

# Boletim de Pesquisa 19 e Desenvolvimento dezembro, 2008

ISSN 1677-9266

## Uma proposta de plataforma de software para integração e interoperabilidade de serviços Web - WebAgritec Estudo de caso: Diagnose Virtual





ISSN 1677-9266  
Dezembro, 2008

*Empresa Brasileira de Pesquisa Agropecuária  
Embrapa Informática Agropecuária  
Ministério da Agricultura, Pecuária e Abastecimento*

*Cooperação Internacional da Embrapa e International Potash Institute*

# **Boletim de Pesquisa e Desenvolvimento 19**

## **Uma proposta de plataforma de software para integração e interoperabilidade de serviços Web - WebAgritec Estudo de caso: Diagnose Virtual**

Sílvia Maria Fonseca Silveira Massruhá  
Helano Póvoas de Lima  
Sílvio Roberto Medeiros Evangelista  
Gabriel Esteves Marques Pinto

Campinas, SP  
2008

**Embrapa Informática Agropecuária**  
**Área de Comunicação e Negócios (ACN)**

Av. André Tosello, 209

Cidade Universitária "Zeferino Vaz" – Barão Geraldo

Caixa Postal 6041

13083-970 – Campinas, SP

Telefone (19) 3211-5700 – Fax (19) 3211-5754

URL: <http://www.cnptia.embrapa.br>

e-mail: [sac@cnptia.embrapa.br](mailto:sac@cnptia.embrapa.br)

**Comitê de Publicações**

*Kleber Xavier Sampaio de Souza (presidente); Marcia Izabel Fugisawa Souza  
Martha Delphino Bambini; Sílvia Maria Fonseca Silveira Massruhá; Stanley Robson  
de Medeiros Oliveira; Suzilei Almeida Carneiro (secretária)*

Supervisão editorial: *Suzilei Almeida Carneiro*

Normalização bibliográfica: *Marcia Izabel Fugisawa Souza*

Revisão de texto: *Adriana Farah Gonzalez*

Editoração eletrônica: *Área de Comunicação e Negócios (ACN)*

**Suplentes**

*Goran Neshich; Leandro Henrique Mendonça de Oliveira e Maria Goretti Gurgel  
Praxedes*

**1ª. edição on-line - 2008**

**Todos os direitos reservados.**

A reprodução não-autorizada desta publicação, no todo ou em parte, constitui violação dos direitos autorais (Lei nº 9.610).

---

Uma proposta de plataforma de software para integração e interoperabilidade de serviços web – WebAgritec : estudo de caso : Diagnose Virtual / Sílvia Maria Fonseca Silveira Massruhá ... [et al.]. – Campinas : Embrapa Informática Agropecuária, 2008.

25 p. : il. — (Boletim de Pesquisa e Desenvolvimento / Embrapa Informática Agropecuária; 19).

ISSN 1677-9266

1. Serviços web. 2. Interoperabilidade. 3. WebAgritec. 4. Sistema especialista. 5. Inteligência artificial. 6. Doença de planta. I. Título. II. Massruhá, Sílvia Maria Fonseca Silveira. III. Série.

CDD - 21<sup>st</sup> ed.  
004.678  
006.3

# Sumário

Resumo.....	5
Abstract.....	6
Introdução.....	7
Material e Métodos.....	9
Resultados e Discussão.....	13
Conclusões.....	21
Referências Bibliográficas.....	22



# Uma proposta de plataforma de software para integração e interoperabilidade de serviços Web - WebAgritec

## Estudo de caso: Diagnose Virtual

---

<sup>1</sup>*Sílvia Maria Fonseca Silveira Massruhá*

<sup>2</sup>*Helano Póvoas de Lima*

<sup>3</sup>*Sílvio Roberto Medeiros Evangelista*

<sup>4</sup>*Gabriel Esteves Marques Pinto*

## Resumo

Este trabalho foi desenvolvido no âmbito do projeto WebAgritec que como objetivo desenvolver uma infra-estrutura, concretizada na forma de um website, que integrará os sistemas da Embrapa Informática Agropecuária. Para tal, é necessário definir a plataforma de software para integração e interoperabilidade destes serviços bem como levantar aplicações e componentes, já desenvolvidos ou em fase de projeto, que possam ser utilizados e, se necessário, adaptá-los para esta plataforma. Um dos principais requisitos deste projeto é interface, mais especificamente, conceito de interface para Web 2.0. Além desse conceito, tecnologias e padrões de projeto próprios para o desenvolvimento de aplicações web e plataformas, bem como comunicação entre seus componentes são estudados no escopo deste trabalho. Outros requisitos não funcionais como desempenho e escalabilidade também devem ser analisados. Nesse contexto, o Openlaszlo foi a tecnologia definida para a infra-estrutura do portal WebAgritec. Neste trabalho, mais especificamente, são apresentados os resultados obtidos na integração do módulo produtor do Diagnose Virtual ao portal WebAgritec visando garantir a interoperabilidade do mesmo.

Palavras-Chave: serviços web, Web 2.0, interface, interoperabilidade, openlaszlo, diagnose virtual.

---

<sup>1</sup> Doutora em Computação Aplicada, Pesquisadora da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: silvia@cnptia.embrapa.br).

<sup>2</sup> Graduado em Ciências da Computação, Analista da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: helano@cnptia.embrapa.br).

<sup>3</sup> Doutor em Engenharia do Software, Analista da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: silvio@cnptia.embrapa.br).

<sup>4</sup> Bacharel em Engenharia da Computação, Consultor da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: gabriel@brainweb.com.br).

# **A software platform for integration and interoperability of Web services - WebAgritec Case Study: Virtual Diagnosis**

---

## **Abstract**

*This work has been developed under the project WebAgritec which aims to develop an framework that will be become available in a website. This project will integrate the systems developed by Embrapa Agricultural Information Technology. Therefore, it is necessary to define the software framework for integration and interoperability of those services as well as acquiring applications and components, have already developed or projected, which can be used and, if necessary, adapt them for this platform. One of the main requirements of this project is interface, more specifically, concept of interface for Web 2.0. In this approach, technologies and design standards to the development of web applications and platforms, as well as communication between its components have been studied. Other non-functional requirements such as performance and scalability must also be examined. In this context, the Openlaszlo technology was specified to implement the framework of the portal WebAgritec. In this work are also presented the results obtained in the integration of the User module of the Virtual Diagnosis under WebAgritec to ensure the interoperability of it.*

*Keywords: web-services, Web 2.0, interface, interoperability, Openlaszlo, virtual diagnosis.*

## Introdução

Este trabalho tem por finalidade apresentar os resultados obtidos no contexto do projeto de pesquisa, cujo objetivo foi definir uma infraestrutura de software para implantação de um portal de integração e interoperabilidade de serviços desenvolvidos pela Embrapa Informática Agropecuária denominado WebAgritec.

A Embrapa Informática Agropecuária tem como missão viabilizar soluções de pesquisa, desenvolvimento e inovação em tecnologia de informação para a sustentabilidade da agricultura, em benefício da sociedade brasileira (Embrapa Informática Agropecuária, 2008).

Ao longo dos anos foram desenvolvidos diversos sistemas especialistas que visam atender a demanda de certos nichos e entidades relacionadas ao negócio agrícola. Dentre eles pode-se destacar o sistema de monitoramento agrometeorológico ([www.agritempo.gov.br](http://www.agritempo.gov.br)) (Romani et al., 2003), o sistema de diagnose virtual de doenças de plantas (<http://diagnose2.cnptia.embrapa.br/diagnose/>) (Massruhá, 2003; Massruhá et al., 2004, 2005, 2006, 2007) e o sistema de previsão de safra de soja (Assad et al., 2007) bem como o projeto do sistema de recomendação para adubação.

O projeto WebAgritec tem como objetivo desenvolver uma infraestrutura, concretizada na forma de um *website*, que integrará esses sistemas desenvolvidos pela Embrapa Informática Agropecuária. As aplicações, que foram definidas durante a fase inicial do projeto, são:

- Módulo de Previsão do tempo
- Módulo de Zoneamento Agrícola
- Módulo de Monitoramento de Plantio
- Módulo Produtor do Diagnose Virtual
- Módulo de Cultivares
- Módulo de Adubação
- Módulo de Videoteca
- Módulo de Notícias

Para tal, foram levantados alguns pontos essenciais para o projeto:

- definir a infraestrutura de software para implantação de uma plataforma de integração e interoperabilidade desses serviços;
- levantar aplicações e componentes, já desenvolvidos ou em fase de projeto, que possam ser utilizados e, se necessário, adaptá-los para essa plataforma;

- investigar e definir requisitos de interface Web 2.0<sup>1</sup>;
- estudar e definir tecnologias e padrões de projeto próprios para o desenvolvimento de aplicações web e plataformas, bem como comunicação entre seus componentes, tais como *portlets*, serviços web (*web-services*), ajax, servidores de aplicação dentre outros;
- implantar protótipo de servidor de componentes e com alguns componentes pré-definidos;
- especificar requisitos não funcionais como desempenho e escalabilidade.

Nesse contexto, o OpenLaszlo foi a tecnologia definida para a infraestrutura do portal WebAgritec. OpenLaszlo (<http://www.openlaszlo.org>) é a implementação de código aberto da tecnologia Laszlo, da empresa Laszlo Systems (2006). Seu principal objetivo é facilitar a criação de interfaces *web* ricas, com características cinemáticas e dinâmicas, e é compatível com os principais navegadores da Internet. O OpenLaszlo roda aplicações escritas em XML (eXtended Markup Language) e JavaScript, e permite utilizar as tecnologias *Flash* e DHTML (*Dynamic HTML*).

A tecnologia OpenLaszlo não é especificamente uma plataforma para a construção de portais, mas, dada sua flexibilidade, pode ser usada de maneira a proporcionar uma ótima experiência de navegação ao usuário do sistema.

Um primeiro protótipo do Webagrítec foi gerado para validar a tecnologia OpenLaszlo onde as aplicações 1,2,3,5,6,7 e 8 foram acopladas. Essas aplicações foram totalmente reescritas na nova tecnologia utilizando uma arquitetura cliente servidor simples, onde a parte cliente (apresentação) é representada por uma aplicação OpenLaszlo e a infraestrutura (*backend*) por uma aplicação *Java Web*, formada por *Servlets* ([www.sun.com/java/](http://www.sun.com/java/)), que realizam as consultas necessárias à base de dados do Agritempo. A reescrita dessas aplicações não faz parte do escopo deste trabalho.

A aplicação 4, Diagnose Virtual, por ser um sistema especialista complexo, demanda uma infraestrutura dedicada e envolve dois diferentes perfis de usuário: Especialistas e Produtores. Os especialistas alimentam o sistema com uma vasta base de conhecimento sobre diferentes culturas e os produtores podem consultar o sistema a fim de diagnosticar desordens em seus cultivos.

O WebAgrítec contém apenas o Módulo Produtor do Diagnose Virtual que foi desenvolvido originalmente com a tecnologia de Java/JSPs (*Java Server Pages*) ([www.sun.com/java/](http://www.sun.com/java/)) e migrado para a tecnologia OpenLaszlo. Entretanto, a integração da aplicação 4 no portal não foi trivial. A hipótese desse trabalho foi que a tecnologia OpenLaszlo seria adequada para integrar o módulo produtor do Diagnose Virtual ao portal e garantir a interoperabilidade do mesmo.

Para validar a viabilidade de uma interface OpenLaszlo para o Diagnose Virtual foi desenvolvido um protótipo do sistema nessa nova tecnologia. Na seção 2 serão apresentados os materiais e métodos bem com a metodologia utilizada para o desenvolvimento deste trabalho. Na seção 3 são discutidos os principais resultados obtidos. Finalmente, na seção 4 são apresentados as principais conclusões bem como sugestões para trabalhos futuros.

<sup>1</sup> Web 2.0 é a mudança para uma internet como plataforma, e um entendimento das regras para obter sucesso nessa nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a "inteligência coletiva".

## Material e Métodos

Nessa seção está descrita a arquitetura proposta para a integração de serviços *web*, bem como o material e os métodos utilizados neste trabalho. Para fins didáticos, antes de detalhar a arquitetura proposta neste trabalho, alguns conceitos básicos são apresentados.

Este projeto de pesquisa foi dividido nas seguintes etapas:

- Levantamento de aplicações e componentes já desenvolvidos na Embrapa Informática Agropecuária, em fase de projeto ou no plano hipotético que possam fazer parte do portal.
- Estudo de tecnologias para o desenvolvimento da infraestrutura do portal.
- Estudo de tecnologia para interoperabilidade.
- Elaboração do modelo de núcleo do portal.
- Elaboração do modelo de componentes.
- Validação da arquitetura proposta, e
- Capacitação de técnicos nas tecnologias utilizadas.

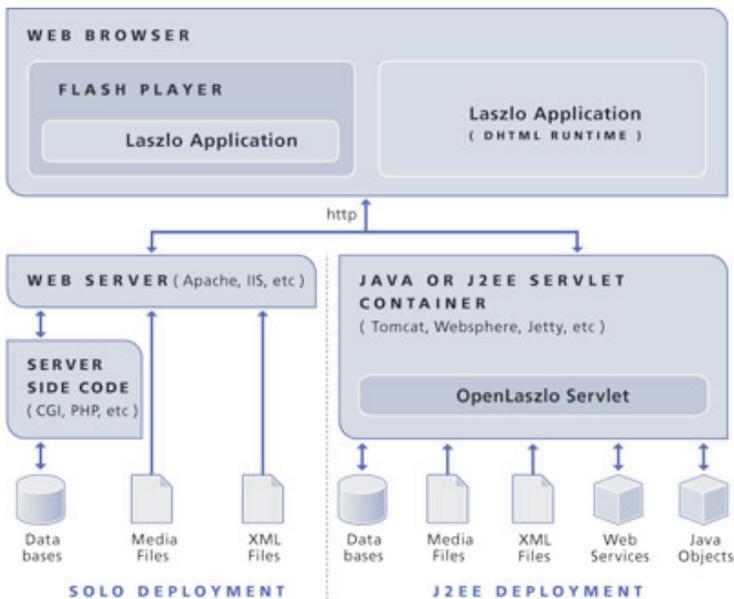
No início das atividades do plano de pesquisa, a situação era a seguinte:

- As aplicações desenvolvidas na Embrapa Informática Agropecuária participantes do portal WebAgritec previamente selecionadas.
- Necessidade de entendimento do conceito de Web 2.0 onde a Web é definida como plataforma e os softwares como serviços.
- Intensificação dos conceitos de modularidade e reusabilidade no processo evolutivo de desenvolvimento de *software* para Internet.
- A proposta da *Web 2.0* das aplicações para Internet ter um uso semelhante ao de aplicativos para desktop, frequentemente fazendo uso de uma combinação de várias tecnologias (como *Web services* e *AJAX - Asynchronous Javascript And XML*) para que tornem a experiência do usuário mais rica, com interfaces mais rápidas e muito fáceis de usar.
- O conteúdo dos sítios também sofreu um enorme impacto com a *Web 2.0*, dando ao usuário a possibilidade de participar, geralmente gerando e organizando as informações. Tecnologias como *blogs* e *wikis* contribuíram para essa massificação.
- O *OpenLaszlo* como uma alternativa para *Web 2.0*, devido a sua facilidade para criação de interfaces web ricas, com características cinemáticas e dinâmicas e compatível com os principais navegadores da Internet.
- O *framework* para o desenvolvimento da infraestrutura do portal previamente definido para utilizar a tecnologias para *Web 2.0*, possivelmente, o *Openlaszlo*.

Frente ao cenário apresentado, focou-se em estudar as novas tecnologias e definições já estabelecidas e desenvolver um protótipo de componente para validar a eficácia das escolhas realizadas.

O OpenLaszlo é uma plataforma de desenvolvimento que serve para criar aplicações em *RIA* (*Rich Internet Applications*). Trata-se de um ambiente completo, gratuito e de código-fonte aberto que é compatível com os principais navegadores da Internet.

Os aplicativos OpenLaszlo são construídos através de linguagem declarativa LZX, que mescla XML (*eXtended Markup Language*) (Halevy, 1999; Widom, 1999) e JavaScripts (<http://javascript.internet.com>). Além de rodar aplicações escritas em XML e JavaScript, o OpenLaszlo pode ser compilado para Flash ou DHTML (*Dynamic HTML*). Na Fig. 1 representa uma visão geral da arquitetura OpenLaszlo retirada do *site* do Openlaszlo (<http://www.openlaszlo.org/architecture>).



Fonte: Laszlo Systems (2006).

**Fig. 1.** Arquitetura do OpenLaszlo

As aplicações cliente-servidor Laszlo podem ser distribuídas de duas formas: Sem Servidor (*Serverless*) e Cliente-Servidor (*Proxied*). Na primeira, o arquivo do código-fonte com a extensão *.LZX* é compilado em um arquivo *Flash .SWF* e disponibilizado para *download*. Na última, o cliente faz a requisição a um arquivo com extensão *.LZX* presente no servidor e a compilação do arquivo é feita sob demanda, podendo gerar tanto um arquivo *Flash* com extensão *.SWF* como uma aplicação *DHTML*.

Essa arquitetura cliente-servidor é a mais usual e foi a escolhida para o uso no

WebAgritec. Na Fig. 2 é apresentado, detalhadamente, o processo de execução de uma aplicação OpenLaszlo retirada do site do OpenLaszlo (<http://www.openlaszlo.org/developmentcycle>).

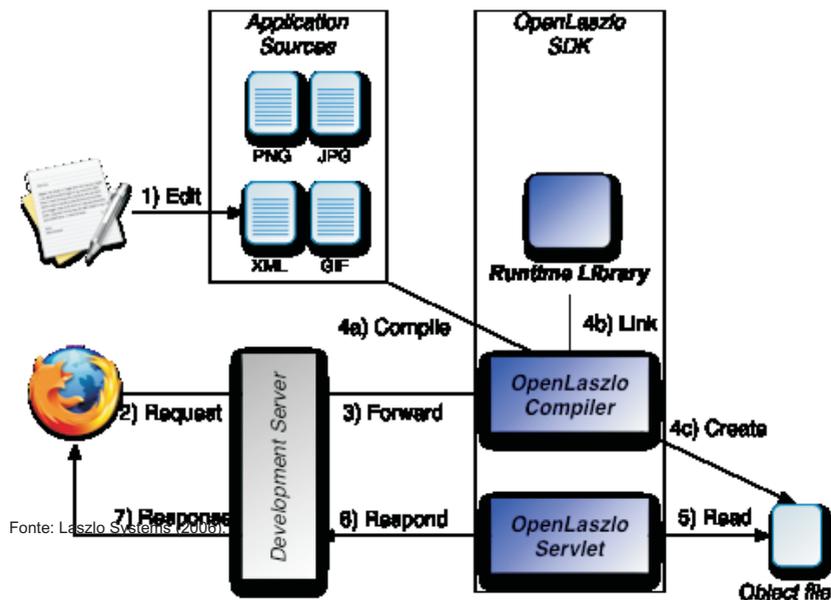


Fig. 2. Execução de uma aplicação OpenLaszlo

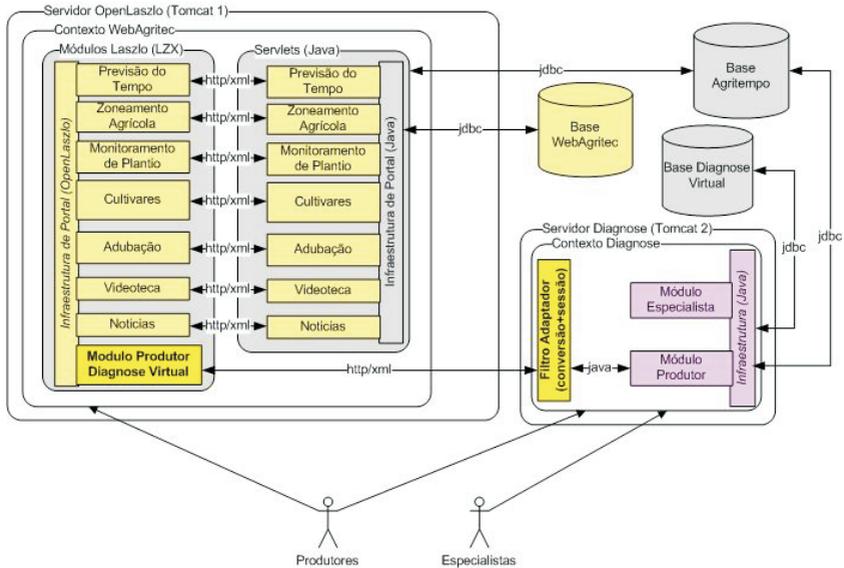
Ao analisar a tecnologia OpenLaszlo e os 8 módulos que integram a arquitetura do WebAgritec (Módulo de Previsão do tempo, Módulo de Zoneamento Agrícola, Módulo de Monitoramento de Plantio, Módulo Produtor do Diagnóstico Virtual, Módulo de Cultivares, Módulo de Adubação, Módulo de Videoteca e Módulo de Notícias), as subdividimos em dois conjuntos distintos: as aplicações simples e as complexas.

O Módulo Produtor do Diagnóstico Virtual se encaixou nesse último conjunto de aplicações e as demais aplicações, foram totalmente reescritas na nova tecnologia utilizando uma arquitetura cliente servidor simples, onde a parte cliente (apresentação) é representada por uma aplicação OpenLaszlo e a infraestrutura (*backend*) por uma aplicação *Java Web*, formada por *Servlets*, que realizam as consultas necessárias à base de dados do Agritempo.

Na Fig. 3 pode ser vista a arquitetura geral do WebAgritec e sua integração com todos os sistemas legados envolvidos no processo. Entretanto, a reescrita das aplicações mais simples que acessam a base de dados do Agritempo não fazem parte do escopo deste trabalho.

A aplicação Diagnóstico Virtual, por ser um sistema especialista complexo, demanda uma infraestrutura dedicada e envolve dois diferentes perfis de usuário:

Especialistas e Produtores. Os especialistas alimentam o sistema com uma vasta base de conhecimento sobre diferentes culturas, e os produtores podem consultar o sistema a fim de diagnosticar desordens em seus cultivos. O WebAgritec contém apenas o Módulo Produtor do Diagnose Virtual que foi desenvolvido originalmente com a tecnologia de Java/JSPs (Java Server Pages) ([www.sun.com/java/](http://www.sun.com/java/)) e migrado para a tecnologia OpenLaszlo (<http://www.openlaszlo.org>), podendo assim ser integrado ao portal.



**Fig. 3.** Arquitetura do WebAgritec e Integrações

Nesse ponto, sabendo de antemão quais as aplicações e os tipos de dados que a interface tem que enviar para o mecanismo de inferência, foi eleito o *XML* como tecnologia de interoperabilidade e comunicação entre os sistemas. Desse modo, todas as requisições feitas entre as aplicações têm como resposta um *XML* precisamente especificado. O *XML* foi escolhido por ser independente de plataforma de desenvolvimento e suportado pelas principais linguagens de programação.

Na Fig. 3 também pode ser identificada a principal diferença entre o Módulo Produtor do Diagnose Virtual e os demais módulos. Notadamente as requisições são realizadas para um sistema que se encontra em um servidor externo ao do OpenLaszlo, enquanto que as outras aplicações, por serem simples, se resumem a uma requisição e uma resposta entre a aplicação OpenLaszlo e o próprio servidor OpenLaszlo. Outra diferença importante entre os módulos é a natureza interativa do Módulo Produtor, no qual o usuário é requisitado a responder inúmeras perguntas, dependendo de suas respostas anteriores.

Outro ponto a ser ressaltado é que na arquitetura do WebAgritec também está prevista uma base própria para armazenar os dados referentes aos serviços exclusivos do portal e que não estão contemplados nas bases de dados do

Diagnose Virtual e do Agritempo. Entretanto, esses serviços não estão ainda totalmente especificados e, portanto, não serão detalhados neste trabalho.

Em resumo, a investigação neste trabalho foi focada em resolver a integração entre a aplicação Diagnose Virtual e a infraestrutura OpenLaszlo previamente definida.

## Resultados e Discussão

Para validar a viabilidade de uma interface OpenLaszlo para o Diagnose Virtual foi desenvolvido um protótipo do sistema nesta nova tecnologia. Os resultados obtidos na validação desse protótipo podem ser discutidos sob cinco aspectos diferentes: interface do protótipo, atualização do módulo produtor, comunicação entre aplicações, integração do módulo produtor ao portal do WebAgritec e comparação de tecnologias.

### Interface do protótipo

O potencial cinemático da tecnologia (Laszlo Systems, 2006) foi aproveitado ao máximo no desenvolvimento da nova interface do sistema. Nesse protótipo, o sistema de diagnóstico foi decomposto em sete etapas:

- Acesso ao sistema (Fig. 4)
- Identificação do produtor (Fig. 5)
- Identificação do cenário (Fig. 6)
- Identificação dos sintomas e severidades (Fig. 7)
- Identificação da desordem (Fig. 8)
- Identificação das manifestações (Fig. 9)
- Relatório do diagnóstico (Fig. 10)

Cada uma das etapas originalmente é representada por uma tela diferente, em um modelo tradicional de requisição/resposta de páginas da *Internet*. A nova interface do Módulo Produtor do Diagnose Virtual segue um modelo de tela única e os preceitos da *Web 2.0* com apoio de tecnologias inovadoras.

Apesar de conseguir produzir resultados visuais muito interessantes, o desenvolvimento de aplicações e algoritmos mais complexos se mostrou muito trabalhoso e pouco produtivo. Muitos dos problemas e entraves encontrados durante o projeto de pesquisa foram devido à tecnologia adotada, que apresentou muitos erros e pouca flexibilidade (*OpenLaszlo Server 4.0.5*). Foram encontrados problemas desde a 'renderização' de componentes básicos (como uma caixa de seleção - *html select*) até a execução de requisições mais elaboradas durante a comunicação com servidores legados, além de alguns problemas relacionados à *performance*.



Fig. 4. Tela principal do WebAgritec



Fig. 5. Identificação do Produtor



Fig. 6. Identificação do Cenário



Fig. 7. Identificação dos Sintomas e Severidades



Fig. 8. Identificação da Desordem



Fig. 9. Identificação das Manifestações



Fig. 10. Relatório do Diagnóstico

## Atualização do Módulo Produtor

No decorrer do trabalho, algumas funcionalidades do Módulo Produtor do Diagnose Virtual foram alteradas para satisfazer as necessidades de uso do sistema:

- Inclusão de novos campos:
  - Data observação: com componente para escolha de datas.
  - Escolha de cidades usando recurso de 'autocompletar' com texto parcial e estado.
- Validações de campos:
  - Nome completo: pelo menos dois nomes separados com espaço.
  - Formatação de email.
  - Data observação: obrigatório.
  - Estado: obrigatório.
  - Cidade: obrigatório.
  - Ramo: obrigatório.
  - Tipo: obrigatório.
  - Fase: obrigatório.
  - Tamanho da produção: numérico.
  - Intervalos iniciais e finais das perguntas: numérico.
- Tratamento de diferentes tipos de imagens (jpg, gif e png): Hoje o sistema considera todas as imagens como 'jpg'.
- Reinício de diagnóstico: Permite que ao final de um diagnóstico o usuário possa realizar um novo diagnóstico, refiná-lo ainda mais ou imprimi-lo.
- Integração com base de dados Agritempo: Para busca de cidades, obtenção de dados de intervalos de temperatura e umidade da data de observação do caso.
- Reimplementação de mecanismos de segurança: Permite que as consultas realizadas pelo Módulo Produtor Laszlo sejam filtradas para evitar alguma brecha de segurança.

## Comunicação entre as aplicações

Conforme a Fig. 3, a arquitetura WebAgritec contempla 2 servidores *Tomcats* (<http://tomcat.apache.org>) envolvidos na arquitetura do sistema: *Tomcat* do OpenLaszlo e *Tomcat* do Diagnose. A nova versão da interface do Módulo Produtor do Diagnose Virtual foi implantada no *Tomcat* do OpenLaszlo e a lógica de negócio de diagnósticos ficou no *Tomcat* do Diagnose. Dado esse cenário, o Módulo Produtor faz suas requisições partindo do *Tomcat* 1 (OpenLaszlo) para o *Tomcat* 2 (Diagnose). As respostas são todas em *XML*.

A aplicação do Diagnose Virtual (*backend no servidor Tomcat 2*) precisou ser adaptada para receber as requisições em formato diferente do convencional e gerar respostas em *XML*, antes formatadas em páginas *HTML*. Na fase de protótipo essa adequação foi implementada criando-se uma versão alternativa da camada de apresentação e

controle do Diagnose Virtual original. A versão final dessa adaptação é baseada em *Servlet Filters* e *JSPs* (*Java Server Pages*) para a geração de *XMLs* como resposta conforme ilustrado da Fig. 3. Dependendo da origem da requisição, a resposta é renderizada por um conjunto específico de *JSPs*, mas a lógica de negócios da aplicação permanece a mesma. Observe o exemplo de código no Trecho de Código 1 e Trecho de Código 2 da classe *CasoAction.class*:

```
<action path="/caso" type="org.springframework.web.struts.DelegatingActionProxy"
  name="casoForm" scope="request" parameter="method" validate="false">
  <forward name="list" path="/casoList.jsp"/>
  <forward name="edit" path="/casoForm.jsp"/>
  <forward name="login" path="/login.jsp"/>
  <forward name="edit2" path="/casoForm2.jsp"/>
  <forward name="edit3" path="/casoForm3.jsp"/>
  <forward name="edit4" path="/casoForm4.jsp"/>
  <forward name="edit5" path="/casoForm5.jsp"/>
</action>
<action path="/casoLaszlo" type="org.springframework.web.struts.DelegatingActionProxy"
  name="casoForm" scope="request" parameter="method" validate="false">
  <forward name="list" path="/casoList.jsp"/>
  <forward name="edit" path="/casoXml.jsp"/>
  <forward name="login" path="/login.jsp"/>
  <forward name="edit2" path="/casoXml2.jsp"/>
  <forward name="edit3" path="/casoXml3.jsp"/>
  <forward name="edit4" path="/casoXml4.jsp"/>
  <forward name="edit5" path="/casoXml5.jsp"/>
</action>
```

### Trecho de Código 1. Configuração de recebimento de requisições Laszlo em *struts-config.xml*

```
<bean name="/casoLaszlo" class="br.embrapa.cnptia.diagnose.web.CasoAction"
  singleton="false">
  <property name="casoBO"><ref bean="casoBO"/></property>
  <property name="cidadeBO"><ref bean="cidadeBO"/></property>
  <property name="problemaBO"><ref bean="problemaBO"/></property>
  <property name="faseBO"><ref bean="faseBO"/></property>
  <property name="severidadeBO"><ref bean="severidadeBO"/></property>
  <property name="categoriaBO"><ref bean="categoriaBO"/></property>
  <property name="usuarioBO"><ref bean="usuarioBO"/></property>
  <property name="manifestacaoBO"><ref bean="manifestacaoBO"/></property>
  <property name="desordemBO"><ref bean="desordemBO"/></property>
  <property name="tratamentoBO"><ref bean="tratamentoBO"/></property>
</bean>
```

### Trecho de Código 2. Configuração de classes de negócio para as requisições Laszlo em *action-servlet.xml*

Dado que as aplicações se encontram em *Tomcats* diferentes e separados fisicamente, as sessões de usuário cada *Tomcat* também são diferentes. A solução para esse problema pode ser feita de diferentes maneiras, dentre elas:

- Persistência dos dados de sessão em meio persistente diferente do tratamento padrão do *Tomcat* (por exemplo, banco de dados) e identificação por uma chave (id).

- Reenvio de parâmetros acumulados de requisição a cada interação
- Configurar a replicação de sessões entre os *Tomcat*
- Disponibilizar o Portal WebAgritec e o Sistema Diagnose Virtual no mesmo *Tomcat*.

Na primeira etapa do projeto foi construído um protótipo com base na opção 2. Essa escolha se mostrou inviável dado o acúmulo de uma quantidade indefinida de informações na aplicação OpenLaszlo. As opções 3 e 4 foram descartadas, pois quebram a autonomia das aplicações terem as configurações de seus *Tomcats* independentes e de fácil implantação. Após muitas tentativas, notou-se que o identificador da sessão de navegação do usuário (*JSESSIONID*) poderia funcionar corretamente caso fosse adicionado manualmente um *Cookie* no navegador do usuário. Essa alteração foi feita no filtro *LaszloFilter* e assumiu a responsabilidade de gerenciar a sessão de navegação do usuário, bem como o registro (autenticação) desse usuário no sistema (veja exemplo no Trecho de Código 3):

```
public class LaszloFilter implements Filter {
    public void init(FilterConfig arg0) throws ServletException {}
    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws IOException, ServletException {
        HttpServletRequest req = (HttpServletRequest) request;
        HttpServletResponse res = (HttpServletResponse) response;
        HttpSession session = req.getSession();

        session = req.getSession();
        req.getSession().setAttribute("idSessao", session.getId());
        res.addCookie(new Cookie("JSESSIONID", session.getId()));
        UsuarioTO usuario = (UsuarioTO) session.getAttribute("USUARIO");
        if (req.getParameter("tipo") != null &&
            req.getParameter("tipo").equals("produtor")) {
            String nome = req.getParameter("nome");
            String email = req.getParameter("email");
            usuario = new ProdutorTO();
            usuario.setLogin("");
            usuario.setSenha("");
            usuario.setNome(nome);
            usuario.setEmail(email);
            session.setAttribute("USUARIO", usuario);
        }
        chain.doFilter(req, res);
        res.addCookie(new Cookie("JSESSIONID", session.getId()));
    }
    public void destroy() {}
}
```

**Trecho de Código 3.** *LaszloFilter*: responsável por manter a sessão de navegação entre os diferentes *Tomcats*.

## Integração do módulo produtor com o portal WebAgritec

O desafio seguinte foi a integração do módulo produtor ao portal WebAgritec. Durante essa etapa, o principal objetivo foi consolidar o protótipo do Módulo Produtor do WebAgritec em uma versão estável e funcional, com todas as características necessárias para seu pleno uso.

Dentre as atividades de integração realizadas pode-se destacar:

- Inclusão do Módulo Produtor do Diagnose Virtual no WebAgritec (veja exemplo da alteração no Trecho de Código 4):

```
<include href="WinVideo/WinVideo.lzx"/>
<include href="WinZoneamento/WinZoneamento.lzx"/>
<include href="WinCultivares/WinCultivares.lzx"/>
<include href="WinMonitoramento/WinMonitoramento.lzx"/>
<include href="WinDiagnose/WinDiagnose.lzx"/>
<include href="resources.lzx"/>
```

Trecho de Código 4. lps-4.0.5/my-apps/agritest/TelaPrincipal.lzx

- Inclusão do ícone do Módulo Produtor do Diagnose Virtual no WebAgritec (veja exemplo da alteração no Trecho de Código 5):

```
<Icône id="icoMonitoramento" img="imgIcôneAcompanhamentoPlantio" text="Monitoramento"
janela="{winMonitoramento}" valign="bottom"/>
<Icône id="icoDiagnose" img="imgIcôneDiagnose" text="Diagnose Virtual"
valign="bottom" janela="{winDiagnose}"/>
<Icône id="icoCultivares" img="imgIcôneCultivares" text="Cultivares"
janela="{winCultivares}" valign="bottom"/>
<Icône id="icoAdubo" img="imgIcôneAdubo" text="Aduba&#231;&#227;o" janela="{winAdubo}"
valign="bottom"/>
<Icône id="icoVideo" img="imgIcôneVideo" text="Videoteca" janela="{winVideo}"
valign="bottom"/>
```

Trecho de Código 5. lps-4.0.5/my-apps/agritest/TelaPrincipal.lzx

- Inclusão da janela do Módulo Produtor do Diagnose Virtual no WebAgritec (veja exemplo da alteração no Trecho de Código 6):

```
<WinMonitoramento id="winMonitoramento" x="{once{(immediateparent.width-width)/2}"
y="{once{(immediateparent.height-height)/2}" icône="{icoMonitoramento}"
minimizada="true"/>
<WinCultivares id="winCultivares" x="{once{(immediateparent.width-width)/2}"
y="{once{(immediateparent.height-height)/2}" icône="{icoCultivares}" minimizada="true"/>
<WinDiagnose id="winDiagnose" x="{once{(immediateparent.width-width)/2}"
y="{once{(immediateparent.height-height)/2}" icône="{icoDiagnose}"
minimizada="true" />
```

Trecho de Código 6. lps-4.0.5/my-apps/agritest/TelaPrincipal.lzx

- Configuração do caminho das requisições para o servidor Diagnose Virtual (*backend*) (veja exemplo da alteração no Trecho de Código 7):

```
<script>
  SERVLET_BASE = "/lps-4.0.5";
  DIAGNOSE_BASE = "http://localhost:7070/diagnose";
</script>
```

**Trecho de Código 7.** lps-4.0.5/my-apps/agritest/index.lzx

- Configuração de *timeout* do servidor OpenLaszlo (veja exemplo da alteração no Trecho de Código 8 e 9): O padrão de timeout do servidor foi insuficiente para o diagnóstico de um caso, portanto é necessário alterar esse valor padrão para um novo valor. Possivelmente, no futuro, com otimizações do mecanismo de diagnóstico, esse tempo possa ser reduzido novamente.

```
# Timeout for back-end http requests
http.backendTimeout=120000
http.backendConnectionTimeout=120000
```

**Trecho de Código 8.** lps-4.0.5/WEB-INF/lps/config/lps.properties

```
<canvas title="Webagrítec" dataLoadTimeout="120000" mediaLoadTimeout="120000">
  <include href="TelaPrincipal.lzx"/>
```

**Trecho de Código 9.** lps-4.0.5/my-apps/agritest/index.lzx

Além das alterações acima listadas é necessário integrar o Módulo Produtor do Diagnose Virtual para o ambiente de produção do WebAgritec. A cópia deve

**Tabela 1.** Quadro Comparativo das tecnologias.

	Módulo Produtor	
	Original / HTML	LASZLO / Flash
<b>Conteúdo</b>	HTML	Flash / SWF / DHTML
Conteúdo dinâmico	JSP	LASZLO / JavaScript
Apresentação	Páginas	Janela
Requisições assíncronas	AJAX	LASZLO / Flash
Backend	No mesmo Servidor	Servidor remoto
Controle de sessão	Feito pelo TOMCAT	Gerenciando manualmente via Cookies (alternativa)

ser feita para dentro do diretório lps-4.0.5/my-apps/agritest tendo como resultado o diretório lps-4.0.5/my-apps/agritest/WinDiagnose contendo todo o conteúdo do módulo.

## Quadro Comparativo de tecnologias

Na tabela 1 é apresentado um quadro comparativo das tecnologias que foram utilizadas na versão original do módulo produtor do diagnose virtual e da tecnologia investigada nessa proposta.

O resultado final alcançado, neste projeto de pesquisa, foi satisfatório tanto do ponto de vista de arquitetura da aplicação quanto do ponto de vista da aplicação final apesar dos problemas encontrados no OpenLaszlo já citados nesta seção. Dentre os principais resultados positivos obtidos pode-se destacar:

- Estudo e análise da tecnologia OpenLaszlo como base para a infraestrutura do portal e desenvolvimento dos componentes; estudo e definição de arquivos XML para a intercomunicação entre as aplicações.
- Validação de conceitos de Web 2.0 tais como a web como plataforma e os softwares como serviços.
- Desenvolvimento de um protótipo funcional do Módulo Produtor do Diagnose Virtual utilizando a tecnologia OpenLaszlo e direcionamento para solucionar os problemas e desafios encontrados para o desenvolvimento das próximas etapas.
- Solução para comunicação entre Módulo Produtor na tecnologia OpenLaszlo e Diagnose Virtual sem necessidade de reimplementação de códigos.
- Integração com sistema legado Agritempo (para obtenção de dados de estados, cidades, temperaturas, umidade, etc.).
- Inclusão de novos campos e validações essenciais para o ambiente de produção.
- Versão funcional de aplicação para execução em ambiente de produção.

## Conclusões

O cerne deste trabalho de pesquisa foi a obtenção de uma versão funcional do Módulo Produtor do Diagnose Virtual dentro do portal WebAgritec validando a tecnologia OpenLaszlo. O uso do OpenLaszlo, pelo menos até a versão utilizada para o WebAgritec, pode não ser o mais recomendado para o projeto

devido a maturidade atual da tecnologia, quantidade de erros existentes, dificuldade na construção de aplicações mais extensas, consumo excessivo de processador durante a sua execução, dentre outros motivos menores. Tais fatores são agravados quando a tecnologia é comparada a outras tecnologias web, baseadas em html, javascript/ajax, que são amplamente documentadas, robustas e de implementação mais simples. Dentre os próximos passos no escopo deste projeto de pesquisa, recomenda-se como trabalhos futuros:

- Disponibilizar protótipo para testes por um grupo de controle;
- Avaliar a continuidade do projeto na tecnologia OpenLaszlo ou substituição da tecnologia caso novas versões não supra os problemas citados neste trabalho;
- Implementar versão definitiva do portal, levando em consideração requisitos não funcionais como desempenho, usabilidade, autenticação unificada, manutenção e modularidade do projeto.

## Referências Bibliográficas

ASSAD, E. D.; MARIN, F. R.; EVANGELISTA, S. R.; PILAU, F. G.; FARIAS, J. R. B.; PINTO, H. S.; ZULLO JUNIOR, J. Sistema de previsão de safra de soja para o Brasil. *Pesq. agropec. bras.*, Brasília, DF, v. 42, n. 5, p. 615-625, maio 2007. Disponível em: <<http://www.repdigital.cnptia.embrapa.br/bitstream/CNPSO/27369/1/42n05a02.pdf>>. Acesso em: 4 nov. 2008.

EMBRAPA INFORMÁTICA AGROPECUÁRIA. *IV Plano Diretor da Embrapa Informática Agropecuária 2008-2011*. Campinas, 2008. (Embrapa Informática Agropecuária. Documentos). No prelo.

HALEVY, A. Y. More on data management for XML. [Seattle]: University of Washington, 1999. Disponível em: <<http://www.cs.washington.edu/homes/alon/widom-response.html>>. Acesso em: 4 nov. 2008.

LASZLO SYSTEMS. *OpenLaszlo: an open architecture framework for advanced Ajax applications*. San Mateo, CA, 2006. 19 p. (Technology white paper). 19 p. Disponível em: <<http://www.openlaszlo.org/whitepaper/LaszloWhitePaper.pdf>>. Acesso em: 4 nov. 2008.

MASSRUHÁ, S. M. F. S. *Uma teoria de coberturas nebulosas para diagnóstico, investigação e tratamento*. 2003. 251 p. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos.

MASSRUHÁ, S. M. F. S.; DUTRA, J. P.; CRUZ, S. A. B.; SANDRI, S.; WAINER, J.; MORANDI, M. An objected oriented framework for virtual diagnosis. In: BIENNIAL CONFERENCE OF THE EUROPEAN FEDERATION OF IT IN AGRICULTURE – EFITA 2007, 6<sup>th</sup>, 2007, Glasgow. EFITA/WCCA 2007 - 6<sup>th</sup> Biennial Conference of the European Federation of IT in Agriculture [proceedings]. Glasgow: Glasgow Caledonian University, 2007.

MASSRUHÁ, S. M. F. S.; SANDRI, S.; WAINER, J. Ordering manifestations for investigation in incomplete diagnosis. Information processing and management of uncertainty in knowledge based systems. In: INTERNATIONAL CONFERENCE IPMU, 10<sup>th</sup>, 2004, Perugia. 10<sup>th</sup> International Conference IPMU 2004 [proceedings]. Perugia, 2004. p. 1153-1160.

MASSRUHÁ, S. M. F. S.; SANDRI, S.; WAINER, J.; MORANDI, M. A fuzzy and abductive framework for clinical problem solving tasks of treatment in agriculture. In: CONFERENCE OF THE ASIÁN FEDERATION FOR INFORMATION TECHNOLOGY IN AGRICULTURE, 5<sup>th</sup>, 2006, Bangalore. Fifth Conference of the Asian Federation for Information Technology in Agriculture [proceedings]. Bangalore, 2006.

MASSRUHÁ, S. M. F. S.; SANDRI, S.; WAINER, J.; MORANDI, M. An integrated framework for clinical problem solving in agriculture. In: EFITA/WCCA 2005, Vila Real; CONGRESS ON IT IN AGRICULTURE, 2005, Vila Real. *Proceedings...* [Vila Real]: Efitas, 2005. p. 1400-1407.



---

*Informática Agropecuária*

Ministério da  
Agricultura, Pecuária  
e Abastecimento

