



# Integração de Sistemas de Modelagem Numérica Oceanográfica e Sistemas Computacionais Autônomos por meio de WEB SERVICE

Patrick Serpa<sup>1,2</sup>, Leila Weitzel<sup>1</sup>, Leandro Calado<sup>2</sup>

<sup>1</sup>Departamento de Computação - Instituto de Ciência e Tecnologia - UFF - Campus Rio das Ostras

<sup>2</sup>Instituto de Estudos do Mar Almirante Paulo Moreira - IEAPM, Arraial do Cabo - RJ, Brasil

**Resumo**—A Oceanografia Operacional hoje é uma realidade, sendo necessário que sistemas computacionais visem o aprimoramento de modelos numéricos e a viabilização de sistemas numéricos oceanográficos. O principal objetivo desta pesquisa é estudar e projetar um Sistema que utiliza os paradigmas de Web Service com intuito de integrar sistemas de modelagem numérica oceanográfica e sistemas computacionais autônomos.

O sistema provê uma camada de abstração entre aplicações tais como Matlab, Fortran, Python, Java Script e a infraestrutura de rede subjacente. O sistema é baseado na arquitetura REST e no protocolo JavaScript Object Notation (JSON). Assim, tanto os módulos internos quanto sistemas externos se comunicam de forma distribuída, com baixo acoplamento e transparente, independentemente de sua localidade, linguagem de desenvolvimento e plataforma de hardware. Para a Oceanografia Operacional, essas características trazem algumas vantagens, por exemplo, proporciona maior poder de processamento para a execução de modelos numéricos usando hardware específico e ainda é possível integrar novos modelos numéricos. Nesse sentido, o sistema proposto possibilita a evolução do Sistema de Previsão Oceanográfico Numérico especialmente nas operações do Instituto de Estudos do Mar Almirante Paulo Moreira, órgão da Marinha do Brasil, bem como outras pesquisas oceanográficas em geral.

**Palavras-chave**—Web Service, REST, Oceanografia, Previsão Numérica

**Abstract**—Operational Oceanography is a reality nowadays, making it necessary for computer systems to improve the numerical models and the feasibility of numerical oceanographic systems. The main goal of this research is to study and design a system that uses Web Services paradigm in order to integrate oceanographic numerical modeling systems and autonomous computational systems. The system aims to provide an abstraction layer between applications such as Matlab, Fortran, Python, Java Script and the underlying network infrastructure. It is based on the REST architecture and the JavaScript Object

Notation (JSON) protocol. Thus, both internal modules and external systems can communicate in a distributed, low-coupling and transparent manner, regardless of their location, development language and hardware platform. For Operational Oceanography, these characteristics bring some advantages including, for instance, the fact that it provides greater processing power for the execution of numerical models using specific hardware and it makes it possible to integrate new numerical models. Hence, the proposed system enables the evolution of the Numerical Oceanographic Prediction System, specially core operations of the Almirante Paulo Moreira Institute of Sea Studies (Brazilian Navy), as well as other oceanographic researches in general.

**Index Terms**—Web Service, REST, Oceanography, Numerical Forecast

## I. INTRODUÇÃO

A Oceanografia (ou Ciências do Mar) é a ciência do ramo das geociências que se dedica ao estudo dos oceanos e zonas costeiras sob os aspectos físicos, químicos, biológicos e geológicos [1].

As previsões a respeito das condições atmosféricas sofreram um aprimoramento sensível desde o início do século XX, no trabalho pioneiro de Richardson [2] até os dias atuais, devido ao melhor conhecimento dos fenômenos meteorológicos / oceanográficos, ao refinamento das técnicas de solução numérica e ao exponencial aumento da capacidade computacional. O advento de novas tecnologias e equipamentos nessa área, tais como, navios de pesquisa, sensores em bóias meteo-oceanográficas, dados orbitais (satélites) e Sistemas de Posicionamento Global (GPS, do inglês *Global Positioning Systems*), alargaram o âmbito e o potencial da investigação no contexto marítimo, tornando-o assim um estudo interdisciplinar em sintonia com as mudanças tecnológicas [3].

Desta forma, desde o final do século XX, a ciência oceanográfica vive a transição do modo de pesquisa para o modo operacional. A Oceanografia Operacional (OO) é

formada por três componentes básicos, a saber:

- Sistema de monitoramento do oceano através de coleta de dados tanto *in situ*, quanto através de sensores remotos;
- Sistema de Previsão Numérica Oceanográfica (SPNO) capaz de fornecer diagnósticos e previsões do oceano, e
- Sistema Operacional de pós processamento e armazenamento dos dados e disponibilização para a comunidade, através da rede mundial de computadores.

O primeiro componente da OO pode ser definido como o fornecimento de informações científicas e de previsões sobre o estado do mar, de forma rotineira, e com rapidez suficiente para que os usuários possam utilizar tais informações para tomar decisões, antes que ocorram alterações significativas das condições ambientais [4]. Hoje tais prognósticos, tanto em pequena escala de tempo (alguns dias), quanto em escala climática (meses a anos), fazem parte das atividades do cotidiano de muitas comunidades.

O segundo componente, um SPNO faz uso de uma série de modelos numéricos obtidos pelos sistemas de monitoramento. Pode-se definir a modelagem numérica como o processo pelo qual um conjunto de equações diferenciais e as condições de contorno associadas são transformadas em sistemas de equações algébricas que são resolvidas por técnicas computacionais [5], gerando os Modelos Numéricos Oceanográficos, que são ferramentas computacionais utilizadas para a investigação e previsão das condições físicas do mar. Lembrando apenas que estas rotinas estão em pleno funcionamento há alguns anos no IEPAM, sendo assim foram desenvolvidas por geólogos, geógrafos, etc. Esses pesquisadores não têm a “expertise” na área de computação, logo não são capazes de implementar a modelagem numérica através de potros paradigmas, como por exemplo, Inteligência Artificial. Por esta justificativa é que se faz importante a presente pesquisa, auxiliar na “orquestração” de um conjunto de rotinas para diminuir o tempo das previsões das condições físicas do mar.

Esses modelos utilizam medições e teorias sobre o comportamento do oceano, de modo a possibilitar simulações e previsões dos processos que nele ocorrem. Para a previsão das condições dos mares e oceanos existem, portanto, diversas medições de diferentes características, como por exemplo, a temperatura à superfície da água, a intensidade e direção do vento ou correntes marítimas, altura e direção das ondas, período de onda, etc. Estas características são exemplos de dados com variações no espaço e no tempo, de modo que existem múltiplas variáveis com diferentes magnitudes a serem representadas.

Os modelos numéricos de oceano evoluíram muito nos últimos anos, não apenas pela melhor compreensão dos processos físicos, mas também graças ao aumento substantivo do poder computacional, que permite não somente maior rapidez de processamento, como uma representação mais fiel das equações diferenciais por seus pares algébricos. Tal

revolução do poder computacional tornou a modelagem do oceano prática e a contribuição da disponibilidade de dados obtidos por sensores remotos a bordo de satélites, a custos cada vez menores, resulta num conjunto de dados globais [4].

O terceiro componente faz parte de um conjunto de sistema operacional para realizar o pós-processamento dos dados, responsável pelo armazenamento destes dados coletados e gerados e proporcionar a distribuição de dados e a prestação de uma variedade de serviços com documentos eletrônicos e disponibilização para a comunidade, através da rede mundial de computadores

A tecnologia Web Services permite que tanto os módulos internos quanto sistemas externos possam se comunicar de forma distribuída, com baixo acoplamento e de forma transparente, independentemente de sua localidade, linguagem de desenvolvimento e plataforma de hardware. Para OO essas características trazem uma série de vantagens, pois propiciam obter maior poder de processamento para execução dos modelos numéricos utilizando hardwares específicos, possibilitam a evolução do SPNO integrando novos modelos numéricos para melhoria das técnicas de previsão e permitem que a coleta e a disponibilização de dados possam ser realizadas de maneira mais eficiente.

Desta forma, baseado no contexto descrito nos parágrafos acima, o objetivo deste trabalho é apresentar um estudo de caso com o uso de Web Services na integração de sistemas de modelagem numérica oceanográfica e sistemas computacionais autônomos, de modo que estas aplicações possam interagir entre si.

Com a integração dos sistemas espera-se subsidiar operações do Instituto de Estudos do Mar Almirante Paulo Moreira (IEAPM), órgão da Marinha do Brasil, bem como pesquisas Oceanográficas na região Sudeste brasileira, através de um módulo de previsão oceanográfica. Em última análise, a integração das aplicações visa permitir maior eficiência e autonomia nas pesquisas neste domínio.

O artigo está organizado da seguinte maneira: na segunda seção é descrita uma breve fundamentação teórica sobre Web Service e REST. Na terceira seção apresenta-se o domínio da aplicação, ou seja, os processos do SPNO. Na quarta é apresentado um desenho experimental. Na quinta e última seção relata-se a discussão dos processos e são sugeridos trabalhos futuros.

## II. O SPNO DO IEPAM

A pesquisa referente ao SPNO no Instituto de Estudos do Mar Almirante Paulo Moreira (IEAPM) é conduzida pelo Grupo de Pesquisas em Modelagem Numérica, onde contribuem para a evolução do sistema desde orientandos até professores estrangeiros convidados. Os estudos baseiam-se em dois componentes, o sistema de coleta de dados, que funciona como plataforma para coleta e disponibilização de dados, e o SPNO, que reúne um conjunto de métodos na previsão da evolução das condições oceanográficas.

Atualmente, coletar, processar, analisar e disponibilizar dados e informações são tarefas realizadas por especialistas. A previsão é realizada de forma isoladas e sem comunicação

entre as diferentes etapas, necessitando que em cada etapa as variáveis de entrada e saída demandem da ação do especialista, o que traz dependência em razão de sua disponibilidade para realizar todo o processo de previsão em tempo hábil.

Explicando de forma simples como se dá todo o processo de previsão das condições do mar no IEAPM. Primeiro, dados brutos são coletados de servidores de Institutos Ambientais espalhados fisicamente ao redor do mundo. Esses dados são coletados por sensores de diferentes tipos. Dados já processados também são coletados. Todos estes dados juntos servem de “input” para diferentes cálculos de previsão numérica. Depois, cálculos são feitos localmente (ou remotamente) em diferentes plataformas com diferentes paradigmas de programação (em fortran, matlab, etc.) cada um produzindo diferentes formatos de saída. Tudo isso deve ser “orquestrado” por um sistema que, automaticamente, busque os dados (ou os resultados de processamento), encaminhem para outros processamentos (em outros servidores ou não) e tenham como resultado a previsão das condições climáticas para determinado período. E que os resultados sejam publicados (e/ou armazenados) em um servidor para que pesquisadores possam ter acesso. Na seção V “Processos Do Sistema De Previsão Numérica Oceanográfica – Spno” é feita toda a descrição de como se dá a previsão em detalhes com figuras ilustrativas. A figura 5 mostra esquematicamente como os webservices serão “orquestrados” nesta pesquisa e como os passos descritos acima foram modelados.

Os dados coletados e gerados, por todos os componentes da OO, são fundamentais não só para o próprio SPNO mas também para o público externo das demais áreas da Pesquisa. No IEAPM, por exemplo, a Divisão de Acústica Submarina utiliza as previsões em seus modelos de previsão acústicos e nas comunicações submarinas, e a área de Meteorologia, que também usa as informações geradas pelo SPNO em seus modelos de previsão e em análises, de modo a facilitar a compreensão de fenômenos climáticos.

### III. TRABALHOS CORRELATOS

No contexto da aplicação de Web Services a Sistemas de Previsão Numérica Oceanográfica para gerenciar a comunicação e a interação entre módulos, não foram encontrados trabalhos com esse propósito, no entanto, Ferreira [6] em sua dissertação “Arquitetura de Middleware para Internet das Coisas” constatou ser capaz de controlar e monitorar o estado de dispositivos, trazendo a devida interoperabilidade, escalabilidade e mobilidade para as partes envolvidas, utilizando as tecnologias UPnP (Universal Plug and Play), REST e ZigBee (IEEE 802.15.4) sobre redes IP.

Já o trabalho elaborado por ONO et al. [7] apresentou para a aplicação Cytoscape, que permite análise, visualização e publicação de redes complexas, já disponível em vários tipos de domínios, incluindo bioinformática, análise de redes sociais e web semântica o aplicativo CyREST.

O CyREST é um módulo RESTful capaz de permitir a comunicação com aplicativos em diversas linguagens, como Python, R, Matlab ou Java Script. Nas solicitações e respostas

os dados são transmitidos no formato JavaScript Object Notation (JSON).

No modelo apresentado por ONO et al. [7], a camada Cytoscape representa ferramentas que realizam as solicitações e recebem os resultados do cyREST, que se comunica com bibliotecas desenvolvidas para intermediar a passagem de parâmetros da camada superior para os sistemas locais disponíveis.

Poulter, Johnston e Cox [8] examinam o uso da pilha de ferramentas de desenvolvimento MEAN (MongoDB, Express, Angular e Node.js) no desenvolvimento de serviços backend e interfaces Web, para dispositivos da Internet das Coisas (IoT, do inglês *Internet of Things*).

Poulter, Johnston e Cox [8] implementaram uma API RESTful - permitindo quatro operações (POST, GET, PUT & DELETE) em todos os dados e, assim como Ferreira [6], apresentou o elemento Controlador Escravo, também utilizaram um micro controlador dedicado para atuar como uma interface para o hardware de nível inferior, e dessa forma modular proporcionaram maior flexibilidade para uma futura substituição do dispositivo.

Poulter, Johnston e Cox [8] também propuseram não somente o envio de dados através do sensor para a repositório central, mas interações inversas, permitindo mudança de parâmetros, como frequência de leitura do sensor e até mesmo o envio de uma atualização de firmware para o dispositivo.

Do ponto de vista da dificuldade do SPNO para com a integração de seus módulos, a tecnologia de Web Services se apresenta como uma solução na qual os sistemas podem interoperar, ou seja, é possível um sistema ou produto de interagir com outro sistema ou produto sem uma intervenção específica por parte do usuário [9].

## IV. REFERENCIAL TEÓRICO

### A. Arquitetura Orientada a Serviços e Web Services

Conforme o Modelo de Referência SOA, publicado pela Organização para o Avanço de Padrões de Informação Estruturada (OASIS) [10], SOA é definida como um paradigma (conceito) para organização e utilização de recursos distribuídos que estão sob controle de diferentes domínios proprietários. Provê uma maneira uniforme de oferecer, descobrir, interagir e utilizar capacidades para produzir efeitos desejados consistentes com pré-condições e expectativas mensuráveis [10].

SOA é uma estrutura independente de tecnologia para definir, registrar e invocar serviços [11]. Um serviço é um mecanismo para habilitar o acesso a um ou mais recursos, onde o acesso é fornecido por meio de uma interface (registro de serviços) descrita com restrições e políticas especificadas pela descrição do serviço, sendo oferecido por uma entidade (provedor de serviço) para uso de terceiros (clientes de serviço) [10].

SOA é um conceito arquitetural, uma forma de pensar e projetar integração entre serviços de negócio. A integração entre vários serviços (cada serviço pode ser uma parte de um

sistema) é que forma um processo de negócio. Em SOA, estes serviços são plugados a um barramento central chamado de ESB (Enterprise Service Bus). Esse barramento possui uma serie de conectores que permitem a comunicação com diferentes tecnologias, desde SOAP (WebService), CORBA, RMI a tecnologias legadas. A plataforma de tecnologia mais utilizada neste modelo é a de Web Services [12], [13].

Web Services consistem em um serviço que se comunica com os clientes através de um conjunto de protocolos e tecnologias padronizadas. Esses protocolos e tecnologias são implementados em plataformas e produtos, permitindo que clientes e serviços se comuniquem de forma consistente em diferentes ambientes operacionais [13].

Segundo W3C [14], a definição de Web Services é descrita como sendo um sistema de software desenvolvido para permitir interações máquina-máquina através de uma rede. Fornece uma interface de descrição de serviços onde outros sistemas interagem com o Web Services através de mensagens SOAP, geralmente enviadas através de HTTP em conjunto com outros padrões relacionados à web. No entanto, esta definição não é restritiva e não pressupõe o uso de SOAP como o único formato de modelo de processamento.

O protocolo SOAP (Simple Object Access Protocol - Protocolo Simples de Acesso a Objetos) surgiu com as primeiras definições para Web Service, porém o estilo arquitetural REST (REpresentational State Transfer - Transferência de Estado Representacional) vem ganhando cada vez mais popularidade entre os desenvolvedores de Web Services. Empresas como Google, Facebook, Twitter e Amazon já disponibilizam Web Services de estilo REST para acesso a seus serviços [15].

A escolha da utilização do REST neste trabalho foi feita pelos motivos acima, além do fato do ERDDAP<sup>1</sup>, sistema integrado a mais de 50 instituições ambientais, também disponibilizar dados utilizados nesta pesquisa através de Web Services RESTful, implementação que segue os princípios da meta-arquitetura REST.

### B. Rest - Representational State Transfer

O estilo arquitetônico REST (*Representational State Transfer*), foi desenvolvido como um modelo abstrato da arquitetura Web baseado em HTTP e URI (Identificador Uniforme de Recurso - *Uniform Resource Identifier*), sendo apresentado não como uma arquitetura, mas como uma forma de avaliar arquiteturas. Esse estilo foi proposto por Roy T. Fielding em sua dissertação de doutorado na Universidade da Califórnia, com o tema "*Architectural Styles and the Design of Network-based Software Architectures*", publicada no ano de 2000. Fielding também é o principal autor do HTTP / 1.1 e coautor dos padrões da Internet HTTP e URI.

Fielding e Reschke [16] definiram HTTP como um

<sup>1</sup> O ERDDAP (Programa de Acesso de Dados da Divisão de Pesquisa Ambiental) é um servidor de dados que oferece uma maneira simples e consistente de baixar subconjuntos de conjuntos de dados científicos em formatos comuns de arquivos e criar gráficos e mapas. Esta instalação particular do ERDDAP possui dados oceanográficos (por exemplo, dados de satélites e bóias).

protocolo de nível de aplicação sem estado para sistemas de informação distribuídos, colaborativos e de hipertexto. Nesse protocolo a formatação e negociação da representação de dados permite que os sistemas sejam construídos independentemente dos dados que são transferidos.

De acordo com Fielding e Reschke [17] os métodos do protocolo HTTP devem ter a mesma semântica quando aplicados a qualquer recurso, embora cada recurso determine por si mesmo se essas semânticas são implementadas ou permitidas. Uma série de métodos padronizados são comumente usados em HTTP, conforme descrito na Tabela I.

O *Internet Engineering Task Force* (IETF), principal órgão de padrões da Internet, através de *Request for Comments* (RFC), descreve os padrões de cada protocolo da Internet. Além dos oito métodos descritos na RFC 7230 [16] o método PATCH também foi padronizado conforme a RFC 5261 [18], que descreve um patch formato para documentos XML, e na RFC 6902 [19], que descreve um formato semelhante para documentos JSON [20].

TABELA I  
MÉTODOS DO PROTOCOLO HTTP

Método	Descrição
GET	Obter uma representação do recurso.
HEAD	Obter os cabeçalhos enviados junto com uma representação do recurso.
POST	Criar um novo recurso com base na representação fornecida.
PUT	Substituir o estado de um recurso pelo descrito na representação fornecida.
PATCH	Modificar apenas algumas informações específicas do recurso.
DELETE	Apagar o recurso.
CONNECT	Encaminhar algum outro protocolo através de um proxy HTTP.
OPTIONS	Descobrir os métodos HTTP aos quais o recurso responde.
TRACE	Depurar proxies.

Segundo Fielding [21] e Richardson, Amundsen e Ruby [20], o URI é o elemento mais simples e mais importante da arquitetura da Web. Havia sistemas de hipertexto antes do HTML e protocolos de Internet antes do HTTP, mas eles não se comunicavam entre si. O URI interligou todos esses protocolos de Internet em uma Web.

A arquitetura inicial da Web definiu o URI como identificadores de documentos. Os autores foram instruídos a definir identificadores em termos da localização de um documento na rede. Os protocolos da Web poderiam então ser usados para recuperar esse documento. No entanto, esta definição revelou-se insatisfatória devido, por exemplo, à necessidade de mudar o identificador sempre que o conteúdo sofresse alteração, ou até mesmo, quando a URI correspondesse a um serviço ao invés de um documento [21].

Com o REST é possível que a referência permaneça estática, embora o resultado de acessar essa referência possa

mudar com o tempo, definindo um recurso como a semântica do que o autor pretende identificar, ao invés do valor correspondente à semântica no momento em que a referência é criada. É então deixado ao autor para assegurar que o identificador escolhido para uma referência realmente identifique a semântica pretendida [21].

O URI fornece um meio simples e extensível para identificar um recurso. Esta especificação da sintaxe URI e semântica é derivada de conceitos introduzidos pela *World Wide Web* [22].

Nas recomendações REST e nas recomendações do W3C, um recurso e seu URI devem ter uma correspondência intuitiva, mantendo uma estrutura e devendo variar de maneira previsível. Cada URI designa exatamente um recurso, no entanto um recurso pode possuir URIs diferentes. Por exemplo: Se a versão atual do software for 1.3, então <http://www.example.com/releases/1.3.tar> e <http://www.example.com/releases/latest.tar>, são duas URIs que se referirão ao mesmo arquivo por um período de tempo [20].

O REST fornece um conjunto de restrições arquitetônicas que, quando aplicadas como um todo, enfatizam a escalabilidade de interações de componentes, a generalidade de interfaces, a implantação independente de componentes e componentes intermediários, para reduzir a latência de interação, reforçar a segurança e encapsular sistemas legados. [23].

A arquitetura RESTful é uma implementação do conceito da meta-arquitetura REST: Um conjunto de diretrizes simples para a implementação de serviços típicos que atendam ao potencial da Web [20].

## V. PROCESSOS DO SISTEMA DE PREVISÃO NUMÉRICA OCEANOGRÁFICA - SPNO

Neste trabalho a coleta de dados é feita na base ERDDAP, que é uma base de dados científicos mantido pela NOAA<sup>2</sup> (*National Oceanic and Atmospheric Administration*).

Próxima etapa, de geração do campo inicial, através de técnicas de assimilação de dados, faz com que as imagens de satélite criem campos tridimensionais utilizando os dados orbitais e climatológicos. Dessa forma, são produzidos, por exemplo, campos tridimensionais de temperatura e salinidade.

Os processos executados com Matlab ou Python são para gerar o campo inicial, que ocorre após os dados orbitais e climatológicos estarem disponíveis.

Na Figura 1 é apresentado a representação gráfica do dado orbital extraído em 24 de novembro de 2017, com as coordenadas de Latitude (-40° e -15°) e Longitude (-50° e -30°), através do sistema ERDDAP. O dado é um produto de análise de Temperatura de Superfície do mar (SST, do inglês *Sea Surface Temperature*) produzido por Jet Propulsion Laboratory (JPL) no âmbito do programa NASA MeaSURES.

<sup>2</sup> A NOAA é uma organização que pertence ao Departamento de Comércio dos Estados Unidos que é apoiada pelo Serviço Nacional de Meteorologia dos Estados Unidos. Sua missão é “Compreender e prever as mudanças climáticas, oceânicas e costeiras, para compartilhar esse conhecimento e informação com outras pessoas e para conservar e gerenciar os ecossistemas e recursos costeiros e marinhos.” [24].

O dado orbital foi extraído da base *Multi-scale Ultra-high Resolution (MUR) SST Analysis fv04.1, global, 0.01°, 2002-present, Daily*. A barra com os valores de 0 a 32 indicam a variação de temperatura em °C.

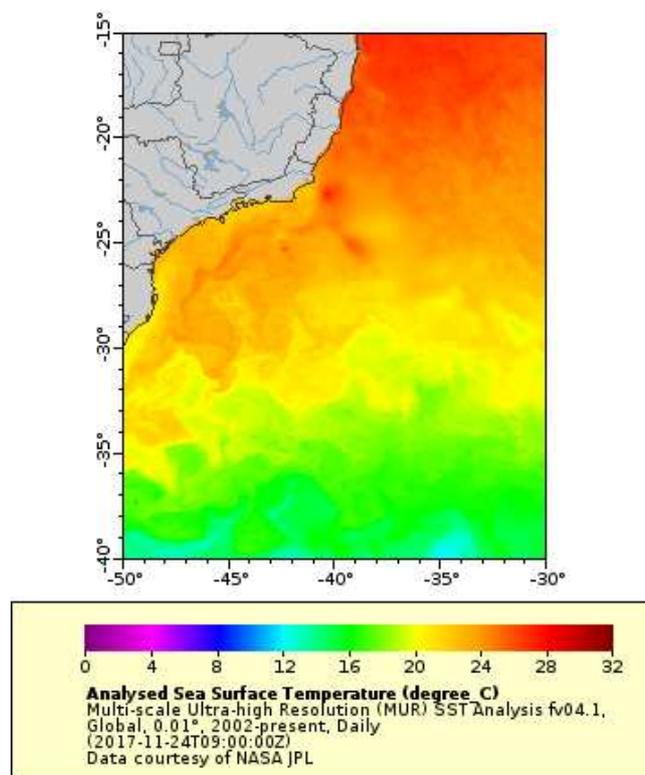


Fig. 1. Representação gráfica do dado orbital.

No bloco de processos do servidor Fortran estão as etapas de processamento do Sistema de Previsão, isto é, onde o modelo numérico *Regional Ocean Modeling System* (ROMS) será executado em três etapas, conforme apresentado na Figura 4. Cada etapa para execução do modelo numérico ROMS atua de forma específica.

O primeiro módulo operacional é responsável por realizar uma suavização dos campos tridimensionais a partir de campos iniciais. Este módulo inicializa o processo e o seu resultado é transferido através de um novo campo inicial para o próximo módulo operacional.

O segundo módulo operacional permite que as forças de temperatura e salinidade permaneçam invariantes no tempo enquanto a velocidade evolui. É referenciado como *Nowcasting* (previsão do tempo presente), que é o resultado do processo de ajuste de um campo de velocidade aos campos de temperatura e salinidade, conforme apresentado na Figura 2.

O último módulo utiliza o campo *Nowcasting* e evolui no tempo com as velocidades, temperaturas e salinidade interagindo entre elas. Dessa forma será gerado o resultado da previsão, com confiabilidade para três dias, conforme apresentado na Figura 3.

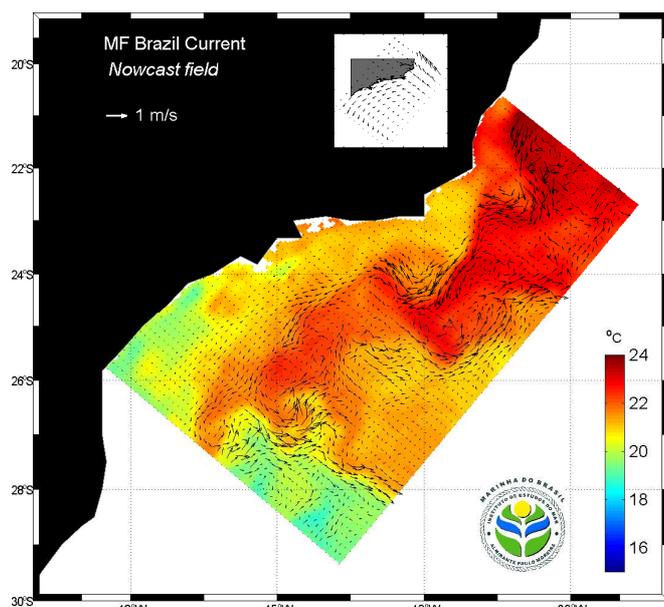


Fig. 2. Nowcast: resultado do tempo presente.

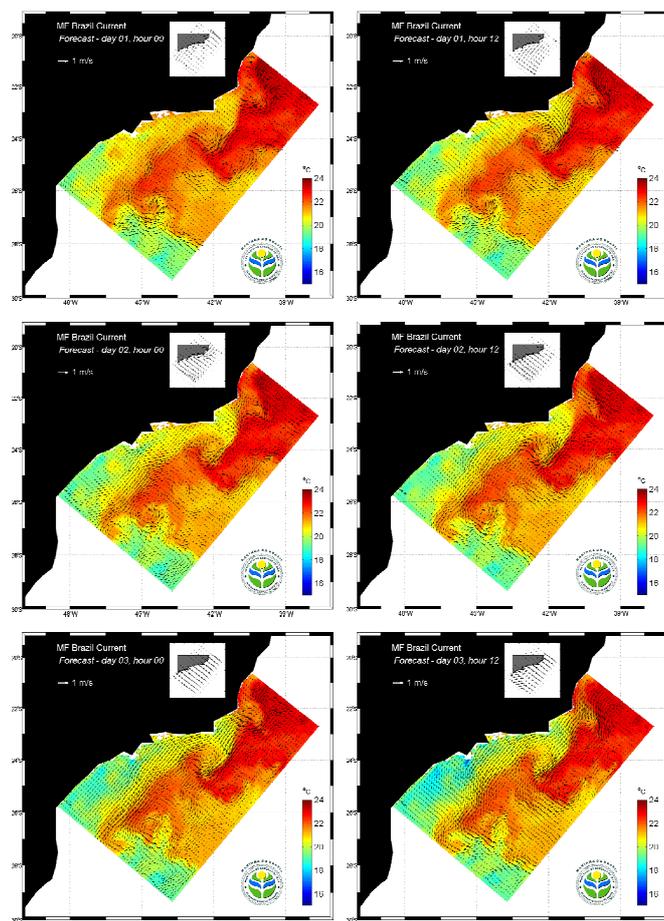


Fig. 3. Resultado de previsão para três dias com intervalo de 12 horas.

## VI. DESENHO EXPERIMENTAL

Os sistemas especialistas utilizados no processo de Previsão Numérica Oceanográfica são atualizados em relação aos métodos utilizados internacionalmente para produção de

previsões, mas também permitem configurações específicas dos especialistas em relação ao conhecimento de técnicas e da região onde os estudos são realizados.

Há necessidade de flexibilidade e independência, para ajustes individuais como período de tempo, tamanho de grade e número de ciclos de processamento para melhorar os resultados, ou até mesmo para a substituição de um novo módulo.

Este trabalho visa manter simplicidade e eficiência da aplicação de Web Services na problemática apresentada, desta forma utiliza em cada um dos módulos a implementação do Restify<sup>3</sup>, que é uma biblioteca desenvolvida em JavaScript para o Node.js que permite a construção de Web Services baseado na arquitetura RESTful.

Mantendo a independência dos sistemas, os arquivos processados por cada sistema são armazenados numa estrutura de diretórios, para que possam ser acessados localmente e remotamente, permitindo a continuidade de todo o processo.

Os arquivos manipulados pelo sistema são basicamente no formato NetCDF<sup>4</sup>, padrão utilizado para acesso e compartilhamento de dados científicos orientados para matrizes.

Em razão do tamanho dos arquivos e disponibilidade de rede, estes arquivos passam por um processo de compactação. Como exemplo, foi utilizado o arquivo resultante na terceira etapa do processamento do modelo numérico que contém as previsões no formato NetCDF. As Figuras 2 e 3 são representações gráficas de resultados gerados a partir desse arquivo. Na avaliação de compactação, o arquivo de resultado passou para 54% do tamanho original, passando de 810 MB para 439 MB.

A estrutura para processamento dos dados não restringe ao uso dentro da rede local, podendo utilizar hardware de maior capacidade de processamento em outras instituições, assim o processo de compactação é importante para uma maior eficiência na transferência dos arquivos e para que os resultados das previsões estejam disponíveis em intervalo de tempo aceitável.

A Tabela II apresenta a estrutura de diretórios dos Web Services e indica um fluxo de processos. Os diretórios que iniciam com “Sis” representam os sistemas especialistas e os diretórios que iniciam com “Dir” representam os diretórios com arquivos gerados pelos sistemas. Esses diretórios ficam disponíveis via Web Service, através de requisições GET e GET: ID, nas quais são executados os módulos fs.readdir, para ler o conteúdo do diretório e fs.readFile para ler o conteúdo de um arquivo.

<sup>3</sup> Grandes empresas como Netflix, Napster, Pinterest, NPM e Joyent utilizam o módulo Restify.

<sup>4</sup> Network Common Data Form (NetCDF) é um conjunto de bibliotecas de software e formatos de dados auto-descreventes, independentes da máquina que suportam a criação, acesso e compartilhamento de dados científicos orientados para matrizes.

TABELA III  
ESTRUTURA DE DIRETÓRIOS DOS WEB SERVICES

WebService1	WebService2	WebService3
1	SisDownload	
2	DirDownload → DirDownload	
3	SisCampoInicial	
4	DirCampoInicial → DirCampoInicial	
5	SisROMS	
6	DirROMS ← DirROMS	
7	SisPrevisaoIMG	
8	DirPrevisaoIMG ← DirPrevisaoIMG	

Na Figura 4 é apresentada uma representação gráfica dos fluxos dos processos. No WebService1 sua estrutura armazena arquivos coletados de sistemas externos e disponibiliza os resultados do processo de previsão a requisições externas.

Nos demais WebServices os dados são transferidos e processados através de suas rotinas, como a geração do Campo Inicial, que disponibiliza seu resultado para o WebService3, e a geração das imagens, que disponibiliza seu resultado para o WebService1. Já no WebService3, após o processamento das três etapas do modelo ROMS o resultado é disponibilizado para o WebService1 e WebService3.

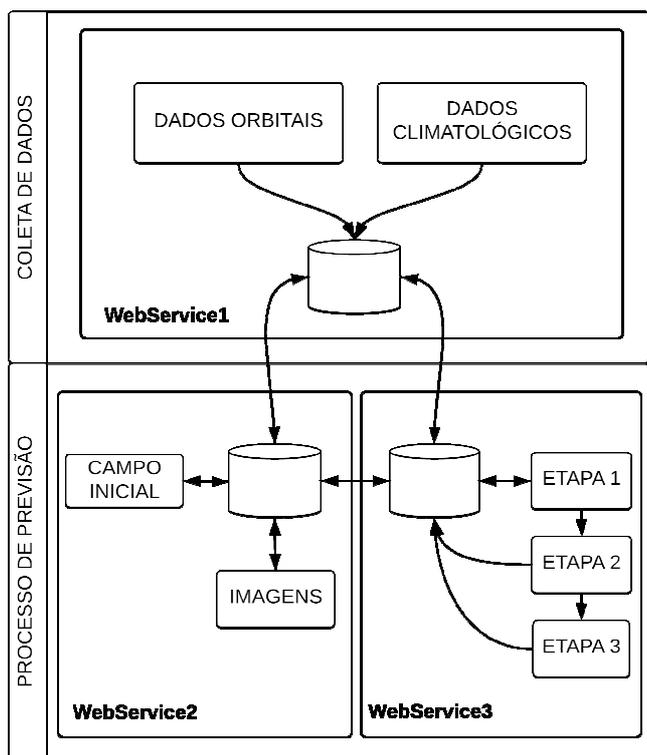


Fig. 4. Componentes de coleta de dados e dos processos de previsão.

Cada servidor Node.js, além da implementação de rotas para os serviços, utiliza uma função CRON<sup>5</sup> para automatizar

<sup>5</sup> Através da função CRON é possível especificar o intervalo de tempo que sua função de verificação de arquivos remotos e seus processos internos devem ser executados.

suas tarefas e o módulo child.process, através da função child.process.execSync()<sup>6</sup>, para gerar um shell e executar um comando para execução de processos. Abaixo seguem funções específicas de cada servidor:

- A rotina no WebService1 é responsável em fazer uma requisição via GET ao servidor especificado (ERDDAP) e, através da função child.process.execSync(), compactar o arquivo. Também realiza, através do método GET, a consulta e busca dos resultados do WebService2.
- A rotina no WebService2 possui duas etapas que são executadas em tempos diferentes, a primeira é a responsável em buscar os arquivos do WebService1, descompactar, transferir para a pasta SisCampoInicial com o nome pré-definido, gerar o Campo Inicial e compactar o resultado no diretório DirCampoInicial; e a segunda etapa é realizar o download do WebService3, descompactar, transferir para a pasta SisPrevisãoIMG com o nome pré-definido, gerar as imagens de previsão e compactar o resultado no diretório DirPrevisãoIMG.
- A rotina no WebService3 busca o arquivo do CampoInicial no WebService2, descompacta, copia com o nome pré-definido para a pasta SisROMS, executa as etapas do sistema ROMS e, após o resultado ser gerado, o arquivo é compactado e disponibilizado na pasta DirROMS.

As funções descritas também são apresentadas através do diagrama de atividades na Figura 5.

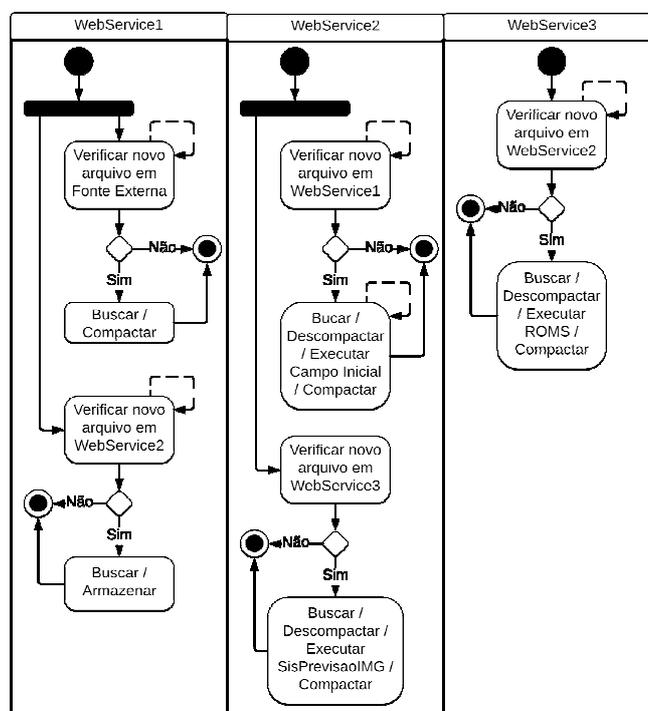


Fig. 5. Diagrama de Atividades

<sup>6</sup> Child.process.execSync() é uma versão síncrona do child\_process.exec() que irá bloquear o loop de eventos Node.js.

O sistema de Gerência permite monitorar e controlar os módulos, passando parâmetros, iniciando processos e analisando a execução das etapas.

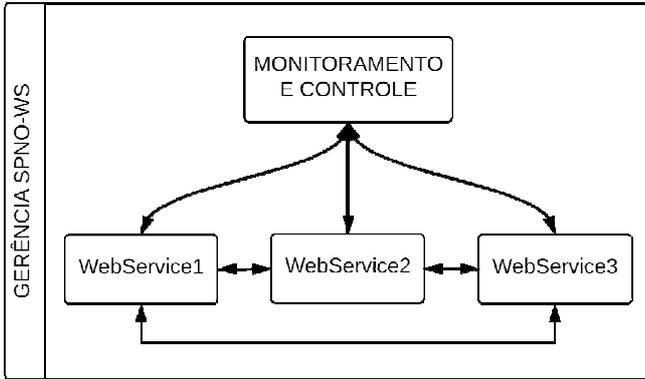


Fig. 6. Comunicação entre os módulos do SPNO-WS.

Conforme demonstrado na Figura 6, todos os módulos de se comunicam. Esta comunicação é necessária para que a transferência de arquivos nas etapas de processamento não crie barreiras e torne mais eficaz a solução proposta. A utilização da arquitetura REST com arquivos no formato JSON, além permitir o envio e recebimento sem barreiras de firewalls, por trabalhar diretamente no protocolo HTTP, busca trabalhar com seus recursos identificados por URI, permitindo passagem de parâmetros aos demais servidores para ajustes dos seus módulos locais.

Nas Figuras 7 e 8 estão detalhadas requisições e respostas de dois processos que utilizam REST como meio para transferência e passagem de parâmetros.

Conforme apresentado na Figura 7, o cliente realiza uma consulta aos arquivos disponíveis através do método GET ao Webservice, que executa uma rotina interna capaz de listar arquivos disponíveis e retornar uma resposta no formato JSON. A partir dessa resposta o cliente identifica através de uma rotina interna se não possui algum arquivo e executa novamente o método GET informando a identificação do arquivo a ser copiado.

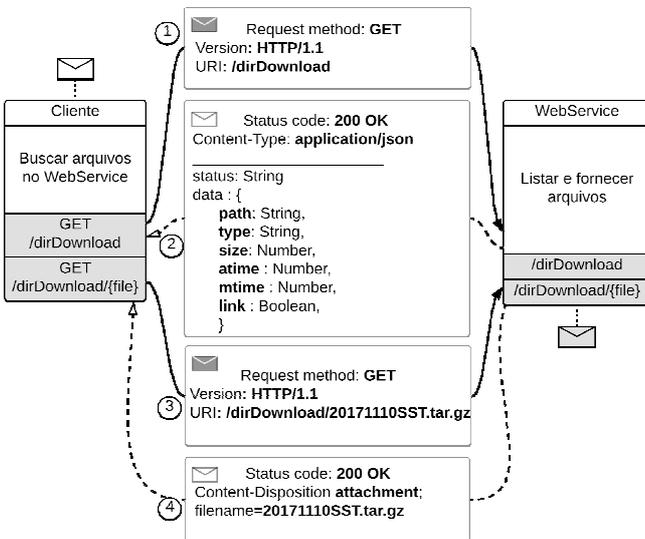


Fig. 7. Exemplo de uma comunicação REST nas etapas de transferência de arquivos.

Conforme apresentado na Figura 8, o cliente realiza uma consulta ao arquivo de configuração, através do método GET, ao Webservice, que executa uma rotina interna capaz de ler o arquivo e retornar o parâmetro atual através de uma resposta no formato JSON. A partir dessa resposta o cliente executa o método PUT passando o valor lido anteriormente e o novo parâmetro.

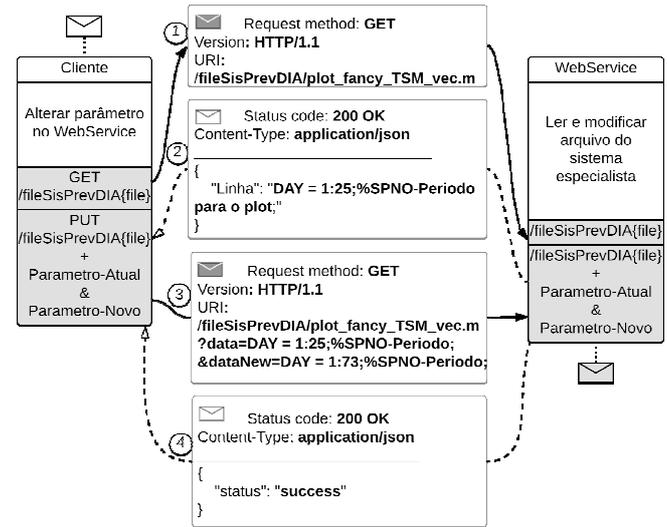


Fig. 8. Exemplo de uma comunicação REST nas etapas de passagem de parâmetros para os sistemas especialistas.

## VII. DISCUSSÃO DAS ETAPAS DE PROCESSAMENTO

A tecnologia JavaScript com o NODE.js, traz simplicidade e a possibilidade de interação com o Sistema Operacional de forma fácil e controlada, algo que não era possível com outras linguagens. Dessa forma o acesso a sistemas especializados ou até mesmo sistemas antigos que não suportam tecnologias recentes, se torna possível, permitindo a utilização de tecnologias mais novas.

A arquitetura RESTful traz clareza e facilidade na implantação dos conceitos REST, permitindo implementação de diversas formas, desde operações de simples busca de arquivos até manipulação de arquivos, com leitura, exclusão e mudança remota de parâmetros nos sistemas especialistas. Nesse sentido, as diversas bibliotecas para Node.js mencionadas neste trabalho vêm a contribuir de forma significativa na agilidade de desenvolvimento, devido à grande contribuição de desenvolvedores de toda Web.

A adaptação de novos sistemas necessita de um tempo de ajuste, no entanto a forma modular permite, além da agilidade na adequação de novos sistemas e parâmetros, uma visão clara e objetiva do sistema. Permite que os próprios especialistas estejam mais próximos para propor evoluções no sistema, de forma a contribuir para os resultados de previsão. Por exemplo, é possível desenvolver comparativos de resultados de previsão, identificando por sistemas intermediários a equivalência da previsão do dia atual em relação à projeção realizada anteriormente. Nesse caso, a elaboração do sistema é

realizada por pesquisadores de forma totalmente independente, para que no futuro possa ser adicionada como um novo módulo.

Como trabalhos futuros pretende-se implementar ajustes no modelo em funcionamento, permitindo que parâmetros nas rotinas legadas existentes sejam ajustados de forma automática. Como por exemplo, ajuste da janela de tempo da previsão, que hoje é feita ainda de forma manual. Além de outros aperfeiçoamentos na pesquisa.

Deve-se destacar que, a pesquisa é baseada nos modelos e procedimentos internos de previsão numérica no IEAPM e que atendem às suas necessidades específicas. Por tanto, para uso em outros institutos de pesquisa é necessário um estudo aprofundado de todo o procedimento, rotinas, estratégias de uso, entre outros fatores para que se possa adaptar o sistema desenvolvido.

#### REFERENCIAS

- [1] SOUZA, Ronald Buss de. Oceanografia por satélites. 2. ed. Oficina de Textos, 2009.
- [2] RICHARDSON, Lewis Fry. Weather prediction by numerical process Cambridge University Press. Monthly Weather Review, n. February, p. 219, 1922. Disponível em: <https://docs.lib.noaa.gov/rescue/mwr/050/mwr-050-02-0072.pdf>. Acesso em: 25 mar. 2017.
- [3] KALYVAS, Christos; KOKKOS, Athanasios; TZOURAMANIS, Theodoros. A Survey of Official Online Sources of High-quality Free-of-charge Geospatial Data for Maritime Geographic Information Systems Applications. Information Systems, v. 65, n. September 2016, p. 36–51, 2016.
- [4] FLEMMING, Nicholas Coit. Strategic Planning for Operational Oceanography. Ocean Forecasting. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 1–17. Disponível em: <http://link.springer.com/10.1007/978-3-662-22648-3\_1>. Acesso em: 24 mar. 2017.
- [5] BAPTISTA, António M. Modern Paradigms for Modeling and Visualisation of Environmental Systems. Encyclopedia of Physical Science & Technology, v. 5, p. 565–581, 2002.
- [6] FERREIRA, Hiro Gabriel Cerqueira. Arquitetura de middleware para internet das coisas. 2014. 111 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade de Brasília, Brasília, 2014. Disponível em: <http://repositorio.unb.br/handle/10482/17251>. Acesso em: 20 maio 2017.
- [7] ONO, Keiichiro et al. CyREST: Turbocharging Cytoscape Access for External Tools via a RESTful API. F1000Research, 2015. Disponível em: <http://f1000research.com/articles/4-478/v1>. Acesso em: 1 jun. 2017.
- [8] POULTER, Andrew John; JOHNSTON, Steven J.; COX, Simon J. Using the MEAN stack to implement a RESTful service for an Internet of Things application. IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings, p. 280–285, 2016.
- [9] IEEE. Standards Glossary. Disponível em: <https://www.ieee.org/education\_careers/education/standards/standards\_glossary.html>. Acesso em: 24 mar. 2017.
- [10] MCCABE, Francis et al. Reference Model for Service Oriented 1.0. p. 31, 2006. Disponível em: <https://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>. Acesso em: 20 abr. 2017.
- [11] PAPAOGLOU, Mike P.; VAN DEN HEUVEL, Willem Jan. Service oriented architectures: Approaches, technologies and research issues. VLDB Journal, v. 16, n. 3, p. 389–415, 2007. Disponível em: <https://link.springer.com/article/10.1007/s00778-007-0044-3>. Acesso em: 3 maio 2017.
- [12] ERL, THOMAS. SOA Principios de Design de Serviços. São Paulo: Pearson Prentice Hall, 2009.
- [13] ORT, Ed. Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools. 2005. Disponível em: <http://www.oracle.com/technetwork/articles/javase/index-139840.html>. Acesso em: 23 abr. 2017.
- [14] W3C, World Wide Web Consortium. Web Services Architecture, 2004. Disponível em: < https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>. Acesso em: 25 mar. 2017.
- [15] DA SILVA, Ricardo Frenedoso; GONÇALVES, Pablo Rodrigo. Web Services – Uma Análise Comparativa. Revista das Faculdades Integradas Claretianas–No5–janeiro/dezembro de, p. 8, 2012.
- [16] FIELDING, Roy Thomas; RESCHKE, J. RFC 7230 - Hypertext Transfer Protocol HTTP/1.1: Message Syntax and Routing. 2014a. Disponível em: <https://tools.ietf.org/pdf/rfc7230.pdf>. Acesso em: 24 maio 2017.
- [17] FIELDING, Roy Thomas; RESCHKE, J. RFC 7231 - Hypertext Transfer Protocol HTTP/1.1: Semantics and Content. 2014b. Disponível em: <http://www.rfc-editor.org/info/rfc7231.>. Acesso em: 25 maio 2017.
- [18] URPALAINEN, J. RFC 5261 - An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors. RFC, 2008. Disponível em: <https://tools.ietf.org/pdf/rfc5261.pdf>. Acesso em: 11 jun. 2017.
- [19] BRYAN, P.; NOTTINGHAM, M. RFC 6902 - JavaScript Object Notation (JSON) Patch. 2013. Disponível em: <http://www.rfc-editor.org/info/rfc6902.>. Acesso em: 11 jun. 2017.
- [20] RICHARDSON, Leonard; AMUNDSEN, Mike; RUBY, Sam. RESTful Web APIs: Services for a Changing World. 5. ed. Sebastopol, CA: 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2013.
- [21] FIELDING, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. 2000. 162 f. Doctor of Philosophy in Information and Computer Science, University of California, IRVINE, 2000. Disponível em: <http://www.ics.uci.edu/~fielding/pubs/dissertation>. Acesso em: 24 mar. 2017.
- [22] BERNERS-LEE, T; FIELDING, R; MASINTER, L. RFC 3986 - Uniform Resource Identifier - URI: Generic Syntax. 2005. Disponível em: <https://tools.ietf.org/pdf/rfc3986.pdf>. Acesso em: 24 maio 2017.
- [23] FIELDING, Roy Thomas; TAYLOR, Richard N. Principled design of the modern Web architecture. ACM Transactions on Internet Technology, v. 2, n. 2, p. 115–150, 2002. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/webarch\_icse2000.pdf>. Acesso em: 24 maio 2017.
- [24] NOAA, Administração Nacional Oceânica e Atmosférica. Sobre. Disponível em: <http://www.noaa.gov/about-our-agency>. Acesso em: 26 mar. 2017.