

Serviços web para integração de bases de dados de recursos naturais

Felipe Todra Pavan¹

Luciano Vieira Koenigkan²

Alan Massaru Nakai²

Carla Geovana do Nascimento Macário²

O projeto NATDATA (MACÁRIO et al., 2011), tem como objetivo geral o de prover para a Empresa Brasileira de Pesquisa Agropecuária (Embrapa) um ambiente tecnológico para a gestão da informação de recursos naturais dos biomas, visando a geração de inteligência competitiva para a agricultura brasileira.

O problema encontrado no projeto é o de integrar várias bases de dados heterogêneas, construídas com sistemas gerenciadores de bancos de dados (SGBD) diferentes e que necessitam se comunicar para troca de informações com a plataforma em questão.

Neste contexto, a aplicação da tecnologia de serviços web constitui uma peça importante para o desenvolvimento da plataforma pretendida sendo este o foco deste trabalho. Um serviço web é uma ferramenta muito útil para a integração de sistemas e em comunicação de aplicativos e plataformas diferentes.

O objetivo geral do trabalho é o de desenvolver e aplicar a tecnologia de serviços web para a integração de dados de recursos naturais, como aqueles oriundos de bancos de dados de solos e clima.

A primeira etapa do trabalho consistiu no estudo da especificação JPA ou Java *Persistence* API (SUN MICROSYSTEMS, 2009) e de sua implementação Hibernate, bem como foram realizados estudos das ferramentas Jboss AS (BRAUN, 2013) e do SGBD PostgreSQL.

¹ Fatec/Americana - felipe.pavan@colaborador.embrapa.br

² Embrapa Informática Agropecuária - {luciano.vieria, alan.nakai, carla.macario}@embrapa.br

A segunda etapa envolveu o desenvolvimento de serviços web, onde foram estudados os padrões existentes, tendo sido selecionado e utilizado o padrão Representational State Transfer (REST) (FIELDING, 2000).

Na etapa seguinte, iniciou-se o desenvolvimento do mecanismo de filtro flexível, onde as rotinas desenvolvidas preparam cláusulas de seleção em SQL de acordo com a escolha do usuário, permitindo a realização de consultas com encadeamento de critérios para seleção.

A arquitetura do filtro flexível consiste em três classes: Clausula, ClausulaComposta e ClausulaParser. A classe Clausula permite uma seleção envolvendo atributos e valores utilizando operadores de comparação. Já a classe ClausulaComposta recebe objetos da classe Clausula e faz a junção com outras cláusulas já existentes utilizando operadores lógicos, e possui recursos que tornam possível a delegação de prioridades na seleção. A classe ClausulaParser faz com que, após o cliente enviar uma requisição ao servidor contendo os parâmetros necessários, seja organizado em uma instrução de seleção em linguagem SQL.

Como exemplo temos: (A == 10) && (B > 5), onde serão selecionados os dados em que o atributo A possui valor 10, sendo está comparação representada por um objeto da classe clausula, bem como no termo seguinte onde são selecionados itens com o valor de B maiores que 5 e por fim, as cláusulas são unidas pelo operador lógico && (and), representado por um objeto da classe ClausulaComposta.

Dessa forma o serviço, através da classe ClausulaParser criará a instrução de seleção SQL equivalente.

Uma vez que a implementação permite o encadeamento de mais de uma cláusula uma cláusula simples ou composta, torna-se possível expressar por meio destes objetos critérios complexos de seleção dos dados.

Após a finalização do mecanismo de filtragem, o mesmo foi aplicado na forma de serviços web para a recuperação de dados.

Por fim, foi criado um serviço web que permite a recuperação de informações sobre os dados presentes nos bancos de dados, a partir de suas descrições em formato XML. Em tal serviço, o cliente envia uma requisição para o servidor através de um serviço web, e recebe as informações dos bancos de dados disponíveis no sistema. A lista recebida se transforma em uma espécie de cardápio de dados que auxilia o usuário na identificação dos campos presentes nas bases.

A Figura 1 ilustra a arquitetura da aplicação, na qual o cliente por meio de um serviço web, faz uma requisição proveniente de algum banco ligado ao sistema. A aplicação acessa o serviço da ação a ser tomada e consulta a base de dados desejada para a coleta de informações, que serão repassadas ao usuário da plataforma.

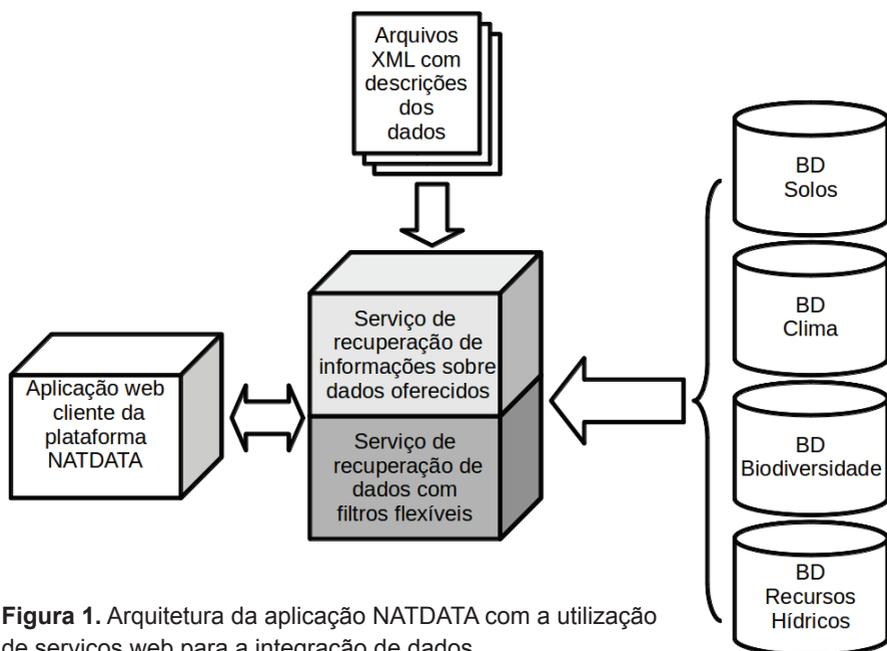


Figura 1. Arquitetura da aplicação NATDATA com a utilização de serviços web para a integração de dados.

Os resultados globais obtidos pela execução do trabalho foi o desenvolvimento de serviços web que contribuem para a implementação da plataforma de integração de dados NATDATA.

Referências

BRAUN, H. **Developer guide**. JBOSS. Disponível em: <<https://docs.jboss.org/author/display/AS71/Developer+Guide>>. Acesso em: 25 mar. 2013.

FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. 2000. 180 p. Doctor of Philosophy (Dissertation in Information and Computer Science) - Information and Computer Science, University of California, Irvine. Disponível em: <http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>. Acesso em: 20 jun. 2013.

MACÁRIO, C. G. do N.; ASSAD, E. D.; PIEROZZI JUNIOR, I.; EVANGELISTA, S. R. M. **NATDATA: integrando dados de recursos naturais dos biomas brasileiros**. In: CONGRESSO BRASILEIRO DE AGROINFORMÁTICA, 8., 2011, Bento Gonçalves. Anais ... Florianópolis: UFSC; Pelotas: UFPel, 2011. 1 CD-ROM. SBIAgro 2011.

SUN MICROSYSTEMS. **JSR-000317 Java persistence 2.0 - final release**. Disponível em: <http://jcp.org/aboutJava/communityprocess/final/jsr317/>. Acesso em: 8 maio 2013.