

MVT Generator – Gerador de MVT a partir de dados geográfico complexos

INTRODUÇÃO AO QUE SÃO TILES E O FORMATO MVT

Geralmente, dados geográficos são muito grandes, o que dificulta sua análise e exibição em ambiente web. Uma maneira de resolver esse problema é carregar somente os dados da parte do mapa que está sendo exibida. Para isso, o dado geográfico é recortado, dependendo do nível de zoom, em diversos quadrados (*tiles*). Assim, quando estamos em um zoom próximo (ex: zoom 16), veremos somente um pedaço do mapa, sendo necessário carregar somente os tiles(quadrados) deste pedaço visível. Já quando estamos com um zoom muito afastado (ex: zoom 0), veremos em poucos *tiles* quase o mapa inteiro.

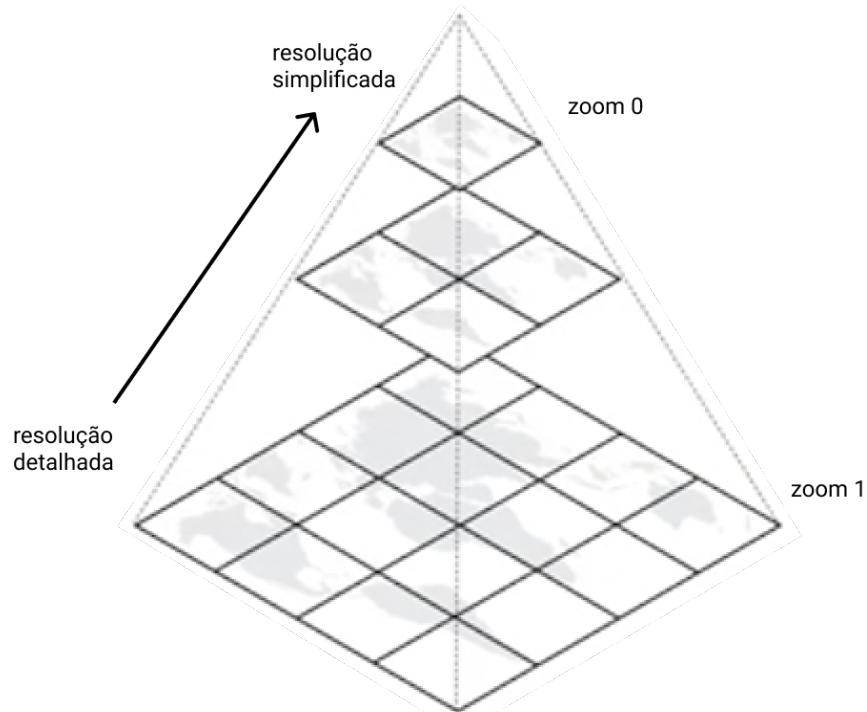


Figura 01 - Os dados geográficos são recortados em diversos quadrados (*tiles*), dependendo do zoom.

Serviços que fornecem, em tiles, os mapas já são implementações utilizadas há muito tempo, porém a maioria ainda utiliza/fornece o tile em formatos bitmaps (imagens .png ou .jpg) geralmente com a resolução de 256 pixels por *tile*. Mas, afinal, o que isso quer dizer?

Que cada *tile* é uma imagem de 256 pixels de largura por 256 pixels de altura. E essa imagem é gerada (renderizada) no servidor, já com sua representação visual. Tiles como bitmap têm a vantagem de serem bem leves em sua transmissão via web, porém tem a

desvantagem de não possibilitar interações com as geometrias, já que tudo o que se vê na tela são imagens, como uma fotografia.

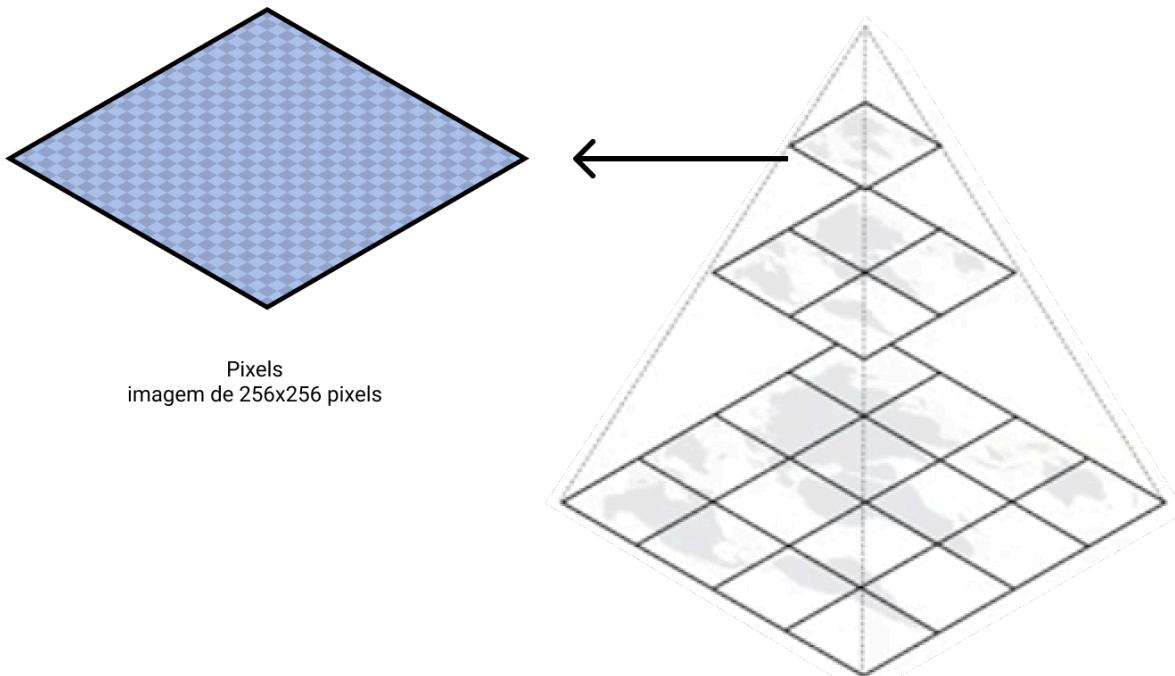


Figura 02 - Cada tile é uma imagem de 256x256 pixel, que já vem "pintada" do servidor.

Já no formato MVT, o que vem nesses tiles são representações vetoriais das geometrias (representações matemáticas) das formas que existem em cada *tile*, e não imagens.

Assim o dado transmitido é somente as formas geométricas e seus atributos, e quem renderiza (pinta sua aparência) é o navegador onde o mapa está sendo exibido.

Isso traz enormes vantagens, pois permite interagir com as formas que são exibidas nos mapas. Também permite analisar a forma, determinar sua aparência, mudá-la conforme suas propriedades, conforme o mouse interage com a forma etc.

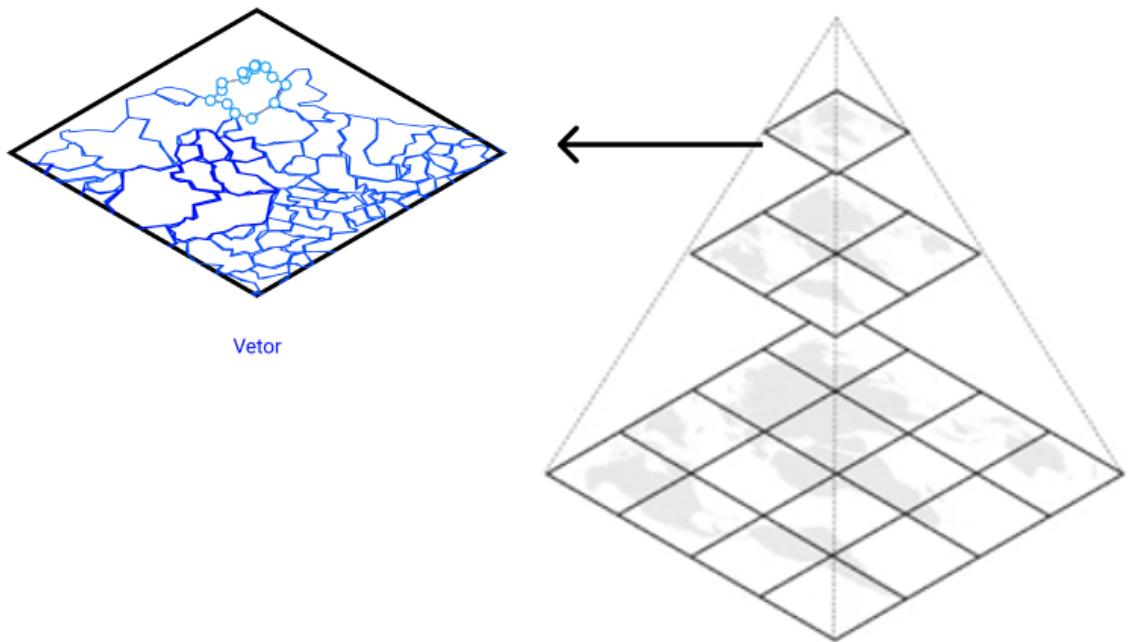


Figura 03 - Cada tile é uma representação matemática da forma, sendo o navegador responsável por interpretar essas formas e renderiza-las ("pintar sua aparência").

Mas esse formato representa um desafio. Por exemplo, quando estamos no zoom 0, e todo um mapa vem representado em poucos tiles? Se o tile vem em formato bitmap não tem problema, pois ele vai ser simplesmente algumas imagens de 256x256 pixel com a representação bem pequena do mapa. Mas e como vetor? Todos os pontos e linhas que compõem o mapa inteiro são exibidos em um tile. Sem simplificação das formas, todos os pontos que formam uma geometria, que pode ser enorme, serão enviados ao navegador. Por tanto, no formato mvt, os pontos, as linhas, os polígonos precisam ser simplificados de maneira que em cada nível de zoom a geometria representa a forma, e sem as geometrias que não são visíveis aos olhos. Por exemplo, num zoom muito distante, a geometria que representa uma rua não vai ser visível, então ela não precisa contar nesse tile.

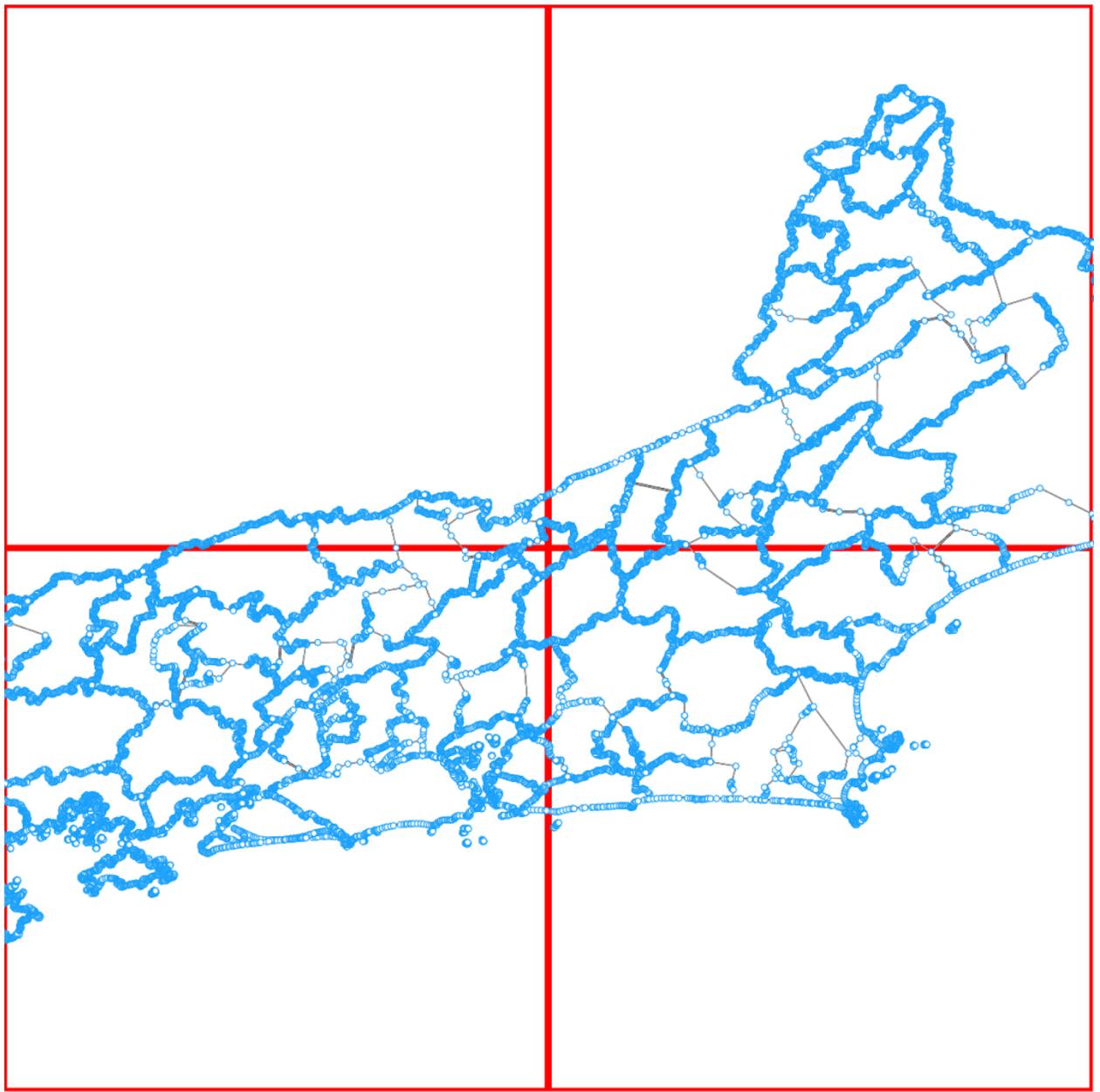


Figura 4 - Em um zoom distante, o estado do Rio de Janeiro aparece quase inteiro. Se não otimizar, todos os pontos da sua geometria são carregados, o que deixaria muito lento o carregamento dos dados.

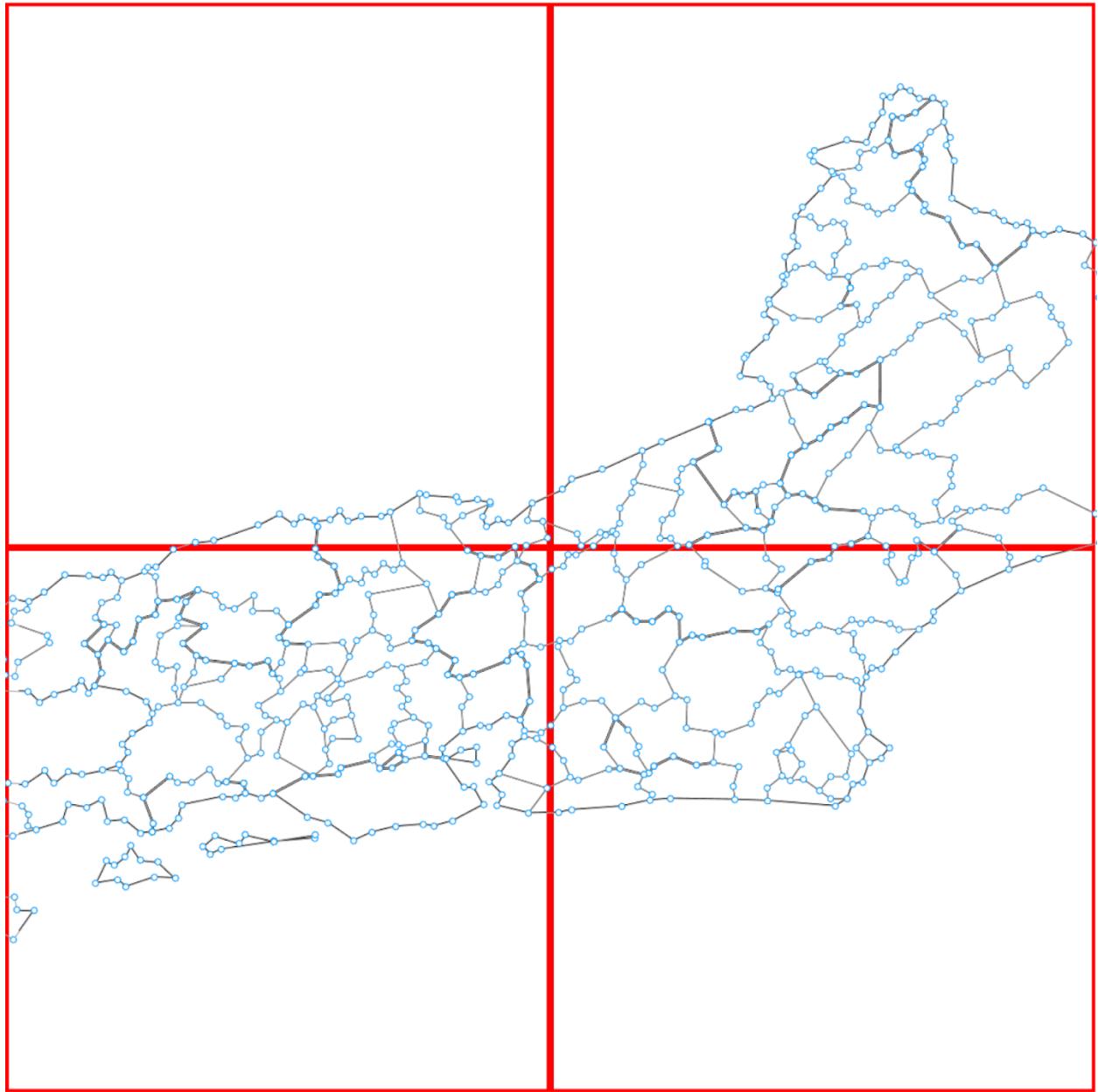


Figura 5 - Aqui a geometria do estado do Rio de Janeiro simplificada, que no final representa bem sua aparência, sem a necessidade de carregar todos os pontos, aumentando a performance.

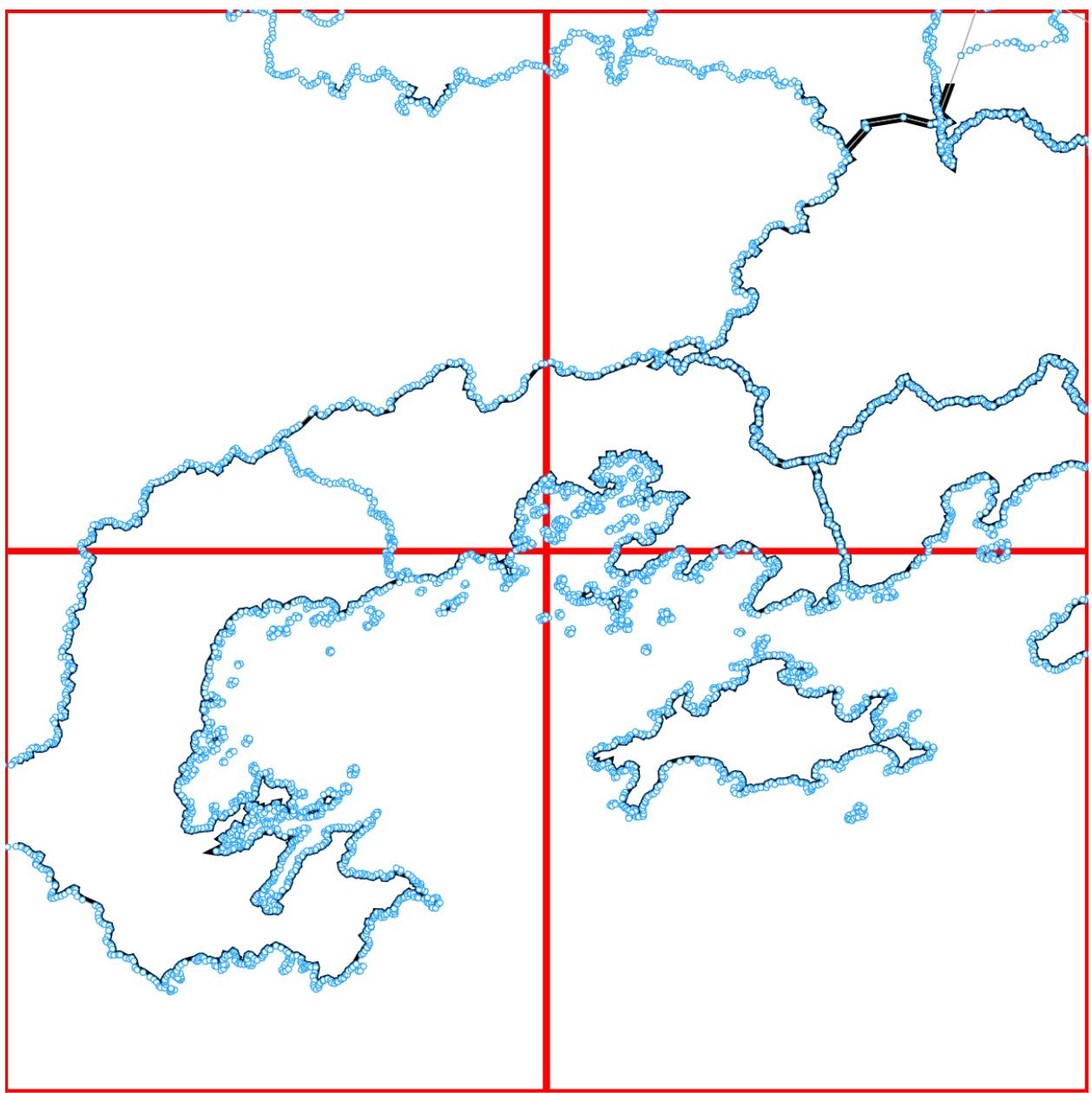


Figura 6 - A imagem do estado do Rio de Janeiro num zoom mais próximo. Aqui podemos carregar a geometria mais detalhada, pois somente um pequeno pedaço do estado vai ser carregado.

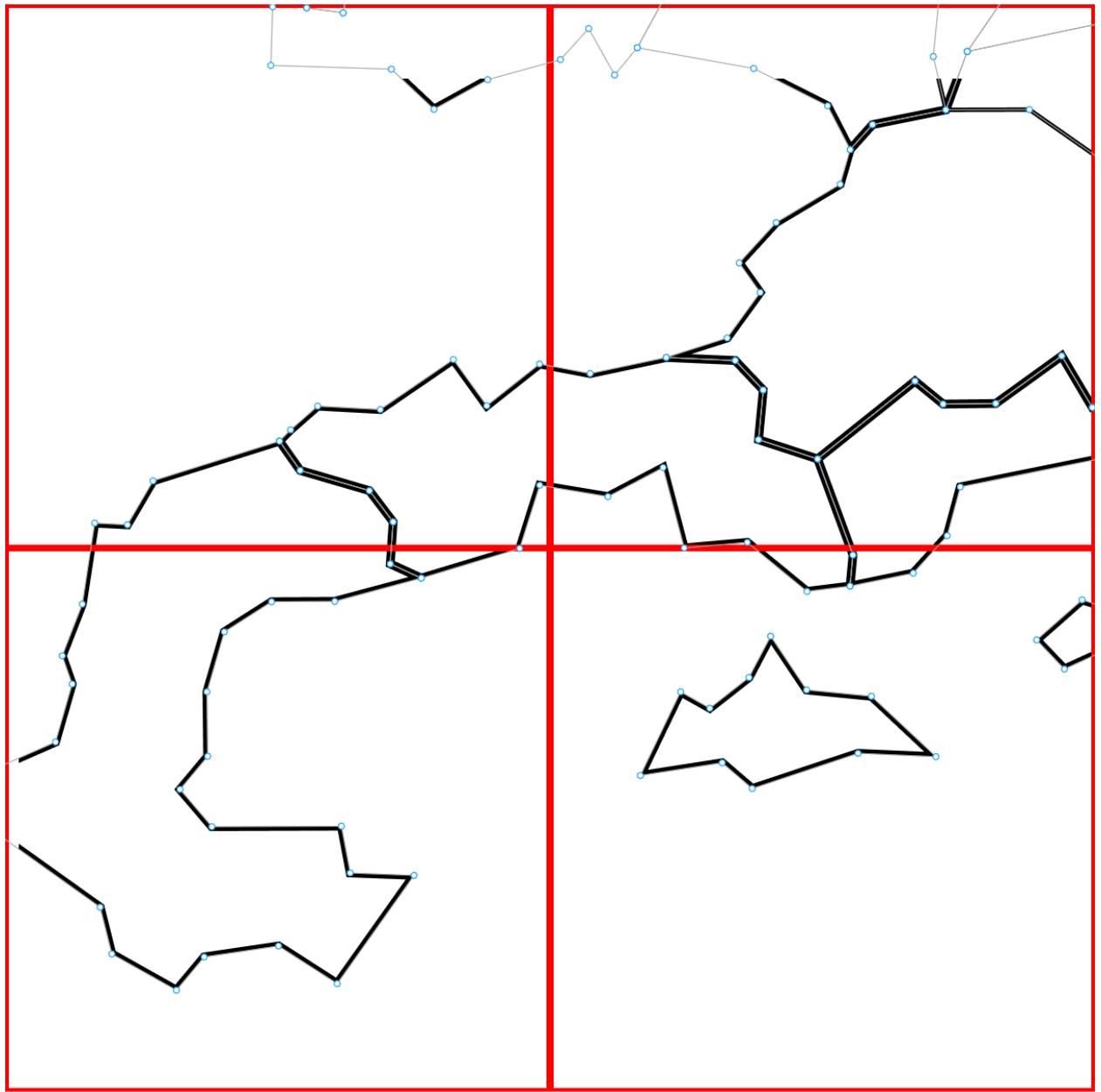


Figura 7 - Se usarmos uma geometria muito simplificada nesse nível de zoom, perdemos a definição da geometria, apesar da menor transmissão de dados.

Então, o maior desafio do mvt está em achar a forma otimizada para cada nível de zoom, que tenha o menor número de pontos possíveis para a sua representação geométrica sem perder sua forma.

AVALIAÇÃO DA SOLUÇÃO ATUAL

Na solução atual é utilizado o formato mvt para exibição dos mapas. Isso permite a grande interação existente na solução, assim como a fácil definição da aparência das camadas. Essa possibilidade de interação é muito importante, pois no decorrer desse projeto capacidade como a mudança da aparência das camadas pelo usuário vão ser implementadas.

Mas o maior problema enfrentado atualmente na utilização do sig-web é a performance da exibição de algumas camadas, principalmente, as denominadas geo-biofísicas.

Algumas soluções alternativas foram implementadas a fim de contornar esses problemas, como algumas camadas sendo exibidas somente a partir de um determinado zoom. Porém, com isso, o usuário perde o panorama da visualização nacional dessas camada. Também para algumas camadas, foi utilizado um serviço externo, que fornecem os tiles como bitmap. Mas esses serviços se encontram instáveis, às vezes, fugindo o seu funcionamento do controle do CGEE, além de não permitir a interação e as configurações de renderização (aparência) das camadas.

Então, para o melhor funcionamento, precisamos de uma solução com as seguintes características:

- Possibilidade de simplificação de dados geográficos que possuem uma quantidade muito grande de geometrias sendo exibidas em determinado tile.
- Limitar o tamanho máximo que cada tile em formato vetorial pode ter em kilobytes.
- Fornecimento dos tiles pelo servidor ao navegador da forma mais rápida possível.

Figura 8 - características necessárias para a solução.

AVALIAÇÃO DA FERRAMENTAS DISPONÍVEIS

A seguir serão analisadas algumas soluções e ferramentas existentes para a geração de MVT, para verificar qual que melhor atende as características mínimas necessárias (figura 8).

Para testes das ferramentas, foi utilizada a camada geo-biofísica de **uso e cobertura da terra**. Essa camada, gerada e disponibilizada pelo [MapBiomass](#) em seu formato original é composta por um total de 359.017 polígonos, totalizando 4.737.733 vértices (pontos).

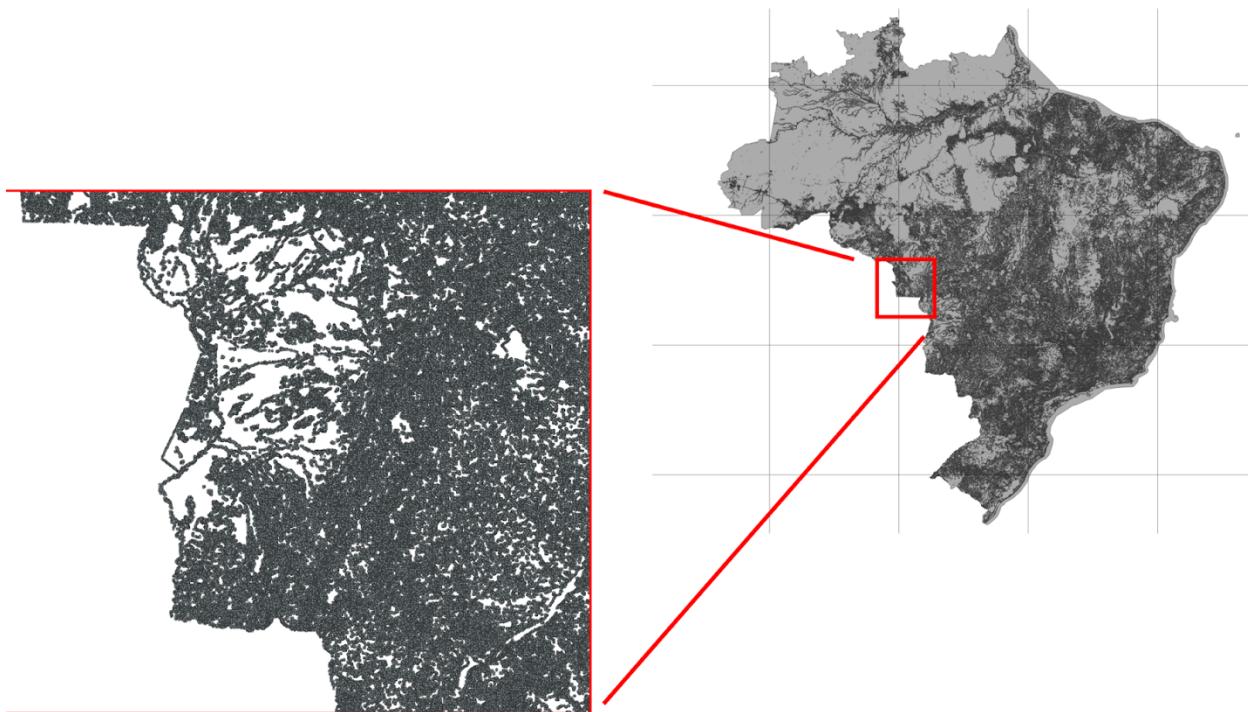
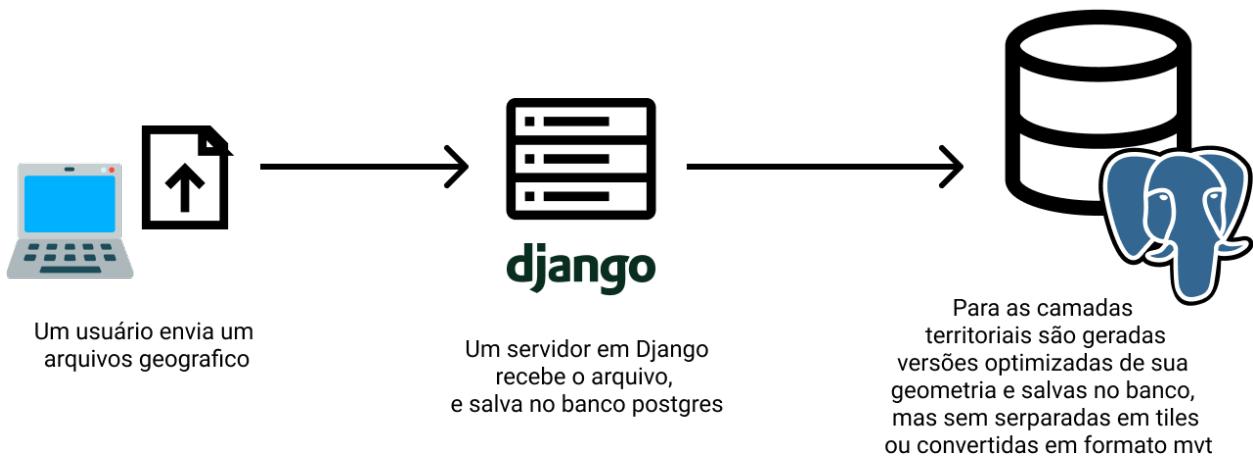


Figura 8 - Camada original de uso e cobertura da terra, no detalhamento demonstrando a quantidade de vértices existentes..

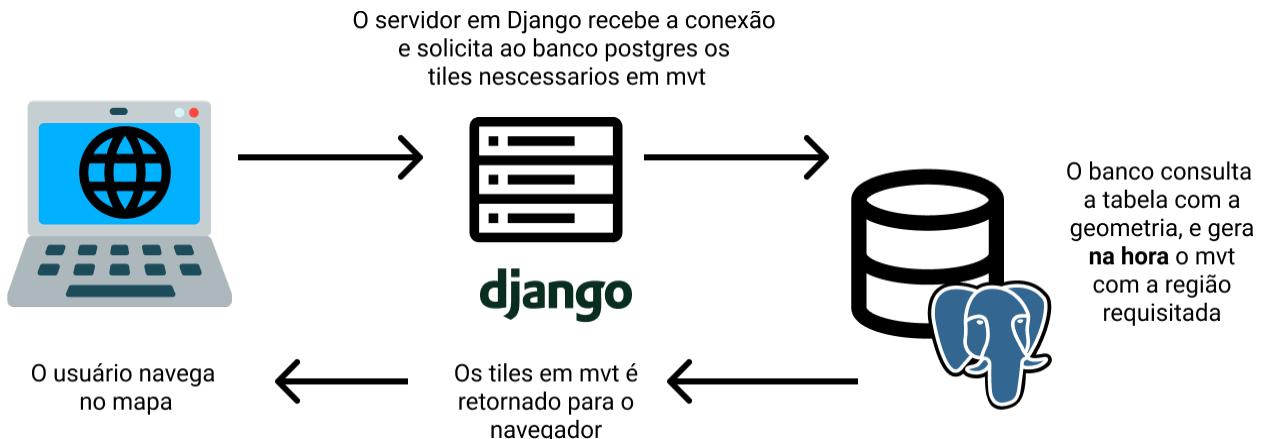
POSTGIS

Muitas ferramenta que fornece camadas em formato MVT, com a opção de pré processamento de simplificação das geometrias. A ferramenta utiliza o banco de dados PostgreSQL/Postgis. O seu funcionamento é ilustrado abaixo.

Para a criação de uma camada:



Para a requisição de uma camada:



As vantagens da ferramenta são:

- Geração de camadas territoriais a partir de agregações de algumas regiões, por exemplo, gerar a camada de estados a partir das camadas de municípios.
- Manutenção da topologia das camadas regionais.

As desvantagens são:

- A geração dos tiles e mvt ocorrem **na hora da requisição**, o que diminui a performance.
- Depende de infraestrutura mais complexa, com a dependência do banco de dados PostgreSQL/Postgis mais atuais para seu funcionamento.

Um dos problemas é que a geração do formato final, o MVT, ocorre na hora da requisição. Isso quer dizer que para cada tile requisitado, existe um processamento no banco para a geração do tile. Apesar de ser razoavelmente rápido, para regiões complexas, com muitas geometrias, isso aumenta consideravelmente o tempo de execução.

Outro apontamento é a utilização de Django, que é uma ótima framework desenvolvida em python, porém exige mais recurso de máquina, pois responde a requisição de maneira bloqueada (quando alguém faz uma requisição ao servidor, um processo é iniciado e fica bloqueando o processamento até o seu retorno. Mas vários processamentos podem ser iniciados, só exigem um hardware mais potente se houver muitas demandas).

TIPPECANOE

Tippecanoe é uma ferramenta para conversão de arquivos GeoJSON em MVT. Desenvolvida pela Mapbox, a mesma que criou a especificação do formato MVT. Ela é capaz de gerar um tipo de arquivo denominado MBTILES. Esses são formatos de arquivos com especificação do SQLite, contendo todos os tiles necessários pela geometria nos diversos níveis de zoom.

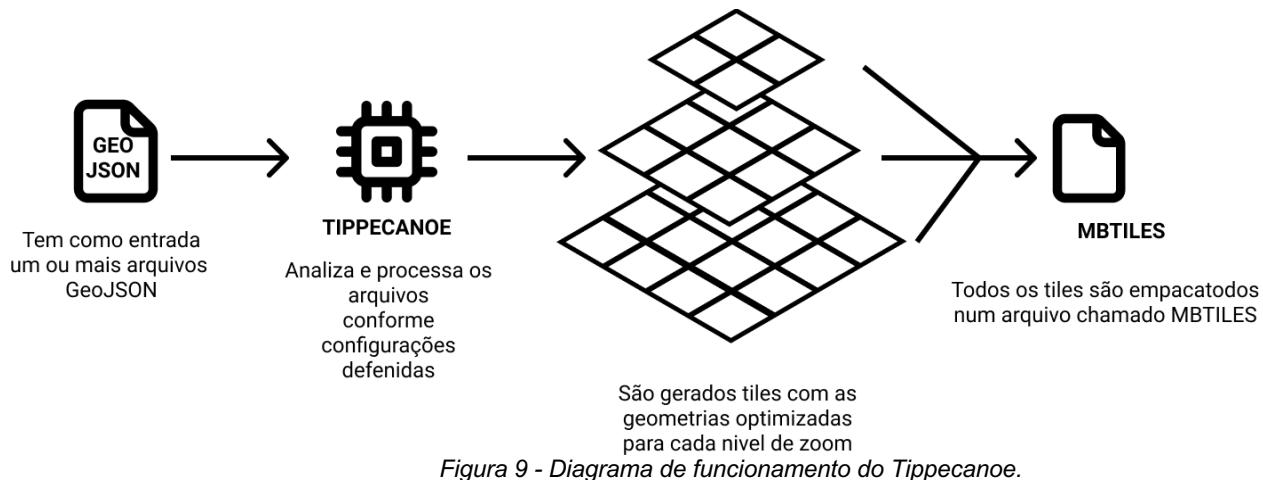


Figura 9 - Diagrama de funcionamento do Tippecanoe.

As vantagens da ferramenta são:

- Simplificação e conversão muito rápida, por ser uma ferramenta em C++ e rodar em paralelismo se os GeoJSON são fornecidos seguindo a especificação [RFC8142](#).
- Geração e armazenamento de todos os tiles em 1 arquivo, o que facilita sua distribuição e consumo.
- Manutenção da topologia das camadas regionais.

As desvantagens são:

- Não é um "fornecedor" de mvt, somente conversor para este formato.
- Em sistemas linux é necessário compilar a ferramenta.

O tippecanoe apresenta uma performance excelente, além de possibilitar várias formas de otimização das geometrias dependendo do nível de zoom, como remoção de geometrias que não vão ser visíveis, agregações de geometrias próximas e que apresentem uma série de propriedades similares, simplificação da geometria até o tile ficar com no máximo um determinado tamanho em kb especificado (o padrão são 512kb). Outra vantagem é o arquivo gerado, que não exige nenhum processamento no servidor quando consumido. Apesar de ser necessário outro serviço para consumo desse arquivo, existem diversas opções disponíveis, por se tratar de um serviço muito simples.

Na conversão das camadas regionais, o tippecanoe teve um desempenho excelente, convertendo em 5 minutos a camada de Municípios, mantendo sua tipologia.

Mas a conversão da camada de uso e cobertura da terra, com os ajustes para simplificação máxima ocorreu em algumas horas e o resultado apresentou alguns problemas.

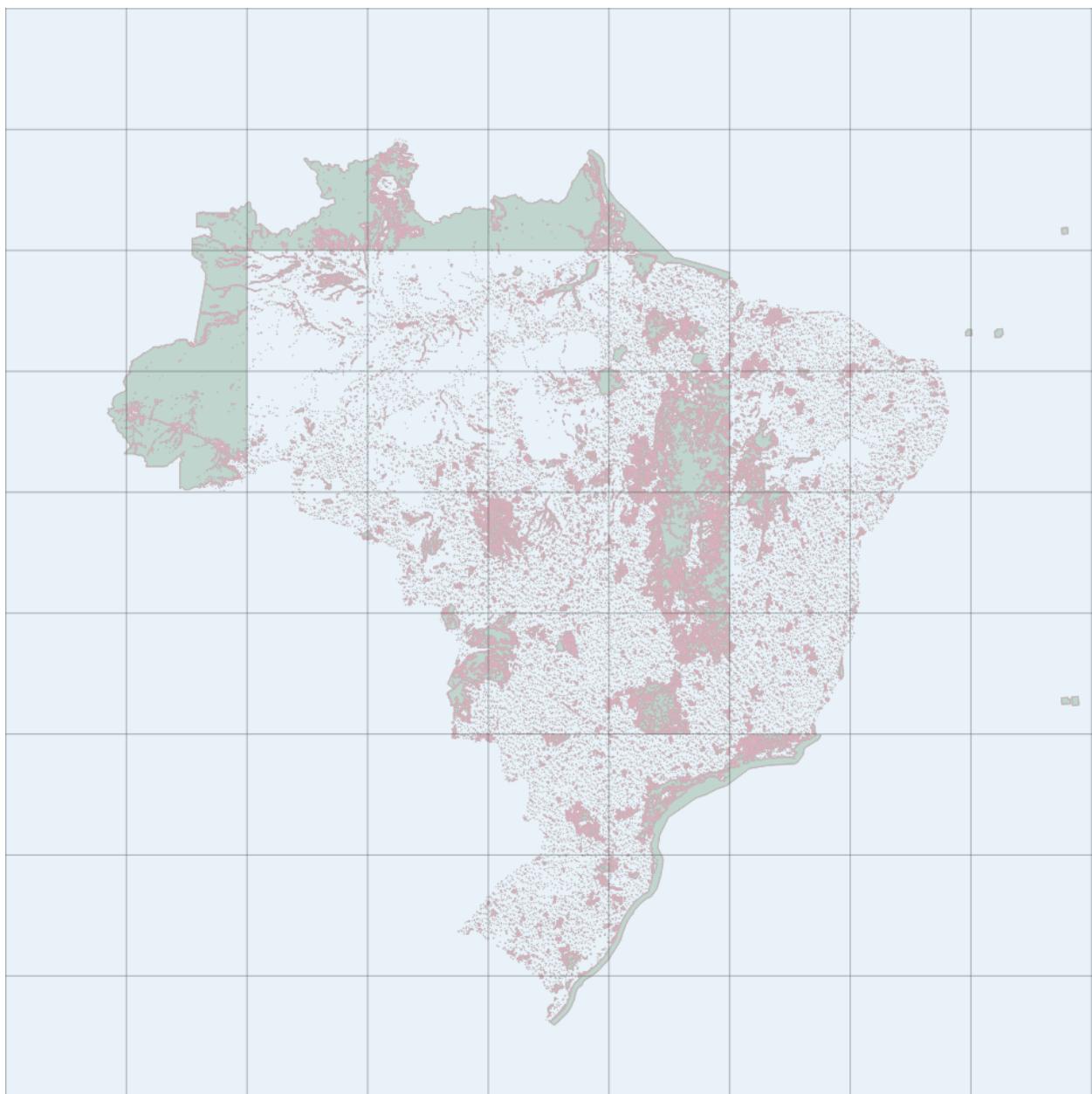


Figura 10 - Resultado em Zoom 6 dos tiles gerados pelo tippecanoe. Alguns problemas na geometria final.

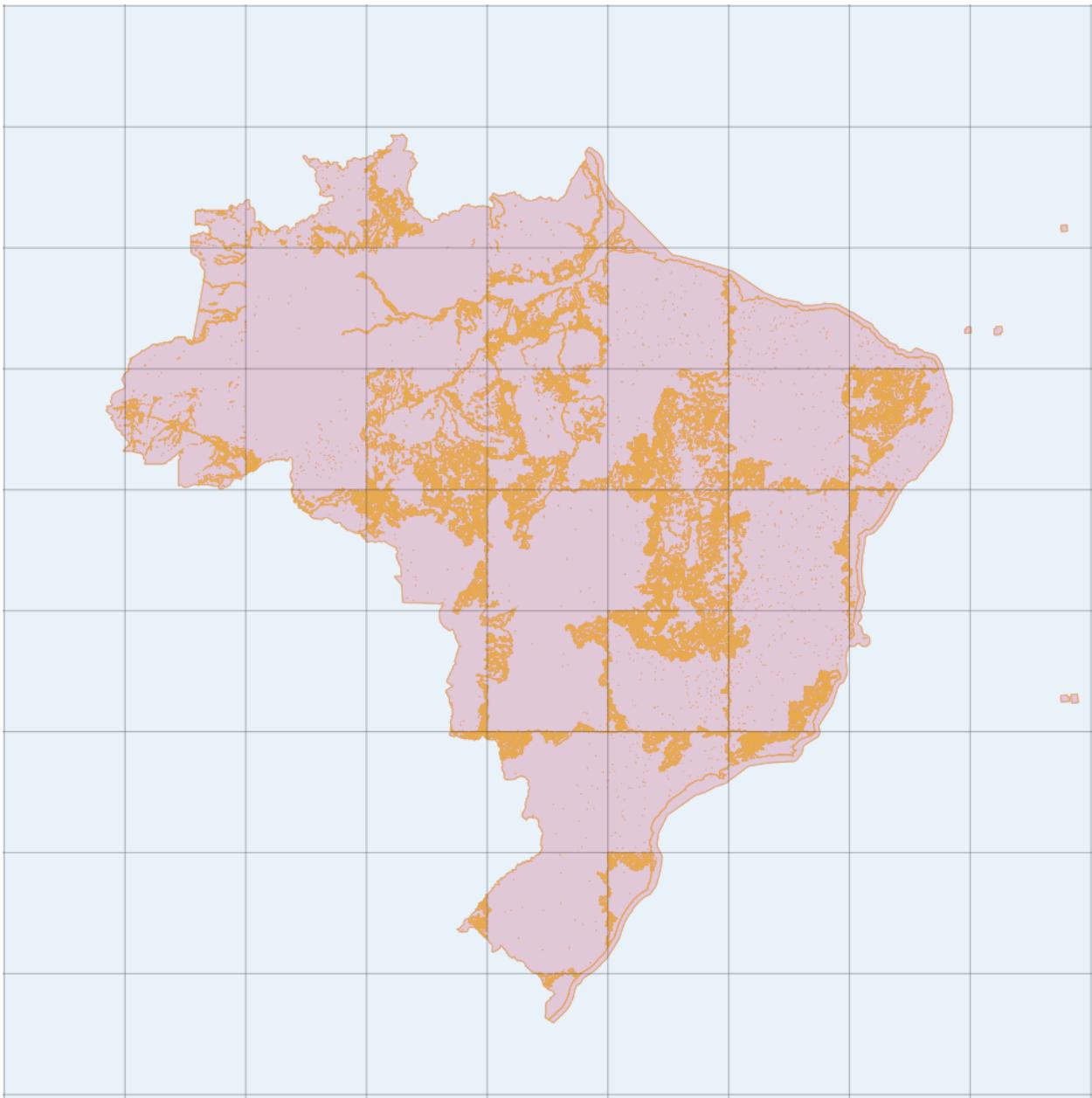


Figura 11 - Resultado em Zoom 6 dos tiles gerados pelo tippecanoe com outros parâmetros de conversão.
Alguns problemas na geometria final.

GDAL

O GDAL possui driver para conversão entre formatos diversos, incluindo MVT. Porém, a versão oficial disponibilizada através do docker possui incompatibilidade com o driver.

```
root@93e409d0cfae:/app# ogr2ogr -f MVT uso_cobertura_gdal.mbtiles /data/geojson/uso_cobertura_terra_original.json -dsco MINZOOM=6 -dsco MAXZOOM=6 -dsco FORMAT=MBTILES  
ERROR 1: ogr2ogr was compiled against GDAL 3.3, but the current library version is 3.0  
root@93e409d0cfae:/app#
```

Figura 12 - Erro na ferramenta GDAL utilizando o driver MVT

4. SOLUÇÃO

Depois de analisar algumas maneiras de geração de MVTs, o tippecanoe apresenta a melhor performance e opções de geração e optimização. Além disso, o formato de saída, o MBTILES, por se tratar apenas de um arquivo é muito fácil de distribuir e, por já ter todos os tiles produzidos e salvos no formato MVT, permite a maneira mais eficiente para consumo.

Apesar do tippecanoe funcionar muito bem para a maioria das camadas, em camadas muito complexas, o resultado apresenta problemas em zoom distantes. Isso ocorre porque é muito difícil o algoritmo calcular a forma final de muitas geometrias pequenas em um zoom muito distante.

Uma maneira de resolver esse problema seria a criação de tiles bitmaps (imagens). Porém, como bitmap se perde a possibilidade de definir no navegador como será a aparência das camadas como a possibilidade de interação.

Para poder manter essas camadas como vetoriais, mas ter sua aparência final preservada foi desenvolvida a uma nova metodologia, que consiste na seguinte ideia.

Se computacionalmente é complicado simplificar e determinar o formato final de milhares de geometrias pequenas (quando visto de um zoom distante), por outro lado é computacionalmente muito simples redimensionar uma imagem(bitmap grande em uma imagem pequena).

Então, para resolver o problema a ideia é gerar uma imagem de alta resolução a partir de um formato vetorial. Então essa versão é redimensionada para um tamanho menor (como o tamanho da imagem que estaríamos vendo na tela) e, depois, convertemos essa imagem pequena de volta para o formato vetorial. Isso para gerar geometrias vistas de um zoom distante. Esse processo tem a vantagem de ser muito rápido também. Porém, para zoom muito próximos, como somente pequenos pedaços da geometria são exibidas na tela, convertemos do formato vetorial original para o formato vetorial mvt.

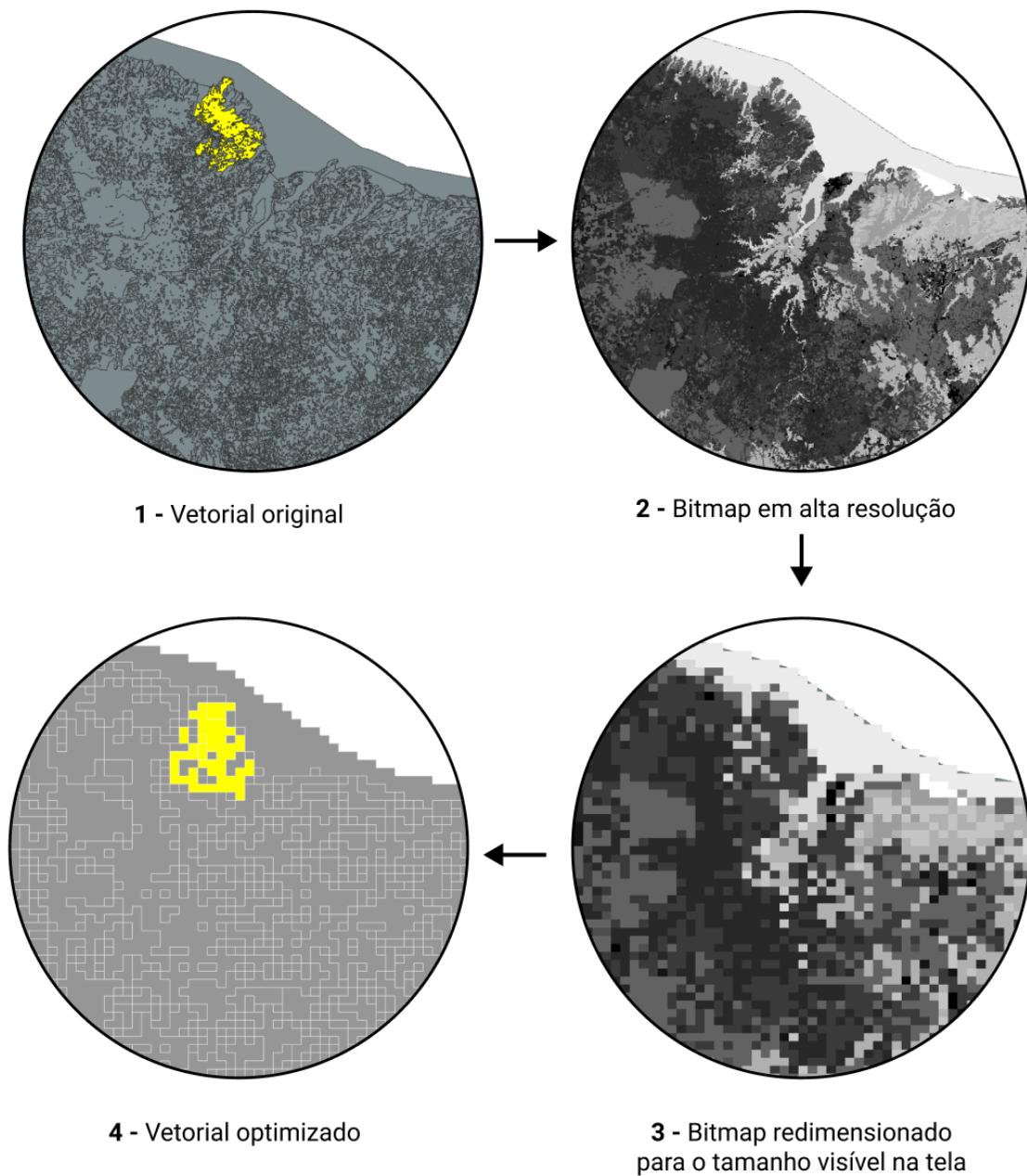


Figura 12 - Estratégia utilizada para a geração do mvt final.

Como pode-se observar na imagem abaixo, o vetor resultante visto na tela do computador a um zoom distante (zoom 5) não apresenta diferença perceptiva ao usuário final.

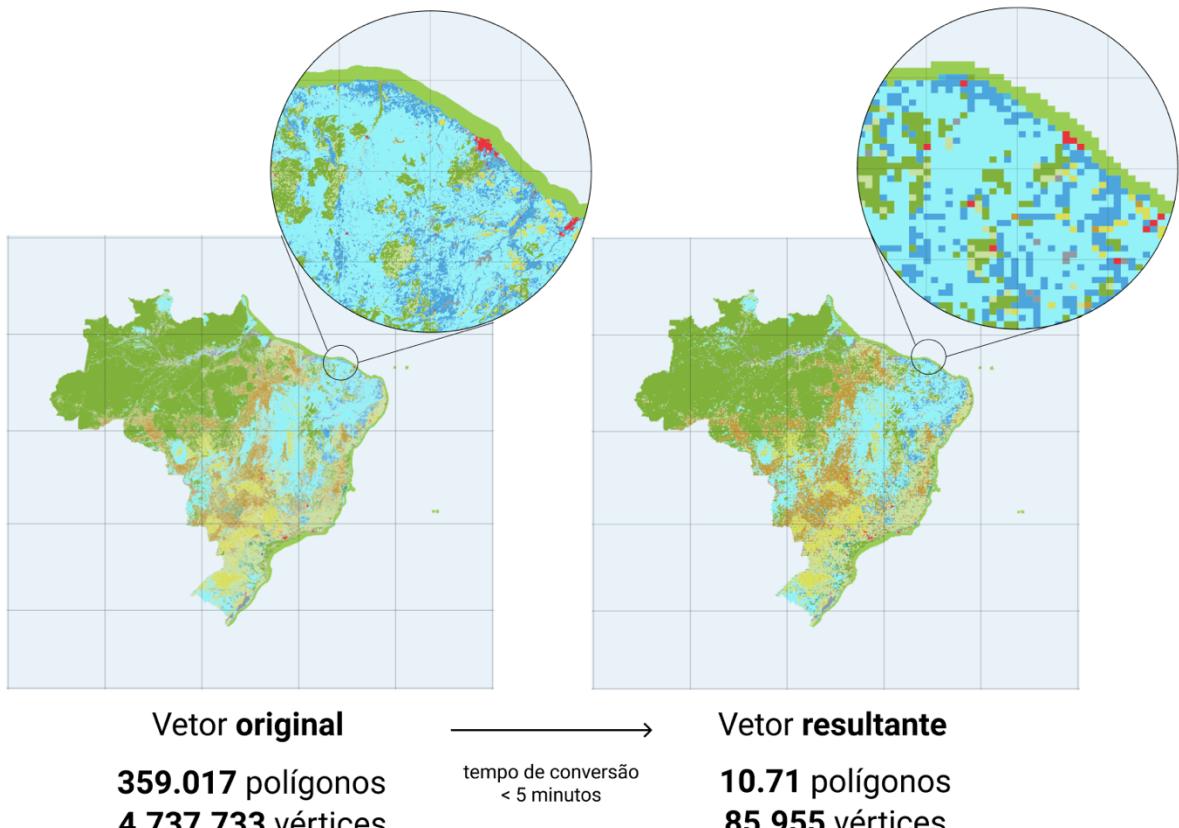


Figura 13 - Visto a um nível de zoom 5, onde a camada é exibida completamente na tela com 1280 pixels (5 tiles * 256 pixel), não existe diferença entre o formato original e o resultante, com uma diminuição de 98,18% no número de vértices em menos de 5 minutos de processamento.

Essas conversões representam também alguns desafios técnicos, por exemplo, como mantemos os atributos de cada geometria do formato vetorial, uma vez que convertido para bitmap essas informações são perdidas? Para resolver essa questão, geramos no formato bitmap GeoTiff, onde separamos cada geometria, ou geometrias que pertencem a uma mesma classe em diferentes bandas. Depois de redimensionadas, essas bandas são convertidas em formato vetorial e, nessa conversão, podemos saber a que banda cada uma das geometrias pertence, injetando novamente seus valores.

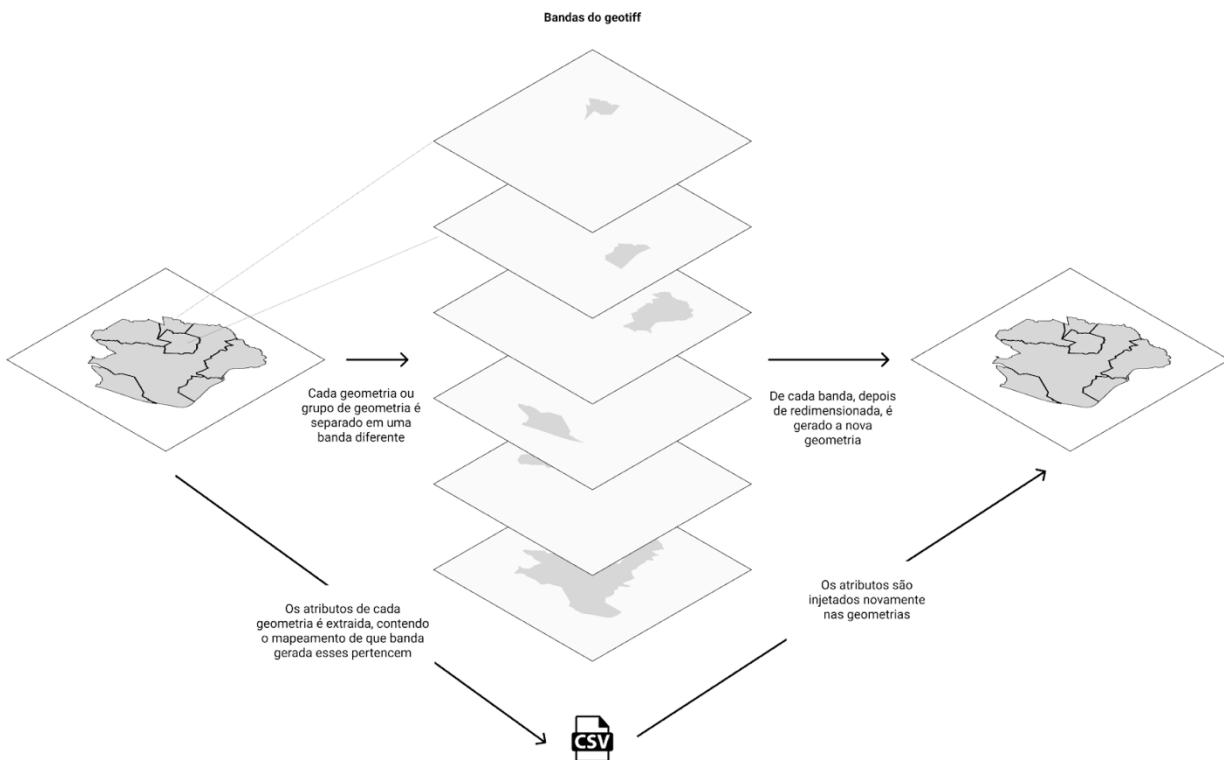


Figura 14 - Detalhamento da conversão de vetor para bitmap redimensionado e depois, novamente, para vetor, mantendo os atributos (como nome ou classe etc).

No final das conversões, temos alguns arquivos mbtiles para alguns níveis de zoom mais distantes e, a partir de uma determinada aproximação, o mbtile é gerado da geometria original para não haver perda de detalhes. Depois, juntamos todos esses mbtiles em 1 mbtile final, contendo todos os níveis de zoom. O funcionamento da solução final pode ser ilustrado conforme a figura abaixo:

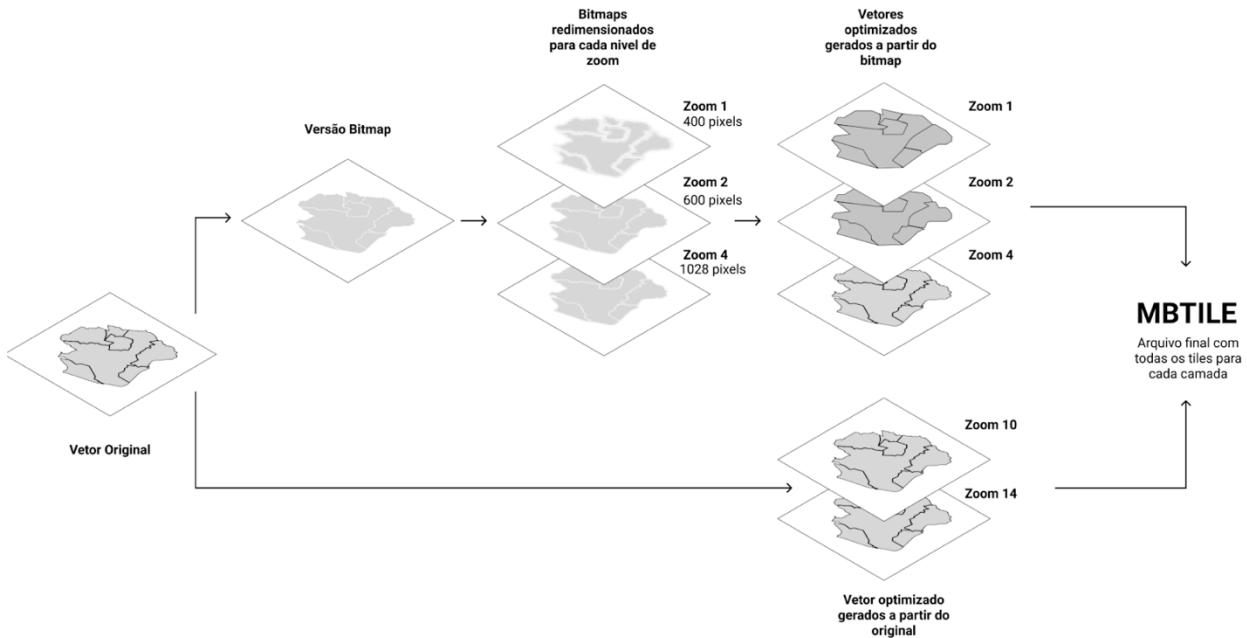


Figura 15 - Panorama da solução final.

Num teste real na aplicação sig-web, a performance ganha foi muito significante. No **Anexo A** e **Anexo B**, enviado junto deste documento, tem um vídeo demonstrando o tempo que a camada demora para ser exibida no zoom inicial do sig-web.

No anexo A, que é a demonstração da camada utilizando a abordagem relatada, a camada é exibida em menos de 2 segundos.



Figura 16 - Captura de tela do vídeo disponibilizado como Anexo A, demonstrando que o tempo para a exibição da camada é inferior a 2 segundos.

No anexo B, que é a demonstração da camada utilizando a versão disponibilizada atualmente, a camada ainda não tinha sido completamente exibida após 1min e 30 segundos



Figura 17 - Captura de tela do vídeo disponibilizado como Anexo B, demonstrando que o tempo para a exibição da camada atualmente utilizando não carrega completamente após 90 segundos.

Inclusive, a solução final exibe com mais eficiência o mapa do que o serviço fornecido pelo próprio [Mapabiomas](#), que usa um serviço WMS, utilizando tiles em formato bitmap (essa comparação não é válida para comparar a melhor tecnologia, o seu intuito demonstrar somente a experiência do usuário que acessar os 2 serviços)