

Ambiente de desenvolvimento para o projeto Banco de Dados de Pragas Quarentenárias¹

Tiago Augusto Pimenta²

Isaque Vacari³

Marcos Cezar Visoli³

Carlos Alberto Alves Meira³

Anderson Carlos Bueno dos Santos²

Bruno Franciscon Mazzotti²

O Projeto Análise de Risco de Pragas (ARP) tem o objetivo de auxiliar a Defesa Sanitária Vegetal do Brasil no controle e monitoramento de produtos com risco potencial de introdução de pragas quarentenárias no país durante sua importação. Parte desse projeto inclui a catalogação e modelagem de um banco de dados e o desenvolvimento de um sistema de informação (SI) de pragas quarentenárias, chamado de BD-Pragas.

A etapa da definição da plataforma computacional para o desenvolvimento do sistema deve ser realizada criteriosamente, com base na arquitetura adotada para o sistema e nas tecnologias mais modernas possíveis, considerando robustez, qualidade, quantidade de projetos que estão utilizando, e licenças que as caracterizem como tecnologias livres.

¹ *Agradecimentos ao CNPq pelo auxílio financeiro. Projeto Banco de Dados sobre Pragas Quarentenárias Associadas às Culturas Relevantes para o Agronegócio Brasileiro e a Defesa Fitossanitária no País (número 558949/2009-6, chamada 2 do edital 32/2009).*

² *Bolsista do CNPq; tiagoap@cnptia.embrapa.br; andersoncbs@cnptia.embrapa.br; brunofm@cnptia.embrapa.br*

³ *Embrapa Informática Agropecuária; isaque@cnptia.embrapa.br; visoli@cnptia.embrapa.br; carlos@cnptia.embrapa.br*

A arquitetura de sistema adotada para o BD-Pragas é a Model-View-Controller - MVC (ALMEIDA, 2006) cujo diagrama está ilustrado na Figura 1. A plataforma de software é baseada na tecnologia Java, utilizada de forma consolidada nos projetos da Embrapa Informática Agropecuária. O Sistema Gerenciador de Banco de Dados (SGBD) adotado é o PostgreSQL, que além de robusto e estável, provê boa performance e escalabilidade (FERREIRA, 2004).

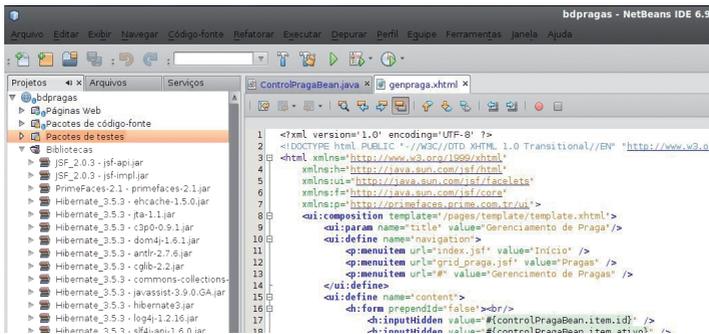


Figura 1. Camadas do BD-Pragas.

Para a camada de persistência, optou-se pelo uso do *framework* Hibernate, que encapsula o sistema de transações do banco de dados com a filosofia de abstrair do desenvolvedor a execução de consultas SQL fazendo uso da *Criteria Query API*. Trata-se de um conjunto de classes para elaborar consultas ao banco de dados com código Java que são traduzidas para o dialeto SQL do SGBD pelo Hibernate. No entanto, ainda é possível utilizar a *Hibernate Query Language* (HQL), que é uma extensão da linguagem SQL que faz referência a objetos em vez de tabelas (LINHARES, 2005).

Concluiu-se que para a camada de interface, o *framework* a ser adotado seria o Mojarra JavaServer Faces (JSF) com componentes PrimeFaces para apresentação, encaixando-se perfeitamente na arquitetura MVC. Porém, a maioria dos desenvolvedores web habituados com linguagens *action-like* têm dificuldades ao desenvolver em JSF, adquirindo uma série de maus hábitos. O estudo adequado da tecnologia e do ciclo de vida de uma aplicação JSF pode evitar esses

problemas. Gonçalves (2010) explica o funcionamento do JSF e seu ciclo composto por seis fases que manipulam a interface.

Para garantir a continuidade do SI e a integridade de suas regras de negócio é importante realizar testes sobre blocos de códigos isolados, denominados testes unitários. Esses testes são fundamentais para sistemas flexíveis como o BD-Pragas e podem ser aplicados utilizando o *framework* JUnit.

Adotou-se o sistema de controle de versão CVS por diretivas internas da Embrapa Informática Agropecuária, uma vez que existe uma equipe que oferece suporte a esse padrão.

Para tornar possível a integração de todas essas ferramentas, optou-se pelo uso do ambiente de desenvolvimento NetBeans, que oferece os recursos necessários para manipular todo esse conjunto de tecnologias, além de ser um excelente auxílio na escrita e geração de códigos.

A Figura 2 ilustra uma tela do NetBeans com o ambiente construído.

Detalhes da configuração desse ambiente, com os comandos necessários, estão na documentação interna do projeto, construído

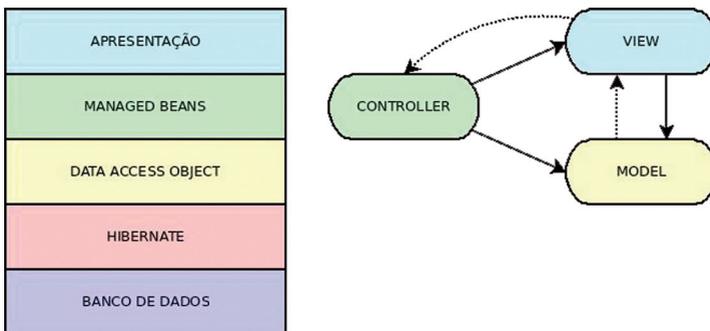


Figura 2. Ambiente de desenvolvimento.

a partir de instruções disponíveis em cada uma das ferramentas. Baseado no desenvolvimento do sistema realizado até o momento, com a confecção de diversos formulários de entrada de dados, regras de negócio e conexão com o banco de dados por meio de uma camada de persistência, o ambiente mostra-se adequado às necessidades do projeto.

Referências

ALMEIDA, R. R. **Model-View-Controller (MVC)**. 2006. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>. Acesso em: 10 ago. 2010.

FERREIRA, A. **Principais características do PostgreSQL**. 2004. Disponível em: http://www.sqlmagazine.com.br/Artigos/Postgre/01_Caracteristicas.asp. Acesso em: 11 ago. 2010.

GONÇALVES, E. **JavaServer faces 2.0 na prática – Parte 3**. 2010. Disponível em: <http://www.edsongoncalves.com.br/tag/jsf-2-0/#LifeCicle>. Acesso em: 14 jul. 2010.

LINHARES, M. **Introdução ao Hibernate 3**. 2005. Disponível em: <http://www.guj.com.br/article.show.logic?id=174>. Acesso em: 18 jun. 2010.