MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA CARTOGRÁFICA

RODRIGO VIEIRA CASANOVA MONTEIRO
TIAGO ALMEIDA DE CARVALHO

PROTÓTIPO DE VECTOR TILES PARA O BDGEX.

RIO DE JANEIRO
2023

RODRIGO VIEIRA CASANOVA MONTEIRO
TIAGO ALMEIDA DE CARVALHO

PROTÓTIPO DE VECTOR TILES PARA O BDGEX.

Projeto de Final de Curso apresentado ao Curso de Graduação em Engenharia Cartográfica do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Bacharel em Engenharia Cartográfica.

Orientador(es): Marcos de Meneses Rocha, D.Sc.
Maurício Carvalho Mathias de Paulo, M.Sc.

Rio de Janeiro

2023

©2023
INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 – Praia Vermelha
Rio de Janeiro – RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

Vieira Casanova Monteiro, Rodrigo; Almeida de Carvalho, Tiago.
Protótipo de Vector Tiles para o BDGEx. / Rodrigo Vieira Casanova Monteiro e Tiago Almeida de Carvalho. – Rio de Janeiro, 2023.
40 f.

Orientador(es): Marcos de Meneses Rocha e Maurício Carvalho Mathias de Paulo.

Projeto de Final de Curso (graduação) – Instituto Militar de Engenharia, Engenharia Cartográfica, 2023.

1. BDGEx. 2. Vector Tiles. 3. Serviços de Mosaico. 4. WebMapping. 5. MapServer. i. de Meneses Rocha, Marcos (orient.) ii. Carvalho Mathias de Paulo, Maurício (orient.) iii. Título

RODRIGO VIEIRA CASANOVA MONTEIRO
TIAGO ALMEIDA DE CARVALHO

Protótipo de Vector Tiles para o BDGEx.

Projeto de Final de Curso apresentado ao Curso de Graduação em Engenharia Cartográfica do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Bacharel em Engenharia Cartográfica.

Orientador(es): Marcos de Meneses Rocha e Maurício Carvalho Mathias de Paulo.

Aprovada em 09 de outubro de 2023, pela seguinte banca examinadora:

Documento assinado digitalmente
MARCOS DE MENESES ROCHA
Data: 19/10/2023 13:21:04-0300
Verifique em https://validar.iti.gov.br

Cel **Marcos** de Meneses Rocha - D.Sc. do IME - Presidente

Documento assinado digitalmente
IVANILDO BARBOSA
Data: 18/10/2023 14:35:48-0300
Verifique em https://validar.iti.gov.br

TC **Ivanildo** Barbosa - D.Sc. do IME

Documento assinado digitalmente
FELIPE FERRARI
Data: 18/10/2023 15:05:53-0300
Verifique em https://validar.iti.gov.br

Maj Felipe **Ferrari** - M.Sc. do IME

Documento assinado digitalmente
MAURICIO CARVALHO MATHIAS DE PAULO
Data: 18/10/2023 15:44:13-0300
Verifique em https://validar.iti.gov.br

Maj **Maurício** Carvalho Mathias de Paulo - M.Sc. do IME

Rio de Janeiro
2023

Este projeto é dedicado à pequena Ayla, que nos acompanhou indiretamente ao longo do trabalho e que certamente terá um futuro brilhante pela frente.

AGRADECIMENTOS

Agradecemos a nossos professores e instrutores, em especial aos nossos orientadores, Cel Marcos de Meneses Rocha e Maj Maurício Carvalho Mathias de Paulo, que nos acompanharam ao longo do desenvolvimento desse projeto, sempre dispostos a dedicar tempo e recursos para nos auxiliar.

Agradecemos também ao IME e a todos os amigos que fizemos ao longo dos anos passados nesta casa. Os momentos bons e ruins que dividimos em nossa formação irão nos manter unidos, não importa o quão espalhados pelo Brasil ou pelo mundo estivermos.

Por fim, agradecemos às nossas famílias. Eu, Rodrigo, agradeço em especial a minha mãe Rosemary, meu pai Ricardo, minha avó Maria Aparecida, minha irmã Mariana e minha tão aguardada sobrinha Mavie. Eu, Tiago, agradeço em especial a minha mãe Alessandra e meu padrasto Luis Antônio. Vocês foram essenciais para nossa formação pessoal e profissional, nos apoiando e ajudando ao longo de toda a vida. Esperamos que este trabalho retribua uma fração dos sacrifícios e investimentos que vocês fizeram por nós.

"Não adentre a boa noite apenas com ternura,
A velhice queima e clama ao cair do dia;
Fúria, fúria contra a luz que já não fulgura."
(Dylan Thomas)

RESUMO

O Banco de Dados Geográficos do Exército (BDGEx) é um sistema que disponibiliza produtos de geoinformação fornecidos pela Diretoria de Serviços Geográficos do Exército (DSG). As funcionalidades de visualização dos dados vetoriais atualmente sofrem de problemas de colisão e discrepância de densidade de informação ao visualizar dados vetoriais em pequenas escalas. Foi proposta uma solução utilizando o mosaico vetorial juntamente com a estrutura de visualização em múltiplas escalas já presente no BDGEx. Foi preparado um ambiente virtual para simular a arquitetura cliente-servidor, no qual foi instalada e configurada uma solução utilizando contêineres. Em seguida foram feitos testes de disseminação e visualização dos dados multi-escala utilizando o padrão Mapbox Vector Tiles (MVT). Os dados foram armazenados em um banco PostgreSQL, disseminados por meio do conjunto MapServer e MapCache, e foram apresentados nos clientes QGIS e OpenLayers. Foi identificado que o padrão MVT não transmite aos clientes o sistema de referência de coordenadas (SRC), o que dificulta a utilização no BDGEx. Isto pode ser contornado utilizando o SRC padrão (EPSG:3857) ou declarando no sistema cliente. O protótipo demonstra que há viabilidade técnica de implantar o padrão MVT para representar dados vetoriais no BDGEx. Além disso, foi identificado que a API de mosaicos vetoriais da OGC está em desenvolvimento no MapServer, podendo ser utilizada futuramente.

Palavras-chave: BDGEx. Vector Tiles. Serviços de Mosaico. WebMapping. MapServer.

ABSTRACT

The Banco de Dados Geográficos do Exército (BDGEx) is a system that provides geoinformation products supplied by the Diretoria de Serviços Geográficos do Exército (DSG). The vector data visualization functionalities currently suffer from collision problems and information density discrepancies when visualizing vector data at small scales. A solution was proposed using the vector tiles together with the multiscale visualization structure already present in the BDGEx. A virtual environment was prepared to simulate the client-server architecture, in which a solution using containers was installed and configured. Tests were then carried out to disseminate and visualize the multi-scale data using the Mapbox Vector Tiles (MVT) standard. The data was stored in a PostgreSQL database, disseminated using MapServer and MapCache, and displayed in the QGIS and OpenLayers clients. It was identified that the MVT standard does not transmit the coordinate reference system (CRS) to clients, which makes it difficult to use in BDGEx. This can be circumvented by using the standard CRS (EPSG:3857) or by declaring it in the client system. The prototype demonstrates that it is technically feasible to implement the MVT standard to represent vector data in BDGEx. In addition, it was identified that the OGC vector tiles API is under development and could be used in the future.

Keywords: BDGEx. Vector Tiles. Tile Services. WebMapping. MapServer.

LISTA DE ABREVIATURAS E SIGLAS

CSW	Catalogue Service for the Web
HTTP	Hypertext Transfer Protocol
MVT	Mapbox Vector Tiles
OGC	Open Geospatial Consortium
SSH	Secure Shell
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service

SUMÁRIO

1 INTRODUÇÃO . . . 12
1.1 OBJETIVO . . . 13
1.2 REQUISITOS . . . 13

2 REVISÃO DA LITERATURA . . . 14
2.1 SERVIÇOS OGC UTILIZADOS NO BDGEX . . . 14
2.2 *WEBMAPPING* COM SERVIÇOS DE MOSAICO . . . 14
2.3 *VECTOR TILES* . . . 15
2.4 SISTEMAS DE REFERÊNCIA EM SERVIÇOS DE MOSAICO . . . 16
2.5 ESTRUTURA DE SERVIÇOS DE MOSAICO DO BDGEX . . . 17
2.6 MÁQUINAS VIRTUAIS . . . 17
2.7 DOCKER E CONTEINERIZAÇÃO . . . 17

3 MATERIAIS E MÉTODOS . . . 19
3.1 COMPONENTES DA SOLUÇÃO PROPOSTA . . . 19
3.2 PREPARAÇÃO DO AMBIENTE . . . 20
3.2.1 CONFIGURAÇÃO DOS SERVIÇOS WMS E WFS NO MAPSERVER . . . 21
3.2.2 CONFIGURAÇÃO DOS SERVIÇOS WMS-C NO MAPCACHE . . . 22
3.2.3 CONFIGURAÇÃO DOS SERVIÇOS MVT NO MAPCACHE . . . 23
3.2.4 CONFIGURAÇÃO DA VISUALIZAÇÃO EM MULTIESCALA NO MAPSERVER E NO MAPCACHE . . . 24
3.3 TESTES DE CADA ETAPA DO PROCESSO DE DISSEMINAÇÃO . . . 25
3.3.1 POSTGIS . . . 25
3.3.2 WMS . . . 25
3.3.3 WMS-C . . . 26
3.3.4 MVT - MAPSERVER . . . 27
3.3.5 MVT - MAPCACHE . . . 28
3.4 EXPERIMENTOS UTILIZANDO A ESTRUTURA DE MOSAICO MULTI-ESCALA DO BDGEX . . . 29
3.4.1 QGIS . . . 29
3.4.2 OPENLAYERS . . . 30

4 RESULTADOS E DISCUSSÃO . . . 31
4.1 TESTES DE CADA ETAPA DO PROCESSO DE DISSEMINAÇÃO . . . 31
4.1.1 POSTGIS . . . 31
4.1.2 WMS . . . 31

4.1.3 WMS-C 32
4.1.4 MVT - MAPSERVER 32
4.1.5 MVT - MAPCACHE 33
4.2 EXPERIMENTOS UTILIZANDO A ESTRUTURA DE MOSAICO MULTI-ESCALA DO BDGEX 33
4.2.1 QGIS 33
4.2.2 OPENLAYERS 34

5 CONCLUSÃO 35

REFERÊNCIAS 36

APÊNDICE A – PROCEDIMENTOS DE INSTALAÇÃO 38
A.1 KUBUNTU 38
A.2 VIRTUAL BOX 38
A.3 INTERFACES DE CLIENTE REMOTO 39
A.3.1 WINDOWS 39
A.3.2 LINUX 39
A.4 DOCKER 39

1 INTRODUÇÃO

O BDGEx é um serviço do Exército Brasileiro que visa prover geoinformação em formatos matricial e vetorial, tendo como público alvo o Exército e as forças armadas, além da sociedade civil. Dessa forma, sua principal função é disponibilizar dados cartográficos provenientes da DSG na *web*. (DSG, 2023b)

O BDGEx permite a visualização dos produtos em mosaicos, separados por categorias e subcategorias de produtos (DSG, 2023a). Uma das principais funcionalidades do BDGEx é a representação visual dos dados, que é utilizada tanto na consulta por navegação, quanto consumida por outros sistemas do Exército por meio dos serviços de mapas na *web*. Dessa forma, este trabalho concentra-se nos Serviços de Mosaico (*Tile Services*) e nas diferenças entre os tipos desses serviços - *Raster tiles* e *Vector tiles*, quando aplicados a dados vetoriais. Os serviços de mosaico padronizam a forma de solicitar os dados vetoriais, dividindo o espaço geográfico em *tiles* que os clientes podem solicitar individualmente. (OGC, 2023)

Os serviços do tipo *Raster tiles* são utilizados no BDGEx atualmente para representar dados vetoriais e matriciais, mas possuem algumas limitações em sua essência que afetam a visualização do cliente em algumas situações - especialmente representando dados vetoriais em escalas pequenas (DSG, 2023b). Problemas como a alta densidade de informações, espessura das linhas e tamanhos dos símbolos presentes nos produtos cartográficos não podem ser resolvidos pelo cliente, o que motiva a busca pelo uso de *Vector tiles*. A figura 1 apresenta um exemplo de uma representação de rodovias em diferentes escalas utilizando *Raster tiles*.

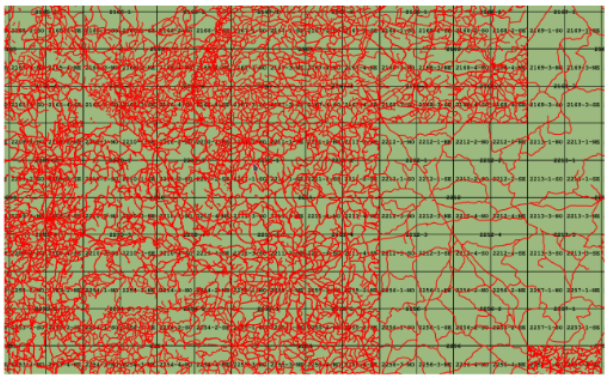


Figura 1 – Rodovias no BDGEx, em diferentes escalas

Ao utilizarmos serviços de *Vector tiles*, os dados são transmitidos em formato vetorial para o visualizador do cliente, que efetua o desenho do produto cartográfico e a respectiva simbolização (MAPBOX, 2023). Isto permite que o cliente realize alterações que

favoreçam a sua visualização. Assim, a principal vantagem dos *Vector tiles* é a flexibilidade, que permite alterar a simbolização conforme a necessidade do cliente.

O serviço de *Vector tiles* é muito utilizado no OpenStreetMap, que é um sistema de mapeamento colaborativo, livre, aberto e editável (MARTINELLI; ROTH, 2016). No entanto, esse serviço ainda não foi implementado no BDGEx. Tal lacuna constitui a principal motivação para a construção do protótipo cujo desenvolvimento será apresentado neste trabalho.

1.1 Objetivo

O objetivo deste projeto é produzir um protótipo do serviço de *Vector tiles* compatível com o BDGEx, visando melhorar a visualização de dados geográficos vetoriais em escalas pequenas, para fins de futura implementação.

1.2 Requisitos

O BDGEx é um sistema já consolidado, ao qual o protótipo produzido neste projeto deve se integrar. Portanto, foram adotados os seguintes requisitos para o desenvolvimento do protótipo:

- Os *tiles* devem ser atualizados quando uma carta for carregada no banco de dados PostGIS.
- Os dados armazenados no banco de dados PostGIS não devem ser alterados pelo sistema que produz os *Vector tiles*.
- O armazenamento de dados intermediários do processo de produção dos *Vector tiles* deve ser evitado.
- A introdução de novos componentes no sistema deve ser evitada.
- O sistema de referência do BDGEx (EPSG:4326) deve ser mantido.
- A compatibilidade com os níveis de zoom atualmente implementados no BDGEx deve ser garantida.

Essas restrições são necessárias para garantir a operacionalidade, compatibilidade e manutenibilidade da implementação no BDGEx.

2 REVISÃO DA LITERATURA

2.1 Serviços OGC utilizados no BDGEx

Os serviços de geoinformação permitem que vários serviços consumam informação de um mesmo servidor, através de uma padronização que garante a compatibilidade dos dados. A utilização recomendada no BDGEx é (DSG, 2023b):

- Web Map Service (WMS) (OGC, 2006b): Para visualização rápida dos produtos. A "Consulta por Navegação" do BDGEx utiliza o WMS, onde o usuário pede por uma representação da região e o servidor retorna uma imagem.
- Web Map Tile Service (WMTS) (OGC, 2010): Para visualização de produtos solicitando recortes do mosaico armazenados em cache. O cache permite armazenar os resultados do WMS permitindo reuso dos *tiles*.
- Web Feature Service (WFS) (OGC, 2006a): Para acesso aos mosaicos de dados vetoriais. Como trata-se de enviar um arquivo de banco de dados para o cliente, pode ser um processo demorado.
- Catalog Service for the Web (CSW) (NEBERT; VOGES; BIGAGLI, 2016): Para acesso ao catálogo de produtos do BDGEx, de forma similar à "Consulta por Metadados".

2.2 *Webmapping* com serviços de mosaico

O *Webmapping* é a prática de criar e compartilhar mapas interativos na web (MEDEIROS, 2010). Já os serviços de mosaico (ou *tile services*) são uma forma de publicar dados, podendo ser *Raster* ou *Vector tiles*, em que o mapa é dividido em pequenos *tiles* (quadrados) que podem ser armazenados em cache. A figura 2 ilustra esse conceito. Quando um usuário solicita um determinado trecho do mapa, o servidor entrega os *tiles* correspondentes em vez de enviar a imagem inteira, tornando o processo mais rápido e eficiente (OGC, 2023).

Os serviços de mosaico são uma das principais tecnologias usadas para implementar o *Webmapping*, pois permitem a exibição rápida de grandes quantidades de dados *Raster* e *Vector* em uma interface amigável ao usuário (OGC, 2023). Além disso, podem ser usados em conjunto com clientes de *Webmapping*, como a biblioteca OpenLayers, para criar mapas interativos que permitem ao usuário explorar diferentes camadas de dados, realizar zoom e pan, e interagir com informações geoespaciais em tempo real (OPENLAYERS, 2023).

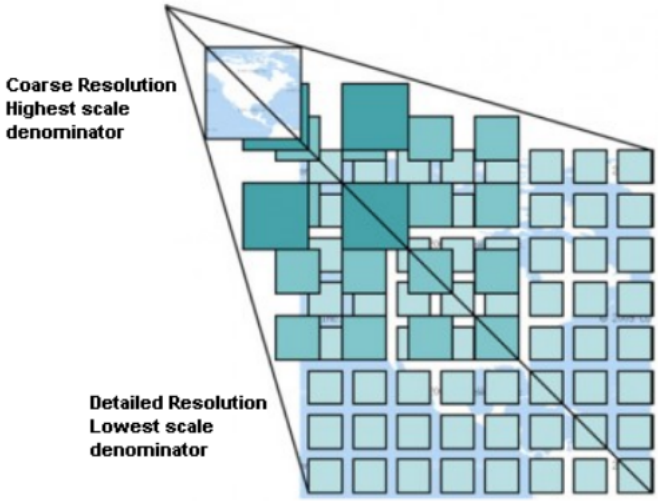


Figura 2 – Representação de um serviço de mosaico (QGIS, 2022)

Devido à divisão do espaço geográfico em *tiles*, os serviços de mosaico podem armazená-los em cache, evitando que o mesmo *tile* seja gerado várias vezes, devido a diferentes requisições relacionadas à mesma região. Isso permite que o serviço funcione com maior eficiência, além de reduzir o tempo de resposta experienciado pelo usuário. (OGC, 2023)

2.3 *Vector tiles*

Os *Vector tiles* são serviços de dados vetoriais otimizados para visualização em mapas online. Existem diversos serviços e formatos de arquivos que são distribuídos em cada serviço. Eles dividem o mapa em pequenos *tiles* (quadrados) que contêm informações vetoriais sobre a área representada, conforme ilustra a figura 3. Esses *tiles* podem ser carregados sob demanda, permitindo que o mapa seja exibido de forma mais rápida e responsiva. (OGC, 2023)

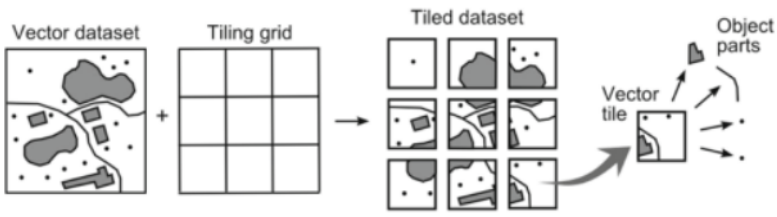


Figura 3 – Composição dos Vector tiles (GAFFURI, 2012)

O MVT oferece várias vantagens em relação a outros formatos de dados vetoriais no contexto de *webmapping*, como a possibilidade de estilo dinâmico, que permite a customização dos elementos de um mapa, como cores e espessuras de linhas. Os formatos de arquivos de MVT foram projetados para serem enviados pela internet, otimizados

para frações dos arquivos poderem ser representadas durante a transmissão. Os principais formatos disponíveis atualmente no MapServer são do padrão MVT: ".mvt" e ".pbf". (MAPBOX, 2023)

Os padrões WFS e MVT transmitem dados vetorias, mas foram desenvolvidos para aplicações diferentes. A tabela 1 apresenta algumas diferenças.

Tabela 1 – Diferenças entre os padrões WFS e MVT.

WFS	MVT
O dado é transmitido integralmente.	O dado é otimizado para transmissão e visualização.
Os vetores são baixados sem generalização.	Respeita níveis de visualização (pode haver generalização).
A área a recortar pode ser determinada pelo cliente.	A área é recortada conforme a estrutura de tiles.
Os dados são transformados para o formato escolhido para a transmissão.	Os arquivos do cache já estão prontos para a transmissão.
Formatos mais comuns: GML e SHP.	Formatos mais comuns: MVT e PBF.

2.4 Sistemas de referência em serviços de mosaico

Os sistemas de referência são um componente importante dos serviços de mosaico, pois garantem que as imagens constituem representações geograficamente coerentes com a realidade. Eles são usados para projetar as imagens em um sistema de coordenadas geográficas, permitindo que diferentes camadas de dados sejam sobrepostas e visualizadas em conjunto.

Embora a projeção UTM seja amplamente utilizada na cartografia, não é comumente utilizada nos serviços de *webmapping* em grande parte por três motivos:

1. Representação Global - A projeção UTM representa um fuso de 6 graus (GMBH, 2023). Enquanto isso atende bem representações locais, geralmente os sistemas de *webmapping* representam grandes porções do globo, portanto exigem representações que contemplam grandes áreas de maneira uniforme.

2. Deformações - A projeção UTM é cilíndrica, o que significa que quanto mais distante dos paralelos de interseção, maiores são as deformações (GMBH, 2023). Como os usuários podem navegar livremente entre as diferentes regiões, é interessante que a representação minimize essas distorções.

3. Desempenho - A projeção UTM exige cálculos complexos para tradução de coordenadas geográficas para coordenadas UTM e vice-versa. Em sistemas de *webmapping*

que trabalham com quantidades grandes de dados e precisam ser responsivos em tempo real, é melhor uma projeção cuja tradução seja mais simples.

Dentre os sistemas de coordenadas mais utilizados no MapServer e MapCache, destacam-se o WGS 84 / Pseudo-Mercator (EPSG:3857), o Google Maps Global Mercator - Spherical Mercator (EPSG:900913) e o WGS 84 (EPSG:4326) (MAPSERVER, 2023).

Os dois primeiros exemplos baseiam-se na projeção de Mercator original, a qual representa a superfície da Terra em unidades de comprimento lineares, tendo como unidade de medida o metro. O último, em contrapartida, representa a superfície da Terra em graus decimais, medindo a latitude e a longitude relativas ao WGS 84 (GMBH, 2023).

2.5 Estrutura de serviços de mosaico do BDGEx

O BDGEx utiliza uma estrutura de divisão de *tiles* específica, que foi projetada para representar apenas o espaço geográfico brasileiro, utilizar coordenadas geográficas, e níveis de visualização similares aos de escalas da cartografia sistemática (ex.: 1:100.000, 1:50.000). Desta forma, a grade do BDGEx tem extensão geográfica delimitada por longitude -125º a 0º, latitude -55º a 15º, no sistema de referência EPSG:4674 (SIRGAS 2000, coordenadas geográficas). (DSG, 2023b)

2.6 Máquinas Virtuais

Máquinas virtuais (VMs) são uma representação de uma máquina física, incluindo hardware virtualizado, sistema operacional, aplicativos e outros recursos de computação (DOCKER, 2021). O objetivo das VMs é fornecer uma maneira de compartilhar um único servidor físico (hardware) em várias máquinas virtuais isoladas, cada uma com seu próprio sistema operacional e conjunto de aplicativos. Dessa forma, é possível simular o ambiente de um servidor sem a necessidade de adquiri-lo ou alugá-lo.

As VMs são usadas em muitas áreas, incluindo desenvolvimento de software, teste, treinamento, ambientes de produção e muito mais. Elas são particularmente úteis para aqueles que precisam criar e executar sistemas operacionais e aplicativos diferentes em um único servidor físico, permitindo que os usuários economizem espaço, tempo e dinheiro.

2.7 Docker e Conteinerização

Docker é uma plataforma de conteinerização que permite aos desenvolvedores criar, implantar e executar aplicativos em contêineres. Os contêineres são um tipo de virtualização em que o sistema operacional do *host* é compartilhado com os contêineres,

permitindo que eles sejam executados de maneira isolada, mas com menos sobrecarga do que as VMs. (DOCKER, 2022)

Os contêineres Docker são leves, portáteis e escaláveis, o que os torna ideais para desenvolvimento de software e implantação de aplicativos em vários ambientes, incluindo ambientes locais, em nuvem e em dispositivos móveis. O uso de contêineres Docker também simplifica a implantação e gerenciamento de aplicativos, já que os contêineres podem ser implantados e atualizados rapidamente.

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os componentes e os procedimentos utilizados no protótipo para disponibilização de *Vector tiles* no BDGEx. Inicialmente, foi feita a preparação do ambiente, a qual compreende a instalação da solução Docker e configuração inicial dos serviços. Os serviços foram implementados e testados na seguinte sequência:

- WMS e WFS no MapServer
- WMS-C no MapCache
- MVT no MapServer
- MVT no MapCache

Esses testes foram realizados primeiramente com dados de uma única escala e posteriormente seguindo a estrutura do mosaico multi-escala do BDGEx. Os testes de consumo dos serviços de *Vector Tiles* foram efetuados tanto no QGIS quanto no OpenLayers.

3.1 Componentes da solução proposta

Foi utilizado o ambiente WebMapDocker, disponibilizado em <https://github.com/mauriciodev/webmapdocker/>, para configurar os contêineres Docker. O ambiente prepara um contêiner para cada um dos *softwares* NGINX, PyCSW, MapServer, MapCache, OpenLayers e PostGIS.

O NGINX cumpre o papel de *proxy* reverso, provendo um *Uniform Resource Locator* (URL) externo para conexão com os contêineres do Docker. Este é o ponto de entrada e saída principal do WebMapDocker, e é o contêiner que responde na porta 80.

O PyCSW age como catálogo de produtos de geoinformação, permitindo que sejam registrados e disseminados os metadados relativos aos produtos. Este serviço não tem relação direta com os *Vector tiles*, mas ele pode ser utilizado para consultar as informações do produto que contém a feição vetorial visualizada nos *Vector tiles*.

MapServer e MapCache fazem os papéis, respectivamente, do servidor responsável por produzir os *tiles* e do servidor responsável por montar o cache de *tiles* produzidos pelo MapServer e solicitar *tiles* quando for o caso.

Por fim, o PostGIS foi utilizado para armazenamento, pesquisa e filtragem dos dados geoespaciais. A forma de acesso na qual os testes foram conduzidos é no *Desktop* através do cliente QGIS. Foi testado também o acesso através do OpenLayers para comprovar

que o sistema pode ser acessado por outros clientes. A figura 4 representa visualmente os componentes descritos nesta seção.

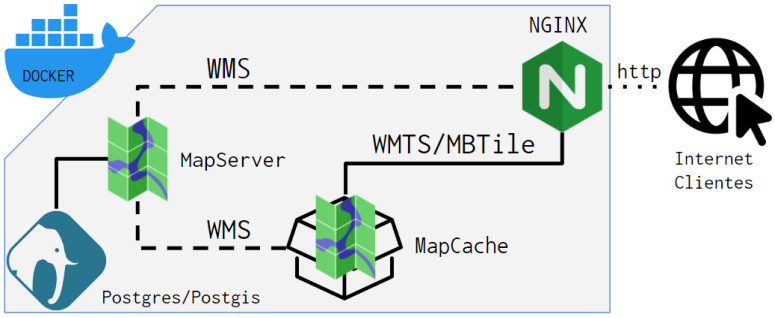


Figura 4 – Componentes do WebMapDocker

3.2 Preparação do ambiente

Foi criada uma máquina virtual no aplicativo Oracle VirtualBox, usando o sistema operacional Kubuntu, além do Docker com contêineres de MapServer, MapCache, PostGIS e NGINX. O passo a passo detalhado da instalação e configuração encontra-se no Apêndice A. Foram mapeadas as portas 22, 80 e 5432 da máquina virtual para a máquina hospedeira; essas portas correspondem respectivamente ao SSH, HTTP e POSTGRESQL.

Em seguida, foi instalado na máquina virtual o software openssh-server, para permitir o acesso remoto. O putty e o winscp foram instalados na máquina cliente, para permitir o acesso à máquina virtual. Desta forma, foi simulado um ambiente cliente-servidor, conforme ilustrado na figura 5, similar ao utilizado no BDGEx.

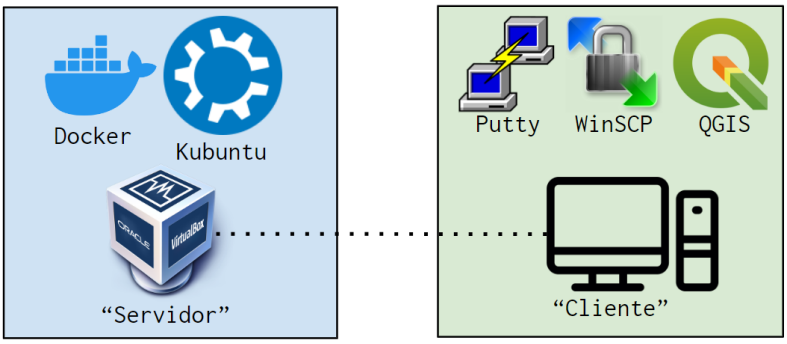


Figura 5 – Ilustração da arquitetura cliente-servidor simulada

Foram executados os procedimentos explicitados no Apêndice A utilizando o terminal do putty. Em seguida foram executados os passos descritos no repositório do MapServer, resultando na instalação e configuração inicial da solução na máquina virtual.

3.2.1 Configuração dos serviços WMS e WFS no MapServer

O WebMapDocker inclui na sua configuração padrão cinco (5) bancos de dados, intitulados geo25, geo50, geo100, geo250 e pycsw. Além destes, um banco de dados nomeado multiescala, posteriormente adicionado, armazena dados que possuem regras de representação em multiescala.

No arquivo docker-compose.yml, na configuração inicial da solução, estão definidos o usuário e senha do postgres, que conferem acesso ao banco de dados. Este usuário e senha foram inseridos no gerenciador de bancos de dados do QGIS para permitir o acesso e importação de dados.

A camada Trecho_Drenagem_L do conjunto de dados geográficos vetoriais relativo à carta de mapa índice 2987-2-NO, na escala 1:25.000, de 2019, obtida no BDGEx, foi importada no banco de dados. O arquivo .map foi atualizado incluindo uma declaração da camada do MapServer, utilizando uma conexão ao banco de dados PostgreSQL.

No arquivo .map, já existe uma camada que pode ser usada de modelo para inserção de outra. O tipo de geometria deve ser alterado para POINT, LINE ou POLYGON, de acordo com a geometria da nova camada.

Como a nova camada foi alimentada ao banco de dados, o código abaixo deve constar na sua declaração:

```
LAYER
    (...)
    CONNECTIONTYPE POSTGIS
    CONNECTION "host=postgis dbname=<db> user=postgres password=test port=5432"
    DATA "geom FROM <camada> USING UNIQUE id USING srid=<srid>"
    #EXTENT <lat_west> <lon_south> <lat_east> <lon_north>
    PROCESSING "CLOSE_CONNECTION=DEFER"
    (...)
END
```

Aqui foram substituídos os seguintes parâmetros, de acordo com a camada importada:

- <db> - O banco de dados no qual foi importada a camada, em letras minúsculas;
- <camada> - O nome da camada, exatamente como consta no banco de dados. É boa prática que o nome não possua espaços; se for o caso, renomeie a camada no banco de dados utilizando o Gerenciador de Bancos de Dados do QGIS;

- <srid> - O índice EPSG que identifica o sistema de referência de coordenadas da camada;
- <lat_west> <lat_south> <lat_east> <last_north> - Os limites oeste, sul, leste e norte da extensão da camada, respectivamente.

No caso em que ocorreu erro por causa de projeção, foi adicionado o sistema de projeção ao metadado:

```
LAYER
  METADATA
    "wms_srs"   "EPSG:4326"
  END
END
```

Por fim, utilizando o QGIS, foi acessado o menu Camada - Adicionar Camada - WMS/WMTS e criada uma conexão utilizando o URL abaixo, e foi verificado que o WMS da camada podia ser requerido como esperado:

<http://localhost:80/ms?MAP=/etc/mapserver/teste.map&mode=tile&tilemode=gmap&tile={x}+{y}+{z}&layers=all&map.imagetype=mvt>.

3.2.2 Configuração dos serviços WMS-C no MapCache

Para configurar o serviço WMS-C no MapCache, foram executados os seguintes passos:

1 - Acessar o link <http://localhost/mapcache?request=GetCapabilities> pelo navegador e verificar que o MapCache está online e funcional;

2 - Utilizando o terminal do putty, executar o arquivo ./permissions.sh no diretório do webmapdocker dentro da máquina virtual;

3 - No QGIS, no menu Camada - Adicionar Camada - WMS/WMTS com o URL <http://localhost/mapcache>. O QGIS reconhece automaticamente o WMTS e abre uma aba diferente. Caso não seja aberta outra aba, o QGIS não reconheceu o serviço;

4 - No diretório do webmapdocker, subdiretório config, abrir o arquivo mapcache.xml;

5 - Criar uma nova instância dos elementos <source> e <tileset>, copiando uma instância já presente, e substituir os nomes e valores de acordo e;

6 - Utilizar o comando "docker restart mapcache" no terminal putty para reiniciar o MapCache. Isso deve ser feito toda vez que o arquivo for alterado para que as modificações entrem em efeito.

É importante observar na configuração do MapCache que apenas o grid "g" e o SRC 900913 funcionam adequadamente, e os nomes dos elementos <source> e dos elementos <tileset> devem ser distintos, de forma que um elemento <source> e um elemento <tileset> não podem ter o mesmo nome, pois caso contrário o serviço não funcionará corretamente.

Quaisquer dados multiparte que o QGIS venha a reconhecer como única-parte devem ser pré-processados pelo plugin "Promover a multiparte" antes de serem importados para o PostGIS.

3.2.3 Configuração dos serviços MVT no MapCache

O trecho do arquivo de configuração do MapCache a seguir exemplifica a principal alteração que deverá ser feita nas camadas do BDGEx para viabilizar a utilização de *Vector tiles*:

Os elementos <source> e <tileset> foram copiados e preenchidos de acordo com o modelo <https://github.com/sdlime/mvt-demo/blob/master/mapcache.xml>.

Além disso, caso o elemento <format> de nome ou extensão MVT não esteja presente, o mesmo deve ser criado no mapcache.xml.

```
 <format name="MVT" type="RAW">
  <extension>mvt</extension>
  <mime_type>application/vnd.mapbox-vector-tile</mime_type>
</format>

 <!-- WMS returns Mapbox Vector tiles (.mvt) - for mvt_demo -->
<source name="mvt-demo-mvt_wms" type="wms">
  <http>
    <url>http://mapserver.localhost/cgi-bin/mapserv?</url>
  </http>
  <getmap>
    <params>
      <LAYERS>wma,sna,lakes,major_roads,places</LAYERS>
      <FORMAT>application/vnd.mapbox-vector-tile</FORMAT>
      <MAP>/mapserver/mvt-demo/demo.map</MAP>
    </params>
  </getmap>
</source>

 <tileset name="mvt-demo-mvt">
```

```
    <source>mvt-demo-mvt_wms</source>
    <grid>g</grid>
    <cache>disk</cache>
    <format>MVT</format>
  </tileset>
```

É importante ressaltar que, no MapCache, estão funcionando apenas o grid “g” e o SRC 900913 para o serviço de MVT.

3.2.4 Configuração da visualização em multiescala no MapServer e no MapCache

O mosaico multiescala filtra o que deve ser apresentado e o que deve ser ocultado para o visualizador do cliente, filtrando as feições presentes na camada. Na padronização do BDGEx, o atributo ‘minusablelevel’ é utilizado para filtrar as feições que requerem uma escala maior do que a presente, enquanto o atributo ‘regime’ é utilizado para apresentar visualizações diferentes para rios permanentes e não-permanentes.

No arquivo .map foi declarada a camada da mesma forma que foi declarada anteriormente, porém com algumas mudanças. Foi inserido um elemento CLASS dentro do LAYER para cada regra de visualização diferente. Uma instância disso é explicitada no trecho de código a seguir onde o elemento EXPRESSION descreve a condição para a visualização, enquanto o elemento STYLE descreve a simbologia da camada.

```
CLASS
  EXPRESSION (('[regime]' eq 'Permanente') and ([minusablelevel]>=5000000))
  STYLE
    COLOR "#1089f9"
    WIDTH 1
  END # STYLE
END # CLASS
```

Além disso, o elemento METADATA foi alterado para incluir o sistema de referência EPSG:900913 no argumento `"wms_src"` e foi incluído também o argumento `"gml_include_items" "all"` para que os metadados acompanhem a camada.

```
METADATA
    "wms_srs"     "EPSG:4326 EPSG:900913"
    "gml_include_items" "all"
END
```

O MapCache foi configurado para uma camada multiescala seguindo os mesmos passos utilizados para a camada de escala única.

3.3 Testes de cada etapa do processo de disseminação

De forma a verificar se os componentes do sistema estavam funcionando corretamente, foram feitos testes de conexão utilizando o QGIS como cliente. Foram testadas a conexão direta ao banco PostGIS, e aos serviços WMS, WFS, WMS-C e *Vector tiles*.

3.3.1 PostGIS

O primeiro teste foi a solicitação da camada diretamente ao banco de dados, utilizando a interface PostGIS do QGIS, logo após a camada ter sido alimentada ao banco de dados.

No QGIS, foi acessado o menu Camada - Adicionar Camada - Adicionar camada do PostGIS. Em seguida, foi adicionada uma nova conexão configurada como demonstrado na figura 6.

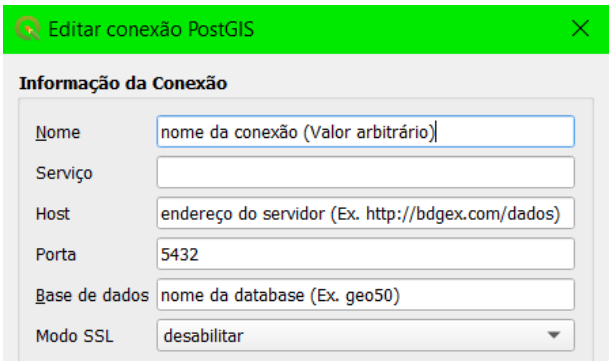


Figura 6 – Configuração da conexão PostGIS

Depois, foi pressionado o botão "Conectar". Na mesma janela, apareceram as camadas carregadas no banco de dados, como mostrado na figura 7.

Por fim, foi selecionada uma camada e pressionado o botão "Adicionar".

3.3.2 WMS

Para testar se o serviço WMS estava funcionando, foi acessado o menu do QGIS Camada - Adicionar Camada - Adicionar Camada WMS/WMTS. Foi criada uma nova conexão configurada de acordo com a figura 8; foram pressionados os botões "OK" e "Conectar".

O servidor apresentou as camadas disponíveis, como mostrado na figura 9; então foi selecionada uma camada e pressionado o botão "carregar".

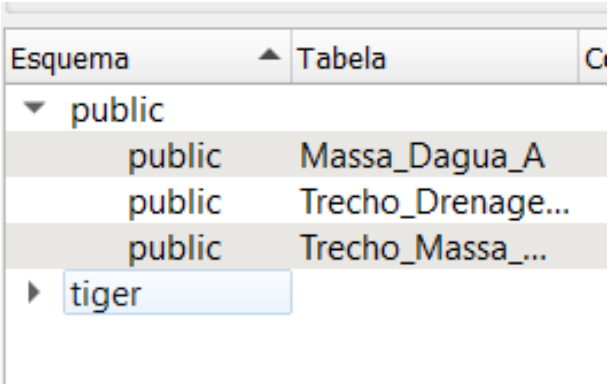


Figura 7 – Camadas identificadas no banco de dados conectado

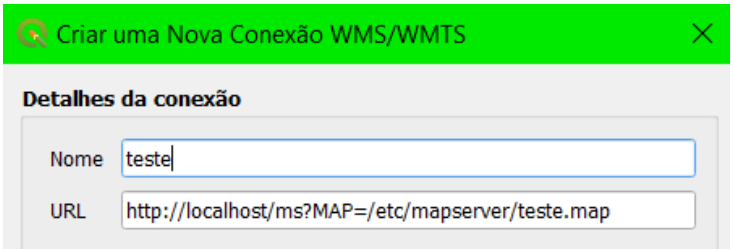


Figura 8 – Configuração da conexão

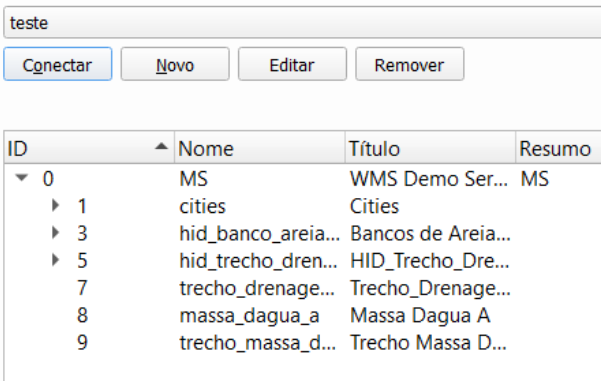


Figura 9 – Camadas disponíveis no servidor para WMS

3.3.3 WMS-C

Para testar se o cache do serviço WMS estava funcionando, no mesmo menu onde foi configurada a conexão WMS, foi criada uma nova conexão configurada de acordo com a figura 10.

Foi pressionado “Conectar” e o servidor retornou as camadas disponíveis em cache, conforme a figura 11. Foi selecionado a camada e pressionado “Adicionar”.

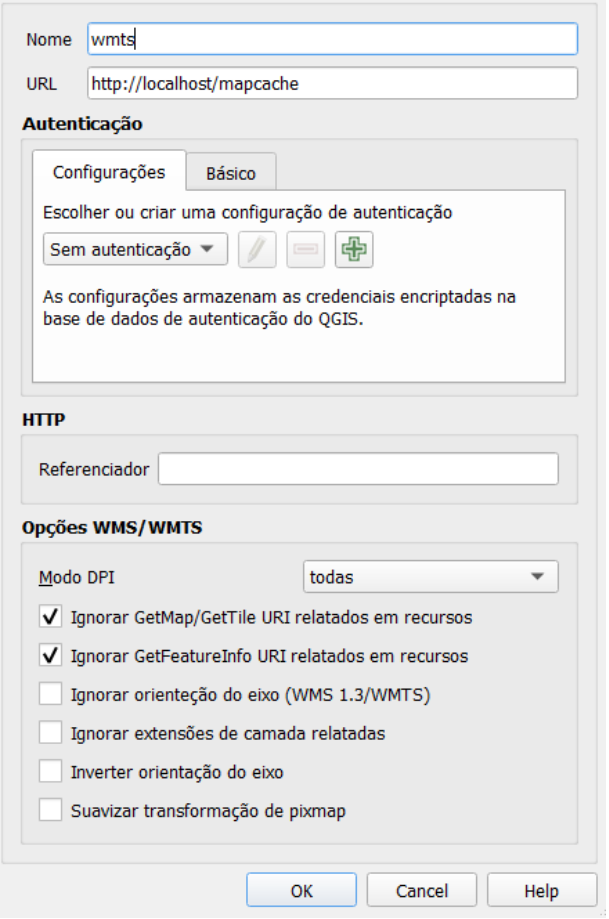
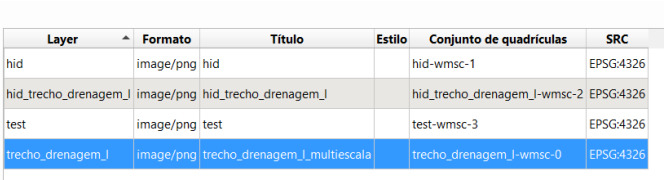


Figura 10 – Configuração da conexão no WMS-C



Layer	Formato	Título	Estilo	Conjunto de quadrículas	SRC
hid	image/png	hid		hid-wmsc-1	EPSG:4326
hid_trecho_drenagem_l	image/png	hid_trecho_drenagem_l		hid_trecho_drenagem_l-wmsc-2	EPSG:4326
test	image/png	test		test-wmsc-3	EPSG:4326
trecho_drenagem_l	image/png	trecho_drenagem_l_multiescala		trecho_drenagem_l-wmsc-0	EPSG:4326

Figura 11 – Camadas disponíveis para WMS-C

3.3.4 MVT - MapServer

Primeiramente, utilizamos os seguinte URL no navegador para obter as capacidades do servidor e verificar que constavam as capacidades do serviço de mosaico vetorial:

<http://localhost:80/ms?MAP=/etc/mapserver/teste.map&request=GetCapabilities&service=WMS>

Foram localizados os seguintes elementos em Service - Capabilities - GetMap, conforme mostra a figura 12, confirmando que o servidor possuía as capacidades desejadas:

```
<Format>application/vnd.mapbox-vector-tile</Format>
<Format>application/x-protobuf</Format>
```

```
  </GetCapabilities>
▼<GetMap>
    <Format>image/png</Format>
    <Format>image/jpeg</Format>
    <Format>image/png; mode=8bit</Format>
    <Format>image/vnd.jpeg-png</Format>
    <Format>image/vnd.jpeg-png8</Format>
    <Format>application/x-pdf</Format>
    <Format>image/svg+xml</Format>
    <Format>image/tiff</Format>
    <Format>application/vnd.google-earth.kml+xml</Format>
    <Format>application/vnd.google-earth.kmz</Format>
    <Format>application/vnd.mapbox-vector-tile</Format>
    <Format>application/x-protobuf</Format>
    <Format>application/json</Format>
  ▼<DCPType>
    ▼<HTTP>
      ▼<Get>
          <OnlineResource xmlns:xlink="http://www.w3.org/199
```


Figura 12 – Capacidades disponíveis

No QGIS, foi adicionada uma camada através do menu Camada - Adicionar Camada de Mosaico Vetorial. Foi criada uma nova conexão genérica e preenchida com o seguinte URL, além de um nome, conforme mostra a figura 13:

<http://localhost:80/ms?MAP=/etc/mapserver/teste.map&mode=tile&tilemode=gmap&tile={x}+{y}+{z}&layers=all&map.imagetype=mvt>

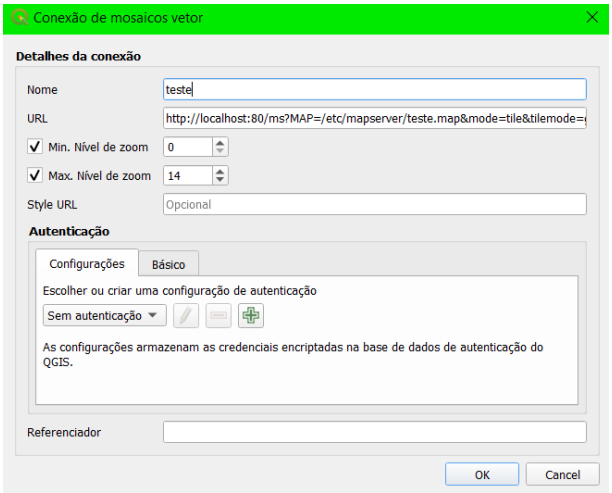


Figura 13 – Configuração da conexão MVT do MapServer

Em seguida, foi pressionado o botão “Adicionar”.

3.3.5 MVT - MapCache

No QGIS, foi adicionada uma camada através do menu Camada - Adicionar Camada de Mosaico Vetorial. Foi criada uma nova conexão genérica e preenchida o com o URL <http://localhost/mapcache/gmaps/trecho_drenagem_l_g@g/{z}/{x}/{y}.mvt> e com um nome conveniente, conforme a figura 14.

Em seguida, foi pressionado o botão “Adicionar”.

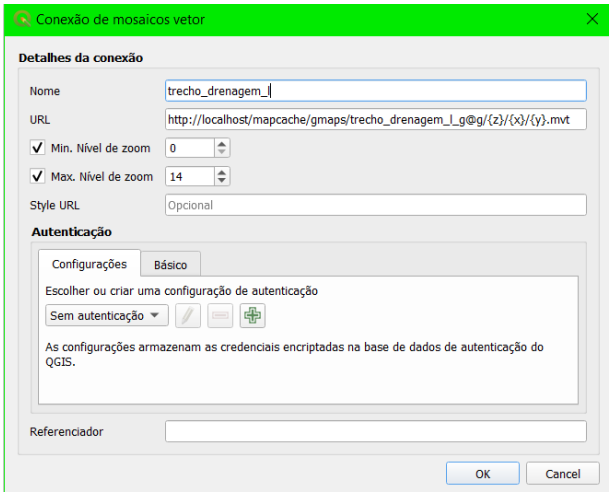


Figura 14 – Configuração da conexão MVT do MapCache

3.4 Experimentos utilizando a estrutura de mosaico multi-escala do BDGEx

Os experimentos dos quais tratamos na seção anterior foram repetidos para uma camada em multi-escala, de forma que, em cada um deles, foi adicionada a verificação de se o dado multi-escala foi transmitido com as regras de visualização adequadas. Foram utilizados como clientes o QGIS e um aplicativo NodeJS que utiliza a biblioteca OpenLayers.

Os dados foram obtidos por meio de uma requisição WFS ao serviço de mosaico multi-escala do BDGEx, utilizando o comando CURL. Foram selecionadas as classes Trecho_Drenagem_L, Massa_Dagua_A e Trecho_Massa_Dagua_A. A requisição WFS utilizada foi:

```
curl -o multiscale.gml
↪  "http://bdgex.eb.mil.br/wfs?service=wfs&request=GetFeature&typeN⌋
↪  ame=Trecho_Drenagem_L,Massa_Dagua_A,Trecho_Massa_Dagua_A&version⌋
↪  =1.0.0&resultType=results&bbox=-52,-31,-50.0,-29"
```

3.4.1 QGIS

Os dados foram inseridos em um banco de dados PostgreSQL/PostGIS por meio do DB Manager do QGIS. Foram utilizadas as funcionalidades do menu Camada para visualizar cada etapa da disseminação, desde o acesso direto ao PostgreSQL até a visualização do MVT fornecido pelo MapCache, seguindo os mesmos procedimentos descritos na seção 3.3.

3.4.2 OpenLayers

Para verificar que o funcionamento correto da solução não depende do cliente QGIS, foi testado o acesso utilizando a biblioteca OpenLayers, empregada no BDGEx.

O contêiner "webmapclient", existente no WebMapDocker, contém uma aplicação OpenLayers básica que utiliza NodeJS. Esta aplicação foi alterada, utilizando os arquivos disponíveis no endereço <https://openlayers.org/en/latest/examples/mapbox-vector-tiles.html>. O acesso à aplicação OpenLayers foi feito pelo endereço <http://localhost:8001>.

4 RESULTADOS E DISCUSSÃO

4.1 Testes de cada etapa do processo de disseminação

4.1.1 PostGIS

O resultado do teste foi o recebimento e carregamento integral dos dados contidos na camada, conforme mostra a figura 15.

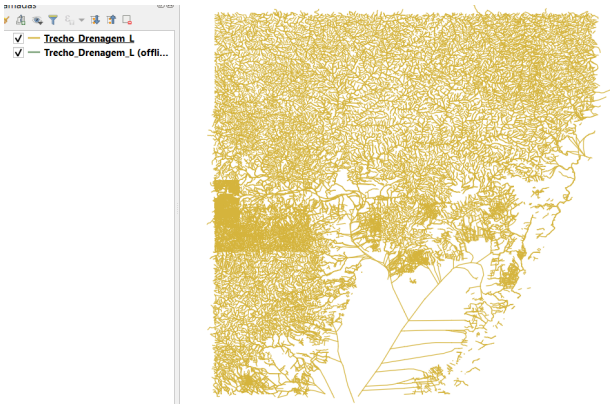


Figura 15 – Resultado do teste do PostGIS

A camada confere com o carregamento manual do shapefile no QGIS, mostrando que os dados foram transmitidos para e do banco de dados com sucesso.

4.1.2 WMS

O resultado do teste foi o recebimento do WMS referente a camada, como mostrado pela figura 16.

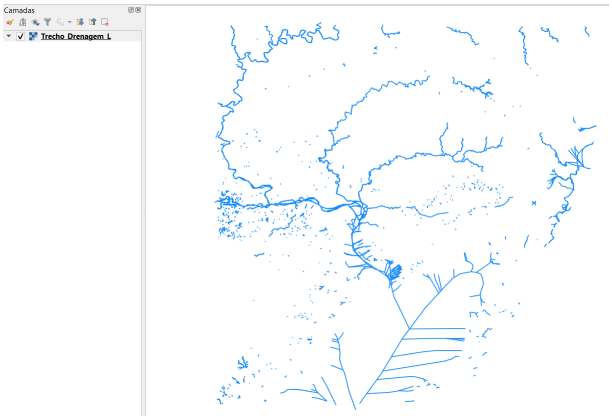


Figura 16 – Resultado do teste do WMS

Com isso, foi comprovado que os dados foram adequadamente transferidos do banco de dados para o contêiner do MapServer; o desenho foi feito corretamente e os dados foram recebidos pelo cliente.

4.1.3 WMS-C

O resultado do teste foi o recebimento da camada em cache, conforme mostra a figura 17. Foi verificado visualmente que a geometria da camada recebida confere com a geometria da camada no WMS sem cache.

Figura 17 – Resultado do teste WMS-C

Isso comprova que o cache e a re-distribuição dos dados ocorreram adequadamente.

4.1.4 MVT - MapServer

O resultado do teste foi o recebimento do mosaico vetorial, conforme mostra a figura 18. Foi verificado visualmente que a geometria da camada recebida confere com a geometria da camada no serviço WMS (com resposta em PNG), e que nas feições constavam os todos os campos da camada original, em formato Shapefile.

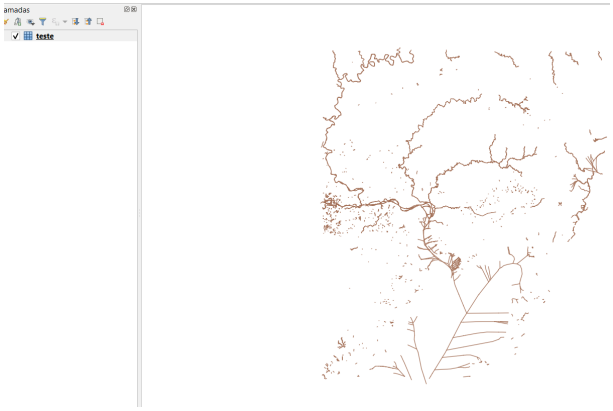


Figura 18 – Resultado do teste do MVT do MapServer

O mosaico é compatível com o shapefile inserido, portanto confirmando que o mesmo foi gerado adequadamente.

4.1.5 MVT - MapCache

Foram feitos testes com duas configurações de sistemas de coordenadas: EPSG:4326 e EPSG:3857. O BDGEx utiliza o EPSG:4326, porém nos testes o QGIS não reconheceu o sistema de coordenadas da camada disponibilizada pelo MapCache. Desta forma, os *tiles* foram representados em regiões erradas do espaço geográfico. Foi identificado que este problema ocorre pois o padrão MVT não transmite informações de sistemas de coordenadas. Este problema geralmente é abordado nos padrões OGC por meio da requisição GetCapabilities, que não existe no MVT.

O resultado do teste com EPSG:3857 não apresentou problemas, por ser o sistema de coordenadas adotado no padrão MVT, conforme mostra a figura 19. Foi verificado visualmente que a geometria da camada recebida confere com a geometria da camada em disco, e que nas feições constavam os todos os campos da camada original.

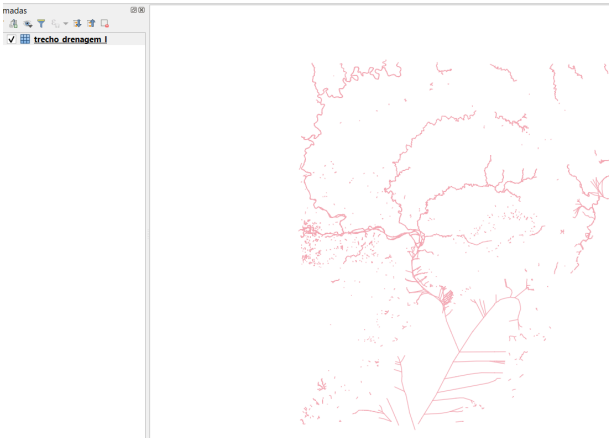


Figura 19 – Resultado do teste do MVT do MapCache

Comparando com o recebido do servidor, verifica-se que o cache foi feito adequadamente.

4.2 Experimentos utilizando a estrutura de mosaico multi-escala do BDGEx

4.2.1 QGIS

O QGIS utiliza o sistema de referência de coordenadas EPSG:3857 (WGS 84 - Pseudo Mercator), pois esse é o SRC do serviço padrão do MVT atualmente. O padrão MVT não transmite o SRC para o cliente, pois não possui uma requisição como a GetCapabilities,

existente no WMS-C. A OGC possui uma API de *Vector Tiles* em desenvolvimento, e nela o mosaico vetorial declara o sistema de referência. Quando esta entrar em operação, o uso dela corrigirá esta limitação.

Além disso, o estilo deve ser publicado separadamente no padrão MVT. O BDGEx não tem estilos neste padrão atualmente, havendo a necessidade de conversão do estilo do MapServer para MapBox.

4.2.2 OpenLayers

O OpenLayers, da mesma forma que o QGIS, utiliza o sistema de referência EPSG:3857 e deverá ter essa limitação corrigida com a adoção da API de Mosaicos Vetoriais da OGC.

O resultado do experimento em questão foi a exibição da camada, conforme mostra a figura 20, comprovando que a solução funciona como esperado em clientes diferentes do QGIS.

Figura 20 – Camada apresentada no Open Layers

5 CONCLUSÃO

Nosso trabalho atingiu o objetivo de gerar um protótipo que pode ser implementado nos serviços do BDGEx para utilização do mosaico vetorial em multiescala, melhorando a visualização em pequenas escalas dos dados vetoriais. Tal implementação pode ser feita através de um servidor dedicado, com as atualizações dos arquivos .map e .xml feitas automaticamente através de *scripts*.

Nos experimentos com dados do mosaico multi-escala do BDGEx, foi identificado que o padrão MVT permite filtrar geometrias e outros atributos no cliente, além de permitir formas alternativas de apresentar os dados. Isto pode ser utilizado para reduzir a densidade de informações apresentadas na tela conforme a necessidade dos usuários.

O protótipo é viável, porém com restrições. Atualmente, o sistema de referência de coordenadas dos mosaicos vetoriais é obrigatoriamente o EPSG:3857, e o mosaico vetorial não herda o estilo estabelecido no MapServer. Assim a reprojeção do mosaico e estilização dos vetores ficariam ambas a cargo do cliente. Nossa projeção é de que, com a inclusão futura da API de Mosaicos Vetoriais da OGC, a restrição referente ao sistema de referência será corrigida.

Finalizando, nossos testes comprovam que este protótipo imbuirá os serviços BDGEx de mais uma funcionalidade, além de contribuir para seu processo de modernização, antecipando a implementação desta funcionalidade ou similar. Tal contribuição, acreditamos, contribuirá positivamente para a transmissão de geoinformação por parte do BDGEx para seus diversos usuários, ajudando-o a cumprir sua atividade-fim.

REFERÊNCIAS

DOCKER. *What is a Container? | Docker*. 2021. Disponível em: <https://www.docker.com/resources/what-container/>.

DOCKER. *Docker: Accelerated, Containerized Application Development*. 2022. Disponível em: <https://www.docker.com/>.

DSG. *BDGEx | Login*. Diretoria de Serviço Geográfico do Exército Brasileiro, 2023. Disponível em: <https://bdgex.eb.mil.br/bdgexapp>.

DSG. *Geoportal - Serviços OGC*. Diretoria de Serviço Geográfico do Exército Brasileiro, 2023. Disponível em: <https://bdgex.eb.mil.br/portal/index.php?option=com_content&view=article&id=77&Itemid=221&lang=pt>.

GAFFURI, J. *Toward Web Mapping with Vector Data*. 2012. 87-101 p.

GMBH. *EPSG.io: Coordinate Systems Worldwide*. Klokan Technologies, 2023. Disponível em: <https://epsg.io>.

MAPBOX. *Mapbox Vector Tile Specification*. Mapbox, 2023. Original-date: 2014-04-12T19:30:28Z. Disponível em: <https://github.com/mapbox/vector-tile-spec>.

MAPSERVER. *OGC API : Features — MapServer 8.0.1 documentation*. 2023. Disponível em: <https://mapserver.org/ogc/ogc_api.html#ogcapi>.

MARTINELLI, L.; ROTH, M. *Updatable Vector Tiles from OpenStreetMap*. Tese (Doutorado) — HSR Hochschule für Technik Rapperswil, 2016.

MEDEIROS, A. M. L. de. *Web Mapping*. 2010. Disponível em: <www.infoescola.com/geografia/web-mapping/>.

NEBERT, D.; VOGES, U.; BIGAGLI, L. Implementation Standard, *OGC® Catalogue Services 3.0 - General Model*. 2016. Publisher: Open Geospatial Consortium. Disponível em: <https://docs.ogc.org/is/12-168r6/12-168r6.html>.

OGC. *Corrigendum for the OpenGIS® Web Feature Service (WFS) implementation specification 04-095*. Open Geospatial Consortium Inc., 2006. Disponível em: <https://portal.ogc.org/files/?artifact_id=14145>.

OGC. *OpenGIS® Web Map Server Implementation Specification*. Open Geospatial Consortium Inc., 2006. Disponível em: <https://portal.ogc.org/files/?artifact_id=14416>.

OGC. *OpenGIS® Web Map Tile Service Implementation Standard*. Open Geospatial Consortium Inc., 2010. Disponível em: <https://portal.ogc.org/files/?artifact_id=35326>.

OGC. *OGC API - Tiles - Part 1: Core*. Open Geospatial Consortium Inc., 2023. Disponível em: <https://docs.ogc.org/is/20-057/20-057.html#toc72>.

OPENLAYERS. *OpenLayers v7.3.0 API - Class: VectorTile*. 2023. Disponível em: <https://openlayers.org/en/latest/apidoc/module-ol_source_VectorTile-VectorTile.html>.

QGIS. *22. working with OGC / ISO protocols*. 2022. Disponível em: <https://docs.qgis.org/3.28/pt_BR/docs/user_manual/working_with_ogc/ogc_client_support.html>.

APÊNDICE A – PROCEDIMENTOS DE INSTALAÇÃO

A.1 Kubuntu

Aqui será descrito o procedimento para download, instalação e configuração do Kubuntu, usado para todo o projeto.

1. Baixe o Kubuntu Desktop 22.04.2 LTS através do link <https://cdimage.ubuntu.com/kubuntu/releases/22.04.2/release/kubuntu-22.04.2-desktop-amd64.iso> ou através do site do Kubuntu Desktop.
2. Aguarde o download concluir.

A.2 Virtual Box

Aqui será descrito o procedimento para download, instalação e configuração do Virtual Box, usado para montar a máquina virtual do Linux.

Primeiramente, para download e instalação:

1. Acesse o site <https://www.virtualbox.org/wiki/Downloads>;
2. Escolha a versão adequada ao sistema operacional;
3. Aguarde o download concluir;
4. Execute o instalador;
5. Clique em “next” até aparecer “install”. Clique em “install”;
6. Marque para abrir o aplicativo e clique em “finish”.

Agora o Virtual Box está baixado e instalado. O próximo passo é a configuração:

1. Clique em “Novo”.
2. Configure a máquina virtual, selecione o Kubuntu Desktop que foi baixado como Imagem.
3. Inicie a máquina virtual e proceda com a instalação do Kubuntu Desktop. Anote o IP dá VM.

4. Clique em “Máquina”, “Configurações”, “Rede”, “Redirecionamento de Portas”.
5. Acrescente uma nova regra de direcionamento para as portas 22, 5432 e 80.
6. Preencha os campos “Porta do Hospedeiro” e “Porta do Convidado” com a porta a ser remapeada.
7. Autorize acesso ao firewall, caso seja socilitado.
8. Abra o Konsole e digitar o comando `"sudo apt-get install openssh-server"`.
9. Digite o comando `"ps -e"` e verificar a presença do “sshd” na lista.

Para abrir uma máquina virtual já existente, use o “Acrescentar”.

A.3 Interfaces de cliente remoto

A.3.1 Windows

Faça o download do Putty através da URL <https://putty.org/> e do WinSCP através da URL <https://winscp.net>. Depois de instalar o os dois, abra o Putty. Na janela que abrir, em “HostName (or IP address)” preencha “localhost” e clique em “Open”. Se aparecer uma janela avisando que a porta não está em cache, clique em “Aceitar” e faça o Login. Com isso o Putty estará pronto para uso. Para o WinSCP, abra-o, preencha “localhost”, o Usuário e a Senha, e clique em “Login” e “Sim”.

A.3.2 Linux

No navegador de arquivos Dolphin, preencha com a URL <sftp://localhost:22/>. No terminal Konsole digite: ssh localhost -p 22.

A.4 Docker

Siga os passos no link: <https://docs.docker.com/engine/install/ubuntu/>.

Adicione o usuário ao docker group, conforme o link: <https://docs.docker.com/engine/install/linux-postinstall/>.

Após a configuração, feche e reabra o cliente para atualizar as permissões, e verifique com o comando `"docker run hello-world"` se as mesmas foram atualizadas.

Execute os seguintes comandos:

```
sudo apt-get install git docker-compose
```

```
git clone https://github.com/mauriciodev/webmapdocker.git
cd webmapdocker
docker-compose up
```