

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Danilo da Silveira Figueira

Arquitetura para análise e visualização de dados geoespaciais na Web.

Belo Horizonte
2019

Danilo da Silveira Figueira

Arquitetura para análise e visualização de dados geoespaciais na Web.

Versão Final

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Clodoveu Augusto Davis Junior

Belo Horizonte
2019

2019, Danilo da Silveira Figueira.
Todos os direitos reservados

Figueira, Danilo da Silveira.

F475a Arquitetura para análise e visualização de dados geoespaciais na Web [recurso eletrônico] / Danilo da Silveira Figueira. – 2019.

1 recurso online (74 f. il., color.) : pdf.

Orientador: Clodoveu Augusto Davis Júnior

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Referências: f. 71-74.

1. Computação – Teses. 2. Sistemas de informação geográfica – Teses. 3. Infraestrutura de dados espaciais – Teses. 4. Visualização de dados – Teses. 5. Cartografia – álgebra – Teses. 6. Amazônia – Aspectos ambientais – Teses. I. Davis Júnior, Clodoveu Augusto. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. III. Título.

CDU 519.6*72(043)

Ficha catalográfica elaborada pela bibliotecária Irenquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais – ICEX



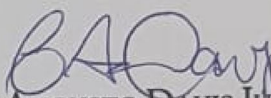
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

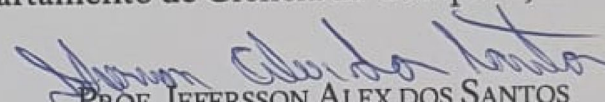
FOLHA DE APROVAÇÃO

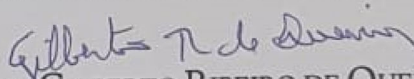
Arquitetura para análise e visualização de dados geoespaciais na Web

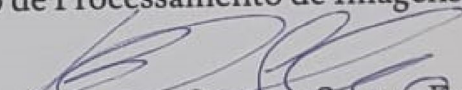
DANILO DA SILVEIRA FIGUEIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. CLODOVEU AUGUSTO DAVIS JUNIOR - Orientador
Departamento de Ciência da Computação - UFMG


PROF. JEFERSSON ALEX DOS SANTOS
Departamento de Ciência da Computação - UFMG


PROF. GILBERTO RIBEIRO DE QUEIROZ
Divisão de Processamento de Imagens - INPE


PROF. BRITALDO SILVEIRA SOARES FILHO
Departamento de Cartografia - UFMG

Belo Horizonte, 28 de Junho de 2019.

Dedico este trabalho à minha mãe, avós, tios, primos, namorada e amigos.

Acknowledgments

Em especial agradeço a minha mãe por todo o carinho, confiança, ensinamentos e investimento em meus estudos. A toda minha família, em especial minha avó pelas experiências, à meus tios por todo trabalho duro e cuidados, aos primos pelo suporte e à namorada por todo amor e apoio constante. Agradeço também aos professores por todos ensinamentos, aos amigos pela alegria. Agradeço também ao meu orientador pela ajuda no trabalho e inspiração que sua atitude proporciona. Agradeço ao Centro de Sensoriamento Remoto da UFMG e a todos os colegas que ali trabalham. Por fim, agradeço a todos que já participam da minha jornada, e direta ou indiretamente me ajudam a me tornar uma pessoa melhor.

*“Descobrir consiste em olhar
para o que todo mundo está vendo
e pensar uma coisa diferente.”*
(Roger Von Oech)

Resumo

O processamento de grandes volumes de dados geoespaciais através da Web é tipicamente implementado usando uma entre duas abordagens: concentrando o processamento no servidor ou transferindo dados para processamento no cliente. Na primeira abordagem, é necessário promover o aumento da infraestrutura, uma vez que cada requisição inicia um processo de alto custo computacional. A segunda alternativa exige a transferência de grandes volumes de dados através da rede, com impacto significativo sobre o tempo para geração do resultado. Esta dissertação propõe uma abordagem mista que otimiza o processamento online de dados geográficos. Nessa proposta, o servidor inicia com o pré-processamento dos dados aplicando algoritmos de agregação simples, para reduzir o volume de dados transferidos. Graças a isso, o cliente irá receber menos dados e deverá finalizar o processamento desses dados otimizados de acordo com a operação específica desejada. Por meio desse pré-processamento garantimos que o resultado será correto, apesar de aplicado em um conjunto simplificado de dados. O processo de simplificação pode ser executado no servidor apenas uma vez, e reutilizado em todas as requisições seguintes para o mesmo dado. Esse serviço é escalável pela sua simplicidade, bastando apenas disponibilizar esses pré-processamentos em mais servidores. Garantindo assim que o serviço possa pré-computar os resultados e apresentar melhor desempenho do que as abordagens anteriores. O trabalho inclui a implementação de simplificadores e operações algébricas em dados espaciais "álgebra de mapas", em que os dados de entrada e saída são matriciais. Assim, a validação dos resultados obtidos é feita através de um estudo de caso, desenvolvido no software de modelagem ambiental DinamicaEGO, e envolvendo dados de pesquisa da Amazônia. No estudo, o desempenho da abordagem híbrida proposta é comparado com o processamento realizado exclusivamente no servidor. O resultado mostra que é possível realizar, em uma plataforma online, consultas espaciais complexas envolvendo grandes volumes de dados, com respostas rápidas que viabilizem o compartilhamento da plataforma com diversos usuários simultâneos.

Palavras-chave: serviço web; dados geográficos; visualização de dados; álgebra de mapas.

Abstract

Processing large volumes of geospatial data over the Web is typically implemented using one of two approaches: all processing on the server side or transferring data for processing on the client. In the first approach, it is necessary to gradually increase the computational infrastructure, since each request initiates a process of high computational cost. The second alternative requires transferring large volumes of data across the network, with significant impact on the time to produce the result. This thesis proposes an architecture and presents a mixed implementation approach of an online service of geographical data processing. In this service, the server simplifies the stored data, with the intention of reducing the volume of information to be transferred to clients. Thanks to that, the clients work with data that is closer to what its operation needs and following that, completes necessary processing. The simplification process is applied only once, and the result can be reused in subsequent requests. The resulting service is more stable, more scalable, and performs better than previous approaches. This work includes the implementation of simplifiers and operations for map algebra where the input and output are matrices. Thus, the validation of the results obtained is done through a case study, developed in the software DynamicaEGO using the data from the Amazon Forest. In the study, the performance of the proposed hybrid approach is compared to the processing performed exclusively on the server. The result shows that it is possible to perform, in an online platform, complex queries involving large volumes of data, with performance that allows multiple concurrent users.

Keywords: web service; geographic data; data visualization; map algebra.

Lista de Figuras

2.1	Exemplo de formato matricial	15
2.2	Operações da álgebra de mapas.	16
2.3	Exemplo de operação local.	16
2.4	Exemplo de operação focal.	17
2.5	Exemplo de operação zonal.	17
2.6	Exemplo de operação zonal.	18
2.7	Piramide de imagens.	18
2.8	Arquitetura do serviço para infraestrutura de dados espaciais.	23
2.9	Esquema resumido que lista os padrões de distribuição de dados geográficos e os requisitos que eles atendem.	26
3.1	Exemplo da aplicação das funções de agregação.	35
3.2	Exemplos de simplificadores de agregação.	36
3.3	Exemplos dos simplificadores de área, vizinho mais próximo e soma.	38
3.4	Comparação de resultados entre simplificadores.	44
3.5	Imagem codificando valores das células.	49
4.1	Diversidade mapas escolhidos.	52
4.2	Exemplo de implementação da arquitetura.	52
4.3	Composição arquitetural do serviço.	54
4.4	Tempo de execução das operações de teste no servidor.	57
4.5	Média de tempo gasto pelas operações no mapa original.	57
4.6	Tempo de processamento para gerar o mapa simplificado até o nível oito.	59
4.7	Tempo médio por milhares de células processadas.	59
4.8	Proporção das células nulas e não nulas.	60
4.9	Comparação da área dos dados de teste.	60
4.10	Volume dos dados usados para teste.	61
4.11	Tempo médio de download por mapa.	62
4.12	Taxa de compressão dos mapas de teste.	62
4.13	Tamanho total do mapa simplificado por nível de zoom.	64
4.14	Tamanho total do dado após simplificação.	65

Sumário

1	Introdução	12
1.1	Objetivos	14
1.2	Organização	14
2	Conceitos e trabalhos relacionados	15
2.1	Álgebra de mapas	15
2.1.1	Escala de análise	17
2.2	Serviços Web geoespaciais padrão ISO/OGC	19
2.3	Avaliação de serviços OGC: WFS, WCS, WPS, WMS, WCPS	24
2.4	Serviços interativos de acesso a grandes volumes de dados geográficos	26
3	Metodologia	29
3.1	Estudo do problema	29
3.2	Definição dos simplificadores	32
3.2.1	Tipos de simplificadores	35
3.2.1.1	Simplificador de Soma	35
3.2.2	Simplificador de Média	36
3.2.3	Simplificador de Máximo	37
3.2.4	Simplificador de Mínimo	37
3.2.5	Simplificador de Quantidade de Células	37
3.2.6	Simplificador de Área	38
3.2.7	Simplificador de Centro	38
3.2.8	Simplificador de Mapa Integral	39
3.3	Definição das Operações	39
3.3.1	Tipos de operadores	40
3.3.1.1	Operador de Soma na região	41
3.3.1.2	Operador de Média na região	42
3.3.1.3	Operadores de Máximo e Mínimo na região	42
3.3.1.4	Operador de Quantidade de Células na região	42
3.3.1.5	Operador de Área da região	43
3.3.1.6	Operador de Álgebra	43
3.3.1.7	Operador de Filtragem	43

3.3.1.8	Operador de transformação, soma por célula em distribuição espacial	44
3.4	Arquitetura	45
3.4.1	Definição do padrão de requisições da API	46
3.4.2	Describe	47
3.4.3	GetMap	48
3.4.4	RunModel	49
3.4.5	Codificação e decodificação dos valores dos <i>pixels</i> da imagem	50
4	Validação	51
4.1	Processamento e publicação dos mapas utilizando o Dinamica EGO	53
4.2	Requisição dos dados	55
4.3	Processamento concentrado no servidor	55
4.3.1	Processamento concentrado no cliente	56
4.3.2	Processamento dividido entre servidor e cliente	58
4.4	Transferência dos dados	61
4.5	Visualização do resultado	63
4.5.1	Análise da visualização da multiplicação	63
4.5.2	Análise da visualização da soma	64
4.5.3	Análise dos tempos de resposta em cada abordagem	65
5	Conclusão	67
	Referências	71

Capítulo 1

Introdução

A floresta amazônica é a maior floresta tropical em extensão do mundo [19] e está permanentemente sujeita a todo tipo de pressão econômica, e ações de grande relevância são frequentemente realizadas sem que se tenha uma noção prévia de sua efetividade e impacto. Por isso é importante um estudo para embasar estratégias de exploração eficiente dos recursos naturais, potencial econômico e dificuldades associadas. Por exemplo, identificar as regiões da Amazônia com grande valor para a preservação da biodiversidade, quais áreas podem influenciar o regime de chuvas, onde é mais propício para a produção de borracha ou madeira, e quais regiões têm elevados riscos de incêndios florestais. Existe um argumento segundo o qual “a floresta vale mais em pé do que derrubada” [32] que precisa ser testado e comprovado segundo uma base de comparação do valor dos diversos produtos, de forma regionalizada, para indicar o potencial econômico de cada parte da Amazônia, levando em conta as alternativas de exploração de cada área. Portanto, é importante conhecer a dinâmica espacial de cada produto para a tomada de decisões sobre quais áreas devem ser priorizadas para conservação ou extrativismo.

Esta pesquisa surge como uma resposta ao desafio de divulgar análises dos cenários da floresta amazônica, como parte do projeto AMAZONES¹, financiado pelo Banco Mundial. O objetivo desse projeto envolve realizar análises espaciais de cenários calculando o valor da Amazônia em diferentes regiões, variando os preços dos insumos e produtos em cada cenário. Para viabilizar essa análise, a região foi dividida em células de tamanho 500 x 500 metros, e cada variável foi avaliada independentemente para cada área. Considerando que a cobertura da Amazônia é de aproximadamente 5 milhões de km², ao dividir em áreas de estudo de 500m², temos um total de 20 milhões de áreas por variável. Cada área é avaliada em relação a cada uma das variáveis, resultando em mais de 100 milhões de áreas processadas. Apesar do grande volume de dados, o projeto AMAZONES demanda duas funcionalidades importantes: a visualização espacial da produtividade de cada produto amazônico e a visualização em tempo quase real de cenários de exploração. Os usuários podem testar diferentes cenários selecionando valores de preço por tonelada e propensão à exploração para cada produto. A plataforma deve então utilizar o valor

¹AMAZONES - AMAZON ECO SERVICES, CSR. Disponível em: <http://maps.csr.ufmg.br/calculator/?queryid=106>. Acesso em 16 de outubro de 2018.

de intensidade para cada produto multiplicado pelo seu preço, a fim de gerar um mapa combinado que indique as áreas que devem ser priorizadas.

Para que os objetivos do projeto AMAZONES sejam plenamente alcançados, faz-se necessário recursos computacionais que permitam a execução de simulações sobre um grande volume de dados a partir de plataformas online e de forma interativa. Espera-se que múltiplos usuários simultâneos interajam com uma ferramenta em que parâmetros da simulação possam ser definidos e o resultado possa ser visualizado. Portanto, o objetivo deste trabalho é descrever a criação de uma plataforma web para a divulgação e o processamento em tempo real de dados geoespaciais com baixo custo computacional. São pontos importantes para esse objetivo: a divisão do custo de processamento entre servidor e cliente, a otimização do volume de dados trafegados na rede, o baixo tempo de resposta do servidor e o baixo custo computacional ou monetário da solução. Atualmente, a maioria dos servidores de mapas com capacidades de processamento utiliza arquiteturas em que todos os dados são processados no servidor e apenas o resultado é enviado ao usuário. No entanto, essa abordagem demanda alto custo computacional, que aumenta proporcionalmente à quantidade de clientes do serviço. E até onde pudemos determinar, nenhuma ferramenta existente atualmente é capaz de prover serviços que atendam a todos os requisitos necessários para processar o volume de dados disponível sobre a Amazônia. Por exemplo, o Google Earth Engine [11] permite processar imagens, mas sua licença gratuita apenas possibilita criar os dados iniciais. Assim, não constitui uma plataforma operacional de processamento².

Neste trabalho é proposta uma abordagem mista de processamento com partes executadas no servidor e outras no cliente. O primeiro é responsável por preparar o dado para agilizar o processamento e a transmissão pela internet, pré-processar cada mapa para reduzir o volume de dados, sem alterar o resultado, já o segundo irá processar apenas o dado reduzido e otimizado para cada operação desejada. Para isso, foi necessário realizar uma pesquisa de técnicas de compressão específicas para o processamento e visualização de mapas online. **Pesquisou-se** também métodos de simplificação de dados, como a reamostragem, que permitam que as operações realizadas tanto no mapa original quanto no mapa simplificado sejam exatas e produzam o mesmo resultado, quando aplicadas à mesma área.

²Noncommercial - Google Earth Engine. 2023. Disponível em: <https://earthengine.google.com/noncommercial/>. Acesso em: 30 maio 2018.

1.1 Objetivos

O objetivo geral desta dissertação de mestrado é projetar uma plataforma online que permita a divulgação de um grande volume de dados matriciais para visualização interativa na Web. A plataforma deve suportar pequenas simulações em tempo real, envolvendo um conjunto significativo de operações pixel a pixel. A arquitetura proposta deve buscar maximizar o desempenho e minimizar o custo computacional, especialmente reduzindo a carga no servidor. A aplicação imediata dessa plataforma será direcionada aos dados do projeto AMAZONES, permitindo a seleção e visualização de cenários de exploração da Amazônia. Esses dados serão utilizados para validar o projeto e sua implementação, mas as contribuições deste trabalho podem ser aplicadas a outros casos de uso.

Para alcançar esse objetivo, são definidos cinco objetivos específicos:

- Otimizar operações pixel a pixel entre mapas para a análise de cenários em tempo real para visualização na web;
- Potencializar o pré-processamento e evitar reprocessamento dos dados matriciais no servidor;
- Reduzir o tempo de resposta e o custo computacional de requisições simultâneas;
- Minimizar o processamento sobre demanda no servidor;
- Minimizar a transferência de dados.

1.2 Organização

Este trabalho está organizado da seguinte maneira. O Capítulo 2 apresenta trabalhos relacionados e conceitos importantes como: a álgebra de mapas e os serviços Web geoespaciais. No Capítulo 3 é detalhada a estrutura da solução, incluindo o detalhamento das operações utilizando os simplificadores. O Capítulo 4 apresenta a implementação da plataforma proposta, e demonstra sua funcionalidade em um estudo de caso. Finalmente, o Capítulo 5 apresenta conclusões e uma discussão sobre trabalhos futuros.

Capítulo 2

Conceitos e trabalhos relacionados

Este capítulo apresenta os conceitos-chave e descreve trabalhos relacionados à proposta da dissertação. A Seção 2.1 discorre sobre o formato dos dados e os tipos de operações que compõem a álgebra de mapas. A Seção 2.2 descreve os padrões Web OGC utilizados para disseminação e acesso a dados geospaciais, incluindo uma discussão sobre os formatos disponíveis para transmissão de dados. O uso dos padrões OGC para grandes volumes de dados matriciais foi conduzido e avaliado em alguns trabalhos de pesquisa, apresentados na Seção 2.3 Ao final do Capítulo, apresentamos uma discussão sobre as limitações das técnicas apresentadas, levando à proposta de arquitetura no Capítulo 2.

2.1 Álgebra de mapas

A álgebra de mapas é um conjunto de operações, largamente adotadas, para a manipulação de dados geográficos, no formato matricial (raster), proposto por [33]. Os mapas raster são compostos por uma matriz de células, também conhecidas como pixels, que contêm informações valor de uma determinada variável geográfica [Figura 2.1]. Cada

Figura 2.1: Representação visual de um mapa, como os *pixels* compõem um dado matricial

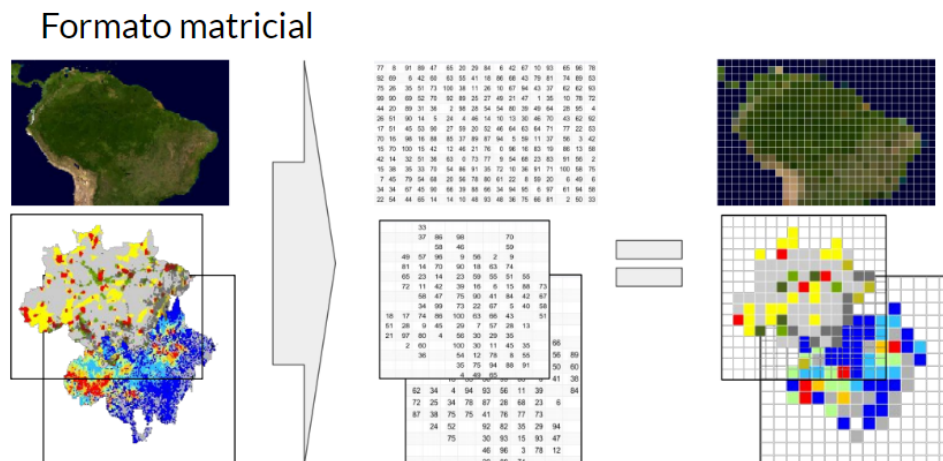
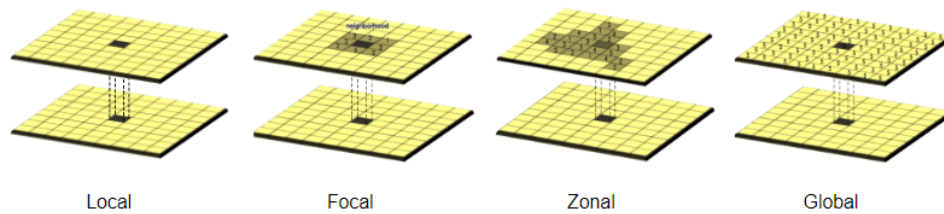
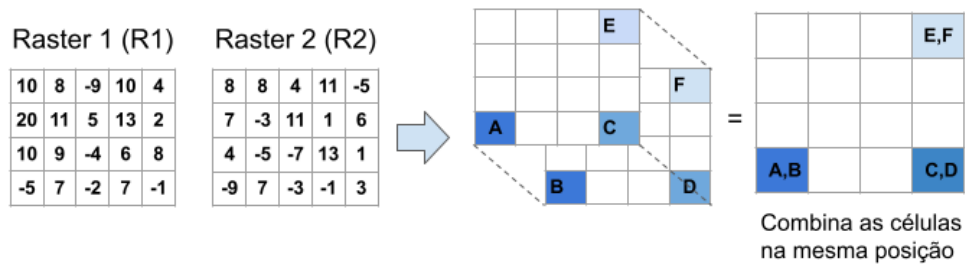


Figura 2.2: Tipos de operações contidas na álgebra de mapas.



Operação local: utiliza a célula na mesma posição.

Exemplos de dados de entrada e operação célula a célula:



Exemplos operações:

Soma: $R1 + R2$

18	16	-5	21	-1
27	8	16	14	8
14	4	-11	19	9
-14	14	-5	6	2

Subtração: $R1 - R2$

2	0	-13	-1	-9
13	-14	-6	12	-4
6	14	3	7	7
4	0	1	8	-4

Multiplicação: $R1 * R2$

80	64	-36	110	-20
140	-33	55	13	12
40	-45	28	78	8
45	47	6	-7	-3

Figura 2.3: Exemplos de operações locais de: soma, subtração e multiplicação.

célula tem uma posição espacial definida pela sua coordenada X, Y e também pelo seu tamanho, que é determinado pela escala do mapa. No contexto geoespacial esse formato pode detalhar o território em pequenas áreas representadas com um valor representando a intensidade ou classe que melhor represente cada uma. Esses dados podem ser classificados quanto ao significado dos valores em suas células em dois tipos: estoque ou categórico. No caso dos dados de estoque, cada valor representa a intensidade da característica estudada, sendo os valores comparáveis. Desta forma, uma célula com valor maior que outra indica a intensidade maior que a da outra região. Já em mapas categóricos, cada valor representa uma característica distinta ou uma classe diferente.

Os tipos de operações sobre dados rasters [Figura 2.2] podem ser classificadas quanto ao padrão de células envolvidas, em quatro grupos: Locais, Focais, Zonais e Globais [33]. As operações locais utilizam apenas a célula na mesma posição nos mapas de entrada como exibido [Figura 2.3], são comuns operações de: soma, multiplicação, subtração, entre outras. Já as operações focais processam dados na vizinhança da célula, a região de vizinhança é arbitrária e alguns exemplos podem ser consultados na [Figura 2.4] que permitem operações como: soma, multiplicação e máximo. As operações zonais são úteis para obter resultados em regiões pré-definidas, que podem ser, por exemplo,

Figura 2.4: Exemplos de operações focais de: soma, máximo e multiplicação por matriz.

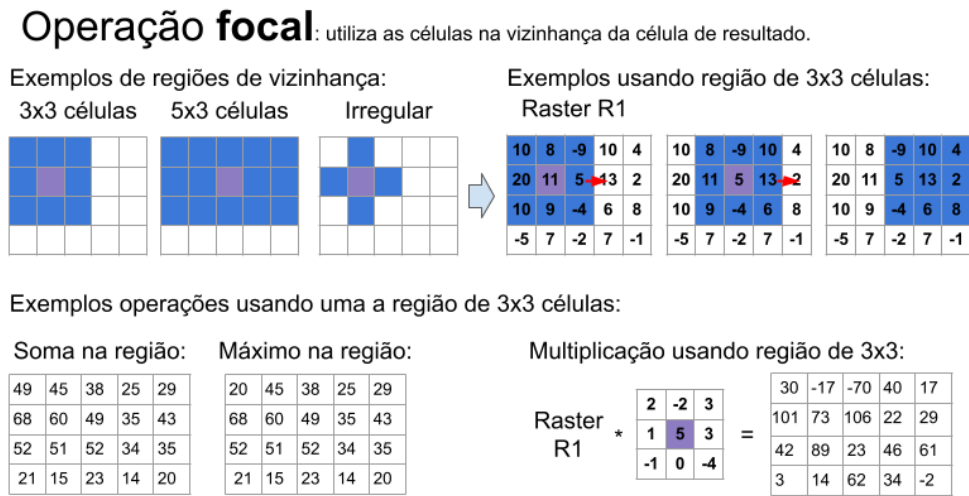
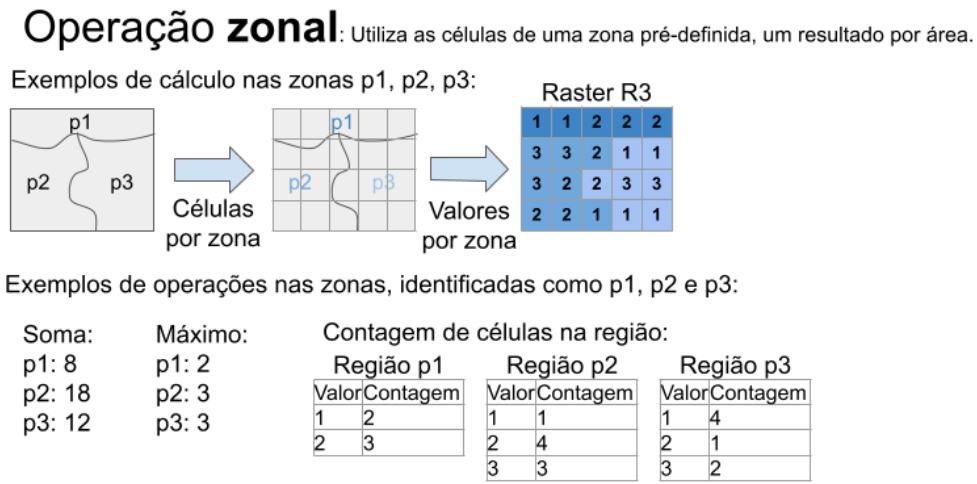


Figura 2.5: Exemplos de operações zonais de: soma, máximo e contagem de células.



limites de municípios, biomas ou quaisquer regiões de interesse. O zonal retorna um resultado para toda a região, podendo ser um valor único ou mesmo um conjunto de valores, como os exemplos [Figura 2.5]. As operações globais, por sua vez, consideram todas as células do mapa, e o resultado pode ser também um valor único ou mesmo um conjunto de valores que, nesse caso, representa todo o mapa [Figura 2.6].

2.1.1 Escala de análise

A escala é um conceito fundamental na análise de mapas raster. Ela se refere à relação entre a representação cartográfica de um mapa e a realidade geográfica na região. A escala indica o nível de detalhamento de um mapeamento correlacionando a distância

Figura 2.6: Exemplos de operações zonais de: soma, máximo e contagem de células.

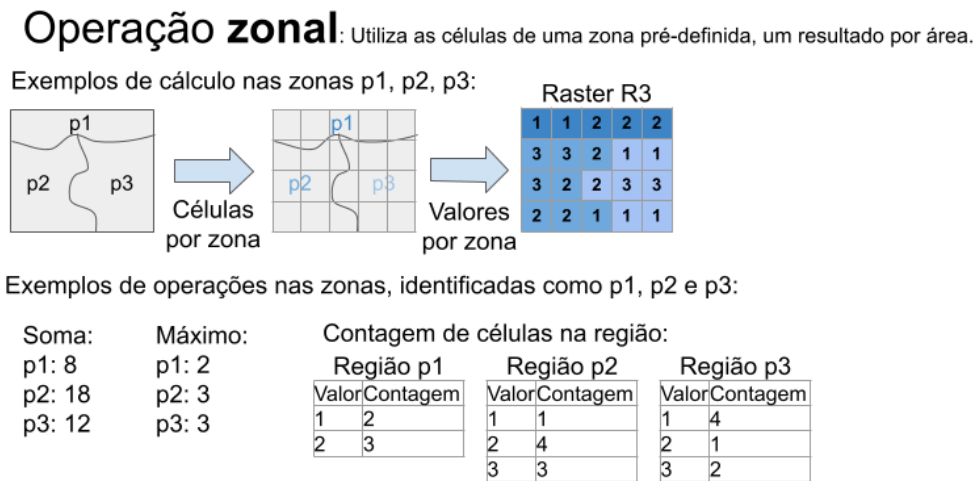
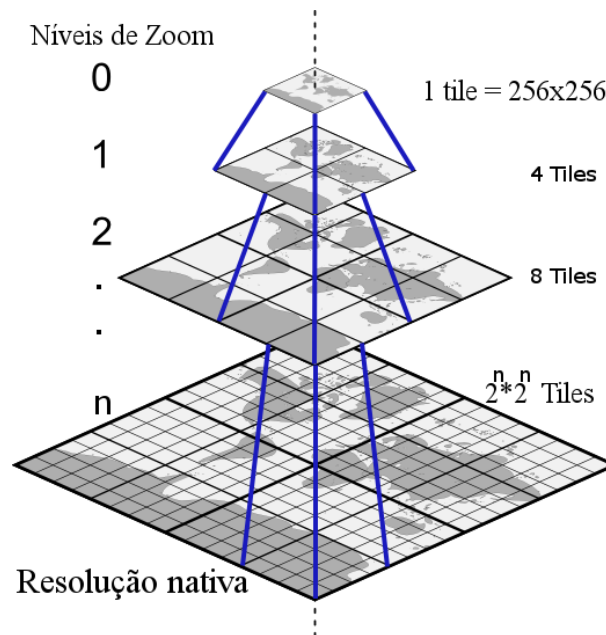


Figura 2.7: Piramide de imagens: A representação gráfica de uma piramide de imagens. Onde as linhas na vertical representam as células que serão usadas em cada aplicação da regra de agregação das células, geralmente a cada janela de 2x2 resultando em uma célula gerando imagem com metade do tamanho e da largura.



entre dois pontos no mapa e a distância no mundo real. Quanto maior a escala, maior é o nível de detalhes e a informação mais fidedigna. A escala pode ser expressa como uma razão, por exemplo de (1:50.000) indicando que a distância real entre dois pontos no mundo real é 50.000 vezes maior do que a distância entre esses mesmos pontos no mapa.

Os mapas podem ser representados em diferentes escalas, dependendo da utilização pode ser interessante mais ou menos detalhes. O método de alteração de escala do mapa é chamado de reamostragem, que consiste em agrupar informações de pixels existentes aplicando métodos de interpolação bilinear, bicúbica e Lanczos ou métodos estatísticos como média, soma, máximo, mínimo, entre outros. É importante escolher o método

apropriado para cada caso específico, considerando o tipo de imagem e o objetivo da reamostragem. Uma técnica para otimizar o processamento de imagens consiste em gerar e armazenar múltiplas versões da mesma imagem com escalas diferentes, conhecida como *pirâmide de imagens* [1]. Cada nível da pirâmide representa uma versão da imagem com resolução reduzida, geralmente por meio da reamostragem. A reamostragem é aplicada sucessivamente em cada nível da pirâmide, resultando em versões cada vez menores da imagem original. A vantagem da criação de uma pirâmide de imagens é que ela permite acessar rapidamente versões da imagem com resoluções diferentes, sem precisar realizar a reamostragem da imagem a cada vez. Isso é especialmente útil em aplicações que requerem o processamento de imagens de grande escala. Além disso, a pirâmide de imagens permite uma representação hierárquica da imagem, facilitando a seleção da resolução adequada para cada tarefa específica.

Os servidores de mapa na web utilizam o conceito de *nível de zoom* que define quais as escalas os mapas devem ser disponibilizados e permite a interoperabilidade entre diferentes aplicações de mapas. Quanto maior o nível de zoom, menor é a escala e maior é a quantidade de detalhes que podem ser exibidos. O nível 0 permite a representação do mundo em uma imagem de 256 por 256 pixels. Enquanto o próximo nível terá 2 por 2 dessas imagens e cresce continuamente em potência de 2 [22]. Por exemplo, um nível de zoom mais elevado permite que o usuário visualize ruas e prédios com mais clareza, enquanto um nível de zoom mais baixo mostra somente regiões maiores como cidades e países. Além disso, esses níveis de zoom são dinâmicos e podem ser ajustados pelo usuário em tempo real, o que torna a experiência de visualização de mapas ainda mais interativa e personalizada. Por exemplo, ao aumentar o nível de zoom de 10 para 11, a área representada por cada pixel é dividida por 4 e permite maior detalhamento de uma área menor.

Nesse trabalho é definido o conceito de *simplificadores*, que define operações de reamostragem que podem ser aplicadas no mapa em diferentes níveis de zoom com algumas restrições garantindo que o resultado seja exato se aplicado na mesma área do dado original, como descrito na seção 3.2.

2.2 Serviços Web geoespaciais padrão ISO/OGC

A OGC (Open Geospatial Consortium) e a ISO (International Standards Organization) definiram em conjunto padrões para a disseminação de dados geográficos e a

disponibilização de mapas pela internet¹. Esses padrões foram estruturados no modelo de comunicação cliente-servidor. Nessa abordagem todos os clientes devem enviar requisições para um servidor ou conjunto de servidores que, por sua vez, retorna a resposta necessária ao cliente. A padronização permitiu a criação de serviços interoperáveis construídos sobre tecnologias de serviços Web. Um serviço é uma aplicação que permite o compartilhamento de dados independente do sistema operacional e da plataforma onde estas operações são executadas [28]. Vários serviços foram definidos pela OGC, como o Web Coverage Service (WCS), Web Feature Service (WFS), o Web Map Service (WMS) e o Web Processing Service (WPS). Além destes, vários outros padrões foram definidos, como o Web Coverage Processing Service (WCPS), que define a sintaxe para descrever operações entre mapas, e o CWS, que descreve os mapas presentes em cada serviço, parte essencial de infraestruturas de dados espaciais (IDEs). A utilização destes padrões é desejada, pois, como mostram [10] os projetos que utilizaram os padrões da OGC consumiram 25% a menos de recursos para serem implementados, tomando como base o serviço WCS².

Em 2000 a OGC publicou a primeira versão do padrão **Web Map Service (WMS)**, que permitiu a padronização dos serviços de visualização de mapas georreferenciados pela internet de maneira rápida utilizando o protocolo HTTP (hypertext transfer protocol) para comunicação³. Esse serviço foi projetado para permitir a visualização de partes de um banco de dados geoespacial através da Web. Ele funciona de maneira simples: para cada escala uma versão do mapa é exibida. Essas imagens são produzidas no servidor, para isso o dado original é reamostrado para a escala de visualização desejada e aplica-se a transformação dos valores em cores por meio da aplicação da legenda. A visualização do dado é feita sob demanda, por meio dos pedidos do cliente, que especifica a área e o nível de detalhamento das partes do mapa de interesse.

O protocolo de comunicação WMS oferece diversas funções importantes, como o (GetCapabilities), que permite ao cliente obter informações sobre os dados disponíveis e formatos de saída. Com essas informações, o cliente pode realizar consultas (GetMap) ao servidor, especificando os limites espaciais, classes de dados, escala, formato de saída, entre outros detalhes opcionais. Além disso, o controle destes parâmetros permite otimizar o volume de dados, pois é possível exibir regiões de forma mais detalhada ou menos detalhada sem a necessidade de carregar todos os dados. Outro ponto importante é que o servidor pode tratar qualquer dado independente do seu tamanho original, pois todos os dados são reamostrados para o mesmo nível de zoom. or conta do grande volume de dados e níveis de zoom, é necessário o uso de um serviço de cache de imagens para

¹OGC - Standards Implementation Specifications. 2017. Disponível em: <http://www.opengeospatial.org/standards/specifications/list>. Acesso em: 30 maio 2017.

²OGC® WCS 2.0 Interface Standard - Implementation Specification. 2012. Disponível em: <http://www.opengis.net/doc/IS/wcs-core-2.0.1>. Acesso em: 25 maio 2019.

³OpenGIS® Web Map Server Interface Implementation Specification. Disponível em: https://portal.ogc.org/files/?artifact_id=7196. Acesso em: 20 de maio de 2019.

evitar o sobrecarregamento do servidor [20]. Além disso, é possível consultar elementos do banco de dados através da imagem, indicando um *pixel* da imagem, e a função (GetFeatureInfo) transforma coordenadas na imagem em coordenadas geográficas, produzindo uma operação de consulta ao servidor, que retorna ao cliente no formato desejado [6]. Embora o WMS seja eficiente para a visualização de dados, ele é insuficiente para processamento de cálculos nos mapas, já que os dados enviados ao cliente apenas representam as cores de visualização em tela, que substituíram os valores das células do dado original. No entanto, devido à grande quantidade de dados e aos avanços tecnológicos, é cada vez mais imprescindível implementar processos computacionais complexos de análise para transformá-los em informação, conforme apontado por [33]. Para esse fim, a OGC definiu vários serviços, incluindo o WFS e o WCS que permitem o download do dado bruto para análise no computador local, bem como o WPS que realiza o processamento no servidor e envia apenas o resultado.

O serviço **Web Feature Service (WFS)**, ao contrário do WMS, recebe do servidor objetos geográficos codificados em GML, em lugar da imagem. Esses objetos permitem que o cliente tenha a liberdade de apresentar os dados vetoriais com qualquer simbologia desejada. Além disso, o dado fica disponível para quaisquer análises locais com exatidão. No entanto, esse formato textual é relativamente maior que o arquivo binário. O WFS é um formato extensível, e, portanto, cada servidor pode acrescentar a sua peculiaridade. Devido a isso, este formato pode ser acrescido de novas informações e adaptado a diversas necessidades, mas que também pode gerar problemas de interpretação entre diferentes servidores. Contudo, para que o cliente possa processar o dado, é preciso transferir os dados vetoriais em GML pela rede. O serviço WFS também possui a função GetCapabilities para a listagem dos dados e operações disponíveis. Com isso, é possível especificar a geometria na requisição (GetFeature), sendo que o servidor se encarrega de filtrar e gerar a resposta no formato GML para envio ao cliente. A função Transaction permite editar os dados do servidor, enviando uma requisição com o tipo de operação e o conteúdo da alteração em formato GML. As operações suportadas são criar, modificar e deletar atributos, e também suas coordenadas e projeção [36].

Dados matriciais são acessados por meio do serviço **Web Coverage Service (WCS)**. Esse serviço dispõe de uma função (GetCoverage) que retorna dados em formato matricial, em diversos formatos, alguns inclusive similares aos do WMS, porém, nesse caso, com informações adicionais como metadados e os valores exatos de cada célula. Porém, os dados geográficos podem ser BigData [27], como afirma a OGC, pois não são incomuns dados que, mesmo após compressão, ocupam vários gigabytes. Esse serviço também implementa a função GetCapabilities para a listagem de mapas e operações. Outra operação importante é a DescribeCoverage, que permite consultar informações adicionais, como sistema de coordenadas, tipo de célula e limites dos valores do mapa⁴. Mapas matriciais

⁴OpenGIS® Web Map Server Interface Implementation Specification. Disponível em: <https://>

gerados pelo WCS podem ser gerados em vários formatos, tais como JSON, GeoJSON, GeoTIFF, TIFF, PNG e NET-CDF.

A OGC define o **Web Processing Service (WPS)**⁵, que permite ao usuário obter informações sobre as funções disponíveis através da função `GetDescription`. Através do comando `DescribeProcess` obtém-se a descrição detalhada da assinatura das funções e dos parâmetros. A chamada pode ser síncrona ou assíncrona, dependendo da escolha do usuário. O processamento síncrono é recomendado para operações rápidas, enquanto o processamento assíncrono é apropriado para operações de longa duração. A resposta inicial do servidor é o identificador da operação em execução. O usuário pode gerenciar a requisição através deste identificador, verificando o estado da operação (aguardando, pausada, em execução, cancelada ou finalizada). Quando a execução é concluída, os resultados são armazenados no servidor por um período de tempo e podem ser obtidos mediante uma requisição `GetExecuteStatus`.

Com esses e outros padrões, a OGC propõe uma arquitetura de serviços Web padronizada que facilita o acesso a dados geográficos por meio de padrões abertos de dados e distribuição de metadados. Essa arquitetura viabiliza a construção de *infraestruturas de dados espaciais (IDE)*, que, segundo [4], abrangem políticas, tecnologias, padrões e pessoas para a coleta, gerenciamento, acesso, distribuição e uso de dados espaciais pela comunidade. Esse modelo foca em três elementos principais: o conteúdo, a tecnologia e as pessoas. Ele integra diversas fontes de dados que podem ser processadas de forma conjunta, mesmo com diferenças técnicas, permitindo disponibilizar os resultados amplamente. Os clientes precisam identificar os serviços disponíveis e compreender os dados e suas metainformações, o que é viabilizado por um catálogo fornecido pelo serviço *Catalog Service for the Web (CSW)*⁶. Com essas informações, o cliente pode acessar os serviços via internet ou por softwares SIG que implementem esses padrões, garantindo o uso correto dos dados ao considerar projeções, tipos de célula e metadados, como ilustrado na figura 2.8. Essa arquitetura assegura que todos tenham acesso aos dados mais atualizados diretamente na fonte, evitando perda de sincronização e retrabalho. Consultas aos metadados permitem identificar dados úteis que não precisam ser gerados novamente, otimizando o processo. Além disso, a ampla divulgação dos dados permite alcançar os agentes que promovem mudanças. Dessa forma, facilitar o acesso aos dados aumenta o impacto positivo e promove decisões mais embasadas e efetivas.

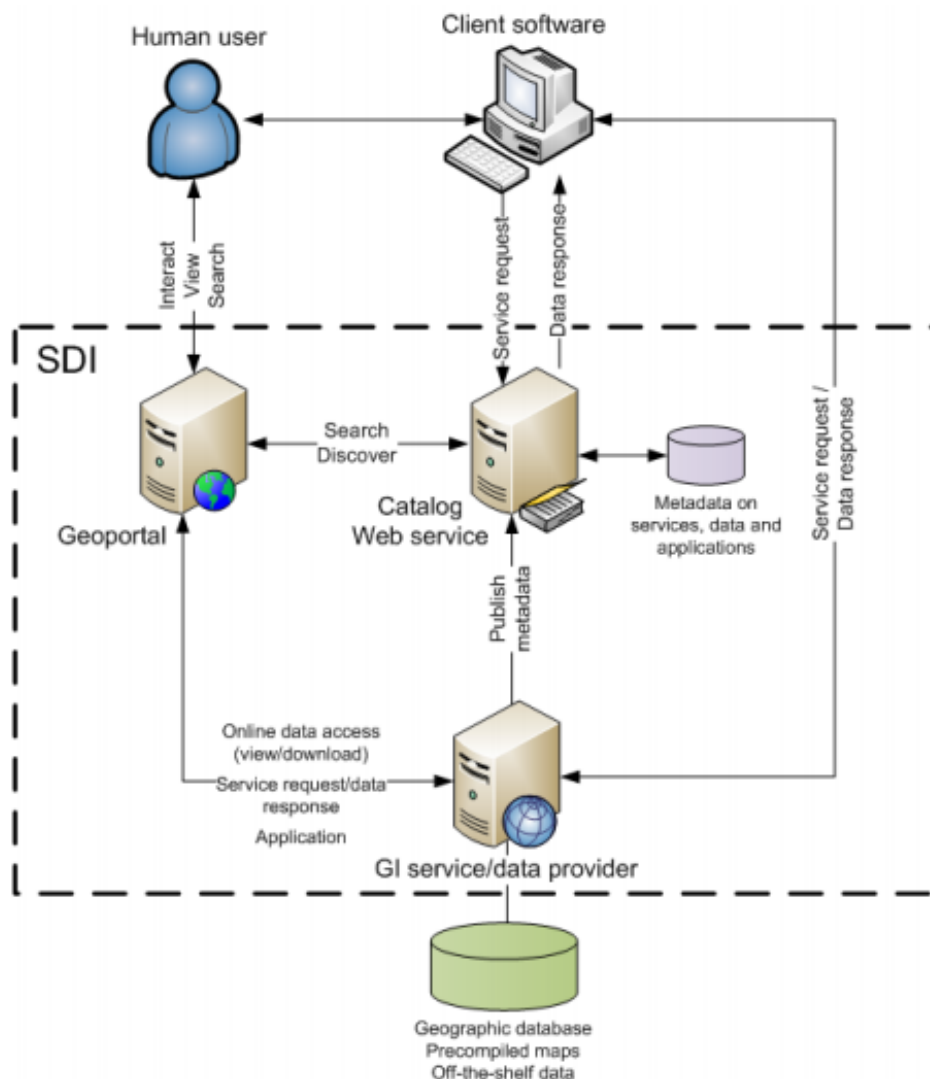
A transmissão de dados entre clientes e servidores, na arquitetura básica da OGC, pode ser feita em diferentes formatos, de acordo com a finalidade dos servidores de aplicação. São padrões estabelecidos pela OGC para dados vetoriais os formatos GML,

portal.ogc.org/files/?artifact_id=7196. Acesso em: 20 de maio de 2019.

⁵Web Processing Service (WPS). Disponível em: <http://www.opengeospatial.org/standards/wps>. Acesso em: 20 de maio de 2019.

⁶OGC® Catalogue Services 3.0 - General Model. 2016. Disponível em: <https://docs.ogc.org/is/12-168r6/12-168r6.html>. Acesso em: 23 de janeiro de 2025.

Figura 2.8: Arquitetura do serviço para infraestrutura de dados espaciais.



Fonte: (Davis Junior et al. 2009.) Disponível em: https://www.researchgate.net/publication/228776340_A_Web_Map_Service_for_Mobile_Computers Acesso em: 30 maio 2018

KML [36], SVG⁷, e GeoJSON⁸. No entanto, para a transmissão de dados matriciais, são mais usados padrões tradicionais: TIF, JPEG, PNG, GIF, entre outros. Essa diversidade de formatos possibilitou uma grande abrangência de utilização desses serviços, pois o formato a utilizar poderá ser escolhido de acordo com a necessidade de cada utilização. Assim, como queremos implementar a utilização desses dados no navegador do cliente, é preciso verificar que formatos são suportados completamente ou parcialmente nos navegadores mais relevantes⁹, adotamos o

⁷Scalable Vector Graphics (SVG) 1.1 (Second Edition). Disponível em: <https://www.w3.org/TR/SVG11/>. Acesso em 20 de abril de 2017.

⁸The gejson format. 2016. Disponível em: <https://www.rfc-editor.org/info/rfc7946>. Acesso em 24 de janeiro de 2025.

⁹Browser market share, NetMarketShare. Disponível em: <https://www.netmarketshare.com>. Acesso em 16 de outubro de 2018.

formato PNG para dados matriciais. Os formatos TIFF, GeoTIFF e NET-CDF, apesar de serem amplamente utilizados para armazenar dados de cobertura, não são suportados por várias versões dos navegadores. O formato PNG possui vários pontos positivos, como ampla utilização, ser suportado no serviço WMS, e permitir compressão sem perda. Quanto ao suporte, a grande maioria dos navegadores implementa a especificação desse formato, utilizando 4 bytes por *pixel*, sendo um byte para transparência e três para a definição da cor.

Desde sua introdução, os serviços Web e a arquitetura proposta pela OGC têm sido intensivamente usados em IDEs e outros ambientes correlatos. Existem diversos estudos que analisam esses serviços Web do ponto de vista de eficiência, escalabilidade e outros aspectos de interesse para este trabalho. A próxima subseção avalia os padrões e suas utilizações nos projetos.

2.3 Avaliação de serviços OGC: WFS, WCS, WPS, WMS, WCPS

O modelo de arquitetura da OGC foi desenvolvido na década de 1990 e ganhou maior destaque a partir de meados dos anos 2000. Inicialmente, seu foco era garantir a interoperabilidade e a disseminação de dados entre sistemas. Contudo, com a popularização dos serviços web, tornou-se essencial implementar soluções que priorizem o processamento de dados. Apenas em 2007 a OGC introduziu o padrão WPS, que trouxe a capacidade de realizar operações de processamento no servidor. O WCPS complementa essa abordagem ao definir uma linguagem conceitual para o processamento de imagens raster. Entretanto, as operações descritas por esse padrão são geralmente executadas por serviços como o WPS ou o WCS, e, por essa razão, não serão analisadas separadamente neste trabalho. Para entender as capacidades e avaliar a aplicação desses serviços, serão analisados os padrões WFS, WCS, WPS, WMS e WCPS, considerando sua utilização em diferentes projetos.

Como em [34] que discute a implementação desses padrões em software de SIG. São avaliadas as limitações e peculiaridades dos serviços WMS, WFS e WCS, e também discute-se uma nova proposta, que busca melhorar a performance utilizando uma combinação de padrões, de acordo com o formato mais adequado para cada tarefa. A arquitetura propõe a escolha automática do serviço, de acordo com a ação requerida pelo usuário. Considere por exemplo duas situações, em uma o mapa é usado apenas para uma consulta visual, e em outra, é usado no cálculo da média dos valores dos *pixels* de

uma região. Na primeira, o WMS é mais indicado, por gerar uma imagem de dimensões reduzidas e reamostradas para a visualização na escala requerida. Já no segundo, o WFS ou o WCS são mais adequados, por enviarem o dado bruto na resolução original e com o valor real de atributo, deixando a execução da operação de média a cargo do cliente. Dessa maneira, o software cliente poderia alternar automaticamente o serviço a ser usado, dependendo de qual operação será realizada, para que somente os dados necessários sejam enviados em cada caso.

Interfaces de acesso a serviços Web geoespaciais no padrão OGC foram implementadas em várias ferramentas de SIG (Sistemas de Informação Geográfica) tais como ARCGIS ¹⁰, QGIS ¹¹, dentre outras. No entanto, os serviços de processamento, embora funcionais, não escalam bem e exigem alto poder de processamento. Por isso, são insuficientes para construir serviços de cálculo de mapas online com baixo custo computacional e em tempo quase real. Isso ocorre por dois motivos principais: a sobrecarga do servidor e a necessidade de enviar grandes volumes de dados para a visualização dos resultados. Esses problemas são críticos para serviços Web geográficos de grande volume, dificultando a transferência para os clientes e resultando em processamento que pode levar horas ou dias.

Os artigos [26] e [35] apresentam questionamentos sobre os padrões da OGC. O primeiro trabalho analisa pontos positivos e negativos do WFS, comparando interoperabilidade, transporte e qualidade dos dados. Os autores destacam que, embora esses serviços viabilizem interoperabilidade e acesso a dados de múltiplos servidores, possuem limitações, como as do formato GML. Por ser extensível, o GML pode gerar inconsistências na interpretação, já que cada servidor pode estender seu conteúdo de forma diferente, causando incompatibilidades. Em relação à qualidade dos dados, no entanto, o serviço apresenta vantagens, permitindo a escolha entre formatos como SVG, que garantem a visualização correta de dados vetoriais em qualquer escala. No segundo estudo, é proposto um serviço chamado WTSS (Web Time Series Service) [7], que melhora o desempenho de acesso aos *pixels* em séries temporais. Ao requisitar um *pixel*, o cliente recebe os valores correspondentes a cada instante presente na série temporal, ao invés de receber apenas um valor. A proposta se justifica porque nenhum dos padrões da OGC é otimizado para esse tipo de acesso. No entanto, não permite processar mapa, por ser focada em acessar apenas um *pixel* e verificar sua evolução ao longo do tempo.

























Outros trabalhos propõem arquiteturas diferentes para o processamento e análise de mapas via internet. [34] sugere uma arquitetura de balanceamento de carga que busca maior escalabilidade e disponibilidade do serviço. Por outro lado, [29] discute decisões

¹⁰Esri ArcGIS 10.5 Leading Platform in OGC Compliance. 2017. Disponível em: <http://www.esri.com/esri-news/releases/17-2qtr/esri-arcgis-105-leading-platform-in-ogc-compliance>. Acesso em: 26 março 2018.

¹¹QGIS as OGC Data Client. 2018. Disponível em: https://docs.qgis.org/2.8/en/docs/user_manual/working_with_ogc/ogc_client_support.html. Acesso em: 10 outubro 2018.

Figura 2.9: Análise dos padrões frente aos requisitos.

Análise dos padrões frente aos requisitos

Atende ao requisito	Não atende ao requisito	Requisitos	WFS	WMS	WPS	WCS	WTSS
		Processamento interativo					
		Transferência particionada					
		Escalável					
		Resultado exato					

arquiteturais voltadas para a escalabilidade do servidor por meio da utilização dos serviços WFS e WCS. Nesse caso, o processamento dos dados é realizado pelo cliente, o que aumenta a escalabilidade, mas pode ser insuficiente devido à transferência de grandes quantidades de dados pela rede. Outra alternativa é o uso de serviços WPS distribuídos, que permitem o processamento de grandes quantidades de dados sem a necessidade de transferência dos arquivos originais. Essa abordagem utiliza várias máquinas trabalhando juntas e tem a vantagem de se concentrar no aumento contínuo da infraestrutura, já que depende de mais computadores ao invés de computadores mais potentes. Essa abordagem é virtualmente ilimitada, pois é proporcional ao número de nós de processamento na rede.

2.4 Serviços interativos de acesso a grandes volumes de dados geográficos

Uma dificuldade está em conhecer as limitações e capacidades de cada padrão de dados geográficos. Por isso, pretende-se analisar as utilizações que foram documentadas. Para isso, foram analisadas várias plataformas baseadas nos padrões da OGC, a fim de levantar as operações e interações possíveis, além de permitir conhecer as vantagens e desvantagens das decisões arquiteturais empregadas em cada serviço. Atualmente, existem diversos servidores de mapas, por isso, serão descritos com detalhes apenas os que apresentarem alguma propriedade relevante para o presente trabalho.

O Global Forest Watch (GFW) foi criado em 1997, com o intuito de divulgar com precisão dados das florestas do mundo¹². Esse projeto cresceu, e atualmente utiliza o Google Earth Engine [12] para processamento dos dados, que são servidos utilizando o

¹²Global forest watch. 2018. Disponível em: http://pdf.wri.org/wri_annual_review_2002_gfw.pdf. Acesso em 24 de janeiro de 2025.

serviço WMS. No GFW é possível enviar um polígono ou mesmo delimitá-lo na própria plataforma, e requisitar a análise sobre a área de desmatamento em uma área interativamente. Porém, o resultado dessa análise pode conter erros, o que é informado na própria plataforma, destacando que o erro é proporcional à escala de visualização do mapa.

O GeoTrellis [18] é uma biblioteca de processamento geoespacial de código aberto que oferece recursos avançados para a manipulação e análise de dados geográficos em um ambiente distribuído. Essa ferramenta é especialmente projetada para o processamento de dados geoespaciais no servidor, permitindo a execução eficiente de operações em grandes conjuntos de dados geográficos. Nele é possível realizar tarefas como manipulação de imagens, processamento de mosaicos, análise espacial e criação de serviços de mapas, fornecendo uma solução escalável e flexível para o processamento de dados geográficos em ambientes distribuídos. Esse serviço depende da distribuição do processamento e, por isso, não é suficiente para a resolução do problema proposto neste trabalho.

Outro servidor com funcionalidades únicas é o Map Of Life [16], criado em 2012. Esse projeto busca reunir múltiplas fontes de dados sobre biodiversidade em uma única plataforma, categorizando quanto à fonte e à espécie. Esse projeto também utiliza o Earth Engine [16] como ferramenta de análise, e os dados são exibidos utilizando o WMS. Esse estudo buscou identificar a correlação de ocorrência das espécies, a fim de permitir criar um mapa mundial relacionando a importância pontual de cada região à diversidade de espécies. Nessa ferramenta existem várias funcionalidades únicas, tais como filtrar os dados de uma área selecionada e ter análises mais detalhadas e adequadas para aquela região. Outra funcionalidade interessante é a utilização do WMS como proxy para obtenção do valor do *pixel*. No mapa de biodiversidade mundial, o usuário pode passar o mouse em qualquer *pixel* e obter o valor em tempo real e sem uma nova requisição ao servidor.

O MapBiomias [23], criado em 2015 com o intuito de divulgar anualmente o mapeamento do uso do solo e a distribuição dos biomas brasileiros, é mais uma plataforma com características de interesse. A plataforma realiza análises que consistem em etapas automatizadas e semiautomatizadas, que podem ser divididas em dois passos. Um passo é executado no Earth Engine [12], e o outro é executado por um software chamado Temporal Visual Inspection (TVI) [24]. A primeira etapa consiste na execução do algoritmo no Earth Engine. A segunda consiste em uma análise do índice de confiabilidade [24] sobre os valores gerados na primeira etapa, o que é feito com base em alguns pontos de controle definidos manualmente pelo usuário. Assim, baseado nos valores assumidos pelos pontos selecionados, é calculada a confiabilidade por região do mapa. Nessa ferramenta, existem algumas funções interessantes, tais como a possibilidade de filtrar as legendas que serão exibidas no mapa, permitindo, por exemplo, não exibir um ou mais biomas no mapa. Outra função possibilita visualizar o mapa em uma linha do tempo e poder indicar, para cada período, quais as áreas sofreram transição. Por exemplo, é possível

visualizar no mapa apenas quais as áreas que em 2012 eram vegetação nativa mas em 2015 foram transformadas em pastagem. Existe também a calculadora de carbono online ¹³. Nessa plataforma o usuário pode selecionar uma dentre as possíveis fontes de dados de carbono para uma área selecionada pelo usuário. O cálculo é executado na nuvem e, após o resultado, é gerado um relatório que pode ser impresso ou baixado no formato de tabela. Outro estudo [2] utilizou as informações da CCAL como uma das fontes de dados para poder avaliar a correlação do desmatamento com a existência de terrenos públicos na Amazônia.

A proposta desta dissertação se diferencia dos trabalhos apresentados por ser uma proposta que visa a utilização dos dados originais re-escalados e re-codificados para permitir o processamento em tempo real de grande montante de dados via internet. Para isso, é proposta uma arquitetura com divisão de trabalho entre servidor e cliente, de maneira que, para o cliente, ocorram em tempo real: a resposta do servidor, recebimento do dado e a execução da análise desejada. Essa distribuição é iniciada pelo servidor, que, ao receber uma requisição do cliente, inicia as operações de agregação das células, mantendo apenas as informações necessárias à operação requerida. Com isso gera-se uma versão contendo apenas as células necessárias ao cálculo desejado no cliente e todas as outras células tenham sido removidas. Nessa proposta, busca-se maximizar a possibilidade de utilização de cache no servidor com a possibilidade de reutilização dos mapas reescalados, além de diminuir o tempo de envio e processamento do dado no cliente, visando a execução em tempo real.

¹³CCAL - BRASIL, IPAM. Disponível em: <http://www.carboncal.org.br/view/brasil/>. Acesso em 26 de março de 2018.

Capítulo 3

Metodologia

Este capítulo descreve a metodologia adotada para a proposta de arquitetura. Um cenário ilustrativo é apresentado, no qual as operações são aplicadas para exemplificar o funcionamento de cada solução proposta. O principal destes requisitos é o fornecimento de um serviço em tempo real, já que a natureza dos dados geográficos implica em altos requisitos computacionais de processamento e de rede, mas ao mesmo tempo, esses dados devem ser processados e transferidos rapidamente. Outro objetivo crítico é garantir a exatidão dos resultados, o que implica em obter uma resposta idêntica para a mesma região de análise, independentemente da escala considerada. Além disso, a carga de trabalho no servidor deve ser minimizada para atender múltiplos usuários. Por fim, é importante garantir a similaridade com os padrões estabelecidos pela OGC nos formatos dos arquivos e na comunicação com o servidor. O conteúdo deste capítulo está dividido em cinco partes: estudo do problema, escolha das operações, definição dos operadores, definição da arquitetura e validação da arquitetura.

3.1 Estudo do problema

Na primeira parte, estudo do problema, foi realizado o levantamento das características dos serviços de disponibilização de mapas interativos que realizam algum processamento em tempo real. A análise de casos foi utilizada para identificar as funcionalidades e modos de uso de diversas plataformas. Plataformas com padrões oficiais da OGC e outras com extensões não oficiais foram analisadas, com o objetivo de conhecer as peculiaridades e os pontos fortes e fracos de cada padrão. Além disso, foi realizado o levantamento das implementações e eficiências dos padrões em ferramentas SIG, com o objetivo de elevar a compreensão das dificuldades e problemas técnicos em cada abordagem.

Requisitos de serviço em tempo real e de não ampliação da infraestrutura se mostraram impossíveis de cumprir utilizando apenas os serviços atuais. Por exemplo, os padrões WFS e WCS, que enviam os dados originais completos, que podem possuir vários

gigabytes de dados, porém o simples envio desse volume de dados já torna inviável sua utilização em uma abordagem de tempo real pela internet. Uma alternativa é proposta por [35], em que o cliente faz uma requisição ao serviço WPS e o servidor realiza todo o processamento. No final, apenas o resultado reescalado para visualização é enviado ao cliente. No entanto, essas requisições são iniciadas sob demanda, mas os mapas podem demorar para serem processados, fazendo com que a utilização dessa abordagem não resolva o problema. O padrão WMS apenas apresenta os mapas visualmente e não permite aplicar análises e processamento, pois apenas envia o mapa com os *pixels* representando as cores e não valores.

Apesar das limitações desses serviços, foi possível observar conceitos úteis para o novo serviço proposto, como o nível de zoom, a divisão do mapa em *tiles*. O nível de zoom define uma escala independentemente da escala original que é comum para todos os mapas. Todo mapa no mesmo nível de zoom terá a mesma quantidade de células na mesma região. As divisões do mapa em regiões pré-definidas, são chamadas de *tiles*, e facilita a obtenção de partes do mapa. Essas regiões têm sempre a mesma quantidade de *pixels* o que muda com o zoom é o tamanho de cada *pixel* e, por isso, variam também a área representada em cada *tile*. Um tamanho padrão amplamente utilizado é o de 256 linhas e colunas, e a quantidade de *tiles* e *pixels* representando o mundo em cada nível de zoom podem ser calculada, conforme [Equação 3.1] e [Equação 3.2] respectivamente. Dessa forma, é possível calcular o limite superior de tamanho do dado a ser enviado e processado no cliente, pois depende apenas do nível de zoom e da área requerida. Tornando possível estimar os custos computacionais de rede e tempo de execução das requisições no servidor.

$$Tiles = TilesX * TilesY = (2^{Zoom}) * (2^{Zoom}) = 2^{2*Zoom} \quad (3.1)$$

$$Pixels = 2^{16} * Tiles \quad (3.2)$$

Buscou-se alguma alternativa que combinasse o envio de poucas informações pela rede, o tempo de resposta baixo e a computação distribuída utilizando o próprio cliente, para garantir a qualidade do serviço frente a usuários simultâneos. Nessa nova abordagem, os dados originais devem ser pré-processados no servidor e o resultado deve ser salvo em cache. Que deve aplicar operadores predefinidos para gerar dados simplificados e otimizados para cada operação no cliente. Foi proposto utilizar o servidor apenas para pré-processar dados que pudessem ser enviados prontamente ao cliente como resposta da requisição. Esse dado vai ser pré-processado pelo servidor e deve ser abrangente o suficiente pra não precisar ser regerado a cada requisição, tanto para maximizar a eficiência de cache como minimizar o volume de dados enviados pela rede. O cliente, ao receber esse dado, deve poder executar a sua parte do processamento rapidamente. Com essa divisão, o servidor tem o custo de processamento amortizado pela utilização de cache, e

um resultado vai ser reutilizado em muitas operações diferentes. O cliente, por sua vez, vai receber um dado já parcialmente processado e não precisa requerer o mesmo mapa para aplicar outra operação, consumindo assim menos recursos de rede.

Foi decidido utilizar um formato similar às requisições do WMS e o processamento no servidor como o WPS, criando um novo serviço. Os comandos da API WMS foram mantidos e um novo parâmetro foi adicionado, o do identificador da operação que o tile vai representar. As diferenças para o WMS são em dois pontos principais. Primeiramente, os dados possuem os valores exatos adequados à escala visível ao invés de apenas as cores definidas na legenda. Em segundo lugar, a aplicação do pré-processamento como no WPS, os dados são simplificados antes do envio para minimizar as transferências na rede. Por exemplo, da API REST foram mantidos os parâmetros, nomes de comandos, e metadados, apenas adicionou-se um novo argumento, que indica o simplificador a ser utilizado. No entanto, existe uma grande diferença entre as respostas de ambos os serviços, uma vez que o WMS retorna os mapas já rasterizados com as cores da legenda, enquanto o serviço proposto retorna *pixels* cujos valores correspondem ao resultado da operação.

O processo de geração dos simplificadores deve ocorrer no servidor, ao processar os dados originais. Esse processo consiste em aplicar uma função ao mapa original, e essa função tem por objetivo garantir uma propriedade do mapa original. Essa propriedade garante que o mapa simplificado poderá substituir o dado original no cenário onde o usuário iria também aplicar a mesma operação àquele mapa. Por exemplo, se o usuário iria calcular a soma dos valores para cada região do mapa original, ele poderá dividir as mesmas regiões no mapa simplificado e o resultado deve ser o mesmo. O resultado desse processamento é armazenado em cache para ser reutilizado para responder a requisições subsequentes. Com isso, dois problemas são evitados: o dado original, potencialmente grande, não precisa ser enviado, contornando assim o gargalo de rede, e, além disso, o resultado pode ser reutilizado inúmeras vezes, atendendo a múltiplas requisições simultâneas. Após o processamento vem o segundo passo, o envio do dado. Esse deve acontecer de maneira rápida, pois são enviados os dados na escala de visualização e somente para a região requerida. Para envio do dado, foram avaliados os formatos de imagens que são mais amplamente suportados pelos navegadores, como JPG, GIF e PNG. Optou-se pelo PNG por ser o único dentre eles que atende a todas as necessidades deste projeto: compressão sem perdas, diferentes tipos de célula, e suporte dos navegadores. A compressão sem perda de dados é necessária porque permite a compressão do dado matricial sem alteração de valores. Além disso, esse formato permite a representação de diferentes tipos de célula ao se variar a quantidade e o tamanho das bandas, possibilitando tanto a representação de dados booleanos quanto de dados em ponto flutuante de precisão dupla.

O envio de mapas simplificados precisa garantir os requisitos de exatidão e de eficiência. O dado simplificado, portanto, deverá ser de menor volume que o dado original

e ainda assim permitir a obtenção do resultado exato ao limitar o uso ao contexto de escala e operação desejada. Para esse fim, modificou-se o processo de disponibilização dos mapas, os mapas serão pré-processados com algoritmos de agregação e o processamento dividido entre cliente e servidor. O método de agregação cria um dado simplificado, que é uma versão do dado pré-processado para uma operação e uma escala. Por definição, uma compressão sem perda é uma versão com menos dados, mas que permite recriar o dado original. No caso dos simplificadores, o resultado é o mesmo, se aplicadas as operações respeitando a escala do mapa reescalado. O resultado não é aproximado, mas sim um resultado exato. Esse conceito consiste na reamostragem do dado para cada cenário específico, ou seja, criar uma versão diferente do mapa para cada operação, escala e região. Dessa maneira, o dado pode ser comprimido mais eficientemente por descartar todos os dados não relevantes para a operação.

Por fim, a etapa da finalização do processamento no cliente. O dado deve ser interpretado para servir efetivamente de entrada para as operações da álgebra de mapas, que produzirão resultados exatamente iguais aos que seriam obtidos se executadas no mapa original na mesma região. O dado PNG recebido pelo cliente deve ser interpretado, pois cada *pixel* da imagem, na verdade, codifica um valor em quatro bytes, e não mais um conjunto de cores. Por isso, o cliente consulta o servidor para determinar qual o tipo de célula e como cada célula deverá ser interpretada. Essa informação está contida no metadado da operação no mapa. A requisição dos metadados deverá ocorrer uma vez por operação de mapa, pois após saber o tipo de célula da operação não é necessário fazer outra consulta ao servidor. Os tipos de células podem ser variados, porém para simplificação foram implementados apenas três: inteiro de 32 bits, inteiro sem sinal de 32 bits e ponto flutuante de 32 bits. Os outros formatos são interpretados e transformados em um desses três. Esse serviço pode ser utilizado independentemente da arquitetura e da terminação de memória da máquina, pois o servidor sempre envia o dado em big-endian, e basta que o cliente o interprete dessa maneira. Se uma máquina possui a arquitetura little-endian, basta que leia os bytes na ordem inversa para fazer a interpretação dos valores dos *pixels*.

3.2 Definição dos simplificadores

Na etapa de definição dos simplificadores são definidas as transformações que permitem alterar o nível de detalhamento do mapa, porém, mantendo a propriedade necessária para aplicar ao menos uma operação específica. Essa ideia remete a outro conceito do processamento de imagens, a pirâmide de imagens [1], pois ambos armazenam um dado em múltiplos níveis de escala. Como em um armazém de dados, poderia-se ter,

pré-calculadas, camadas de resoluções progressivamente menos detalhadas (1km, 2km, 4km, 8km, etc.). A premissa para a definição dos simplificadores é de que ao reduzir a quantidade de dados o resultado não seja alterado. Nos métodos de compressão sem perda para dados matriciais, como o Lempel-Ziv Welch (LZW) [37], são úteis para diminuir o tamanho do dado, permitindo a reconstrução do dado original. A taxa de compressão desse método pode ser extremamente variada pois depende do conteúdo e, apesar de útil e amplamente utilizado, não é suficiente para garantir envio rápido pela internet. Por isso, considerou-se a criação de um método similar a "pirâmide de imagens", que vai permitir reescalar o mapa antes do envio pela internet. O simplificador pode ser entendido como um tipo de "pirâmide de imagens" mais restrito, com basicamente duas regras adicionais. A primeira é que, na pirâmide de imagens, um *pixel* não precisa representar a totalidade reamostrada, enquanto que o simplificador cada pixel irá representar com exatidão o valor de toda a área reamostrada para cada simplificador, como: soma, mínimo, máximo, entre outros. A segunda é que o resultado da operação em qualquer escala, se aplicada a exata mesma região do mapa original e ao simplificado, o resultado deve ser exatamente o mesmo. Por exemplo, o simplificador de soma, que sua utilização permite calcular a soma da produção em uma região, instantaneamente, sem utilização de recursos adicionais do servidor. A diferença entre uma pirâmide de imagens e o simplificador é que, no primeiro caso, todo o valor da célula pode ser somada em qualquer sobreposição, enquanto que, no segundo caso é garantida que a soma seja exata e por exemplo precise ser ponderada pela área de cobertura e garantir que nenhuma parte do mapa seja somada duas vezes. Assim, somar células do mapa simplificado resulta no mesmo valor de somar a mesma área no mapa original, porém com a vantagem óbvia de ser uma operação muito mais rápida e de menor custo computacional.

Os simplificadores permitem pré-computar resultados dos mapas independentemente da escala e são funções aplicadas no servidor, para que apenas o dado simplificado seja enviado ao cliente. Esse método pode ser entendido como uma compressão com perda, pois a rigor não permite recriar o dado original [14], no entanto permite obter o resultado exato. Para isso, cada simplificador tem um contexto específico representando apenas uma operação em um contexto específico, por isso pode descartar os dados não essenciais e manter apenas os essenciais. Cada simplificador mantém uma propriedade distinta, representada pela sua operação, e mais três propriedades comuns e obrigatórias. A primeira é que qualquer operação precisa ser expressa na resolução do mapa simplificado, ou seja, ao especificar os limites de uma determinada operação não deve ser usado o detalhamento original, mas sim o nível de detalhes do zoom. Isso garante que a propriedade é mantida livre da escala, pois não tenta agrupar e nem dividir células a partir de um mapa já re-amostrado. A segunda regra diz que o resultado da operação aplicada à mesma região do mapa deve ser igual em qualquer nível de detalhamento do mapa. Já a terceira é de que toda simplificação precisa partir do mapa original e nunca de qualquer outro mapa

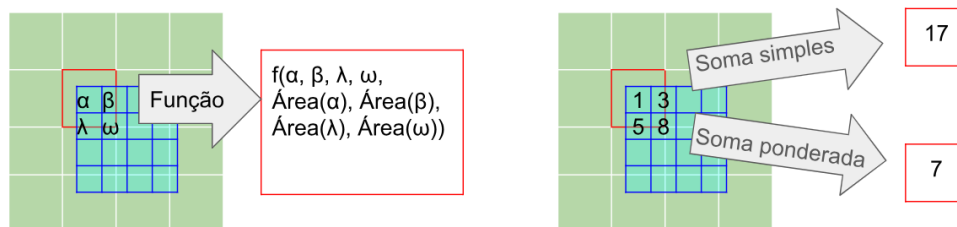
simplificado. Assim, qualquer operação executada no mapa de maior tamanho de célula ou mapa simplificado, terá o mesmo resultado caso a operação fosse aplicada no mapa de maior resolução, que é o mapa original, para uma região igual em ambos os casos.

Assim, apenas parte do dado é enviado ao cliente, que é todo o dado necessário, na escala da requisição, para gerar o mesmo resultado se utilizasse o mapa original. Além disso, esses simplificadores podem ser combinados para gerar operações complexas a partir da combinação de operações distintas. Por exemplo o cálculo da média por área ao combinar os simplificadores de soma e de área, essa nova operação pode ser executada da divisão do simplificador de soma pelo de área, e essa nova operação gerada dessa combinação, maximiza a reutilização dos resultados sem precisar de novas análises. Assim, garante-se a execução de uma análise sem erro, de maneira instantânea sem ser necessário processar o dado a cada nova requisição e portanto diminui o uso de recursos computacionais do servidor, como espaço em disco, transferência de dados e processamento.

Os rasters podem ser classificados em duas categorias distintas: categóricos e não categóricos. A diferenciação entre esses dois tipos é devido à representação dos valores armazenados em suas células. No caso dos mapas categóricos, cada valor representa a ocorrência de uma classe, que pode possuir características completamente distintas de outras classes no mesmo mapa. Nesse contexto, a tentativa de agrupar esses *pixels*, que podem ser contraditórios em um único tipo pode ser impossível. Portanto, geralmente não existem métodos de simplificação aplicáveis a esses dados. Por outro lado, os mapas não categóricos utilizam valores que representam a intensidade de um mesmo evento e, por essa razão, podem ser agregados calculando a intensidade resultante dessa combinação de intensidades. Por exemplo, mapas que representam a produção de madeira por região ou o estoque de madeira por região são exemplos de dados não categóricos. Em tais mapas, o valor de cada célula corresponde a uma quantidade e é fácil realizar a união desses valores por meio de operações de soma para obter o total na região. Além da soma outros métodos algébricos podem ser empregados para gerar mapas simplificados, que podem ser utilizados individualmente ou combinados em expressões algébricas.

Outra etapa importante foi o levantamento dos algoritmos candidatos a simplificadores, para isso, foram analisados os algoritmos que realizam alguma forma de agregação de células, para identificar quais funções mantêm alguma propriedade do dado independentemente do nível de detalhamento. Por exemplo, a agregação por meio da soma, além de outros métodos amplamente utilizados para reamostragem implementados em ferramentas de SIG, como o QGIS, ou mesmo em bibliotecas como o [9]. Dentre os métodos implementados estão: Cúbica, Lanczos, Vizinho mais próximo, Moda, Máximo, Mínimo, Mediana. Além desses, outros métodos também foram avaliados como: mapa integral e a soma ponderada por área que consiste em uma variante do algoritmo de soma, por sua vez, os detalhes de cada simplificador assim como do seu funcionamento foram descritas abaixo, focando principalmente na propriedade mantida de cada dado.

Figura 3.1: Funções de agregação: a soma simples cada valor é inteiramente somado, e na soma ponderada o valor é proporcional a área coberta. Por isso na soma simples a célula tem valor 17, resultante da soma das células tocadas 1, 3, 5 e 8, já a soma ponderada o resultado é 7 pois toca em metade das células 3 e 5, um quarto da célula 8 e completamente na célula 1.



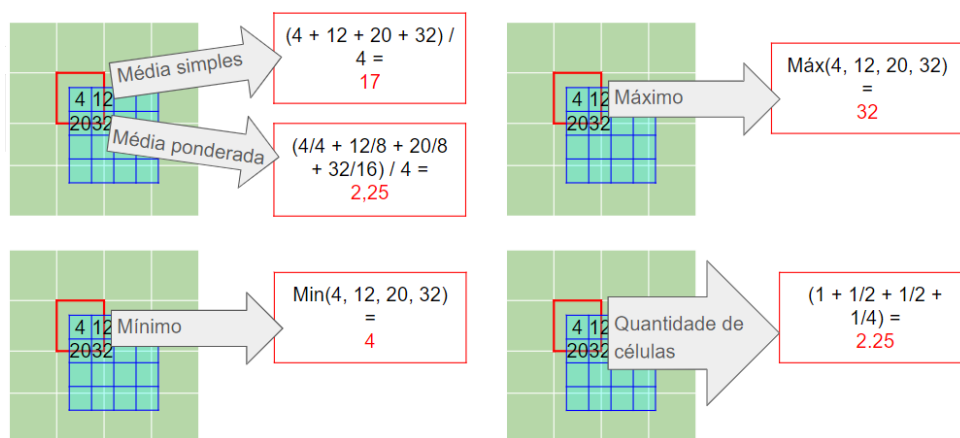
3.2.1 Tipos de simplificadores

Os simplificadores, como dito anteriormente, são funções de agregação que garantem certas propriedades durante a reamostragem do mapa. Essas operações são executadas por célula do mapa reamostrado mas sempre utilizam os dados originais como base para os cálculos. Cada simplificador é diferenciado pela sua função característica, que é aplicada às células da região em análise, como é ilustrado na [Figura 3.1], e a região de cálculo que limita as células envolvidas no cálculo de cada valor é conhecida pelo termo "janela". Essa janela, por sua vez, define os *pixels* do mapa original que devem ser considerados na geração do novo mapa, então uma função utiliza todas as células contidas na região e aplica a função simplificadora. A simplificação ocorre de maneiras distintas, pois as funções de simplificação garantem propriedades diferentes, mas, independentemente do nível de detalhamento do mapa a propriedade deve ser mantida válida. São descritas a seguir as funções simplificadoras propostas nesta abordagem.

3.2.1.1 Simplificador de Soma

Um método comumente implementado pelas ferramentas é o chamado soma simples ou não ponderada, que basicamente soma todas as células que tocam a janela de interseção. Dessa maneira, caso uma mesma célula seja tocada por múltiplas janelas o resultado é incrementado proporcionalmente à quantidade de toques, portanto aumenta-se o valor total da região. Por isso, a célula coberta mais vezes em partes por diferentes regiões tem o seu valor aumentado proporcionalmente à quantidade dessas interseções, e o resultado da soma, portanto, não pode ser usado como resultado simplificado de soma do dado original, uma vez que o total poderá ser completamente diferente ao total real do mapa

Figura 3.2: Exemplo de aplicação dos simplificadores de agregação, onde um conjunto de 4 células são agrupadas em um nova célula marcada de vermelho. São descritos os métodos de simplificação por média, máximo, mínimo, e quantidade de células.



[Figura 3.1]. Por isso, nessa pesquisa utilizou-se outro método, a soma ponderada por interseção com a janela, ou seja, cada *pixel* a ser somado é ponderado pela porcentagem de interseção que a célula ocupa em relação à janela de cálculo. Assim, independentemente do nível de zoom, o valor da célula sempre representa a totalidade dos valores que estão contidos naquela região.

3.2.2 Simplificador de Média

O método comumente implementado pelas ferramentas é o de média não ponderada, que calcula a média somando todas as células que têm alguma área interna à janela, e divide-se a soma pela quantidade de células. Porém esse método aumenta a influência de determinada célula com base na quantidade de vezes que ela é tocada, causando erro. Por isso, foi implementado um algoritmo que pondera a soma por seu percentual de área de interseção com a janela. Portanto, a janela avalia os valores das células que possuem alguma área comum, cada um desses valores é multiplicado por um valor entre zero e um, que representa a porcentagem coberta da célula. Assim, se a janela é coberta por um único valor esse peso é 1 porém se for 10% da célula o valor será 0.1, como ilustrado na [Figura 3.2]. Dessa maneira, todos os valores ponderados são somados para então dividir-se pela quantidade de células, garantindo que, independente do nível de detalhamento, a média na região seja mantida.

3.2.3 Simplificador de Máximo

Este simplificador utiliza a ideia de janela onde todas as células tocadas são avaliadas. Para isso compara-se todas as células a fim de encontrar a que possui o maior valor, é importante observar que a célula nula é ignorada, portanto qualquer valor exceto nulo é maior que nulo. Dessa maneira é garantido que não exista valor maior que o resultado em nenhuma parte da região simplificada, como é ilustrado na [Figura 3.2], e portanto, essa é a característica que será mantida em todos os níveis de detalhamento.

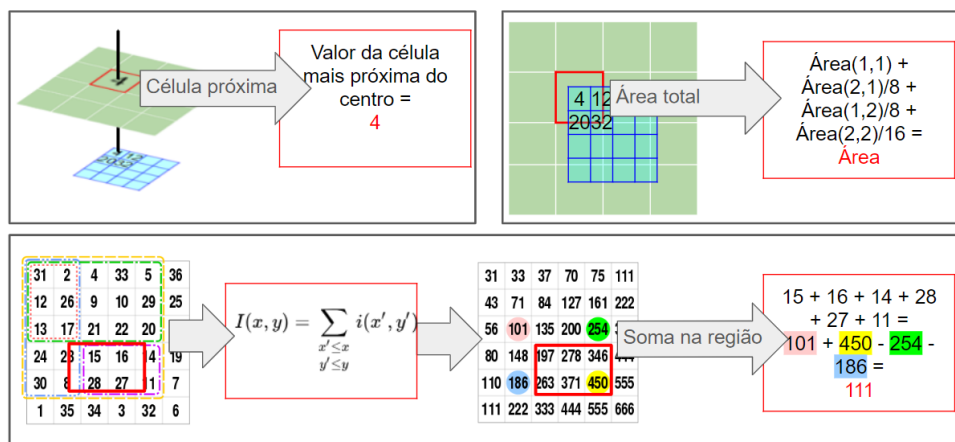
3.2.4 Simplificador de Mínimo

Este simplificador utiliza a ideia de janela onde todas as células tocadas são avaliadas. Para isso, contabiliza-se a quantidade de células internas à região. Nesse cálculo, a proporção de cada célula é usada como peso para a soma. Assim, as proporções de cobertura são somadas e o valor resultante é atribuído à célula correspondente, conforme ilustrado na [Figura 3.2]. Neste simplificador, ele não depende do nível de detalhamento, garantindo que a quantidade de células cobertas será precisa.

3.2.5 Simplificador de Quantidade de Células

Este simplificador utiliza uma janela para determinar o número de células do mapa original representadas em cada região do mapa simplificado. Para isso, conta-se a quantidade de células internas à região. Nesse cálculo, a porção de cada célula é usada como peso para a soma. As proporções de cobertura são somadas, e o valor resultante é atribuído à célula correspondente, conforme ilustrado na [Figura 3.2]. Neste simplificador, a ausência de detalhamento garante que sempre será considerada a quantidade de células cobertas.

Figura 3.3: Exemplo da aplicação dos simplificadores de cálculo de área, vizinho mais próximo, e soma.



3.2.6 Simplificador de Área

Este simplificador utiliza a janela para saber quais células uma determinada janela está tocando, e calcula a área dessa janela em termos das células do mapa original, utilizando o algoritmo de soma ponderada das áreas das células envolvidas, como ilustrado na [Figura 3.3]. Esse algoritmo basicamente atribui a cada célula não nula do mapa original o valor da sua respectiva área, então utiliza o algoritmo de reamostragem por soma ponderada. Portanto, a nova célula conterá a área da janela em termos do mapa original, de modo que, mesmo que o novo mapa possua erro em relação ao mapa original, os valores da área fiquem corretos por terem sido somados no mapa original.

3.2.7 Simplificador de Centro

O simplificador de centro é basicamente a utilização do método de reamostragem conhecido como 'nearest-neighbor' ou 'vizinho mais próximo' [30]. Nesse caso, o simplificador utiliza a proximidade do centro da janela dentre as células do mapa original, como ilustrado na [Figura 3.3]. Esse simplificador possui características únicas, pois nesse caso não se busca agrupar os dados em sua totalidade, mas apenas selecionar um dentre os possíveis. Por esse motivo, esse é o único simplificador aplicável a dados categóricos, já que a propriedade que é mantida é apenas de representação do dado original, ou seja, toda célula possui apenas um valor dentre os que estão na região e, portanto, dado um

valor de célula, é garantido que exista pelo menos uma célula com aquele valor naquela região.

3.2.8 Simplificador de Mapa Integral

Esse simplificador utiliza o método do mapa integral, que se baseia na soma dos valores de todas as células anteriores, atribuindo resultado a cada célula, como descrito em [5]. A construção desse mapa é dada a partir da [Equação 3.3], onde $i(x, y)$ é o valor da célula na posição x, y , e $\phi(x, y)$ é o somatório até aquela célula.

$$\phi(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y') \quad (3.3)$$

Esse mapa permite o cálculo da soma na área, calculando apenas diferenças entre os pontos extremos, como é ilustrado na [Figura 3.3]. Portanto, para saber a soma do valor total em uma região retangular não é necessário somar todas as células, basta avaliar quatro células, como descrito na [Equação 3.4].

$$\phi(x, y) = i(x, y) + \phi(x, y - 1) + \phi(x - 1, y) - \phi(x - 1, y - 1) \quad (3.4)$$

Porém, após criar esse mapa é preciso redimensioná-lo, garantindo que a propriedade será mantida em todos os níveis de zoom. Para isso é utilizada a função de reamostragem de máximo. Essa função garante que o valor do *pixel*, em qualquer escala, representa a soma de todos os valores até a última célula do mapa original na região delimitada.

3.3 Definição das Operações

Nesta seção são definidas operações com base em problemas comuns e soluções populares em álgebra de mapas, de modo a identificar operações que podem ser implementadas e garantir a maior quantidade de soluções possíveis. Além disso, pretende-se também entender as peculiaridades de cada operação para assim definir os operadores de maneira mais geral, especificando uma API. Essa API, por sua vez, precisa ter o máximo de abrangência com o menor conjunto de operadores possível, e reduzir a quantidade

de versões dos dados a serem criadas. Potencialmente, o dado tem uma versão para cada operação suportada, e ao definir operações genéricas é possível reutilizar a mesma operação mais vezes. Com isso, é diminuída a quantidade de espaço em disco necessária, além de diminuir também a quantidade de pré-processamento para a publicação do dado. Essa abordagem visa, portanto, identificar as operações a implementar e definir a melhor maneira de especificá-las de maneira a minimizar a carga no servidor, de forma que cada simplificador possa ser combinado com outros simplificadores para disponibilizar mais operações da álgebra de mapas para o cliente.

É importante esclarecer as diferenças entre "operação" e "simplificador" no contexto desse trabalho, uma vez que são conceitos diferentes porém complementares. Os simplificadores são as transformações executadas sobre os dados originais que ocorrem no servidor, pois servem para simplificar os dados e permitir a transmissão e utilização rápida pelo cliente. Já as operações são funções aplicadas aos dados simplificados, no lado do cliente, nos mapas resultantes do processo de simplificação. No cliente, o mapa simplificado pode ser o alvo de qualquer operação algébrica simples, porém são definidos operadores que permitam novas interações com os dados, e também especificar formas de utilização dos simplificadores no processamento dos dados. Por isso, as operações são executadas no cliente, enquanto o servidor simplifica os dados. Essa divisão permite a utilização do processamento de ambos os agentes, que inclusive podem ser combinadas. Detalhes sobre as operações e sua combinação são descritos a seguir.

3.3.1 Tipos de operadores

Os operadores foram categorizados quanto à sua funcionalidade e característica. Os operadores utilizam um ou mais simplificadores, portanto existe ao menos um operador para cada simplificador. Cada operação representa uma maneira de interação com o dado. Por isso, temos as seguintes operações simples, que utilizam apenas um simplificador: soma, média, máximo, mínimo, quantidade de células e área da célula. Utilizando a combinação de simplificadores outras operações são possíveis, como filtro baseado em média, cálculo da variação entre maior e menor valor na região, cálculo da produtividade por área, entre outros.

Por exemplo, considere-se um cenário em que, num mapa de estoque de carbono, deseja-se filtrar as regiões que possuam proporção acima de um limiar definido pelo usuário, de maneira que somente essas regiões fiquem visíveis. Essa operação pode ser executada no cliente, sem nenhum processamento adicional do servidor pois, ao combinar dois simplificadores, é possível obter um terceiro que garante uma propriedade derivada

dessa combinação. No caso, pode-se agrupar os simplificadores de soma e de área para calcular a proporção de carbono para cada célula, e num segundo passo basta aplicar a operação de filtragem nos valores da proporção para gerar o filtro de produtividade. Dessa maneira, utilizando apenas o processamento no cliente, uma nova operação é possível, pois criou-se um mapa interativo que permite a filtragem de limiar de carbono por área. Esses cálculos podem ser executados em tempo quase real, uma vez que o dado foi simplificado e otimizado para aquela operação, para a escala e região de análise.

A seguir são descritos os operadores, definidos em função de argumentos que o usuário poderá usar em diferentes execuções. No entanto, a definição dos operadores não é exaustiva, ou seja, não esgota todas as possíveis operações, mas buscou-se apresentar um conjunto representativo necessário para a implementação de operações bastante utilizadas. É importante lembrar também que cabe ao usuário identificar qual operação é possível em cada caso, pois os operadores garantem que a utilização dos simplificadores será feita de maneira correta com o correto uso de suas propriedades. No entanto, nesta pesquisa não se buscou fazer a identificação completa de quais operações fazem sentido em um mapa ou em outro. Portanto, cabe ao usuário o conhecimento do dado, não sendo necessário conhecimento do algoritmo de simplificação, pois o usuário deve identificar qual é a melhor operação para determinada finalidade, e também identificar quais operações fazem sentido em um mapa mas não fazem sentido em outro.

3.3.1.1 Operador de Soma na região

Esse operador utiliza o simplificador de mesmo nome para somar regiões do mapa. Ele permite selecionar uma área qualquer do mapa e calcular o somatório, de maneira que o resultado dessa soma será igual à execução da mesma operação no mapa original. A função óbvia aplicada é a soma dos valores de célula na região, e que por isso utiliza o simplificador de soma, garantindo que a soma das células simplificadas representa o dado original. Para implementação dessa operação, é preciso evitar a soma indiscriminada de valores muito grandes com valores muito pequenos pois existe a tendência dos valores pequenos se perderem, por esse motivo é recomendável a utilização de algum algoritmo para compensação de erro, como por exemplo o [17]. Esse método basicamente soma os valores pequenos separados dos valores grandes, e somente soma o valor pequeno ao valor grande quando ele se torna relevante e não será mais ignorado.

3.3.1.2 Operador de Média na região

Esse operador utiliza o simplificador de mesmo nome para calcular o valor da média em uma região do mapa. O usuário previamente define a área de análise desenhando diretamente no mapa a região onde os valores serão calculados. A média dos valores contidos no mapa simplificado será igual à obtida ao aplicar a mesma operação no mapa original. Para realizar essa operação, é utilizado o simplificador de média, e o resultado é calculado em tempo quase real. A execução desse operador consiste em somar todas as células internas à região definida e calcular a média, dividindo o valor total pela quantidade de células na região selecionada.

3.3.1.3 Operadores de Máximo e Mínimo na região

Esses operadores funcionam de maneira semelhante, apesar das operações serem diametralmente opostas pois utilizam o simplificador de máximo e mínimo respectivamente. Ao se utilizar um desses operadores é preciso aplicar a mesma função porém sobre o mapa simplificado, assim, torna-se possível calcular o máximo e o mínimo de uma região em tempo quase real de uma maneira interativa.

3.3.1.4 Operador de Quantidade de Células na região

Esse operador permite saber em uma região qualquer, a quantidade de células do mapa original que estão sendo representadas. Essa operação utiliza o simplificador de mesmo nome e soma todas as células da região, de maneira que o resultado é a quantidade total de células na área especificada.

3.3.1.5 Operador de Área da região

Esse operador permite calcular a área de uma região arbitrária no mapa simplificado de maneira que essa região seja exatamente igual à mesma região no mapa, sem por sua vez adicionar erro ao dado na projeção original. Esse aspecto se deve à maneira como os valores são calculados, pois é calculada a área de cada *pixel* do mapa original e esse é o valor usado na simplificação através da soma ponderada, portanto cada célula do mapa resultante contém exatamente a área relativa ao mapa original em hectares que equivale a mil metros quadrados.

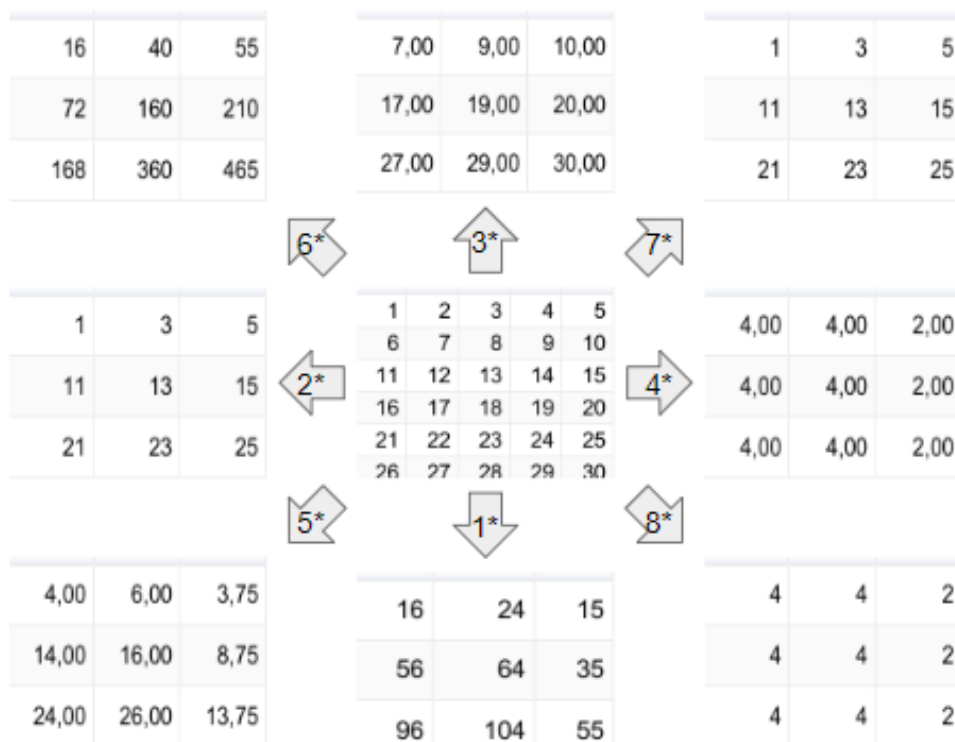
3.3.1.6 Operador de Álgebra

Para cada célula do mapa, o valor pode ser combinado utilizando operações algébricas de mapas. Esse operador permite realizar operações do tipo local, que utilizam o valor de cada pixel na mesma posição e o combinam de acordo com a expressão algébrica definida. O operador pode ser aplicado em qualquer mapa e com qualquer simplificador, desde que o resultado seja um mapa ou uma tabela. Além disso, o valor retornado pela tabela também pode ser reutilizado.

3.3.1.7 Operador de Filtragem

Esse operador avalia se o valor do mapa é maior menor ou igual a um determinado limiar. Se a avaliação da expressão for verdadeira retorna o valor da célula e caso contrário retorna o valor de nulo para a célula. Dessa maneira, é possível filtrar os valores de células que não devem ser usadas nos cálculos. É importante salientar que a filtragem deve ser usada com cuidado, pois não é equivalente à uma consulta a um banco de dados, uma vez que os dados que estão sendo filtrados já passaram por um processo de transformação e por isso é importante lembrar que somente a propriedade do simplificador é mantida. Por exemplo, o mapa de máximo só irá possuir os máximos mas não possui os mínimos e nem quaisquer outros valores de células que não fossem o máximo de cada célula simplificada.

Figura 3.4: Comparação entre o resultado da aplicação de diferentes simplificadores: 1º Soma, 2º Média, 3º Máximo, 4º Quantidade de células, 5º Mínimo, 6º Mapa Integral, 7º Vizinho mais próximo, 8º Média



3.3.1.8 Operador de transformação, soma por célula em distribuição espacial

Este operador permite a transformação entre dois tipos de mapas: um mapa que apresenta o valor total por célula e outro mapa que apresenta o valor médio por área. Por exemplo, considerando um mapa de produção de borracha que apresenta a quantidade de toneladas de borracha produzidas na região de cada célula, é possível aplicar simplificadores de soma e área por região para obter um mapa de produtividade por área. Esse resultado é obtido ao dividir a produção total na região pela área correspondente, resultando na distribuição do valor por área. Além disso, há também a possibilidade de calcular o total na região ao multiplicar a média pela quantidade de células.

3.4 Arquitetura

Propõe-se uma arquitetura cliente-servidor em que cada requisição de consulta seja executada em duas etapas, uma no servidor e outra no cliente. Que de maneira geral deve seguir os seguintes passos. Primeiro o cliente especifica a escala, o simplificador, o formato de saída, a área de interesse e faz uma requisição ao servidor. Que recebe esse pedido e retorna os dados simplificados, reescalados e divididos em tiles em um formato conhecido pelo cliente e salva o resultado no cache. Com isso pode responder imediatamente ao receber uma requisição. Optou-se por estender um servidor que implemente os padrões da OGC que já implemente os serviços WMS, WPS, WCS e WFS para validar a proposta. A comunicação com o WMS é feita pela interface REST [8] possibilitando o acesso direto aos arquivos, legendas e outras metainformações. As legendas são definidas no formato *Styled Layer Descriptor* (SLD) [21], que é uma linguagem de marcação baseada no XML para descrever as camadas de dados matriciais e vetoriais. Dessa maneira, criou-se um serviço externo que pode ser integrado a qualquer implementação do WMS desde que tenha o acesso ao mapa original e a legenda. Na nova arquitetura, diferentemente do WMS, onde os *pixels* representam cores e ocupam 1 byte para cada componente de cor [3], nessa abordagem, os *pixels* armazenam valores numéricos. Para isso, foi desenvolvido um novo padrão baseado no WMS, com parâmetro adicional dizendo o simplificador e que envia o dado com os valores das células, mas que mantém o nome dos comandos. Utilizou-se também conceitos como o de distribuição dos dados, o envio em tiles e a utilização de metainformações para descrever os dados. Por exemplo, o comando *GetCapabilities* para listar os mapas e seus metadados, e o *GetMap* para realizar o download do mapa na região de interesse. No novo serviço foi adicionado o parâmetro na requisição para a obtenção dos dados na nova arquitetura mantendo a compatibilidade com a API do WMS. Pode-se inclusive utilizar o *GetCapabilities* do WMS para listar os mapas disponíveis no novo serviço. Dessa maneira, após a publicação, o mapa é fornecido em ambos os padrões, além de garantir que cada mapa possua uma identificação única entre ambos os protocolos.

Em ambos os serviços, as requisições podem ser feitas em regiões do mapa, permitindo obter cada parte independentemente, pois é possível dividir em tiles contendo apenas as partes relevantes de um mapa, ignorando as partes inúteis. Isso possibilita visualizar uma área específica do mapa com muito detalhamento sem a necessidade de receber todo o restante no mesmo nível de detalhes. A maior diferença entre os dois serviços é referente ao conteúdo, pois no WMS cada *pixel* representa uma cor, que é resultado da aplicação da legenda ao dado original, enquanto que no novo serviço cada *pixel* representa o valor real do *pixel*. Portanto, na nova abordagem o cliente pode utilizar os valores reais dos *pixels* em cálculos para a combinação de múltiplos mapas, porém fica responsável pela geração da legenda da maneira que for mais conveniente. Por isso, criou-se

simplificadores que transformam os dados no servidor, a fim de permitir a aplicação das operações no cliente como se tivessem sido executadas nos mapas originais. É importante que o cliente possa definir quais operações e variáveis serão utilizadas em cada consulta. Como já mencionado, o dado tem uma versão específica uma para cada nível de zoom e simplificador. Desse modo, o cliente requisita os dados simplificados que são necessários, e por sua vez o servidor envia apenas a versão mais adequada. Gerar simplificadores tem elevado custo computacional, por isso a utilização do cache é tão importante, de modo a evitar o retrabalho e ampliar a escalabilidade. A escalabilidade desse serviço pode ser facilmente feita ao se distribuir o cache em diferentes nós. A exemplo dos serviços WMS, a primeira requisição é mais demorada, e ao término do primeiro processamento é garantida resposta instantânea.

A implementação do serviço envolverá o processamento, o armazenamento e o envio dos dados ao usuário. Na primeira etapa, o pré-processamento dos dados é realizado no servidor, gerando versões dos dados para cada escala e simplificador. Esse processo é computacionalmente custoso e envolve a simplificação dos dados para cada simplificador, com os resultados sendo armazenados. Em seguida, os dados pré-processados e resumidos são enviados ao cliente, que pode aplicar as operações desejadas. O cliente recebe os dados com os valores das células e pode realizar expressões no mapa recebido, combinando os dados com diferentes valores e operações algébricas em tempo quase real. Isso resulta em tempos de resposta imediatos, pois os resultados das requisições já estão prontos e o processamento no cliente pode ocorrer quase instantaneamente.

É importante observar que o armazenamento e o custo de processamento no servidor são proporcionais à quantidade de versões disponibilizadas do mapa. O pré-processamento dos dados permite uma redução significativa no custo computacional no servidor, uma vez que cada versão dos dados é processada apenas uma vez. Em geral, o cache já está presente e, ao receber uma requisição, o resultado previamente calculado e armazenado no servidor pode ser enviado imediatamente. Somente em casos excepcionais, nos quais os dados simplificados não estão em cache, inicia-se a geração da resposta, que é então armazenada em cache. O armazenamento e o processamento aumentam proporcionalmente à quantidade de versões do mapa. Portanto, é recomendável limitar a quantidade de projeções suportadas, por exemplo, selecionando apenas uma.

3.4.1 Definição do padrão de requisições da API

A comunicação com o serviço é baseada em requisições REST, que é utilizada em todas as etapas: obter informações descritivas dos dados, obter os dados e também iniciar

processamento dos dados. O novo serviço utiliza o banco de dados dos mapas publicados no WMS, portanto a requisição para a listagem dos mapas publicados e disponíveis é feita primeiramente ao WMS, por meio do comando `GetCapabilities`. Esse comando, de acordo com a própria especificação, obtém metainformações sobre o serviço, incluindo as operações e parâmetros, além de uma lista das camadas disponíveis¹. No entanto, os mesmos mapas disponíveis para o WMS são disponíveis para o novo serviço, portanto com o resultado dessa operação o usuário sabe quais mapas estão disponíveis para serem utilizados e pode requerer para qualquer um deles quais simplificadores estão disponíveis. Estas requisições se assemelham às padronizadas pelo WMS, e o novo padrão contém os comandos a seguir.

3.4.2 Describe

Este comando descreve as operações que estão disponíveis no cache, para todos os mapas publicados. Se acionado sem parâmetros, retorna todos os simplificadores de todos os layers. Pode conter dois parâmetros: o primeiro, "LAYERS", deve ser o identificador único do mapa em formato string. O segundo, "STYLE", obtém o índice da camada a ser analisada, que começa em 1. Portanto, a primeira camada do layer tem o índice um e não o índice zero. O retorno dessa requisição é um objeto JSON, no qual cada mapa é representado por um array e cada operação é representada por quatro elementos na seguinte ordem: tipo de célula, valor nulo do mapa, identificador da operação e identificador da camada utilizado. Os tipos de célula suportados atualmente são de até 32 bits, sendo int para representar inteiros de 32 bits, float para representar ponto flutuante de precisão simples, uint para representar inteiro sem sinal, além dos tipos byte e bit. O valor nulo é usado para saber qual *pixel* será interpretado como nulo no mapa, o que permite ao cliente identificar um *pixel* como sendo desprovido de conteúdo. Já o descritor da operação é uma string que identifica unicamente cada operação e informa qual simplificador cada entrada do metadado descreve. O último parâmetro é o indicador da camada utilizada no processamento, que descreve o índice do layer que foi utilizado no simplificador.

Método GET

URL /describe/

Parâmetros LAYERS, STYLE

¹OpenGIS Web Map Server Implementation Specification. 2006. Disponível em: https://portal.ogc.org/files/?artifact_id=14416. Acesso em: 23 de janeiro de 2025.

Exemplo de requisição `./describe/?layers=CSR:rent_brazil_nut&style=1`

Exemplo de resposta `["float",-9999","raw","1"]`

3.4.3 GetMap

Essa operação é responsável por obter um tile do mapa simplificado, e possui o mesmo nome da requisição do WMS. Essa requisição tem similaridades no funcionamento e finalidade em ambas as abordagens, mas nesta proposta é utilizada para a obtenção do tile simplificado do mapa para a operação requisitada. O dado é sempre enviado no formato de imagem. O formato escolhido foi o PNG, e as informações de georreferenciamento são provenientes da própria requisição. Uma vez que cada dado é requisitado para um determinado limite, pode-se atribuir esse mesmo limite ao tile. Essa requisição possui os seguintes parâmetros: `layers`, `style`, `bbox` e `operation`. O parâmetro `"layers"` recebe o identificador do mapa a ser processado, e o parâmetro `"style"` é o identificador da camada a ser utilizada, considerando a primeira camada como índice 1. O parâmetro `bbox` define a região da requisição definida por dois pares de coordenadas no sistema de projeção do mapa (mínimo x, mínimo y, máximo x, máximo y, onde x e y são coordenadas no sistema de projeção do mapa). Através desse limite é interpretado também o nível de detalhamento, pois independentemente da área coberta, o resultado sempre possui tamanho fixo, e assim o sistema consegue identificar o tile que deve ser utilizado. Outro ponto importante é que, apesar do formato PNG não possuir georreferenciamento, a própria região da requisição define a região que aquele tile representa. O parâmetro `"operation"` indica o simplificador do mapa a ser requisitado. Essa requisição é descrita abaixo:

Método GET

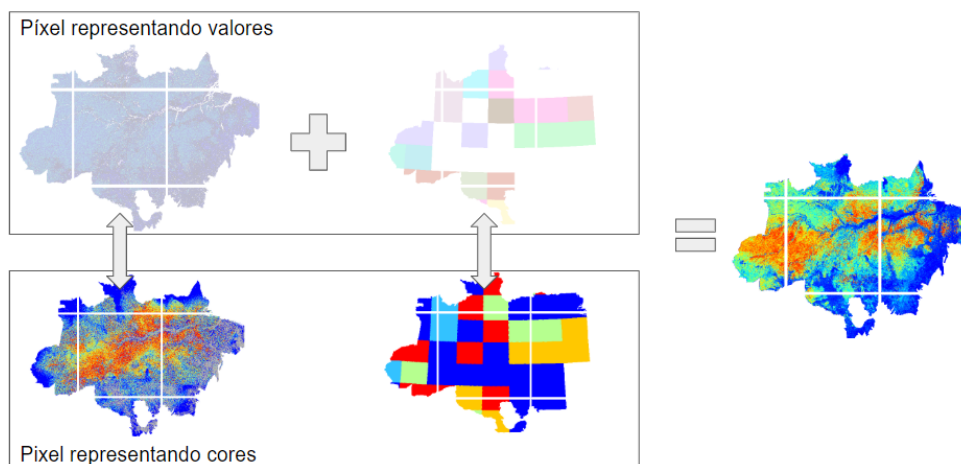
URL `/calc/OPERATION/?`

Parâmetros LAYERS, STYLES, BBOX e OPERATION

Exemplo de requisição `./calc/raw/?LAYERS=CSR:change_revenue_soy
&STYLES=1
&BBOX=-7514065.6275,-2504688.5425
,-5009377.085,4.6566128730774e-10`

Exemplo de resposta Ilustrada na [Figura 3.5].

Figura 3.5: Imagem representando uma parte do mapa PNG, com os *pixels* codificando valores numéricos e não cores.



3.4.4 RunModel

Essa operação permite executar um modelo utilizando os parâmetros da requisição. Por meio dela é iniciado o processo de geração do tile. Esse processo interpreta os parâmetros da requisição e carrega o script referente à operação que foi requerida. Os parâmetros são o identificador do mapa, identificador da camada e operação a ser executada. O identificador é o nome utilizado pelo WMS mas sem o workspace; por exemplo, para o identificador "CSR:prodes" deve-se enviar apenas "prodes". Além disso, o parâmetro operação identifica unicamente a operação e por sua vez o modelo a ser executado, como definido abaixo:

Método GET

URL /calc/run/

Parâmetros I_BASEMAP, STYLE, MODELNAME e OPERATION.

Exemplo de requisição ./calc/run/?i_basemap=prodes&styles=1
&modelname=map_general&i_operation=0

Exemplo de resposta Ilustrada na [Figura 3.5]

3.4.5 Codificação e decodificação dos valores dos *pixels* da imagem

A codificação dos *pixels* no formato PNG utiliza quatro camadas: três destinadas às cores e uma à transparência. Cada camada possui 8 bits, totalizando 32 bits por célula, o que permite armazenar valores inteiros ou de ponto flutuante de precisão simples. Os valores são gravados no formato little-endian, em que os bytes menos significativos ocupam as posições de menor ordem. No formato PNG, os bytes seguem a sequência RGBA. O byte do vermelho representa o menor valor, enquanto o byte do alpha representa o maior. Em arquiteturas big-endian, é necessário reordenar os bytes, colocando o alpha no início e o vermelho no final. A decodificação desses valores ocorre no cliente. Quando o cliente é um navegador, essa operação é realizada em Javascript, utilizando ponto flutuante duplo. Essa abordagem assegura precisão suficiente para evitar erros na representação dos dados.

Capítulo 4

Validação

A fim de validar a arquitetura, comparou-se a eficiência das diferentes abordagens. As métricas escolhidas para isso foram o tempo de processamento, quantidade de dados processados, tempo de transferência, uso de recursos computacionais e a possibilidade de pré-processamento em cada abordagem. Os testes são executados em duas máquinas diferentes, o servidor naturalmente com mais recursos computacionais que o cliente. O servidor com o hardware: quatro processadores Intel(R) Xeon(R) modelo E7-8837 de oito núcleos físicos de 2.67 Giga hertz, memória de 64 Gigabytes e sistema operacional Windows 2008 Server. Já o cliente, um notebook, com as especificações: processador Intel I7-4510U com 2 núcleos físicos de 2.0 Giga hertz e 16 Gigabytes de memória e Windows 10 Home. Os tempos calculados vão seguir a ordem que ocorrem, portanto, pré-processamento anterior a requisição, seguido da requisição do mapa pelo cliente e finalizada com a obtenção do resultado. Os dados coletados vão ser usados para a identificação dos pontos positivos e negativos de cada alternativa. Dessa maneira, o processo foi dividido em cinco etapas: requisição dos dados, processamento, transferência e visualização do resultado.

Os dados utilizados na pesquisa são provenientes das bases do CSRMaps¹, referentes aos anos de 2013 e 2018, e foram coletados de diferentes projetos para evitar viés nos testes de performance. Esta base de dados está disponível publicamente online, possui documentação e foi validada pelo CSR. Estes dados possuem categorias como: zoneamento do uso da terra, recursos naturais, infraestrutura, divisão política, entre outros. Utilizou-se dados oficiais do Centro de Sensoriamento Remoto da UFMG² e resultados de projetos como: AMAZONES, uma pesquisa sobre valor dos serviços da Amazônia³ e o projeto FIP, que monitora a ocorrência de fogo no Cerrado⁴. Para garantir uma análise mais ampla, foram selecionados dados de diferentes regiões do Brasil e de Moçambique. No total, foram escolhidos sete mapas com características distintas, incluindo: volume, resolução, tipo de célula, e área de cobertura, como ilustrado na figura 4.1. O volume dos dados variou de 6 milhões até 2 trilhões de células, a resolução variou de 30 a 500 metros

¹CSR Maps, CSR. Disponível em: <https://maps.csr.ufmg.br>. Acesso em: 20 de maio de 2019.

²Centro de Sensoriamento Remoto, CSR. Disponível em: <https://maps.csr.ufmg.br>. Acesso em: 20 de maio de 2019.

³AMAZONES - AMAZON ECO SERVICES, CSR. Disponível em: <http://maps.csr.ufmg.br/calculator/?queryid=106>. Acesso em 16 de outubro de 2018.

⁴FIP, CSR. Disponível em: <http://csr.ufmg.br/fipcerrado>. Acesso em: 20 de maio de 2019.

Figura 4.1: Características diversas dos mapas escolhidos.

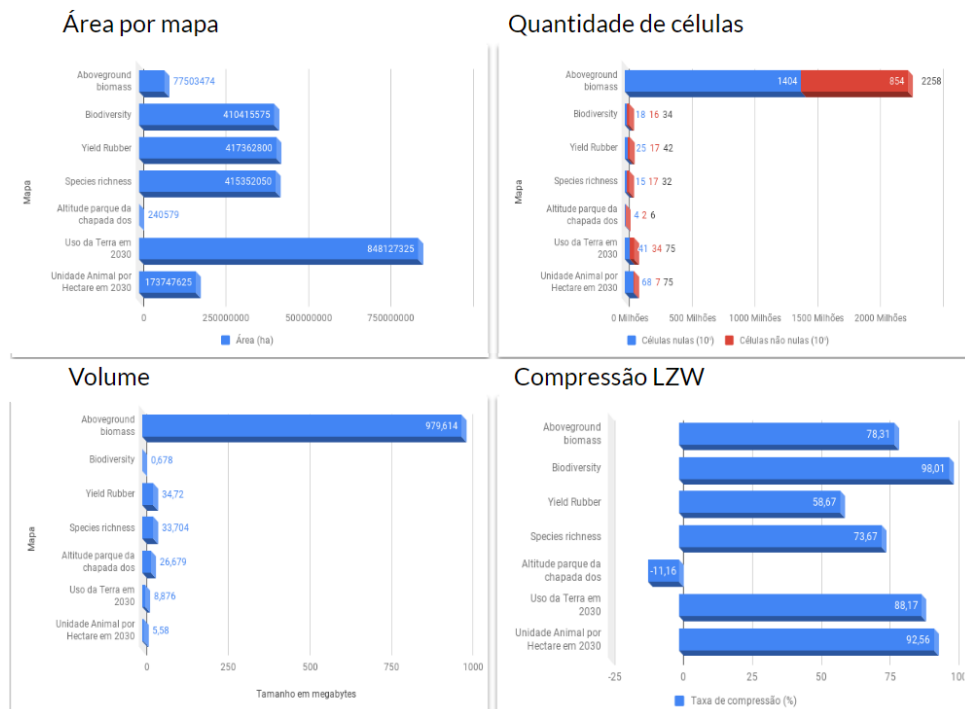
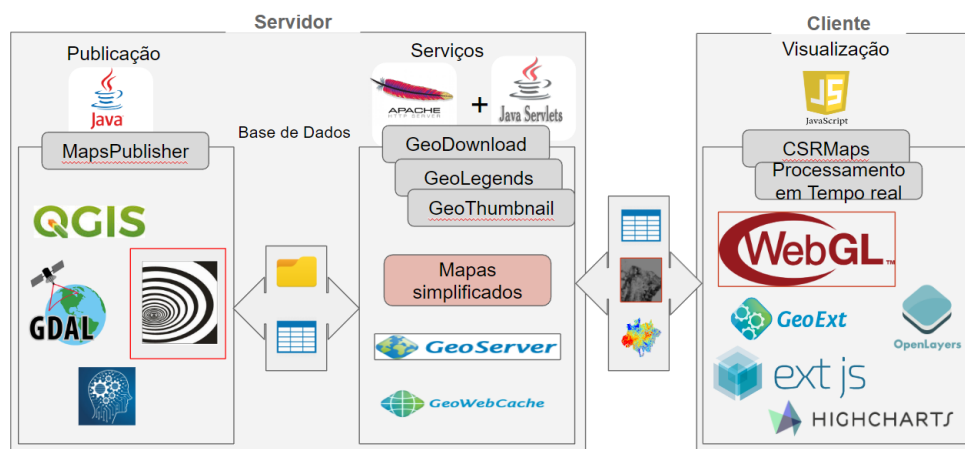


Figura 4.2: Exemplo de componentes utilizado na implementação dessa arquitetura.



e os tipos de células variaram de inteiro de 8 bits sem sinal a 32 bits de ponto flutuante. Para simplificar essa análise, todos os dados estarão na mesma projeção "EPSG:4326", compatível com servidores de mapas e software de SIG [13]. O objetivo dessa validação foi demonstrar as vantagens da arquitetura proposta e justificar as decisões tomadas, além de apresentar os resultados obtidos com a implementação desse serviço em aplicações em tempo real.

Para a avaliação dos serviços da OGC optou-se pelo servidor de mapas Geoserver [25] que além de amplamente utilizado já implementa os serviços WMS, WPS, WCS e WFS. Sendo o WMS utilizado na visualização dos dados, o WPS para processamento no servidor além dos WCS e WFS que permitem processar os dados no cliente. Essa implementação já vem com um módulo integrado chamado GeoWebCache que otimiza

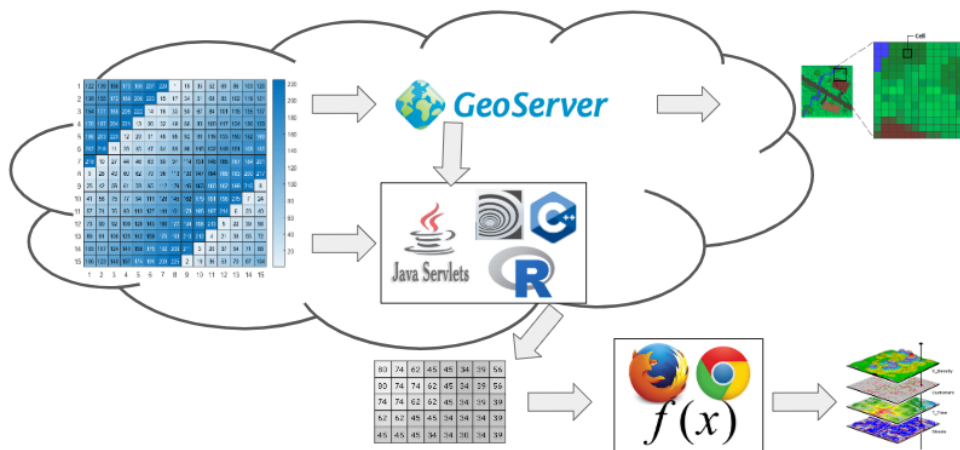
a performance do WMS com a utilização do cache. Para o cliente dessas abordagens vamos utilizar o QGIS para avaliar os serviços WPS, WCS, WFS e WMS. Já para a implementação da arquitetura proposta nesta dissertação, optamos por implementar apenas o controle de requisições na linguagem Java com servlets *servlets* e o processamento das imagens no software Dinamica EGO [31]. Nesse software que serão realizadas as transformações, quando necessárias como as simplificações dos dados matriciais. Esse software é feito para automatização e processamento de dados geográficos e simulações em formato raster. Ele possui funtores, que representam funções de transformação do dado, que podem ser encadeados a fim de se criar uma rotina de transformação dos dados com alta performance. Esses funtores são representados visualmente em um diagrama de fluxo que cada passo representa uma transformação e que os dados transformados podem ser salvos. Os funtores oferecem uma grande variedade de possibilidades para processar dados matriciais, tabelas, imagens, além de poderem ser implementados nativamente em outras linguagens de programação como R e Python. Esses fluxos pré-definidos podem ser salvos para gerar um script que pode ser aplicado a outros mapas, por exemplo para gerar os mapas simplificados, divididos em tiles.

O caminho da requisição pode ser descrito seguindo em sequência os seguintes passos. Por regra, o cache já vai existir e, ao receber uma requisição, o resultado previamente calculado e armazenado no servidor pode ser enviado imediatamente. Nos casos excepcionais, onde o dado simplificado não esteja no cache, o Java vai instanciar o Dinamica EGO com um script pré-definido. O cliente nesse trabalho foi implementado no navegador, utilizando a linguagem Javascript. E pode executar as análises desejadas em tempo real e de maneira interativa. Todo o processamento de dados e a geração da legenda no cliente foram implementados nessa linguagem, e, portanto, podem ser facilmente aplicadas localmente por qualquer navegador, para quaisquer operações, mapas e variáveis. Um ponto importante é notar que o custo computacional de gerar um mapa simplificado é extremamente elevado, podendo demorar várias horas, por isso a utilização do cache é de extrema importância, de modo que ao final da primeira resposta todas as seguintes serão instantâneas.

4.1 Processamento e publicação dos mapas utilizando o Dinamica EGO

O software Dinamica EGO permite a criação de modelos para o processamento de dados geográficos por meio de uma linguagem visual. Nele é possível implementar

Figura 4.3: Composição arquitetural do serviço demonstrando as etapas no servidor e no cliente. No servidor o dado é publicado no Geoserver e é processado por um conjunto de softwares. No cliente o dado otimizado do processamento do servidor é usado para aplicar a equação e gerar a representação visual do dado.



qualquer análise e transformação dos dados matriciais, pois permite processar mapas com muitos dados, mesmo em computadores pessoais, além de permitir também ampla utilização de arquiteturas paralelas. Esse software é amplamente utilizado para pesquisas em grandes volumes de dados, e foi utilizado em inúmeras publicações. Por esses motivos, e pelo conhecimento prévio da ferramenta, foi a escolhida para processar os dados geográficos. A comunicação entre o serviço Web e o software é feita por meio do utilitário DinamicaConsole, que permite a execução de modelos pela linha de comando. Por sua vez, os modelos foram previamente criados e recebem os parâmetros provenientes da requisição, como por exemplo, identificador do mapa, camada a ser processada e tipo de simplificador.

A partir do simplificador é selecionado o modelo e é iniciado o processamento. O resultado de cada mapa é armazenado separadamente, permitindo assim que, caso o servidor possua os recursos, seja possível executar mais de um processo paralelamente. Os modelos, por sua vez, implementam diferentes simplificadores, porém de maneira geral todos possuem aspectos em comum, como a obtenção da camada do mapa, a aplicação do algoritmo de alteração do valor da célula e a geração e armazenamento dos tiles para cada um dos níveis de detalhamento disponibilizados. No entanto, cada algoritmo de simplificação possui diferentes características na função de agregação de *pixels*. Por exemplo, no caso do simplificador de quantidade de células, onde a cada célula é atribuído o valor 1 e é aplicado o processo de reamostragem de soma ponderada, fazendo com que o valor represente a quantidade de valores do mapa original na região. Já o processo de geração do dado em diferentes níveis é feito com base nos níveis de zoom. Portanto, o mapa é reprojetoado para cada nível de zoom utilizando um dos métodos de reamostragem: soma, célula mais próxima, média, entre outros. O mapa reamostrado é cortado em tiles, de

tamanho 256x256, que devem ser armazenados em cache. Tanto os tiles em formato PNG quanto o metadado informando o tipo de célula usado na simplificação são armazenados em disco. Esse armazenamento utiliza uma pasta com o nome do mapa. Dentro dessa pasta existe outra que indica o índice da camada utilizada, e subsequentemente uma pasta por operação, dentro da qual existe uma pasta por nível de zoom. Essa estrutura é similar à utilizada pelo WMS do Geoserver, que integra o serviço GeoWebCache, um middleware para caching de tiles WMS. A arquitetura proposta sugere o seu próprio sistema de cache, pois é preciso armazenar vários caches para o mesmo mapa, um para cada simplificador, segundo a estrutura de pastas descrita. Dessa maneira, pode-se limitar a quantidade de dados em uma mesma pasta, definindo um sistema simples para determinar o caminho do arquivo de cache a partir de uma requisição. Assim, determina-se com facilidade se o cache já foi criado e está pronto para uso, ou se é necessário criá-lo.

4.2 Requisição dos dados

A primeira etapa começa com uma requisição do cliente que inclui os parâmetros necessários para identificar a operação e orientar o servidor sobre como proceder. Com base nessa requisição, o servidor determina quais dados são solicitados e verifica se eles já foram processados e podem ser enviados imediatamente ao cliente. No caso do WMS, WPS e da nossa proposta, o dado é buscado no cache. Caso não seja encontrado, é necessário inicializar o processamento. Já nos serviços WFS e WCS, que não utilizam cache, basta identificar os dados originais e codificá-los para envio. Assim, ao final dessa etapa, determina-se se há ou não a necessidade de processamento antes da transferência do dado.

4.3 Processamento concentrado no servidor

Nesta seção, são tratados os serviços WMS e WPS que processam os dados integralmente no servidor para envio apenas do resultado ao cliente. No contexto de um serviço em tempo real, busca-se a minimização do tempo de resposta. Por isso, será considerada a máxima e permanente utilização do cache, de maneira que, ao finalizar uma execução, o resultado seja armazenado permanentemente. Dessa maneira, cada resultado

é processado uma única vez, e as consultas subsequentes apenas o reutilizam. No padrão WMS o dado é processado aplicando-se a legenda e enviando a imagem ou representação gráfica do dado ao cliente. Essa imagem pode ser dividida em tiles de maneira a limitar a quantidade de dados apenas para a região de interesse. Porém, esse serviço WMS é definido apenas para a visualização e não permite a execução da álgebra de mapas. Por isso, o tempo do WMS não será avaliado, uma vez que não permite as operações propostas. No entanto, no caso do WPS existe sim a possibilidade de execução dos testes e por isso será avaliado mais detalhadamente no contexto proposto nesse trabalho.

O padrão WPS foi usado para processar cada teste separadamente a fim de evitar a interferência no desempenho entre eles. No entanto, esse serviço suporta múltiplas requisições paralelas, que não será avaliado por depender de crescente aumento na infraestrutura para que o desempenho não seja degradado. Portanto, executou-se os testes e os tempos em cada cenário foram descritos na [Figura 4.4]. Um aspecto notável, não necessariamente do padrão mas pelo menos na implementação utilizada, é a limitação de 65 mil linhas ou colunas nas dimensões do raster. Por exemplo, o mapa *Aboveground biomass* que não possui nenhum resultado pois ultrapassa a maior dimensão suportada pelo serviço. O tempo de execução varia muito por operação pois os tempos de agregação dos valores demorou aproximadamente trinta vezes mais que o tempo de multiplicação da matriz por uma constante. Outro ponto importante é que o tempo analisado corresponde apenas ao processamento do dado e não considera o pós-processamento para envio ao usuário. Esse envio pode depender de um processamento equivalente ao do WMS para gerar imagens que representem o resultado. No entanto, apenas o tempo de processamento já inviabiliza a sua utilização para um serviço em tempo real para os mapas de teste. A última alternativa é a de reutilização do cache gerado para responder as requisições em tempo real. No entanto, esse serviço depende de todos os parâmetros definidos na expressão de cálculo para processamento inteiramente no servidor. E portanto, ao trocar qualquer valor na expressão é necessário gerar um novo resultado, praticamente uma versão a cada requisição. Por isso é inviável pré armazenar todas as combinações de cache e não garante o resultado em tempo real.

4.3.1 Processamento concentrado no cliente

Nesta seção são tratados os padrões WCS e WFS que basicamente permitem o envio dos dados ao cliente, para que esse possa processar os dados. Nesse contexto o servidor necessita apenas codificar o dado para o formato requerido pelo cliente. Os dados podem ser requisitados em vários formatos possíveis dados rasters por meio do

Figura 4.4: Tempo de execução via WPS das operações de soma e multiplicação nos mapas de teste.

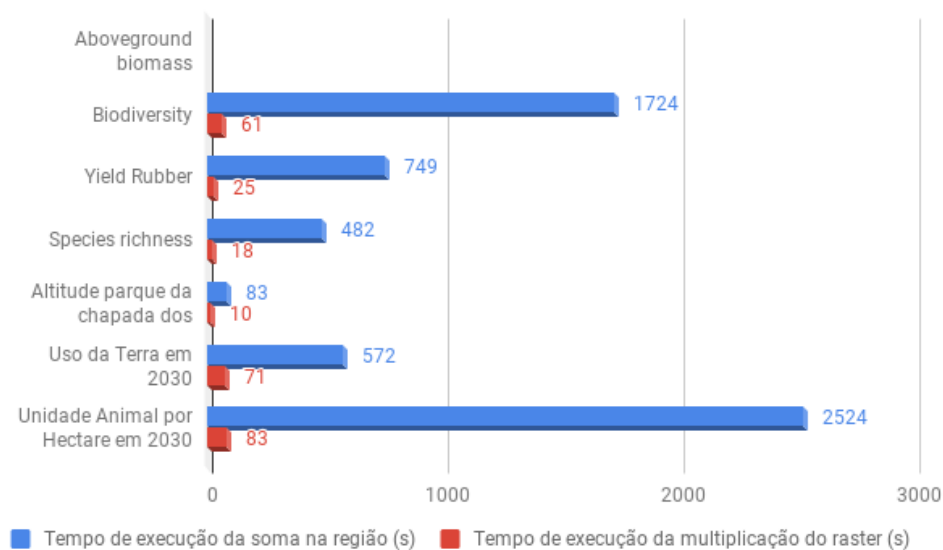
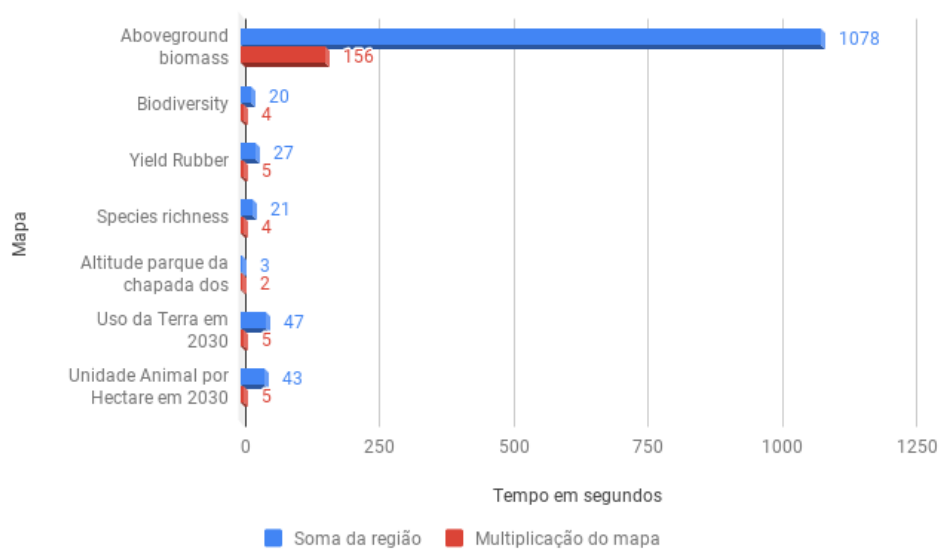


Figura 4.5: Média de tempo em segundos para executar cada operação no mapa original.



WCS e vetoriais por meio do WFS. Para avaliar a performance desses serviços optou-se pela obtenção do dado por meio do WCS no formato *TIF* e o software de processamento *Dinamica EGO*. Não será comparado portanto, com as implementações em ferramentas SIG, e por isso deve ser avaliado apenas o tempo de processamento do dado no software de processamento. Portanto, ao para avaliar o tempo total é preciso somar uma das seguintes etapas: transferência, processamento e eventual geração e aplicação de uma legenda.

4.3.2 Processamento dividido entre servidor e cliente

Esta seção analisa o tempo gasto nas etapas de processamento da arquitetura híbrida, ocorrendo parcialmente no servidor e no cliente. O tempo de gerar os mapas simplificados no servidor tende a ser elevado, porém, o resultado é armazenado em cache. As requisições posteriores, dos mesmos simplificadores serão executadas instantaneamente sem a necessidade de reproprocessamento. Em outras palavras, se a expressão utilizar os mesmos simplificadores, mesmo que com valores diferentes, o cache pode ser instantaneamente enviado sem reproprocessamento. O servidor processa os *pixels* do dado original, reescala e gera os tiles. Esses dados são transferidos ao cliente, que aplica as operações algébricas com os valores definidos pelo usuário. Os tempos descritos aqui incluem o pré-processamento e geração do cache, e não o tempo de resposta a uma requisição, pois é razoável considerar que o cache já foi criado. Dessa maneira, as requisições poderão ser avaliadas apenas em relação ao atraso devido à transferência do dado acrescido do processamento no cliente.

Para analisar a performance da simplificação dos mapas verificou-se quais aspectos influenciam no tempo de execução. Para isso, foi considerado o tempo de geração de alguns mapas de até a resolução de seiscentos metros por *pixel*, que representa o nível de zoom oito. Considerando que a projeção utilizada seja a de código (EPSG:3857)⁵ que representa a terra com área de aproximadamente $1.6 * 10^{15}$ m². De acordo com a [Equação 3.2], no oitavo nível de zoom, a area total da terra é representada por mais de quatro bilhões de *pixels* e possui resolução de 600 metros quadrados. Todos os mapas nesse nível de zoom estarão na mesma escala, por isso o tempo de processamento deve ser proporcional a area. Outro ponto importante é que no servidor a quantidade de células processadas aumenta em quatro vezes a cada nível de zoom. No entanto, o processamento no cliente geralmente mantém-se constante uma vez que a quantidade de *pixels* na tela deve ser aproximadamente a quantidade de *pixels* visíveis.

Ao analisar o gráfico 4.6 é possível perceber que o tempo de publicação varia não linearmente ao tamanho do mapa e que, de maneira geral, demanda elevado tempo de processamento. No entanto, a variação entre os tempos de publicação não pode ser explicada apenas com base nesses dados, uma vez que não existe uma correlação direta entre o tamanho do mapa e o tempo de publicação. Portanto, foi necessária a realização de outras análises. Na primeira, buscou-se correlacionar a ocorrência de células nulas no mapa, ou seja, células sem um valor. Para isso, correlacionou-se o tempo médio de cada simplificador com a quantidade de células nulas a fim de identificar essa possível correlação.

⁵Spherical Mercator, EPSG. Disponível em: <https://epsg.io/3857>. Acesso em 11 de abril de 2019.

Figura 4.6: Tempo em segundos de processamento do serviço para aplicar cada simplificador e gerando o cache até o nível oito de zoom.

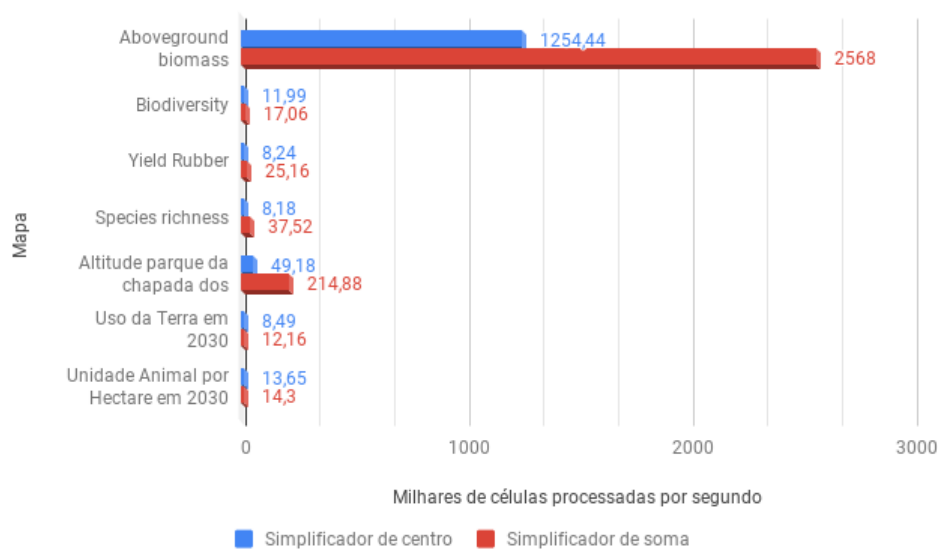
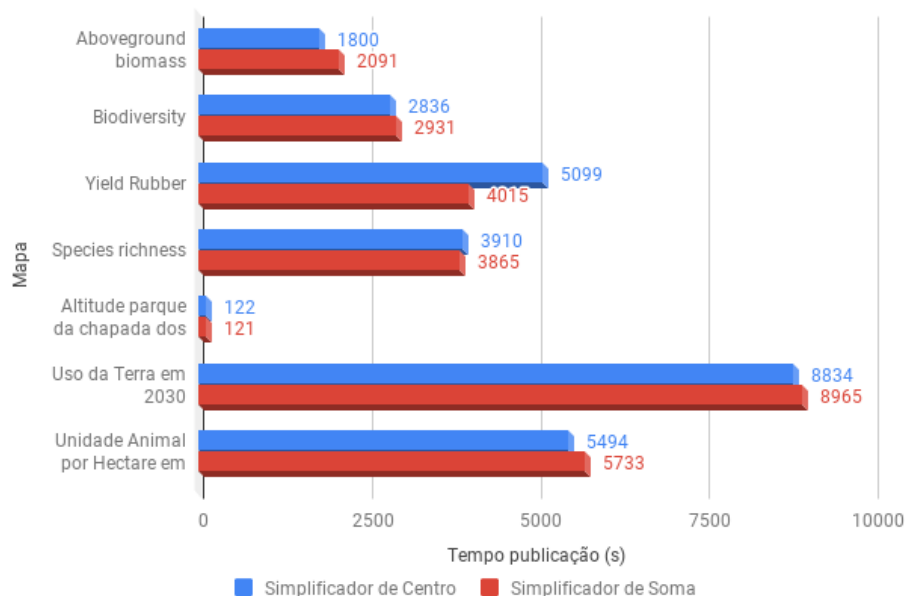


Figura 4.7: Tempo médio em segundos por milhão de células processadas por mapa.

Ao analisar a [Figura 4.7] e a [Figura 4.8], observa-se que o algoritmo de soma não é influenciado pela quantidade de células nulas, enquanto o algoritmo de centro melhora sua performance nessa condição. Contudo, essa evidência isolada não explica completamente a diferença nos tempos de execução entre os simplificadores de cada mapa. Uma análise adicional considerou a resolução por célula, definida pela área ocupada por cada uma. Como os mapas possuem diferentes tamanhos de células, essa característica foi apresentada na [Figura 4.9]. Verificou-se que mapas que ocupam áreas maiores demandam mais tempo de execução em comparação aos de menor área. Apesar de contraintuitiva, essa

Figura 4.8: Proporção das células nulas e não nulas dos mapas de teste.

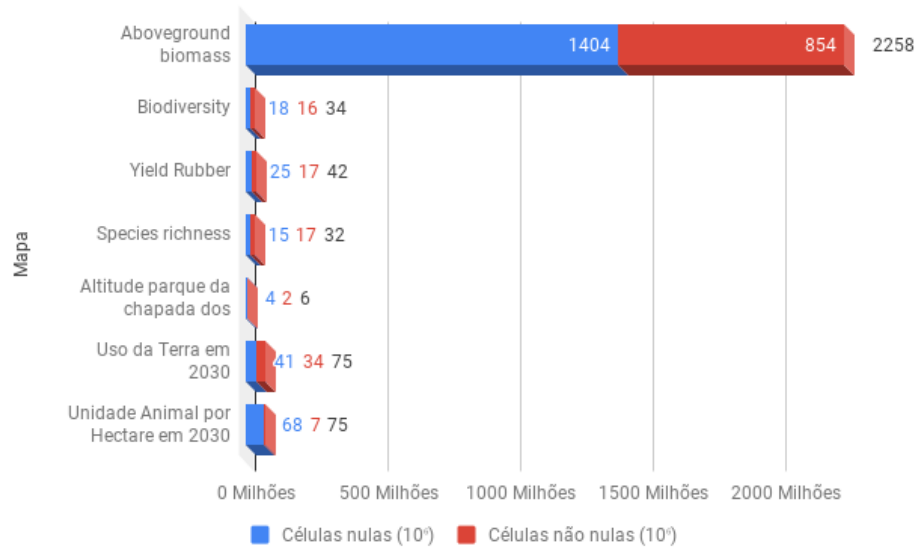
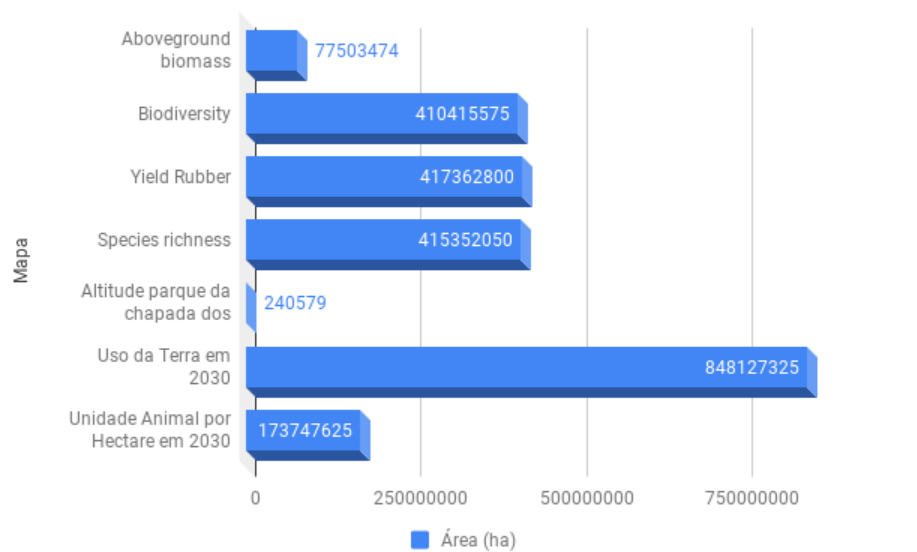
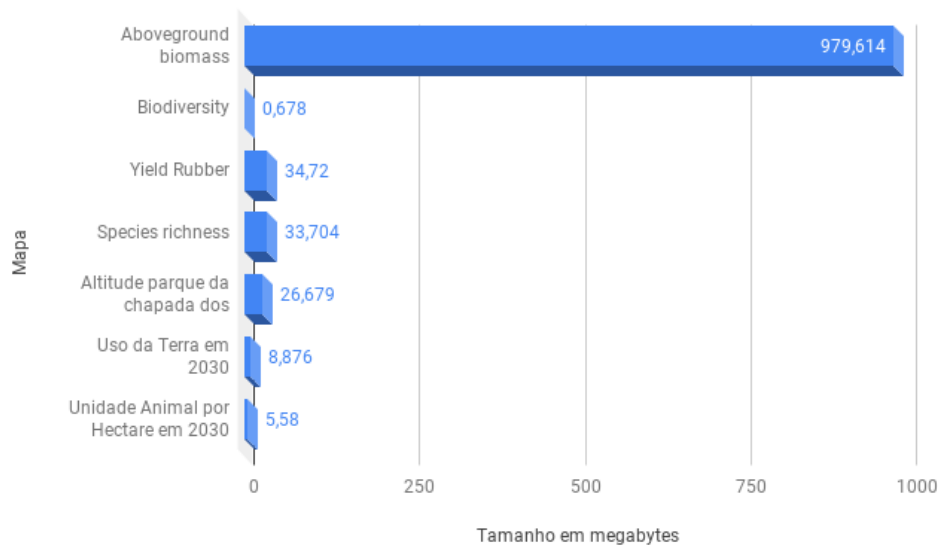


Figura 4.9: Área da região coberta por cada mapa em hectares.



constatação se justifica pela forma de geração dos tiles. O dado original é sempre reescalado para os diversos níveis de zoom, resultando em uma maior quantidade de células em mapas com maior área. Assim, a variação no nível de zoom e na área de cobertura altera a quantidade de tiles gerados, o que aumenta o tempo de processamento necessário.

Figura 4.10: Volume original dos dados em megabytes com compressão LZW.



4.4 Transferência dos dados

Esta seção trata da transferência dos dados entre o servidor e o cliente e faz parte de todas as abordagens. Será avaliado o tempo que o servidor demora para gerar a resposta acrescido do tempo de transferência pela rede.

Analisando o gráfico 4.10, e sabendo que a velocidade de download média da banda larga brasileira é de 24.77 Megabits⁶ é possível estimar o tempo médio de download de cada mapa original.

Na [Figura 4.11] é possível observar o registro de tempo em segundos para envio de cada mapa original. Essa variação é diretamente proporcional ao tamanho do arquivo. Além disso, também foi observada a taxa de compressão de cada mapa avaliado, como mostrado na [Figura 4.12], que exemplifica a grande variação da compressão do dado. Nesse gráfico é possível verificar que a maioria dos dados apresentou boa taxa de compressão. Apesar disso, a variação entre as taxas é bastante alta, enquanto um mapa foi comprimido em 98% outro apresentou compressão negativa de -11,16%. Essa compressão utilizou um método de dicionário e a taxa de compressão é imprevisível pois depende do conteúdo [15]. Por isso, é difícil estimar os tempos de envio com base na quantidade de dados requerida. Essa incerteza e enorme variação, pode causar instabilidade e é indesejável para um serviço em tempo real. Além disso, outro problema grave é que a simples alteração dos valores dos *pixels* podem influenciar a taxa de compressão e causar aumento

⁶Barómetro das ligações à Internet fixa no Brasil. 2018. Disponível em: https://media.nperf.com/files/publications/BR/2018-07-18_Barometro-internet-fixa-nPerf-2018-S1.pdf. Acesso em: 23 janeiro 2025.

Figura 4.11: Tempo médio de download por mapa em segundos.

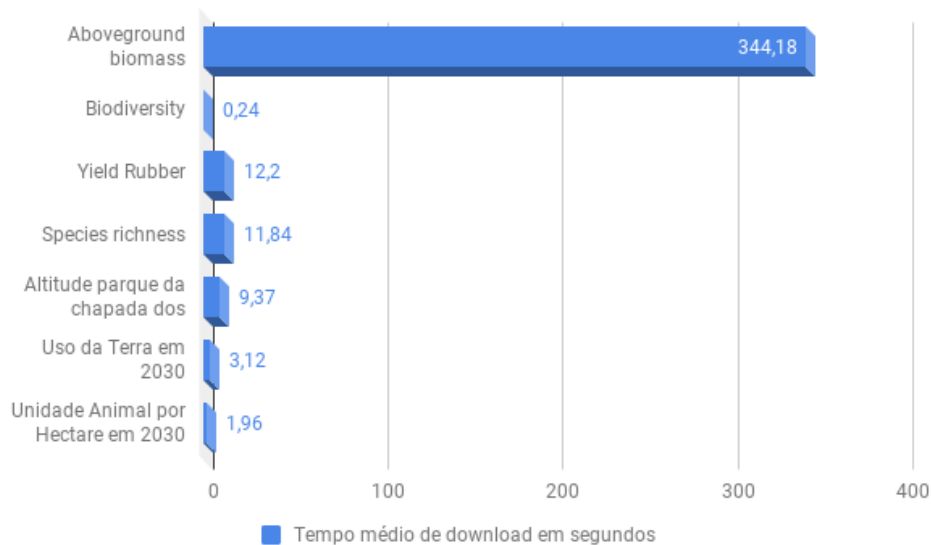
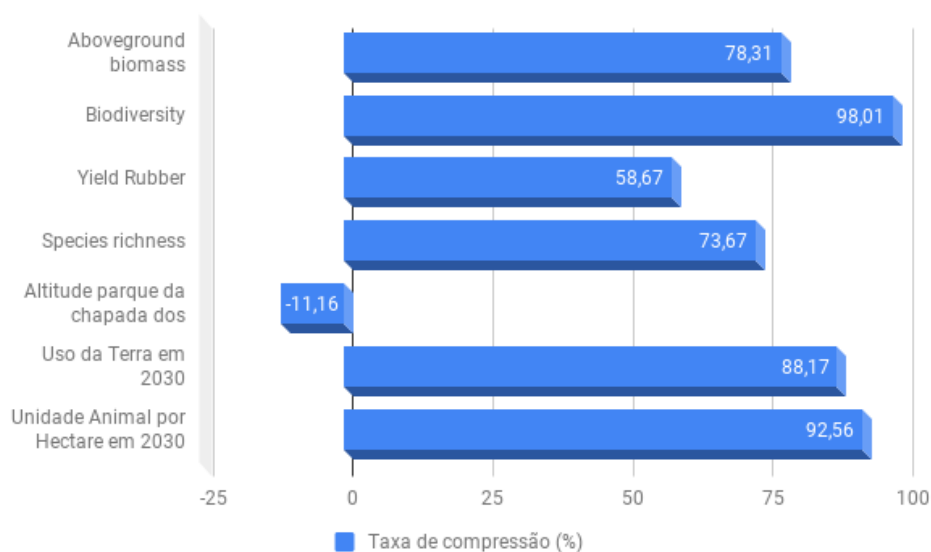


Figura 4.12: Relação entre o tamanho do arquivo espacial comprimido pelo algoritmo LZW e o dado original.



substancial no tamanho do mapa. Por isso, se mostrou imprescindível para um serviço em tempo real que os dados não fossem enviados em sua totalidade, uma vez que muito comumente os dados geográficos, principalmente os matriciais, são constituídos de uma enorme quantidade de dados.

4.5 Visualização do resultado

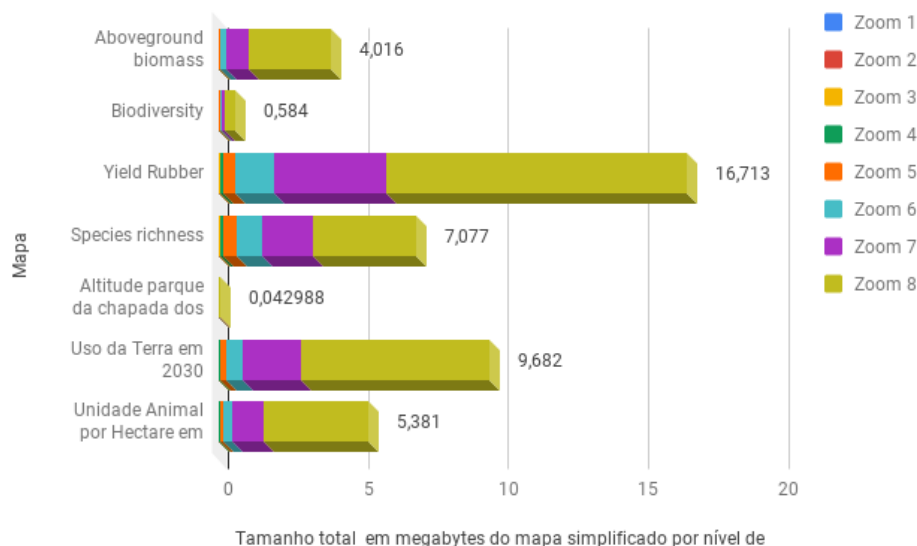
Nesta seção é analisado o tempo relacionado a visualização do dado no cliente em cada abordagem. No WMS o dado já foi processado e aplicada a legenda, porém como não foi possível aplicar as operações não iremos avaliar essa abordagem. No caso dos serviços WPS, WCS, WFS pode ser necessário reescalar o dado e aplicar legenda ao dado original. Já na arquitetura aqui proposta o dado requisitado já foi reescalado mas será necessário a aplicação da legenda. É importante observar que, para cada operação, o tipo de dado final pode ser diferente e, por isso, a visualização pode variar. Por exemplo, o resultado da soma da região pode ser simplesmente um número, já a multiplicação do mapa por um valor deve necessariamente resultar em um novo mapa. Por essas diferenças as análises serão feitas separadamente abaixo.

4.5.1 Análise da visualização da multiplicação

Este tópico aborda os processos após o processamento e disponibilização do resultado da operação de multiplicação do mapa por uma constante. Para a visualização do mapa, é necessário escolher a parte visível e a escala. Existem vários algoritmos de reamostragem, incluindo o algoritmo "Nearest-neighbor" ou "vizinho mais próximo". Identificamos as similaridades e peculiaridades de cada um dos três cenários propostos: processamento no cliente, no servidor e o híbrido. Nos serviços WPS, WCS e WFS, os dados precisam ser reamostrados para a escala visível. Na abordagem híbrida, o mapa é enviado após a simplificação, mas antes da aplicação da legenda, o que resulta no envio de dados em menor volume, apenas com os valores necessários naquela escala. Para a visualização, é preciso processar apenas o dado simplificado, o que agiliza a aplicação da expressão e da legenda. Garantindo uma maior estabilidade do serviço, devido a menor interferência do tamanho do dado, e redução drástica do tempo total de processamento da requisição.

Dessa maneira, a resolução de todos os mapas é a mesma, definida pelo nível de zoom, e portanto pode-se facilmente ser estimada. Para o cálculo é considerada a área do mapa e o nível de detalhamento, que possibilita estimar a carga de uso dos recursos computacionais como rede e processamento. Para isso, analisou-se a quantidade de dados enviados em cada nível anotando o tempo de publicação de cada mapa. De acordo com a [Figura 4.13] mesmo que seja necessário carregar todos os tiles dos níveis de zoom até o nível oito a quantidade de dados é menor que o carregamento do mapa original.

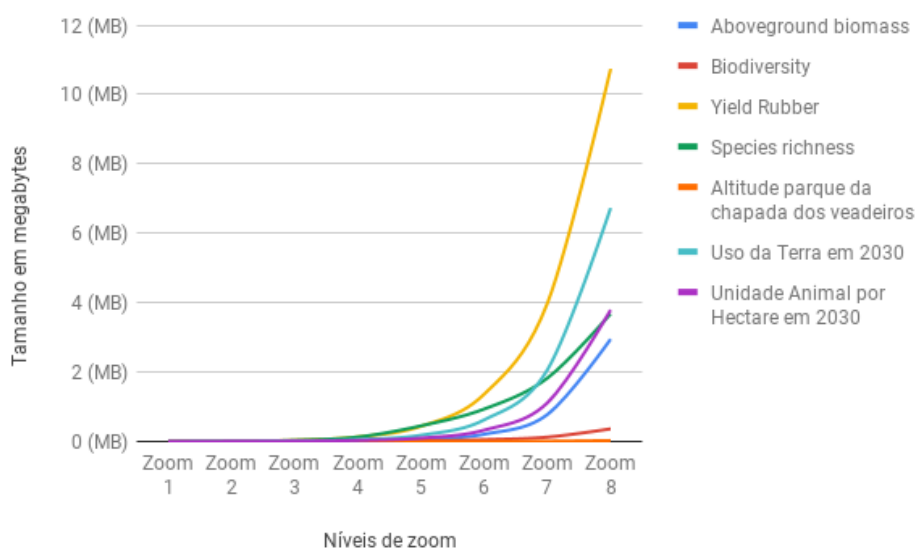
Figura 4.13: Tamanho total do mapa simplificado por nível de zoom somado até o nível oito. Para cada nível de zoom foi calculado o total de megabytes necessários para armazenamento de todos os tiles.



4.5.2 Análise da visualização da soma

Esta seção descreve a soma por região e resulta em um valor final para cada região de processamento no servidor e no cliente. Porém, na abordagem híbrida o dado foi simplificado no servidor e deverá ser enviado ao cliente para que termine o cálculo. Esse processamento permite que o usuário faça a soma de qualquer região no dado simplificado. Por isso o tempo de envio é acrescido do tempo de processamento do mapa de soma no cliente. Para estimar a quantidade de dados a ser processado foi considerado o nível de zoom da operação e o tamanho da área. Primeiramente analisou-se o volume total dos tiles gerados em diversos níveis de zoom, e esses valores foram dispostos no gráfico da [Figura 4.14]. Nesse gráfico é possível perceber que a quantidade de dados aumenta exponencialmente com o aumento do nível de zoom. Porém, com o mapa simplificado a análise pode ser feita no nível de zoom referente ao detalhamento usado na especificação da região analisada. Portanto, em nosso exemplo a maior resolução dessa região deverá respeitar o limite que definimos nessa avaliação de 600m. Caso a resolução seja menor, não será necessário enviar todo o mapa para o cliente, mas somente a parte requisitada na resolução equivalente. Sabendo disso, em nossa comparação avaliaremos a relação entre o tamanho do mapa em vários níveis de detalhamento do mapa. Comparando o total do dado simplificado com o dado original 4.10 é possível observar que inicialmente o mapa é menor mas o crescimento do dado pode chegar a ultrapassar o tamanho total do dado

Figura 4.14: Tamanho total em megabytes por nível de zoom até o nível de zoom 8 para a simplificação por soma.



nos casos em que a resolução exibida ultrapasse a resolução do dado original. No entanto é importante notar que quanto maior o mapa, maior o nível de zoom para equiparação entre a versão simplificada e a original, ou seja para mapas pequenos a equiparação pode acontecer em níveis de zoom baixo, porém para mapas grandes essa equiparação ocorre somente nos níveis elevados. Portanto a maior vantagem desse método é na sua utilização para mapas com grandes volumes de dados, e não em mapas com pequenos volumes de dados. No entanto, mesmo em mapas pequenos, a análise pode ser realizada em partes e portanto a quantidade de dados enviados também será uma fração do mapa original.

4.5.3 Análise dos tempos de resposta em cada abordagem

Nesta seção como já foram descritos todos os passos de cada abordagem, foram correlacionados os dados obtidos em cada uma delas e analisar o desempenho da arquitetura híbrida em relação às alternativas. É importante destacar que, em todas as abordagens, foram identificados gargalos, sendo os principais o tempo de transferência, tempo de resposta e o alto consumo de recursos computacionais.

O tempo de download, é um gargalo forte para os serviços em tempo real pela internet, especialmente dados que possuam magnitude de gigabytes. É essencial, portanto, estimar e limitar o tamanho do conteúdo de download, porém o algoritmo Lempel-Ziv Welch (LZW) [37], sem perda apresenta alta variação na taxa de compressão. Por exem-

plô, um mapa com menos informações pode demorar mais tempo para ser enviado que um mapa com mais detalhes por causa dessa variação. Essa imprevisibilidade em um sistema do tipo 'cliente pesado e servidor leve' pode gerar instabilidade no serviço para responder a algumas requisições e, por isso, é inapropriada para a sua utilização em tempo real. No caso da abordagem híbrida, os tiles, apesar de sofrerem também com a imprevisibilidade intrínseca da compressão, ficam limitados ao nível de zoom e área de interesse da análise, que serve de limite superior. Por isso, no pior caso cada tile com dimensões de 256x256 e cada *pixel* armazena um inteiro de quatro bytes que resulta em 260 kbytes, que pode ser transferido em poucos segundos pela internet.

Outro gargalo, comum a todas as abordagens, é o elevado custo computacional de cada operação, que pode tornar inviável o processamento em paralelo de muitas requisições sem o investimento na ampliação da infra-estrutura. Ambas as abordagens, puramente no servidor quanto a puramente no cliente, são executadas no mapa original e precisam processar o mapa todo a cada nova operação. Já na abordagem híbrida, o mapa original só vai ser processado uma vez, na primeira requisição. O tempo de execução é bastante elevado, como pode-se observar na [Figura 4.7], que variam de alguns minutos até algumas horas. No entanto, nessa abordagem o mapa simplificado vai para o cache, e não precisa ser reprocessado. Essa versão do mapa tem apenas a informação útil àquela operação e pode ser processada rapidamente no cliente. Assim, o processamento do servidor é reutilizado, e o cliente processa apenas o dado simplificado, melhorando o tempo de resposta, requisitos essenciais para um sistema em tempo real.

A abordagem híbrida consiste na utilização de poucos recursos computacionais do servidor para responder às requisições, pois, em geral, apenas envia os dados simplificados, sem carga adicional de processamento. Isso distribui a carga sem sobrecarregar o servidor e permite que o processamento desejado pelo cliente seja executado localmente em cada máquina. Todos os testes foram realizados no hardware especificado e que processou um mapa por vez, para simplificar a análise da performance. Outra vantagem da abordagem híbrida é a possibilidade de reutilização do mesmo dado simplificado em várias operações, uma vez que cada versão enviada ao cliente pode ser usada em diferentes expressões algébricas. Por exemplo, um mapa simplificado que será multiplicado por uma constante, o cliente pode usar o mesmo mapa independente do valor da constante. Por outro lado, na abordagem de servidor pesado, todo valor alterado implica em reexecutar tudo novamente. O cliente pesado também depende de processar o dado inteiro, e pode apresentar elevado tempo de resposta. Portanto, a abordagem híbrida amplia a utilização do cache do servidor e diminui drasticamente a carga e o tempo de resposta, mesmo em casos com acessos simultâneos.

Capítulo 5

Conclusão

A disseminação de informações geográficas envolve o envio e a transferência de grandes volumes de dados, o que pode resultar em processos demorados e elevados custos computacionais. Um exemplo dessa situação é o projeto AMAZONES¹, que analisa o valor potencial de vários aspectos relacionados à floresta amazônica. Cada produto ou variável considerado nesse projeto é representado por matrizes compostas por mais de cinco milhões de células, tais como a produção de borracha, o risco de incêndio e o índice de importância da biodiversidade, entre outros. Apesar do grande volume de dados envolvido, um dos principais objetivos desse projeto é calcular o valor da Amazônia em diferentes cenários, o que requer calcular a influência de cada produto ou variável e obter valores instantâneos de algumas variáveis. Cada parâmetro deve ser configurável pelo usuário, como a taxa de câmbio, o preço de mercado da carne bovina e da energia elétrica. Além disso, é necessário que a análise seja oferecida como um serviço em tempo real, capaz de atender múltiplos usuários simultaneamente.

Para cumprir com esse tipo de requisitos, foram avaliados os serviços de disponibilização de mapas e dados geográficos na internet. Foram avaliadas alternativas como o padrão definido pelo Open Geospatial Consortium (OGC), e variações. No entanto, nenhuma das alternativas existentes permite suprir as necessidades deste projeto e de outros semelhantes, pois cada abordagem oferecida possui características que a inviabilizam para a solução do problema, seja por dificuldades na escalabilidade, seja por consumo de recursos computacionais. O serviço Web de processamento dos dados no servidor (WPS), por exemplo, requer grande poder computacional do servidor e apresenta tempo de resposta elevado, mesmo sem considerar múltiplos acessos concorrentes ao sistema. Uma abordagem como a do WPS poderia funcionar bem caso os dados fossem de menor volume, como geralmente é o caso de dados vetoriais, porém a utilização de processamento em tempo real de enormes quantidades de dados implicaria em limites drásticos à qualidade do serviço. A abordagem oposta, que consiste no envio dos dados do servidor ao cliente para que este execute toda a análise em um computador local, reduz o processamento no servidor, mas acarreta uma alta demanda de rede e um custo computacional

¹AMAZONES - AMAZON ECO SERVICES, CSR. Disponível em: <http://maps.csr.ufmg.br/calculator/?queryid=106>. Accessed on October 16, 2018.

elevado no cliente, além de poder resultar em tempos de resposta prolongados. Ademais, mesmo que o cliente possua um poder computacional significativo e atenda aos requisitos necessários, certos conjuntos de dados envolvem quantidades tão grandes de informações que exigiriam um tempo considerável para manipulação, tornando inviável a realização de análises em tempo real. Além disso, o cliente também teria que preparar um ambiente adequado para esse processamento, o que nem todos os interessados nos dados estariam habilitados ou motivados a fazer, dificultando assim o acesso à informação e sua utilização. Conseqüentemente, essa abordagem centrada no cliente também é insuficiente.

Para solucionar o problema em questão, foi adotada uma abordagem que combina parcialmente os padrões existentes por meio de uma arquitetura mista, utilizando tanto o poder de processamento do servidor quanto do cliente. A proposta concentra-se nos principais desafios que impedem o uso em tempo real: o tempo de transferência de grandes volumes de dados pela rede e o processamento de grandes volumes de dados. A fim de reduzir o tempo de transferência, buscou-se alternativas que permitem a redução do volume de dados a transmitir, sendo o dado simplificado ou mesmo comprimido. Portanto, foram criados métodos que minimizam o volume de dados, removendo tudo o que não é absolutamente necessário. Para isso, foi analisado o contexto mais específico de utilização do dado, para encontrar o mínimo de dados necessário em cada cenário.

A utilização dos dados é influenciada por duas importantes características: a escala de análise e a operação aplicada. A escala do dado determina o nível de detalhamento com o qual os dados serão visualizados. Assim, quando uma análise é realizada em uma escala diferente da escala original dos dados, é necessário realizar reamostragem dos dados ou do resultado. Outro aspecto relevante diz respeito à operação a ser aplicada aos dados. Ao realizar a agregação de *pixels* utilizando diferentes operações, os resultados obtidos serão distintos. Portanto, é possível criar diversas versões do mapa, cada uma específica para uma combinação de operação e nível de zoom, de forma que uma versão seja adequada para uma operação e desconsiderada para qualquer outra. Foi desenvolvida uma versão específica dos dados que é adequada apenas para um determinado nível de zoom e tipo de operação e que não vai ser usada em outras condições.

Foi superado o desafio do elevado tempo de processamento, comumente encontrado em diversos serviços, por meio da aplicação de técnicas de pré-processamento e armazenamento em cache, utilizando o conceito de simplificadores definido na presente dissertação de mestrado. Para isso, foi adotada uma abordagem semelhante ao uso de pirâmides em imagens [1], criando-se uma versão para cada nível de zoom. Essas versões são geradas com base nos simplificadores, que são métodos de reamostragem otimizados para uma combinação específica de nível de zoom e operação.

Os simplificadores proporcionam a capacidade de reamostrar e agregar os dados de forma a preservar a exatidão das operações, quando aplicadas na mesma área, independente da escala. Cada simplificador mantém uma propriedade específica, estabelecida

neste estudo, que assegura a consistência da operação aplicada. Um exemplo é o simplificador de soma que garante que o valor total da soma seja mantido após a reamostragem do mapa. Dessa forma, o cálculo da soma em uma determinada região pode ser realizado no mapa simplificado, que possui um volume menor de dados, sem comprometer a acurácia do resultado.

Esta solução permite pré-calcular todos os resultados necessários nos servidores e armazenar em cache as respostas, tornando o tempo de resposta do servidor instantâneo. E ao somar regiões no mapa simplificado, o resultado será sempre igual ao da aplicação da mesma operação sobre o mapa original, guardada a limitação da escala na especificação da região. Assim, o cálculo de soma da região pode acontecer também no mapa simplificado, com volume menor de dados, garantindo que o resultado continue exatamente o mesmo, e isso também é válido para outras operações.

A arquitetura proposta neste estudo demonstrou ser adequada para alcançar os objetivos estabelecidos. A adoção de mapas simplificados viabiliza a transferência rápida de arquivos, independentemente da resolução original do mapa. O servidor realiza o processamento de cada simplificador do mapa apenas uma vez e armazena o resultado em cache, possibilitando sua reutilização em múltiplas ocasiões. Essa técnica substitui o processamento em tempo real por meio do pré-processamento armazenado em cache, que melhora drasticamente o tempo de resposta. Para criar esse cache, é necessário aplicar cada operação de simplificação em todos os níveis de zoom em que o mapa estará disponível. Portanto, o tempo necessário para gerar o mapa simplificado pode ser significativamente maior do que o tempo exigido para aplicar uma operação similar no mapa original. No entanto, esse processo é executado apenas uma vez no servidor, não inviabilizando assim essa estratégia. Além disso, o processamento que o cliente vai realizar em tempo real é reduzido, pois trabalha apenas com os dados simplificados. Dessa forma, tanto o processamento no servidor quanto o tempo de resposta e o tempo de processamento no cliente são relativamente baixos.

A arquitetura proposta foi utilizada como base para demonstrar a visualização de cenários em tempo real na plataforma AMAZONES. Os mapas foram categorizados e aplicaram-se operadores de vizinho mais próximo. Dessa forma, foi possível combinar os mapas com os pesos definidos pelo usuário, visualizar diferentes cenários em tempo real e verificar a estabilidade do serviço em conformidade com os requisitos estabelecidos. A utilização dessa plataforma não acarretou em aumento significativo no consumo de espaço em disco nem no consumo de processamento da máquina. O consumo de espaço é diretamente proporcional à quantidade de versões simplificadas, e que não resulta em problemas graves de armazenamento. Outro aspecto relevante é que o desempenho do serviço não é afetado pela resolução original do mapa disponibilizado, mesmo em casos de acessos simultâneos. O limitador da qualidade do serviço é a quantidade de mapas em exibição, que pode afetar o tempo de resposta na visualização dos dados, uma vez que

cada mapa é enviado separadamente e a largura de banda da rede é um recurso escasso. Portanto, quanto mais mapas forem carregados, proporcionalmente maior será o tempo de transferência dos dados.

Esta abordagem é fundamentada na capacidade de reescrever as operações em várias escalas, preservando o resultado. No entanto, existem operações que requerem o uso de todas as células dos dados originais e, portanto, não podem ser adequadamente implementadas nesta arquitetura. Por exemplo, em uma análise de fluxo, é necessário comparar os valores das células adjacentes. Nesse caso, é impossível agrupar as células de forma a manter o resultado consistente, pois a informação relativa às células adjacentes seria perdida. Essas operações não podem ser realizadas nesta abordagem e representam uma das suas principais limitações. Outro ponto observado é que a adição de cada novo simplificador aumenta linearmente o consumo de espaço em disco e o tempo de pré-processamento. Além disso, a quantidade de dados a ser enviada ao cliente é proporcional ao número de simplificadores usados. Trabalhos futuros podem explorar alternativas que minimizem ou evitem o envio de múltiplos conjuntos de dados, permitindo a união de simplificadores combinados em um único simplificador. Criando um conjunto de operações que ocupe um volume menor do que a soma das operações individuais.

Este trabalho apresenta uma contribuição significativa por meio de uma arquitetura baseada na utilização de cache no servidor, combinada com o uso de simplificadores, que viabiliza a execução de operações em diferentes níveis de zoom em tempo quase real. Essa abordagem permitiu a realização de simulações em tempo real com um baixo consumo de recursos computacionais, ao mesmo tempo em que manteve a qualidade do serviço, mesmo diante de acessos simultâneos por múltiplos usuários. Como resultado, os objetivos iniciais estabelecidos foram plenamente alcançados, e um serviço funcional foi implementado como prova de conceito no site de divulgação do projeto AMAZONES.

Referências

- [1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.
- [2] Claudia Azevedo-Ramos and Paulo Moutinho. No man’s land in the brazilian amazon: Could undesignated public forests slow amazon deforestation? *Land Use Policy*, 73:125 – 127, 2018.
- [3] Thomas Boutell. Png (portable network graphics) specification version 1.0, March 1997. Disponível em: <https://www.rfc-editor.org/rfc/rfc2083>. Acessado em: 2018-05-30.
- [4] David J Coleman and JD McLaughlin. Defining global geospatial data infrastructure (ggdi): components, stakeholders and interfaces. *GEOMATICA-OTTAWA-*, 52:129–143, 1998.
- [5] Franklin C Crow. Summed-area tables for texture mapping. In *ACM SIGGRAPH computer graphics*, volume 18, pages 207–212. ACM, 1984.
- [6] JEFF De la Beaujardiere. Opengis web map server implementation specification (ogc 06–042). *Open Geospatial Consortium Inc. Accessed November, 10:2016*, 2006.
- [7] Gilberto Ribeiro de Queiroz, Karine Reis Ferreira, Lúbia Vinhas, Gilberto Camara, Raphael Willian da Costa, Ricardo Cartaxo Modesto de Souza, Victor Wegner Maus, and Alber Sanchez. Wtss: um serviço web para extração de séries temporais de imagens de sensoriamento remoto. *XVII SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO*, 2015.
- [8] Roy Fielding. Representational state transfer. *Architectural Styles and the Design of Network-based Software Architecture*, pages 76–85, 2000.
- [9] GDAL/OGR contributors. *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation, 2018.
- [10] Gregory Giuliani, Alain Dubois, and Pierre Lacroix. Testing ogc web feature and coverage service performance: Towards efficient delivery of geospatial data. *Journal of Spatial Information Science*, 2013(7):1–23, dec 2013.

- [11] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 2017.
- [12] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202:18–27, 2017.
- [13] Zhipeng Gui, Jun Cao, Xiaojing Liu, Xiaoqiang Cheng, and Huayi Wu. Global-scale resource survey and performance monitoring of public ogc web map services. *ISPRS International Journal of Geo-Information*, 5(6):88, 2016.
- [14] Apoorv Gupta, Aman Bansal, and Vidhi Khanduja. Modern lossless compression techniques: Review, comparison and analysis. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–8. IEEE, 2017.
- [15] Amit Jain and Kamaljit I Lakhtaria. Comparative study of dictionary based compression algorithms on text data. *International Journal of Computer Science and Network Security (IJCSNS)*, 16(2):88, 2016.
- [16] Walter Jetz, Jana M. McPherson, and Robert P. Guralnick. Integrating biodiversity distribution knowledge: toward a global map of life, 2012. Disponível em: <https://mol.org/about>. Acesso em: 24 janeiro 2025.
- [17] William Kahan. Pracniques: further remarks on reducing truncation errors. *Communications of the ACM*, 8(1):40, 1965.
- [18] Ameet Kini and Rob Emanuele. Geotrellis: Adding geospatial capabilities to spark. *Spark Summit*, 2014.
- [19] Jorge Luiz Dos Santos LEAL. A amazônia brasileira e o seu caráter transnacional: o aproveitamento do seu patrimônio estratégico para garantia do desenvolvimento. *Itajaí-SC, Abril de*, 2019.
- [20] Alexander J Loechel and Stephan Schmid. Comparison of different caching techniques for high-performance web map services. *IJSDIR*, 8:43–73, 2013.
- [21] M Lupp. Styled layer descriptor profile of the web map service implementation specification. ogc 05-078r4. *Open Geospatial Consortium Inc.*, 22:2011, 2007.
- [22] Joan Maso, Keith Pomakis, and Nuria Julia. Opengis web map tile service implementation standard. *Open Geospatial Consortium Inc*, pages 102–105, 2010.

- [23] Alana Kasahara Neves, Thales Sehn Korting, Leila Maria Garcia Fonseca, Gilberto Ribeiro de Queiroz, Lúbia Vinhas, Karine Reis Ferreira, and Maria Isabel Sobral Escada. Terraclass x mapbiomas: Comparative assessment of legend and mapping agreement analysis. In *Brazilian Symposium on Geoinformatics - GEOINFO*, pages 295–300, 2017.
- [24] SHM Nogueira, LL Parente, and LG Ferreira. Temporal visual inspection: Uma ferramenta destinada à inspeção visual de pontos em séries históricas de imagens de sensoriamento remoto. In *XXVII Congresso Brasileiro de Cartografia; Instituto Brasileiro de Geografia e Estatística: Rio de Janeiro, Brazil*, 2016.
- [25] Open Source Geospatial Foundation. Geoserver: Open source server for geospatial data, 2001. Disponível em: <https://geoserver.org>. Acesso em: 18 abril 2023.
- [26] Zhong-Ren Peng and Chuanrong Zhang. The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS). *Journal of Geographical Systems*, 6(2), Junho 2004.
- [27] George Percivall. Big processing of geospatial data, 2013. Disponível em: <http://www.opengeospatial.org/blog/1866>. Acesso em: 25 maio 2019.
- [28] K Sahin and MU Gumusay. Service oriented architecture (soa) based web services for geographic information systems. In *XXIst ISPRS Congress. Beijing*, pages 625–630. Citeseer, 2008.
- [29] John T Sample, Kevin Shaw, Shengru Tu, and Mahdi Abdelguerfi. *Geospatial services and applications for the internet*. Springer Science & Business Media, 2008.
- [30] Sam Linton Savage. *The Solution of Discrete Linear Optimization Problems by Neighborhood-search Techniques*. PhD thesis, Yale University, New Haven, CT, USA, 1973. AAI7329245.
- [31] Britaldo S Soares-Filho, Hermann O Rodrigues, William L S Costa, and P Schlesinger. Modeling environmental dynamics with dinamica ego. *Belo Horizonte: IGC/UFMG: Universidade Federal De Minas Gerais*, 2009.
- [32] Jon Strand, Britaldo Soares-Filho, Marcos Heil Costa, Ubirajara Oliveira, Sonia Carvalho Ribeiro, Gabrielle Ferreira Pires, Aline Oliveira, Raoni Rajao, Peter May, Richard van der Hoff, et al. Spatially explicit valuation of the brazilian amazon forest’s ecosystem services. *Nature Sustainability*, 1(11):657–664, 2018.
- [33] C Dana Tomlin. Map algebra: one perspective. *Landscape and Urban Planning*, 30(1-2):3–12, 1994.

-
- [34] Ranga Raju Vatsavai, Shashi Shekhar, Thomas E. Burk, and Stephen Lime. Umn-mapserv: A high-performance, interoperable, and open source web mapping and geo-spatial analysis system. In Martin Raubal, Harvey J. Miller, Andrew U. Frank, and Michael F. Goodchild, editors, *Geographic, Information Science - 4th International Conference, GIScience 2006, Proceedings*, volume 4197 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 400–417. Springer Verlag, 1 2006.
- [35] Lúbia Vinhas, Gilberto Ribeiro de Queiroz, Karine Reis Ferreira, and Gilberto Câmara. Web services for big earth observation data. In *Brazilian Symposium on Geoinformatics - GeoInfo*, pages 166–177, 11 2016.
- [36] P Vretanos. Web feature service (wfs) implementation specification. version 1.1. 0. ogc 04-094. open geospatial consortium. *Inc.*, 131pp, 2005.
- [37] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.