MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Conselho Nacional de Desenvolvimento Científico e Tecnológico

DESENVOLVIMENTO DE UMA CAMADA DE ACESSO AOS DADOS AMBIENTAIS E IMAGENS DE SATÉLITE WEBSERVICE AMBIENTAL

Ruan Fagundes Rita (FATEC Cruzeiro, Bolsista PIBITI/CNPq)
E-mail: ruan.rita@inpe.br

Daniel Alejandro Vila (DSA/CPTEC/INPE, Orientador)
E-mail: daniel.vila@cptec.inpe.br

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO EM DESENVOLVIMENTO TECNOLOGICO
(PIBITI/CNPq/INPE)

CO-ORIENTADOR

MÁRIO LEMES DE FIGUEIREDO NETO (DSA/CPTEC/INPE)

Julho de 2019

RELATÓRIO PARCIAL DE INICIAÇÃO EM DESENVOLVIMENTO TECNOLÓGICO DO PROGRAMA:
PIBITI/INPE - CNPq

PROJETO

DESENVOLVIMENTO DE UMA CAMADA DE ACESSO AOS DADOS AMBIENTAIS E IMAGENS DE SATÉLITE WEBSERVICE AMBIENTAL

PROCESSO: 105405/2019-1

Relatório elaborado por Ruan Fagundes Rita relativo ao período de
Março de 2019 a Agosto de 2019

Ruan Fagundes – Bolsista PIBITI/CNPq
E-mail: ruan.rita@inpe.br

Daniel Alejandro Vila– Orientador
DSA/CPTEC/INPE
E-mail: daniel.vila@inpe.br

Julho de 2019

AGRADECIMENTOS

Primeiramente devo agradecer a minha família, que me ensinou a ser uma pessoa educada, talentosa, dedicada e na maioria das vezes estava lá me orientando, me cuidando para continuar sempre estudando. Nos dias em que fiquei desanimado sempre me deram forças para continuar lutando, para nunca desistir e continuar sempre otimista.

Aos meus amigos, fazendo do meu dia sempre mais feliz, resultando nos dias de trabalhos eficientes. Me distraindo dos problemas cotidianos conseguindo ter bastante disposição e foco.

Agradeço muito ao meu orientador e colaborador que destrincharão as minhas dúvidas ocasionando no bom desenvolvimento do projeto e sempre deram conselhos bons.

Por fim, agradeço a todos as pessoas que fizeram ou ainda faz parte da minha vida, pois elas me influenciam no meu jeito de viver, de ser e de como eu reajo com outras pessoas, fui coletando os costumes bons de cada um tornando-se uma pessoa melhor e grato por todos.

LISTA DE FIGURAS

Figura 1: Estrutura do Projeto

Figura 2: Interface PostgreSQL

Figura 3: Interface Eclipse – pom.xml

Figura 4: Interface Eclipse – web.xml

Figura 5: Interface Eclipse – classe Conexao

Figura 6: Interface Eclipse – classe ImagemDAO

Figura 7: Interface Eclipse – classe sensor

Figura 8: Interface Web Service – Login

Figura 9: Interface Web Service – Exibindo os dados

LISTA DE ABREVIATURAS E SIGLAS

CSS	–	Cascading Style Sheets
DAO	–	Data Access Object
DSA	–	Divisão de Satélites e Sistemas Ambientais
EAI	–	Enterprise Application Integration
HTML	–	Hypertext Markup Language
JSON	–	JavaScript Object Notation
MVC	–	Model, View e Controller
REST	–	Representational State Transfer
SGBD	–	Sistema Gerenciador de Banco de Dados
URL	–	Uniform Resource Locator
URI	–	Uniform Resource Identifier
XML	–	Extensible Markup Language

RESUMO

Neste trabalho, abordamos o desenvolvimento de uma camada padronizada de acesso aos dados ambientais e imagens de satélites, utilizando a tecnologia de Web Service, que tem como objetivo a integração de sistemas entre diferentes aplicações e padrões. Sua estrutura é composta por diversos bancos de dados ambientais, que possuem diferentes estruturas como Imagens de Satélite, Plataformas de Coleta de Dados entre outras aplicações. Para o desenvolvimento da aplicação utilizamos o conceito de MVC para estruturação da aplicação, que está sendo desenvolvida em Java e ainda utilizamos os frameworks VRAPTOR e JERSEY (JAX/RS). O projeto permitirá ao CPTEC/DSA oferecer essa camada de serviço de dados ambientais, permitindo a integração de diferentes aplicações de forma padronizada.

Palavras-chave: Web Service. MVC. REST

ABSTRACT

In this work, we discuss the development of a standardized layer of access to environmental data and satellite images, using Web Service technology, which aims to integrate systems between different applications and standards. Its structure is composed of several environmental databases, which have different structures such as Satellite Images, Data Collection Platforms and other applications. For the development of the application we use the MVC concept for structuring the application, which is being developed in Java and we still use the VRAPTOR and JERSEY (JAX / RS) frameworks. The project will allow CPTEC/DSA to offer this layer of environmental data service, allowing the integration of different applications in a standardized way.

Word-key: Web Sevice. MC. REST

SUMÁRIO

1. INTRODUÇÃO.....9

2. OBJETIVO.....10

2.1 Geral.....10

2.2 Específico.....10

3. FUNDAMENTAÇÃO TEÓRICA.....11

4. METODOLOGIA.....12

4.1 Ferramentas.....12

4.1.1 Eclipse ide.....12

4.1.2 MAVEN.....13

4.1.3 Tomcat 8.5.....13

4.1.4 Jax-rs.....14

4.1.5 Vraptor 4.....14

4.1.6 PostgreSQL.....14

4.2 Linguagens.....14

4.2.1 JAVA.....15

4.2.2 XML.....15

4.2.3 JSON.....15

4.2.4 HTML.....16

4.3 Estruturação.....16

5. ANÁLISE E RESULTADOS.....18

6. CONSIDERAÇÕES FINAIS.....22

7. REFERÊNCIAS.....23

1. INTRODUÇÃO

O projeto tem o tema de desenvolver uma camada de acesso aos dados ambientais e imagens de satélite proporcionando uma integração com vários tipos de sistema e ajudando a DSA nos seus objetivos. Para isso, é necessário a utilização de uma tecnologia que visa esse fácil acesso ao banco de dados e disponibilização das imagens.

Vem se utilização cada vez mais os Web Service pois é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

Com isto, podemos disponibilizar os produtos de forma padronizada e o meio na qual as imagens serão exibidas, através das classes em Java no formato JSON para que outras aplicações possam trabalhar com formatos padronizados, obtendo uma integração de sistemas.

No presente trabalho, é composto por diversos capítulos sendo bem-conceituados. No segundo Capítulo aborda todos objetivos do projeto destacando-se o objetivo geral e o específico que deve ser conquistado durante o desenvolvimento deste presente trabalho para que conclua o projeto.

No Capítulo 3, mostra os conceitos priorizados na construção do Web Service tendo uma visão ampla do projeto.

No Capítulo 4, são os métodos adotados nesses trabalhos, mostrando as ferramentas utilizadas, linguagem de programação e outras tecnologias que foram necessários, pois a partir desse ponto o projeto começa a desenvolver o sistema.

No Capítulo 5, mostra as análises, os progressos do sistema e as interfaces das classes java e dos projetos.

No Capítulo 6, são as considerações finais abordando uma análise crítica e conclusivas e todas as possíveis ideias que possam ser integradas no sistema para os futuros projetos.

No Capítulo 7, segue com as referências.

2. OBJETIVO

2.1 Geral

Desenvolver uma camada de acesso aos dados ambientais e imagens de satélite, disponibilizando diferentes tipos de formatos de dados. Fornecendo às diversas aplicações. Construir um Web Service onde a DSA e comunidade possa acessar as imagens e dados ambientais.

2.2 Específico

Desenvolver uma capitação de técnica de programação essencialmente com a linguagem Java que faz parte de todo o desenvolvimento da aplicação.

Conhecer os conceitos e devolver habilidades da ferramenta de desenvolvimento de projetos, o Eclipse e aprofundar no software responsável pelo gerenciamento de banco de dados, o PostgreSQL que é de extrema importância no projeto pois visa os acessos aos dados ambientais necessitando de um tratamento apropriado.

O Web Service implementa diversas tecnologias que aumentam o seu desenvolvimento. Framework é uma dessas tecnologias, na qual o Vraptor se destaca, pela sua facilidade desenvolver aplicações em estruturas MVC e tratando das requisições.

3. FUNDAMENTAÇÃO TEÓRICA

Web Service é um conjunto de sistemas que se comunicam entre si através da Internet. Esta comunicação é realizada com intuito de facilitar a integração de sistemas (EAI) que por sua vez traz interoperabilidade entre a informação que circula numa organização nas diferentes aplicações

A estrutura MVC tem extrema importância por separar as classes em pacotes, que são grupos voltados para uma determinada funcionalidade com diferentes níveis.

> "*MVC é o acrônimo de Model-View-Controller (em português,* ***Modelo-Visão-Controle****) e que, como o próprio nome supõe, separa as camadas de uma aplicação em diferentes níveis. Ao contrário do que muitos desenvolvedores dizem, o MVC não é um Padrão de Projeto, mas sim um Padrão de Arquitetura de Software."* **(André Luis Celestino).**

O projeto torna-se mais limpo e compreensível para o programador, na busca de algum erro ou melhoramento por que para cada pacote desempenha um padrão de tarefas e é fácil de identificar a classe comprometida.

O padrão REST aborda os conceitos e as características fundamentais para a construção de aplicações Web. Adotando os principais métodos http, destacando o GET utilizado quando é preciso adquirir um recurso, o POST para criar algum recurso no servidor, PUT permite a atualização de recurso no servidor e DELETE para apagar o recurso. O Recurso são documentos que pode ser identificada, nomeada ou endereçada, ou seja, um elemento utilizado para mapeamento de uma URL. No Web Service **RESTful**, as solicitações feitas ao URI (utilizada para identificar ou denominar um recurso na Internet) de um recurso provocará uma resposta com uma carga útil formatada em XML ou JSON.

4. METODOLOGIA

O desenvolvimento do projeto visa a utilização de ferramentas livres, que permitam utilizar um ambiente heterogêneo, com sistemas operacionais Linux/Unix e Windows. Será utilizado também conceitos de computação distribuída e camadas de serviços, onde o processamento das requisições pode ser dividido entre os equipamentos do ambiente computacional. Essa metodologia distribuída permite a otimização dos recursos computacionais, permitindo diminuição nos investimentos futuros na aquisição de servidores.

4.1 Ferramentas

Para o desenvolvimento de uma camada de acesso ao banco de dados é necessário conhecer as tecnologias que serão trabalhadas. São todas as ferramentas utilizadas que contribuiu para o desenvolvimento da aplicação Web Service.

4.1.1 Eclipse ide

Software para desenvolvimento Java, possibilita criar diversas aplicações desde de um produto inicial ao final, pois esse é opensource no que facilita no custo, ou seja, em vez de pagar diversas licenças para desenvolver a sua aplicação utiliza-se o eclipse que é livre de pagamento.

Tem muitas possibilidades para trabalhar com diferentes plataformas sendo elas: criar programas, aplicativos mobile e as aplicações Web Services. Além de ser opensource, traz uma variedade de biblioteca, recursos permitindo uma aplicação profissional e uma comunidade de programadores Java sempre à disposição para ajudar nas aplicações.

4.1.2 MAVEN

É uma ferramenta do Eclipse que automatiza o uso de outras bibliotecas ou Frameworks para usá-los através das dependências. As dependências em Java são arquivos comprimidos e armazenados com a extensão “.jar”. Alguns dessas dependências possuem uma ou mais subdependências que essas dependem de outras.

Tem um repositório oficial do Maven onde faz os downloads das dependências e além dele existem muitos outros repositórios é através do site http://mvnrepository.com/ que indexa diversos dependências e disponibiliza em um formato padrão trazendo muita facilidade para a sua implementação.

E as dependências, veem no formato XML para serem adicionado no pom.xml. Após salvar o documento, o Maven faz automaticamente o downloads das bibliotecas e incrementar no projeto.

Formato de uma dependência:

```
<dependency>
   <groupId>link/distrubuidora</groupId>
   <artifactId>tecnologia</artifactId>
   <version>10.5</version>
</dependency>
```

4.1.3 Tomcat 8.5

O Tomcat é um servidor web Java, mais especificamente, um container de servlets. O Tomcat implementa, dentre outras de menor relevância, as tecnologias Java Servlet e JavaServer Pages. Desenvolvido pela Apache Software Foundation, é distribuído como software livre.

4.1.4 Jax-rs

A api JAX-RS nos permite trabalhar com o que foi denominado como Restful WebServices. E segundo a arquitetura REST nós devemos expor as informações importantes de nossa aplicação como *recursosresouce*. Para isso precisamos criar uma classe que é definida pela especificação como Root. Para criação das URLs damos por anotações escrito encima das classes: @Path(“/pedido/{id}”). E o próprio JAX-RS reconhece {id} como sendo um parâmetro que é enviado através da URL.

4.1.5 Vraptor 4

O VRaptor, iniciativa brasileira para Controller MVC, é utilizado para facilitar o desenvolvimento web, evitando o contato com as classes pouco amigáveis do javax.servlet, deixando o código legível e desacoplado, ideal para testes. Fornecendo uma alternativa mais simples e eficiente para desenvolver aplicações MVC utilizando recursos como: anotações, injeção de dependências.

4.1.6 PostgreSQL

É software que tem a funcionalidade de ser um Sistema Gerenciador de Banco de Dados relacional de código aberto e nos permite gerenciar toda a administração e a manutenção de banco de dados, como o controle dos privilégios de acesso, ajustes de performance, backup e recuperação.

4.2 Linguagens

São necessários a utilização de linguagem no progresso do projeto, que moldará a sua estrutura e interface com relação as ferramentas utilizadas que foram abordadas acima nesse relatório.

4.2.1 JAVA

O Java é uma linguagem de programação multiplataforma, com uma sintaxe até certo ponto parecida com o C++, porém com bibliotecas diferentes. Os programas em Java podem ser executados em qualquer sistema operacional, desde que o interpretador esteja instalado. Uma linguagem de Programação Orientada a Objetos, isto é, implementada a partir de classes que definem características de objetos que são utilizados no momento de programar.

No projeto, tem importante função de “dar vida ao sistema”, como por exemplo de tratar as requisições web acompanhada do VRaptor, ou seja, pra cada função do Web Service tem uma classe responsável que é composta por algoritmos em Java. Por exemplo: ao clicar em um botão do web Service, envia esse parâmetro pelo servidor recebido pela classe de Controller que ajusta o botão. Além disso, disponibiliza as imagens através das consultas que são feitas ao banco de dados e é retornado no Web Service através de outras classes Java responsável nesta parte de exibição.

4.2.2 XML

É um tipo linguagens capaz formatar diversos tipos de dados, tendo como propósito facilitar a integração de outras linguagens que receberão esse padrão (XML), contendo os dados, facilitando seu compartilhamento de informações através da Internet.

4.2.3 JSON

O JSON é um formato mais leve conhecido por transferência de dados, conceitos parecidos com o XML pois tem a mesma utilidade, mesma façanha, porém é mais leve, não é necessário trabalhar com JavaScript porque a maioria das linguagens atuais dão suporte ao JSON. Implementamos ele para retornar dados vindo de um servidor e nos permite também atualizar as informações em tempo real.

4.2.4 HTML

Esta é uma linguagem de programação voltada para o desenvolvimento Web, ou melhor, para construção de páginas aquela acessada pelo browser dos computadores, smartphones e outros dispositivos que tem essa tecnologia.

4.2.5 CSS

Essa linguagem que define os estilos composta por camadas e utilizada para definir os elementos visual e como eles será apresentado, ou seja, esta é responsável pela estética da aplicação, das páginas ou do Web Service que adotam para o seu desenvolvimento acompanhado de outras linguagens de marcação (como XML, HTML e outras).

4.3 Estruturação

O projeto faz o uso de diversas tecnologias, tornando o trabalho complexo necessitando criar uma estrutura para se expressar visualmente os dados, os elementos, de maneiras diferentes, assim facilitando a compreensão dos mesmos. A estrutura se inicia com VRaptor, tratando as requisições e disponibilizando em formato JSON no Web Service, adiante apresenta-se a camada Model (DAO) interagindo com diversos bancos de dados ambientais, que possuem diferentes estruturas como Imagens de Satélite, Plataformas de Coleta de Dados e a Rindat, logo abaixo mostra uma ilustração da estrutura:

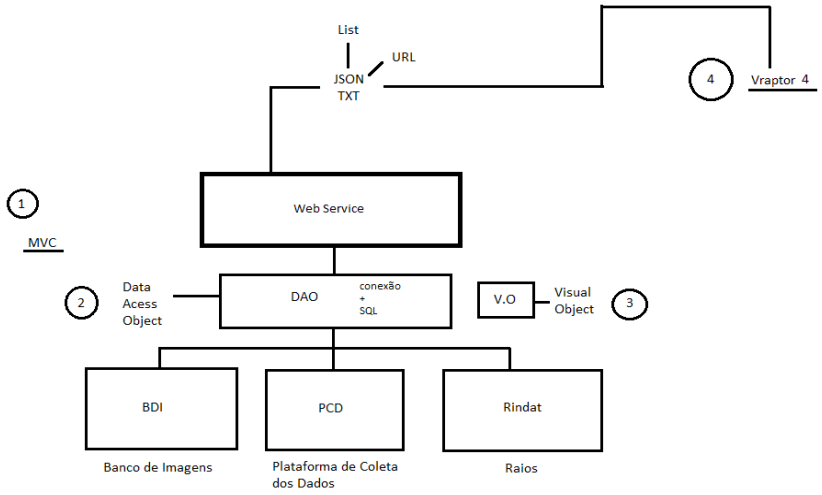

Figura 1: Estrutura do Projeto

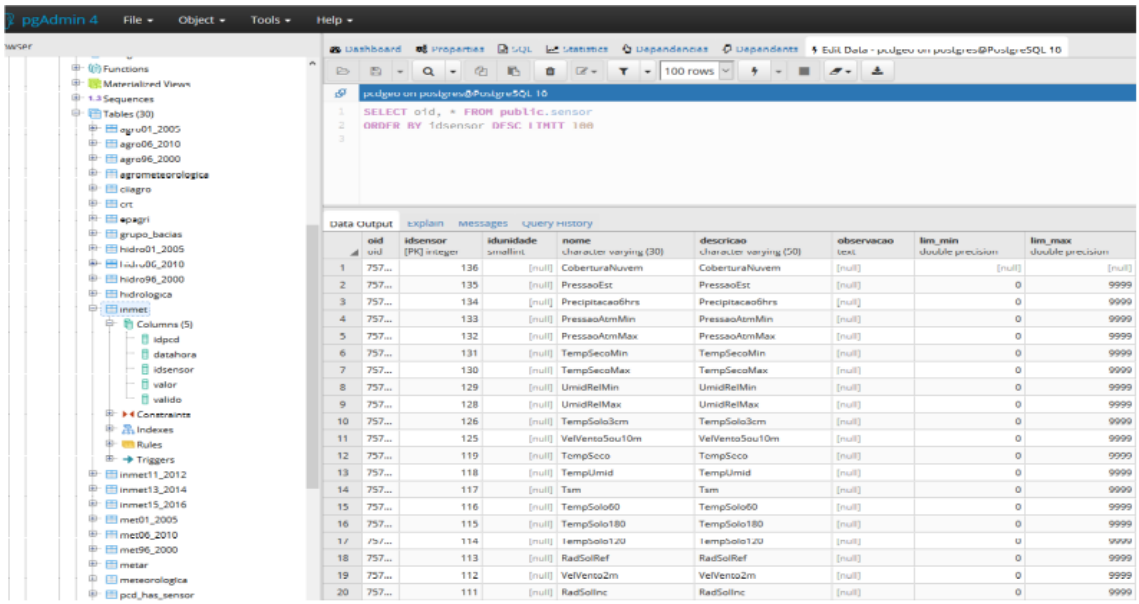

Figura 2: Inteface PostgreSQL

Adotado a utilização do PostgreSQL como nosso SGBD, o pgAdmin 4 é a nova interface web muita intuitiva pode ser trabalhada com diversas aplicações ou projetos inclusive o Web Service. Essa é a grande vantagem de se trabalhar com Web Service pois integra diversos sistemas.

5. ANÁLISE E RESULTADOS

Após o eclipse gerar uma aplicação inicial com artefatos e groupId definidos, passa pra etapa de preparação do projeto, implementos das ferramentas e configurar a nossa Project Facets, adicionando o módulo Dynamic Web, JavaServe Faces e JAX-RS. É importante notar a versão do java tem que ser a mais recente para não ocasionar em erros mais pra frente.

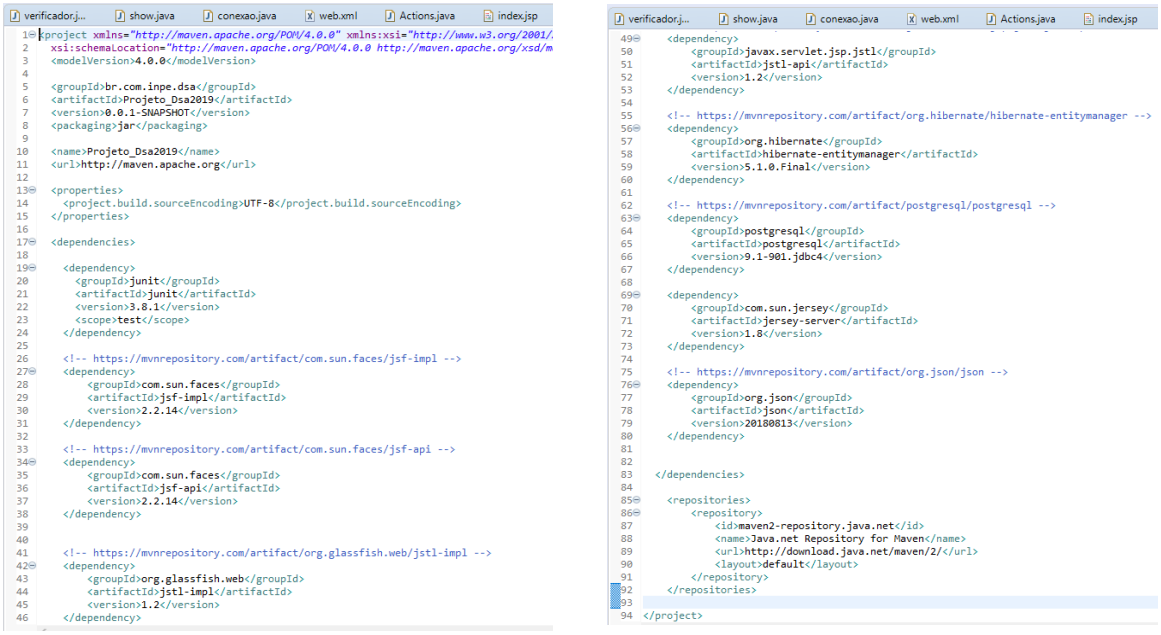

Figura 3: Interface Eclipse – pom.xml

Configurar o pom.xml para colocar novas tecnologias, através das inserções de dependências que abordamos na metodologia deste trabalho. Como por exemplo, implementação do PostgreSQL para fazer a conexão e todas atividades voltadas para o banco de dados. Depois de salvar, o Maven atualiza automaticamente o projeto fazendo download do seu repositório dos respectivos elementos.

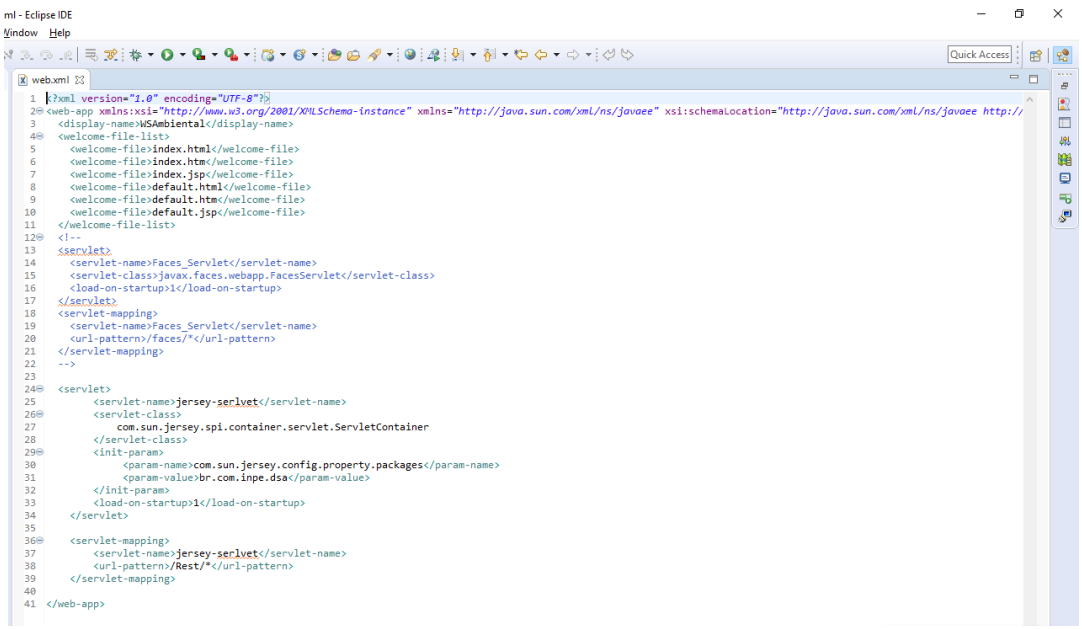

Figura 4: Interface Eclipse – web.xml

No web.xml, configura-se o percurso da URL em conjunto do Jersey. Após o nome do projeto passaria para o “Rest”(nome adotado para utilização do Jersey), e adiante os nomes adotado cada classe usando as anotações Path. Então o endereçamento da URL ficou desta forma “Localhost:8080//WSAMBIENTAL/Rest/” sendo que Localhost é servidor local da aplicação e o “8080” é a porta que servidor está usando e “WSAMBIENTAL” o nome do projeto no eclipse.

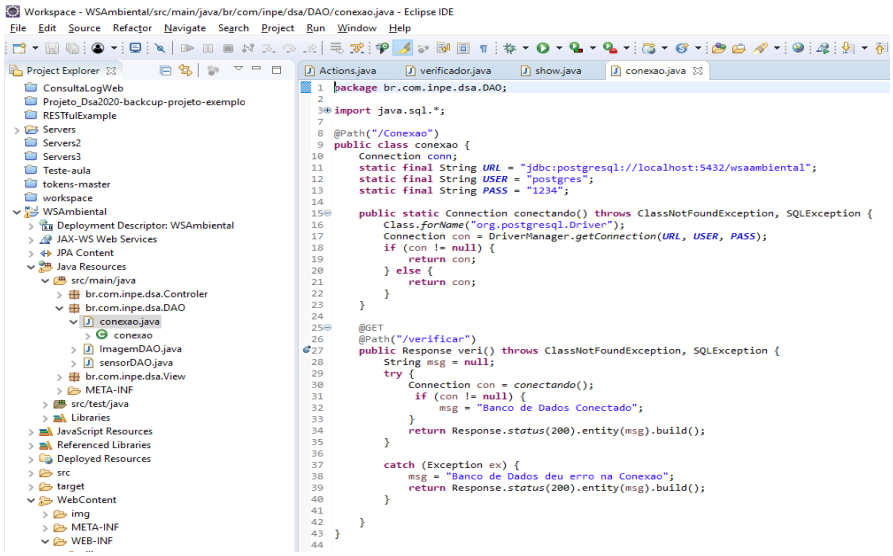

Figura 5: Inteface Eclipse – classe Conexao

A classe de Conexao, tem como objetivo fazer a conexão ao banco de dados recebendo uma anotação do tipo Path para verificar se a conexão foi bem-sucedida quando é acessada pela URL no browser, deve retornar uma mensagem do tipo "`Banco de Dados Conectado`" ou se houver falha traz uma mensagem de erro senão é possível fazer busca ou alterações no banco de dados.

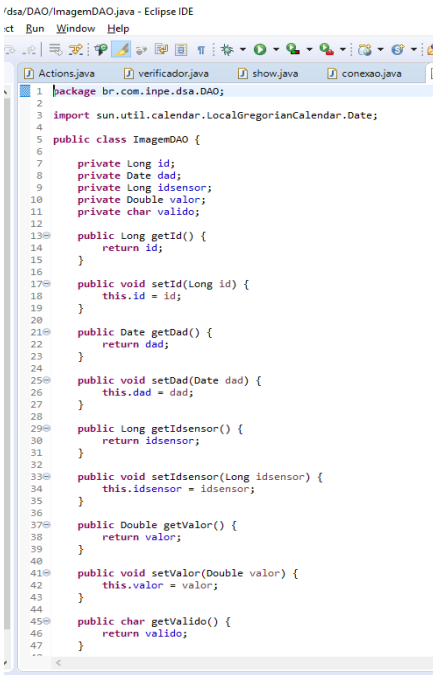

Figura 6: Inteface Eclipse – classe ImagemDAO

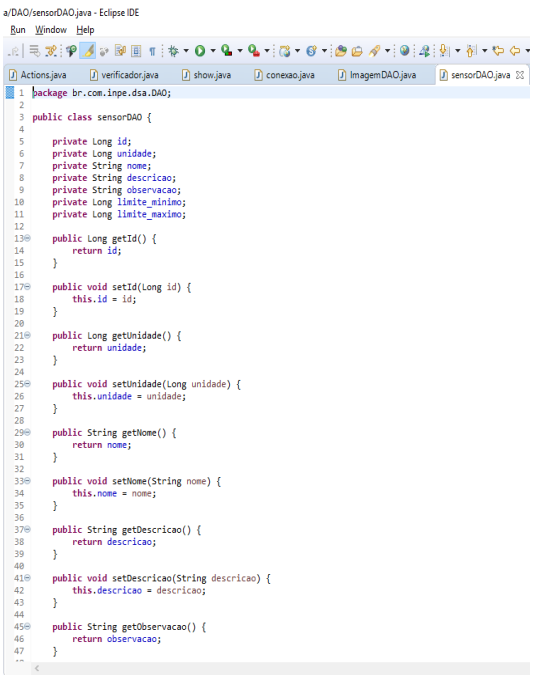

Figura 7: Inteface Eclipse – classe sensor

Ao criar os getter e setter das variáveis que corresponde itens das tabelas no banco de dados, servem para obter informações de variáveis da classe que são definidas como "private". Atualmente são duas classes com get e set para as duas tabelas do banco de dados, sendo trabalhadas a sensor e a inmet.

A seguir mostra as interfaces do Web Service:

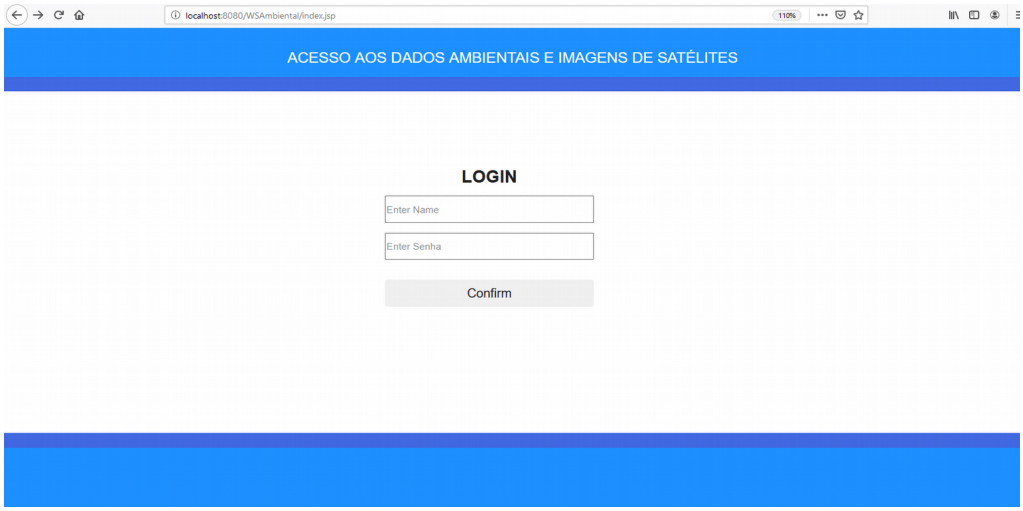

Figura 8: Interface Web Service – Login

Interface de login, isso nos permite fazer uma restrição dos usuários, ou seja, apenas as pessoas permitidas poderão utilizar os recursos do Web Service.

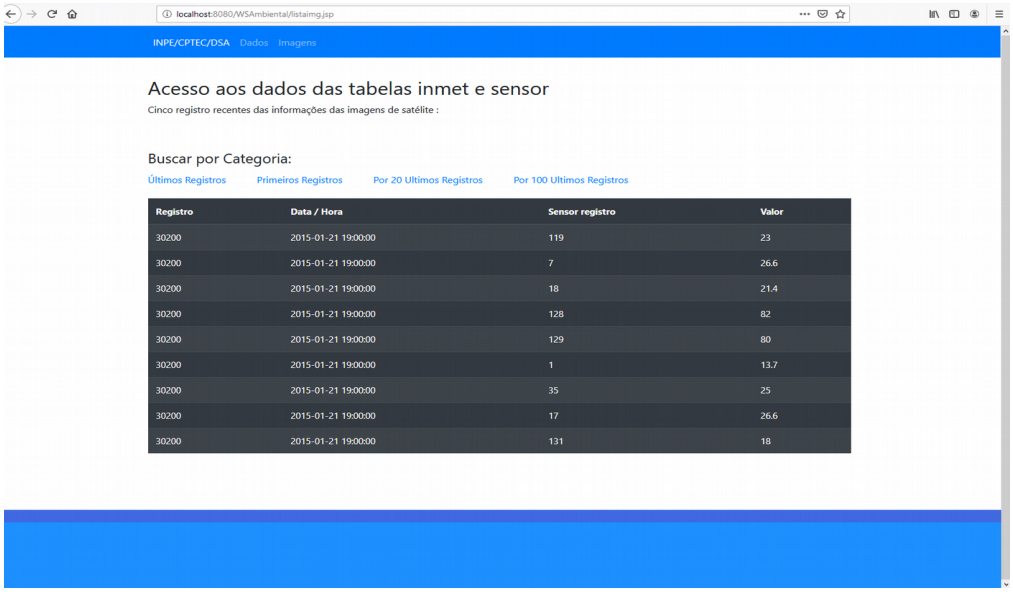

Figura 9: Interface Web Service – Exibindo os dados

Ao acessar a URL desta página exibirá os dados relacionado as imagens e os dados ambientais.

6. CONSIDERAÇÕES FINAIS

Pode concluir que, o projeto tem grande eficiência na disponibilização nas imagens e informações ambientais com diversos formatos fornecendo uma camada padronizada e organizada de acesso e distribuição dos dados ambientais armazenados na Divisão de Satélite e Sistemas Ambientais (DSA), permitindo a integração desse conjunto de dados com os demais sistemas também fornecidos pela divisão.

Proporcionando ao usuário os dados de satélite e com fácil acesso e disponibilidade pelo meio do Web Service, sendo uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes, é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

O próximo passo necessário para o melhoramento do projeto é dar funcionalidade nas classes de busca por imagens no banco e em seguida retornar o resultado em formato JSON na web e também criar um processo de validação do login a fim de torná-lo funcional.

7. REFERÊNCIAS

JUNIOR, Sérgio Azevedo. **Arquitetura REST com java: JAX-RS.** Disponível em: <https://blog.caelum.com.br/arquitetura-rest-com-java-jax-rs/> Acesso em:14 julho 2019

Matheus C. S. **Utilização do MAVEN.** Disponível em: <https://pt.stackover flow.com/questions/59240/para-que-serve-o-maven> Acesso em: 10 julho 2019

WEB SERVICE. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2019. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Web_service &oldid=55562041>. Acesso em: 24 junho 2019

Web ágil com VRaptor, Hibernate e AJAX. In: **Caelum, Apostila do curso FJ-28. Desde 2004.** Disponível em: <https://www.caelum.com.br/apostila-vraptor-hibernate/o-curso/#1-2-sobre-o-curso-hibernate-vraptor-jsp-e-ajax> Acesso em: 15 abril 2019

APACHE MAVEN. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2018. Disponível em: <https://pt.wikipedia.org/w/index.php? title=Apache_Maven&oldid=52030705>. Acesso em: 9 abril 2019.

Carlos E. Morimoto**. Java. Disponível em: <https://www.hardware.com.br/termos /java>** Acesso em: 15 julho 2019.

André Luis Celestino. **O conceito e as dúvidas sobre o MVC.** Disponível em: <https://ww w.profissionaisti.com.br/2014/10/o-conceito-e-as-duvidas-sobre-o-mvc/> Acesso em: 20 maio 2019.

Ana Paula Pereira. **O que é xml?** Disponível em: <https://www.tecmundo.com.br/pro gramacao /1762-o-que-e-xml-.htm> Acesso em 17 jul. 2019

Paulo Roberto. **O que é Json? Para que serve e como funciona?** Disponível em: <https://pt.stackoverflow.com/questions/4042/o-que-%C3%A9-json-para-que-serve-e-como- funciona> Acesso em 17 jul. 2019