

Sistema de apoio à análise da vulnerabilidade natural à poluição do recurso hídrico subterrâneo

Caio M. Marques¹, Ivan R. Guilherme¹, Eduardo G. Foster², Eduardo R. N. Assis²
Fábio J. Meaulo³

¹Programa de Pós-Graduação em Ciências da Computação - Universidade Estadual Paulista (UNESP) 13.500-171 – Rio Claro – SP – Brasil

²Caiena – Soluções em Gestão do Conhecimento – Incubadora de Base Tecnológica da UNESP - INCUNESP (INCUNESP) 13.500-171 – Rio Claro – SP – Brasil

³FortGEO – Geociências&Meio Ambiente – Incubadora de Base Tecnológica da UNESP - INCUNESP (INCUNESP)

caiena.caio.marques@gmail.com, ivan@rc.unesp.br,
{foster,eduardo}@caiena.net, meaulo@ig.com.br

Abstract. *This article presents a Web Geographical Information System Architecture, that meets the necessary requirements to visualize geographical data, environmental scenarios simulation and to implement environmental data analyzes methods. The architecture has been built integrating the concepts of a Services Oriented Architecture (SOA) and, also, of Rich Internet Applications (RIA). To validate the presented architecture, a Geographical Information System prototype was developed and it shows a method to analyze water resource's natural vulnerability to pollution.*

Resumo. *Neste artigo é apresentada uma arquitetura de um Sistema de Informação Geográfica para a Web, que atende os requisitos necessários para a visualização de dados geográficos, simulação de cenários ambientais e para a implementação de métodos de análise de dados ambientais. A arquitetura foi concebida integrando os conceitos de uma Arquitetura Orientada a Serviços (SOA) e das Aplicações Ricas para a Internet (RIA). Com o objetivo de validar a arquitetura, é apresentada a implementação do protótipo de um Sistema de Informação Geográfica, onde é sistematizado o método de análise da vulnerabilidade natural à poluição do recurso hídrico.*

1. Introdução

A informação georeferenciada é especialmente relevante no contexto das aplicações relacionadas à gestão do meio ambiente. Durante décadas, pesquisadores e profissionais de georeferenciamento dedicaram-se em criar infra-estruturas de dados espaciais (*Spatial Data Infrastructures – SDI*) para apoiar as atividades humanas. Inicialmente, estas iniciativas foram realizadas isoladamente, resultando em infra-estruturas monolíticas, fortemente acopladas e com baixa capacidade de interoperar.

Com a expansão da Web, a sociedade tornou-se mais interconectada e colaborativa. Além disso, os problemas ambientais apresentam-se cada vez mais complexos, exigindo a integração e colaboração de grupos de pesquisas distribuídos

geograficamente. Este cenário determinou a evolução das SDIs no sentido das Arquiteturas Orientadas a Serviços – SOA. Considerando as aplicações ambientais, este cenário é ainda mais evidente [Gomes 2007].

Neste contexto, este trabalho descreve uma arquitetura de um Sistema de Informação Geográfica para a Web, que suporte os requisitos necessários para a visualização de dados geográficos, simulação de cenários ambientais e implementação de métodos de análise de dados ambientais. A arquitetura é concebida com bases em conceitos de Arquiteturas Orientadas a Serviços (SOA) e de Aplicações Ricas para Internet (RIA), com o objetivo de obter avanços em performance, incremento na usabilidade, interoperabilidade de dados e métodos de análise, além do baixo acoplamento entre o cliente e o servidor.

Este trabalho está organizado conforme segue: na Seção 2, são apresentados os aspectos teóricos relacionados às Aplicações Ricas para a Internet (RIA) e à Computação Orientada a Serviços (SOC); na Seção 3, a arquitetura proposta é descrita; na Seção 4, são apresentadas funcionalidades relacionadas à análise de dados ambientais, na Seção 5 são destacadas as conclusões e os trabalhos futuros.

2. Aspectos teóricos

Nesta seção são apresentados os aspectos teóricos relacionados às Aplicações Ricas para a Internet (RIA), à Computação Orientada a Serviços (SOC) e ao método de análise da vulnerabilidade natural à poluição do recurso hídrico subterrâneo.

2.1. Aplicações ricas para a internet

O termo *Rich Internet Application (RIA)* foi introduzido por Jeremy Allaire em 2002 para definir a integração entre as aplicações *desktop* e Web tradicionais, com base nas melhores características das duas abordagens [Bozzon e Comai e Fraternali e Carughi 2006]. As evoluções propostas pelas RIAs estão relacionadas à solução de problemas recorrentes em aplicações web tradicionais, como performance e, principalmente, usabilidade. A finalidade principal da abordagem RIA é proporcionar aos usuários de aplicativos Web melhor usabilidade, baseada em elementos de interface (listas, barras de ferramentas, menus de contexto, entre outros) estabelecidos para as aplicações *desktop* tradicionais.

Embora a implementação da abordagem RIA possa variar conforme a técnica utilizada [Bozzon e Comai e Fraternali e Carughi 2006], há características que são compartilhadas, e que diferenciam as RIAs das aplicações Web tradicionais:

- Requisições ao servidor são realizadas em plano de fundo. Esta característica permite que a interface não fique inoperante até que uma resposta seja fornecida pelo servidor. Além disso, possibilita que partes das informações sejam recuperadas sob demanda do servidor, não necessitando que, a toda ação do usuário, todas as informações sejam retransmitidas;
- Sincronia de dados entre cliente e servidor. Os dados presentes na camada cliente e no servidor devem estar sempre sincronizados para garantir a consistência das operações e da aplicação como um todo.
- Lógica de negócios no cliente. Nas aplicações ricas é comum realizar boa parte da lógica de negócios na camada cliente.

- Camada de apresentação independente do servidor. A aplicação é a responsável por tratar os aspectos relacionados à camada de apresentação, independentemente do servidor.
- Elementos de interface ricos. As aplicações RIA fazem uso de primitivas de interface (árvores, listas, menus suspensos, entre outras) que proporcionam parte do incremento na usabilidade.

A natureza interativa dos SIGs e os grandes volumes de dados, característicos de aplicações geográficas, determinam a usabilidade e a performance como requisitos fundamentais de projeto. A abordagem RIA utilizada na implementação permite atingir estes requisitos.

2.2. Computação orientada a serviços

A computação orientada a serviços consiste em um paradigma que utiliza serviços como blocos básicos para a construção de aplicativos, tendo por finalidade, atender os requisitos de composição e execução de componentes de software em ambientes distribuídos.

Serviços são componentes de software auto-descritos, auto-contidos, independentes de plataforma, disponíveis através de uma rede, possuindo baixo custo de composição e baixo nível de acoplamento [Breivold e Larsson 2007][Papazoglou e Heuvel 2006].

Para suportar tais requisitos, foi proposto o modelo para as Arquiteturas Orientadas a Serviços (*Service Oriented Architecture – SOA*). Uma arquitetura orientada a serviços deve conter três partes: o provedor, o consumidor e o registro. O provedor registra os serviços disponíveis para que o cliente possa localizar e compor os serviços adequados à sua demanda.

Uma das implementações para o modelo SOA é referenciada como Serviços Web. Os Serviços Web foram padronizados pelo World Wide Web Consortium – W3C e utilizam a Web como plataforma [W3C 2002].

No contexto da Web, é possível implementar serviços sem utilizar os padrões e a arquitetura proposta pelo modelo SOA. Esta abordagem é referenciada como Serviços *Rest – Representational State Transfer* [Fielding 2000] e destaca-se por sua simplicidade de implementação.

No contexto dos serviços de informações geográficas, ambas abordagens descritas podem ser utilizadas. Os serviços geográficos são especificados e padronizados pelo *Open Geospatial Consortium – OGC* [OGC 2009]. Esta padronização garante que haja uniformidade no acesso aos dados geográficos através da Web, potencializando a interoperabilidade. Os principais serviços padronizados pelo OGC são classificados conforme segue:

- *Web Map Service*: padroniza o acesso e o intercâmbio de mapas na Web;
- *Web Feature Service*: padroniza o acesso e o intercâmbio de dados geográficos vetoriais na Web;
- *Web Coverage Service*: padroniza o acesso e o intercâmbio de imagens georeferenciadas.

Neste trabalho, os dados geográficos, o método de análise da vulnerabilidade natural e as funcionalidades gerais são implementadas por serviços Rest. Os serviços de dados geográficos estão implementados conforme os padrões estabelecidos pelo OGC.

2.3. Vulnerabilidade natural

O método de mapeamento da vulnerabilidade natural é o primeiro passo na avaliação do risco à poluição do recurso hídrico subterrâneo. Os mapas de vulnerabilidade natural à poluição combinam diferentes categorias de informações, que por sua vez, consideram as propriedades e/ou capacidade de proteção que cada tipo de material geológico apresenta ante aos diversos tipos de poluentes [Meaulo 2004].

O mapa de vulnerabilidade natural contribui para o planejamento urbano, auxiliando e disciplinando as diversas formas de intervenção antrópica no meio físico (instalação de indústrias de porte, aterros sanitários, sistemas de saneamento, entre outros) [Meaulo 2004].

O método de análise da vulnerabilidade natural à poluição do recurso hídrico está ilustrado na Figura 1 e pode ser sistematizado como segue:

Primeira fase: identificação do tipo de ocorrência da água subterrânea, apresentada no intervalo de 0-1. Nesta fase são especificadas 6 classes que representam os tipos de ocorrência da água subterrânea. Estas classes estão ilustradas na Figura 1;

Segunda fase: consiste na caracterização geológica da zona não saturada, representada numa escala de 0,4-1. A caracterização geológica é realizada com base em 15 classes de tipos de rocha. Estas classes estão ilustradas na Figura 1;

Terceira fase: estimativa da profundidade do nível d'água, exibido numa escala 0,6-1. O nível d'água é descrito a partir de 5 classes. Estas classes estão ilustradas na Figura 1;

Produto das fases: o produto dos três parâmetros será o índice de vulnerabilidade natural do aquífero, expresso numa escala de 0,1-1. Na Figura 1, nota-se a associação de intervalos às classes de vulnerabilidade natural.

O tipo de ocorrência da água subterrânea, a caracterização geológica e a profundidade do nível d'água são atributos associados a polígonos que representam a localização espacial do fenômeno em observação. Basicamente, são utilizados dois planos de informação para o cálculo da Vulnerabilidade Natural: o plano de informação Geologia, composto de polígonos que caracterizam a área de estudo segundo as classes geológicas expressas na Figura 1; e o plano de informação Hidrologia, composto de polígonos caracterizados pela classe de profundidade do nível d'água e pelo grau de confinamento. O índice de vulnerabilidade natural é obtido através do cálculo da intersecção entre os polígonos e da multiplicação dos atributos relevantes para o método. É importante ressaltar que os coeficientes adotados para cada classe, bem como os demais parâmetros utilizados no método, não são universais e foram propostos em [Meaulo 2004].

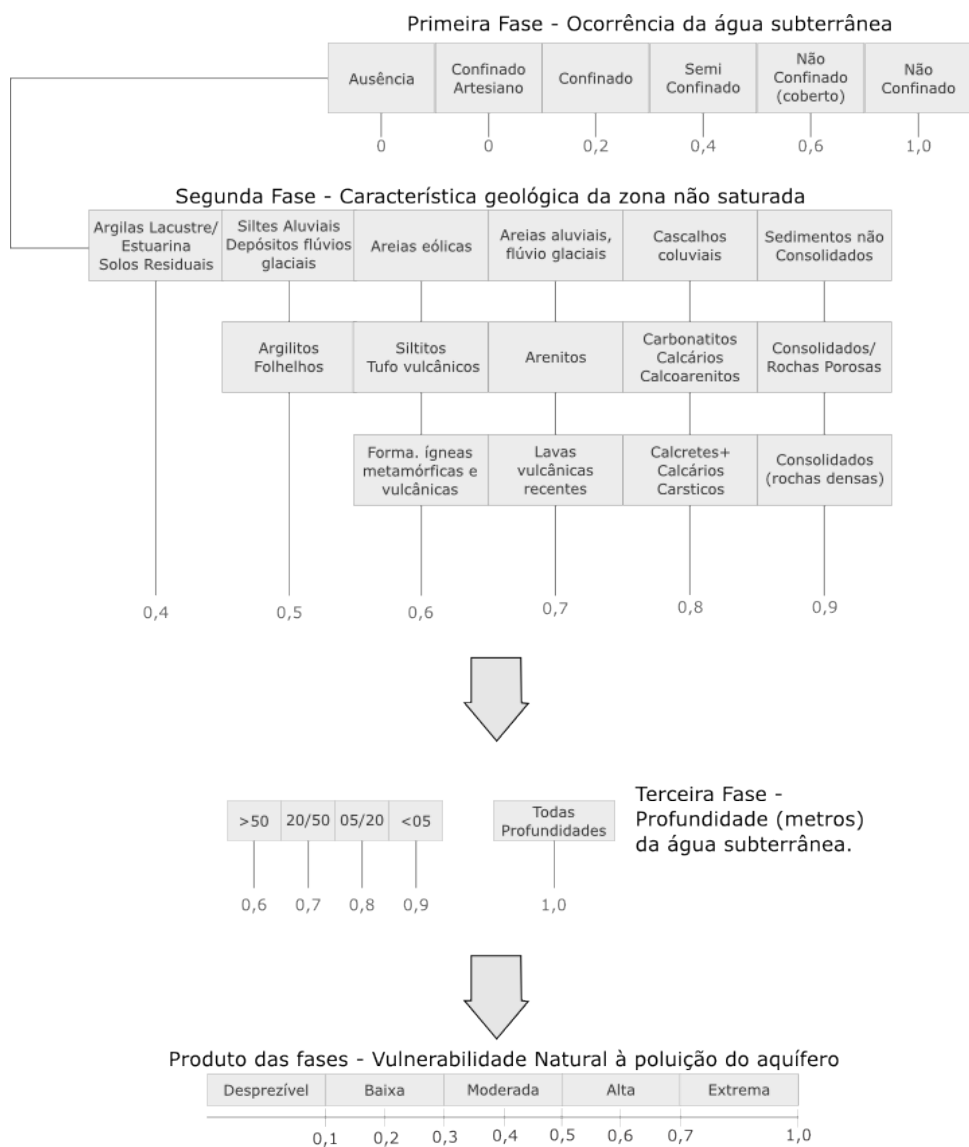


Figura 1. Método de análise da vulnerabilidade natural [Meaulo 2004].

3. Arquitetura

A arquitetura proposta neste trabalho é dividida nas camadas cliente, servidor e persistência. A camada cliente consiste na máquina virtual e nas bibliotecas utilitárias necessárias para a criação de aplicações RIA. A camada servidora consiste nos serviços que provêm acesso aos dados geográficos, aos métodos de análise e às funcionalidades gerais. A camada de persistência contém os dados geográficos e os arquivos XML que descrevem os métodos de análise. A arquitetura descrita está ilustrada na Figura 2.

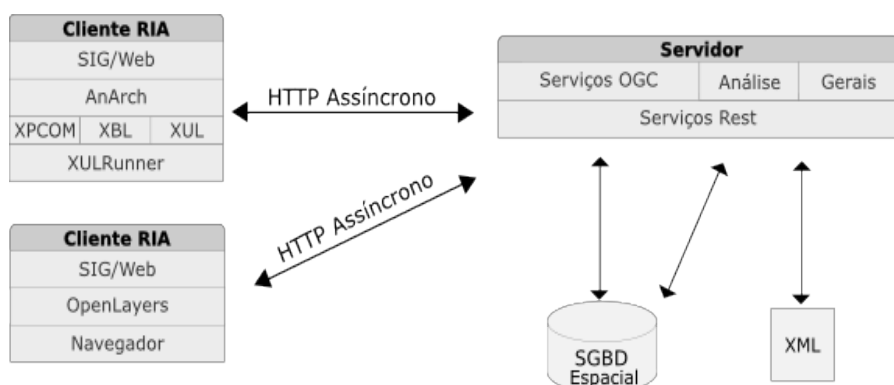


Figura 2. Arquitetura proposta.

3.1. Cliente

A camada cliente foi definida considerando os requisitos das seguintes funcionalidades: a visualização de dados geográficos, o simulador de cenários ambientais e a execução do método de análise da vulnerabilidade natural à poluição do recurso hídrico. Essas funcionalidades requerem a utilização da abordagem RIA (Seção 2.1), que consiste dos seguintes requisitos: requisições HTTP em plano de fundo, lógica de apresentação no cliente, além do baixo acoplamento entre o cliente e o servidor.

Para satisfazer os requisitos estabelecidos, a camada cliente consiste em uma máquina virtual e bibliotecas/*frameworks* utilitários para gerenciar requisições HTTP, facilitar operações DOM, entre outras.

O visualizador de dados geográficos e o simulador de cenários ambientais foram implementados utilizando o XULRunner da Fundação Mozilla [Mozilla 2009] e o *framework* AnArch [Foster 2007]. O XULRunner fornece ferramentas para o desenvolvimento de aplicações ricas, destacando-se sua linguagem para interfaces gráficas *Xml User interface Language* – XUL e seu padrão para componentização *eXtensible Binding Language* - XBL. O *framework* AnArch trata-se de um conjunto de ferramentas e bibliotecas que auxiliam na integração entre clientes desenvolvidos com o XULRunner e servidores de serviços.

A funcionalidade que executa o método de análise, utiliza o navegador web como máquina virtual e a biblioteca OpenLayers [OpenLayers 2009].

3.2. Servidor

Na camada servidora, encontram-se os serviços implementados conforme a abordagem Rest, e dividem-se em serviços de dados geográficos, análise de dados ambientais e gerais.

Os serviços de dados geográficos seguem o padrão WMS e WFS para serviços. Na implementação dos serviços é utilizado o provedor de serviços GeoServer [GeoServer 2009].

Os serviços de análise implementam os métodos de análise. No contexto deste trabalho, o método de análise da vulnerabilidade natural foi implementado. A descrição do cálculo do índice de vulnerabilidade natural foi implementado utilizando-se um arquivo no padrão *eXtensible Markup Language* – XML. Este arquivo fornece a

descrição dos parâmetros de entrada e sua relação, para a execução do método. Os parâmetros de entrada são: tipo de geologia, profundidade do nível d'água e grau de confinamento do lençol freático.

A obtenção do índice se dá através da caracterização de uma região de acordo com os parâmetros de entrada descritos pelo arquivo XML. Esta caracterização é então enviada ao serviço responsável por executar o método e retornar a vulnerabilidade natural.

Os serviços gerais são responsáveis pelas funcionalidades gerais da aplicação, como autenticação, permissões de acesso, entre outras. A organização dos serviços está ilustrada na Figura 2.

3.3. Persistência

A camada de persistência consiste em um banco de dados relacional com extensão espacial e um conjunto de arquivos XML que descrevem os métodos de análise disponíveis.

A camada de persistência é acessada pelos serviços e provê as informações geográficas e a descrição da sistemática dos métodos de análise.

Na implementação realizada neste trabalho, o servidor de banco de dados utilizado foi o PostgreSQL [PostgreSQL 2009] com sua extensão espacial PostGIS [PostGIS 2009].

4. Análise de dados ambientais

Utilizando a arquitetura, foi realizada a implementação de duas formas de análise ambiental: o simulador de cenários ambientais e o método de análise da vulnerabilidade natural à poluição do recurso hídrico.

O simulador de cenários ambientais consiste em uma funcionalidade de apoio ao planejamento urbano, com foco em variáveis ambientais. Esta funcionalidade permite determinar áreas de interesse para a instalação de empreendimentos potencialmente poluidores. O principal objetivo do simulador é proporcionar um meio eficaz para que os usuários realizem o cruzamento de variáveis ambientais e obtenham áreas de interesse, chamadas de cenários ambientais. No contexto deste trabalho, o simulador torna-se relevante, pois pode auxiliar na caracterização de áreas de interesse com base na vulnerabilidade natural da região de estudo.

Para criar um cenário é necessário caracteriza-lo com base nos fenômenos geográficos disponíveis. Como um exemplo, um cenário pode ser caracterizado como segue: “Todas as regiões geográficas onde ocorram os fenômenos Vulnerabilidade Natural Alta, nível d'água entre 5 e 10 metros e região de planejamento ambiental 2”. Este cenário retorna regiões que não seriam recomendadas para a instalação de empreendimentos potencialmente poluidores. Na Figura 3, está ilustrada a interface com o usuário responsável por fornecer os parâmetros para caracterizar o cenário.

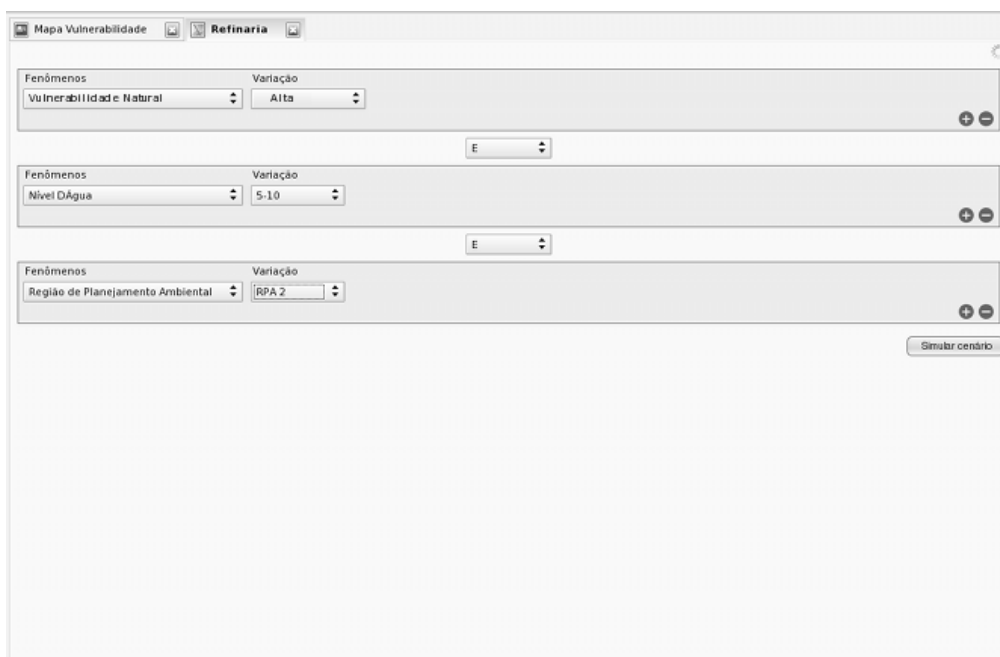


Figura 3. Caracterização de cenário ambiental.

Após caracterizar o cenário, a simulação pode ser executada. A consulta especificada pelo usuário é passada como parâmetro para um serviço que executa as operações espaciais necessárias com a finalidade de encontrar as áreas de interesse. Após o processamento, o serviço retorna a representação gráfica da simulação. A caracterização exemplo, resulta no mapa ilustrado na Figura 4.

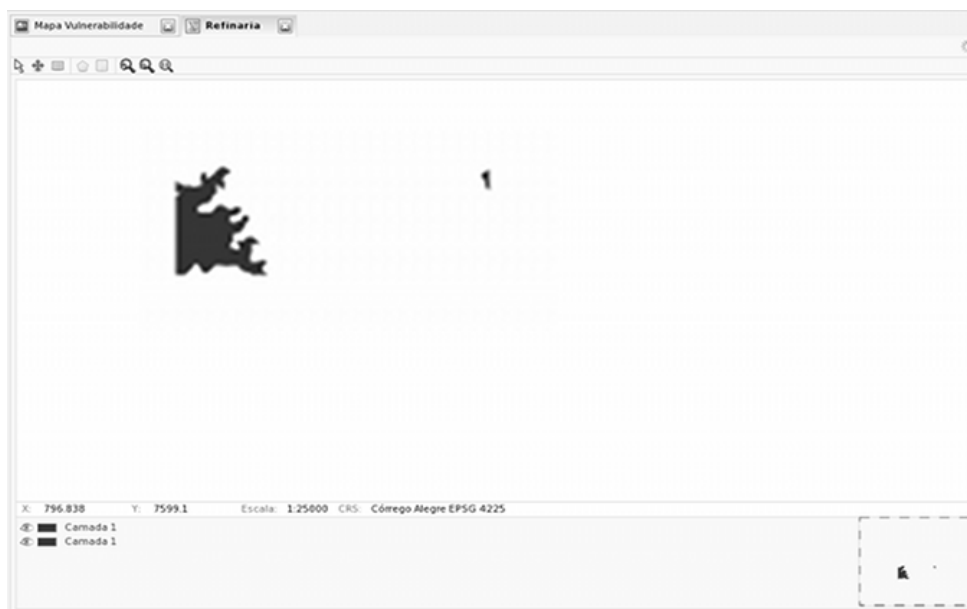


Figura 4. Resultado da simulação de um cenário ambiental.

A implementação do método de análise da vulnerabilidade natural permite que o usuário obtenha a classe e o índice de vulnerabilidade natural de acordo com a caracterização de determinada região geográfica.

Inicialmente, o especialista deve fornecer a caracterização da região de acordo com as variáveis ambientais utilizadas pelo método. No caso da vulnerabilidade natural, as variáveis são: grau de confinamento do lençol freático, extrato geológico e profundidade do nível d'água. Após, os parâmetros são enviados a um serviço, que interpreta o arquivo XML que descreve o método e retorna a classe de vulnerabilidade e o índice. Na Figura 5, está ilustrada a interface gráfica que permite a obtenção da classe de vulnerabilidade natural.

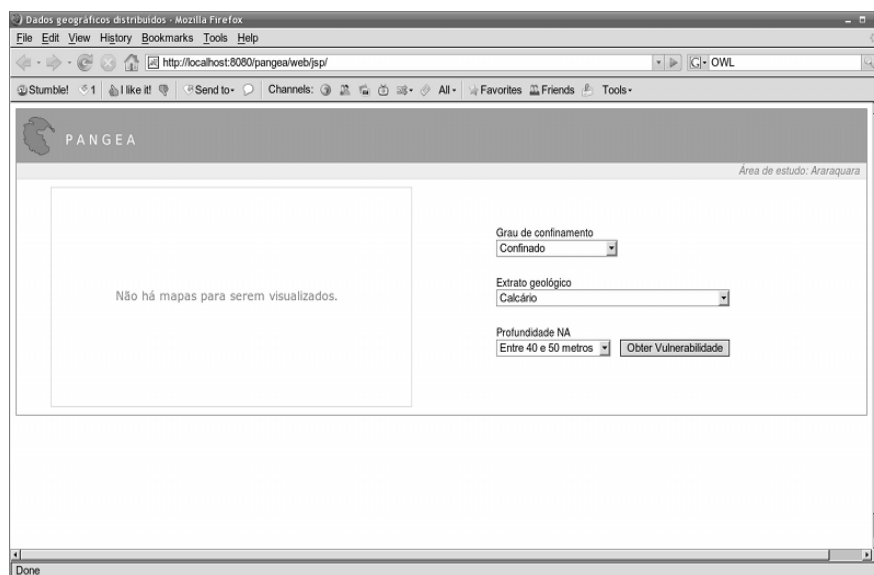


Figura 5. Cliente que executa o método de análise da vulnerabilidade natural.

5. Conclusões

Neste artigo é apresentada uma arquitetura de um Sistema de Informação Geográfica para a Web, que atende os requisitos necessários para a visualização de dados geográficos, simulação de cenários ambientais e para a implementação de métodos de análise de dados ambientais. A arquitetura foi concebida com base nos conceitos de uma Arquitetura Orientada a Serviços (SOA) e de Aplicações Ricas para a Internet (RIA).

Os requisitos de projeto estabelecidos para esta arquitetura são: incrementar usabilidade, interoperabilidade, performance e baixo acoplamento entre os clientes e o servidor.

A arquitetura foi validada com base na implementação de dois protótipos RIA: um deles baseado em navegador e na biblioteca OpenLayers e o outro no XULRunner e no “framework” AnArch.

As aplicações desenvolvidas com a abordagem RIA, bem como o método de análise da vulnerabilidade natural foram concebidos e validados em conjunto com um especialista. Os resultados demonstraram usabilidade mais rica comparadas às aplicações Web tradicionais. Ainda, por conta das requisições assíncronas, foi possível obter ganhos em performance. Por fim, a abordagem RIA permitiu maior independência entre o cliente e o servidor, na medida em que a responsabilidade de criar a camada de apresentação foi toda transferida para a camada cliente. A camada de serviços também contribuiu com o baixo acoplamento, uma vez que era responsável apenas por fornecer

dados, sem detalhes de apresentação. Os padrões OGC possibilitaram criar aplicações clientes completamente distintas, e ainda assim, ter acesso aos dados geográficos. Além da arquitetura, foi possível validar a técnica utilizada para sistematizar o método de análise da vulnerabilidade natural à poluição dos recursos hídrico.

Na continuidade do trabalho a arquitetura proposta esta sendo readequada para suportar serviços semânticos, com o objetivo de facilitar a localização, composição e execução dos serviços.

Referências

- Bozzon, A. and Comai, S. and Fraternali, P. and Carughi, G. (2006) “Conceptual modeling and code generation for rich internet applications”, In: International Conference on Web Engineering, 6.; International Conference on Web Engineering. Palo Alto: ACM, ISBN 1-59593-352-2.
- Breivold, H. and Larsson, M. (2007) “Component-based and service-oriented software engineering: key concepts and principles”, In: Euromicro, 33.; Conference on Software Engineering and Advanced Applications. Lübeck: IEEE.
- Fielding, R. T. (2000) “Architectural styles and the design of network-based software architectures”, Irvine, Dissertação de PhD – Universidade da Califórnia.
- Foster, E. (2007) “AnArch: uma arquitetura orientada a serviços para aplicações ricas para a internet”, Rio Claro, Trabalho de conclusão de curso – Universidade Estadual Paulista Júlio de Mesquita Filho.
- GeoServer (2009) “GeoServer”, <http://geoserver.org>, Último acesso: Março 2009.
- Gomes, L. C. J. (2007) “Uma arquitetura para consultas a repositórios de biodiversidade na Web”, Campinas, Dissertação de mestrado – Universidade Estadual de Campinas.
- Meaulo F. (2004) “Vulnerabilidade natural à poluição dos recursos hídricos subterrâneos da área de Araraquara (SP)”, Rio Claro, Dissertação de Mestrado – Programa de Pós-Graduação em Geociências e Meio Ambiente – Universidade Estadual Paulista Júlio de Mesquita Filho.
- Mozilla (2009) “XULRunner”, <https://developer.mozilla.org/en/XULRunner>, Último acesso: Março de 2009.
- OGC (2009) “OGC”, <http://www.opengeospatial.org/standards>, Último acesso: Março de 2009.
- OpenLayers (2009) “OpenLayers”, <http://openlayers.org/>, Último acesso : Março de 2009.
- Papazoglou, M. e Heuvel, W. (2006) “Service Oriented Design and development methodology”, In: International Journal Web Engineering and Technology, Vol. 2, No. 4.
- PostGIS (2009) “PostGIS” <http://postgis.refractory.net>, Último acesso: Março de 2009.
- PostgreSQL (2009) “Postgre” <http://postgre.refractory.net>, Último acesso: Março de 2009.
- W3C (2002) “Web services activity”, <http://www.w3.org/2002/ws/>, Último acesso: Março de 2009.