15º Congresso de Cadastro Multifinalitário e Gestão Territorial
3º Encontro de Professores de Cadastro Territorial
COBRAC
7 a 9 de NOVEMBRO|2022 | EVENTO VIRTUAL
Florianópolis - SC
Realização: UFSC
Através do: PPGTG Programa de Pós-graduação em Engenharia de Transportes e Gestão Territorial
E GOTT GRUPO DE OBSERVAÇÃO E TRANSFORMAÇÃO DO TERRITÓRIO

PROGRAMAÇÃO BASEADA EM SQL APLICADA AO CADASTRO TERRITORIAL MULTIFINALITÁRIO

SQL-Based Programming Applied to Territorial Management Engineering

Pedro Jardel Barbosa Pinto
Universidade Federal de Viçosa
Departamento de Engenharia Civil
pedro.jardel@ufv.br

Éder Teixeira Marques
Universidade Federal de Viçosa
Departamento de Engenharia Civil
eder@ufv.br

Caroline Fani Teixeira
Universidade Federal de Viçosa
Departamento de Engenharia Civil
caroline.fani@ufv.br

João Felipe Emerick Hubner
Universidade Federal de Viçosa
Departamento de Engenharia Civil
joao.emerick@ufv.br

Rafael Antunes Pereira
Universidade Federal de Viçosa
Departamento de Engenharia Civil
rafael.a.pereira@ufv.br

Resumo:
O Cadastro Territorial Multifinalitário (CTM) é uma ferramenta poderosa de organização e ordenamento territorial, sendo um instrumento fundamental para auxiliar no planejamento e na gestão pública. Neste cenário, é interessante que no CTM seja empregada a S*tructured Query Language* (SQL), e a partir disso algumas plataformas pertinentes sejam definidas para o uso da SQL, dentre elas o *PostgreSQL*, explicitado neste trabalho. Dessa maneira, o PostgreSQL viabiliza o uso de diversas funções, como vistas, regras, gatilhos e simulações. Fundamentado nisto, pode-se obter resultados de forma mais eficientes ao utilizar a programação estruturada em SQL, o que impulsiona a estruturação e organização de uma base cartográfica cadastral, proporcionando até mesmo a espacialização de dados complexos. Nesse caso, um exemplo é a aplicação para o cálculo do Imposto Territorial Predial Urbano (IPTU), uma operação que pode ser realizada aplicando essa técnica.

Palavras-chave: CTM, SQL, *PostgreSQL*.

Abstract

The Territorial Management Engineering (CTM) is a powerful tool for territorial organization and planning and a fundamental instrument for public planning and management. In this scenario, it is desirable to use the Structured Query Language (SQL) in the CTM. Based on this, some relevant platforms are defined for the use of SQL, includingPostgreSQL, which is explained in this work. *PostgreSQL* enables the use of several functions, such as views, rules, triggers and simulations. This allows more optimized results with the use of SQL, which boosts the structuring and organization of a cadastral cartographic base, thus providing even the spatialization of complex data. In this case, it can be exemplified by the application for the calculation of the Urban Property Land Tax, an operation that can be carried out through the use of this technique.

Keywords: CTM, SQL, POSTGRESQL.

1 INTRODUÇÃO

O Cadastro Territorial Multifinalitário (CTM) é o inventário territorial oficial e sistemático de um Município e baseia-se no levantamento dos limites de cada parcela, que recebe uma identificação numérica inequívoca (MINISTÉRIO DAS CIDADES, 2009).

Compete aos Municípios promover, no que couber, adequado ordenamento territorial, mediante planejamento e controle do uso, do parcelamento e da ocupação do solo urbano (ESTATUTO DA CIDADE, 2001). Sabe-se que o CTM assegura tal função por auxiliar na organização e desenvolvimento das Cidades.

Dessa forma, o CTM pode ser compreendido como uma estruturação em grande escala de informações urbanísticas, sociais, econômicas e de caráter público. É essencial que tais informações estejam vinculadas a um Sistema de Referência (SR), ou seja, as informações podem ser georreferenciadas. Uma das principais características do CTM é a sua utilização como alicerce para a gestão territorial urbana, substanciando o planejamento público em âmbito municipal, regional e nacional.

Segundo Oliani (2006), em caráter nacional, é de suma importância ter dimensão do território e sabe-se que uma forma de alcançar este cenário é por meio da utilização do CTM. Apesar disso, no Brasil, existem dificuldades de colocar em prática essa ação, visto que a base de dados cadastrais em alguns municípios é desatualizada e muitas vezes inexistente. Além disso, em algumas Prefeituras há a falta de recursos para atuar com essas bases.

Sendo assim, a gestão territorial carece, em sua concepção, do mapeamento detalhado das regiões de interesse de forma a direcionar e otimizar análises, sejam elas na esfera pública, social, econômica, voltadas ao planejamento urbano ou a medidas corretivas. A organização, armazenamento e manutenção dos dados mapeados emergem com uma importância fundamental para prover a gestão territorial urbana de seu insumo central, que são as bases cartográficas cadastrais.

Neste contexto, surgem diversas metodologias para a manutenção de dados espaciais: Sistemas de Informação Geográficas (SIG), plataformas de *webmapping*, estruturas de servidores de arquivos, dentre outros. Muitas destas estruturas são associadas a arquivos georrelacionais vetoriais (como o *shapefile*), nos quais as informações são anexadas a uma geometria básica, podendo estar ou não atreladas a um Sistema de Referência de Coordenadas (SRC).

Contudo, com a disseminação da Linguagem de Consultas Estruturadas, ou *Structured Query Language* (SQL), foram incrementados artifícios que otimizam e potencializam a organização e estruturação de uma base de dados. Em sua concepção, o SQL trata-se de uma linguagem de programação voltada a dados relacionais, ou seja, dados organizados em tabelas. É uma tecnologia utilizada em larga escala em sistemas comerciais, bancários, reservas de voo, ou aplicações com dados estruturados. SQL é a linguagem de consulta orientada para estas aplicações (NETO, 2021).

Em um ambiente de cadastro territorial multifinalitário existem plataformas específicas que viabilizam o tratamento de informações cartográficas e espaciais. É o caso do Sistema Gerenciador de Banco de Dados (SGBD) *PostgreSQL*, por meio de sua extensão *PostGIS*. Existem trabalhos que comprovam a aplicação deste SGBD para o CTM, como o proposto por Marotta (2021).

Portanto, para o caso do CTM, é de grande adequação a utilização de sistemas gerenciadores de banco de dados relacionais baseados em SQL. São inúmeras vantagens que visam potencializar o armazenamento, a disponibilização dos dados e as grandezas essenciais utilizadas para a gestão territorial, principalmente as de caráter público. O *PostgreSQL* (SGBD tomado como referência para este trabalho) possui características primordiais para uma base de dados unificada, como as bases voltadas ao CTM, as quais serão abordadas na sequência.

2 OBJETIVO

O objetivo do presente trabalho é apresentar pontos que substanciam a utilização de ferramentas fundamentadas em SQL para construção, organização e manutenção de uma base cartográfica cadastral. O intuito é mostrar as principais vantagens e benefícios da utilização desta linguagem de consultas para a organização de dados geoespaciais, tomando como diretriz a estrutura de dados orientada a objeto.

3 APLICAÇÕES DE SQL

3.1 Estruturação de uma base cartográfica cadastral

O *PostgreSQL*, assim como a maioria dos SGBD's são providos de interfaces diversas e trabalham em um sistema de Servidor-Cliente: o núcleo dos dados é o servidor e as diversas interfaces que são capazes de se conectar ao núcleo são chamadas de clientes. Para o âmbito do CTM, onde as informações armazenadas são acessadas de diversas esferas simultaneamente, a configuração Servidor-Cliente se destaca em detrimento das demais, nas quais nem sempre é possível realizar o acesso ao mesmo dado através de interfaces distintas ao mesmo tempo.

É válido ressaltar que o *PostgreSQL* teve sua origem em 1986 e é um poderoso banco de dados de código aberto com vários recursos capazes de armazenar e mensurar cargas de trabalho muito complexas. Justifica-se o seu uso uma vez que essa plataforma apresenta segurança, possibilidade de recuperação de dados e a integração desses.

É importante ressaltar que, anterior a composição da base de dados, é preciso fundamentar o produto em normas técnicas que dão o suporte e norteiam a estrutura que deve ser

implementada. A cartografia cadastral pode ser considerada uma especialização da cartografia digital e, portanto, carece de uma definição do modelo de dados vetoriais a serem desenvolvidos. São diversas as estruturas de dados vetoriais presentes na literatura, como a estrutura *Spaghetti*, a estrutura topológica, a georrelacional e a estrutura de dados orientada a objetos.

As normatizações vigentes dão destaque a estrutura de dados orientadas a objetos. É o caso da ISO 19.152/2012 – *Land Administration Domain Model* e também da ET-EDGV 3.0/2017 – Especificações Técnicas para Estruturação de Dados Geoespaciais Vetoriais. A ISO 19.152 é direcionada ao cadastro e prevê a padronização de conceitos e terminologias voltadas à gestão territorial. Já a ET-EDGV 3.0 é uma norma brasileira que padroniza a construção de bases cartográficas digitais, mas garante a possibilidade de adequações e personalizações para temas específicos, como o CTM (MAROTTA, 2021).

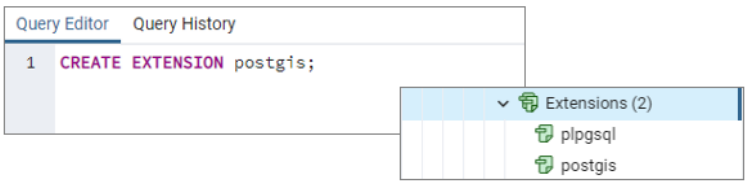
Seja qual for a normatização definida para a construção da base de dados, é imprescindível o desenvolvimento de um modelo conceitual. De forma sintetizada, um modelo conceitual consiste em um diagrama elaborado em alto nível que detalha e caracteriza a organização, as relações e as condições entre as classes de objetos entre si, bem como a relação entre os atributos de tais classes de objetos. Para a elaboração de uma base de dados em SQL é essencial que seja definido um modelo conceitual completo.

A partir do modelo conceitual estruturado é possível realizar a implementação de classes de objetos organizados em coleções semelhantes quanto a determinado atributo. Tais coleções são nomeadas de esquemas, ou *schemas*, e são essenciais para a administração da segurança de informações na base de dados. Com o advento dos esquemas, os dados deixam de ser relacionados diretamente a usuários e passam a ser anexados às coleções. Assim, pode-se afirmar que os *schemas* são o meio de conexão entre um grupo de usuários e uma classe de objetos.

Além disto, a organização dos dados em esquemas é análoga à distribuição de pastas e arquivos em um sistema computacional convencional. Esta analogia é mais facilmente compreendida por usuários básicos e contribui para o entendimento de uma base de dados cartográfica.

Por fim, para a atribuição da grandeza espacial a uma base de dados, comumente é preciso que se ative extensões específicas à mesma. Para o caso de bases de dados estruturadas pelo *PostgreSQL*, a extensão espacial referida consiste no *PostGIS*. Uma das interfaces possíveis de acesso à plataforma SGBD é o aplicativo *PgAdmin4*, no qual é possível administrar a base de informações. A partir deste aplicativo é possível realizar consultas e comandos em SQL, inclusive a ativação da extensão citada, conforme ilustrado na Figura 1.

Figura 1 - Ativação da Extensão Espacial PostGIS.



Fonte: PgAdmin (2022).

Depois de ativada a extensão é possível inserir dados na base, e assim organizá-la de

acordo com as necessidades do projeto de CTM em andamento. Na sequência serão apontadas as principais vantagens da utilização do *PostgreSQL* na organização da base de dados cadastrais.

3.2 Aplicação do PostgreSQL em base de dados cadastrais

Partindo do ponto de existência da base de dados cadastrais e da verificação do carregamento da mesma com informações tratadas, espera-se que haja o manuseio frequente dos dados provenientes de tal base. Para isto, conforme já apresentado, são disponibilizadas credenciais de usuários e senhas, que permitem aos usuários acessar determinada carga de dados.

Uma das diretrizes de um CTM é a presença de unicidade de parcelas cadastrais e, com a utilização de um SGBD como o *PostgreSQL* essa unicidade é facilmente garantida: a base de dados trabalha com um sistema de chaves primárias e secundárias, garantindo a relação inequívoca entre as informações. Dado este evento, é possível realizar a composição da base de dados em tabelas de grandezas bem definidas, únicas e fundamentais para a organização de um cadastro de caráter multifinalitário. Inicialmente, as tabelas utilizadas em um CTM podem ser: a) tabela de unidades cadastrais; b) tabela de lotes; c) tabela de edificações; d) tabela de seções de logradouro e; e) tabela de proprietários. Ambas englobam informações referentes às suas grandezas, compondo o básico para o desenvolvimento de um CTM (MAROTTA, 2021).

A partir destas tabelas-chave é possível o direcionamento e adequação do CTM para quaisquer finalidades. Uma alternativa muito interessante para viabilizar este direcionamento é a elaboração de vistas, também chamadas de *views*. Estas vistas podem ser entendidas como particularizações sobre as principais tabelas da base de dados. Para criar uma vista, a título de exemplo, basta executar um código similar ao código apresentado na Figura 2.

Figura 2 - Exemplo de Vistas em SQL.

Query Editor | Query History

```
1  CREATE OR REPLACE VIEW auditoria.bci_edificacao_telheiro_com_parede
2   AS
3   SELECT row_number() OVER () AS id,
4      cad_edificacao_a.id_edificacao,
5      cad_edificacao_a.inscricao,
6      cad_edificacao_a.tipo,
7      cad_edificacao_a.paredes,
8      cad_edificacao_a.geom
9     FROM dados.cad_edificacao_a
10   WHERE cad_edificacao_a.tipo = 'Telheiro' AND cad_edificacao_a.paredes <> 'Sem';
```

Fonte: PgAdmin (2022).

É importante salientar que, conforme o próprio nome sugere, por padrão, uma vista é utilizada apenas para visualização. Logo, não é comum a existência de vistas que possibilitam um usuário realizar modificações no dado espelhado e, consequentemente, no dado da tabela que foi particularizada.

Além das vistas, existem as vistas materializadas (*materialized views*). Estes elementos possuem a mesma finalidade das vistas dinâmicas, com uma característica peculiar: as vistas

materializadas permitem a visualização momentânea das tabelas e das informações originais e, por mais que os dados passem por atualizações, as vistas materializadas seguirão apresentando as informações existentes no momento da materialização.

Com o intuito de simplificar a edição de atributos e geometria das tabelas constituintes de um CTM, foram desenvolvidos adereços em SQL que otimizam o trabalho de analistas em uma linha de produção. Tais adereços consistem em regras (*rules*) e gatilhos (*triggers*). Estes fundamentos viabilizam até mesmo a edição de vistas e têm como principal finalidade a automatização de determinadas funções recorrentes na administração de um banco de dados.

Uma aplicação comum e muito útil de uma regra (*rule*) no âmbito do CTM consiste no direcionamento de edições em uma vista para a camada original. Este procedimento é essencial para manter a unicidade de informações, uma vez que o armazenamento de informações diferentes para o mesmo dado em duas camadas distintas (a vista e tabela) é inviabilizado. A Figura 3 exemplifica a execução de uma *rule* que redireciona as modificações na camada "auditoria.bci_edificacao_telheiro_com_parede" para a camada "dados.cad_edificacao_a" no ato da modificação.

Figura 3 - Exemplo de Regras em SQL.

Query Editor | Query History

```
1  CREATE RULE bci_edificacao_telheiro_com_parede_cad_edificacao_a AS
2      ON UPDATE TO auditoria.bci_edificacao_telheiro_com_parede
3      DO INSTEAD
4  (UPDATE dados.cad_edificacao_a SET tipo = new.tipo, paredes = new.paredes
5    WHERE (cad_edificacao_a.id_edificacao = new.id_edificacao));
```

Fonte: PgAdmin (2022).

Outro artifício muito utilizado em bases cartográficas cadastrais é a implementação de gatilhos. Um caso corriqueiro consiste na necessidade de atualização em tempo real de áreas das parcelas cadastrais, lotes e edificações. Esta funcionalidade pode ser facilmente acionada via SQL para o instante seguinte a atualização da geometria da feição, conforme ilustrado na Figura 4, na qual a área construída da edificação é atualizada após a modificação da geometria da mesma.

Figura 4 - Exemplo de Gatilhos em SQL.

Query Editor | Query History

```
1  CREATE OR REPLACE FUNCTION att_area_construida()
2    RETURNS TRIGGER
3    LANGUAGE PLPGSQL
4    AS
5  $$
6  BEGIN
7      UPDATE dados.cad_edificacao_a
8      SET area_construida = st_area(geom);
9  END;
10 $$
11
12 CREATE TRIGGER att_area_construida()
13   AFTER UPDATE
14   ON dados.cad_edificacao_a
15   FOR EACH STATEMENT
16   EXECUTE PROCEDURE att_area_construida();
```

Fonte: PgAdmin (2022).

Observa-se que a ativação do *trigger* acontece para a atualização da geometria (*geom*) de

determinada feição. De imediato, quando há a modificação da geometria, o valor da área é atualizado.

No âmbito fiscal do Cadastro Territorial Multifinalitário pode-se, com o auxílio das regras, vistas e gatilhos, gerar simulações para o Imposto Territorial Predial Urbano (IPTU). Estas simulações consistem em códigos completos que utilizam diversos insumos de natureza descritiva, geométrica e tributárias para efetuar cálculos imprescindíveis para o Cadastro Fiscal.

Um exemplo de simulação que pode ser desenvolvido é a simulação de valores do Imposto Territorial Predial Urbano (IPTU) através da SQL. Os códigos das consultas devem considerar camadas distintas para subsidiar os cálculos. É importante salientar que a fórmula de cálculo de IPTU varia de acordo com cada Código Tributário Municipal, ou seja, cada município está sujeito a uma fórmula de cálculo distinta. Contudo, em termos gerais, este tributo é normalmente calculado com base em: a) valores unitários de terreno e edificação, b) áreas dos lotes e áreas construídas; c) fatores de correção geral, de terreno e de edificações; d) fração ideal e alíquotas. Para efeitos de ilustração, considere a equação 1 como exemplo de fórmula de cálculo de IPTU:

$$V_{IPTU} = Alq \times \left[(VUT \times FI \times FC_T \times AT) + \sum_{i=1}^{ne} (VME \times AE \times FC_E) \right] \times FC_G \qquad (1)$$

Onde:

V_{IPTU} = Valor do IPTU da unidade cadastral;

Alq = Alíquota aplicada sobre o imóvel;

VUT = Valor unitário do terreno;

$FI = \frac{Áreaconstruídadaunidade}{Áreaconstruídatotalnolote}$ = Fração ideal da unidade cadastral;

FC_T = Fator de correção de valores sobre o terreno;

AT = Área do terreno;

ne = Número total de edificações pertencentes a unidade cadastral;

VME = Valor do m² edificado (valor unitário da edificação);

AE = Área da edificação;

FC_E = Fator de correção de valores sobre a edificação;

FC_G = Fator de correção de valores sobre a unidade cadastral;

A partir disso, viabiliza-se a determinação dos valores calculados via SQL, por um código conforme o apresentado na Figura 5. Nota-se que este código puxa informações de diversas camadas, também implementadas em SQL na mesma base de dados.

Observa-se que o código gera uma camada chamada “simulacao_iptu” que fica armazenada no esquema nomeado “simiptu”. São explicitados os fatores notáveis da fórmula de cálculo, como a alíquota, as áreas, as frações ideais, dentre outros.

Figura 5 - Exemplo de Código para Cálculo de IPTU.

Query Editor | Query History

```
CREATE OR REPLACE VIEW simiptu.simulacao_iptu AS
SELECT DISTINCT ON (vv.inscricao),unid.inscricao,vv.inscricao_anterior,vv.area_terreno,vv.area_construida_terreno,
    vv.area_construida_unidade,vv.area_construida_comum,vv.fracao_ideal,vv.venal_terreno,vv.nome_proprietario,
    vv.tipo_logradouro,vv.nome_logradouro,vv.vut,vv.vme,vve.venal_edificacao_total,
    COALESCE(vve.venal_edificacao_total + vv.venal_terreno,vv.venal_terreno) as vvi,
    vv.aliquota,vv.fce,vv.fct,vv.fcg,vv.imune_isento_iptu,
    CASE
        WHEN vv.imune_isento_iptu = 1 THEN 1    -- Não há imunidade/isenção
        ELSE 0                                  -- Há imunidade/isenção
    END AS imune_isento_iptu_fator,
    CASE
        WHEN vv.imune_isento_iptu = 1 AND venal_area_comum IS NOT NULL
            THEN vv.aliquota * (COALESCE(vve.venal_edificacao_total + vv.venal_terreno,vv.venal_terreno) +
                 vva_comum.venal_area_comum*vv.fracao_ideal)
        WHEN vv.imune_isento_iptu = 1 AND venal_area_comum IS NULL
            THEN vv.aliquota * COALESCE(vve.venal_edificacao_total + vv.venal_terreno,vv.venal_terreno)
        ELSE 0
    END AS iptu_calculado,
    COALESCE(vve.geom,unid.geom) AS geom
FROM dados.cad_unidade_cadastral_p unid
```

Fonte: PgAdmin (2022).

4 RESULTADOS ESPERADOS

A partir das informações apresentadas neste trabalho, podem ser listados alguns resultados esperados depois da aplicação da SQL para manutenção de bases cartográficas cadastrais. Tais resultados consistem em vantagens significativas na administração das informações, acessos, gerência e disponibilização da base de dados.

Com a utilização de vistas espera-se resultados que potencializam a validação de dados. É possível filtrar as informações a serem consideradas pelos analistas em determinados casos e, a partir disso, potencializar a identificação de erros lógicos/semânticos nas classes de objetos. Também é possível criar camadas de visualização que evidenciam o dado de interesse.

Em relação aos gatilhos e às regras, cabe ressaltar que são artifícios muito utilizados e otimizam o funcionamento de uma base cartográfica. O usuário pode ter sua quantidade de trabalho reduzida e determinadas ações, como o cálculo de áreas e a composição de caracteres nas unidades cadastrais podem ser realizados de forma automática. Todos estes pontos agregam maior robustez à base, tornando-a moderna, funcional e otimizada para gerência de um CTM.

Por fim, em relação aos códigos apresentados, os ganhos foram consideráveis quando se trata de observação e espacialização de informações mais complexas. No caso apresentado, foram calculados os valores de IPTU para todas as unidades cadastrais de uma base de dados, tratando-se de um cálculo derivado de diversas outras camadas. A Figura 6 mostra o resultado com a aplicação SQL.

Figura 6 – Resultados obtidos pelo simulador de IPTU obtido pela aplicação da SQL.

Fonte: Os autores (2022).

Verifica-se que os resultados obtidos com a aplicação da SQL mostram-se apropriados do ponto de vista prático e permite o aprimoramento do SIT, por meio da inserção de novas regras, e a segurança das bases cartográficas cadastrais.

5 CONCLUSÃO

Portanto, é perceptível que o Cadastro Territorial Multifinalitário auxilia sobremaneira na organização territorial e apresenta a funcionalidade de ampliar desenvolvimento socioeconômico do Município. A utilização da programação baseada em SQL potencializa o CTM e deve ser considerada fundamental na estruturação e manipulação dos dados cadastrais.

Diante do exposto, verifica-se que a aplicação da plataforma *PostgreSQL* na análise dos dados possibilita, a partir dados geoespaciais bem estruturados, construir uma base cartográfica cadastral muito mais organizada. O que é uma vantagem, visto que em muitos Municípios há a falta de organização destes dados, dificultando o entendimento da realidade do território municipal e causando prejuízos na gestão territorial.

Conclui-se que, o uso da linguagem computacional SQL tornou possível a filtragem de informações que devem ser analisadas, facilitando a identificação de possíveis erros gerados. A partir da implementação da programação em SQL verifica-se ganhos substanciais nas análises realizadas, tanto em termos de redução de tempo quanto de acurácia. Dessa forma, a gestão municipal torna-se mais eficaz e pode dar respostas mais rápidas às questões relacionadas à administração do território, como é o caso de possíveis alterações no valor da terra e o seu impacto para a administração pública e na vida dos munícipes.

Referências

COMISSÃO NACIONAL DE CARTOGRAFIA – CONCAR. **NCB-CC/E 0001B08**: ESPECIFICAÇÕES TÉCNICAS PARA ESTRUTURAÇÃO DE DADOS **GEOESPACIAIS**

VETORIAIS. Brasil: Ministério do Planejamento, Desenvolvimento e Gestão, Comissão Nacional de Cartografia, 2017. 44 p. Disponível em <https://docs.ufpr.br/~deni_ern/CD2020/A1/ET-EDGV_versao_3.0.pdf>. Acesso em: 19 ago. 2022.

GONÇALVES, Rômulo Parma. **Modelagem conceitual de bancos de dados geográficos para cadastro técnico multifinalitário em municípios de pequeno e médio porte**. 2008. 170 f. Dissertação (Mestrado) - Curso de Engenharia Civil, Universidade Federal de Viçosa, Viçosa - Mg, Brasil, 2008. Disponível em <https://www.locus.ufv.br/bitstream/123456789/3700/1/texto%20completo.pdf>. Acesso em: 23 ago. 2022.

GONÇALVES, R.P. LISBOA FILHO, J. VIEIRA, C. A. O. **Modelagem Conceitual de Banco de Dados Geográficos aplicada ao Cadastro Técnico Multifinalitário**. Brasil. 2009. Disponível em <https://seer.ufu.br/index.php/revistabrasileiracartografia/article/view/43647/22911>. Acesso em: 20 ago. 2022.

MAROTTA, Victor dos Santos. **Um framework para a disponibilização de informação geográfica** municipal na Web. 2021.

NORMAS BRASIL. **PORTARIA N.° 511 DE 07 DE DEZEMBRO DE 2009**: Diretrizes para a criação, instituição e atualização do Cadastro Territorial Multifinalitário nos municípios brasileiros. Brasília: Norma Federal, 2009. Disponível em <https://www.normasbrasil.com.br/norma/portaria-511-2009_217279.html>. Acesso em: 22 ago. 2022.

OLIANI, Luiz Octávio. **NOÇÕES DE CADASTRO TERRITORIAL MULTIFINALITÁRIO - CTM**. Paraná: Conselho Regional de Engenharia e Agronomia do Paraná, 2016. 36 p. Disponível em: <https://www.mundogeo.com/arquivos/news/nocoes_cadastro_territorial-final.pdf>. Acesso em: 19 ago. 2022.

OLIVEIRA, Q. G. LOPES, L. F. B. **Aumento de Performance por meio da Otimização de Banco de Dados.** Universidade Paranaense- UNIPAR- Paranavaí - PR - Brasil. Disponível em <https://silo.tips/download/otimizaao-e-performance-de-banco-de-dados-utilizando-sql-tuning>. Acesso em: 21 ago. 2022.

SOBRE o PostgreSQL. Disponível em <https://www.postgresql.org/about/>. Acesso em: 18 ago. 2022.