78, 2016. Disponível em: <https://tools.ietf.org/html/rfc7946>. Acesso em: 11 de maio de 2019.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6 ed. São Paulo: Pearson, 2011.

GOODCHILD, M. F. The quality of big (geo) data. **Dialogues in Human Geography**, v.3, n.3, 2013.

GOODCHILD, M.F. Citizens as sensors: the world of volunteered geography. **GeoJournal**, v.69, n.4, 2007.

GRAPHVIZ. **Welcome to Graphviz**. Disponível em: <http://www.graphviz.org/>. Acesso em: 10 de maio de 2019.

HAKLAY, M.; WEBER, P. Openstreetmap: User-generated street maps. **IEEE Pervasive Computing**, v.7, n.4, 2008.

HEDA, M. R.; CHIKURDE, S. V. A Review: Geo-Information Technology for WebMapping Application. **International Journal of Advanced Research in Computer and Communication Engineering**, v. 5, p. 582-585, 2016.

JOHANSSON, L. P.; JUNIOR MACHADO, D. M.; SANTINI, J. C.; FILHO, O. S., Solução webmapping para elaboração de projetos de adequação ambiental em propriedades rurais na Bacia do Paraná 3. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, 14., 2009, Rio Grande do Norte. **Anais...**, Natal: INPE, 2009.

KITCHIN, R. The real-time city? Big data and smart urbanism. **GeoJournal**. v.79, n.1, 2014.

LEAFLET. **Leaflet Quick Start Guide**. Disponível em: <https://leafletjs.com/examples/quick-start/>. Acesso em: 15 de maio de 2019.

LISBOA-FILHO, J.; IOCHPE, C.; HASENACK, H.; WEBER, E. J. Modelagem conceitual de banco de dados geográficos: o estudo de caso do projeto PADCT/CIAMB. In: UFRGS/CENTRO DE ECOLOGIA. **Energia e Meio Ambiente: a questão do carvão no Rio Grande do Sul**. Porto Alegre: Editora da Universidade/UFRGS, 2000.

LISBOA-FILHO, J.; IOCHPE, C. Mecanismo de reutilização em sistemas de informação. **Acta Scientiar**, v.1, n.2, 1999.

LISBOA FILHO, J.; RODRIGUES JUNIOR, M. F.; DALTIO, J. ArgoCASEGEO - Uma ferramenta CASE de código-aberto para o modelo UML-GeoFrame. In: Workshop Iberoamericano de Ingeniería de Requisitos y Desarrollo de Ambientes de Software IDEAS, 7, 2004, Arequipa-Perú. **Proceedings...** Arequipa-Perú: Universidad Católica de Santa María, 2004.

LISBOA-FILHO, J.; IOCHPE, C. Modeling with a UML profile. In: Shashi Shekhar and Hui Xiong. **Encyclopedia of GIS**. New York, Springer, p. 691-700, 2008.

LXML. **Lxml – XML e HTML com python**. Disponível em: <https://lxml.de/>. Acesso em: 12 de maio de 2019.

MILLER, H. J. The data avalanche is here. Shouldn't we be digging? **Journal of Regional Science**, v.50, n.1, 2010.

MIRANDA, T. S. **Uma arquitetura para contribuição geográfica voluntária em infraestruturas de dados espaciais: um estudo de caso do município de Viçosa**. 2010. Dissertação (Mestrado em Ciência da Computação) – Departamento de informática, Universidade Federal de Viçosa, Viçosa, MG, 2010.

MOONEY, P.; CORCORAN, P. The annotation process in OpenStreetMap. **Transactions in GIS**, v.12, n.4, 2012.

NEIS, P.; ZIPF, A. Analyzing the contributor activity of a Volunteered Geographic Information project—The case of OpenStreetMap. **International Journal of Geo-Information**, v.1, n.2, 2012.

OPEN DATA COMMONS. **Open Data Commons Open Database License (ODbL)**. Disponível em: <https://opendatacommons.org/licenses/odbl/>. Acesso em 20 de julho de 2018.

OPENSTREETMAP. **OpenStreetMap Blog**. Disponível em: <https://blog.openstreetmap.org/about/>. Acesso em 12 de maio de 2019.

PRUVOST, H.; MOONEY, P. Exploring Data Model Relations in OpenStreetMap. **Future Internet**, v.9, n.4, 2017.

QGIS DEVELOPMENT. **QGIS geographic information system**. Open Source Geospatial Foundation Project. Disponível em: <http://www.qgis.org/>. Acesso em: 19 de maio de 2019.

GOVERNO DE MINAS GERAIS. **Regiões de planejamento**. Disponível em: <https://www.mg.gov.br/conteudo/conheca-minas/geografia/regioes-de-planejamento>. Acesso em: 10 mai. 2019.

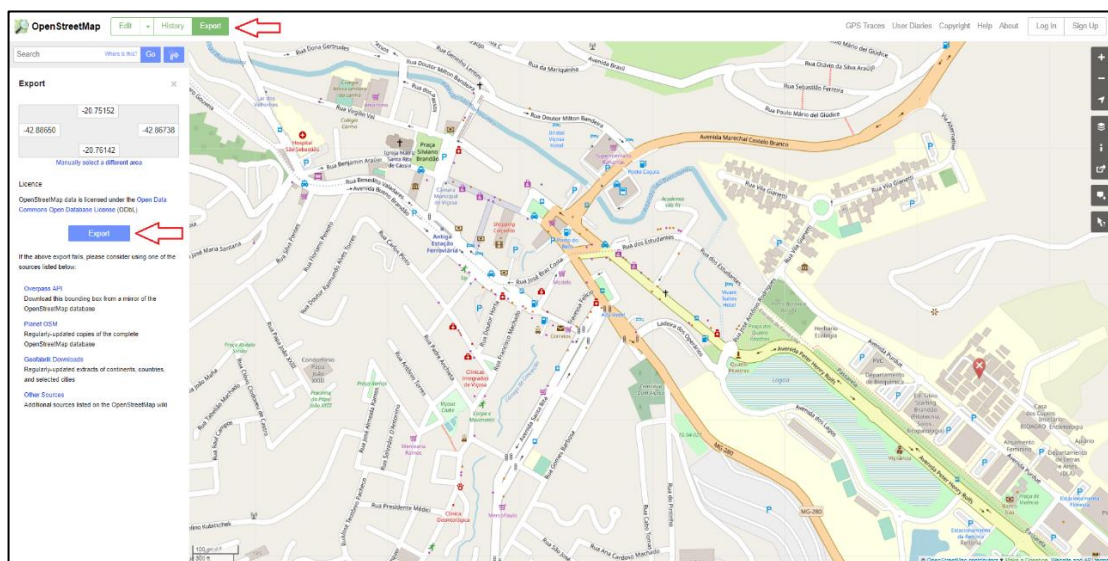
STEMPLIUC, S. M. **Modelagem de restrições de integridade espaciais em aplicações de rede através do modelo UML-GeoFrame**. 2008. Dissertação (Mestrado em Ciência da Computação) – Departamento de Informática, Universidade Federal de Viçosa, Viçosa, MG, 2008.

WARMERDAM, F. The Geospatial Data Abstraction Library. **Advances in Geographic Information Science**. Berlin, Springer, v.2, p. 87-104, 2008.

# APÊNDICE A - MANUAL DE INSTALAÇÃO E USO DA FERRAMENTA OSM2DIAGRAM

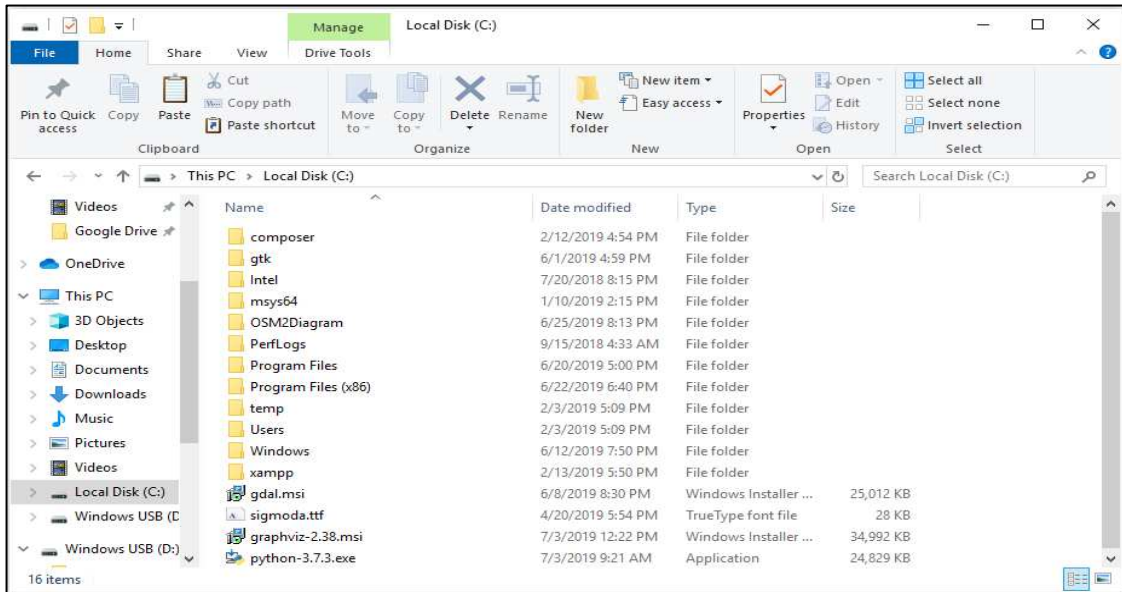
## 1. Exportando dados da plataforma OpenStreetMap

- Acesse o site da plataforma OpenStreetMap
- Escolha a opção Export localizada na barra superior do site
- Selecione a área que deseja exportar
- Clique no botão Export localizado na barra lateral esquerda
- O arquivo chamado 'map.osm' sera baixado contendo os dados da área escolhida

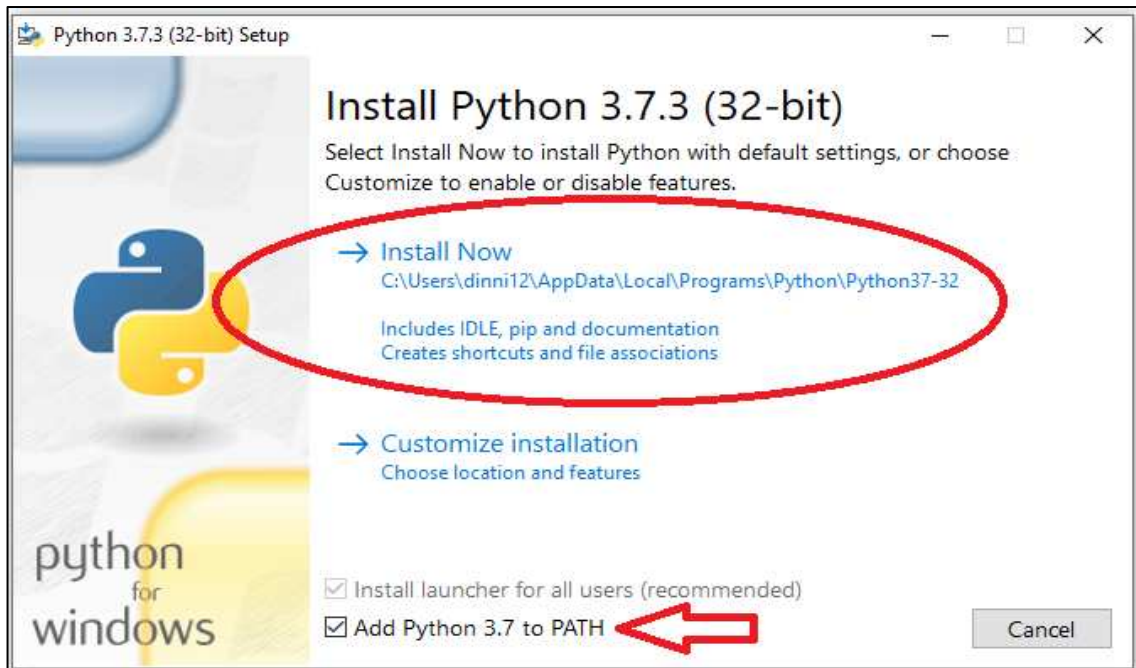


## 2. Instalando Python e plugins

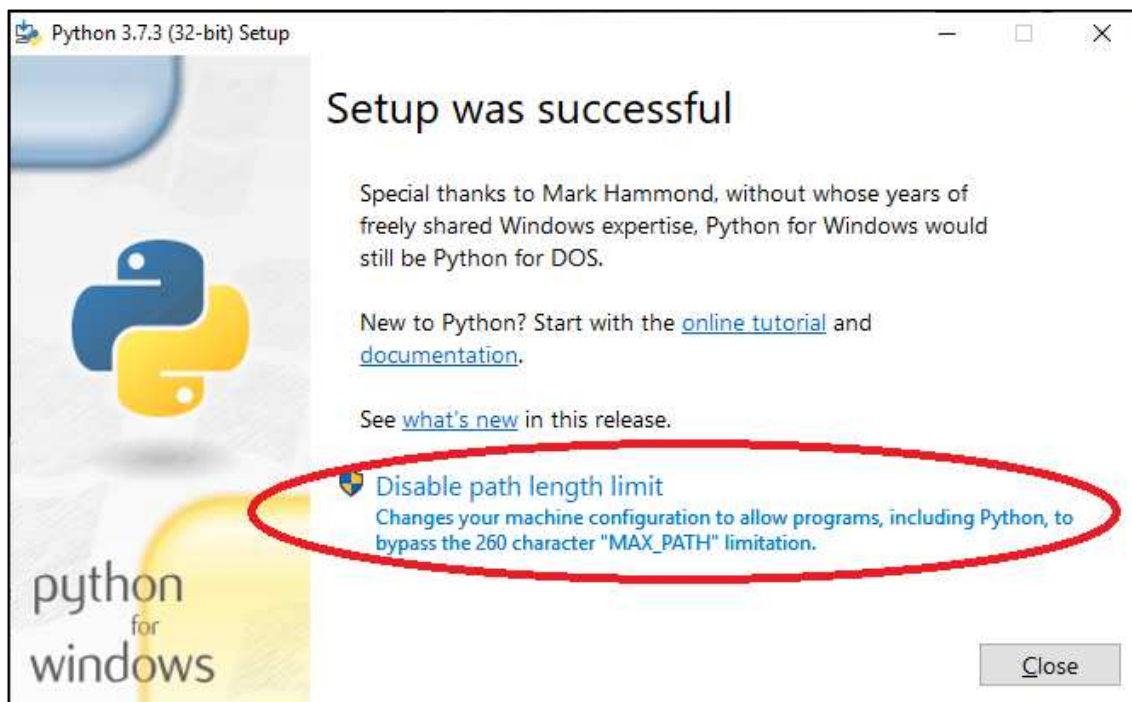
- Faça o download da Ferramenta OSM2Diagram
- Faça a extração dos arquivos para o diretório C:



- Faça a instalação do python-3.7.3.exe
- Marque a opção "Add Python 3.7 to PATH" e clique em "Install Now"



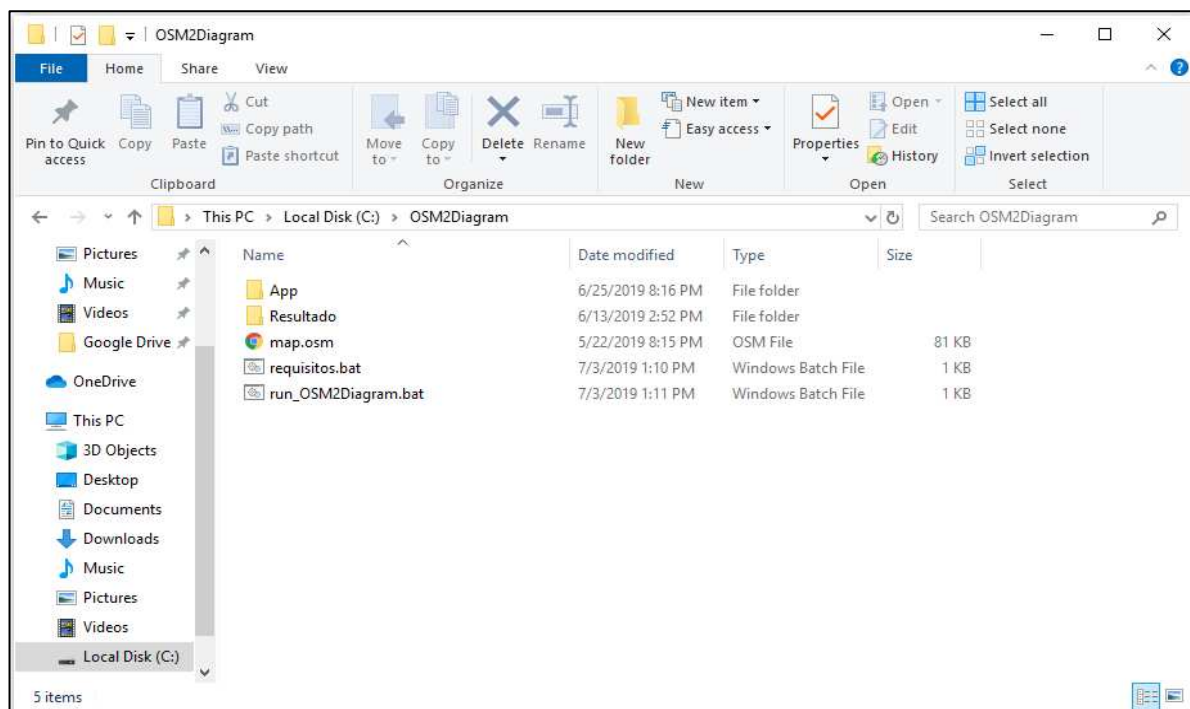
- Ao final da instalação clique na mensagem "Disable path length limit"



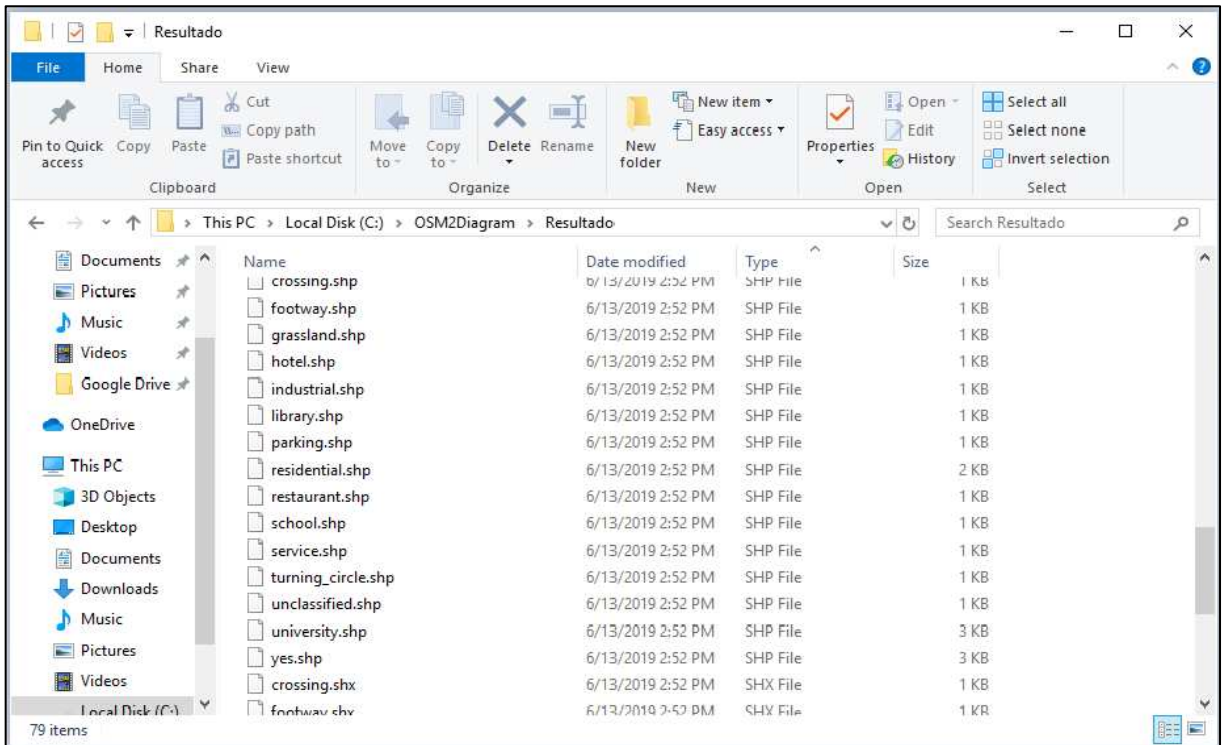
- Faça a instalação dos arquivos restantes (GDAL.msi, sigmoda.ttf, graphviz-2.38.msi)
- Entre no diretório C:\OSM2Diagram e execute o arquivo 'requisitos.bat' para instalar as dependências do Python

### 3. Gerando Shapefiles e GeoJson

- Copie o arquivo 'map.osm' para dentro da pasta 'OSM2Diagram'
- Execute o arquivo 'run\_OSM2Diagram.bat'



- Após a execução, os arquivos ShapeFile, GeoJson e o esquema conceitual ('schema.png') estarão na pasta Resultado

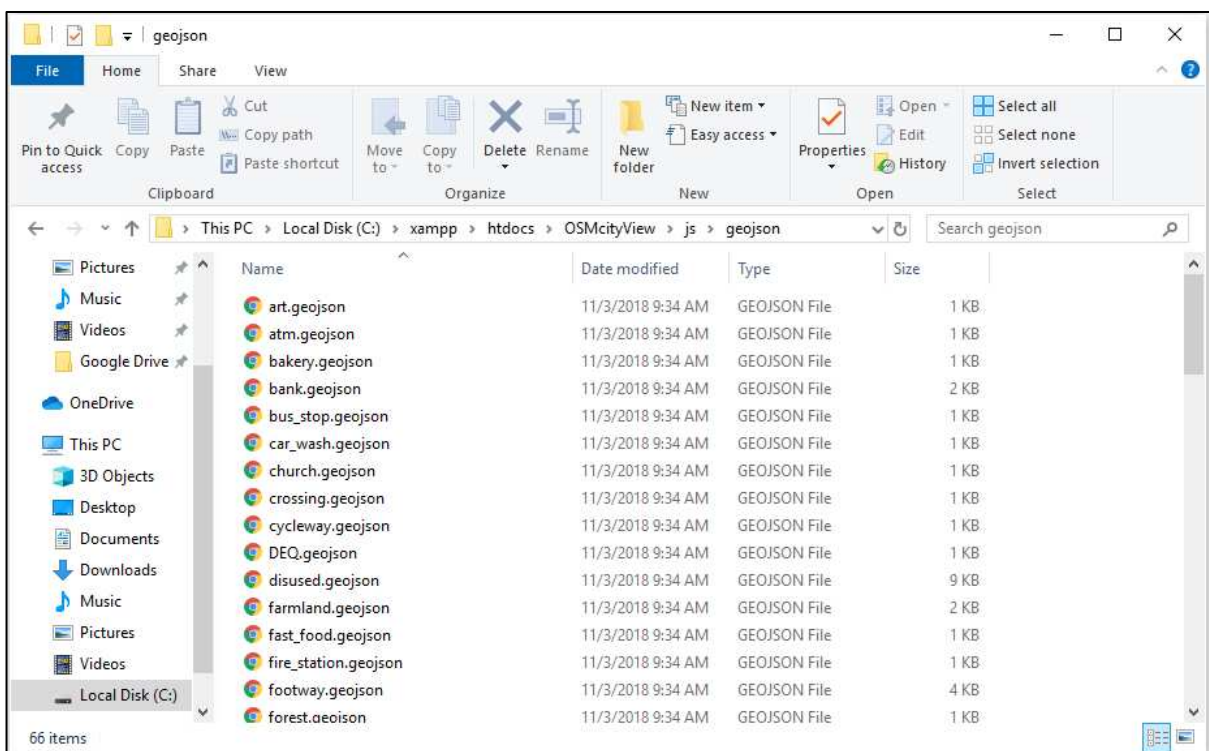


- OBS: Sempre que for executar a Etapa 3, antes excluir o conteúdo da pasta Resultado

## APÊNDICE B - MANUAL DE CONFIGURAÇÃO DO OSMCITYVIEW

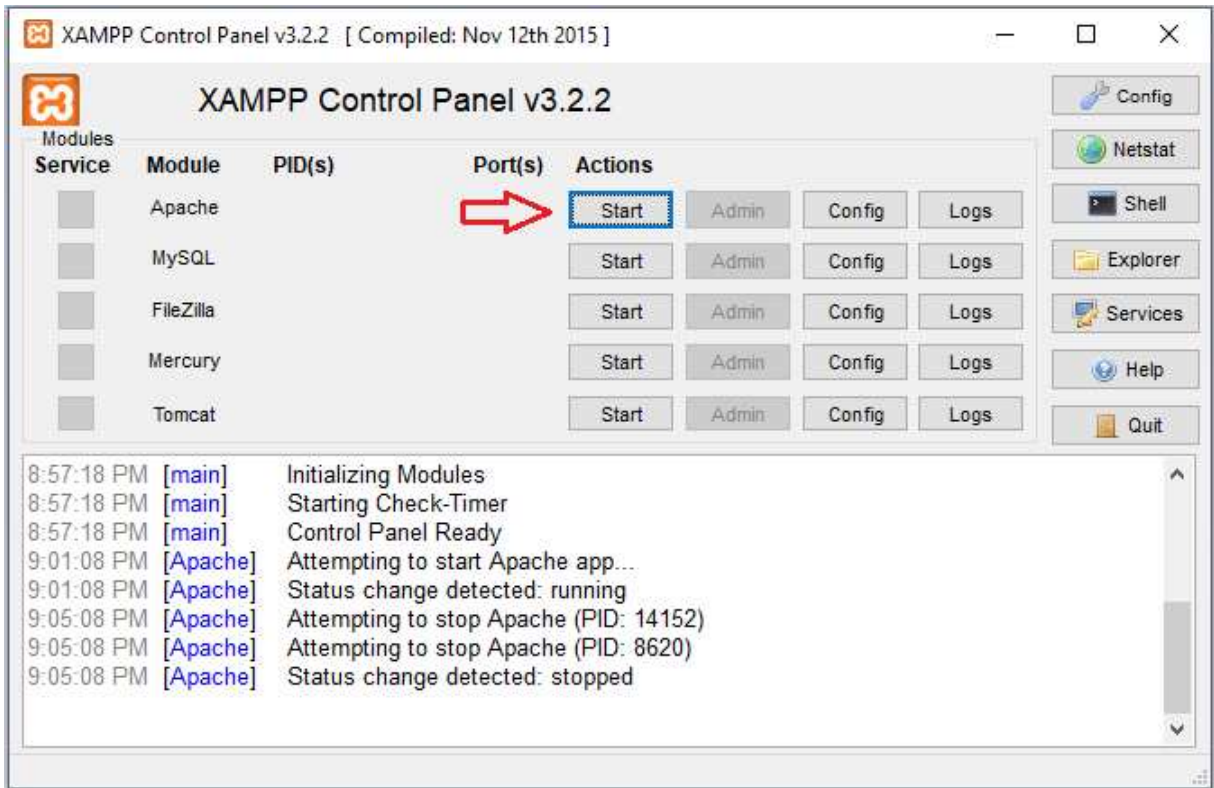
### 1. Instalando o OSMcityView

- Faça o Download do visualizador OSMcityView
- Instale o arquivo 'xampp.exe'
- Copie a pasta 'OSMcityView' para o caminho C:\xampp\htdocs
- Copie os arquivos GeoJson do diretório C:\OSM2Diagram\Resultado para o diretório C:\xampp\htdocs\OSMcityView\js\geojson



### 2. Inicializando o Visualizador

- Abra o Xampp
- Clique em Start da opção Apache



- Abra o navegador e acesse <http://localhost/OSMcityView/>
- Os arquivos GeoJson estão representados por Layers na barra lateral direita

