

VISUALIZAÇÃO DE DADOS ESPAÇO-TEMPORAIS DE OBSERVAÇÃO DA TERRA

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA (PIBIC/CNPq/INPE)

Roger Victor (FATEC, Bolsista PIBIC/CNPq)
E-mail: roger-victor@hotmail.com

Gilberto Ribeiro de Queiroz (DPI/OBT/INPE, Orientador)
E-mail: gribeiro@dpi.inpe.br

COLABORADOR

Eng. Ricardo Cartaxo (DPI/OBT/INPE)

Junho de 2016

Ficha será revisada pelo SID.

Dados Internacionais de Catalogação na Publicação

Cutter Sobrenome, Prenome(s) Completos do(s) Autor(es).
 Título da publicação / Nome Completo do Autor(es). - São José
 dos Campos: INPE, ano da publicação.

 Grau(Mestrado ou Doutorado em Nome do Curso) - Instituto
Nacional de Pesquisas Espaciais, São José dos Campos, ano de
defesa.

 Orientador: Nome completo do orientador(es).

 1. Assunto. 2. Assunto. 3. Assunto. 4. Assunto. 5. Assunto.
I. Título

CDU

Copyright AAAA do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright AAAA by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming or otherwise, without written permission from the INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

FOLHA DE APROVAÇÃO

CONFECCIONADA PELO SPG E INCLUÍDA PELO SID.

RESUMO

Os satélites de observação da Terra representam uma das fontes de dados mais importantes sobre o nosso planeta. Diversos estudos de sensoriamento remoto, como modelagem de ecossistemas terrestres, estudos de mudanças climáticas, estimativa de produtividade de culturas agrícolas e classificação do uso da terra, requerem a utilização do grande volume de dados geoespaciais gerados por esses satélites. Assim tem-se a necessidade de ferramentas de visualização de dados espaço-temporais na web, que forneçam de forma eficiente uma boa capacidade analítica, além de bons mecanismos de visualização para análise exploratória por parte dos usuários. Neste contexto, este projeto de Iniciação Científica tem por objetivo desenvolver uma ferramenta computacional para visualização de dados espaço-temporais de observação da Terra para bases massivas de dados geoespaciais, gerenciadas pelo servidor de bancos de dados matricial SciDB. Tal ferramenta, desenvolvida na forma de um serviço web em conformidade com o padrão OGC *Web Map Service* (WMS), irá propiciar aos usuários dos dados armazenados no SciDB uma visualização dinâmica que ajude a compreender a variação espaço-temporal do fenômeno observado. Através da interface WMS, este serviço poderá ser integrado a outras aplicações do INPE, como Aplicação de Validação do Projeto TerraClass Cerrado, Banco de Dados de Queimadas e a aplicação TerraBrasilis.

SPATIO-TEMPORAL DATA VISUALIZATION FOR EARTH OBSERVATION DATA

ABSTRACT

Earth Observation (EO) satellites are one of the most important sources of data on our planet. Several remote sensing studies, such as terrestrial ecosystems modeling, climate change studies, productivity estimate of agricultural crops and classification of land use, require the use of large volumes of geospatial data generated by these satellites. Thus, there is a need for spatio-temporal data visualization tools on the web capable of providing effective and good analytical support as well as mechanisms for exploratory data analysis by users. In this context, this research project aims to develop a computational tool for viewing space-time EO data for massive geospatial databases. These databases are managed by the array database system SciDB. Our intend is to develop these tool according to the OGC standard Web Map Service (WMS). Through this standard interface, SciDB's users will have mechanisms for dynamic visualization of EO data. It will also help users to understand the spatio-temporal variation of the observed phenomenon. The WMS interface for SciDB will enable us to integrate it to other INPE's applications, such as the TerraClass Cerrado Validation Tool, Fire Monitoring System and the general viewer TerraBrasilis.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1.1 - Classificação automática de cobertura da Terra.....	3
Figura 2.1 - Arquitetura Conceitual da Plataforma TerraLib 5	8
Figura 2.2 - Arquitetura do SciDB.....	10
Figura 2.3 - Matriz 3D representando uma série temporal de imagens	12
Figura 3.1 - Arquitetura do serviço WMS.....	13
Figura 3.2 - Diagrama de classes do serviço WMS	14
Figura 3.3 - Fragmento de código do arquivo CMake de construção do módulo WMS..	15
Figura 3.4 - Fragmento de código contendo definições dos métodos da classe WMS..	16
Figura 3.5 - Fragmento de código contendo definições de tipos especificados pelo OGC	17
Figura 3.6 - Fragmento de código utilizando a API C++ do SciDB.....	21
Figura 3.7 - Fragmento de código contendo a criação da imagem de resposta	22
Figura 4.1 - Documento XML com os metadados do serviço WMS	24
Figura 4.2 - Mensagem de tratamento do erro na URL da requisição GetCapabilities..	25
Figura 4.3 - Imagem gerada com o array teste armazenado no SciDB.....	26
Figura 4.4 - Mensagem de tratamento do erro na URL da requisição GetMap	26

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 3.1 - Parâmetros de requisição da operação GetCapabilities.....	19
Tabela 3.2 - Parâmetros de requisição da operação GetMap.....	21
Tabela 4.1 - Valores dos parâmetros da requisição teste da operação GetMap	25

LISTA DE SIGLAS E ABREVIATURAS

INPE	Instituto Nacional de Pesquisas Espaciais
OGC	Open Geospatial Consortium
WMS	Web Map Service
TWS	TerraLib GeoWeb Services
SIG	Sistema de Informações Geográficas
SGBD	Sistema de Gerenciamento de Banco de Dados

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	2
1.1. Objetivo.....	4
1.2. Estrutura do Trabalho	4
2 FUNDAMENTAÇÃO TEÓRICA.....	5
2.1. Visualização Dinâmica de Dados Espaço-Temporais.....	5
2.2. Serviços OGC	6
2.3. TerraLib 5	7
2.4. TerraLib GeoWeb Services (TWS).....	9
2.5. SciDB 9	
2.6. Multiresolução para Visualização de Dados Geográficos.....	11
3 DESENVOLVIMENTO	13
3.1. Definição de Classes.....	14
3.2. Criação do Módulo do Serviço WMS.....	14
3.3. Registro das operações do serviço no TWS.....	16
3.4. Registro das operações do serviço no TWS.....	17
3.5. Implementação da Operação GetMap.....	19
4 RESULTADOS	24
4.1. Operação GetCapabilities.....	24
4.2. Operação GetMap.....	25
5 CONCLUSÕES E TRABALHOS FUTUROS	27
5.1. Trabalhos Futuros	27

1 INTRODUÇÃO

A popularização dos Sistemas de Informação Geográfica (SIG) tem contribuído para um melhor uso e análise de dados geográficos dentro de várias disciplinas. Com base em avanços na tecnologia da informação, a dependência da sociedade em tais dados está crescendo. Conjuntos de dados geográficos estão cada vez mais sendo compartilhados e utilizados para diversos fins, como estudos de sensoriamento remoto, análise de tráfego urbano, sistemas de geoposicionamento, entre outros.

Os satélites de observação da Terra representam uma das fontes de dados mais importantes sobre o nosso planeta. O acervo de dados LANDSAT da USGS possui mais de cinco milhões de imagens da superfície terrestre, coletadas de forma contínua por um período de quase quarenta anos, compreendendo um volume de cerca de um petabyte de dados (Camara et al., 2014). Esses dados podem ser utilizados em diversas áreas de aplicação, desde estudos de mudanças climáticas, modelagem de ecossistemas terrestres, classificação automática de cobertura da Terra (Figura 1.1), aplicações agrícolas, aplicações socioeconômicas até projetos operacionais como PRODES (PRODES, 2015) e DETER (DETER, 2015).

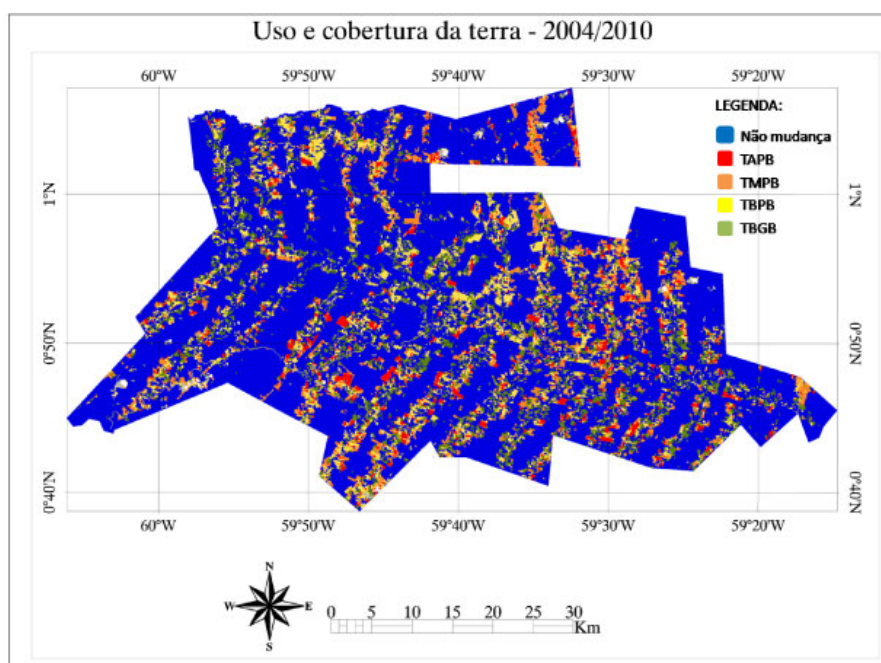


Figura 1.1 - Classificação automática de cobertura da Terra.

Fonte: Xaud e Epiphonio (2014)

Atualmente, um dos maiores desafios enfrentados pelos centros de dados de observação da Terra é organizar estas verdadeiras montanhas de dados, com volumes cada vez maiores, e prover ferramentas computacionais que ajudem a maximizar o valor e o impacto desses dados nessas diversas aplicações. Para isso, é necessário prover infraestruturas de dados que consigam gerenciar bases massivas de dados, que forneçam de forma eficiente uma boa capacidade analítica, além de bons mecanismos de visualização para análise exploratória por parte dos usuários.

Neste cenário, ferramentas de visualização na web que possibilitem aos usuários desses grandes acervos navegarem e interagirem com os dados de forma a realizar uma análise exploratória eficiente e satisfatória dos dados é de grande relevância. É importante ressaltar que o esforço gasto na preparação dos dados para fins de visualização representa uma parte substancial de muitos projetos de pesquisa, principalmente, quando estes têm que lidar com grandes volumes de dados (Fox, 2011).

1.1. Objetivo

Desenvolver uma ferramenta computacional para visualização dinâmica de dados espaço-temporais, desenvolvida na forma de um serviço web em conformidade com a especificação OGC Web Map Service, utilizando o banco de dados matricial SciDB.

1.2. Estrutura do Trabalho

Os capítulos deste trabalho estão organizados da seguinte maneira:

- **Fundamentação Teórica:** apresenta resumidamente alguns tópicos abordados neste trabalho;
- **Desenvolvimento:** descreve como o trabalho foi definido e implementado;
- **Resultados:** exemplifica os resultados obtidos;
- **Conclusão:** com base nos resultados obtidos são apresentadas as conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

2.1. Visualização Dinâmica de Dados Espaço-Temporais

Uma das formas de visualização mais interessante para os usuários de séries temporais de imagens de sensoriamento remoto consiste na visualização de mapas dinâmicos que permitam a análise da variação temporal a partir do conjunto de imagens do acervo de dados. Existem vários portais na web que fornecem visualizações “animadas”, geradas em escalas pré-definidas, e num período específico, como no portal da NASA chamado “*The Earth Observatory*” (NASA, 2015). No entanto, essas ferramentas não fornecem a forma de interação adequada aos usuários desses acervos de dados, que não podem realizar mudanças na escala de visualização para compreender melhor os dados do acervo para regiões mais específicas.

Por outro lado, existem especificações de serviços para visualização de dados na web bem conhecidas, como a especificação OGC Web Map Service - WMS (OGC(a), 2015). Um WMS é um serviço capaz de produzir mapas a partir de informações geográficas de forma dinâmica e consiste de três operações básicas:

- GetCapabilities: esta operação retorna um documento XML contendo metadados obtidos do servidor, como as camadas de informação disponíveis para visualização e os formatos de geração dos mapas;
- GetMap: retorna um mapa (figura) a partir das camadas solicitadas, realizando a renderização na escala desejada;
- GetFeatureInfo: obtém informações sobre um elemento particular de uma ou mais camadas de informação, retornando estes dados em um documento XML.

No contexto da especificação WMS, a palavra geração de mapas “dinâmicos” tem a conotação de que a cada requisição feita pelo cliente do serviço, um

novo mapa é gerado e entregue ao cliente, não havendo relação com o fato dos dados serem espaço-temporais. Tecnologias como MapServer e GeoServer são comumente empregadas para criação deste tipo de serviço.

Já os provedores de serviços WMS, comumente realizam um mapeamento de cada imagem do acervo para uma camada de informação, ou *layer* no jargão WMS, a ser disponibilizada no serviço. Ou seja, imagens de uma mesma cena (ou região) são disponibilizadas como diferentes camadas. O fato é que as ferramentas mais conhecidas, GeoServer e MapServer, não fornecem o suporte adequado para criação e disponibilização do serviço WMS para dados espaço-temporais.

No entanto, existem dois documentos complementares à especificação WMS pouco explorados: (1) o perfil para produtos de observação da Terra, conhecido por *OGC Web Map Services - Profile for EO Products* (OGC(b), 2015); e (2) um documento de melhores práticas para dados com a componente temporal, o documento *OGC Best Practice for using Web Map Services with Time-Dependent or Elevation-Dependent Data* (OGC(c), 2015). O primeiro documento já se encontra em desuso, enquanto o segundo foi produzido com um forte direcionamento as comunidades de Meteorologia e Oceanografia.

Neste trabalho, utilizaremos o documento de melhores práticas, com dados com a componente temporal para a construção de um serviço WMS que permita os usuários de grandes acervos de imagens de sensoriamento remoto visualizar os dados na escala desejada e de forma eficiente.

2.2. Serviços OGC

O *Open Geospatial Consortium* (OGC) é um consórcio internacional com mais de 521 companhias, agências governamentais e universidades, criado para promover o desenvolvimento de tecnologias que facilitem a interoperabilidade entre sistemas envolvendo informação espacial e localização. Os produtos do

trabalho do OGC são apresentados sob a forma de especificações de interfaces e padrões de intercâmbio de dados.

Padrões são a base para o sucesso da internet, tornando interoperáveis milhares de aplicações e tecnologias. A não interoperabilidade impede o compartilhamento de dados e recursos computacionais, causando um maior custo no desenvolvimento de tecnologia de informação geográfica. Assim um padrão descreve requisitos e recomendações que foram acordadas em consenso por um fórum, como a *International Organization for Standardization* (ISO), o *World Wide Web Consortium* (W3C) ou o *Open Geospatial Consortium* (OGC), permitindo que dados e processos fluam e interajam com o mínimo de barreiras possíveis. O OGC tem desempenhado um importante papel em tornar o acesso à informação espacial aberto.

Dentre os diversos padrões desenvolvidos pelo OGC, está a especificação Web Map Service - WMS (OGC(a), 2015), que define o serviço que está sendo implementado neste trabalho.

2.3. TerraLib 5

A TerraLib é definida como um projeto (CAMARA et al., 2008) que visa atender grandes demandas institucionais na área de Geoinformática, criando um ambiente para pesquisa e desenvolvimento de inovações em geoprocessamento.

TerraLib 5 é uma biblioteca de código aberto para o desenvolvimento de aplicações geográficas personalizadas. É escrita em linguagem C++ e usa bibliotecas de terceiros para fornecer funções típicas de um SIG (por exemplo, manuseio de geometria, processamento de imagem, sistemas de referência espacial), funções e algoritmos inovadores (por exemplo, processamento de dados espaço-temporais) (Figura 2.1).

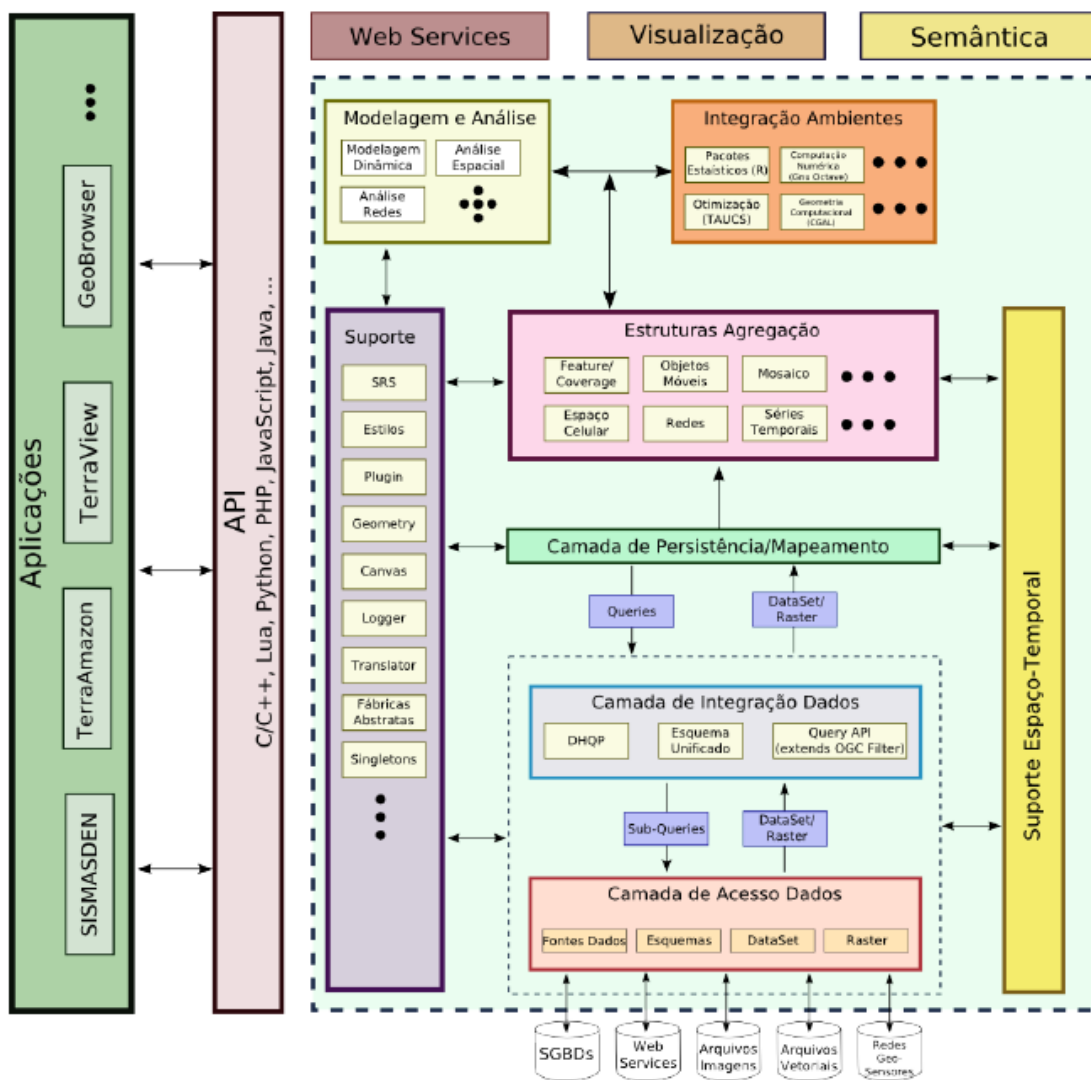


Figura 2.1 - Arquitetura Conceitual da Plataforma TerraLib 5

Fonte: QUEIROZ et al. (2010)

A TerraLib 5 tem como principais características (QUEIROZ et al., 2010):

- Camada de Acesso a Dados: acesso a diferentes tipos de fontes de dados (SGBD's, dados vetoriais, imagens, serviços web, entre outros);
- Camada de Integração de Dados: responsável por fornecer a capacidade de trabalhar com dados provenientes de diversas fontes heterogêneas, de uma maneira mais uniforme e simples;

- Camada de Persistência/Mapeamento: mapeia dados de diversas fontes de dados para diferentes finalidades;
- Camada de Estruturas de Agregação: fornece um mecanismo extensível capaz de introduzir novas representações de alto nível.

2.4. TerraLib GeoWeb Services (TWS)

TerraLib GeoWeb Services (TWS) é uma plataforma livre e de código aberto para o desenvolvimento de serviços web que lidam com dados de Observação da Terra. Construído utilizando os componentes da TerraLib, TWS é parte integrante da família TerraLib de geotecnologias (TWS, 2016).

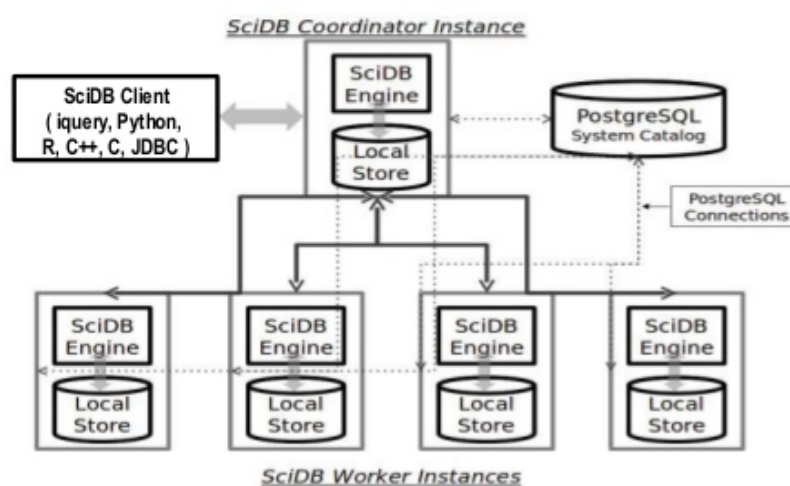
Compondo a lista de serviços a serem disponibilizados pelo TWS, os seguintes módulos estão em desenvolvimento: WTSS (Web Time Series Service), um serviço web para manipulação de séries temporais de dados de imagens de sensoriamento remoto; WCS (Web Coverage Service 2.0), um serviço para troca de dados com o SciDB; e por fim o WMS (Spatio-Temporal Web Map Service), um serviço web para a visualização de dados de imagens de sensoriamento remoto em forma de mapas dinâmicos (espaço-temporal) armazenados como matrizes 3D no SciDB. Todos esses serviços funcionam como módulos do TWS, compartilhando da mesma infraestrutura de armazenamento de dados do SciDB.

2.5. SciDB

O SciDB é uma plataforma aberta para gerenciamento e análise de dados científicos armazenados como matrizes multidimensionais (Stonebraker et al., 2011). Este sistema divide uma grande matriz em pedaços denominados chunks que são distribuídos entre diferentes instâncias de um cluster de bancos de dados (Figura 2.2). Cada instância do cluster controla um armazenamento local próprio, numa arquitetura denominada shared-nothing. Uma das instâncias é designada coordenadora, sendo responsável por mediar

toda a comunicação entre o cluster e as aplicações clientes e também por orquestrar a execução das consultas. As demais instâncias do cluster, denominadas trabalhadoras, apenas participam do processamento da consulta, que pode ser realizada com base em duas linguagens: Array Query Language (AQL) ou Array Functional Language (AFL).

SciDB System Architecture



“Shared Nothing” cluster of commodity hardware nodes
Interconnected with standard ethernet and TCP/IP

© Paradigm4 12

Figura 2.2 - Arquitetura do SciDB

Fonte: PARADIGM4 (2016)

Ao contrário de bancos de dados relacionais convencionais, concebidos em torno de um modelo de dados orientado por tabelas, o SciDB é uma base de dados de matrizes. O modelo de dados de matrizes nativo fornece armazenamento compacto de dados e alto desempenho em operações sobre os dados solicitados, tais como dados espaciais e dados temporais (séries temporais).

2.6. Multiresolução para Visualização de Dados Geográficos

Uma exploração interativa com o acervo de imagens é mais eficiente para o usuário, que pode navegar e definir a escala adequada para observação de um dado fenômeno. O servidor WMS em desenvolvimento terá a capacidade de responder consultas de visualização considerando os aspectos temporais dos dados de observação da Terra armazenados no SciDB. Para isso foi necessário projetar e desenvolver um esquema multiresolução sobre o SciDB.

Bancos de dados relacionais não foram originalmente projetados para gerenciar dados geoespaciais. Já um servidor de banco de dados com extensão espacial, como o PostGIS, suporta nativamente dados com dimensões espaciais, porém a manipulação de dados com mais de duas dimensões despense um grande esforço na elaboração de um modelo de dados compatível. Assim o modelo de banco de dados objeto-relacional é inadequado para a estrutura do serviço WMS que estamos desenvolvendo, visto que propõe uma exploração dos dados não somente espacial, mas também temporal. Dados espaço-temporais não se encaixam perfeitamente em tabelas relacionais, o modelo de dados de matrizes multidimensionais do SciDB é uma escolha natural para dados complexos como dados geoespaciais e temporais (Stonebraker et al., 2011).

Uma estratégia comumente empregada pelos sistemas de informações geográficas (SIGs) para acelerar a visualização de dados é a criação de pirâmides de multiresolução ou overviews, técnica descrita em detalhes em Vinhas et al. (2005). Diferentemente dos esquemas multiresolução encontrados nos tradicionais SIG, que levam em consideração apenas a componente espacial, um esquema multiresolução para matrizes multidimensionais armazenadas no SciDB necessita considerar o aspecto temporal (Figura 2.3).

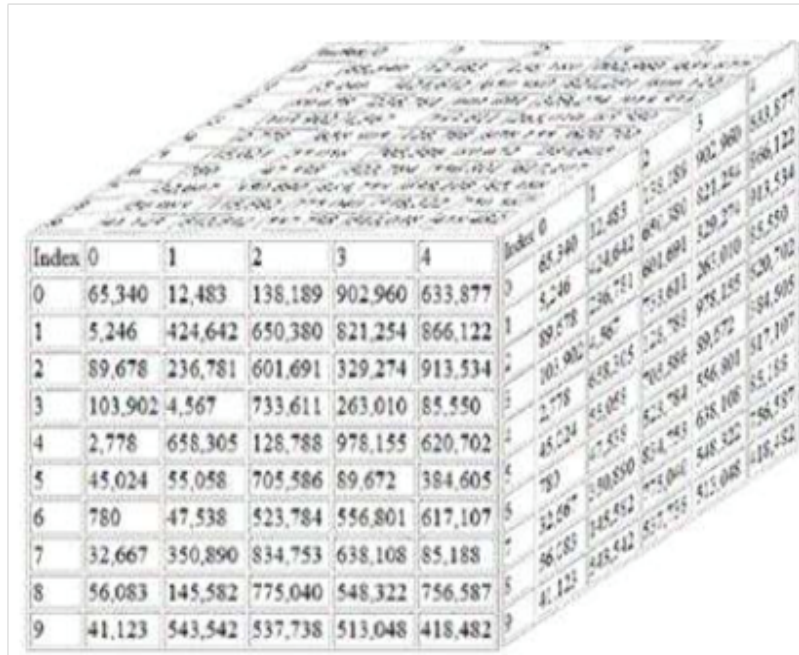


Figura 2.3 - Matriz 3D representando uma série temporal de imagens

Fonte: Adaptado de PARADIGM4 (2016)

3 DESENVOLVIMENTO

Neste capítulo será apresentada a metodologia utilizada no desenvolvimento deste trabalho. Concebido como um módulo do TerraLib GeoWeb Services (TWS), o serviço WMS que foi desenvolvido, se utiliza da arquitetura do mesmo e do armazenamento do SciDB (Figura 3.1). O sistema foi desenvolvido em classes, se utilizando do paradigma de programação orientado a objetos, adotando a linguagem de programação C++ para a implementação, assim como todos os outros serviços que compõe o TWS. Nas seções a seguir será descrito como foi feita a modelagem de classes e a implementação das operações do serviço WMS.

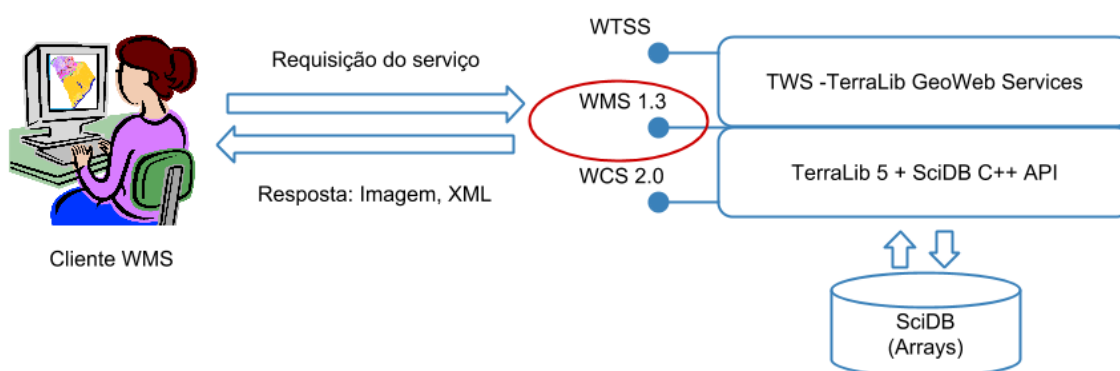


Figura 3.1 - Arquitetura do serviço WMS

Fonte: Autor

3.1. Definição de Classes

A Figura 3.2 contém o diagrama UML com a definição das classes do serviço WMS.

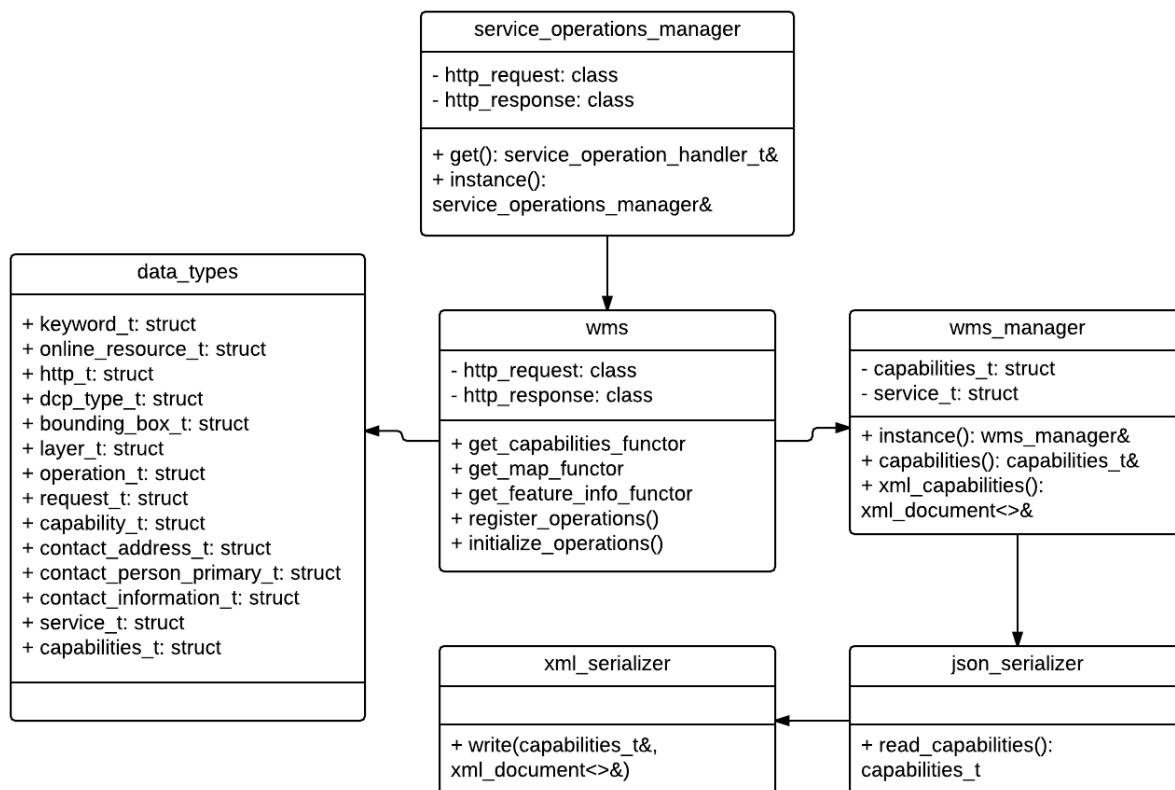


Figura 3.2 - Diagrama de classes do serviço WMS

Fonte: Autor

3.2. Criação do Módulo do Serviço WMS

Para fazer uso da arquitetura e dos recursos do TWS, primeiramente foi necessário criar um módulo para abrigar o serviço WMS. Se utilizando do CMake para a construção deste módulo, foi gerado um novo arquivo contendo os diretórios das bibliotecas inclusas e os módulos com os quais o serviço WMS irá se comunicar (Figura 3.3).

```

29 include_directories(${RAPIDJSON_INCLUDE_DIR})
30 include_directories(${RAPIDXML_INCLUDE_DIR})
31 include_directories(${terralib_INCLUDE_DIRS})
32 include_directories(${SCIDB_INCLUDE_DIR})
33 include_directories(${LIBGD_INCLUDE_DIR})
34
35 file(GLOB TWS_SRC_FILES ${TWS_ABSOLUTE_ROOT_DIR}/src/tws/wms/*.cpp)
36 file(GLOB TWS_HDR_FILES ${TWS_ABSOLUTE_ROOT_DIR}/src/tws/wms/*.hpp)
37
38 add_library(tws_mod_wms SHARED ${TWS_SRC_FILES} ${TWS_HDR_FILES})
39
40 target_link_libraries(tws_mod_wms tws_mod_scidb
41                       tws_mod_geoarray
42                       tws_mod_core
43                       terralib_mod_plugin
44                       terralib_mod_raster
45                       terralib_mod_srs
46                       ${SCIDB_CLIENT_LIBRARY}
47                       ${Boost_FILESYSTEM_LIBRARY}
48                       ${Boost_SYSTEM_LIBRARY}
49                       ${LIBGD_LIBRARY})
50
51 set_target_properties(tws_mod_wms
52                       PROPERTIES VERSION ${TWS_VERSION_MAJOR}.${TWS_VERSION_MINOR}
53                       SOVERSION ${TWS_VERSION_MAJOR}.${TWS_VERSION_MINOR})
54
55 install(TARGETS tws_mod_wms
56         EXPORT tws-targets
57         RUNTIME DESTINATION ${TWS_DESTINATION_RUNTIME} COMPONENT runtime
58         LIBRARY DESTINATION ${TWS_DESTINATION_LIBRARY} COMPONENT runtime
59         ARCHIVE DESTINATION ${TWS_DESTINATION_ARCHIVE} COMPONENT runtime)
60
61 install(FILES ${TWS_HDR_FILES}
62         DESTINATION ${TWS_DESTINATION_HEADERS}/tws/wms COMPONENT devel)
63
64 export(TARGETS tws_mod_wms APPEND FILE ${CMAKE_BINARY_DIR}/tws-exports.cmake)

```

Figura 3.3 - Fragmento de código do arquivo CMake de construção do módulo WMS

Fonte: Autor

No início do fragmento de código da Figura 3.3, são incluídos os diretórios das bibliotecas que são utilizadas pelo serviço WMS, em seguida, são definidos os arquivos de código fonte e os arquivos de cabeçalho do módulo. Então, o mesmo é adicionado ao projeto, juntamente com os arquivos de código fonte e de cabeçalho declarados anteriormente e é feita a conexão com os demais módulos e bibliotecas que são utilizadas pelo serviço WMS.

3.3. Registro das operações do serviço no TWS

Como classe principal do serviço foi definida a classe WMS, sendo responsável pela inicialização do serviço e registro das operações no módulo core do TWS (Figura 3.4).

```
279 void
280 tws::wms::register_operations()
281 {
282     tws::core::service_metadata service;
283
284     service.name = "wms";
285
286     // 1st WMS operation: GetCapabilities
287     {
288         tws::core::service_operation s_op;
289
290         s_op.name = "GetCapabilities";
291         s_op.description = "List the metadata, describing the layers available for visualization.";
292         s_op.handler = get_capabilities_functor();
293
294         service.operations.push_back(s_op);
295     }
296
297     // 2nd WMS operation: GetMap
298     {
299         tws::core::service_operation s_op;
300
301         s_op.name = "GetMap";
302         s_op.description = "Request the server to render a map giving a list of layers.";
303         s_op.handler = get_map_functor();
304
305         service.operations.push_back(s_op);
306     }
307
308     // 3rd WMS operation: GetFeatureInfo
309     {
310         tws::core::service_operation s_op;
311
312         s_op.name = "GetFeatureInfo";
313         s_op.description = "Retrieve information about an element of a particular layer.";
314         s_op.handler = get_feature_info_functor();
315
316         service.operations.push_back(s_op);
317     }
318
319     tws::core::service_operations_manager::instance().insert(service);
320 }
```

Figura 3.4 - Fragmento de código contendo definições dos métodos da classe WMS

Fonte: Autor

Para cada operação do serviço foi definida uma função, utilizando o padrão de projeto functor, que realizando a chamada de métodos pertencentes ao módulo

core do TWS, é capaz de gerenciar as requisições e repostas de cada operação. O método `register_operations()`, mostrado na Figura 7, define o serviço e suas operações, atribuindo nome, descrição e a respectiva função para gerenciamento das requisições e respostas.

3.4. Registro das operações do serviço no TWS

O objetivo da operação `GetCapabilities` é a obtenção das informações do serviço, retornando um documento XML contendo os metadados obtidos do servidor, como as camadas de informação disponíveis para visualização e os valores aceitáveis para os parâmetros de requisição.

Primeiramente foi definida a classe `DATA_TYPES`, um arquivo contendo diversas structs para abrigar os tipos de dados definidos pela especificação OGC WMS (OGC, 2015), necessários para a construção do documento XML que será retornado. A Figura 3.5 mostra a definição das estruturas `capabilities_t` e `service_t`, cada uma com seus atributos e subtipos.

```
277     //!< Base datatype for representing the service metadata.
278     struct service_t
279     {
280         std::string name;
281         std::string title;
282         std::string abstract;
283         std::vector<keyword_t> keyword_list;
284         online_resource_t online_resource;
285         contact_information_t contact_information;
286         std::string fees;
287         std::string access_constraints;
288         uint32_t layer_limit;
289         uint32_t max_width;
290         uint32_t max_height;
291     };
292
293     //!< Base datatype for WMS_Capabilities document.
294     struct capabilities_t
295     {
296         service_t service;
297         capability_t capability;
298         std::string version;
299     };
```

Figura 3.5 - Fragmento de código contendo definições de tipos especificados pelo OGC

Fonte: Autor

Para realizar o controle da leitura dos metadados e da construção do documento XML, foi definida a classe WMS_MANAGER, que através do padrão de projeto Singleton, faz a definição da instância da classe e de seus atributos. Seu método construtor é responsável pelo acesso ao arquivo JSON, armazenado junto aos arquivos de configuração do TWS, contendo os metadados do serviço.

Para a leitura do arquivo JSON, é feita a chamada do método read_capabilities(), definido na classe JSON_SERIALIZER, que através da biblioteca RapidJSON, realiza o parseamento do arquivo, armazenando os dados nas estruturas definidas anteriormente na classe DATA_TYPES.

Em seguida é realizada a chamada do método write(), contido na classe XML_SERIALIZER, responsável pela construção do documento XML de resposta, utilizando a biblioteca RapidXML.

Para a solicitação da operação GetCapabilities, a especificação OGC WMS (OGC, 2015) define como sendo mandatório informar na URL de requisição, os valores dos parâmetros SERVICE e REQUEST. Há também outros parâmetros, que podem ser informados opcionalmente, como mostra a Tabela 1.

Parâmetro	Mandatório/ Opcional	Descrição
VERSION=version	Opcional	Versão
SERVICE=WMS	Mandatório	Tipo do serviço.
REQUEST=GetCapabilities	Mandatório	Nome da requisição.

Parâmetro	Mandatário/ Opcional	Descrição
FORMAT=MIME_type	Opcional	Formato de saída dos metadados do serviço.
UPDATESEQUENCE=string	Opcional	Sequência numérica para controle de cache.

Tabela 3.1 - Parâmetros de requisição da operação GetCapabilities

Fonte: Adaptado de OGC (2015)

A validação dos parâmetros SERVICE e REQUEST, é realizada pelo módulo core do TWS, responsável por gerenciar o recebimento das requisições e tratamento dos parâmetros recebidos, retornando uma exceção caso o nome do serviço, ou da operação requerida, não seja válido. Sendo válidos os parâmetros de requisição, é feita a chamada da função get_capabilities_functor, que irá retornar o documento XML com os metadados do serviço.

3.5. Implementação da Operação GetMap

A operação GetMap é a responsável pela funcionalidade de visualização dos dados, retornando um mapa (figura) a partir das camadas solicitadas, realizando a renderização na escala desejada.

Para se realizar a requisição da operação GetMap, o OGC define diversos parâmetros, mandatórios e opcionais, necessários para a geração do mapa, como os layers desejados para a visualização, intervalo de tempo dos valores, tamanho da imagem a ser gerada, como mostra em detalhes a Tabela 2.

Parâmetro	Mandatário/ Opcional	Descrição
VERSION=1.3.0	Mandatário	Versão
REQUEST=GetMap	Mandatário	Nome da requisição.
LAYERS=layer_list	Mandatário	Lista separada por vírgula com uma ou mais camadas.
STYLES=style_list	Mandatário	Lista separada por vírgula com um estilo por layers para a renderização.
CRS=namespace:identifier	Mandatário	Sistema de Coordenadas de Referência.
BBOX=minx,miny,maxx, maxy	Mandatário	Área de seleção (cantos inferior esquerdo e superior direito) em unidades CRS.
WIDTH=output_width	Mandatário	Largura, em pixels, da imagem do mapa.
HEIGHT=output_height	Mandatário	Altura, em pixels, da imagem do mapa.
FORMAT=output_format	Mandatário	Formato de saída da imagem gerada.
TRANSPARENT=TRUE/F ALSE	Opcional	Transparência do plano de fundo da imagem, por padrão, FALSE.
BGCOLOR=color_value	Opcional	Valor hexadecimal das cores RGB para o plano de fundo da imagem, por padrão, 0xFFFFFFFF.
EXCEPTIONS=exception_ format	Opcional	Formato em que as exceções serão reportadas pelo serviço WMS, por padrão, XML.

Parâmetro	Mandatário/ Opcional	Descrição
TIME=time	Opcional	Intervalo de tempo dos valores das camadas desejadas.
ELEVATION=elevation	Opcional	Valores de elevação das camadas desejadas.

Tabela 3.2 - Parâmetros de requisição da operação GetMap

Fonte: Adaptado de OGC (2015)

A validação do parâmetro REQUEST, assim como na operação GetCapabilities, é realizada pelo módulo core do TWS, que sendo válido o nome da operação na requisição, realiza a chamada da função get_map_functor. Esta função, primeiramente, tem de validar os demais parâmetros mandatórios da requisição, retornando uma exceção caso algum deles seja inválido ou omitido.

Após esta etapa, é feita a preparação para a renderização da imagem a ser retornada. Como primeiro passo é realizada a consulta ao array, armazenado no SciDB, dos layers solicitados (Figura 3.6).

```

223 // get a connection from the pool in order to retrieve the image data
224 std::unique_ptr<tw::scidb::connection> conn(tw::scidb::connection_pool::instance().get());
225
226 std::string str_af1 = "project(regrid( " + layers[0] + ", " + std::to_string(regrid_width) +
227     ", " + std::to_string(regrid_height) + ", avg(val) as val), val)";
228
229 boost::shared_ptr< tw::scidb::QueryResult > qresult = conn->execute(str_af1, true);
230
231 std::vector<double> values;
232
233 const tw::scidb::ArrayDesc& array_desc = qresult->array->getArrayDesc();
234 const tw::scidb::Attributes& array_attributes = array_desc.getAttributes(true);
235 const tw::scidb::AttributeDesc& attr = array_attributes.front();
236
237 std::shared_ptr< tw::scidb::ConstArrayIterator > array_it = qresult->array->getConstIterator(attr.getId());
238
239 tw::scidb::fill(values, array_it.get(), attr.getType());

```

Figura 3.6 - Fragmento de código utilizando a API C++ do SciDB

Fonte: Autor

Como primeiro teste de implementação da operação GetMap, foi gerado no SciDB um array, contendo um atributo de valor inteiro, variando entre 0 e 2 aleatoriamente, a ser utilizado para a geração de uma imagem de teste. Para a consulta deste array armazenado, foi utilizada a função project do SciDB, cujo resultado é um array com um subconjunto dos atributos do array de origem. Os valores retornados desta consulta ao SciDB, são então armazenados em um vetor, declarado dentro da função get_map_functor().

Em seguida, utilizando a biblioteca GD, a imagem é propriamente criada (Figura 3.7).

```
241 // create a GD Image
242
243 gdImagePtr img = gdImageCreateTrueColor(width, height);
244
245 int red = gdTrueColorAlpha(255, 0, 0, 0);
246
247 int green = gdTrueColorAlpha(0, 255, 0, 0);
248
249 int blue = gdTrueColorAlpha(0, 0, 255, 0);
250
251 for(uint32_t i = 0; i != height; ++i)
252     for(uint32_t j = 0; j != width; ++j)
253         if(values[i*100+j] == 0)
254             gdImageSetPixel(img, j, i, red);
255         else if(values[i*100+j] == 1)
256             gdImageSetPixel(img, j, i, green);
257         else
258             gdImageSetPixel(img, j, i, blue);
```

Figura 3.7 - Fragmento de código contendo a criação da imagem de resposta

Fonte: Autor

Para a alocação de memória, necessária a geração da imagem, é feita a chamada da função gdImageCreateTrueColor(), passando como parâmetros a altura e a largura da imagem a ser criada. Para a definição das cores da imagem, a biblioteca GD possui a função gdTrueColorAlpha(), que informando os parâmetros dos valores dos canais RGB (vermelho, verde, azul) e

transparência, é capaz de gerar uma cor correspondente, a ser associada a um pixel da figura.

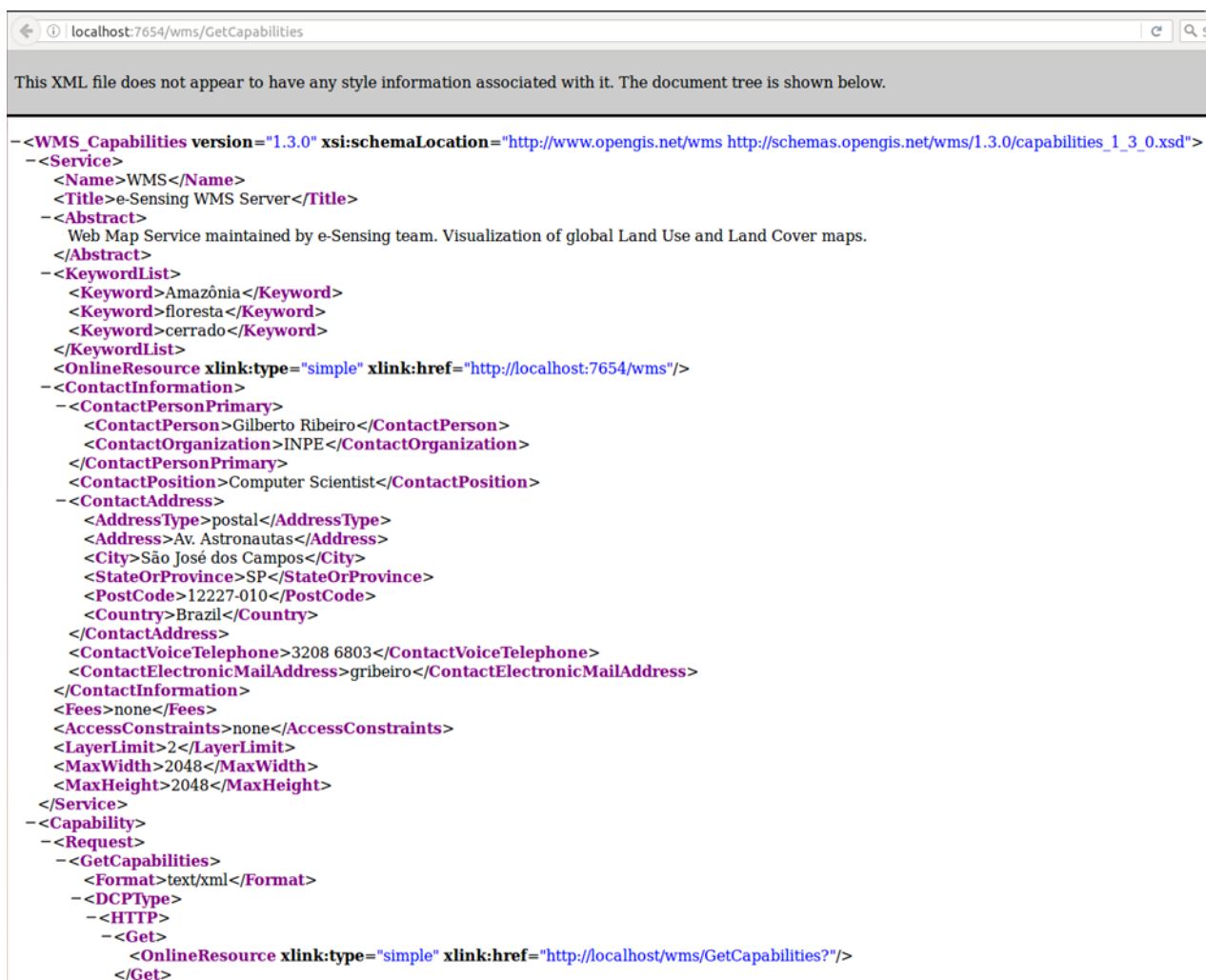
Assim, como mostra o fragmento de código da figura 11, foram definidas as três cores primárias, que foram utilizadas na geração da imagem de teste, onde cada valor do vetor de valores, foi associado a um pixel da figura, através da função `gdImageSetPixel()`, informando a imagem, posição e cor.

4 RESULTADOS

Neste capítulo serão apresentados os resultados obtidos na requisição de cada uma das operações do serviço WMS.

4.1. Operação GetCapabilities

Em uma requisição da operação GetCapabilities, é mandatório informar os valores dos parâmetros SERVICE e REQUEST. Assim, para uma URL de requisição válida (SERVICE=wms, REQUEST=GetCapabilities), a resposta será um documento XML, contendo os metadados do serviço (Figura 4.1).



```
-<WMS_Capabilities version="1.3.0" xsi:schemaLocation="http://www.opengis.net/wms http://schemas.opengis.net/wms/1.3.0/capabilities_1_3_0.xsd">
- <Service>
  <Name>WMS</Name>
  <Title>e-Sensing WMS Server</Title>
  <Abstract>
    Web Map Service maintained by e-Sensing team. Visualization of global Land Use and Land Cover maps.
  </Abstract>
  <KeywordList>
    <Keyword>Amazônia</Keyword>
    <Keyword>floresta</Keyword>
    <Keyword>cerrado</Keyword>
  </KeywordList>
  <OnlineResource xlink:type="simple" xlink:href="http://localhost:7654/wms"/>
- <ContactInformation>
  <ContactPersonPrimary>
    <ContactPerson>Gilberto Ribeiro</ContactPerson>
    <ContactOrganization>INPE</ContactOrganization>
  </ContactPersonPrimary>
  <ContactPosition>Computer Scientist</ContactPosition>
  <ContactAddress>
    <AddressType>postal</AddressType>
    <Address>Av. Astronautas</Address>
    <City>São José dos Campos</City>
    <StateOrProvince>SP</StateOrProvince>
    <PostCode>12227-010</PostCode>
    <Country>Brazil</Country>
  </ContactAddress>
  <ContactVoiceTelephone>3208 6803</ContactVoiceTelephone>
  <ContactElectronicMailAddress>gribeiro</ContactElectronicMailAddress>
</ContactInformation>
  <Fees>none</Fees>
  <AccessConstraints>none</AccessConstraints>
  <LayerLimit>2</LayerLimit>
  <MaxWidth>2048</MaxWidth>
  <MaxHeight>2048</MaxHeight>
</Service>
- <Capability>
  <Request>
    <GetCapabilities>
      <Format>text/xml</Format>
      <DCPType>
        <HTTP>
          <Get>
            <OnlineResource xlink:type="simple" xlink:href="http://localhost/wms/GetCapabilities?"/>
          </Get>
        </HTTP>
      </DCPType>
    </GetCapabilities>
  </Request>
</Capability>
</WMS_Capabilities>
```

Figura 4.1 - Documento XML com os metadados do serviço WMS

Fonte: Autor

No caso de uma solicitação da operação onde a URL de requisição esteja mal formada, por exemplo, o valor do parâmetro REQUEST esteja incorreto, é realizado o tratamento desta exceção retornando uma mensagem alertando para o erro na URL de requisição (Figura 4.2).

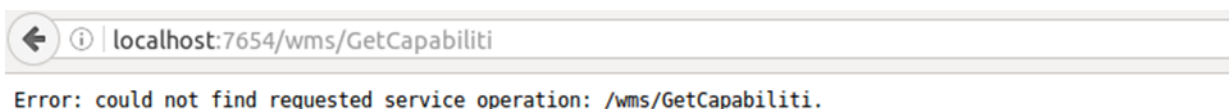


Figura 4.2 - Mensagem de tratamento do erro na URL da requisição GetCapabilities

Fonte: Autor

4.2. Operação GetMap

Para a requisição da operação GetMap, os parâmetros mandatórios na URL são VERSION, LAYERS, CRS, BBOX, WIDTH, HEIGHT e FORMAT. A Tabela 3, mostra os valores utilizados nestes parâmetros, em uma requisição teste da operação.

Parâmetro	Valor
VERSION	1.3.0
LAYERS	mod13q1
CRS	4326
BBOX	-180,-90,180,90
WIDTH	1000
HEIGHT	1000
FORMAT	image/png

Tabela 4.1 - Valores dos parâmetros da requisição teste da operação GetMap

Fonte: Autor

A resposta a esta solicitação do serviço é a imagem gerada com os valores do array de teste armazenado no SciDB (Figura 4.3).

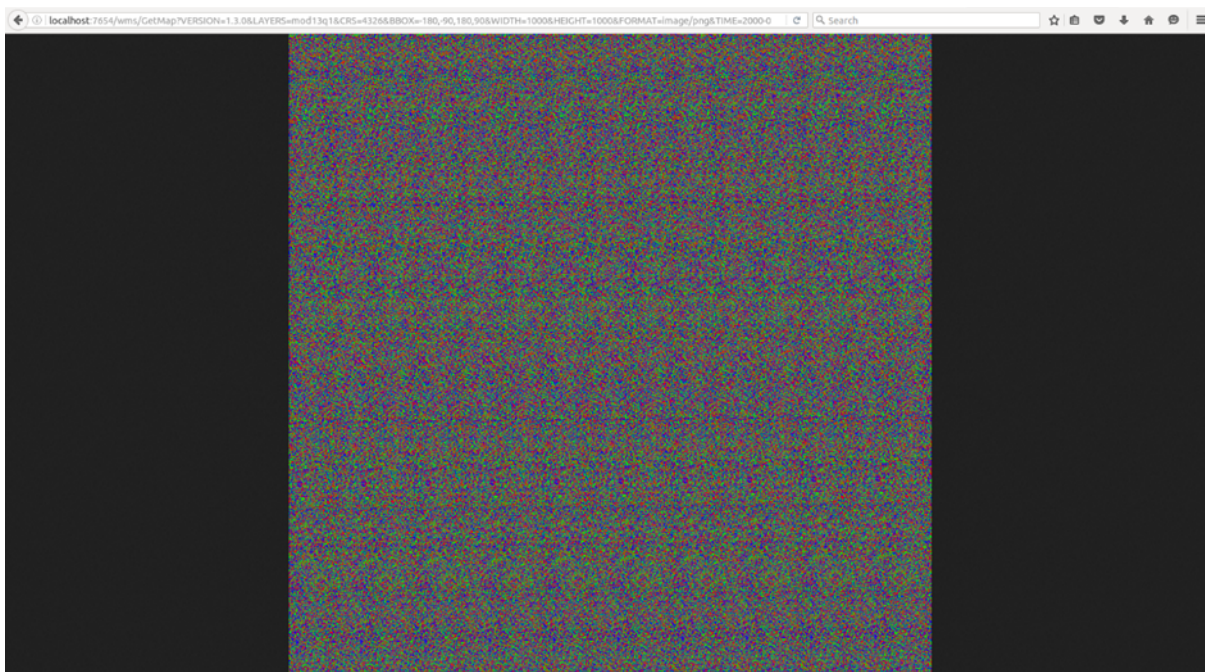


Figura 4.3 - Imagem gerada com o array teste armazenado no SciDB

Fonte: Autor

Assim como na operação GetCapabilities, os erros na informação ou a omissão de um parâmetro obrigatório na URL de requisição da operação GetMap, são tratados através de uma mensagem de erro de resposta ao usuário. Na Figura 4.4 foi feita uma solicitação da operação omitindo os parâmetros necessários, o resultado é uma mensagem informando os valores necessários a requisição da operação GetMap.

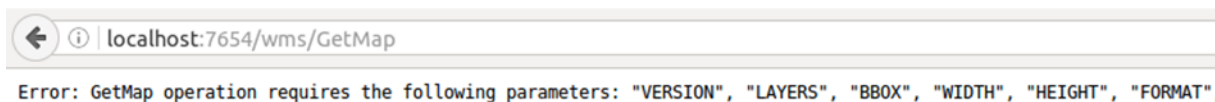


Figura 4.4 - Mensagem de tratamento do erro na URL da requisição GetMap

Fonte: Autor

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho se propôs a desenvolver um serviço WMS baseado na especificação OGC, utilizando a arquitetura de bancos de dados do SciDB. Como definido pela especificação, foram implementadas as operações mandatórias do serviço, GetCapabilities e GetMap.

O desenvolvimento deste trabalho permite concluir que:

- A arquitetura fornecida pelo TerraLib GeoWeb Services (TWS) se mostrou promissora e favorável ao desenvolvimento do trabalho;
- O modelo de dados matricial do SciDB ainda requer uma maior pesquisa e estudo, para se fazer um melhor uso dele;
- A implementação de um web service, de acordo com a especificação OGC WMS, se mostra extensa e desafiadora, com um grande potencial de contribuição a área de Geoinformática.

5.1. Trabalhos Futuros

Este trabalho não encerra as pesquisas e as possibilidades no desenvolvimento deste serviço WMS, sendo possíveis as seguintes melhorias e trabalhos futuros:

- Aperfeiçoamento da operação GetMap, contemplando a funcionalidade de redimensionamento das imagens e definição de mais estilos de visualização;
- Implementação da operação GetFeatureInfo, definida como opcional na especificação OGC;
- Desenvolvimento de uma aplicação cliente para a utilização das imagens geradas pelo serviço WMS.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMI, M. Estimativa da data de plantio da soja por meio de séries temporais de imagens MODIS. 2010. 161 p. Tese (Doutorado em Sensoriamento Remoto) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2010. Disponível em: <<http://urlib.net/8JMKD3MGP7W/3897URE>>. Acesso em: maio de 2016.

CAMARA, G.; EGENHOFER, M.; FERREIRA, K.; ANDRADE, P. R.; QUEIROZ, G.; SANCHEZ, A.; JONES, J.; VINHAS, L. Fields as a Generic Data Type for Big Spatial Data. GIScience, 2014.

CÂMARA, G.; VINHAS, L.; QUEIROZ, G. R.; FERREIRA, K. R.; MONTEIRO, A. M.; CARVALHO, M.; CASANOVA, M. TerraLib: An open-source GIS library for large-scale environmental and socio-economic applications. In: HALL, B.; LEAHY, M (Eds). Open Source Approaches in Spatial Data Handling: Advances in Geographic Information Science. Springer, 2010. p. 247-270.

FOX, P.; HENDLER, J. Changing the equation on scientific data visualization. Science, v. 331, n. 6018, pp. 705-708, 2011.

OGC. Web Map Service. Disponível em: <<http://www.opengeospatial.org>>. Acesso em: : maio de 2016.

NASA – Earth Observatory. Disponível em: <<http://earthobservatory.nasa.gov>>. Acesso: maio de 2016.

OGC(a). Web Map Service. Disponível em: <<http://www.opengeospatial.org/standards>>. Acesso: : maio de 2016.

OGC(b). Web Map Services - Application Profile for EO Products. Disponível em: <<http://www.opengeospatial.org/standards>>. Acesso: maio de 2016.

OGC(c). OGC Best Practice for using Web Map Services (WMS) with Time-Dependent or Elevation-Dependent Data (1.0). Disponível em: <<http://www.opengeospatial.org/standards>>. Acesso: maio de 2016.

PARADIGM4. SciDB Manual. Disponível em: <<http://scidb.org>>. Acesso em: : maio de 2016.

Projeto PRODES - Monitoramento ds Floresta Amazônica Brasileira por Satélite. Disponível em: <<http://www.obt.inpe.br/prodes>>. Acesso: maio de 2016.

QUEIROZ, G. R.; FERREIRA, K. R.; VINHAS, L.; CÂMARA, G.; MONTEIRO, A. M. V.; GARRIDO, J. C. P.; HARA, L.; XAVIER, M.; CASTEJON, E. F.; SOUZA, R. C. M. d. Terralib 5.0: supporting data-intensive giscience. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE, 10. (WORCAP), São José dos Campos. Anais... São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2010. 13, 14

RESCHA, B.; WOHLFAHRTE, R.; WOSNIOK, C. Web-based 4D visualization of marine geo-data using WebGL. Cartography and Geographic Information Science, v. 41, n. 3, pp. 235–247, 2014.

XAUD, M. R.; EPIPHANIO, J. C. N. Dinâmica do uso e cobertura da terra no sudeste de Roraima utilizando técnicas de detecção de mudanças. **Acta Amaz.**, v. 44, n. 1, 2014.

Sistema DETER - Sistema de Detecção de Desmatamento em Tempo Real. Disponível em: <<http://www.obt.inpe.br/deter>>. Acesso: maio de 2016.

STONEBRAKER, M.; BROWN, P.; POLIAKOV, A.; RAMAN, S. The architecture of SciDB. In Proceedings of the 23rd international conference on Scientific and statistical database management (SSDBM'11), Judith Bayard Cushing, James French, and Shawn Bowers (Eds.). Springer-Verlag, Berlin, Heidelberg, 2011, 1-16.

TWS - TerraLib GeoWeb Services. Disponível em: < <https://github.com/e-sensing/tws>>. Acesso: maio de 2016.

VINHAS, L.; SOUZA, R. C. M.; CAMARA, G. Image Data Handling in Spatial Databases. Geoinfo 2003.

