

**UTILIZAÇÃO DE UM SERVIDOR PARA O DESENVOLVIMENTO DE UM
CADASTRO TERRITORIAL MULTIFINALITÁRIO**

**USE OF A SERVER TO THE DEVELOPMENT OF A MULTIPURPOSE
TERRITORIAL CADASTRE**

Rafael Antunes Pereira,
Universidade Federal de Viçosa, Brasil,
rafael.a.pereira@ufv.br

Sabrina Roberta de Moura Martins,
Universidade Federal de Viçosa, Brasil,
sabrina.roberta@ufv.br

Pedro Jardel Barbosa Pinto,
Universidade Federal de Viçosa, Brasil,
pedro.jardel@ufv.br

Natália Aparecida Braga Ferreira,
Universidade Federal de Viçosa, Brasil,
natalia.a.braga@ufv.br

Arthur Gomes Ribeiro,
Universidade Federal de Viçosa, Brasil,
arthur.g.ribeiro@ufv.br

Éder Teixeira Marques,
Universidade Federal de Viçosa, Brasil,
eder@ufv.br

RESUMO

O desenvolvimento de um Cadastro Territorial Multifinalitário (CTM) carece da estruturação de um centro de armazenamento, manuseio e disponibilização de dados. Neste contexto, emergem várias possibilidades para a organização de centrais de dados, com destaque para os Servidores. Tais Servidores podem ser configurados

em máquinas específicas ou através de armazenamento em nuvem. No caso de arquiteturas de máquinas são necessárias especificações técnicas de *hardwares* e *softwares* de forma a consolidar os alicerces necessários para o desenvolvimento de bancos de dados e sistemas diversos envolvidos em um projeto de CTM. O presente trabalho busca elencar e descrever as principais arquiteturas associadas a um Servidor voltado ao Cadastro Territorial Multifinalitário.

Palavras-chave: Servidor, Cadastro Territorial Multifinalitário.

ABSTRACT

The development of a Multipurpose Territorial Cadastre (CTM) lacks the structuring of a storage center, handling and availability of data. In this context, several possibilities emerge for the organization of data centers, with emphasis on the Servers. Such Servers can be configured on specific machines or through cloud storage. In the case of machine architectures, hardware and software technical specifications are required in order to consolidate the foundations needed for the development of databases and diverse systems involved in a CTM project. The present work seeks to list and describe the main architectures associated with a Server dedicated to the Multipurpose Territorial Cadastre.

Keywords: Server; Multipurpose Territorial Cadastre.

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 (ESTATUTO DA CIDADE, 2001). É inerente ao CTM a organização das informações territoriais em bancos de dados que viabilizem análises sobre o território e fundamentem a tomada de decisão dos gestores responsáveis (BARBOSA, 2022). Neste contexto, emerge a necessidade de recorrer a recursos tecnológicos, como os Servidores.

Servidores são estruturas computacionais robustas concebidas para o armazenamento e processamento de informações em massa. Tais estruturas possuem arquiteturas e configurações específicas baseadas em *softwares* e

hardwares adequados para disponibilizar os dados em determinadas plataformas e sistemas – como os sistemas envolvidos na construção de um CTM (AMARAL, 2012).

A presença de um servidor de arquivos durante o desenvolvimento de um Cadastro Territorial Multifinalitário garante a dinamicidade do projeto. Grande parte das estruturas envolvidas em um cadastramento territorial é fundamentada em Sistemas de Informação Geográficas (SIG) e estes têm suas funcionalidades potencializadas ao serem substanciadas por um Sistema Gerenciador de Banco de Dados (SGBD) (BARBOSA, 2022).

As etapas iniciais de um CTM englobam atividades de coleta, organização e armazenamento de informações, arquivos e documentos. Em seguida, são elaboradas as bases cartográficas de referência, contendo geometrias e tabelas descritivas. Tanto a etapa de coleta de informação quanto a etapa de elaboração de conteúdo possuem uma interface direta com um SGBD e, conseqüentemente, com um Servidor. O mesmo ocorre com as etapas subsequentes, envolvendo o desenvolvimento de sistemas *webmapping*, disponibilização de informações cadastrais aos usuários e manutenção cadastral.

Portanto, são muitos os fatores que corroboram com o emprego de uma central de informações, sistemas e dados para o desenvolvimento de um projeto de Cadastro Territorial Multifinalitário. Tal central é correspondente a um Servidor composto por recursos computacionais específicos e dedicados para otimizar a execução de um CTM.

2 OBJETIVOS

O objetivo do presente trabalho consiste em apresentar elementos computacionais que viabilizam a configuração de um servidor de arquivos, visando a plena elaboração de um projeto de Cadastro Territorial Multifinalitário. O trabalho também objetiva justificar o emprego de tal estrutura com a finalidade de potencializar o desenvolvimento do CTM.

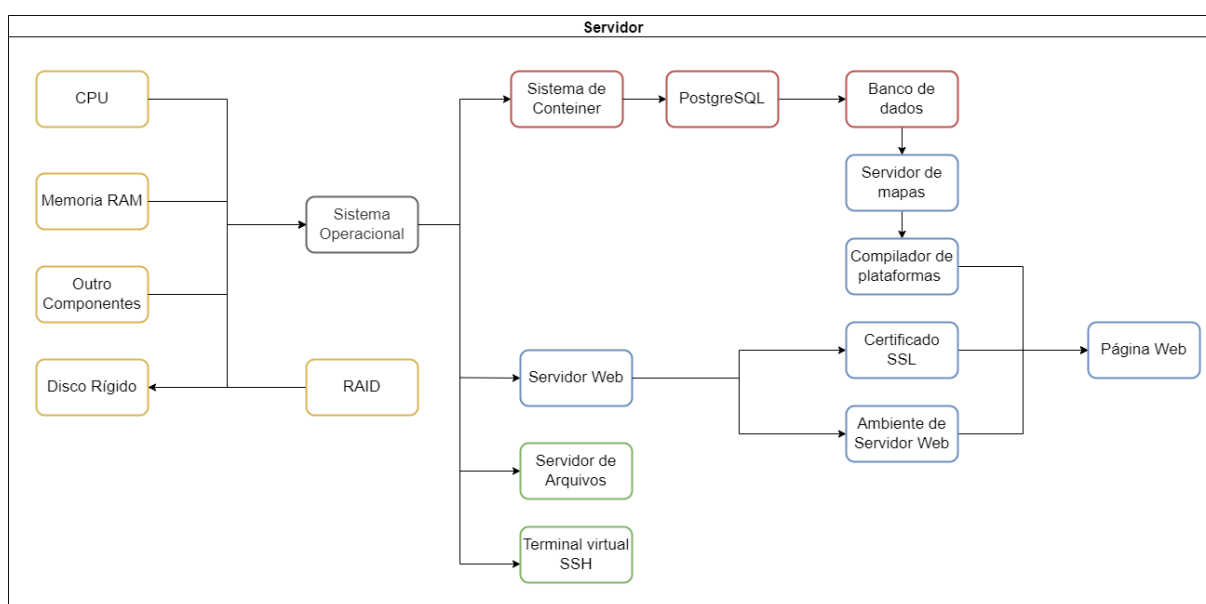
3 METODOLOGIA

Para nortear o desenvolvimento do presente trabalho, serão apresentados os recursos computacionais que devem ser atrelados a uma máquina. Portanto, os tópicos subsequentes são referentes a um servidor físico de arquivos e sistemas.

Cabe ressaltar que o avanço dos serviços de armazenamento em nuvem proporciona uma alternativa à organização de um servidor físico: estruturas implementadas na modalidade *online*.

O passo inicial da configuração consiste no planejamento do Servidor, com a definição dos sistemas e funcionalidades que devem ser implementadas. A Figura 1 apresenta uma organização da construção do Servidor, bem como os processos que se desencadeiam no sistema.

Figura 1 – Fluxograma do servidor



Fonte: Elaborado pelo autor (2022)

De forma sumarizada, os sistemas podem ser divididos entre o *front-end* e *back-end*. O primeiro conta com *softwares* visualizadores de dados e camadas, além de portais de informação, validadores de dados e controle de segurança. O segundo é baseado em programas e *scripts* que vão lidar com os dados de forma direta, encapsulando-os e criando um ambiente estável para o usuário

3.1 O hardware

A disposição física de um Servidor difere de um computador pessoal comum em relação às peças que o compõem. É necessário que as peças que o compõem: o processador, a memória RAM, a fonte de energia e sobretudo o disco rígido sejam adequados para garantir um bom desempenho em cenários de elevado fluxo de dados.

O que mais se diferencia em um servidor é a questão do armazenamento de dados. Um Servidor conta com peças e configurações a nível de máquinas bem específicas. Existem diversas maneiras de estruturar o armazenamento de dados e uma delas é a utilização do *Redundant Array of Inexpensive Drives (RAID)*. Basicamente, são discos rígidos que armazenam os dados em partições de tamanhos iguais, organizados de maneira específica evitando perda permanente de informações. Simultâneo a isso, é facilitada a manutenção do computador com a possibilidade de correções na máquina ligada, garantindo uma maior estabilidade ao sistema.

3.2 Sistema Operacional

Previamente à instalação e configuração de *softwares*, faz-se necessária a definição de um sistema operacional (SO). Um SO faz a ponte entre o usuário e a máquina. Os comandos informados pelo usuário são transformados em linguagem de máquina, na sequência são processados pelo computador e a resposta é transmitida para o usuário. Além disso, o SO é responsável por administrar todos os recursos computacionais disponíveis, dinamizando e garantindo estabilidade para todo o sistema.

Um servidor geralmente vai ter um SO direcionado para esse fim, com modificações pontuais que dão o aporte necessário para o funcionamento da máquina. Existem variados tipos de sistemas disponíveis no mercado, de código fechado ou livre. Dentre elas, destacam-se as soluções Linux, as quais são bem difundidas pelos desenvolvedores.

3.2.1 Terminal virtual SSH

O terminal virtual é a ferramenta que realiza o acesso a um sistema secundário, afastado da máquina que exerce controle. No contexto atual, o protocolo mais adequado em questão de eficiência e proteção é o *Secure Shell*, também conhecido pela sigla SSH. Este protocolo é responsável pela conexão cliente-servidor criptografada, ou seja, estabelece um canal para comunicação cujo os comandos são codificados e entendidos somente pelas máquinas envolvidas no processo.

O *software* servidor SSH separa uma única porta para a conexão (por padrão a 22) que pode ser alterada, possibilitando a administração por meio de linhas de

códigos no terminal ou em ambientes gráficos, nos quais são realizadas atividades tais como execução de comandos, edição de textos e transferência de arquivos. No contexto de administração e configuração de servidores é possível notar a importância da utilização do SSH.

3.2.2 *Servidor de Arquivos*

A produção compartilhada configura-se como uma realidade comum na maioria dos projetos de grande porte, sobretudo na gestão territorial. Nesse contexto, é indispensável a disponibilização de arquivos atualizados e em tempo real para a equipe de desenvolvimento. Tendo em vista essa necessidade, a utilização de Servidores pode garantir o compartilhamento de arquivos e impressoras com as máquinas presentes em uma mesma rede.

Os arquivos de interesse estão alocados em um servidor que disponibilizará o acesso à rede, essa centralização possibilita informações atualizadas ao alcance de todos os usuários e realização de cópias de segurança. Também vale ressaltar a economia de recursos proporcionada, uma vez que os dados estarão apenas em uma máquina.

Os servidores de arquivos, geralmente, são associados ao sistema operacional, dessa forma, o *Windows* utiliza o protocolo *Server Message Block* (SMB), já o *Linux* e distribuições baseadas possuem o conjunto de programas conhecido como *SAMBA*.

3.2.3 *Servidor Web*

O servidor web como *hardware* ou *software* possui a mesma finalidade: interpretar requisições *Hypertext Transfer Protocol* (HTTP) administrando essa comunicação com o cliente. No contexto de *software*, o servidor atua como um gerenciador dos dados, que geralmente são páginas HTML, encaminhados ao usuário. Dentre os servidores mais utilizados podemos citar o *Apache* e *Nginx*.

3.2.3.1 Certificado SSL

Se por um lado o encaminhamento das páginas *web* é gerenciado pelo servidor, por outro essa transmissão precisa de segurança sendo o *Secure Sockets Layer* (SSL) o protocolo que assume tal responsabilidade.

O SSL é a ferramenta responsável por estabelecer um canal de transmissão protegido por meio da criptografia que garante a integridade das informações trocadas, preserva a identidade dos usuários e impede o extravio das mensagens. Para tal, por intermédio do SSL gera-se um certificado para o site no qual é especificado um único caminho de acesso, também conhecido como endereço, autenticando e estabelecendo uma conexão segura.

3.2.3.2 Ambiente de Servidor Web

Os ambientes que dão suporte ao servidor são os *Servlets*, traduzido como um pequeno servidor, esses atuam como um extensor de capacidade como o *Tomcat* e o *GlassFish*. Segundo a documentação da *Oracle* (2014), um *Servlet* é uma classe da linguagem de programação *JAVA* que atua no servidor *web* interagindo com as requisições *HTTP* dos clientes e reportando conteúdos dinâmicos.

3.2.4 Sistema de Container

Uma tecnologia não crucial para o funcionamento de um servidor, porém capaz de erradicar os problemas de compatibilização entre *softwares* é o sistema de container. O *Docker*, traduzido como container, é uma plataforma de código aberto que encapsula aplicações e distribui esse pacote em nuvem na qual o usuário poderá adquirir as versões executáveis chamadas de “imagens” em sua máquina.

De acordo com a documentação oficial do *Docker* (2022), o sistema de containerização pode ser instanciado em qualquer ambiente, encapsular aplicações em conjunto com suas dependências em desenvolvimento e replicar os programas mantendo o padrão. Ademais, a flexibilidade do *Docker* pode ser assegurada pois sua execução está a nível de disco, memória, processamento e rede.

3.2.5 Banco de dados

Um banco de dados é definido por ser uma coleção de informações ou dados organizados em estruturas que normalmente serão armazenados eletronicamente em um computador (CARVALHO, 2008). O SGBD oferece um fácil acesso e organiza os dados permitindo modificações, atualizações e arranjos. A unidade básica do SGBD é organizada em tabelas relacionais, otimizando a eficiência de se estruturar e proporcionar um processamento mais rápido das informações.

Conforme abordado por Marotta (2018), existem vários tipos de SGBD disponíveis destacando-se as soluções *Open Source* como o *PostgreSQL*. Este SGBD é amplamente utilizado no mercado e a vasta documentação disponível justifica a elevada disseminação da plataforma.

No contexto do CTM, uma característica fundamental é o aspecto espacial das informações armazenadas. Com isso, é essencial que o SGB permita o manuseio de dados geográficos referenciados espacialmente. Também por este motivo, justifica-se a utilização do *PostgreSQL*, por meio de sua extensão *PostGIS*. Existem trabalhos que comprovam a aplicação deste SGBD para o CTM, como o proposto por (MAROTTA, 2021).

3.2.6 Servidor de mapas

Para viabilizar a disponibilização de mapas, é necessário o acesso às informações cartográficas. Este acesso deve ser intermediado através de requisições, as quais podem ser realizadas com diferentes especificações tendo em vista os resultados esperados.

De acordo com Parma (2008), os servidores de mapas seguem o padrão cliente/servidor. Neste padrão as solicitações são realizadas pelo cliente, por intermédio do navegador *web*, na sequência são interpretadas pelo servidor e, por fim, são retornadas as informações posteriormente interpretadas e exibidas no formato de páginas *web* com dados espaciais associados. Dessa forma, os servidores de mapas atuam como um SIG customizável e limitado à conexão de rede. Dentre os servidores atualmente disponíveis destacam-se o *Geoserver*, *Mapserver* e o *QGIS Server*.

3.2.7 *Compilador de plataformas*

Uma proposta interessante para o desenvolvimento de aplicações que exibem informações geográficas na *internet*, é utilizar um *runtime* de *JavaScript*, ele garante uma série de vantagens como segurança e estabilidade. Além disso, o compilador é capaz de lidar bem com interações de entradas e saídas em tempo real e não apresenta problemas com diversas chamadas assíncronas.

Um bom exemplo é o *Node.js*, desenvolvido sobre o motor *JavaScript V8*, *engine* desenvolvida pela Google. O *Node.js* tem como objetivo fornecer uma maneira fácil de construir aplicações escaláveis. Funcionando como uma máquina virtual no Servidor, todos os processos em *JavaScript* passam a ser resolvidos fora do navegador evitando uma série de problemas que outros *Frameworks* enfrentam.

4 RESULTADOS

Para substanciar o trabalho, foram aplicados os conhecimentos apresentados na configuração de um servidor com o intuito de suprir as necessidades de projetos de Cadastro Territorial Multifinalitário. As plataformas foram configuradas pelo Grupo de Engenharia para Gestão Territorial (GENTE), instituído na Universidade Federal de Viçosa – MG, Brasil. Os objetivos consistiam em estabelecer uma rede local para produção cadastral, além de disponibilizar *websites* e plataformas de *webmapping* desenvolvidos pelo Grupo de forma a assegurar a eficiência de cada finalidade do CTM.

A princípio, foi formatada uma máquina com o sistema operacional *Ubuntu Server*, arquitetura *Debian*, tendo em vista a segurança e ampla possibilidade de personalização que as aplicações *Linux* permitem. As configurações apresentadas a seguir foram realizadas via linha de comando.

Para *softwares*, foram aderidas as aplicações de servidores disponíveis no *Linux*: o terminal virtual SSL e para distribuição de arquivos, o SAMBA. Já, no armazenamento de informações, foi acatada a imagem do PostgreSQL com a extensão de dados espaciais Postgis obtida via Docker.

Em seguida, o ambiente de distribuição web foi estabelecido utilizando o *Nginx* como servidor, o *Certbot* para emitir os certificados SSL e o *Tomcat* como ambiente

de gerenciamento. Atualmente, a maioria dos sites desenvolvidos pelo Grupo são aplicações *Node.js* e que realizam consultas ao servidor de mapas *Geoserver*.

Alguns dos produtos elaborados de acordo com as especificações supracitadas estão disponíveis na rede, como o *WebGIS* do CTM realizado no município de Bom Despacho – MG (disponível em webgente.bomdespacho.genteufv.com.br), o Geoportal baseado em *QGIS Server* do CTM do município de Itabirito (disponível em itabirito.genteufv.com.br), dentre outros.

5 CONCLUSÕES

Os aspectos apresentados agregam significativo valor aos Servidores e, portanto, tornam a tarefa de configuração de um Servidor meticulosa com necessidade de capacitação técnica de alto nível para os profissionais responsáveis.

É possível observar a importância na utilização de um Servidor bem como sua construção e configuração a fim de armazenar e processar uma quantidade significativa de dados de maneira segura e estável. Quando utilizado durante a realização de um CTM, o processo torna-se dinâmico uma vez que o acesso às informações pode ser feito por diferentes usuários simultaneamente em uma base de dados única.

A utilização de um sistema operacional *Linux* possibilita que os dados gerados durante o cadastro sejam armazenados de forma segura e com ampla capacidade de personalização, visto que as informações são protegidas por um certificado de caminho único, resguardando e impedindo o extravio de informações.

Ademais, conclui-se que a utilização de um servidor com as devidas característica e configuração possibilita um gerenciamento de dados de forma dinâmica durante a execução do Cadastro Territorial Multifinalitário. A centralização dos dados em um único ambiente torna a visualização geral e correção de erros mais simples, além de garantir segurança e integridade dos arquivos.

REFERÊNCIAS

AMARAL, Allan Francisco Forzza. **Redes de Computadores**. Colatina - ES: Instituto Federal do Espírito Santo, 2012. 38-39 p. Disponível em: <http://200.129.0.130/bitstream/handle/123456789/710/Rede%20de%20Computadore>

s%20-%20marca%20de%20corte.pdf?sequence=2&isAllowed=y. Acesso em: 13 out. 2022.

CARVALHO, Carlos Alberto de. Documentos. In: USO de Banco de Dados para Análise Espacial no Sistema de Gestão Territorial da Defesa Agropecuária. [S. l.]: Empresa Brasileira de Pesquisa Agropecuária, 2008. cap. 68. Disponível em: <https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/17724/1/d68analisespacialaftosa.pdf>. Acesso em: 13 out. 2022.

PARMA, Gabriel Cremona. Mapas Cadastrais na Internet: Servidores de mapas. Anais XIII Simpósio Brasileiro de Sensoriamento Remoto, São José dos Campos: INPE, 2007. p. 1311-1319.

NODE JS. [S. l.], 2022. Disponível em: <https://nodejs.org/pt-br/docs/>. Acesso em: 12 out. 2022.

[HTTPS://WWW.ORACLE.COM/BR/DATABASE/WHAT-IS-DATABASE/](https://www.oracle.com/br/database/what-is-database/). [S. l.], 2022. Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 12 out. 2022.

DOCKER overview. [S. l.], 2013-2022. Disponível em: <https://docs.docker.com/get-started/overview/>. Acesso em: 12 out. 2022

THE JAVA EE 6 Tutorial. [S. l.], 2013. Disponível em: <https://docs.oracle.com/javaee/6/tutorial/doc/bnafe.html>. Acesso em: 12 out. 2022.

CANTIERI, ALVARO ROGÉRIO. ANÁLISE DE SEGURANÇA DO PROTOCOLO SSL - SECURE SOCKET LAYER. Orientador: Prof. Dr. Eduardo Parente. 2002. Dissertação (Mestre em Engenharia Elétrica) - Universidade Federal do Paraná, [S. l.], 2002.

MARIMOTO, Carlos E. Linux Redes e Servidores: Guia Prático. 2. ed. atual. [S. l.: s. n.], 2013.

ØSTERGAARD, Jakob; BUESO, Emilio. The Software-RAID HOWTO. [S. l.], 3 jun. 2010. Disponível em: <https://tldp.org/HOWTO/Software-RAID-HOWTO.html>. Acesso em: 12 out. 2022.

TANENBAUM, Andrew S.; BOS, Herbert. SISTEMAS OPERACIONAIS MODERNOS. 4. ed. [S. l.: s. n.], 2016.

MOTA FILHO, João Eriberto. Descobrindo o LINUX: Entenda o sistema operacional GNU/Linux. [S. l.: s. n.], 2012. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=04ZRDQAAQBAJ&oi=fnd&pg=PA27&dq=sistema+operacional&ots=9ECxihWiH7&sig=JsZ2H55W_bt1fUM4w_XPI2fPszU#v=onepage&q=sistema%20operacional&f=false. Acesso em: 12 out. 2022.