
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CARTOGRÁFICA
CURSO DE ENGENHARIA CARTOGRÁFICA E AGRIMENSURA

PRICILA ALVES DA SILVA

ELABORAÇÃO DE UM GEOPORTAL PARA O MUNICÍPIO DE MACAPARANA
UTILIZANDO *SOFTWARES* LIVRES



Recife

2017

PRICILA ALVES DA SILVA

**ELABORAÇÃO DE UM GEOPORTAL PARA O MUNICÍPIO DE MACAPARANA
UTILIZANDO *SOFTWARES* LIVRES**

Monografia apresentada na disciplina Trabalho de Conclusão de Curso de Graduação em Engenharia Cartográfica, do Centro de Tecnologia e Geociências, da Universidade Federal de Pernambuco, como requisito básico para obtenção do grau de Engenheira Cartógrafa.

Orientador: Prof. Dr. Cezário de Oliveira Lima Júnior.

Recife

2017

Catálogo na fonte

Bibliotecária Margareth Malta, CRB-4 / 1198

S586e Silva, Pricila Alves da.
Elaboração de um geoportal para o município de macaparana utilizando
softwares livres / Pricila Alves da Silva. – 2017.
124 folhas, il., gráfs., tabs.

Orientador: Prof. Dr. Cezário de Oliveira Lima Júnior.
TCC (Graduação) – Universidade Federal de Pernambuco. CTG.
Departamento de Engenharia Cartográfica e Agrimensura, 2017.
Inclui Referências, Apêndices e Anexo.

1. Engenharia Cartográfica. 2. Geoportal. 3. Geotecnologia.
4. Municípios de pequeno porte. I. Lima Júnior, Cezário de Oliveira.
(Orientador). II. Título.

UFPE

526.1 CDD (22. ed.) BCTG/2017-409

ELABORAÇÃO DE UM GEOPORTAL PARA O MUNICÍPIO DE MACAPARANA UTILIZANDO *SOFTWARES* LIVRES

Monografia apresentada na disciplina Trabalho de Conclusão de Curso de Graduação em Engenharia Cartográfica, do Centro de Tecnologia e Geociências, da Universidade Federal de Pernambuco, como requisito básico para obtenção do grau de Engenheira Cartógrafa, defendida no Departamento de Engenharia Cartográfica, em 11 de dezembro de 2017.

Aprovado em: 11/12/2017

BANCA EXAMINADORA

Prof. Dr. Cezário de Oliveira Lima Júnior (Orientador)
Universidade Federal de Pernambuco

Profª. Dra. Maria de Lourdes de Aquino Macedo Gonçalves
Universidade Federal de Pernambuco

Prof. MSc. Ligia Albuquerque Alcântara
Universidade Federal de Pernambuco

Aos meus pais amados.

Dedico

AGRADECIMENTOS

A Deus, autor e consumidor da minha vida. “Em tudo daí graças” (1 Ts 5:18a).

Aos meus pais amados, Áurea e Rosival. Obrigada por todos ensinamentos e por sempre me guiar pelos caminhos da verdade. “Ensina a criança no caminho em que se deve andar, e, ainda quando for velho não se desviará dele.” (Pv 22:6).

À minha irmã Débora por sempre ter acreditado que um dia poderia ser engenheira.

À minha sobrinha Hannah, sua chegada tornou as coisas mais simples e belas.

Ao meu orientador Prof. Cezário, pelo conhecimento passado, incentivo, paciência e compreensão durante o transcorrer deste trabalho. O meu muito obrigada!

Ao Departamento de Engenharia Cartográfica e de Agrimensura por terem sido a base da minha graduação.

À Prefeitura de Macaparana e a CONDEPE/FIDEM - Agência Estadual de Planejamento e Pesquisas de Pernambuco por terem disponibilizado os dados necessários para o desenvolvimento desse projeto.

A Livisgthon, pela ajuda fundamental que deu avanço a esse trabalho. Obrigada pela sua disposição em me ajudar.

As minhas amigas de graduação, Renata, Nathalia, Camila, Raquel e Luciana. Esses cinco anos e meio não seriam os mesmos sem vocês.

A Judite por sempre me acolher nas horas que sempre precisei.

A turma do LATEC, vocês são maravilhosos.

A Phillipe por sempre acreditar em mim, mesmo quando nem eu mesma acreditava. Obrigada por todos esses anos de companheirismo.

A todos os meus amigos da Comunidade da Aliança. Obrigada por me fortalecerem a cada dia.

RESUMO

O uso de geotecnologias como ferramenta de auxílio para o conhecimento territorial e tomadas de decisões administrativas vem sendo adotado por diversas prefeituras brasileiras. Atualmente os altos gastos necessários para implantação desse tipo de tecnologia não condizem com a realidade socioeconômica de muitos municípios do Brasil, principalmente os de pequeno porte que dependem do repasse de verbas para se manter, ficando as margens de programas federais para garantir sua sobrevivência. Em meio a esse contexto, o presente trabalho buscou apresentar uma forma para que o avanço tecnológico chegue a esses municípios a custos menores, propondo a criação de um geoportal utilizando ferramentas de natureza livre e não comercial. O estudo de caso foi realizado em uma amostra do município de Macaparana, localizado na Zona da Mata Norte do Estado de Pernambuco. Foram coletadas, tratadas e processadas informações relativas ao local, obtendo-se como resultado um banco de dados a partir das informações descritivas, das feições relacionadas a atual temática do município através dos dados geométricos, a integração entre as informações contidas no banco de dados com as respectivas feições e por fim a elaboração do geoportal. Os Softwares e linguagens computacionais empregados para o desenvolvimento das etapas foram: um editor de planilha e o SGBD PostgreSQL para o tratamento dos dados descritivos; o QGIS para limpeza, produção e envio dos dados geométricos à *Internet*; o editor de texto Notepad++ para edição do geoportal utilizando as linguagens PHP, HTML5, JavaScript, CSS3; e a ferramenta XAMPP que cria um ambiente Web para realização de teste.

Palavras-chave: Geoportal. Geotecnologia. Municípios de pequeno porte.

ABSTRACT

The use of geotechnologies as a tool for support the territorial knowledge and making administrative decisions has been adopted more every day by several Brazilian city halls, highlighting the technology of the Geoportais. However, the high expenses needed to implement this type of technology do not match with the socio-economic realities of many municipalities in Brazil, especially the small ones that do not possess self-sufficiency to maintain their costs, being a part of federal programs to ensure their survival. In between of this context, the present work sought to present a way for the technological advancement reach these municipalities at lower costs, proposing the creation of a Geoportal using free and non-commercial nature tools. The case study was conducted in a sample of t of Macaparana municipality, located in an area called Zona da Mata (northern forest of the state of Pernambuco), where were collected information pertaining to the local. Those information were treated and processed, obtaining a result of a database from the descriptive information, features related to the current thematic of the municipality through geometric data, the integration between the information contained in the database with the respective features and finally the elaboration of the geoportal. The software and computational languages employed to develop of the stages were: a spreadsheet editor and SGBD PostgreSQL for the processing of descriptive data; the QGIS for cleaning, producing and sending the geometric data to the Internet, which together with the SGBD provide the connection between the data; The text editor Notepad + + for editing the geoportal using the PHP, HTML5, JavaScript, CSS3 languages; And the XAMPP tool that creates a WEB environment for testing.

Keywords: Geoportal. Geotechnology. Municipalities of small porte.

SUMÁRIO

1	INTRODUÇÃO.....	10
2	OBJETIVOS.....	11
2.1	OBJETIVO GERAL.....	11
2.2	OBJETIVOS ESPECÍFICOS	11
3	EMBASAMENTO TEÓRICO	12
3.1	CADASTRO TERRITORIAL	12
3.2	BASE DE DADOS	13
3.2.1	Banco de dados.....	13
3.2.2	Banco de dados geográficos	18
3.3	SIG.....	20
4	ÁREA DE ESTUDO	22
5	MEDOTOLOGIA.....	23
5.1	COLETA DE DADOS	24
5.2	DEFINIÇÃO DOS RECURSOS TECNOLÓGICOS	25
5.2.1	Programas computacionais	25
5.2.2	Linguagens computacionais	26
5.3	DEFINIÇÃO DOS PROCEDIMENTOS METODOLÓGICOS.....	26
5.3.1	Tratamento dos dados descritivos.....	26
5.3.2	Tratamento dos dados geométricos	34
5.3.3	Conexão entre os dados descritivos e geométricos	42
5.3.4	Envio dos dados à Internet e edição do geoportal	50
6	RESULTADOS	53
6.1	DADOS DESCRITIVOS	53
6.2	DADOS GEOMÉTRICOS	59
6.3	CONEXÃO ENTRE OS DADOS DESCRITIVOS E GEOMÉTRICOS	64

6.4	GEOPORTAL	73
7	DIFICULDADES	77
8	CONCLUSÕES	78
9	RECOMENDAÇÕES.....	80
	REFERÊNCIAS	81
	APÊNDICES	84
	ANEXO	124

1 INTRODUÇÃO

Os municípios brasileiros de pequeno porte, classificados pelo IBGE – Instituto Brasileiro de Geografia e Estatística como aqueles com população de até 50.000 habitantes, de modo geral apresentam dados cartográficos e alfanuméricos desatualizados.

A visão limitada dos gestores com relação a instituição de um CTM – Cadastro Territorial Multifinalitário, carrega a ideia de que o sistema é apenas um meio seguro que garante o recolhimento do IPTU – Imposto Predial e Territorial Urbano, não enxergando sua multidisciplinaridade como forma de melhoria na gestão pública (LEMES et al., 2017).

Perante a falta de conhecimento gráfico territorial e das informações públicas administrativas para tomadas de decisões, surgem os SIG - Sistemas de Informações Geográficas, que manipulam dados referenciados geograficamente, assim como dados não espaciais, incluindo operações que suportam análises espaciais.

Atrelado ao SIG, temos os chamados Bancos de Dados Espaciais, que armazenam informações sobre o espaço geográfico, referente a um fato no mundo real e são administrados por ferramentas computacionais que visam satisfazer as necessidades de um grupo de usuários específico.

Juntamente com o progresso computacional, associado aos SIG e os bancos de dados, vem ganhando cada dia mais espaço, principalmente em diversas prefeituras, o uso de geoportais para disponibilização de dados geográficos e sua utilização como ferramenta de gestão, que segundo Tait (2005, apud PESSOA et al., 2016, p. 2), “[...] é definido como um Website que representa um ponto de entrada para a descoberta e visualização de conteúdos geográficos”.

Como os municípios de pequeno porte no Brasil são dependentes de programas federais para garantir seu sustento e desenvolvimento, a implantação de um geoportal nesses locais é um grande desafio devido aos altos investimentos aplicados para este tipo de geotecnologia. Porém, atualmente o mercado tecnológico vem desenvolvendo uma série de novos recursos baseando-se na filosofia dos *Softwares* livres, fazendo com que um novo caminho possa ser aberto e novas opções possam ser oferecidas à sociedade civil.

Buscando atender os municípios que dispõem de poucos recursos e tem a necessidade de obter o conhecimento dos seus problemas territoriais, partindo do pressuposto que a administração pública só planeja e/ou gerencia o espaço territorial que reconhece graficamente, o propósito da pesquisa é utilizar meios técnicos especializados de baixo custo para apresentar a realidade do município por meio da cartografia temática.

O projeto de pesquisa apresenta uma nova didática para auxiliar os municípios de pequeno porte sob a ótica dos avanços tecnológicos a um baixo custo e de modo especial, esperando-se que com isso surja um novo recurso no tratamento das Informações Geográficas e conseqüentemente uma nova forma de apoiar os gestores nas tomadas de decisões, sendo estas embasadas em princípios técnicos e científicos deixando de lado as inferências ou predições estabelecidas sobre a realidade.

A área utilizada para o estudo foi o município de Macaparana localizado no interior do Estado de Pernambuco a 116km de Recife. Foram realizados levantamentos sobre informações descritivas e geométricas da cidade em alguns órgãos, tais como na prefeitura do próprio município, na CONDEPE/FIDEM – Agência Estadual de Planejamento e Pesquisas de Pernambuco e na COMPESA - Companhia Pernambucana de Saneamento.

As informações coletadas foram tratadas e processadas em *Softwares* de código aberto e não comerciais, tendo como propósito desenvolver e estruturar uma base cadastral relacionada com a modelagem de um banco de dados associados a um SIG, visando a geração de um geoportal do qual represente a cartografia de uso e ocupação territorial mais adequada vinculada ao espaço urbano em estudo.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Elaborar um geoportal de baixo custo para municípios de pequeno porte.

2.2 OBJETIVOS ESPECÍFICOS

- Aprofundar o conhecimento sobre banco de dados, tanto os convencionais quanto os que possuem extensões espaciais;
- Analisar a funcionalidade e aplicabilidade dos bancos de dados, assim como sua relação com os SIG e conexão com servidores de *Internet*;
- Elaborar as etapas do processo de criação do geoportal considerando os métodos e recursos analisados.

3 EMBASAMENTO TEÓRICO

3.1 CADASTRO TERRITORIAL

O conhecimento do espaço territorial faz com que sua representação e relações de domínio ocorram de maneira justa e competente. A forma mais segura de garantir esse conhecimento é por meio da implantação e manutenção de um cadastro.

A FIG – Federação Internacional de Geômetras, afirma que cadastro é um inventário público de dados sistematicamente ordenados, apoiado no levantamento dos limites das parcelas presentes em um espaço territorial específico (CARNEIRO; ERBA; AUGUSTO, 2012).

A partir dessa definição observa-se a utilização de um termo não muito comum para sociedade civil, que determina a unidade básica do cadastro, que é a parcela, na qual deve apresentar como características os princípios de contiguidade, continuidade e identificação numérica inequívoca, estabelecendo uma relação direta entre pessoa e terra por meio de dados geométricos e descritivos. (Curso de CTM, Ministério das Cidades)

Segundo Carneiro (2003) para que o cadastro seja considerado um sistema completo, é necessário que englobe informações relacionadas a todas as parcelas existentes no ambiente, sejam elas urbanas, rurais, públicas ou privadas. O cadastro deve apresentar um caráter multifinalitário objetivando auxiliar órgãos públicos e privados, assim como atender as necessidades dos cidadãos, servindo como base para os diversos tipos de cadastros temáticos que possam existir em um determinado local. (DALE, 1990 apud CARNEIRO, 2003).

Apesar do cadastro urbano no Brasil não possuir normas federais, o Ministério das Cidades apresenta, por meio da Portaria N° 511, diretrizes para a criação, instituição e atualização do CTM nos municípios brasileiros. Porém a realidade econômica e social desses varia bastante, atingindo principalmente os que apresentam um menor porte, dificultando assim a implantação e manutenção desse tipo de sistema.

Ainda com respeito ao cadastro urbano, os elementos que compõem sua estruturação básica de natureza cartográfica são, a Planta de Quadras, a Planta de Referência Cadastral e Planta de Equipamentos Urbanos, podendo ainda ser incluído a Planta de Valor Genérico. Com respeito aos dados descritivos, as informações são recolhidas por meio de formulários tais como o BL - Boletim de Logradouros, que traz informações sobre os logradouros da cidade, e o BCI – Boletim de Cadastro Imobiliário, que apresenta dados relativos ao imóvel urbano. (CARNEIRO, 2010).

3.2 BASE DE DADOS

3.2.1 Banco de dados

“O termo *banco de dados* possui diferentes interpretações. Ele tem sido usado para indicar tudo, desde uma coleção de fichas de índices até volumes e mais volumes de dados que um governo coleta sobre seus cidadãos”. (KRORNKE, 1999, p. 12, grifo do autor).

Com o surgimento dos primeiros computadores as informações que antes eram registradas em papel por meio de fichas colocadas em pastas e guardadas em arquivos de metal sofreram mudanças e passaram a ser armazenadas em meio digital facilitando o acesso aos dados, otimizando o tempo de pesquisa e conseqüentemente reduzindo a quantidade de documentos gerados em ambiente físico. Com isso, os bancos de dados começaram a ser administrados através de *Softwares* específicos, os chamados Sistemas Gerenciadores de Banco de Dados (SGBD), apresentando assim características próprias tais como: ferramentas de busca, inserção, alteração e recuperação das informações; possível compartilhamento dos dados; controle de acesso por meio de senhas; integridade dos dados; possibilidade de operação em uma ou mais máquinas; entre outras.

Além das características citadas anteriormente, Elmasri e Navathe (2005, apud GONÇALVES, 2008) afirmam que um banco de dados deve apresentar três propriedades básicas:

1. A representação de um “minimundo”, ou seja, a transcrição de uma porção da realidade captada pelo analista;
2. Os dados devem referir a um conjunto lógico e harmonioso com significado intrínseco;
3. Deve responder a um público alvo, no qual a partir das informações relativas a esse público o banco de dados será planejado, produzido e alimentado.

Sendo assim, caso os dados nos quais se está trabalhando não atendam a esses requisitos, o agrupamento deles não pode ser caracterizado como um banco de dados. Logo, pode-se definir um banco de dados como sendo um conjunto de informações relevantes e correlacionadas, relativas a um dado fenômeno real, voltado a atender as necessidades de um grupo específico de pessoas, no qual são gerenciados por programas computacionais possibilitando uma melhor interação tanto com o cliente quanto com o responsável pela manipulação dos dados. (MEDEIROS, 2006, p. 5).

Atualmente a utilização de bancos de dados vem sendo expandida para as diversas áreas da sociedade, das quais pode-se mencionar as aplicações voltadas para informação empresarial, finanças, universidades, companhias aéreas, telecomunicações, entre outros. (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Apesar do crescente progresso que os bancos de dados sofreram melhorando de forma significativa o desenvolvimento no trabalho de diversas pessoas, Gonçalves (2008) afirma que cuidados devem ser tomados para garantir sua continuidade e seu funcionamento.

O primeiro deles é com relação ao tipo de informações que serão registradas, é de extrema importância que ocorra uma filtragem das mesmas antes de serem armazenadas, visto que a não realização dessa filtragem poderá acarretar inconsistências na modelagem do banco.

A segunda cautela que se deve tomar diz respeito as etapas necessárias para organização das informações, pois o não planejamento, ou planejamento ineficiente de como será mantido o banco de dados acarretará em atividades onerosas e dispensáveis.

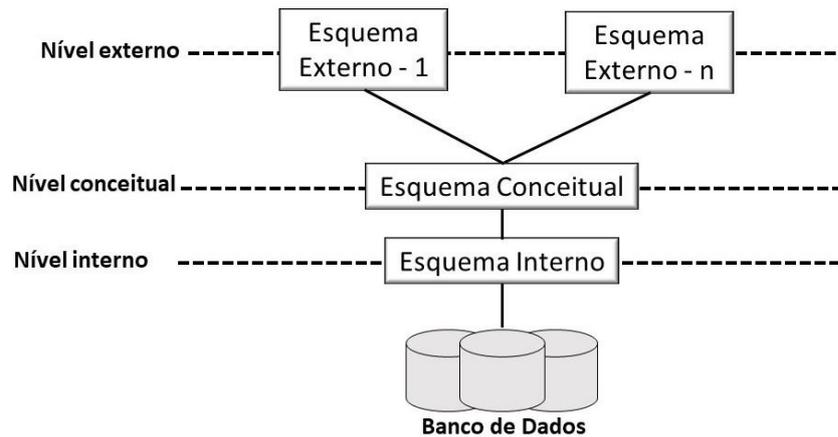
Por fim, um último cuidado que o analista deve tomar é compreender que a representação do mundo real em um meio computacional não será realizada de maneira completa e perfeita, apesar dos grandes avanços na área, logo é indispensável respeitar os limites informatizados procurando sempre moldar os dados a esses limites.

3.2.1.1 Arquitetura de esquemas para sistemas de banco de dados e modelos de dados

Com o propósito de separar as aplicabilidades dos usuários com o banco de dados e seu ambiente físico, a ANSI¹ (*American National Standards Institute*) estabeleceu uma arquitetura para os sistemas de banco de dados baseada em três níveis de esquemas (MELO; SILVA; TANAKA, 1998). Esses níveis podem representados de maneira simples na Figura 1.

¹ Organização Americana destinada à criação de normas padronizadoras.

Figura 1 - Arquitetura para os sistemas de banco de dados baseada em três níveis de esquemas.



Fonte: adaptada de MELO; SILVA; TANAKA (1998).

No nível interno, é detalhada a forma como é feito o armazenamento físico do banco de dados e seu modo de acesso. O nível conceitual prepara o banco de dados como um todo para ser desfrutado pelos clientes, ocultando os processos realizados no nível anterior. Já o nível externo, retrata a perspectiva do usuário relativa as informações contidas no banco de dados, ou seja, para cada grupo específico de pessoas será descrito uma parte dos dados armazenados, segundo os interesses desse grupo, omitindo os demais.

Para que essa arquitetura possa trabalhar de maneira apropriada, é necessário fazer uso de modelos de dados para cada nível de esquema. Um modelo de dados consiste em “uma coleção de ferramentas conceituais para descrever dados, relações de dados, semântica de dados e restrições de consistência” (SILBERSCHATZ; KORTH; SUDARSHAN, 2012, p.5). Esses modelos podem ser conceituais, lógicos e físicos

Segundo Melo, Silva e Tanaka (1998) o modelo conceitual, também chamado de modelo semântico, atua no nível externo e consiste no início da criação de um projeto de banco de dados, porém de maneira abstrata. Sua principal função é retratar os fatos existentes em uma entidade do mundo real, os quais serão armazenados em um banco de dados, estabelecendo assim um cenário geral das informações e seus relacionamentos entre si. O modelo mais comum utilizado nesse processo é o Modelo Entidade-Relacionamento (E-R), que como o próprio nome diz consiste no relacionamento entre os objetos (também chamados de entidades) existentes no mundo real.

O modelo lógico, ou modelo de implementação, atua no nível conceitual e especifica como será a organização do banco de dados, obtendo assim uma representação lógica dos fatos

coletados. Os modelos de implementação mais comuns no mercado são o modelo hierárquico, o modelo em rede e o modelo relacional. (MACHADO e ABREU, 2014).

O último modelo de dados apresentado é o modelo físico. Esse modelo age no nível interno e sua função é estruturar as informações dentro do banco de dados. Para realizar esse processo, utiliza-se uma linguagem específica (DDL - Linguagem de Definição de Dados) que servirá para preparação física das informações em uma base de dados do SGBD. (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

3.2.1.2 Sistema gerenciador de bancos de dados (SGBD)

O ponto inicial para elaboração de um banco de dados corresponde à etapa de introdução das informações no sistema, na qual as mesmas serão inseridas e passarão a ser supervisionadas pelos Sistemas Gerenciadores de Banco de Dados (SGBD). (GONÇALVES, 2008).

Um SGBD é que um conjunto de programas computacionais utilizados para gerir uma base de dados, possibilitando a criação de novas bases, assim como sua modificação, inserindo ou eliminando informações e até mesmo a recuperação das mesmas. Segundo Melo, Silva e Tanaka:

[...] os SGBD surgiram como uma camada lógica entre a aplicação e os dados, fornecendo um mecanismo de abstração de detalhes, como os relacionamentos com os caminhos de acesso aos dados e as estruturas de armazenamento utilizadas, por exemplo. (1998, p. 3).

Esses mecanismos de abstração de detalhes citado por Melo, Silva e Tanaka (1998) no trecho anterior, só é possível graças a uma linguagem própria utilizada nos SGBD. A linguagem comumente empregada, principalmente em ambiente relacional, é a SQL (*Structured Query Language*). Essa linguagem é composta por subconjuntos que darão origem aos programas de aplicação, onde servirão de apoio para as operações das quais queiram se executar sobre o banco de dados. Alguns desses subconjuntos são apresentados por Silberschatz, Korth e Sudarshan (2012), tais como: linguagem de definição de dados (DDL - *Data Definition Language*), linguagem de manipulação dos dados (DML - *Data Manipulation Language*), integridade, definição de visão, controle de transição, SQL embutida e dinâmica (podem ser utilizada junto das linguagens de programação) e autorização.

Um fato importante que deve ser analisado nos SGBD são as atividades relativas a leitura e escrita em um banco de dados que para o cliente essas atividades são realizadas em

uma única etapa denominadas de transições. “Uma transição é a parte de um programa, delimitada por declarações de início e de fim, que acessa e pode atualizar vários registros de várias tabelas, sendo que essa atualização só tem sentido se for feita por inteiro”. (CHALEGRE, 2011, p.20). Portanto, ao realizar qualquer tipo de transição, as informações armazenadas no banco de dados devem permanecer íntegras. Para garantir essa integridade o SGBD deve conservar o modelo denominado ACID, cuja sigla significa as características de atomicidade, consistência, isolamento e durabilidade (SANTOS e MENDES, 2016).

A atomicidade garante que todas as transições executadas sobre o banco de dados sejam realizadas, não havendo transcendência entre as etapas. A consistência, assegura a coerência dos dados durante o processo de transição por meio de recursos que garante a integridade das informações. O isolamento, certifica que mesmo em casos de atividades simultâneas, um processo só será realizado após a conclusão de outro e nunca ao mesmo tempo. E por fim, a durabilidade legitima que os dados permanecerão inalterados ao menos que seja realizada algum tipo de transação que modifique ou exclua o mesmo. (CHALEGRE, 2011).

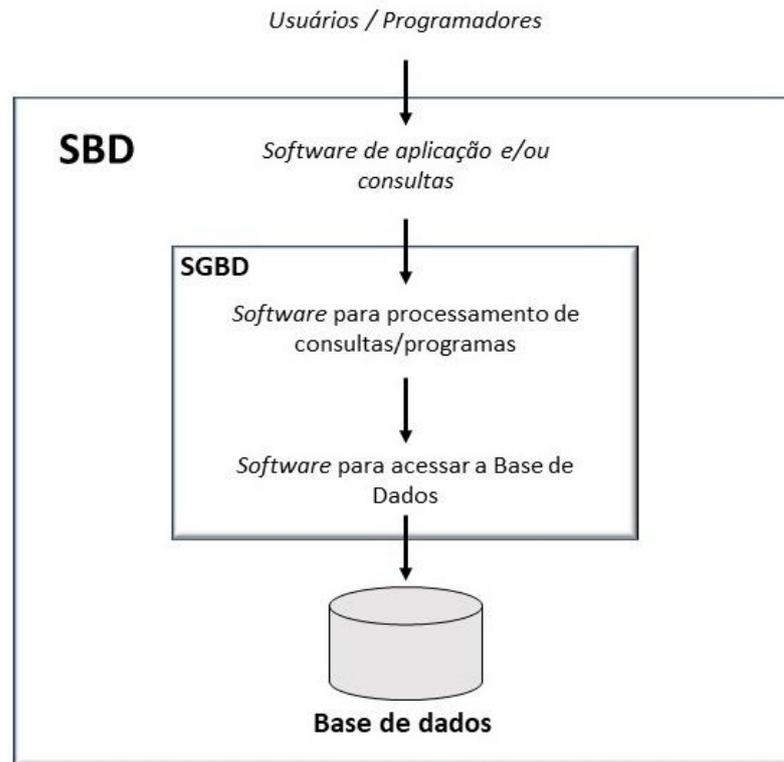
Apesar do modelo ACID ser essencial para conservar a totalidade do SGBD, outros fatores devem ser levados em considerações para garantir o bom funcionamento do mesmo, como por exemplo, a segurança dos dados; simplicidade no processo de execução das tarefas; verificação de repetição dos dados; confiabilidade; eficiência no desempenho; dados corretos dentre outros. (SANTOS e MATTOS, 2017).

3.2.1.3 Sistema de bancos de dados (SBD)

Um Sistema de Banco de Dados (SBD) é uma estrutura composta por uma base de dados, pelos *Softwares* de gerenciamento dos dados (SGBD) e pelos indivíduos na esfera envolvente (usuários e programadores). A principal função desse sistema é conservar o armazenamento das informações de maneira ordenada, oferecendo-as, quando solicitadas, aos usuários (DATE, 2000 apud GONÇALVES, 2008, p.42).

A arquitetura geral de um SGD pode ser analisada na Figura 2. Um serviço é solicitado externamente através de usuários ou programadores. A partir dessa solicitação, o SBD inicia seu processo de consulta ou aplicações de manipulação dos dados por meio dos programas que constituem o SGBD, concedendo assim as respostas solicitadas.

Figura 2 - Arquitetura de um sistema de banco de dados.



Fonte: adaptada de GONÇALVES (2008).

3.2.2 Banco de dados geográficos

Todo fenômeno no qual está relacionado algum tipo de dimensão espacial é dito como um dado espacial. Quando esses dados apresentam uma componente espacial referenciada à superfície da Terra, são chamados de dados geográficos. (CÂMARA et al., 1996, apud BORGES, 2002, p. 5).

Por estarem diretamente ligados as representações terrestres, os dados geográficos, ou geoespaciais, têm como base suas coordenadas, das quais georreferenciam a uma determinada região terrestre, tornando possível realizar atividades de análise do espaço que as contém.

Os dados geoespaciais podem ser representados por meio dos modelos vetoriais e rasters. A estrutura vetorial possibilita a interpretação do mundo real por meio de figuras geométricas simples, tais como pontos, linhas e polígonos das quais fazem uso de coordenadas para representá-las. O modelo raster é composto por uma estrutura matricial organizada em linhas e colunas onde seu cruzamento resultam em células, chamadas de pixels, as quais armazenam as feições observadas. (BOSSLE, 2016).

Logo, em meio a esse cenário o desafio era realizar o armazenamento desses dados geográficos em um ambiente computacional. Como resposta surgiram os chamados banco de dados geográficos tendo um funcionamento semelhante aos bancos de dados relacionais convencionais, assim como suas características e limitações, diferenciando por apresentarem uma extensão espacial em seu sistema na qual armazena as feições geométricas em suas tabelas.

Essa nova tecnologia proporcionou a realização de análises e consultas espaciais, sendo possível realizar tarefas como calcular distancias entre cidades, determinar a área de uma região entre outras operações.

3.2.2.1 Modelagem conceitual para bancos de dados geográficos

O desenvolvimento tecnológico ao longo dos anos, fez com que suas aplicações evoluíssem, surgindo a necessidade de tratar dados mais complexos, como por exemplo as informações geográficas.

Os modelos de banco de dados existentes eram limitados mediante o grau de complexidade apresentado pelos dados geográficos, com isso foi introduzido um novo conceito no mundo da modelagem de dados, o modelo objeto relacional, no qual é utilizado para implementação dos modelos conceituais dos bancos de dados geográficos.

De acordo com Silberschatz, Korth e Sudarshan (2012) o modelo objeto-relacional é uma extensão do modelo de dados relacionais, no qual integra tipos de dados com grau de complexidades maiores e uma orientação voltada ao objeto.

O conceito de orientação voltada a objetos (O-O) consiste em retratar em meio computacional os fenômenos ocorrentes no mundo real compreendendo que o mundo é composto por diversos objetos que interagem uns com os outros, da mesma maneira a orientação a objetos consiste em considerar os sistemas computacionais como uma coleção de objetos que se relacionam entre si. (FARINELLI, 2007).

As vantagens apresentadas por esse modelo são inúmeras, entre elas pode-se citar a possibilidade de se trabalhar com os fenômenos de maneira individual, causando um menor impacto ao modelo como um todo (GONÇALVEZ, 2008).

3.3 SIG

Para Ferreira (2006), todos os serviços realizados pelo homem são executados em alguma parte do território geográfico, logo podem ser geograficamente referenciados, possibilitando as aplicações de SIG em suas atividades.

Um SIG é uma geotecnologia composta por recursos humanos, métodos, dados, *Hardwares* e *Softwares*, onde é possível associar informações descritivas a uma posição geográfica, possibilitando a manipulação dos dados por meio de análises e consultas. (MEDEIROS, 2012).

A Figura 3 mostra um exemplo clássico de como é estruturado um SIG.



Fonte: adaptado de FERREIRA (2006).

Como os SIG estão inseridos em diversas camadas das atividades humanas, é necessário realizar a integração de profissionais habilitados de várias áreas para o desenvolvimento metodológico eficiente do sistema, constituindo uma interdisciplinaridade nos recursos humanos necessários, expandindo também aos usuários finais do produto.

Com respeito aos dados que compõem a estrutura, podem ser de natureza descritiva e geográfica, sendo esse último responsável pela criação de uma base cartográfica.

Uma base cartográfica é um conjunto de dados georreferenciados a um SGR - Sistema Geodésico de Referência, dos quais são empregados para geração de produtos cartográficos de um determinado local.

O SGR tem por objetivo ajustar de maneira local ou global as representações espaciais sobre a superfície terrestre. Através do SGR temos os sistemas de coordenadas que são modelos matemáticos empregados para fornecer a localização espacial de um determinado objeto na superfície da Terra (BOSSLE, 2016).

No Brasil, atualmente o SGR utilizado é o sistema geocêntrico SIRGAS 2000, juntamente com o sistema de coordenadas UTM – Universal Transversa de Mercator.

Com relação aos *Hardwares*, nada mais são do que os equipamentos utilizados tanto para coleta dos dados quanto para o processo dos mesmos.

Uma atenção maior deve ser voltada aos programas computacionais de SIG, que nos últimos anos vem surgindo um grande número de tecnologias apoiadas na filosofia do *Softwares* livres.

Baseada nessa filosofia, o presente trabalho expõe algumas ferramentas e linguagens computacionais existentes no mercado, de natureza livre e não comerciais, que serão abordadas ao longo do processo, das quais pode-se citar:

- PostgreSQL - SGBD com orientação voltada ao objeto que possui uma extensão espacial, o PostGIS, que permite armazenar dados descritivos e geométricos em um único ambiente;
- QGIS – programa que manipula dados geoespaciais nos formatos vetoriais ou rasters;
- XAMPP – é uma ferramenta computacional prática e simples, utilizada por desenvolvedores para criar um servidor *Web* local e um ambiente de desenvolvimento em PHP para fins de testes.
- Notepad++ – editor de texto para ambiente *Windows*, com interface de fácil manipulação onde possibilita o acesso a mais de 40 sintaxes de programação.
- Biblioteca do JavaScript Open Layers – API que fornece funcionalidades para criação de mapas *Web*;
- Linguagem de programação PHP (PHP: Hypertext Preprocessor) – linguagem script² utilizada para desenvolvimento *Web* interpretada pelo servidor;
- Linguagem de marcação – são linguagens que necessitam de interpretação para apresentar os dados, porém não geram dados a partir de sua construção, mas apresenta as informações previamente escritas. Dentre as existentes no mercado as utilizadas serão

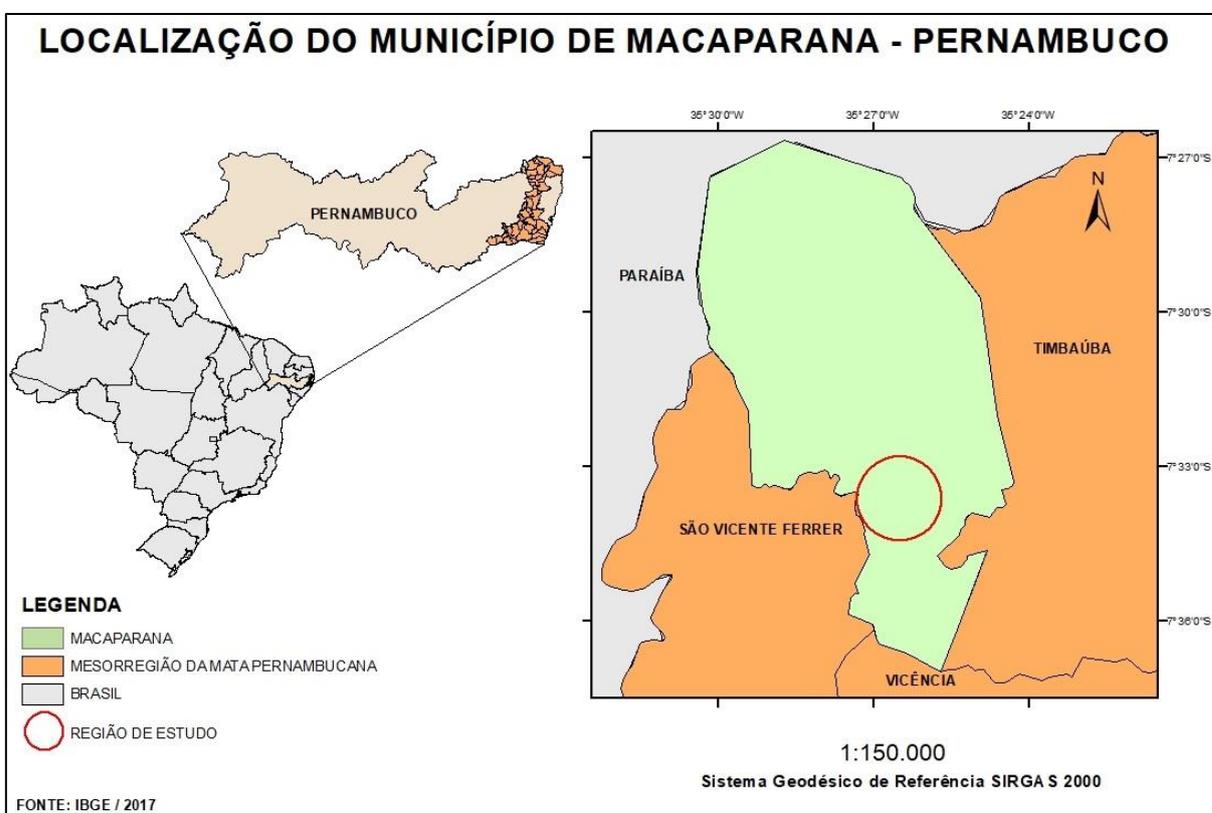
² São roteiros seguidos por sistemas computacionais.

a HTML5, JavaScript e CSS3, para desenvolvimento, interatividade e modificação do estilo do geoportal.

4 ÁREA DE ESTUDO

Das cinco mesorregiões do estado brasileiro de Pernambuco, o município de Macaparana localiza-se na Mesorregião da Mata Pernambucana e Microrregião da Mata Setentrional, também conhecida como Zona da Mata Norte. Limita-se ao norte com o Estado da Paraíba, ao sul com os municípios de São Vicente Ferrer e Vicência, ao leste com Timbaúba e ao oeste com a Paraíba e São Vicente Ferrer. A localização da cidade encontra-se representada na Figura 4.

Figura 4 – Mapa de localização da área de estudo.



Fonte: Autoria própria.

De acordo com o plano diretor do município, o mesmo está introduzido no Planalto da Borborema com variações de altitudes entre 650 a 1000 metros, apresentando um relevo movimentado, solos predominantemente férteis e clima do tipo tropical chuvoso com verão seco.

Sua rede hidrográfica está adentra na Bacia do Rio Goiana com uma malha hídrica que constitui a sub-bacia do Rio Capibaribe Mirim.

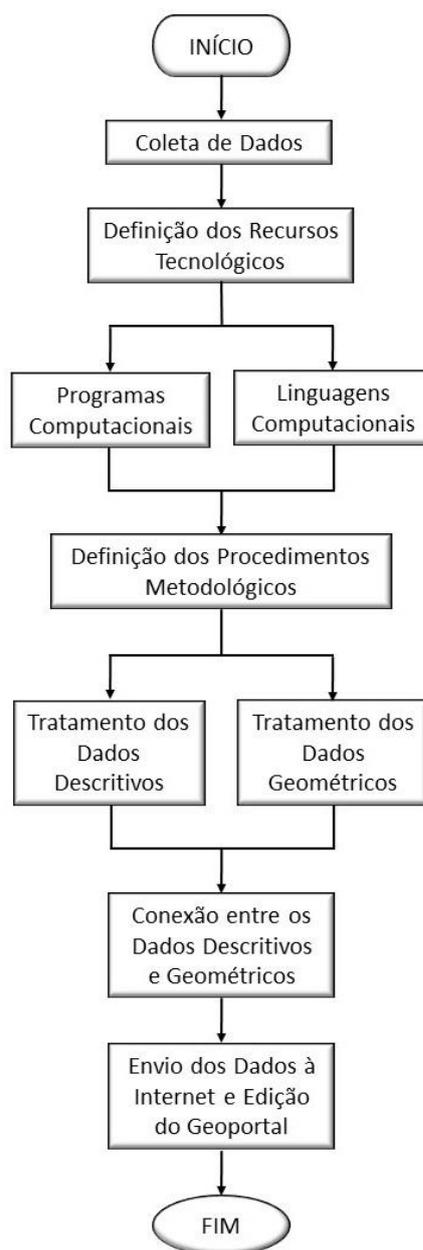
A vegetação original era composta por floresta do tipo Subcaducifólia, porém com o processo de urbanização fez com a vegetação local fosse substituída pelo plantio de cana-de-açúcar, mandioca, batata doce, uva, milho, feijão e outras monoculturas de menor expressão.

Segundo o senso do IBGE em 2010, o município apresenta uma área de 108,049km² com uma população estimada em 23.925 habitantes, sendo boa parte dela residente na zona urbana. Logo de acordo com o mesmo órgão, Macaparana se classifica como um município de pequeno porte por apresentar uma população inferior a 50.000 habitantes, enquadrando-se no requisito proposto para o desenvolvimento do estudo.

5 MEDOTOLOGIA

Os procedimentos metodológicos desenvolvidos durante a realização deste trabalho são apresentados resumidamente na Figura 5.

Figura 5 – Fluxograma representando os processos metodológicos adotados na pesquisa.



Fonte: Autoria própria (2017).

5.1 COLETA DE DADOS

Para composição de uma base de dados que subsidiasse o desenvolvimento do trabalho, buscou-se informações na CONDEPE/FIDEM - Agência Estadual de Planejamento e Pesquisas de Pernambuco, na Prefeitura de Macaparana e na COMPESA - Companhia Pernambucana de Saneamento.

Na CONDEPE/FIDEM obteve-se 25 (vinte e cinco) ortofotos, que recobria toda a cidade, numa escala de 1:5000. Foram utilizadas apenas aquelas que abrangiam o perímetro

urbano, totalizando em 2 (duas) ortofotos, cuja nomenclatura são SB-25-Y-C-V-1-NO-C-IV e SB-25-Y-C-V-1-NO-II.

Essas ortofotos foram resultados do projeto desenvolvido pelo governo de Pernambuco, o PE 3D, que realizou um mapeamento aerofotogramétrico e perfilamento a laser de todo o território pernambucano, executado pela ENGEFOTO e ESTEIO. O conjunto de dados está sendo apresentado exclusivamente em meio digital e todo o conteúdo será disponibilizado integralmente para download. (PE 3D, 2016).

Na prefeitura do município foram obtidos os dados geométricos no formato .dwg, relativos a sede do município, as áreas críticas, o sistema viário, o perímetro urbano e o zoneamento. Apesar da existência de pontos de controles nesses arquivos, não foi possível a realização do georreferenciamento dos mesmos, pois o desconhecimento sobre o sistema de referência era evidente e não se sabia ao certo a procedência dessa documentação. A ausência de escala nos desenhos também era um problema presente, diante disso, optou-se em não manipular essas informações durante a fase de processamento dos dados, visto que sua utilização acarretaria em inconsistências nos resultados. Informações descritivas relacionadas ao Extrato Cadastro Imobiliário, também foram adquiridas, porém, apenas da quadra 001 da cidade foram coletadas, já que o acesso ao banco de dados geral era inviável, já que os responsáveis pelo setor que dispuseram das informações possuíam restrições para aquisição dos dados.

No último órgão, a COMPESA, não se alcançou sucesso em obtenção de dados, pois a empresa não possuía informações referente ao local de estudo.

5.2 DEFINIÇÃO DOS RECURSOS TECNOLÓGICOS

5.2.1 Programas computacionais

Os programas computacionais utilizados na pesquisa foram:

- QGIS 2.18.13 Las Palmas;
- PostgreSQL 9.5.9;
- PostGIS 2.0;
- Xampp;
- Editor de texto Notepad++;
- Editor de planilha.

5.2.2 Linguagens computacionais

As linguagens computacionais utilizadas na pesquisa foram:

- Biblioteca do JavaScript, OpenLayers;
- Linguagem de Programação PHP (PHP: Hypertext Preprocessor);
- Linguagens de Marcação (HTML5, JavaScript e CSS).

5.3 DEFINIÇÃO DOS PROCEDIMENTOS METODOLÓGICOS

5.3.1 Tratamento dos dados descritivos

Inicialmente realizou uma análise do material descritivo obtido na prefeitura referente a quadra 001, onde a avaliação das informações contidas nesse material deu origem a diversos tipos de dados, todavia registrou-se apenas os dados significativos referentes a pesquisa, considerando as relações existentes entre eles, constituindo assim a fase inicial para criação do banco de dados, ou seja, a implementação do modelo conceitual.

Apesar de existirem *Softwares* especializados para realização da etapa lógica na criação de um banco de dados, optou-se em utilizar um editor de planilha simples para execução dessa fase, pois a amostra trabalhada da área de estudo é relativamente pequena possuindo poucos dados, podendo ser facilmente manuseados em programas dessa natureza.

Sendo assim, foram criadas três tabelas relativas as feições das quais possuíam-se os dados: uma tabela para quadras, uma para lotes e uma para edificações. Na tabela de quadras, os registros (colunas) selecionados foram o código de inscrição das quadras, a seção, o distrito, o setor, a quadra, o bairro, a quantidades de lotes, a pedologia e a topografia. Já na tabela de lotes, os registros considerados foram o código de inscrição dos lotes, o distrito, o setor, a quadra, o lote, o endereço, o bairro, o CEP e a área do terreno. Enquanto que na tabela de edificações os registros utilizados foram o código de inscrição das edificações, o distrito, o setor, a quadra, o lote, o número da edificação, o endereço, o bairro, o CEP, a área do terreno, área da unidade a natureza predial.

Além dessas tabelas, foram elaboradas mais duas, uma contendo dados referentes aos funcionários e aos contribuintes do município. Nessas tabelas foram utilizados dados fictícios para seu preenchimento e os registros considerados foram o nome, usuário e senha de acesso, para ambas. Essas tabelas estarão conectadas diretamente com o geoportal e servirá para

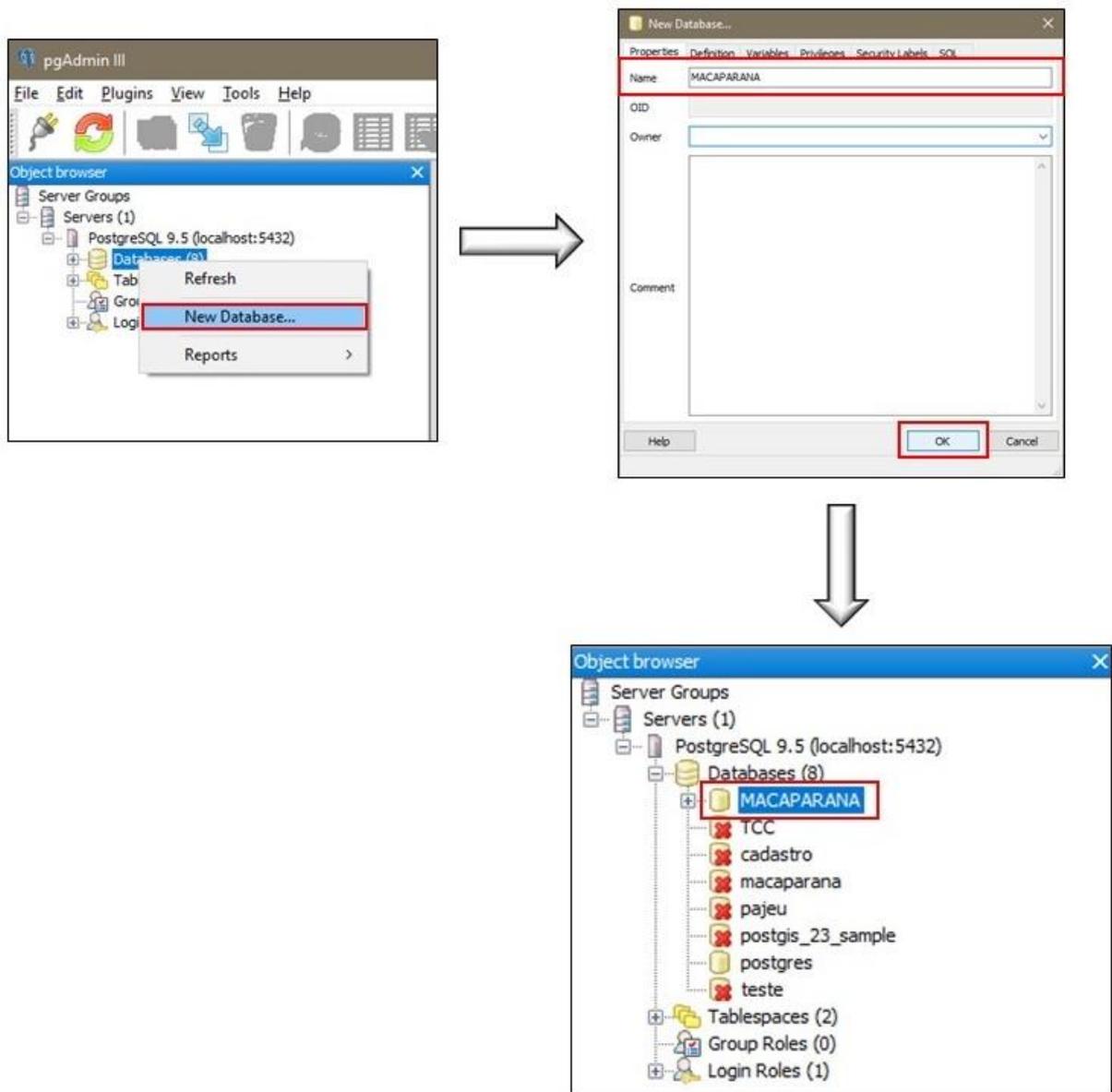
validação das informações inseridas pelo usuário, no acesso restrito que conterà na página. É importante frisar que todos os dados organizados no editor de planilha foram convertidos, no próprio programa, para o formato .csv para que pudessem ser enviados ao SGBD posteriormente.

O próximo passo consistiu na criação do banco de dados por meio do SGBD PostgreSQL. Primeiramente, na janela de navegador de objetos (object browser) criou-se uma nova base de dados, clicando-se com o botão direito do mouse na opção Databases → New Databases. Na janela seguinte criou-se efetivamente o banco de dados, indicando o nome da base na opção “Name” e confirmando o processo em “Ok”. Feito isso, atualiza-se o sistema clicando mais uma vez com o botão direito do mouse em “Databases” selecionando-se a opção



“Refresh” ou através do ícone “Refresh the selected object” localizado na barra de menu, com isso aparecerá no navegador de objetos o novo banco de dados criado. As etapas seguidas anteriormente podem ser visualizadas na figura a seguir.

Figura 6 – Passos para criação de uma base de dados no *Software* PostgreSQL.



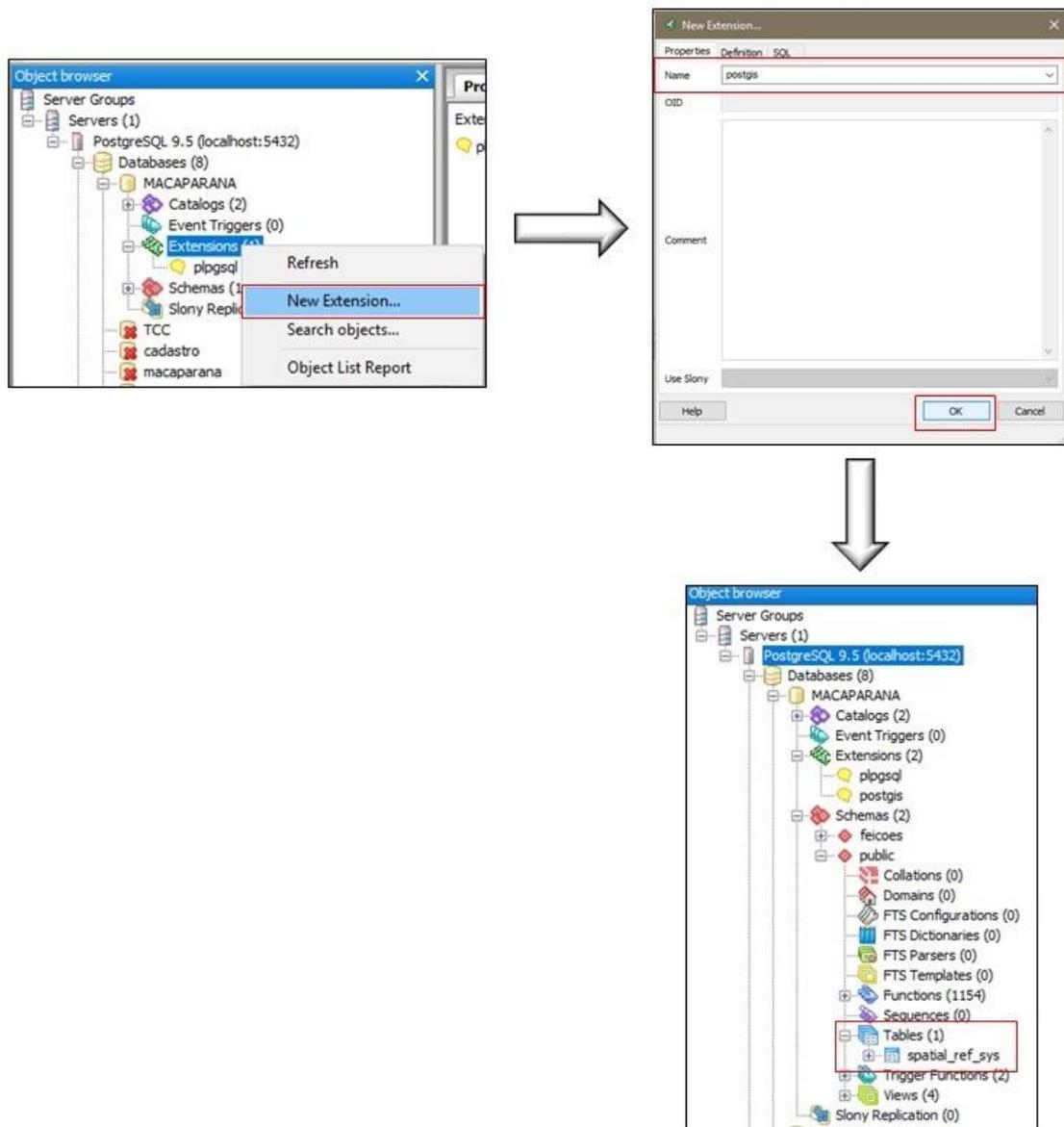
Fonte: Autoria própria (2017).

Como uma das propostas do trabalho é estabelecer uma base de dados espacial para o município, é necessário preparar o ambiente para que ele possa receber, além dos dados descritivos, as informações geométricas. Logo, é fundamental a existência da tabela “spatial_ref_sys” no banco de dados para que as feições possam ser armazenadas no mesmo, pois essa tabela é responsável por guardar os sistemas de referências espaciais e os detalhes necessários para transformá-los e/ou reprojotá-los.

Para criação da tabela “spatial_ref_sys”, basta abrir o banco de dados gerado no próprio painel de navegador de objetos em “Extensions”, com o botão direito do mouse, escolhe-se a opção “New Extension”. Na janela seguinte no campo “Name” procura-se pela extensão

“postgis” e confirma-se o processo em “Ok”. Em seguida atualiza-se a extensão seguindo o mesmo processo realizado para atualização após a criação do banco de dados, porém atualizando-se a opção “Extensions”. Após essa sequência de passos, o resultado da criação da tabela se encontrar na seção Schema → public → Table no banco de dados no qual se está trabalhando. Esse processo pode ser verificado na Figura 7.

Figura 7 – Processo de criação da tabela spatial_ref_sys no *Software PostgreSQL*.



Fonte: Autoria própria (2017).

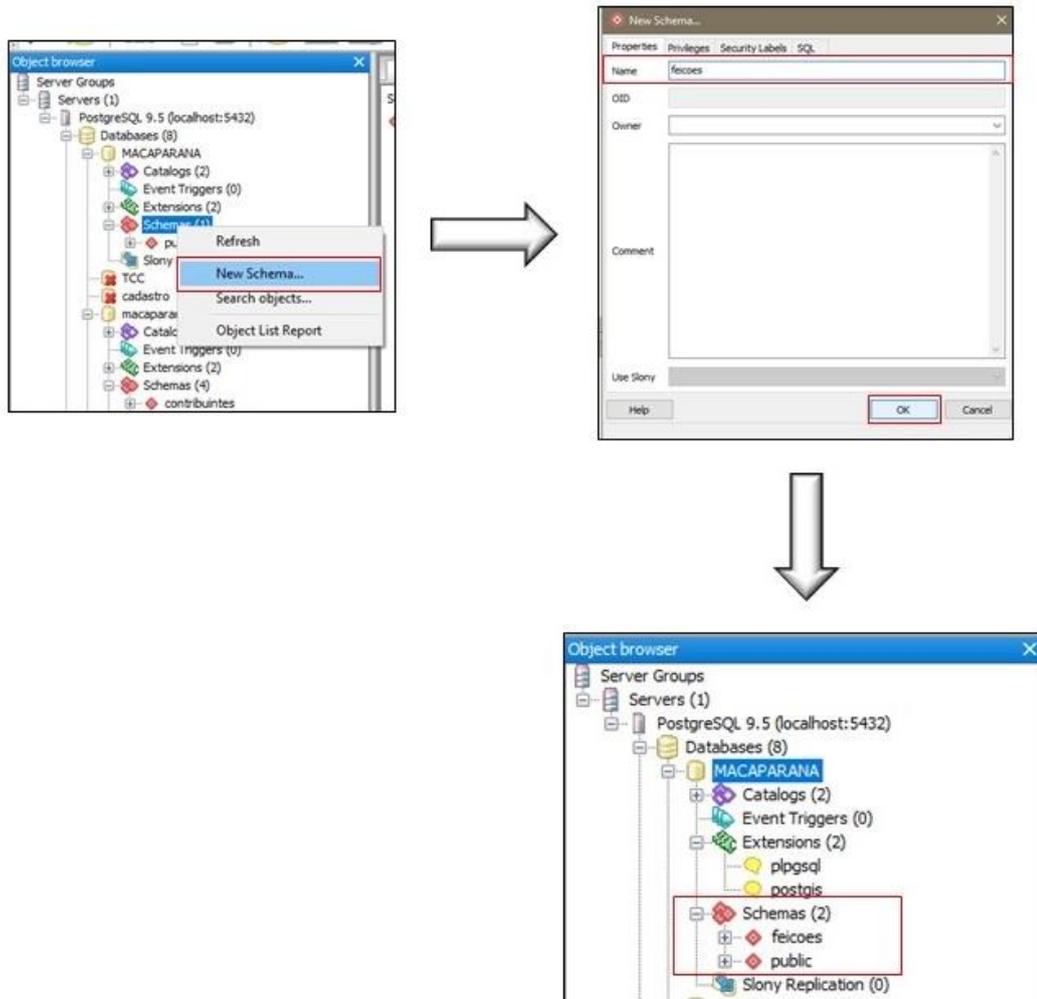
O PostgreSQL apresenta um suporte denominado de “Schema” no qual consiste em um espaço lógico dentro do banco de dados onde é possível realizar o armazenamento de objetos, ou seja, todas as informações contidas dentro do banco de dados são organizadas nesse suporte,

local destinado para criação e manipulação de tabelas, possibilitando o relacionamento de objetos, existentes no mesmo banco de dados, mas em estruturas lógicas (esquemas) diferentes.

Portando, como em qualquer prefeitura existe subdivisões (como por exemplo, secretaria de educação, secretaria de saúde, finanças, etc.), optou-se por estabelecer alguns “schemas” com a finalidade de melhorar a estruturação do banco de dados. Os “schemas” criados foram: feições, para guardar as informações das tabelas referentes as feições; contribuintes, para guardar os dados da tabela com o cadastro dos colaboradores do município; funcionários, para guardar os dados da tabela com o cadastro dos servidores do município. O “schema” public, gerado automaticamente a partir do momento que se cria um novo banco de dados, foi mantido inalterado, sem motivos específicos, e utilizado apenas para receber as informações geométricas que seriam construídas posteriormente.

Para cada “schema” criado, a sequência de passos foram as mesmas. Com botão direito do mouse sobre o suporte principal, seleciona-se a opção “New Schema”, após isso escolhe-se o nome do esquema na opção “Name”, atualizando o processo no final para que o resultado gerado seja apresentado no navegador de objeto (Figura 8).

Figura 8 – Etapas para criação de um novo “schema” no *Software PostgreSQL*.



Fonte: Autoria própria (2017).

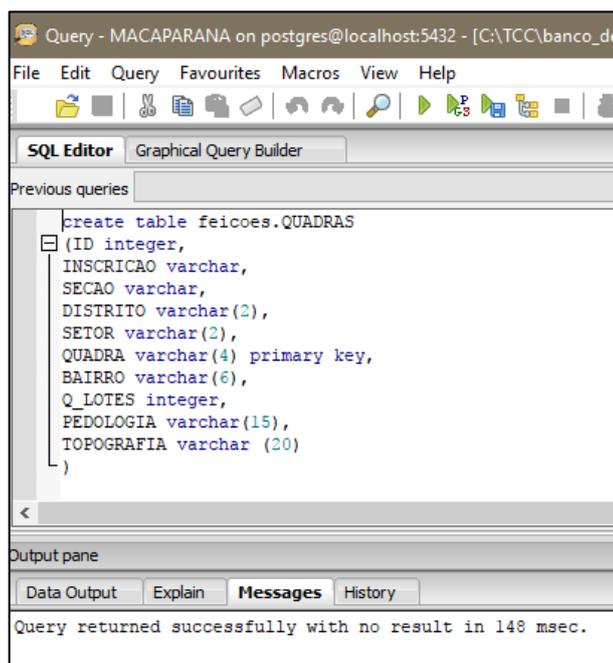
Os dois últimos estágios do tratamento dos dados descritivos consistiram na criação e alimentação das tabelas que comporão o banco de dados.

Existem duas maneiras para criação de uma tabela no PostgreSQL, uma semiautomática e uma manual utilizando comandos SQL. A opção adotada para o presente trabalho foi a manual, onde para realização desse procedimento, seleciona-se, no schema desejado, a opção

“Tables” e em seguida na barra de ferramenta o ícone “Execute arbitrary SQL queries”



. Na área de edição SQL são escritos os comandos necessários para criação da tabela, como mostrado na figura abaixo.

Figura 9 – Comando na linguagem SQL para criação de tabelas no *Software PostgreSQL*.

```
Query - MACAPARANA on postgres@localhost:5432 - [C:\TCC\banco_de
File Edit Query Favourites Macros View Help
SQL Editor Graphical Query Builder
Previous queries
create table feicoes.QUADRAS
(ID integer,
INSCRICAO varchar,
SECAO varchar,
DISTRITO varchar(2),
SETOR varchar(2),
QUADRA varchar(4) primary key,
BAIRRO varchar(6),
Q_LOTES integer,
PEDOLOGIA varchar(15),
TOPOGRAFIA varchar (20)
)
Output pane
Data Output Explain Messages History
Query returned successfully with no result in 148 msec.
```

Fonte: print screen da aplicação no *Software PostgreSQL*.

Algumas considerações devem ser discutidas ao analisar a figura anterior. A primeira diz respeito a necessidade de indicar o “schema” onde será criada a tabela logo após o comando “create table ...”, pois caso essa designação não seja realizada todas as tabelas serão criadas no “schema” padrão, o public.

A segunda diz respeito ao tipo de variável que deve conter em cada registro, que de acordo com a estrutura dos dados que serão armazenados poderão ser do tipo inteiro, real, lógico, data, caracteres, texto, binários e entre outros. Vale ressaltar a importância da limitação do tamanho do campo onde será arquivada as informações, para casos de dados do tipo decimal, texto e caractere, por exemplo, pois fazendo isso possibilitará um consumo menor de memória no banco de dados facilitando até mesmo a realização de buscas no mesmo.

A terceira e última observação, diz respeito a ligação entre diferentes tabelas através de chaves primárias e secundárias, isto é, sempre que existir a relação 1:N entre duas tabelas, a tabela 1 receberá a chave primária e a tabela N receberá a chave estrangeira. Examinando a Figura 12 verifica-se que na tabela intitulada por “quadra” apresenta como chave primária o campo “QUADRA” devido a presença do comando “primary key”, logo a tabela na qual existe relação com a tabela “quadra” apresentará o mesmo campo “QUADRA”, porém será uma chave estrangeira. A Tabela 1 mostra uma lista com as tabelas utilizadas no trabalho e suas respectivas relações.

Tabela 1 - Relação entre tabelas presentes no banco de dados por meio de chaves primárias e secundárias.

TABELA	Chave Primária	Chave Secundária
QUADRAS	quadra	-
LOTES	lote	quadra
EDIFICAÇÕES	-	lote

Fonte: Autoria própria (2017).

Na etapa de alimentação das tabelas do banco, utilizou-se as tabelas pré-organizadas no editor de planilha. Basicamente os passos iniciais são os mesmos dos realizados para criação da tabela, diferenciando na etapa de edição do código SQL que neste caso empregou-se o comando “copy” em vez do “create table”, como mostrado na Figura 10.

Figura 10 – Comando na linguagem SQL para inserção de dados em tabelas no *Software* PostgreSQL.

```

copy feicoes. QUADRAS
(ID,
INSCRICAO,
SECAO,
DISTRITO,
SETOR,
QUADRA,
BAIRRO,
Q_LOTES,
PEDOLOGIA,
TOPOGRAFIA
)
from 'C:\TCC\banco_de_dados\tabelas\quadras.csv'
delimiter ';'
csv header;

```

Fonte: print screen da aplicação no *Software* PostgreSQL.

A estrutura do código é bastante parecida com a da mostrada na Figura 9, apresentando duas peculiaridades, além do uso do comando principal como falado anteriormente. A primeira diz respeito a indicação das colunas nas tabelas, das quais devem apresentar a mesma nomenclatura da tabela original existente no banco não necessitando a indicação do tipo da variável, pois esse processo apenas indica onde os valores deverão ser preenchidos. A segunda diz respeito ao comando sublinhado de vermelho na figura, onde indica o caminho do arquivo

.csv que contém os dados a serem inseridos, sendo necessário a designação de todas pastas onde estão guardados os documentos.

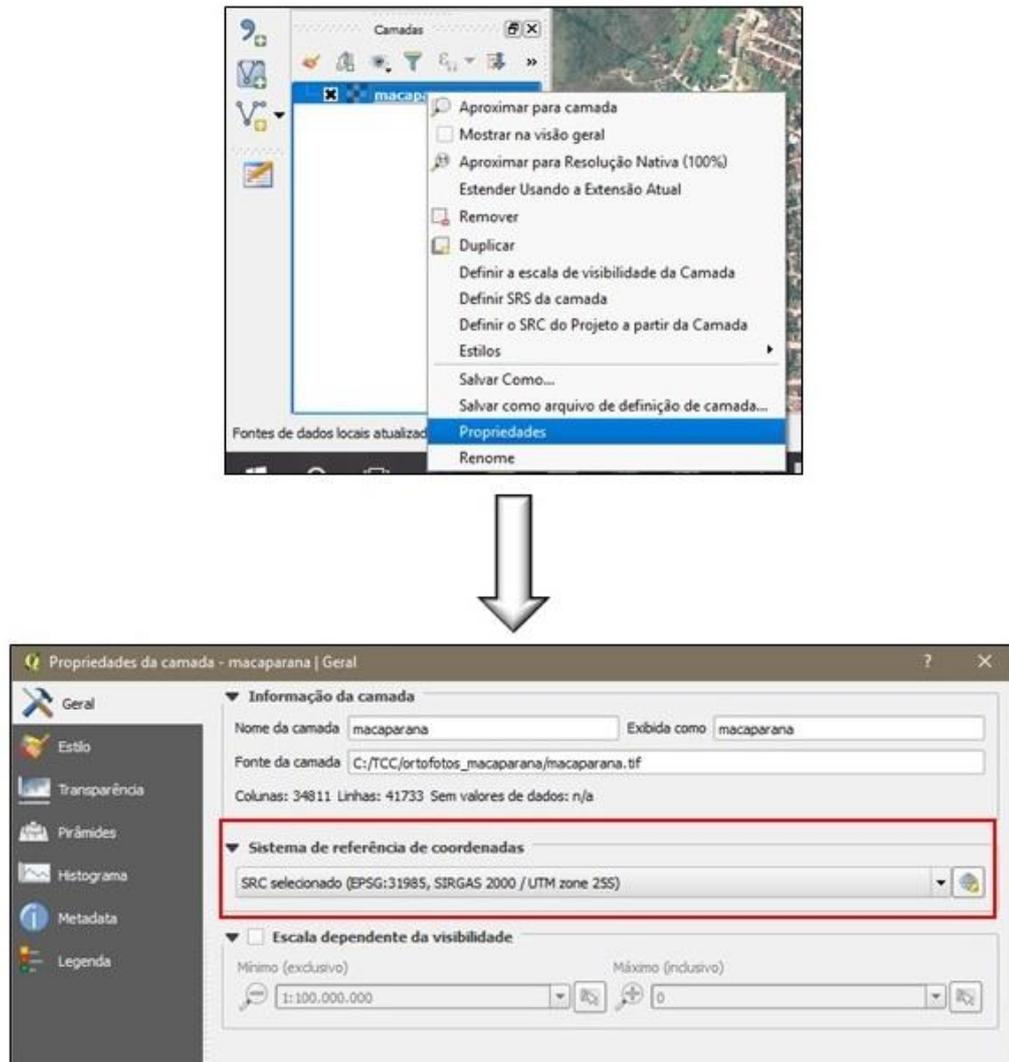
Os procedimentos indicados nesta seção, apesar de terem sido mostrados apenas para um exemplo, foram reproduzidos em todos os arquivos básicos do projeto, apresentando no Apêndice A, Apêndice B e Apêndice C os processos executados nos demais.

5.3.2 Tratamento dos dados geométricos

Conforme visto no item 5.1, o único dado geométrico utilizado para geração dos produtos cartográficos da área de estudo, foram as ortofotos obtidas na CONDEPE/FIDEM, devido as questões discutidas no mesmo item.

Antes de iniciar qualquer tipo de manipulação nas fotografias, verificou-se nas mesmas, o Sistema de Referência na qual se encontravam, se realmente estavam condizentes ao sistema adotado atualmente no Brasil, o SIRGAS 2000. Essa averiguação, assim como toda operação realizada nesta fase, foi efetuada no *Software* QGIS. Logo, com as ortofotos devidamente carregadas no programa, a verificação pode ser feita clicando-se com o botão direito do mouse sobre a ortofoto, localizada na janela “Camadas”, selecionando-se a opção “Propriedades” e na aba “Geral” da janela aberta observa-se a opção presente na caixa de “Sistema de referência de coordenadas” (Figura 11).

Figura 11 – Processo de verificação do Sistema de Referência das ortofotos no *Software* QGIS.

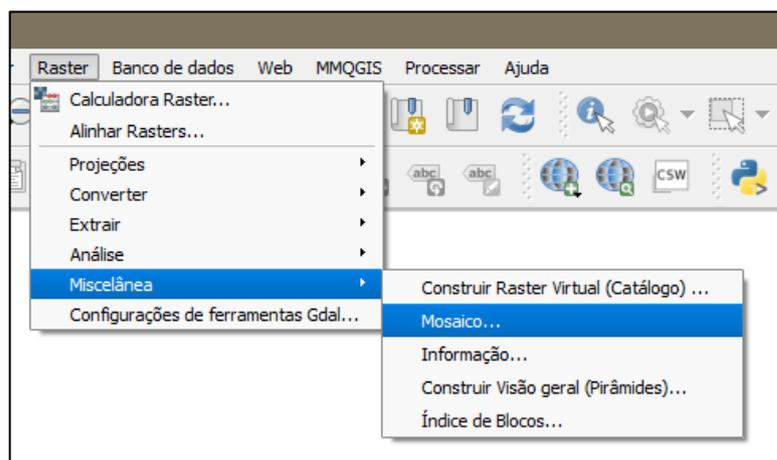


Fonte: print screen da aplicação no *Software* QGIS.

Caso após essa verificação as informações apresentarem divergências, será necessário reprojotá-las, adequando-as aos padrões empregados. Para o projeto não foi necessário realizar esse procedimento, pois as informações relativas ao sistema de referência estavam segundo o modelo brasileiros.

Para facilitar o manuseio das informações, realizou-se um mosaico com as ortofotos do município, seguindo a sequência através do menu Raster → Miscelânea → Mosaico (Figura 12). Vale ressaltar que para efetuar esse processo é necessário que ambas as imagens estejam com o sistema de referência e de coordenadas compatíveis, pois a ocorrência de incompatibilidade interromperá o curso do procedimento.

Figura 12 – Ferramenta utilizada para realização de mosaicagem no *Software* QGIS.



Fonte: print screen da aplicação no *Software* QGIS.

Como os dados geométricos coletados eram inapropriados para o uso, foi necessário vetorizar, sobre o mosaico realizado, os elementos básicos existentes que servirão de apoio para estruturação do geoportal, dentre os quais pode-se citar as quadras, o sistema viário, os lotes, as edificações e a rede hídrica.

Antes de iniciar o processo de vetorização, é preciso criar uma camada para cada feição que será produzida, utilizando o menu Camada → Criar nova camada → Shapefile (Figura 13)

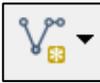
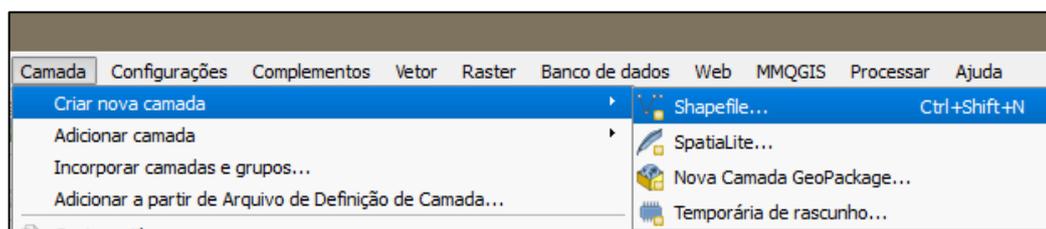
ou pela barra de ferramenta clicando no ícone Shapefile .

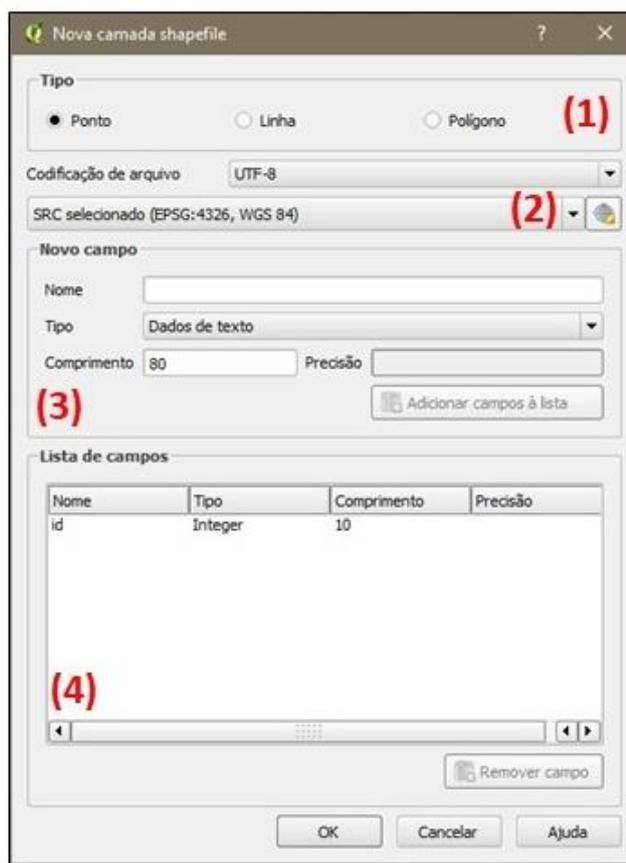
Figura 13 – Ferramenta de criação de camada vetorial no *Software* QGIS.



Fonte: print screen da aplicação no *Software* QGIS.

A janela de criação de uma camada é composta por quatro partes, como mostrada na Figura 14.

Figura 14 – Janela de criação de camada vetorial *Software* QGIS.



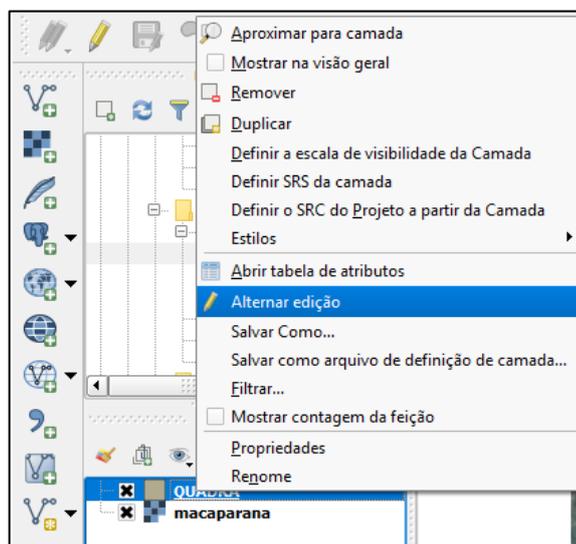
Fonte: print screen da aplicação no *Software* QGIS.

A primeira parte (1) diz respeito a geometria que será utilizada para representar a feição, podendo ser do tipo ponto, linha ou polígono. Nesse projeto utilizou-se como conversão cartográfica o tipo linha para representar o sistema viário e os cursos d'águas e o tipo polígono para descrever as quadras e os lotes. A segunda parte (2) está relacionada com a codificação de arquivo e a escolha do SRC – Sistema de Referência de Coordenadas, onde as opções selecionadas foram UTF-8³ e SIRGAS 2000 / UTM zona 25, respectivamente. Na terceira parte (3) é possível adicionar novos registros a tabela de atributos de cada feição, evitando-se nomes longos, com acentuação e espaços, podendo ser especificada a natureza do registro selecionando-se umas das opções oferecidas pelo programa: texto, número inteiro, número decimal ou data. É possível ainda, na opção “Comprimento”, indicar a quantidade de caracteres que o atributo pode suportar, assim como número de dígitos que devem aparecer após a vírgula no campo “Precisão”. A cada finalização de um novo atributo, clica-se em “Adicionar à lista de atributos” e os mesmos serão inseridos na “Lista de campos”, que é a quarta parte da janela.

³ Codificação binária (UNICODE) utilizada para representar qualquer caracter universal padrão.

Após a criação das camadas, deu-se início ao processo de vetorização. Para isso, basta selecionar a camada a qual deseja-se alterar e com o botão direito do mouse clica-se na opção “Alternar edição” (Figura 15) ou no ícone “Alternar edição”  localizado na barra de ferramenta.

Figura 15 – Comando de “Alternar edição” a partir da camada selecionada no *Software* QGIS.



Fonte: print screen da aplicação no *Software* QGIS.

Para iniciar a digitalização clica-se no ícone “Adicionar feição”  e na ortofoto procura-se os objetos dos quais desejam-se vetorizar, percorrendo toda geometria desejada. Completado o processo, clica-se com o botão direito do mouse e uma tela solicitará que adicione-se os valores nos campos gerados na fase de criação da camada. Inicialmente os registros criados em todas as feições guardavam apenas seus nomes, onde as informações utilizadas para preencher as tabelas de atributos das camadas de quadra e rede viária foram obtidas do arquivo .dwg que continha a sede da cidade. Na camada de curso d’água, apenas o percurso que o Rio Capibaribe Mirim faz foram introduzidas nos campos, os demais trechos não se obtiveram informações referentes.

Com respeito a vetorização dos lotes e edificações, como as informações descritivas obtidas na prefeitura foram apenas de uma porção do município, como comentado no item 5.1, digitalizou-se apenas as informações relativas a essa amostra, ou seja, a quadra 001.

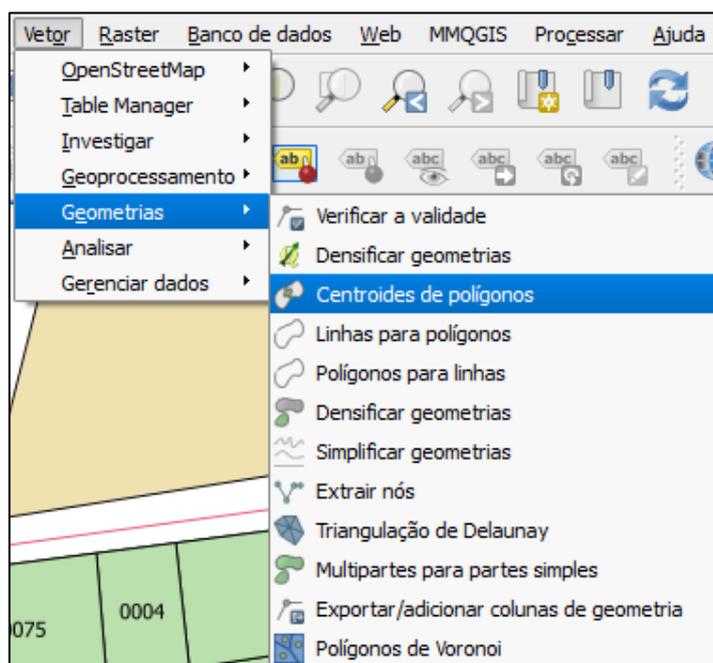
Algumas incompatibilidades surgiram ao realizar esse processo. A primeira está relacionada a quantidade de dados descritivos relativo ao número de lotes da quadra, que eram superiores a realidade apresentada pela ortofoto. Um segundo problema diz respeito a

identificação dos lotes, uma vez que não havia informações concretas nos documentos geométricos cedido pela prefeitura no qual indicasse a nomeação real desses lotes.

Sendo assim, para solucionar essas dificuldades optou-se por nomear os lotes de forma aleatória segundo as informações organizadas na tabela de lotes criada na fase 5.3, no qual pode ser vista no Apêndice A, obedecendo apenas a ordem de indicação das ruas para o qual a frente do terreno era voltada. Vale ressaltar que o presente trabalho é apenas uma proposta para implantação de algo posterior e que o objetivo é mostrar a interação entre dados geométricos e descritivos, ou seja, nenhuma informação será realmente disponibilizada para uso pois as mesmas precisam passar por um processo de atualização para evitar inconsistências dessa natureza.

Um outro fato que deve ser analisado se refere a impossibilidade de se identificar os limites reais da unidade predial. Logo, optou-se por representar as edificações por meio de feições do tipo ponto, extraindo-se os centroides dos polígonos dos lotes por meio do menu Vetor → Geometrias → Centroides de polígonos, como mostrado na Figura 16.

Figura 16 – Processo de extração de centroides de uma camada do tipo polígono no *Software* QGIS.



Fonte: print screen da aplicação no *Software* QGIS.

O último passo a ser realizado no tratamento das informações geométricas foi a organização das tabelas de atributos de cada feição, pois era necessário prepará-las para que as informações descritivas contidas no banco de dados pudessem ser interligadas com as mesmas.

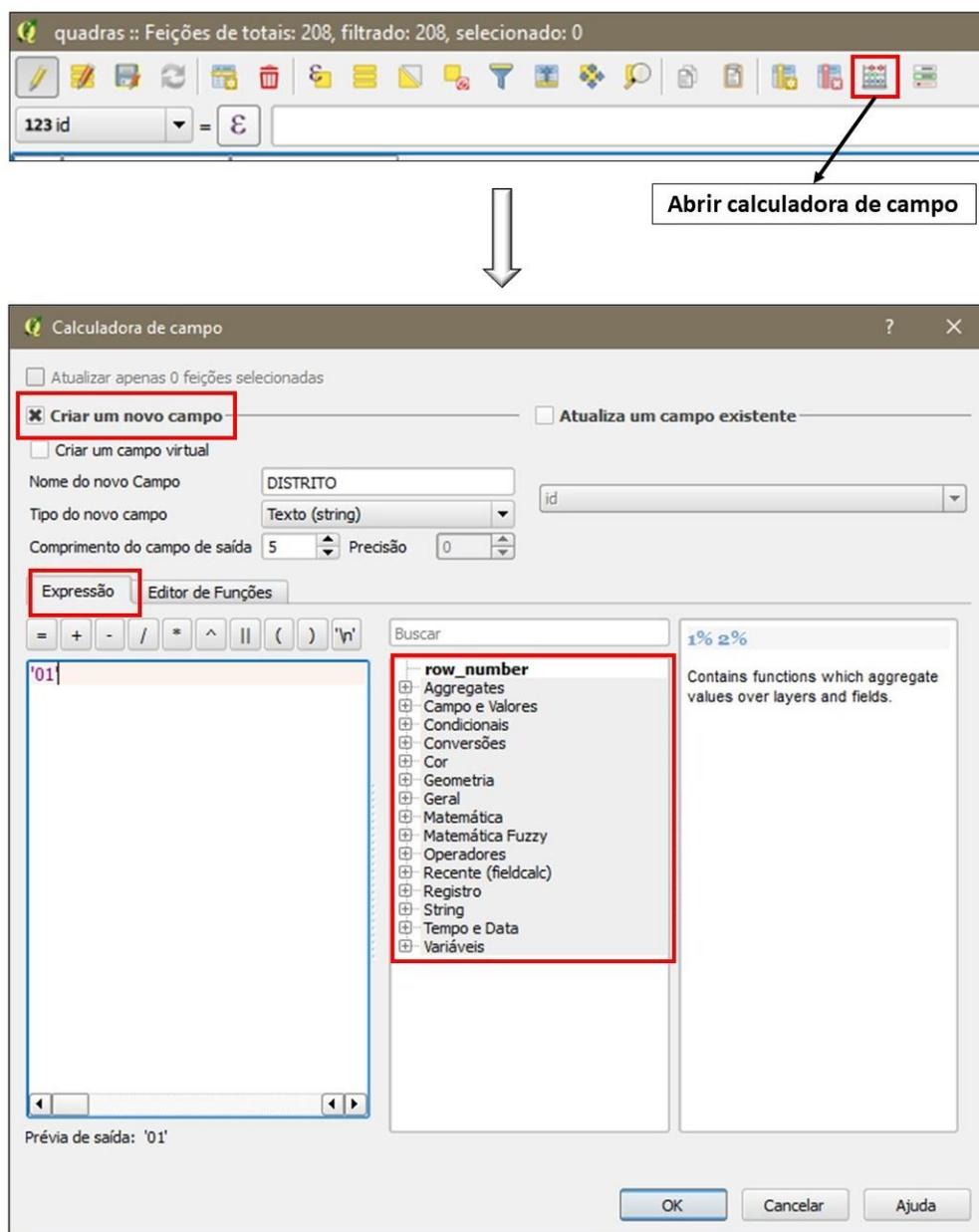
Baseando-se nas tabelas existentes no banco de dados, pode-se verificar que em cada uma possui um número de inscrição composto por uma sequência de caracteres, que obedecem a seguinte ordem:

- Quadras - o código de inscrição é composto pelo número do distrito, seguindo do setor e nome da quadra;
- Lotes - o código de inscrição é composto pelo número do distrito, seguindo do setor, nome da quadra e do lote;
- Edificações - o código de inscrição é composto pelo número do distrito, seguindo do setor, nome da quadra, o lote e o número predial.

A edição da tabela de atributos de cada feição foi realizada acionando-se o botão “alternar edição”, o mesmo utilizado para iniciar a vetorização dos dados, e em seguida selecionando a opção “Abrir tabela de atributos”, onde na janela aberta pôde-se realizar alterações inserindo-se as informações nos campos segundo a necessidade de cada camada.

Como por exemplo, na feição quadra foram inclusas as colunas de distrito e setor por meio da opção “Abrir calculadora de campo”. Na janela de calculadora de campo cria-se um novo campo, informando o nome, tipo e comprimento desejado e na área de “Expressão” informa-se o valor do qual deseja-se preencher a coluna, sendo possível também realizar algum tipo de equação utilizando as sentenças presentes na caixa “row_numer” (Figura 17).

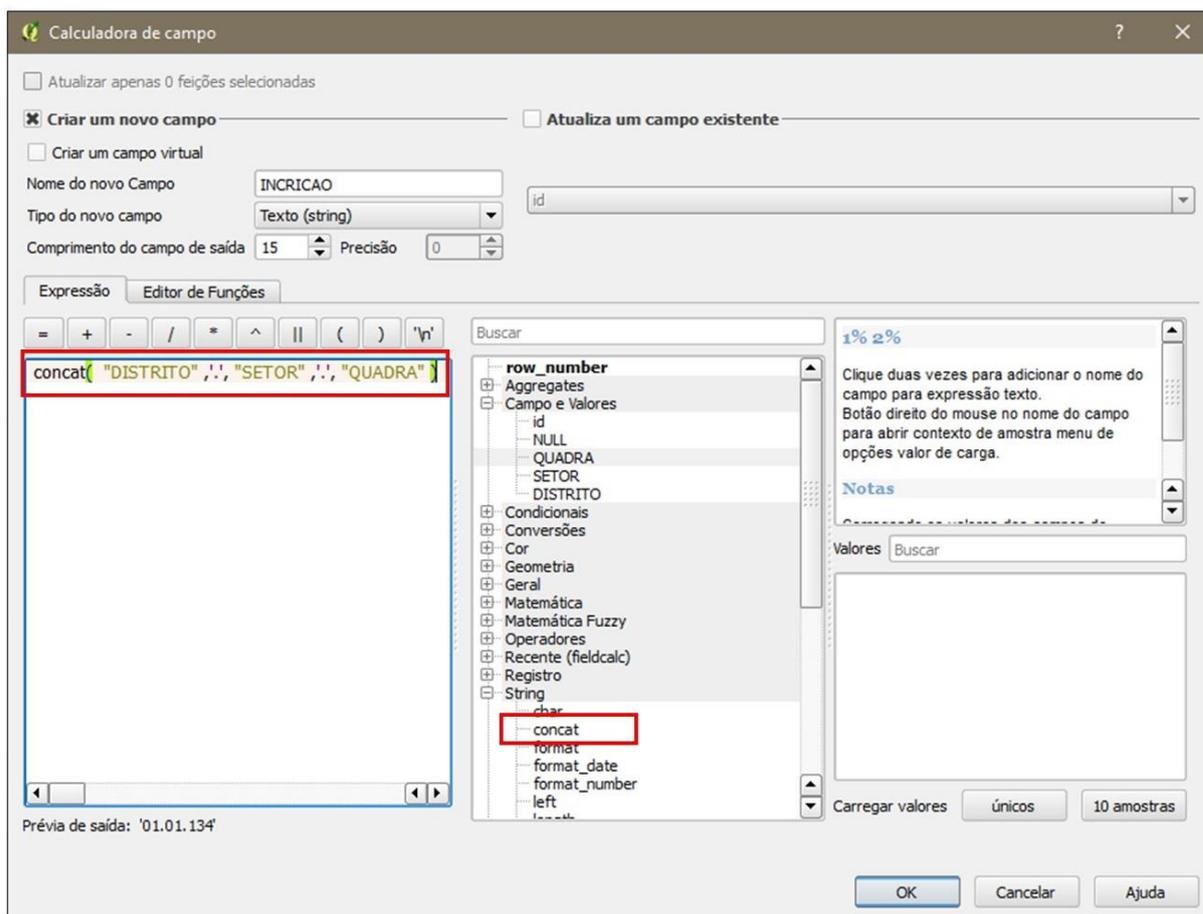
Figura 17 – Processo de criação de uma nova coluna na tabela de atributos em uma camada vetorial no *Software QGIS*.



Fonte: Autoria própria (2017).

A mesma sequência foi realizada para criação da coluna de setor e após gerou-se o código de inscrição das quadras na janela de “Calculadora de campo” utilizando o comando “concat”, caixa “row_number”. no qual concatenou os valores dos campos das colunas “DISTRITO”, “SETOR” e “QUADRA” (Figura 18).

Figura 18 – Processo de concatenação dos campos para geração do código de ligação entre as tabelas no *Software* QGIS.



Fonte: print screen da aplicação no *Software* QGIS.

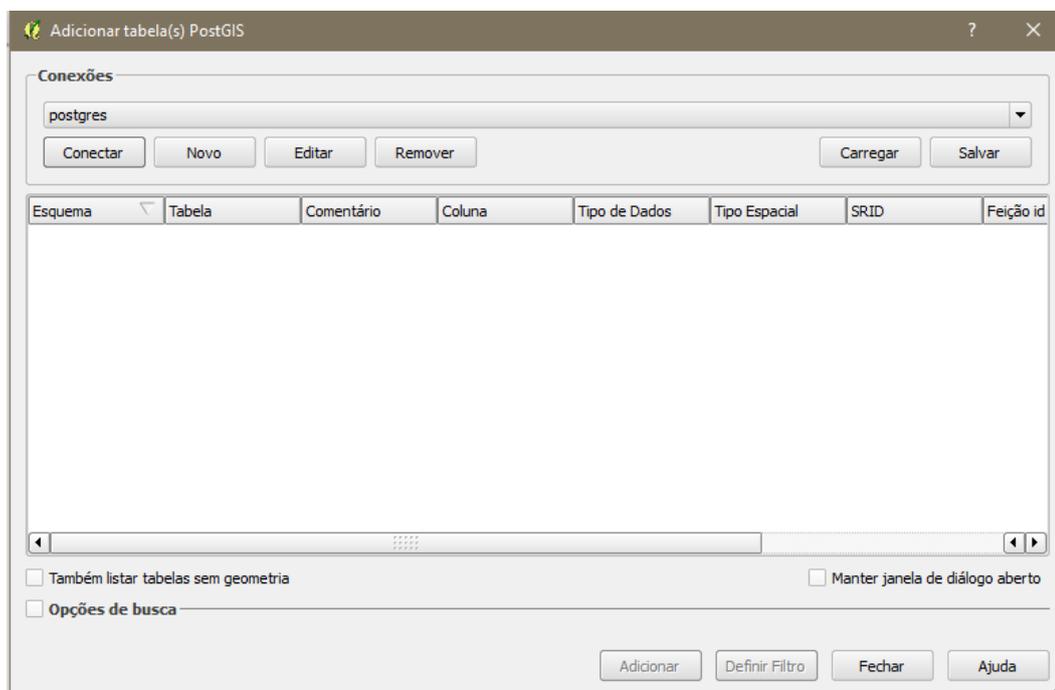
5.3.3 Conexão entre os dados descritivos e geométricos

Com os dados descritivos e geométricos organizados, o próximo passo consistiu em realizar a interligação entre eles. É importante que todo cuidado tenha sido tomado nas etapas anteriores para que a ligação possa obter sucesso, pois esse procedimento se dará por meio das uniões dos códigos de ligações entre as tabelas contidas no banco de dados e a feições geradas, onde esses códigos são uma sequência de caracteres únicos e não nulos, no qual para o presente trabalho escolheu-se para essa função as inscrições que identificam as quadras, os lotes e as edificações.

A conexão é feita através do *Software* PostGIS no qual se encontra presente sua extensão tanto do QGIS quanto no PostgreSQL.

Primeiramente, com o projeto aberto no QGIS, para adicionar as tabelas de um banco de dados existentes no PostgreSQL aciona-se o ícone PostGIS  na barra de ferramenta e inicia-se o processo de conexão entre o programa de SIG e o gerenciador de banco de dados por meio de uma janela como mostrada na figura a seguir.

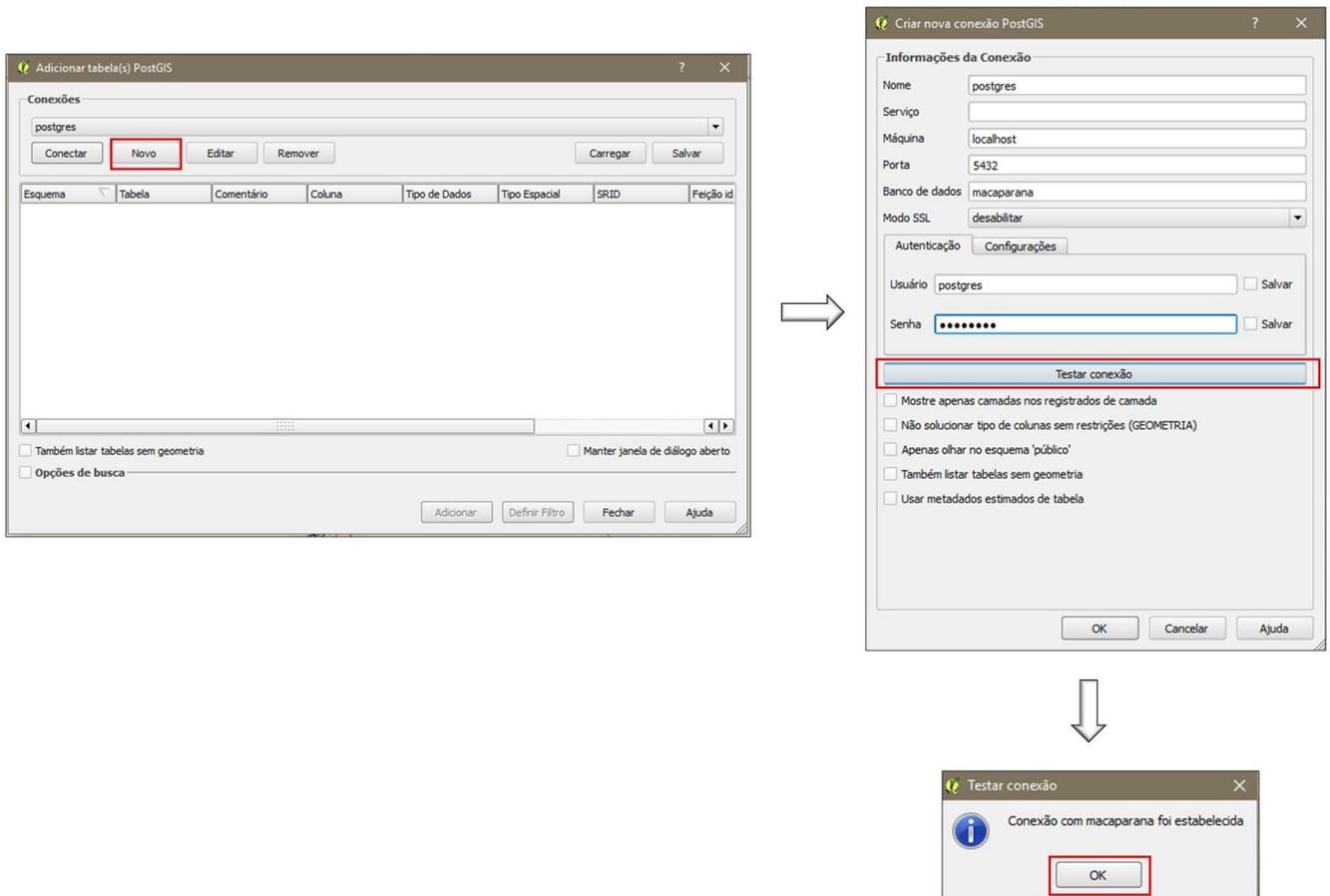
Figura 19 – Janela de conexão com o banco de dados do PostgreSQL no *Software* QGIS.



Fonte: print screen da aplicação no *Software* QGIS.

Nessa janela, cria-se uma nova conexão com a base de dados por meio do botão “Novo” e em seguida preenche-se as informações necessárias para se realizar o processo, tais como nome da conexão, o serviço (caso exista), a máquina (onde nessa opção colocou-se localhost pelo fato do SBGD está conectado a uma máquina local), a porta do *Software*, o nome do banco de dados do qual se deseja conectar, usuário e senha caso possua. Completado os dados básicos verifica-se a conexão no botão “Testar conexão”, caso o procedimento tenha obtido sucesso ou não aparecerá uma caixa informando o resultado final da adição do novo banco. Esses processos podem ser visualizados na Figura 20.

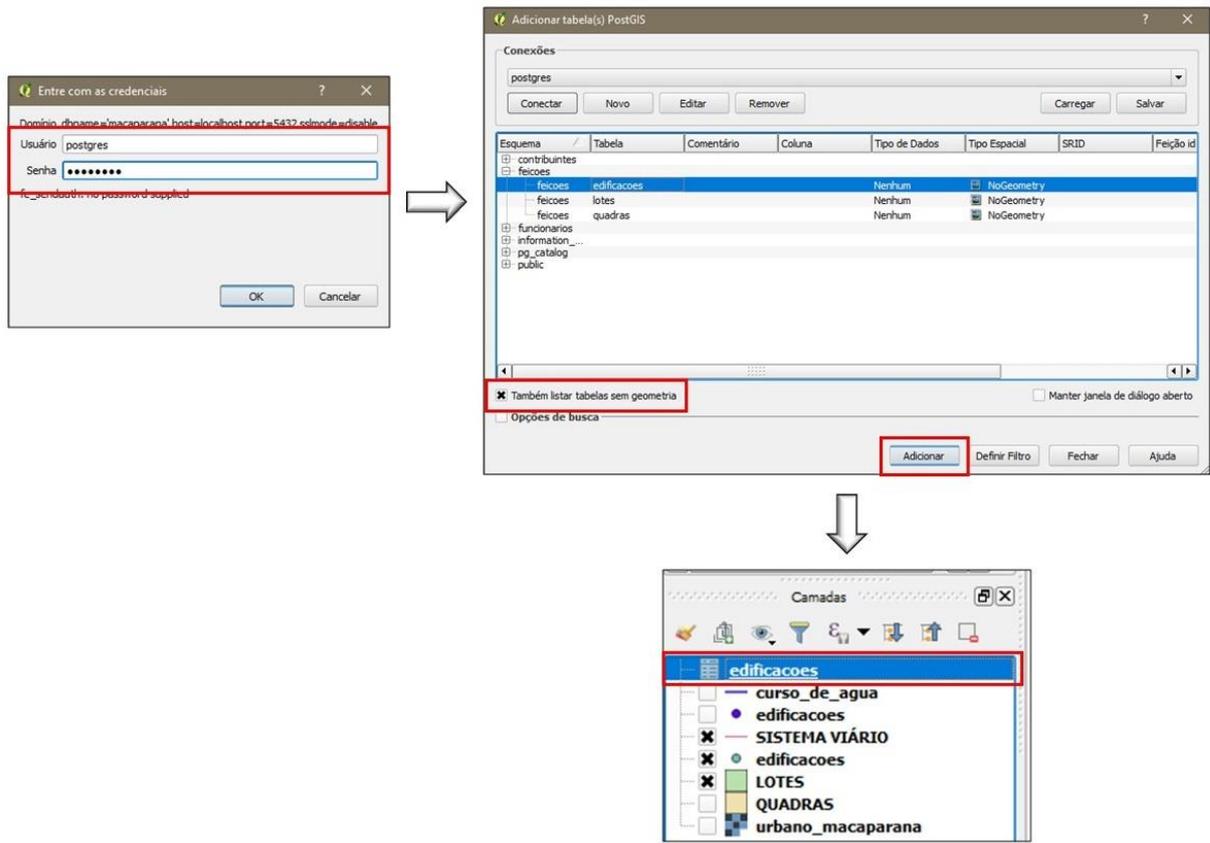
Figura 20 – Etapas para adição de um novo banco de dados do PostgreSQL no *Software* QGIS.



Fonte: Autoria própria (2017).

Feito isso, o usuário será direcionado novamente a interface mostrada na Figura 19, onde no botão “Conectar” irá estabelecer verdadeiramente a conexão com o banco de dados cadastrado anteriormente. Em alguns casos o programa pedirá para entrar com as credenciais, se o SGBD apresentar senha de acesso, e logo após será carregado as informações presentes no banco de dados, sendo necessário ativar a caixa “Também listar tabelas sem geometria” para que as tabelas que não apresentem dados geométricos possam aparecer na interface da janela. Em seguida seleciona-se a tabela desejada e clica-se em “Adicionar” e o resultado esperado se encontrará listado na janela de “camadas” do QGIS (Figura 21).

Figura 21 – Processo de conexão de uma tabela do banco de dados do PostgreSQL no *Software* QGIS.



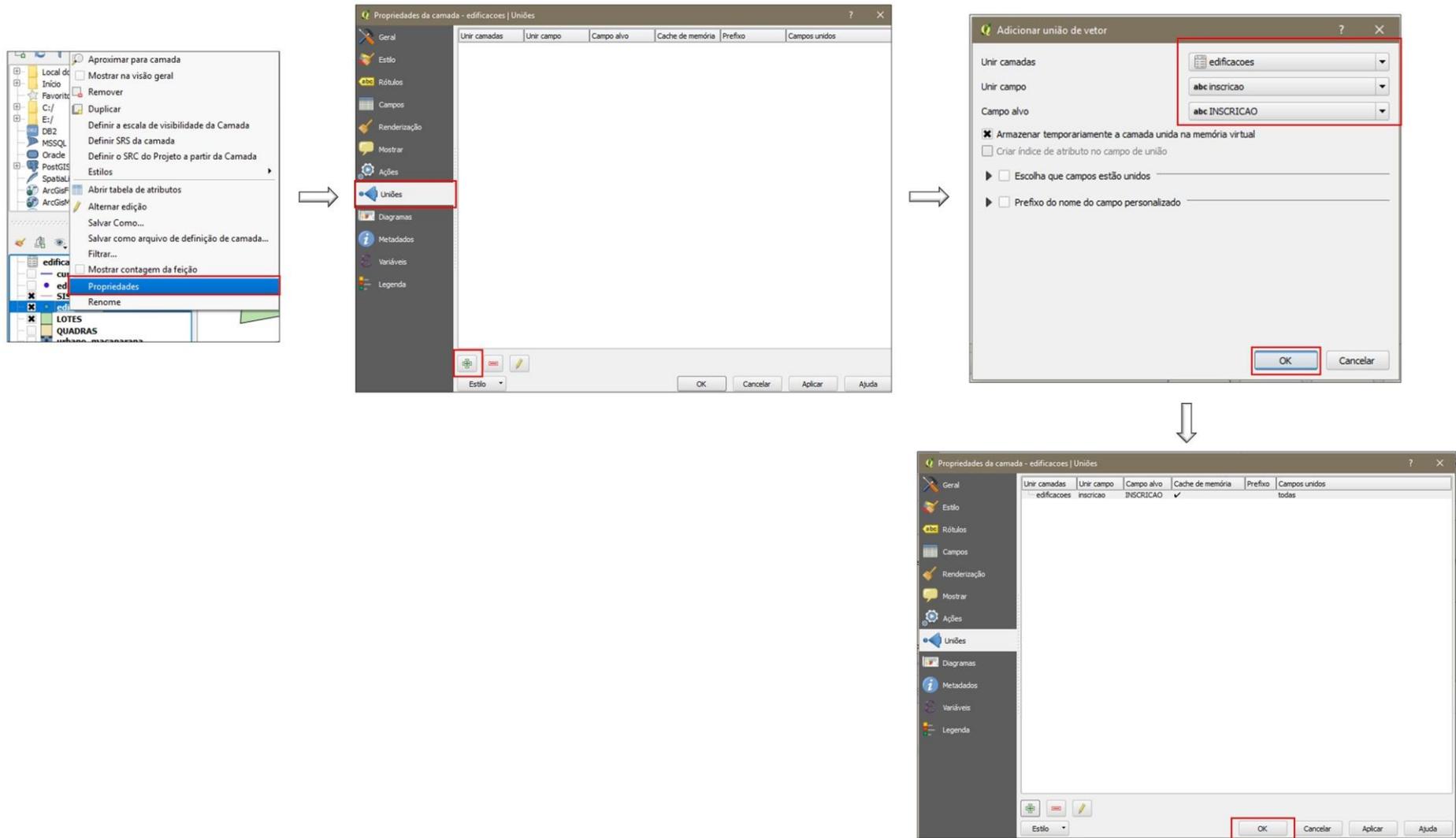
Fonte: Autoria própria (2017).

Em seguida, uni-u-se a tabela adicionada do banco de dados com a camada referente, clicando com o botão direito do mouse sobre a feição desejada, selecionando a opção “Propriedades”. Na janela de “Propriedades da camada”, na aba “União” clica-se no ícone



e uma janela chamada “Adicionar união de vetor” se abrirá, onde será possível escolher a camada na qual deseja-se unir e por quais campos devem-se realizar esse processo. Vale ressaltar que esses campos, pelo qual se realizou a união, não necessariamente devem possuir a mesma nomenclatura, porém os valores que preenchem as colunas devem iguais para que a relação possa ser estabelecida. As etapas seguidas para união dos dados descritivos com os geométricos apresentam-se resumidamente na Figura 22.

Figura 22 – Processo de união entre os dados descritivos e geométricos no *Software* QGIS.



Fonte: Autoria própria (2017).

Para verificar se o procedimento foi realizado com sucesso basta abrir a tabela de atributos das feições e analisar se os dados foram adicionados juntamente com os valores já existentes. É necessário que um novo arquivo seja salvo para que os campos adicionados sejam definitivamente unidos à geometria, pois a não realização desse procedimento não garante que após o encerramento do projeto os dados permanecerão salvos.

Logo após as tabelas de atributos de cada feição passaram por uma edição, onde foram excluídas algumas colunas que ficaram repetidas, devido a existência das mesmas tanto nos dados descritivos quanto nos geométricos. Também passaram por um processo de renomeação

por meio da ferramenta “Table manager” , tendo-se a opção de alteração na ordem de sequência das colunas.

Após a união dos dados descritivos com seus referentes geométricos, esses arquivos foram enviados para o banco de dados levando consigo não apenas a tabela resultante da união, mas também as informações espaciais existentes. Esse procedimento garante a segurança e integridade dos produtos cartográficos produzidos, servindo como um meio de se manterem registrados juntamente com as demais informações presentes, reunindo em uma única base dados de natureza distintas.

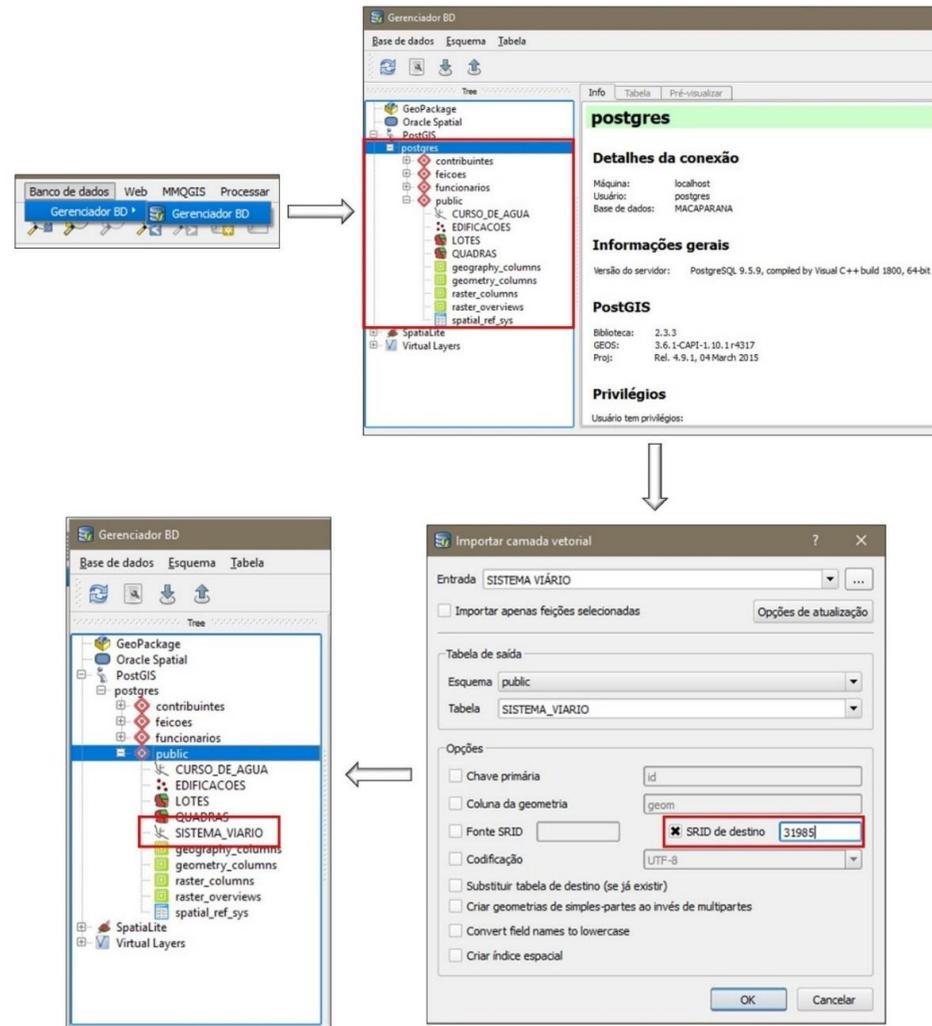
Existem três maneiras para se realizar esse procedimento, uma é acessando diretamente o PostGIS *Shapefile* Import/Export Manager, a segunda é pelo próprio SGBD do qual possui a extensão espacial e a última é pelo *Software* de QGIS.

O meio escolhido para enviar os arquivos ao banco de dados foi através da ferramenta QGIS, pelo fato do programa possuir um *Plugin* que conecta diretamente com o SGBD em estudo, realizando o processo de maneira prática encaminhando os dados que estão em amostra no projeto.

Para isso, antes é necessário estabelecer uma conexão do QGIS com o PostgreSQL como mostrado no início da seção, porém sem a necessidade de se adicionar nenhuma tabela do banco. Posteriormente, seleciona-se na barra de menu a opção Banco de dados → Gerenciador BD. Na janela de “Gerenciador de BD” mostrará uma lista de extensões espaciais que o programa pode se conectar, onde selecionando-se a opção PostGIS → postgres e em seguida “importar camada/arquivo”, preenche-se com as informações necessárias para dá início ao processo de introdução dos dados ao banco, onde será criada uma nova tabela contendo os dados descritivos da camada, assim como a inclusão de uma nova coluna na tabela contendo a informação espacial. É importante frisar a necessidade de se indicar o Sistema de Referência, na opção “SRID destino”, na qual as feições se encontram no ato de envio dos dados, pois a

não identificação do mesmo causará inconsistência e perda na confiabilidade dos dados. Os processos mostrados podem ser observados na Figura 23.

Figura 23 – Representação do processo de envio das camadas vetoriais ao SGBD através do *Software* QGIS.



Fonte: Autoria própria (2017).

Os métodos realizados neste item embora foram desenvolvidos para um modelo, foram repetidos em todos dados elementares da pesquisa.

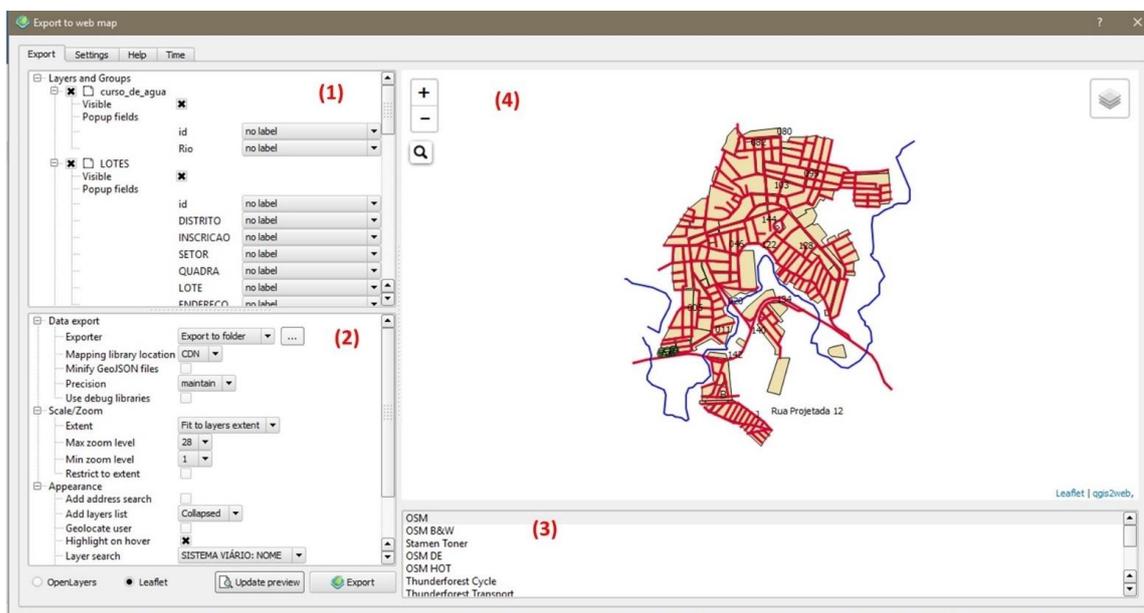
5.3.4 Envio dos dados à *Internet* e edição do geoportal

A última etapa dos procedimentos metodológicos compreendeu em enviar todos os produtos cartográficos produzidos, unidos com suas respectivas informações descritivas, à *Internet*, dando início a criação e edição do geoportal.

Para esse processo utilizou-se o plugin qgis2web, que gera um mapa da *Web* a partir de um projeto no QGIS.

Com o plugin devidamente instalado, para dar início ao envio utiliza-se o menu *Web* → qgis2web → create web map. A janela principal de exportação do mapa é composta por quatro áreas como mostrada na figura a seguir.

Figura 24 – Interface da janela de exportação de mapa para Web no *Software* QGIS.



Fonte: Autoria própria (2017).

A primeira área (1) corresponde ao local destinado para escolha dos layers que comporão o mapa, sendo possível a opção de ativá-los ou desativá-los segundo a vontade do usuário. Lembrando que só aparecerão nessa área as camadas que se encontrem em um projeto principal, não sendo possível adicionar ou criar novos arquivos no ambiente de exportação. Na segunda área (2), apresenta a opção do local onde será salvo os dados, assim como os principais

elementos que podem ser inseridos ao mapa, tais como: barra de zoom, janela de pesquisa, ferramenta de medição, opção de utilização das bibliotecas JavaScript OpenLayer ou Leaflet, entre outros. Na terceira área (3) encontra-se uma vasta lista de opções de mapa base prontos que podem ser introduzido ao projeto de forma a complementar o produto. E por fim, na quarta área (4) pode-se ter uma prévia do que será visto na página *Online*, podendo ser atualizada no botão “Update preview” sempre que se alterar algum elemento do mapa.

Concluído a edição dos dados que constituirão o mapa, antes de exportá-lo é necessário configurar onde o projeto será salvo na opção “Exporter” do item “Data export”, localizada na área (2) como mostrada na Figura 27, e em seguida clica-se no “Export”.

Uma vez executada a exportação, o navegador padrão do computador será aberto e mostrará na *Web* o mapa interativo gerado. Vale destacar que apenas as camadas vetoriais foram selecionadas para o envio à Internet, não sendo possível enquadrar o mosaico das ortofotos do município, pois o mesmo se encontrava com um tamanho do qual dificultava seu envio e não se obteve sucesso na tentativa de se reduzi-lo.

Apesar do plugin qgis2web se mostrar eficiente e de fácil manipulação, ele apresenta algumas limitações, pois é criado apenas um modelo base no qual faz uso de linguagens computacionais, tais como a linguagem de marcação HTML, para estruturação da página *Online*, assim como a formatação JSON do JavaScript que realiza as animações do *Site*. Sendo assim, os arquivos presentes na pasta do computador onde o projeto foi salvo, podem ser facilmente personalizados de acordo com as necessidades de quem o manipula.

Partindo desse pressuposto de personalização, o próximo passo compreendeu na estruturação do geoportal.

A princípio planejou-se a estruturação básica do *Site*, no qual definiu-se que conteria uma página principal onde daria acesso ao mapa público e nesta página do mapa público além de apresentar o produto cartográfico gerado em ambiente SIG, existiria um campo que direcionasse para *Site* da prefeitura do município, assim como um acesso restrito para funcionários e contribuintes.

Nesse acesso restrito, a ideia principal é criar um ambiente exclusivo para os funcionários da prefeitura, segundo as segmentações presente no órgão. Como por exemplo, se um funcionário do setor de educação realizasse o login com seus dados de acesso, ele seria encaminhado a uma seção exclusiva do geoportal onde lhe mostraria todas as informações relacionadas a essa secretaria, por meio de representações cartográficas. O mesmo aconteceria para a população contribuinte, onde o proprietário do imóvel teria acesso privado a sua situação tributária assim como a seus dados pessoais. Para o presente trabalho esse procedimento foi

apenas estruturado como exemplo, podendo ser posteriormente alimentados por meio dos dados adquiridos.

Logo, para realizar essa operação de acesso, a página *Online* deverá estar conectada diretamente com a tabela do banco de dados do SGBD onde registra as informações cadastrais tanto dos funcionários quanto dos contribuintes, das quais foram criadas na seção 5.3.1.

Como o acesso ao banco de dados é restrito a seus operadores, geralmente essa estrutura está condicionada a um servidor. Portanto, para efetivar o mecanismo de acesso restrito no geoportal, utilizou-se a linguagem de programação PHP (PHP: Hypertext Preprocessor), na qual consiste em um código que é executado diretamente no servidor, fazendo com que seja possível estabelecer a ligação entre o banco de dados e o *Site*.

Desta forma, toda a estrutura do *Website* foi desenvolvida alterando-se o código fonte gerado a partir da exportação dos produtos cartográficos através no plugin qgis2web, aplicando a linguagem de programação PHP para estabelecer a conexão do PostgreSQL com o servidor que hospedará o *Site*, assim como linguagens de marcação (HTML5, JavaScript e CSS3) para o desenvolvimento, interatividade e modificação do mesmo, onde toda reorganização foi realizada no editor de texto Notepad++.

Com isso, a conexão entre o banco de dados e a página foi realizada por meio do comando `pg_connect`, onde são necessários indicar o local no qual o banco se encontra hospedado, o usuário, a senha, o nome do banco de dados e a porta do programa, caso não se esteja usando a padrão.

Deste modo, o usuário ao entrar com suas credenciais no seu campo exclusivo, a resposta será enviada ao arquivo no qual se estabeleceu a conexão com o banco de dados e salva em variáveis diferentes, onde a partir dessas variáveis realiza-se uma busca, por meio de comandos SQL, nos campos da tabela referente, a fim de verificar se os dados inseridos se encontram cadastrados no banco. Caso essa verificação seja verdadeira o usuário será encaminhando a sua página de destino, caso contrário aparecerá uma mensagem de erro durante a operação.

Para realização dos testes utilizou-se a ferramenta Xampp, que nada mais é que um programa onde facilita a instalação de um pacote de *Softwares* livres e gratuitos (dentre os quais pode-se destacar o servidor apache, o SGBD MySQL e o PHP), no qual transformou a máquina local em um servidor e conseqüentemente em um ambiente para execução de programação em PHP.

Antes da realização dos testes foi necessário configurar o servidor apache, instalado junto ao Xampp, para que se pudesse utilizar o PostgreSQL com sucesso.

Essa configuração é realizada após a instalação do programa, que primeiramente copiasse o arquivo libpq.dll que se encontra na pasta C:\xampp/php para a pasta C:\xampp/apache/bin. Em seguida habilita-se a extensão do PostgreSQL no PHP, editando-se o arquivo php.ini, localizado na pasta C:\xampp\php\php.ini. Com o arquivo aberto no editor de texto procura-se pela linha que contém o código ;extension=php_pgsq.dll e retira-se o ponto e vírgula do mesmo. Feito isso, a ferramenta estará configurada e habilitada para se trabalhar em conjunto com o SGBD desejado.

Todo os códigos trabalhados no desenvolvimento do geoportal, encontram-se descritos e seus comandos devidamente comentados no Apêndice D.

6 RESULTADOS

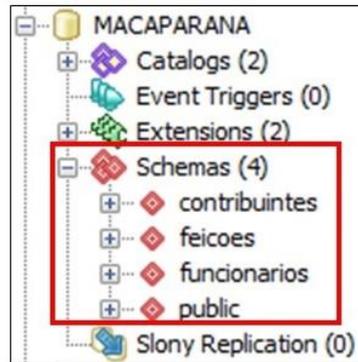
6.1 DADOS DESCRITIVOS

A partir do tratamento dos dados descritivos foi possível estruturar um banco de dados por meio das informações existentes, utilizando o SBGD PostgreSQL.

O programa se mostrou eficiente com respeito a seu manuseio e geração de resultados, apresentando diversos recursos que facilitam a constituição e manipulação de uma base de dados.

Dentre esses recursos pode-se enfatizar a opção de criação de esquemas, que auxiliam na ordenação dos dados segundo sua natureza. Como exemplo disso pode-se verificar na Figura 25, a representação do resultado obtido da criação do banco de dados para o município em estudo, indicando os esquemas que foram levados em consideração de acordo com os dados em mãos e a estruturação final do geoportal.

Figura 25 – Estrutura de um banco de dados criado no *Software PostgreSQL*, enfatizando os schemas que compõe o mesmo.



Fonte: print screen da aplicação no *Software PostgreSQL*.

A Figura 26 retrata as tabelas de cadastramento dos nomes, e-mails e senhas dos contribuintes e funcionários do município, que serão conectadas ao geoportal e servirão para realizar a validação dos dados inseridos na área de acesso restrito, estabelecendo uma interação simultânea entre o serviço *Online* oferecido aos usuários e o banco de dados.

Figura 26 - Tabela de cadastro criada no *Software* PostgreSQL. (a) Contribuintes. (b) Funcionários.

(a)

	id integer	nome character varying(30)	email character varying(30)	senha character varying(10)
1	1	JOSE VITAL LEITAO	jose_vital	jv321
2	2	MANOEL GOMES	manoel_gomes	mg123
3	3	MANOEL PEDRO	manoel_pedro	mp321
4	4	MARIA DA GRAÇAS	maria_graca	ma123
5	6	MARIA DO CARMO	maria_carmo	mc321
6	7	MARIA DO SOCORRO	maria_socorro	ms123
7	8	MARIA DOS PRAZERES	maria_prazeres	mp123
8	9	MARIA FILICIANA	maria_filiciana	mf321
9	11	RENATA T. DA SILVA	renata_t	rt123
10	12	RICARDO DE ARRUDA	ricardo_arruda	ra321
11	14	VALDIR GONCALVES	valdir_goncalves	vg123
12	15	PRICILA ALVES	alvess.pricila@gmail.com	12345

(b)

	id integer	nome character varying(30)	email character varying(30)	senha character varying(10)
1	1	Paula	paula@gmail.com	jv321
2	2	Maria	maria@gmail.com	mg123
3	3	Roberta	roberta@gmail.com	mp321
4	4	Andreia	andreia@gmail.com	ma123
5	6	Paulo	paulo@gmail.com	mc321
6	15	PRICILA ALVES	alvess.pricila@gmail.com	12345

Fonte: print screen da aplicação no *Software* PostgreSQL.

A Figura 27 indica a tabela da quadra utilizada como amostra para realização do estudo, contendo informações sobre o número de inscrição, seção, distrito, setor, o número da quadra, bairro, quantidade de lotes, pedologia e topografia. O retângulo em vermelho indica o campo designado como a chave primária, simbolizado pela sigla [PK].

Figura 30 – Tabela de informações relativas às quadras criada no *Software PostgreSQL*.

	id integer	inscricao character varying	secao character varying	distrito character varying(2)	setor character varying(2)	quadra [PK] character varying(4)	bairro character varying(6)	q_lotes integer	pedologia character varying(15)	topografia character varying(20)
1	1	01.01.001	101001	01	01	001	CENTRO	25	FIRME	PLANO AO NÍVEL
*										

Fonte: print screen da aplicação no *Software PostgreSQL*.

Na Figura 28 pode ser vista a tabela dos lotes contidos na quadra de estudo, indicando o número de inscrição do lote, distrito, setor, quadra, lote, endereço, bairro, CEP e a área do lote. O retângulo vermelho em destaque possui o mesmo significado do comentando na figura anterior. Como as tabelas de quadra e lotes estabelecem um relacionamento direto, a coluna designada como chave primária na tabela de quadra passa a ser uma chave secundária na tabela de lotes (retângulo em amarelo), onde por meio dessa relação é possível obter, por meio de consultas, dados referentes as duas tabelas.

Figura 28 – Tabela de informações relativas aos lotes criada no *Software PostgreSQL*.

	id integer	inscricao character varying(20)	distrito character varying(3)	setor character varying(3)	quadra character varying(4)	lote [PK] character varying(8)	endereco character varying(30)	bairro character varying(8)	cep character varying(15)	area character varying(10)
1	5	01.01.001.0004	01	01	001	0004	AV. 21 DE ABRIL	CENTRO	55065-000	62 m²
2	4	01.01.001.0010	01	01	001	0010	AV. 21 DE ABRIL	CENTRO	55065-000	90 m²
3	14	01.01.001.0020	01	01	001	0020	AV. 21 DE ABRIL	CENTRO	55065-000	150 m²
4	15	01.01.001.0030	01	01	001	0030	AV. 21 DE ABRIL	CENTRO	55065-000	150 m²
5	16	01.01.001.0040	01	01	001	0040	AV. 21 DE ABRIL	CENTRO	55065-000	150 m²
6	17	01.01.001.0049	01	01	001	0049	AV. 21 DE ABRIL	CENTRO	55065-000	150 m²
7	6	01.01.001.0059	01	01	001	0059	AV. 21 DE ABRIL	CENTRO	55065-000	150 m²
8	2	01.01.001.0069	01	01	001	0069	AV. 21 DE ABRIL	CENTRO	55065-000	75 m²
9	9	01.01.001.0075	01	01	001	0075	AV. 21 DE ABRIL	CENTRO	55065-000	103 m²
10	10	01.01.001.0080	01	01	001	0080	AV. 21 DE ABRIL	CENTRO	55065-000	83 m²
11	1	01.01.001.0108	01	01	001	0108	AV. 21 DE ABRIL	CENTRO	55065-000	68 m²
12	13	01.01.001.0142	01	01	001	0142	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55065-000	278 m²
13	3	01.01.001.0152	01	01	001	0152	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55065-000	150 m²
14	8	01.01.001.0157	01	01	001	0157	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55065-000	75 m²
15	7	01.01.001.0162	01	01	001	0162	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55065-000	75 m²
16	12	01.01.001.0210	01	01	001	0210	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55065-000	150 m²
*										

Fonte: print screen da aplicação no *Software PostgreSQL*.

A Figura 29 mostra a tabela das edificações compreendendo a inscrição imobiliária, distrito, setor, quadra, lote, número da casa, endereço, bairro, CEP, área da quadra, área de unidade predial e sua natureza. Além da tabela de quadra, a tabela de lote também estabelece uma condição de relação com a tabela de edificações por meio da coluna “lote” mostrada em destaque de amarelo.

Figura 29 – Tabela de informações relativas as edificações criadas no *Software PostgreSQL*.

	id integer	inscricao character varying(22)	distrito character varying(2)	setor character varying(2)	quadra character varying(3)	lote character varying(5)	n_edificacao character varying(5)	endereço character varying(40)	bairro character varying(7)	cep character varying(10)	area character varying(10)	area_unidade character varying(10)	natureza character varying(15)
1	1	01.01.001.0108.065	01	01	001	0108	0659	AV. 21 DE ABRIL	CENTRO	55065-000	68 m ²	68 m ²	FREDIAL
2	2	01.01.001.0069.071	01	01	001	0069	0717	AV. 21 DE ABRIL	CENTRO	55065-000	75 m ²	41 m ²	FREDIAL
3	3	01.01.001.0152.013	01	01	001	0152	0136	AV. SINHAZINHA DE	CENTRO	55065-000	150 m ²	118 m ²	FREDIAL
4	4	01.01.001.0010.076	01	01	001	0010	0761	AV. 21 DE ABRIL	CENTRO	55065-000	90 m ²	67 m ²	FREDIAL
5	5	01.01.001.0004.076	01	01	001	0004	0769	AV. 21 DE ABRIL	CENTRO	55065-000	62 m ²	50 m ²	FREDIAL
6	6	01.01.001.0059.072	01	01	001	0059	0727	AV. 21 DE ABRIL	CENTRO	55065-000	150 m ²	52 m ²	FREDIAL
7	7	01.01.001.0162.014	01	01	001	0162	0146	AV. SINHAZINHA DE	CENTRO	55065-000	75 m ²	55 m ²	FREDIAL
8	8	01.01.001.0157.014	01	01	001	0157	0140	AV. SINHAZINHA DE	CENTRO	55065-000	75 m ²	65 m ²	FREDIAL
9	9	01.01.001.0075.071	01	01	001	0075	0711	AV. 21 DE ABRIL	CENTRO	55065-000	102 m ²	65 m ²	FREDIAL
10	10	01.01.001.0080.070	01	01	001	0080	0705	AV. 21 DE ABRIL	CENTRO	55065-000	83 m ²	52 m ²	FREDIAL
11	11	01.01.001.0142.012	01	01	001	0142	0126	AV. SINHAZINHA DE	CENTRO	55065-000	75 m ²	64 m ²	FREDIAL
12	12	01.01.001.0210.015	01	01	001	0210	0190	AV. SINHAZINHA DE	CENTRO	55065-000	150 m ²	137 m ²	FREDIAL
13	13	01.01.001.0142.012	01	01	001	0142	0126	AV. SINHAZINHA DE	CENTRO	55065-000	278 m ²	200 m ²	FREDIAL
14	14	01.01.001.0020.075	01	01	001	0020	0755	AV. 21 DE ABRIL	CENTRO	55065-000	150 m ²	0 m ²	FREDIAL
15	15	01.01.001.0030.074	01	01	001	0030	0749	AV. 21 DE ABRIL	CENTRO	55065-000	150 m ²	0 m ²	FREDIAL
16	16	01.01.001.0040.074	01	01	001	0040	0743	AV. 21 DE ABRIL	CENTRO	55065-000	150 m ²	0 m ²	FREDIAL
17	17	01.01.001.0049.073	01	01	001	0049	0737	AV. 21 DE ABRIL	CENTRO	55065-000	150 m ²	0 m ²	FREDIAL

Fonte: print screen da aplicação no *Software PostgreSQL*.

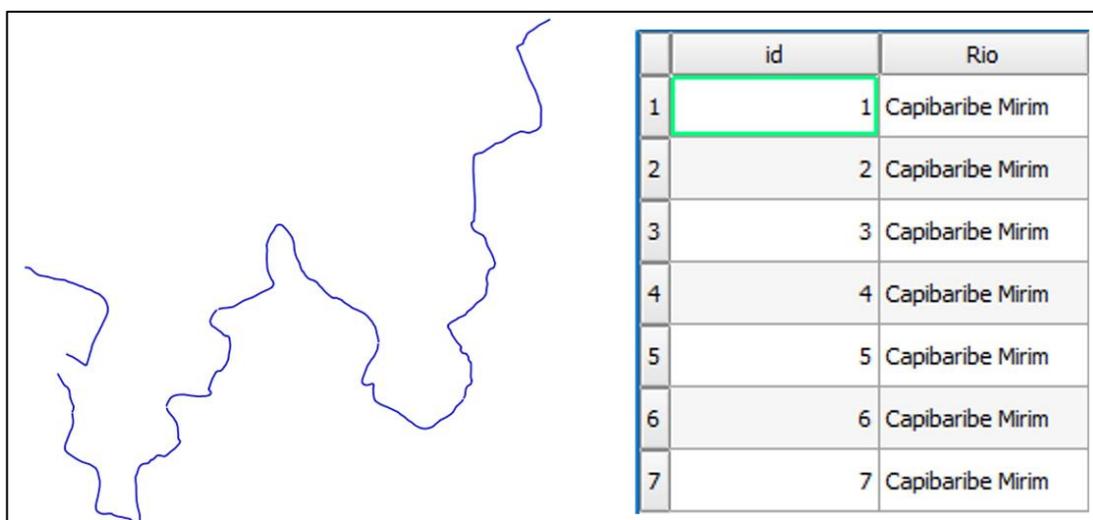
6.2 DADOS GEOMÉTRICOS

Por meio do tratamento dos dados geométricos obteve-se as feições básicas para que a base geométrica do geoportal fosse composta. Não foram criados os elementos cartográficos básicos que constituem um CTM (Planta de Quadra, Planta de Referência Cadastral, Planta de Equipamentos e Planta de Valor Genérico), pois o objetivo do trabalho não é apresentar uma implementação de um cadastro territorial e sim de um geoportal de baixo custo.

A ferramenta empregada, o QGIS, ofereceu os recursos necessários e eficazes para o êxito desta etapa, mostrando-se apto para atividades dessa natureza.

A figura abaixo ilustra o curso d'água da área urbana do município juntamente com sua tabela de atributo, indicando rio que corta a cidade.

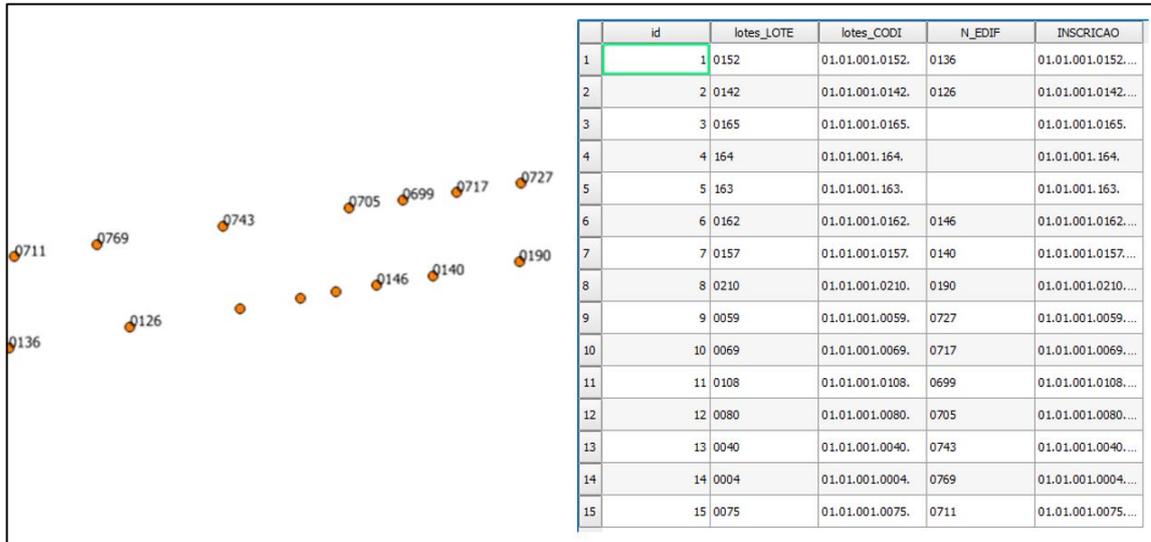
Figura 30 – Representação dos cursos d'águas e sua tabela de atributos criado no *Software* QGIS.



Fonte: Autoria própria (2017).

As edificações da quadra de estudo, que como mencionado no item 5.3.2, foram representadas por feições do tipo ponto como mostrada na figura a seguir. Na tabela da figura, pode-se verificar alguns espaços vazios indicando a não obtenção da numeração predial, consequência de falhas no cadastro territorial do local.

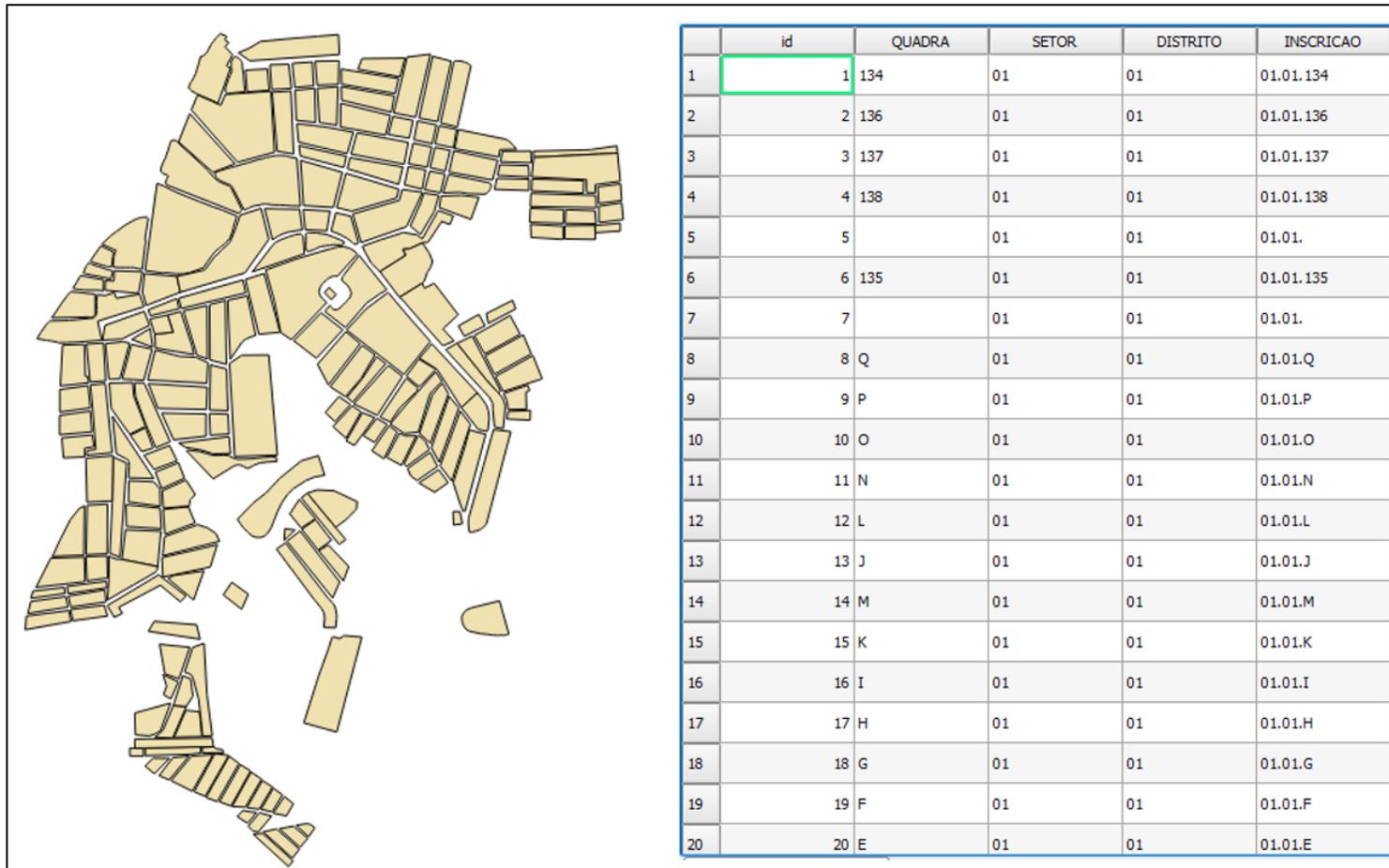
Figura 31 – Representação das edificações e sua tabela de atributos criado no *Software QGIS*.



Fonte: Autoria própria (2017).

As quadras da cidade encontram-se representadas na Figura 32, onde mais uma vez é possível observar na tabela ao lado alguns campos em brancos devido aos mesmos fatores como comentado na figura anterior.

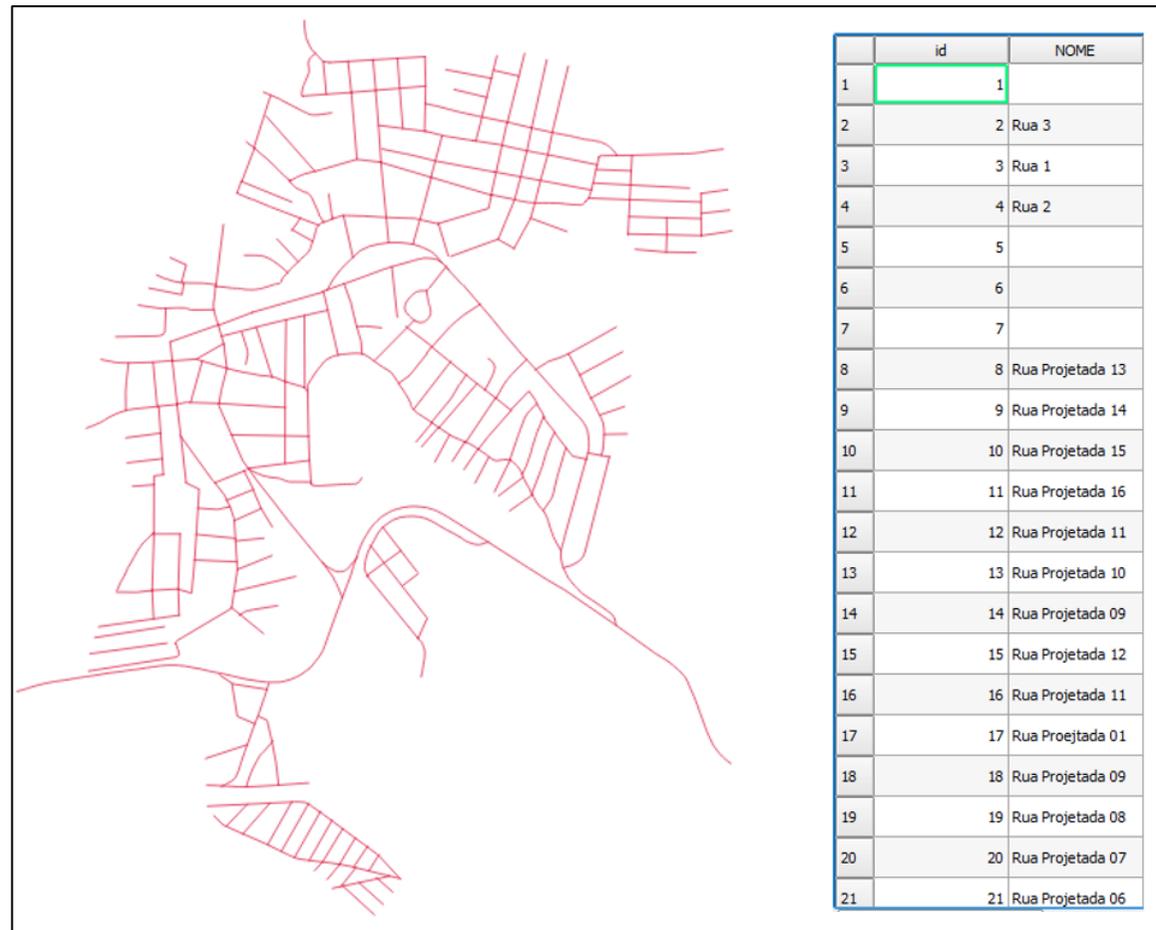
Figura 32 – Representação das quadras e sua tabela de atributos criado no *Software* QGIS.



Fonte: Autoria própria (2017).

A rede viária pode ser vista na Figura 33, assim como sua tabela de atributo, onde novamente é possível verificar a repetição da carência de informações em determinadas áreas como nas feições de quadras e edificações.

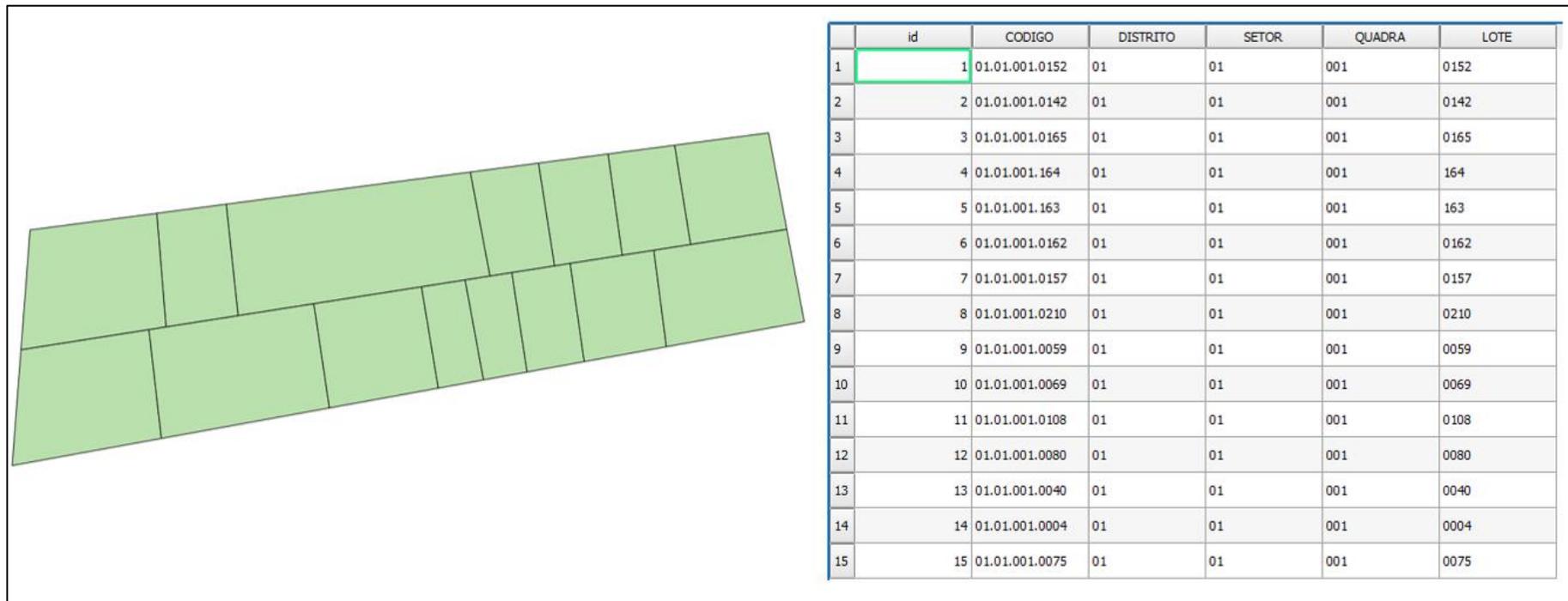
Figura 33 - Representação da rede viária e sua tabela de atributos criado no *Software* QGIS.



Fonte: Autoria própria (2017).

A Figura 34 mostra os lotes da quadra de estudo com sua tabela de informações ao lado. Para essa feição, os dados se mostraram coerentes apresentando cada lote uma identificação correspondente.

Figura 34 - Representação dos lotes e sua tabela de atributos criado no *Software* QGIS.



Fonte: Autoria própria (2017).

6.3 CONEXÃO ENTRE OS DADOS DESCRITIVOS E GEOMÉTRICOS

Os *Softwares* adotados para conexão das informações demonstraram harmonia em sua execução em conjunto, satisfazendo as metas designadas para esta fase.

As figuras a seguir representam o resultado da união entre as tabelas contidas no banco de dados, referentes às quadras, lotes e edificações, com suas respectivas feições no QGIS, onde o retângulo vermelho envolvente indica as informações oriundas do banco de dados, lembrando que após sua junção, as tabelas passaram por um processo de edição, onde algumas colunas foram renomeadas e outras excluídas devido ao fato de sua repetição, pois haviam informações semelhantes tanto nos dados geométricos quanto nos descritivos.

Uma observação que deve ser feita é com respeito aos espaços vazios presentes nos campos, fato que ocorreu devido a não existência da associação entre os dados, ou seja, algumas informações presentes nas geometrias não possuíam um corresponde descritivo.

Na Figura 35 pode ser visualizada a tabela das quadras após a ligação. Como o estudo foi realizado em apenas uma quadra, os demais campos existentes na camada da geometria encontram-se não preenchida.

Figura 35 - Tabela de atributo da feição QUADRAS no *Software* QGIS após a união com os dados da tabela do banco de dados.

QUADRAS :: Feições de totais: 205, filtrado: 205, selecionado: 0

	id	INSCRICAO	SECAO	DISTRITO	SETOR	QUADRA	BAIRRO	Q_LOTES	PEDOLOGIA	TOPOGRAFIA
42	42	01.01.021		01	01	021				
43	43	01.01.022		01	01	022				
44	44	01.01.036		01	01	036				
45	45	01.01.037		01	01	037				
46	46	01.01.		01	01					
47	47	01.01.023		01	01	023				
48	48	01.01.024		01	01	024				
49	49	01.01.019		01	01	019				
50	50	01.01.011		01	01	011				
51	51	01.01.010		01	01	010				
52	52	01.01.009		01	01	009				
53	53	01.01.008		01	01	008				
54	54	01.01.001	101001	01	01	001	CENTRO	25	FIRME	PLANO AO NÍVEL
55	55	01.01.002		01	01	002				
56	56	01.01.003		01	01	003				
57	57	01.01.004		01	01	004				
58	58	01.01.005		01	01	005				
59	59	01.01.006		01	01	006				
60	205	01.01.047		01	01	047				

Fonte: print screen da aplicação no *Software* PostgreSQL.

A Figura 36 mostra o resultado da tabela dos lotes após a conexão.

Figura 36 - Tabela de atributo da feição LOTES no *Software* QGIS após a união com os dados da tabela do banco de dados.

	id	DISTRITO	SETOR	QUADRA	LOTE	INSCRICAO	ENDERECO	BAIRRO	CEP	AREA
1	1 01	01	01	001	0152	01.01.001.0152	AV. SINHAZINHA ...	CENTRO	55865-000	150 m ²
2	2 01	01	01	001	0142	01.01.001.0142	AV. SINHAZINHA...	CENTRO	55865-000	278 m ²
3	3 01	01	01	001	0165	01.01.001.0165				
4	4 01	01	01	001	164	01.01.001.164				
5	5 01	01	01	001	163	01.01.001.163				
6	6 01	01	01	001	0162	01.01.001.0162	AV. SINHAZINHA...	CENTRO	55865-000	75 m ²
7	7 01	01	01	001	0157	01.01.001.0157	AV. SINHAZINHA...	CENTRO	55865-000	75 m ²
8	8 01	01	01	001	0210	01.01.001.0210	AV. SINHAZINHA...	CENTRO	55865-000	150 m ²
9	9 01	01	01	001	0059	01.01.001.0059	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²
10	10 01	01	01	001	0069	01.01.001.0069	AV. 21 DE ABRIL	CENTRO	55865-000	75 m ²
11	11 01	01	01	001	0108	01.01.001.0108	AV. 21 DE ABRIL	CENTRO	55865-000	68 m ²
12	12 01	01	01	001	0080	01.01.001.0080	AV. 21 DE ABRIL	CENTRO	55865-000	83 m ²
13	13 01	01	01	001	0040	01.01.001.0040	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²
14	14 01	01	01	001	0004	01.01.001.0004	AV. 21 DE ABRIL	CENTRO	55865-000	62 m ²
15	15 01	01	01	001	0075	01.01.001.0075	AV. 21 DE ABRIL	CENTRO	55865-000	103 m ²

Fonte: print screen da aplicação no *Software* PostgreSQL.

A Figura 37 mostra o resultado da tabela das edificações após a união.

Figura 37 - Tabela de atributo da feição EDIFICAÇÕES no *Software* QGIS após a união com os dados da tabela do banco de dado.

	id	INSCRICAO	DISTRITO	SETOR	QUADRA	LOTE	EDIFICACAO	ENDEREÇO	BAIRRO	CEP	AREA	A_UNIDADE	NATUREZA
1	1	01.01.001.0152.0136	01	01	001	0152	0136	AV. SINHAZINHA...	CENTRO	55865-000	150 m ²	118 m ²	PREDIAL
2	2	01.01.001.0142.0126	01	01	001	0142	0126	AV. SINHAZINHA...	CENTRO	55865-000	278 m ²	200 m ²	PREDIAL
3	3	01.01.001.0165.				0165							
4	4	01.01.001.164.				164							
5	5	01.01.001.163.				163							
6	6	01.01.001.0162.0146	01	01	001	0162	0146	AV. SINHAZINHA...	CENTRO	55865-000	75 m ²	55 m ²	PREDIAL
7	7	01.01.001.0157.0140	01	01	001	0157	0140	AV. SINHAZINHA...	CENTRO	55865-000	75 m ²	65 m ²	PREDIAL
8	8	01.01.001.0210.0190	01	01	001	0210	0190	AV. SINHAZINHA...	CENTRO	55865-000	150 m ²	137 m ²	PREDIAL
9	9	01.01.001.0059.0727	01	01	001	0059	0727	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²	52 m ²	PREDIAL
10	10	01.01.001.0069.0717	01	01	001	0069	0717	AV. 21 DE ABRIL	CENTRO	55865-000	75 m ²	41 m ²	PREDIAL
11	11	01.01.001.0108.0699	01	01	001	0108	0699	AV. 21 DE ABRIL	CENTRO	55865-000	68 m ²	68 m ²	PREDIAL
12	12	01.01.001.0080.0705	01	01	001	0080	0705	AV. 21 DE ABRIL	CENTRO	55865-000	83 m ²	52 m ²	PREDIAL
13	13	01.01.001.0040.0743	01	01	001	0040	0743	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²	0 m ²	PREDIAL
14	14	01.01.001.0004.0769	01	01	001	0004	0769	AV. 21 DE ABRIL	CENTRO	55865-000	62 m ²	50 m ²	PREDIAL
15	15	01.01.001.0075.0711	01	01	001	0075	0711	AV. 21 DE ABRIL	CENTRO	55865-000	103 m ²	65 m ²	PREDIAL

Fonte: print screen da aplicação no *Software* PostgreSQL.

Após a ligação de todas as feições com suas referentes tabelas, essas foram enviadas ao SGBD, onde durante o processo foi possível analisar a capacidade do PostgreSQL ao trabalhar com dados geoespaciais, devido sua extensão espacial PostGIS, no qual garantiu uma fácil exportação e armazenamento dos mesmos.

Nas figuras adiante, as colunas em destaque de vermelho correspondem a representação geométrica dos dados.

A Figura 40 retrata a tabela no banco de dados dos lotes.

Figura 40 – Tabela de informações descritivas e geométricas relativas aos Lotes gerados a partir do envio da feição ao *Software* PostgreSQL.

	id [PK] bigint	geom geometry(MultiPolygon,31985)	codigo character varying(20)	distrito character varying(2)	setor character varying(2)	quadra character varying(4)	lote character varying(10)	endereco character varying(254)	bairro character varying(6)	cep character varying(254)	area character varying(254)
1	1	0106000020F17C000001000	01.01.001.0152	01	01	001	0152	AV. SINHAZINHA DE	CENTRO	55065-000	150 m²
2	2	0106000020F17C000001000	01.01.001.0142	01	01	001	0142	AV. SINHAZINHA DE	CENTRO	55065-000	278 m²
3	3	0106000020F17C000001000	01.01.001.0165				0165				
4	4	0106000020F17C000001000	01.01.001.164				164				
5	5	0106000020F17C000001000	01.01.001.163				163				
6	6	0106000020F17C000001000	01.01.001.0162	01	01	001	0162	AV. SINHAZINHA DE	CENTRO	55065-000	75 m²
7	7	0106000020F17C000001000	01.01.001.0157	01	01	001	0157	AV. SINHAZINHA DE	CENTRO	55065-000	75 m²
8	8	0106000020F17C000001000	01.01.001.0210	01	01	001	0210	AV. SINHAZINHA DE	CENTRO	55065-000	150 m²
9	9	0106000020F17C000001000	01.01.001.0059	01	01	001	0059	AV. 21 DE ABRIL	CENTRO	55065-000	150 m²
10	10	0106000020F17C000001000	01.01.001.0069	01	01	001	0069	AV. 21 DE ABRIL	CENTRO	55065-000	75 m²
11	11	0106000020F17C000001000	01.01.001.0108	01	01	001	0108	AV. 21 DE ABRIL	CENTRO	55065-000	60 m²
12	12	0106000020F17C000001000	01.01.001.0080	01	01	001	0080	AV. 21 DE ABRIL	CENTRO	55065-000	83 m²
13	13	0106000020F17C000001000	01.01.001.0040	01	01	001	0040	AV. 21 DE ABRIL	CENTRO	55065-000	150 m²
14	14	0106000020F17C000001000	01.01.001.0004	01	01	001	0004	AV. 21 DE ABRIL	CENTRO	55065-000	62 m²
15	15	0106000020F17C000001000	01.01.001.0075	01	01	001	0075	AV. 21 DE ABRIL	CENTRO	55065-000	103 m²

Fonte: print screen da aplicação no *Software* PostgreSQL.

Na Figura 41 pode ser visto parte da tabela no banco de dados das quadras.

Figura 41 – Tabela de informações descritivas e geométricas relativas às Quadras gerada a partir do envio da feição ao *Software* PostgreSQL.

	id [PK] bigint	geom geometry(MultiPolygon,31985)	inscricao character varying(10)	secao character varying(254)	distrito character varying(3)	setor character varying(3)	quadra character varying(6)	bairro character varying(6)	q_lotes character varying(10)	pedologia character varying(15)	topografia character varying(20)
40	40	0106000020F17C000001000	01.01.017		01	01	017				
41	41	0106000020F17C000001000	01.01.020		01	01	020				
42	42	0106000020F17C000001000	01.01.021		01	01	021				
43	43	0106000020F17C000001000	01.01.022		01	01	022				
44	44	0106000020F17C000001000	01.01.036		01	01	036				
45	45	0106000020F17C000001000	01.01.037		01	01	037				
46	46	0106000020F17C000001000	01.01.		01	01					
47	47	0106000020F17C000001000	01.01.023		01	01	023				
48	48	0106000020F17C000001000	01.01.024		01	01	024				
49	49	0106000020F17C000001000	01.01.019		01	01	019				
50	50	0106000020F17C000001000	01.01.011		01	01	011				
51	51	0106000020F17C000001000	01.01.010		01	01	010				
52	52	0106000020F17C000001000	01.01.009		01	01	009				
53	53	0106000020F17C000001000	01.01.008		01	01	008				
54	54	0106000020F17C000001000	01.01.001	101001	01	01	001	CENTRO	25	FIRME	PLANO AO NÍVEL
55	55	0106000020F17C000001000	01.01.002		01	01	002				
56	56	0106000020F17C000001000	01.01.003		01	01	003				
57	57	0106000020F17C000001000	01.01.004		01	01	004				
58	--	--	--	--	--	--	--				

Fonte: print screen da aplicação no *Software* PostgreSQL.

Na Figura 42 mostra parte da tabela do banco de dados do sistema viário.

Figura 42 – Tabela de informações descritivas e geométricas relativas às Quadras gerada a partir do envio da feição ao *Software* PostgreSQL.

	id [PK] bigint	geom geometry(MultiLineString,31985)	nome character varying(50)
1	1	0105000020F17C0000010000000102000000200000007533B3465D150C41E2DD6214757A6141DA9B4F4E25150C418FEF3AD3747A6141	
2	2	0105000020F17C000001000000010200000020000000CB8760DF650C0C41EF77C3DE6A7A6141310FD8D626100C418F66B294587A6141	Rua 3
3	3	0105000020F17C000001000000010200000030000000DC07C9D2860D0C412AD4A88A797A614119EDB5F428100C41E266B2ED6B7A6141	Rua 1
4	4	0105000020F17C0000010000000102000000300000006FC0A30A00C0C4185035B02747A61412423E967D4100C419755A1995E7A6141	Rua 2
5	5	0105000020F17C0000010000000102000000B000000310FD8D626100C418F66B294587A6141AE30C46159100C41F49DA79A577A6141	
6	6	0105000020F17C000001000000010200000020000000664EF597D90E0C41A19A5EB3727A614176646A15520C0C416A88E95F6B7A6141	
7	7	0105000020F17C0000010000000102000000D00000018C593E2C50E0C410D101D1A117A61412C3D1C2BE50A0C414E4350DF147A6141	
8	8	0105000020F17C000001000000010200000020000000CB9F9302C0090C419BA936460E7A61416EA55F53FD0D0C416D1B1DDD117A6141	Rua Projetada 13
9	9	0105000020F17C000001000000010200000020000002501D886D50C0C419BCB5834067A614196AAFBA740A0C41175D6FDF0E7A6141	Rua Projetada 14
10	10	0105000020F17C00000100000001020000002000000030370BFA800D0C412143D0B5097A6141A2093E4ABF0B0C4125D718F70F7A6141	Rua Projetada 15
11	11	0105000020F17C000001000000010200000020000000EF06D8C6320E0C41627603FC0C7A61412311ED200C0D0C413B78E610117A6141	Rua Projetada 16
12	12	0105000020F17C0000010000000102000000200000003E475106F0B0C413A6A3D59147A614118B9A5C9500A0C4149B7CAC00E7A6141	Rua Projetada 11
13	13	0105000020F17C0000010000000102000000200000003308E9974B090C41838714E3107A6141D16D87CA3A0A0C41FC94A151157A6141	Rua Projetada 10
14	14	0105000020F17C000001000000010200000020000000D310FA287C080C41D3CB5899137A6141CAD26B0708090C417B1A33FF157A6141	Rua Projetada 09
15	15	0105000020F17C00000100000001020000002000000068F1C6F525D0C415D43D018187A61411B61F7106C0C0C41BF324063137A6141	Rua Projetada 12
16	16	0105000020F17C00000100000001020000002000000086A982F14B0C0C41DFBA479C1A7A6141B34CE4A5450B0C4189188281147A6141	Rua Projetada 11
17	17	0105000020F17C00000100000001020000002000000230CE957670B0C415021AE301D7A61413A0DE4AD380A0C4130550C53157A6141	Rua Proejtada 01
18	18	0105000020F17C000001000000010200000020000000548AE9DF840A0C41E5EF2FCE1F7A61412D33FEB206090C41F2D7CDF157A6141	Rua Projetada 09
19	19	0105000020F17C0000010000000102000000200000004C807DD2A7090C41E09FAE5B227A61419DA8AB20F6070C41B7808A84167A6141	Rua Projetada 08
20	20	0105000020F17C000001000000010200000020000000AF135A53E5070C4193E0324D277A6141A7D7FC93E0050C4187446337197A6141	Rua Projetada 07
21	21	0105000020F17C000001000000010200000020000000AF135A53E5070C4193E0324D277A6141A7D7FC93E0050C4187446337197A6141	Rua Projetada 06
22	22	0105000020F17C00000100000001020000002000000042C8549EA3060C41F85445CF277A6141FBD158F7DA040C4179976A311B7A6141	Rua Projeta 05
23	23	0105000020F17C0000010000000102000000D00000009E3C0B6AC60E0C41B4BA671F117A614168F1C6F525D0C415D43D018187A6141	Rua Projetada 01
24	24	0105000020F17C0000010000000102000000200000005DB638C6FD040C412FE5B879277A6141A48C4FA6FE020C41A84F17771E7A6141	Rua Projetada 04

Fonte: print screen da aplicação no *Software* PostgreSQL.

6.4 GEOPORTAL

As linguagens computacionais aplicadas para o desenvolvimento do geoportal ofereceram diversos recursos que favoreceram para sua estruturação e interação.

A Figura 43 retrata a página principal do *Site*, organizada de maneira amigável e com o acesso direto ao mapa público no link “Acesso ao Geoportal”.

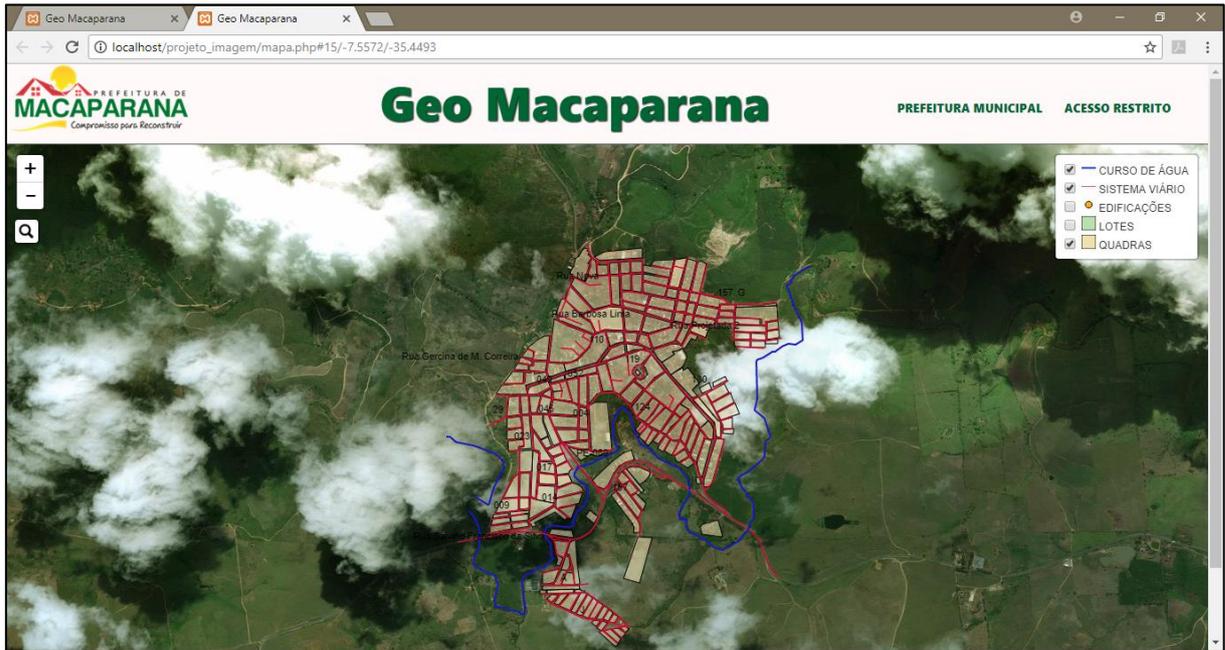
Figura 43 – Interface principal do geoportal do município de Macaparana.



Fonte: Autoria própria (2017).

Na área do mapa público (Figura 44) pode ser visto a representação geométrica das principais feições da cidade, podendo as mesmas serem ativadas ou desativadas por meio de uma caixa localizada na região próxima ao canto superior direito, assim como ferramenta de busca e zoom perto ao canto superior esquerdo. A página também apresenta um *link* direto para o Site da Prefeitura e acesso restrito a funcionários e contribuintes.

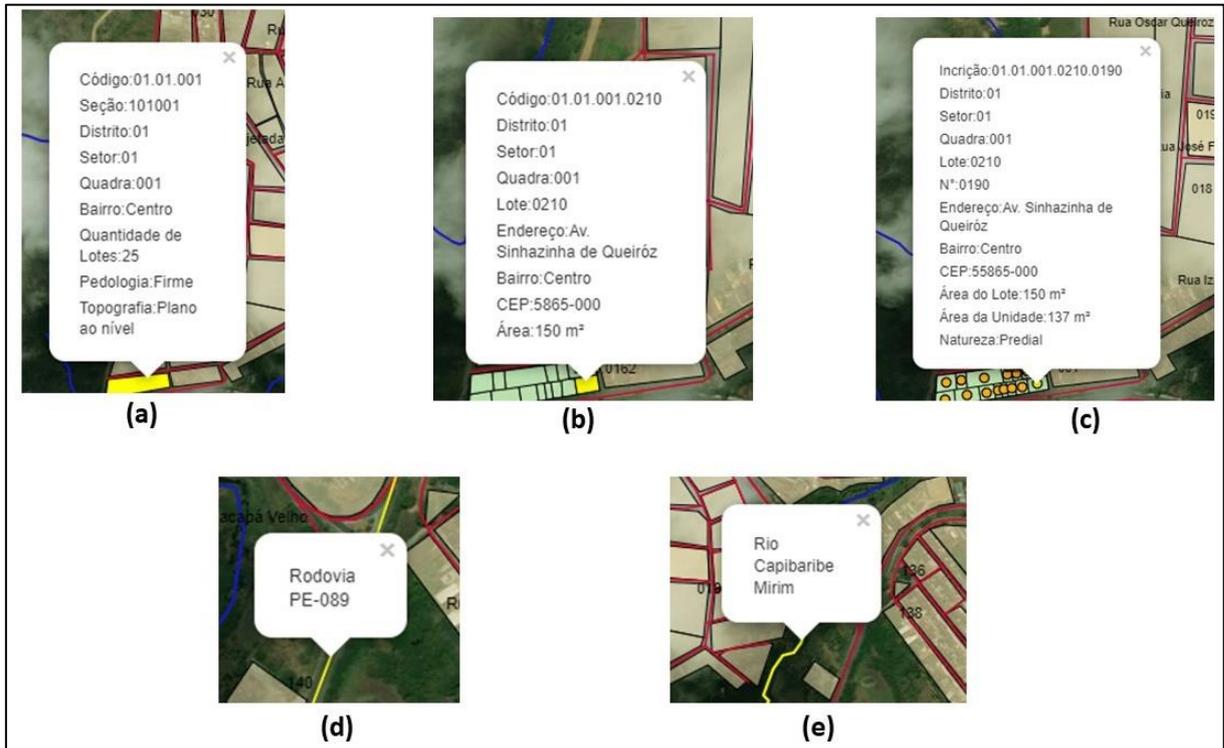
Figura 44 – Mapa público geoportal do município de Macaparana.



Fonte: Autoria própria (2017).

Ainda na página do mapa público, pode-se verificar as informações relativas as respectivas camadas ao clicar-se em cada feição, propriedade adquirida a conexão entre os dados descritivos e geométricos. A Figura 45 mostra os dados referentes a cada geometria.

Figura 45 – Informações ligadas as feições. (a) Informações sobre a quadra. (b) Informação sobre o lote. (c) Informação sobre a edificação. (c) Informação sobre a via. (d) Informação sobre o curso d'água.



Fonte: Autoria própria (2017).

A Figura 46 mostra o Site da prefeitura do qual o usuário é direcionado ao clicar na opção “PREFEITURA MUNICIPAL” na página do mapa público, onde poderá obter notícias atualizadas relativas ao município.

Figura 46 – Site da cidade de Macaparana.



Fonte: Autoria própria (2017).

A Figura 47 exibe o campo destinado ao acesso restrito para funcionários e contribuintes.

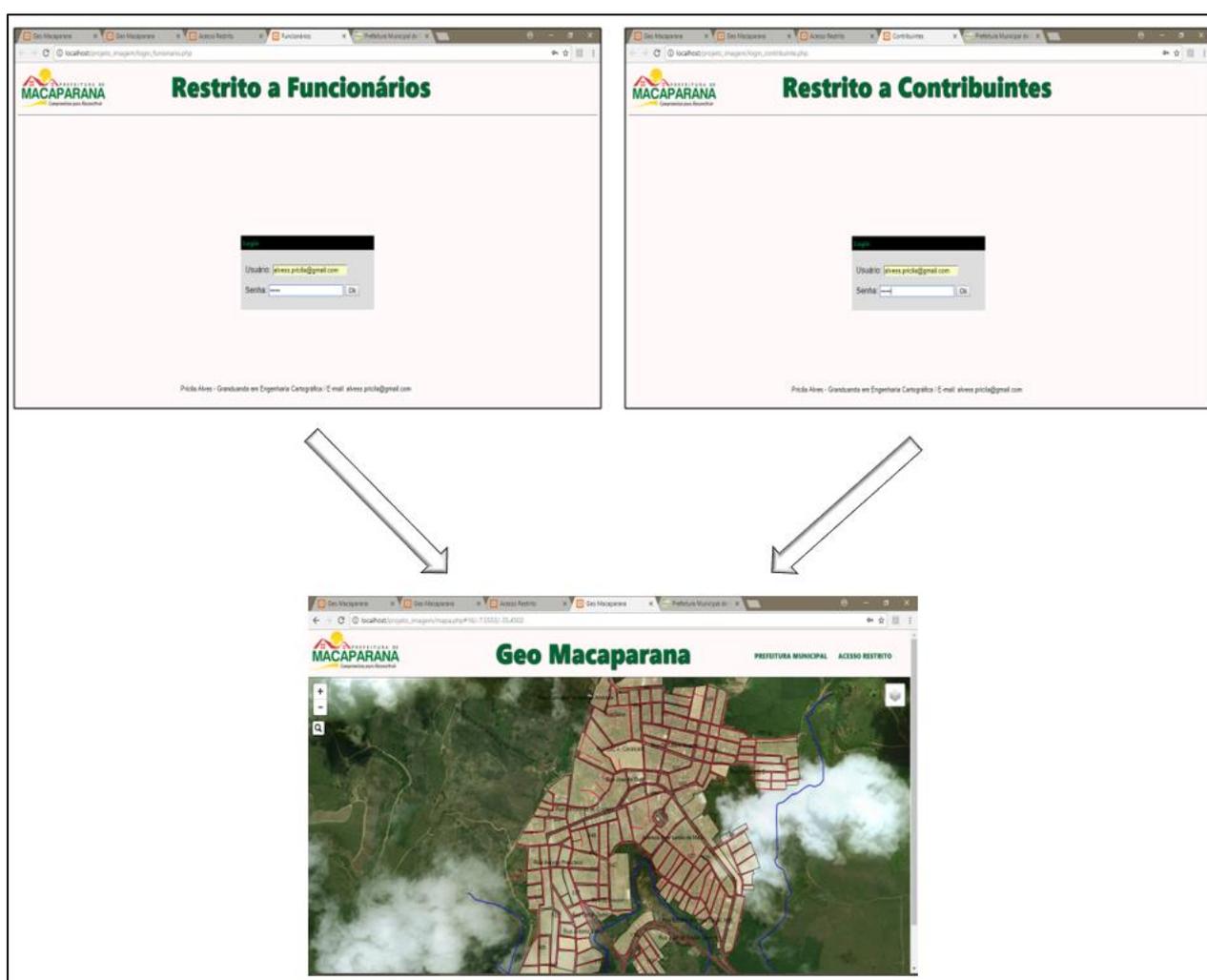
Figura 47 – Área principal de acesso restrito aos funcionários e contribuintes.



Fonte: Autoria própria (2017).

Ao escolher uma das possibilidades oferecidas pelo *Website*, o operador será encaminhado a uma janela onde entrará com suas credenciais. Caso essas informações estejam cadastradas no banco de dados geral, o funcionário ou contribuinte será conduzido a um local de mapa restrito com a temática e descrições alusivas a cada um, caso contrário o acesso de entrada será negado e bloqueado. Para o presente trabalho esse mapa restrito não foi gerado, apenas estabeleceu-se uma ligação com a página de mapa público para servir como exemplo e sugestões futuras. Essas áreas de acesso podem ser visualizadas na Figura 48.

Figura 48 – Área de acesso restrito aos contribuintes.



Fonte: Autoria própria (2017).

7 DIFICULDADES

Durante as etapas das atividades alguns obstáculos surgiram tanto no tempo de excursão quanto na coleta de dados e processamento.

O tempo de desenvolvimento das atividades foi de três meses prazo considerado pequeno para quantidade de produtos gerados, porém graças aos conhecimentos prévios da autora como manipular os *Softwares* e as linguagens computacionais utilizados foi possível cumprir com o que vinha sendo proposto na pesquisa.

Na coleta dos dados, apesar de terem sido obtidos vários produtos geométricos direto com a prefeitura do município, os mesmos se encontravam desatualizados e fora dos padrões cartográficos. Após várias manipulações viu-se que seu uso seria improvável e comprometeria o desenvolvimento do trabalho devido as inconsistências presentes fazendo com que fosse gasto mais tempo realizando a vetorização dos elementos básicos e impossibilitando a agregação dos mapas temáticos da cidade ao geoportal.

Um outro problema é com relação aos dados descritivos, onde não foi possível ter acesso às informações de toda a cidade, mas apenas de uma amostra, devido ao fato de não ter se estabelecido um contato direto com os responsáveis pelo servidor que contém o banco de dados geral do município e sim pelo setor de finanças que possui acesso as informações por quadras, fazendo com que a disponibilização das mesmas como um todo seja uma tarefa onerosa.

Com relação ao processamento dos dados, o plugin *qgis2web* apresentou alguns contratempos no momento do envio dos produtos cartográficos à *Internet*. Por diversas vezes a ferramenta apresentou falhas no carregamento das camadas e a exportação não foi obtida com sucesso. Porém, o inconveniente não alterou o curso do projeto e o problema foi sanado automaticamente pelo próprio *Software* chegou-se à conclusão de que poderia ter sido algum tipo de falha no código do plugin.

O último empecilho ocorrido foi com respeito a conexão entre o banco de dados e a página na *Internet*, pois para realizar esse procedimento são necessários conhecimentos prévios sobre linguagem de programação PHP, até então desconhecida para autora. Para suprir essa necessidade buscou-se auxílio com um estudante da área de tecnologia da computação, assim como buscas pela Web sobre o tema.

8 CONCLUSÕES

O uso de geotecnologias como ferramenta de auxílio para gestão dos municípios brasileiros de pequeno porte, atualmente ainda é um desafio, devido aos altos custos financeiros que acarreta esse tipo de tecnologia.

Em meio esse contexto, a realização do presente estudo possibilitou uma análise sobre a aplicação de recursos tecnológicos de baixo custo para criação do geoportal do município de

Macaparana, utilizando dados descritivos e geométricos relativos ao local, dos quais foram processados em programas especializados e manipulados por meio de linguagens computacionais. Além disso, também foi permitido explorar a funcionalidade de cada ferramenta, assim com sua eficiência para o desenvolvimento desse tipo de trabalho.

De modo geral, foi possível estruturar, a partir de uma amostra, uma base de dados espaciais, feições geométricas referentes a realidade atual da cidade, integração de informações de naturezas diferentes e elaboração de um geoportal.

Não foi possível analisar a qualidade geométrica dos dados pois tratou-se apenas de uma representação da temática atual do município para criação de um sistema protótipo para *Web*, assim como não foi utilizado nenhuma norma cartográfica para elaboração dos mesmos, logo o geoportal criado não poderá ser usufruído sem passar por uma validação de seus dados para que se tenha uma segurança técnica e jurídica.

Na fase de coleta de dados verificou-se a dificuldade de ser obter informações cadastrais atualizadas e consistentes, apresentando a verdadeira realidade que a maioria dos municípios de pequeno porte enfrentam.

A escolha dos recursos tecnológicos partiu da premissa que deveriam ser de caráter livre e não comercial, já que a proposta é minimizar os gastos do serviço. A seleção das linguagens computacionais para construção da página *Online* baseou-se nas funções que a mesma deveria oferecer aos usuários.

O SGBD PostgreSQL mostrou bastante competência, tanto para o armazenamento de dados descritivos quanto os geométricos, não apresentando nenhum tipo de falha nas operações que impossibilitasse o cumprimento do que foi proposto para a ferramenta.

O *Software* QGIS também se mostrou eficiente para o tratamento dos dados geométricos, no qual garantiu a vetorização das feições de maneira simples e eficaz, assim como a ligação entre as tabelas do banco de dados e a exportação das camadas para o banco, apresentando problemas, que foram contornados, pelo plugin qgis2web no ato de exportação dos produtos finais.

Com respeito ao uso das ferramentas PostgreSQL e QGIS de maneira conjunta, percebeu-se o quanto foi harmonioso o trabalho executado por elas, efetivando com sucesso todos os procedimentos solicitados.

Na construção do geoportal, as linguagens computacionais empregadas simultaneamente se mostraram capazes para atender toda formação do mesmo, garantindo uma perfeita conexão com o banco de dados e uma interface amigável aos usuários. Os programas

Xampp e Notepad++ também foram essenciais para o desenvolvimento da etapa final, que assim como os demais *Softwares* operados nesta pesquisa apresentaram bons funcionamentos.

Sendo assim, a proposta retratada nesta pesquisa alcançou seus objetivos finais, podendo ser aplicada em uma situação real que servirá de grande assistência aos municípios que apresentam recursos financeiros limitados.

9 RECOMENDAÇÕES

A metodologia aplicada neste trabalho se mostrou eficiente, apesar da limitação dos dados brutos utilizados para o desenvolvimento da mesma.

Algumas medidas futuras poderão ser tomadas visando o aprimoramento do geoportal. Com respeito ao cadastro territorial do município, onde, com o desenrolar do projeto pôde-se confirmar que o mesmo deve passar por um processo de reestruturação, pois a base cartográfica existente não acompanhou as mudanças ocasionadas ao longo dos anos.

Uma outra medida que poderá ser tomada é o aperfeiçoamento das funções do geoportal com adição de novas aplicações, tais como: ferramentas de medição, escala, opção de impressão e etc. Poderá também, por meio de outras ferramentas das quais não foram empregadas neste estudo, exportar o mosaico da ortofoto do município para que se tenha mais uma opção de mapa base no *Website*.

As informações contidas nas camadas de feições podem ser conectadas com uma tabela no banco de dados, para que seja feita atualização automática no *Site* quando ocorrer algum tipo de alteração no banco.

No acesso restrito aos funcionarios podem ser criados novos campos para os demais segmentos da prefeitura segundo a temática de cada setor.

Por fim, a última recomendação é hospedar o geoportal, com autorização da prefeitura do município, em um servidor no qual expanda o produto a todos os usuários da *Internet*.

REFERÊNCIAS

BILENE, A. R. R.; SAIS, A. C.; GONÇALVES, R. H. Banco de dados geográfico para cidades de pequeno porte: estudo de caso da Cidade de Andradas, MG. In: Simpósio Brasileiro de Sensoriamento Remoto, 13., 2007, Florianópolis. *Anais...* São Paulo: Instituto Nacional de Pesquisas Espaciais, 2007. p. 5121-5121.

BORGES, K. A. V. **Curso de especialização em geoprocessamento**. UFMG, 2002. Disponível em: <<http://www.csr.ufmg.br/geoprocessamento/publicacoes/ALEXANDRE%20DE%20CARVALHO.PDF>>. Acesso em: 10 de outubro de 2017.

BOSSLE, R. C. QGIS do ABC ao XYZ. In: _____. **Coordenadas e escalas**. 1. ed. Paraná: Íthala, 2016. cap. 2, p. 5.

BRASIL. Lei n° 10.267, de 28 de agosto de 2001. Disponível em: <http://www.planalto.gov.br/ccivil_03/Leis/LEIS_2001/L10267.htm>. Acesso em: 19 de setembro de 2017.

CARNEIRO, A. F. T. Cadastro imobiliário e registro de imóveis. In: _____. **Cadastro e sistemas cadastrais**. 1. ed. Porto Alegre: IRIB. 2003. cap. 1, p. 22.

CARNEIRO, A. F. T. *Cadastro brasileiro: uma visão crítica*. 2010. 49f. Monografia (Master en Catastro Internacional Multiproposito Avanzado0 - Universidad de Jaén, Recife. 2010.

CARNEIRO, A. F. T.; ERBA, D. A.; AUGUSTO, E. A. A. Cadastro multifinalitário 3D: conceito e perspectivas de implantação no Brasil. **Revista Brasileira de Cartografia**, Rio de Janeiro, n. 64/2, p. 257-271, abr. 2012.

CHALEGRE, A. M. T. *Avaliação da Integração entre Sistemas de Informações Geográficas e Sistemas Gerenciadores de Bancos de Dados com Extensões Espaciais*. 2011. 74f. Dissertação (Mestrado em Cartografia e Sistemas de Geoinformação) – Departamento de Engenharia Cartográfica, Universidade Federal de Pernambuco, Recife. 2011.

FARINELLI, F. Conceito básicos de programação orientada a objetos. In:_____. **Fundamentos da programação orientada a objetos**. 2007. p. 4. Disponível em: <<https://sites.google.com/site/wagnerkc/POO.pdf>>. Acesso em: 10 de novembro de 2017.

FERREIRA, N. C. Apostila de sistema de informações geográficas. In:_____. **Introdução**. GOIÂNIA. 2006. p. 7. Disponível em: <http://www.faed.udesc.br/arquivos/id_submenu/1414/apostila_sig.pdf>. Acesso em: 3 de novembro de 2017.

GONÇALVES, R. P. *Modelagem Conceitual de Bancos de Dados Geográficos para Cadastro Técnico Multifinalitário em Municípios de Pequeno e Médio Porte*. 2008. 170 f. Dissertação (Mestrado em Engenharia Civil) – Departamento de Engenharia Civil, Universidade Federal de Viçosa, Minas Gerais. 2008.

KROENKE, D. M. Banco de Dados: fundamentos, projeto e implementação. In: _____. **Introdução ao processamento de banco de dados**. 6. ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 1998. cap. 1, p. 12.

LEMES, I. R. et al. Implementação de um geoportal GEOWEB para auxiliar na gestão territorial. In: Simpósio Brasileiro de Geomática e Jornadas Lusófonas – Ciências e Tecnologia de Informação Geográfica, 4., 2., 2017, Presidente Prudente. *Anais...* Presidente Prudente: unesp, 2017. p. 104-109.

MEDEIROS, A. M. L. Artigos sobre conceitos em geoprocessamento. In: _____. **O geoprocessamento e suas tecnologias**. João Pessoa. 2012. Disponível em: <<http://www.andersonmedeiros.com/wp-content/uploads/2012/09/E-book-Artigos-sobre-Conceitos-em-Geoprocessamento-Anderson-Medeiros.pdf>>. Acesso em: 10 de outubro de 2017.

GOVERNO DE PERNAMBUCO. Pernambuco tridimensional. Disponível em: <<http://www.pe3d.pe.gov.br/>>. Acesso em: 13 de dezembro de 2017.

MEDEIRO, M. Banco de dados para sistemas de informação. In: _____. **Conceitos gerais de bancos de dados**. Florianópolis: Visual Books Ltda, 2006. cap. 1, p.5.

MELO, R. N.; SILVA, S. D.; TANAKA, A. K. Banco de dados em aplicações cliente-servidor. In: _____. **Introdução**. Rio de Janeiro: Infobook, 1998. cap. 1, p. 5-13.

MINISTÉRIO DAS CIDADES. Portaria n° 51, de 7 de dezembro de 2009. Disponível em: <https://www.cidades.gov.br/images/stories/ArquivosSNPU/Biblioteca/PlanelamentoUrbano/Portaria_511_PROEXT.pdf>. Acesso em: 25 de outubro de 2017.

PESSOA, G. G.; OLIVEIRA, R. F.; DESTRO, R. D. C.; JULIÃO, R. P.; AMORIM, A. Construção e utilização de um geoportail na administração pública municipal. In: COBRAC, 2016, Florianópolis. *Anais...* Florianópolis: Universidade Federal de Santa Catarina, 2017.

PREFEITURA DE MACAPARANA. Plano diretor do município de Macaparana.

PROGRAMA NACIONAL DE CAPACITAÇÃO DAS CIDADES. Diretrizes para o Cadastro Territorial Multifinalitário In: CARNEIRO, A. F. T. **Disposições gerais do cadastro territorial multifinalitário**. 2016. p. 3.

SANTOS, R. C.; MENDES, L. A. M. **Estudo comparativo dos sistemas gerenciadores de bancos de Dados: Oracle, SQL Server e PostgreSQL**. Disponível em: <<http://www.unipac.br/site/bb/tcc/tcc-6cda526f17394bae9bbbed711b2188920.pdf>>. Acesso em: 25 de setembro de 2017.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de banco de dados. In: _____. **Introdução**. 6. ed. Rio de Janeiro: CAMPUS, 2012. cap. 1, p. 1-19.

APÊNDICES

Apêndice A – Dados descritivos organizados em um editor de planilha em forma de tabelas.

QUADRAS									
ID	INSCRICAO	SECAO	DISTRITO	SETOR	QUADRA	BAIRRO	Q.LOTES	PEDOLOGIA	TOPOGRAFIA
1	01.01.001	101001	01	01	001	CENTRO	25	FIRME	PLANO AO NÍVEL

LOTES									
ID	INSCRICAO	DISTRITO	SETOR	QUADRA	LOTE	ENDERECO	BAIRRO	CEP	AREA
1	01.01.001.0108	01	01	001	0108	AV. 21 DE ABRIL	CENTRO	55865-000	68 m ²
2	01.01.001.0069	01	01	001	0069	AV. 21 DE ABRIL	CENTRO	55865-000	75 m ²
3	01.01.001.0152	01	01	001	0152	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	150 m ²
4	01.01.001.0010	01	01	001	0010	AV. 21 DE ABRIL	CENTRO	55865-000	90 m ²
5	01.01.001.0004	01	01	001	0004	AV. 21 DE ABRIL	CENTRO	55865-000	62 m ²
6	01.01.001.0059	01	01	001	0059	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²
7	01.01.001.0162	01	01	001	0162	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	75 m ²
8	01.01.001.0157	01	01	001	0157	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	75 m ²
9	01.01.001.0075	01	01	001	0075	AV. 21 DE ABRIL	CENTRO	55865-000	103 m ²
10	01.01.001.0080	01	01	001	0080	AV. 21 DE ABRIL	CENTRO	55865-000	83 m ²
11	01.01.001.0142	01	01	001	0142	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	75 m ²
12	01.01.001.0210	01	01	001	0210	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	150 m ²
13	01.01.001.0142	01	01	001	0142	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	278 m ²
14	01.01.001.0020	01	01	001	0020	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²
15	01.01.001.0030	01	01	001	0030	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²
16	01.01.001.0040	01	01	001	0040	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²
17	01.01.001.0049	01	01	001	0049	AV. 21 DE ABRIL	CENTRO	55865-000	150 m ²

EDIFICAÇÕES												
ID	INSCRICAO	DISTRITO	SETOR	QUADRA	LOTE	N_EDIFICACAO	ENDERECO	BAIRRO	CEP	AREA	AREA_UNIDADE	NATUREZA
1	01.01.001.0108.0699	01	01	001	0108	0699	AV. 21 DE ABRIL	CENTRO	55865-000	68 m²	68 m²	PREDIAL
2	01.01.001.0069.0717	01	01	001	0069	0717	AV. 21 DE ABRIL	CENTRO	55865-000	75 m²	41 m²	PREDIAL
3	01.01.001.0152.0136	01	01	001	0152	0136	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	150 m²	118 m²	PREDIAL
4	01.01.001.0010.0761	01	01	001	0010	0761	AV. 21 DE ABRIL	CENTRO	55865-000	90 m²	67 m²	PREDIAL
5	01.01.001.0004.0769	01	01	001	0004	0769	AV. 21 DE ABRIL	CENTRO	55865-000	62 m²	50 m²	PREDIAL
6	01.01.001.0059.0727	01	01	001	0059	0727	AV. 21 DE ABRIL	CENTRO	55865-000	150 m²	52 m²	PREDIAL
7	01.01.001.0162.0146	01	01	001	0162	0146	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	75 m²	55 m²	PREDIAL
8	01.01.001.0157.0140	01	01	001	0157	0140	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	75 m²	65 m²	PREDIAL
9	01.01.001.0075.0711	01	01	001	0075	0711	AV. 21 DE ABRIL	CENTRO	55865-000	103 m²	65 m²	PREDIAL
10	01.01.001.0080.0705	01	01	001	0080	0705	AV. 21 DE ABRIL	CENTRO	55865-000	83 m²	52 m²	PREDIAL
11	01.01.001.0142.0126	01	01	001	0142	0126	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	75 m²	64 m²	PREDIAL
12	01.01.001.0210.0190	01	01	001	0210	0190	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	150 m²	137 m²	PREDIAL
13	01.01.001.0142.0126	01	01	001	0142	0126	AV. SINHAZINHA DE QUEIRÓZ	CENTRO	55865-000	278 m²	200 m²	PREDIAL
14	01.01.001.0020.0755	01	01	001	0020	0755	AV. 21 DE ABRIL	CENTRO	55865-000	150 m²	0 m²	PREDIAL
15	01.01.001.0030.0749	01	01	001	0030	0749	AV. 21 DE ABRIL	CENTRO	55865-000	150 m²	0 m²	PREDIAL
16	01.01.001.0040.0743	01	01	001	0040	0743	AV. 21 DE ABRIL	CENTRO	55865-000	150 m²	0 m²	PREDIAL
17	01.01.001.0049.0737	01	01	001	0049	0737	AV. 21 DE ABRIL	CENTRO	55865-000	150 m²	0 m²	PREDIAL

CADASTRO DOS CONTRIBUINTES			
ID	NOME	EMAIL	SENHA
1	JOSE VITAL LEITAO	jose_vital	jv321
2	MANOEL GOMES	manoel_gomes	mg123
3	MANOEL PEDRO	manoel_pedro	mp321
4	MARIA DA GRAÇAS	maria_graca	ma123
6	MARIA DO CARMO	maria_carmo	mc321
7	MARIA DO SOCORRO	maria_socorro	ms123
8	MARIA DOS PRAZERES	maria_prazeres	mp123
9	MARIA FILICIANA	maria_filiciana	mf321
11	RENATA T. DA SILVA	renata_t	rt123
12	RICARDO DE ARRUDA	ricardo_arruda	ra321
14	VALDIR GONCALVES	valdir_goncalves	vg123
15	PRICILA ALVES	alvess.pricila@gmail.com	12345

CADASTRO DOS FUNCIONÁRIOS			
ID	NOME	EMAIL	SENHA
1	Paula	paula@gmail.com	jv321
2	Maria	maria@gmail.com	mg123
3	Roberta	roberta@gmail.com	mp321
4	Andreia	andreia@gmail.com	ma123
6	Paulo	paulo@gmail.com	mc321
15	PRICILA ALVES	alvess.pricila@gmail.com	12345

Apêndice B – Comandos para criação das tabelas no SGBD PostgreSQL.

- Tabela de QUADRAS

```
CREATE TABLE feicoes. quadras  
(ID integer,  
INSCRICAO varchar (20),  
SECAO varchar (5),  
DISTRITO varchar (3),  
SETOR varchar (3),  
QUADRA varchar (4) primary key,  
BAIRRO varchar (6),  
Q_LOTES integer,  
PEDOLOGIA varchar (15),  
TOPOGRAFIA varchar (20)  
);
```

- Tabela de LOTES

```
CREATE TABLE feicoes. lotes  
(ID integer,  
INSCRICAO varchar (20),  
DISTRITO varchar (3),  
SETOR varchar (3) ,  
QUADRA varchar (4),  
LOTE varchar (8),  
ENDERECO varchar (20),  
BAIRRO varchar (8),  
CEP varchar (15),  
AREA varchar (10),  
CONSTRAINT fk_QuaLotes FOREIGN KEY (quadra) REFERENCES feicoes.quadras (quadra)
```

);

- Tabela de EDIFICAÇÕES

```
CREATE TABLE feicoes. edificacoes
```

```
(ID integer,
```

```
INSCRICAO varchar (22),
```

```
DISTRITO varchar (3),
```

```
SETOR varchar (3),
```

```
QUADRA varchar (3),
```

```
LOTE varchar (5),
```

```
N_EDIFICACAO varchar (5),
```

```
ENDERECO varchar (40),
```

```
BAIRRO varchar (7),
```

```
CEP varchar (10),
```

```
AREA varchar (10),
```

```
AREA_UNIDADE varchar (10),
```

```
NATUREZA varchar (15),
```

```
CONSTRAINT fk_LotEdf FOREIGN KEY (lote) REFERENCES feicoes.lotes (lote)
```

);

- Tabela de CADASTRO DOS CONTRIBUINTES

```
CREATE TABLE contribuinte. cadastro
```

```
(id integer,
```

```
nome varchar (30),
```

```
usuario varchar (30),
```

```
senha varchar (10) );
```

- Tabela de CADASTRO DOS FUNCINÁRIOS

```
CREATE TABLE funcionarios. cadastro
```

```
(id integer,
```

```
nome varchar (30),  
usuario varchar (30),  
senha varchar (10) );
```

Apêndice C - Comandos para inserção de dados em tabelas no SGBD PostgreSQL.

- Tabela de QUADRAS

```
COPY feicoes. quadras
(ID,
INSCRICAO,
SECAO,
DISTRITO,
SETOR,
QUADRA,
BAIRRO,
Q_LOTES,
PEDOLOGIA,
TOPOGRAFIA)
from 'C:\TCC\banco_de_dados\tabelas/quadras.csv'
delimiter ';'
csv header;
```

- Tabela de LOTES

```
COPY feicoes. lotes
(ID,
INSCRICAO,
DISTRITO,
SETOR,
QUADRA,
```

```
LOTE,  
ENDERECO,  
BAIRRO,  
CEP,  
AREA)  
from 'C:\TCC\banco_de_dados\tabelas/lotos.csv'  
delimiter ';' csv header;
```

- Tabela de EDIFICAÇÕES

```
COPY feicoes.edificacoes  
(ID,  
INSCRICAO,  
DISTRITO,  
SETOR,  
QUADRA,  
LOTE,  
N_EDIFICACAO,  
ENDERECO,  
BAIRRO,  
CEP,  
AREA,  
AREA_UNIDADE,  
NATUREZA)  
from 'C:\TCC\banco_de_dados\tabelas/edificacoes.csv'  
delimiter ';' csv header;
```

- Tabela de CADASTRO DOS CONTRIBUINTES

COPY contribuinte. cadastro

(ID,

NOME,

USUARIO,

SENHA)

from 'C:\TCC\banco_de_dados\tabelas\contribuintes.csv'

delimiter ';' csv header;

- Tabela de CADASTRO DOS FUNCINÁRIOS

COPY funcionarios. cadastro

(ID,

NOME,

USUARIO,

SENHA)

from 'C:\TCC\banco_de_dados\tabelas\funcionarios.csv'

delimiter ';' csv header;

Apêndice D – Comandos para criação do Geoportall

Página Principal (arquivo index1.php)

```

<!doctype html>
<html lang = "pt-br" > <! -- lang="pt-br": Idioma do site -->
  <head> <! -- head: área onde ocorre as configurações comportamentais do site -->
    <meta charset="utf-8"> <! -- meta: tag que representa vários tipos de metadados / charset: atributo que indicar o formato de codificação
de caracteres -->
    <link rel="stylesheet" href="css/estilo.css"/> <! -- chama a função estilo.css -->
    <title>Geo Macaparana</title> <! -- título da página -->
  </head>
  <body> <!-- body: área onde deve ser inserido todas as informações visíveis do site -->
    <header id="texto"> <!-- header: tag que delimita a área de cabeçalho -->
      <hgroup> <!-- hgroup: tag semântica que agrupa títulos-->
        <h1>Bem-Vindo! </h1>
        <a href= "mapa.php" target="_blank"><h2>Acesso ao Geoportall<h2></a></br> <!-- a: tag que define uma hiperligação -->
      </hgroup>
    </header>
    <div id="interface"> <!-- div: tag que gera uma delimitação da página -->
      <figure class="principal"> <!--figure: tag que especifica que a imagem faz parte de uma artigo-->
        <img src= "imagens/principal1.png" > <!-- img: tag de inserção de figura -->
      </figure>

```

```

</div>
<footer id="rodape"> <!-- footer: tag cria um rodapé na página -->
<p>Pricila Alves - Graduanda em Engenharia Cartográfica/E-mail:alvess.pricila@gmail.com
</footer>
</body>
</html>

```

Mapa público (arquivo mapa.php)

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1,user-scalable=no,maximum-scale=1,width=device-width">
<meta name="mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-capable" content="yes">
<link rel="stylesheet" href="http://unpkg.com/leaflet@1.0.3/dist/leaflet.css">
<link rel="stylesheet" href="css/qgis2web.css">
<link rel="stylesheet" href="css/leaflet-search.css">
<link rel="stylesheet" href="css/estilo.css"/>

```

```
<title>Geo Macaparana</title>
<style>
html, body, #map {
  width: 100%;
  height: 100%;
  padding: 0;
  margin: 0;
}
</style>
</head>
<body>
<header id="cabecalho">
  <nav id="menu">

    <ul type="square">
      <b><li><a href="http://macaparana.pe.gov.br/portal/" target="_blank">Prefeitura Municipal</a></li></b>
      <b><li><a href="login_principal.php" target="_blank">Acesso restrito</a></li></b>
    </ul>

  </nav>
  <nav id="titulo">
    <ul type="square">
      <li><a>Geo Macaparana</a>
```

```
</ul>
</header>

<div id="map">
</div>
<script src="js/qgis2web_expressions.js"></script>
<script src="http://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>
<script src="js/leaflet.rotatedMarker.js"></script>
<script src="js/leaflet.pattern.js"></script>
<script src="js/leaflet-hash.js"></script>
<script src="js/Autolinker.min.js"></script>
<script src="js/rbush.min.js"></script>
<script src="js/labelgun.min.js"></script>
<script src="js/labels.js"></script>
<script src="js/proj4.js"></script>
<script src="js/proj4leaflet.js"></script>
<script src="js/leaflet-search.js"></script>
<script src="data/QUADRAS_0.js"></script>
<script src="data/LOTES_1.js"></script>
<script src="data/EDIFICAES_2.js"></script>
<script src="data/SISTEMAVIRIO_3.js"></script>
<script src="data/cursodeagua_4.js"></script>
```

```
<script>
var highlightLayer;
function highlightFeature(e) {
    highlightLayer = e.target;

    if (e.target.feature.geometry.type === 'LineString') {
        highlightLayer.setStyle({
            color: '#ffff00',
        });
    } else {
        highlightLayer.setStyle({
            fillColor: '#ffff00',
            fillOpacity: 1
        });
    }
}
var map = L.map('map', {
    zoomControl:true, maxZoom:28, minZoom:1
}).fitBounds([[[-7.56195214373,-35.4553852019],[-7.548616,-35.444928]]]);
var hash = new L.Hash(map);
map.attributionControl.addAttribution('<a href="https://github.com/tomchadwin/qgis2web" target="_blank">qgis2web</a>');
var bounds_group = new L.featureGroup([]);
```

```

var basemap0 = L.tileLayer('http://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}', {
  attribution: 'Mapas &copy; OpenCycleMap, Dados do Mapa &copy; contribuidores do OpenStreetMap',
  maxZoom: 28
});
basemap0.addTo(map);
function setBounds() {
}
function pop_QUADRAS_0(feature, layer) {
  layer.on({
    mouseout: function(e) {
      for (i in e.target._eventParents) {
        e.target._eventParents[i].resetStyle(e.target);
      }
    },
    mouseover: highlightFeature,
  });
  var popupContent = '<table>\
    <tr>\
      <td colspan="2">' + (feature.properties['id'] !== null ? Autolinker.link(String(feature.properties['id'])) : '') + '</td>\
    </tr>\
    <tr>\

```

```

                <td colspan="2"> + (feature.properties['INSCRICAO'] != null ?
Autolinker.link(String(feature.properties['INSCRICAO'])) : ") + </td>\
            </tr>\
            <tr>\
                <td colspan="2">' + (feature.properties['SECAO'] != null ? Autolinker.link(String(feature.properties['SECAO'])) : ") +
'</td>\
            </tr>\
            <tr>\
                <td colspan="2">' + (feature.properties['DISTRITO'] != null ? Autolinker.link(String(feature.properties['DISTRITO'])) : ") +
+ '</td>\
            </tr>\
            <tr>\
                <td colspan="2">' + (feature.properties['SETOR'] != null ? Autolinker.link(String(feature.properties['SETOR'])) : ") +
'</td>\
            </tr>\
            <tr>\
                <td colspan="2">' + (feature.properties['QUADRA'] != null ? Autolinker.link(String(feature.properties['QUADRA'])) : ") +
+ '</td>\
            </tr>\
            <tr>\
                <td colspan="2">' + (feature.properties['BAIRRO'] != null ? Autolinker.link(String(feature.properties['BAIRRO'])) : ") +
'</td>\

```

```

    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['Q_LOTES'] !== null ? Autolinker.link(String(feature.properties['Q_LOTES'])) : '')
+ '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['PEDOLOGIA'] !== null ?
Autolinker.link(String(feature.properties['PEDOLOGIA'])) : '') + '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['TOPOGRAFIA'] !== null ?
Autolinker.link(String(feature.properties['TOPOGRAFIA'])) : '') + '</td>\
    </tr>\
</table>';
layer.bindPopup(popupContent, { maxHeight: 400 });
}

```

```

function style_QUADRAS_0_0() {
    return {
        pane: 'pane_QUADRAS_0',
        opacity: 1,
        color: 'rgba(0,0,0,1.0)',
    };
}

```

```
        dashArray: "",
        lineCap: 'butt',
        lineJoin: 'miter',
        weight: 1.0,
        fillOpacity: 0.5,
        fillColor: 'rgba(240,225,176,1.0)',
    }
}
map.createPane('pane_QUADRAS_0');
map.getPane('pane_QUADRAS_0').style.zIndex = 400;
map.getPane('pane_QUADRAS_0').style['mix-blend-mode'] = 'normal';
var layer_QUADRAS_0 = new L.geoJson(json_QUADRAS_0, {
    attribution: '<a href=""></a>',
    pane: 'pane_QUADRAS_0',
    onEachFeature: pop_QUADRAS_0,
    style: style_QUADRAS_0_0,
});
bounds_group.addLayer(layer_QUADRAS_0);
map.addLayer(layer_QUADRAS_0);
function pop_LOTES_1(feature, layer) {
    layer.on({
        mouseout: function(e) {
```

```

    for (i in e.target._eventParents) {
        e.target._eventParents[i].resetStyle(e.target);
    }
},
mouseover: highlightFeature,
});
var popupContent = '<table>\
    <tr>\
        <td colspan="2">' + (feature.properties['id'] !== null ? Autolinker.link(String(feature.properties['id'])) : ") + '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['CODIGO'] !== null ? Autolinker.link(String(feature.properties['CODIGO'])) : ") +
'</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['DISTRITO'] !== null ? Autolinker.link(String(feature.properties['DISTRITO'])) : ") +
+ '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['SETOR'] !== null ? Autolinker.link(String(feature.properties['SETOR'])) : ") +
'</td>\
    </tr>\

```

```

<tr>\
  <td colspan="2">' + (feature.properties['QUADRA'] != null ? Autolinker.link(String(feature.properties['QUADRA'])) : ")
+ '</td>\
</tr>\
<tr>\
  <td colspan="2">' + (feature.properties['LOTE'] != null ? Autolinker.link(String(feature.properties['LOTE'])) : ") + '</td>\
</tr>\
<tr>\
  <td colspan="2">' + (feature.properties['ENDERECO'] != null ? Autolinker.link(String(feature.properties['ENDERECO']))
: ") + '</td>\
</tr>\
<tr>\
  <td colspan="2">' + (feature.properties['BAIRRO'] != null ? Autolinker.link(String(feature.properties['BAIRRO'])) : ") +
'</td>\
</tr>\
<tr>\
  <td colspan="2">' + (feature.properties['CEP'] != null ? Autolinker.link(String(feature.properties['CEP'])) : ") + '</td>\
</tr>\
<tr>\
  <td colspan="2">' + (feature.properties['AREA'] != null ? Autolinker.link(String(feature.properties['AREA'])) : ") + '</td>\
</tr>\
</table>';

```

```
layer.bindPopup(popupContent, {maxHeight: 400});
}

function style_LOTES_1_0() {
  return {
    pane: 'pane_LOTES_1',
    opacity: 1,
    color: 'rgba(0,0,0,1.0)',
    dashArray: "",
    lineCap: 'butt',
    lineJoin: 'miter',
    weight: 1.0,
    fillOpacity: 1,
    fillColor: 'rgba(186,224,174,1.0)',
  }
}

map.createPane('pane_LOTES_1');
map.getPane('pane_LOTES_1').style.zIndex = 401;
map.getPane('pane_LOTES_1').style['mix-blend-mode'] = 'normal';
var layer_LOTES_1 = new L.geoJson(json_LOTES_1, {
  attribution: '<a href=""></a>',
  pane: 'pane_LOTES_1',
```

```

    onEachFeature: pop_LOTES_1,
    style: style_LOTES_1_0,
  });
  bounds_group.addLayer(layer_LOTES_1);
  function pop_EDIFICAES_2(feature, layer) {
    layer.on({
      mouseout: function(e) {
        for (i in e.target._eventParents) {
          e.target._eventParents[i].resetStyle(e.target);
        }
      },
      mouseover: highlightFeature,
    });
    var popupContent = '<table>\
      <tr>\
        <td colspan="2">' + (feature.properties['id'] !== null ? Autolinker.link(String(feature.properties['id'])) : '') + '</td>\
      </tr>\
      <tr>\
        <td colspan="2">' + (feature.properties['INSCRICAO'] !== null ?
Autolinker.link(String(feature.properties['INSCRICAO'])) : '') + '</td>\
      </tr>\
    </table>';
  }

```

```

        <td colspan="2">' + (feature.properties['DISTRITO'] != null ? Autolinker.link(String(feature.properties['DISTRITO'])) : ")
+ '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['SETOR'] != null ? Autolinker.link(String(feature.properties['SETOR'])) : ") +
'</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['QUADRA'] != null ? Autolinker.link(String(feature.properties['QUADRA'])) : ")
+ '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['LOTE'] != null ? Autolinker.link(String(feature.properties['LOTE'])) : ") + '</td>\
    </tr>\
    <tr>\
        <td
            colspan="2">' + (feature.properties['EDIFICACAO'] != null ?
Autolinker.link(String(feature.properties['EDIFICACAO'])) : ") + '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['ENDERECO'] != null ? Autolinker.link(String(feature.properties['ENDERECO']))
: ") + '</td>\
    </tr>\

```

```

        <tr>\
            <td colspan="2">' + (feature.properties['BAIRRO'] != null ? Autolinker.link(String(feature.properties['BAIRRO'])) : ") +
'</td>\
        </tr>\
        <tr>\
            <td colspan="2">' + (feature.properties['CEP'] != null ? Autolinker.link(String(feature.properties['CEP'])) : ") + '</td>\
        </tr>\
        <tr>\
            <td colspan="2">' + (feature.properties['AREA'] != null ? Autolinker.link(String(feature.properties['AREA'])) : ") + '</td>\
        </tr>\
        <tr>\
            <td
                colspan="2">'
                +
                (feature.properties['A_UNIDADE']
                    !=
                    null
                    ?
Autolinker.link(String(feature.properties['A_UNIDADE'])) : ") + '</td>\
        </tr>\
        <tr>\
            <td
                colspan="2">'
                +
                (feature.properties['NATUREZA']
                    !=
                    null
                    ?
Autolinker.link(String(feature.properties['NATUREZA'])) : ") + '</td>\
        </tr>\
    </table>;
    layer.bindPopup(popupContent, { maxHeight: 400});
}

```

```
function style_EDIFICAES_2_0() {
  return {
    pane: 'pane_EDIFICAES_2',
    radius: 4.0,
    opacity: 1,
    color: 'rgba(0,0,0,1.0)',
    dashArray: "",
    lineCap: 'butt',
    lineJoin: 'miter',
    weight: 1,
    fillOpacity: 1,
    fillColor: 'rgba(255,172,12,1.0)',
  }
}

map.createPane('pane_EDIFICAES_2');
map.getPane('pane_EDIFICAES_2').style.zIndex = 402;
map.getPane('pane_EDIFICAES_2').style['mix-blend-mode'] = 'normal';
var layer_EDIFICAES_2 = new L.geoJson(json_EDIFICAES_2, {
  attribution: '<a href=""></a>',
  pane: 'pane_EDIFICAES_2',
  onEachFeature: pop_EDIFICAES_2,
  pointToLayer: function (feature, latlng) {
```

```

var context = {
    feature: feature,
    variables: {}
};
return L.circleMarker(latlng, style_EDIFICAES_2_0(feature));
},
});
bounds_group.addLayer(layer_EDIFICAES_2);
function pop_SISTEMAVIRIO_3(feature, layer) {
    layer.on({
        mouseout: function(e) {
            for (i in e.target._eventParents) {
                e.target._eventParents[i].resetStyle(e.target);
            }
        },
        mouseover: highlightFeature,
    });
    var popupContent = '<table>\
        <tr>\
            <td colspan="2">' + (feature.properties['id'] !== null ? Autolinker.link(String(feature.properties['id'])) : '') + '</td>\
        </tr>\
        <tr>\

```

```

        <td colspan="2">' + (feature.properties['NOME'] !== null ? Autolinker.link(String(feature.properties['NOME'])) : '') +
'</td>\
    </tr>\
</table>';
layer.bindPopup(popupContent, { maxHeight: 400});
}

function style_SISTEMAVIRIO_3_0() {
    return {
        pane: 'pane_SISTEMAVIRIO_3',
        opacity: 1,
        color: 'rgba(214,5,47,1.0)',
        dashArray: "",
        lineCap: 'square',
        lineJoin: 'bevel',
        weight: 1.5,
        fillOpacity: 0,
    }
}

map.createPane('pane_SISTEMAVIRIO_3');
map.getPane('pane_SISTEMAVIRIO_3').style.zIndex = 403;
map.getPane('pane_SISTEMAVIRIO_3').style['mix-blend-mode'] = 'normal';

```

```

var layer_SISTEMAVIRIO_3 = new L.geoJson(json_SISTEMAVIRIO_3, {
  attribution: '<a href=""></a>',
  pane: 'pane_SISTEMAVIRIO_3',
  onEachFeature: pop_SISTEMAVIRIO_3,
  style: style_SISTEMAVIRIO_3_0,
});
bounds_group.addLayer(layer_SISTEMAVIRIO_3);
map.addLayer(layer_SISTEMAVIRIO_3);
function pop_cursodeagua_4(feature, layer) {
  layer.on({
    mouseout: function(e) {
      for (i in e.target._eventParents) {
        e.target._eventParents[i].resetStyle(e.target);
      }
    },
    mouseover: highlightFeature,
  });
  var popupContent = '<table>\
    <tr>\
      <td colspan="2">' + (feature.properties['id'] !== null ? Autolinker.link(String(feature.properties['id'])) : '') + '</td>\
    </tr>\
    <tr>\

```

```

        <td colspan="2">' + (feature.properties['Rio'] !== null ? Autolinker.link(String(feature.properties['Rio'])) : '') + '</td>\
    </tr>\
</table>';
layer.bindPopup(popupContent, { maxHeight: 400});
}

function style_cursodeagua_4_0() {
    return {
        pane: 'pane_cursodeagua_4',
        opacity: 1,
        color: 'rgba(24,24,197,1.0)',
        dashArray: "",
        lineCap: 'square',
        lineJoin: 'bevel',
        weight: 2.0,
        fillOpacity: 0,
    }
}

map.createPane('pane_cursodeagua_4');
map.getPane('pane_cursodeagua_4').style.zIndex = 404;
map.getPane('pane_cursodeagua_4').style['mix-blend-mode'] = 'normal';
var layer_cursodeagua_4 = new L.geoJson(json_cursodeagua_4, {

```

```

    attribution: '<a href=""></a>',
    pane: 'pane_cursodeagua_4',
    onEachFeature: pop_cursodeagua_4,
    style: style_cursodeagua_4_0,
  });
  bounds_group.addLayer(layer_cursodeagua_4);
  map.addLayer(layer_cursodeagua_4);
  var baseMaps = {};
  L.control.layers(baseMaps,{' CURSO DE ÁGUA': layer_cursodeagua_4,' SISTEMA VIÁRIO': layer_SISTEMAVIRIO_3,'
EDIFICAÇÕES': layer_EDIFICAES_2,' LOTES': layer_LOTES_1,'
QUADRAS': layer_QUADRAS_0,}).addTo(map);
  setBounds();
  var i = 0;
  layer_QUADRAS_0.eachLayer(function(layer) {
    var context = {
      feature: layer.feature,
      variables: {}
    };
    layer.bindTooltip((layer.feature.properties['QUADRA'] !== null?String('<div style="color: #000000; font-size: 8pt; font-family:
\AcadEref', sans-serif;">' + layer.feature.properties['QUADRA']) + '</div>:'), {permanent: true, offset: [-0, -16], className:
'css_QUADRAS_0'});

```

```

    labels.push(layer);
    totalMarkers += 1;
    layer.added = true;
    addLabel(layer, i);
    i++;
  });
  var i = 0;
  layer_LOTES_1.eachLayer(function(layer) {
    var context = {
      feature: layer.feature,
      variables: {}
    };
    layer.bindTooltip((layer.feature.properties['LOTE'] !== null?String('<div style="color: #000000; font-size: 8pt; font-family: \MS
Shell Dlg 2\', sans-serif;">' + layer.feature.properties['LOTE'] + '</div>:'), {permanent: true, offset: [-0, -16], className: 'css_LOTES_1'});
    labels.push(layer);
    totalMarkers += 1;
    layer.added = true;
    addLabel(layer, i);
    i++;
  });
  var i = 0;
  layer_EDIFICAES_2.eachLayer(function(layer) {

```

```

var context = {
  feature: layer.feature,
  variables: {}
};
layer.bindTooltip((layer.feature.properties['EDIFICACAO'] !== null?String('<div style="color: #000000; font-size: 8pt; font-family: \MS Shell Dlg 2\, sans-serif;">' + layer.feature.properties['EDIFICACAO']) + '</div>:'), {permanent: true, offset: [-0, -16], className: 'css_EDIFICAES_2'});
labels.push(layer);
totalMarkers += 1;
layer.added = true;
addLabel(layer, i);
i++;
});
var i = 0;
layer_SISTEMAVIRIO_3.eachLayer(function(layer) {
  var context = {
    feature: layer.feature,
    variables: {}
  };
  layer.bindTooltip((layer.feature.properties['NOME'] !== null?String('<div style="color: #000000; font-size: 8pt; font-family: \MS Shell Dlg 2\, sans-serif;">' + layer.feature.properties['NOME']) + '</div>:'), {permanent: true, offset: [-0, -16], className: 'css_SISTEMAVIRIO_3'});

```

```
labels.push(layer);
totalMarkers += 1;
layer.added = true;
addLabel(layer, i);
i++;
});
map.addControl(new L.Control.Search({
  layer: layer_SISTEMAVIRIO_3,
  initial: false,
  hideMarkerOnCollapse: true,
  propertyName: 'NOME'}));
resetLabels([layer_QUADRAS_0,layer_LOTES_1,layer_EDIFICAES_2,layer_SISTEMAVIRIO_3]);
map.on("zoomend", function(){
  resetLabels([layer_QUADRAS_0,layer_LOTES_1,layer_EDIFICAES_2,layer_SISTEMAVIRIO_3]);
});
map.on("layeradd", function(){
  resetLabels([layer_QUADRAS_0,layer_LOTES_1,layer_EDIFICAES_2,layer_SISTEMAVIRIO_3]);
});
map.on("layerremove", function(){
  resetLabels([layer_QUADRAS_0,layer_LOTES_1,layer_EDIFICAES_2,layer_SISTEMAVIRIO_3]);
});
</script>
```

```
</body>
```

```
</html>
```

- Área de acesso restrito principal (arquivo login_principal.php)

```
<!doctype html>
```

```
<html lang="pt-br"> <!-- lang="pt-br": Idioma do site -->
```

```
<head> <!-- head: área onde ocorre as configurações comportamentais do site -->
```

```
<title>Acesso Restrito</title> <!-- título da página -->
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<link rel="stylesheet" href="css/estilo.css"/> <!-- chama a função estilo.css -->
```

```
<link rel="stylesheet" href="css/estilo_login_principal.css"/> <!-- chama a função estilo_login_principal.css -->
```

```
</head>
```

```
<body>
```

```
<header id="cabecalho"> <!-- header: tag que delimita a área de cabeçalho -->
```

```
<nav id="acesso"> <!--nav: tag que indica uma seção de navegação-->
```

```
<ul type="square"> <!--ul: tag que cria lista de itens / li: cria itens das listas-->
```

```
    <b><li><a href="login_funionario.php" target="_blank">Funcionários</a></li></b>
```

```
    <b><li><a href="login_contribuinte.php" target="_blank">Contribuintes</a></li></b>
```

```
</ul>
```

```
</nav>
```

```
<nav id="titulo">
```

```

<ul type="square">
<li><a>Acesso Restrito</a>
</ul>
</nav>
</header>
<div id="interface"> <!-- div: tag que gera uma delimitação da página -->
<figure class="principal"> <!-- figure: tag que especifica que a imagem faz parte de uma artigo -->
 <!-- img: tag de inserção de figura -->
</figure>
</div>
<footer id="rodape"> <!-- footer: tag cria um rodapé na página -->
<p>Pricila Alves - Graduanda em Engenharia Cartográfica / E-mail: alvess.pricila@gmail.com
</footer>
</body>
</html>

```

- Área de acesso restrito aos contribuintes (arquivo login_contribuinte.php)

```

<!doctype html>
<html>
<head>
  <title>Contribuintes</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

```

```
        <link rel="stylesheet" href="css/estilo_contribuinte.css"/>
</head>
<body>
    <header id="cabecalho">
        <nav id="titulo">
            <ul type="square">
                <li><a>Restrito a Contribuintes</a>
            </ul>
        </nav>
    </header>

    <div>
        <form action="func_login_contribuinte.php" method="post">
            <br> Usuário: <input type="text" name="email" value="" /> <br>
            <br> Senha: <input type="password" name="senha" value="" />
            <input type="submit" name="enviar" value="Ok"/>
        </form>
    </div>
    <footer id="rodape">
        <p>Pricila Alves - Granduanda em Engenharia Cartográfica / E-mail: alvess.pricila@gmail.com
    </footer>
</body>
```

```
</html>
```

- Função de validação dos dados dos contribuintes (arquivo func_login_contribuinte.php)

```
<?php
```

```
    session_start(); //Inicia uma nova sessão ou resume uma sessão existente
```

```
$dbh = pg_connect("host=localhost user=postgres password=postgres dbname=MACAPARANA"); /*conexão com o banco de dados */
```

```
$email = $_POST['email']; // variável recebe o valor do email //
```

```
$senha = $_POST['senha']; // variável que recebe o valor da senha
```

```
$logar = pg_query($dbh, "SELECT * FROM contribuintes.cadastro where email ='$email' AND senha='$senha'");
```

```
$verificarLogar = pg_num_rows($logar);
```

```
if ($verificarLogar == 1) {
```

```
    $_SESSION['emailSession'] = $email;
```

```
    $_SESSION['senhaSession'] = $senha;
```

```
    echo "<meta http-equiv=refresh content='0; URL=mapa.php';>";
```

```
}else{
```

```
    if($verificarLogar == 0){
```

```
        echo "<script> alert ('Login incorreto e/ou não existe!');</script><meta http-equiv=refresh content='0;
```

```
URL=login_contribuinte.php';>";
```

```
    }
```

```
}
```

```
?>
```

- Acesso restrito aos funcionarios (arquivo login_funcionario.php)

```
<!doctype html>
<html>
<head>
  <title>Funcionários</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <link rel="stylesheet" href="css/estilo_funcionario.css"/>
</head>
<body>
  <header id="cabecalho">
    <nav id="titulo">
      <ul type="square">
        <li><a>Restrito a Funcionários</a>
      </ul>
    </nav>
  </header>
  <div>
    <form action="func_login_contribuinte.php" method="post">
      <br> Usuário: <input type="text" name="email" value="" /> <br>
      <br> Senha: <input type="password" name="senha" value="" />
      <input type="submit" name="enviar" value="Ok"/>
    </form>
  </div>
</body>
</html>
```

```

</form>
</div>
  <footer id="rodape">
    <p>Pricila Alves - Graduanda em Engenharia Cartográfica / E-mail: alvess.pricila@gmail.com
  </footer>
</body>
</html>

```

- Função de validação dos dados dos funcionários (arquivo func_login_funcionario.php)

```

<?php
    session_start(); //Inicia uma nova sessão ou resume uma sessão existente
    $dbh = pg_connect("host=localhost user=postgres password=postgres dbname=MACAPARANA"); /*conexão com o banco de dados */
    $email = $_POST['email']; // variável recebe o valor do email //
    $senha = $_POST['senha']; // variável que recebe o valor da senha
    $logar = pg_query($dbh, "SELECT * FROM funcionarios.cadastro where email ='$email' AND senha='$senha'");
    $verificarLogar = pg_num_rows($logar);
    if ($verificarLogar == 1) {
        $_SESSION['emailSession'] = $email;
        $_SESSION['senhaSession'] = $senha;
        echo "<meta http-equiv=refresh content='0; URL=mapa.php';>";
    }else{

```

```
        if($verificarLogar == 0){
            echo "<script> alert ('Login incorreto e/ou não existe!');</script><meta http-equiv=refresh content='0;
URL=login_funionario.php';>";
        }
    }
?>
```

ANEXO

Anexo 1 - Modelo de extrato cadastro imobiliário do município de macaparana

		PREFEITURA MUNICIPAL DE MACAPARANA SECRETARIA DE FINANÇAS SEAC Endereço: RUA DR. ANTÔNIO XAVIER, S/N-CENTRO Telefone: (081)3639-1156 CNPJ: 11.361.888/0001-04	
		EXTRATO CADASTRO IMOBILIÁRIO	
Data Inclusão: Nº do Cadastro:	Nº Cad. CGM: Nº Cad. Anterior:	Data Intervenção: Último Usuário:	
Identificação do Imóvel			
Natureza: Situação:	Contribuição IPTU: Contribuição TAXAS:	Validade: Validade:	
Localização do Imóvel			
Inscrição: Seção: Logradouro: Número: Bairro: Loteamento:	Face: Complemento:	Referência Loteamento: Quadra: Insc. Ant.:	Cod. Lote: CEP:
Identificação do Contribuinte			
Nome: Identidade:	Tipo Pessoa:	CPF/CNPJ:	
Endereço para Correspondência			
Logradouro: Número: Bairro: UF: CEP:	Complemento:	Cidade: Reside:	
Medidas da Unidade			
Testada Princ.: Prof. Principal: Área Terreno: Área Unidade:	Testada Fictícia: Fração Ideal: Área total de Construção: Área Comum da Unidade:	0,01 0,0000 50,00 0,00	Nº de Pavimentos: 1 Ano Construção: 2013 Área Coberta: 0,00 Zona Fiscal: 57
Informações Sobre o Terreno			
Situação da Quadra: Patrimônio: Ocupação Terreno:	Pedologia: Topografia: Limitação:	Nivelamento: Tipo Terreno:	
Identificação da Edificação			
Situação Rua: Situação do Lote: Tipo de Construção: Patrimônio Edificação: Regime Ocupação: Cobertura: Esquadria: Instalação Sanitária:	Uso do Imóvel: Padrão Construção: Estado Conservação: Estrutura: Revestimento Superior: Revestimento Interno: Vidros: Condomínio:		
Água () Esgoto ()	Limpeza () Pavimento ()	Galerias Pluviais () Guias Sargetas ()	Energia () Iluminação () Telefone () Col. Lixo () Passeio ()
Reavaliação		Tributos	
Nº Processo:		Valor IPTU: Taxa de Conservação de Vias: Taxa de Limpeza Pública: Taxa de Coleta de Lixo: Taxa de Iluminação:	
		Valor Venal do Imóvel	