

# ARQUITETURA PARA DESENVOLVIMENTO DE SISTEMAS WEB BASEADO EM FERRAMENTAS DE CÓDIGO ABERTO

## UNIVERSIDADE FEDERAL DO PARANÁ

Alessandre Sena de Oliveira [sena@ufpr.br](mailto:sena@ufpr.br), Antonio Rodrigues Barros [antonio@ufpr.br](mailto:antonio@ufpr.br), Denise Cristiane dos Santos [denisantos@ufpr.br](mailto:denisantos@ufpr.br), Sedimar Antonio Bortolin [sedimar@ufpr.br](mailto:sedimar@ufpr.br)

*Divisão de Sistemas de Informação – CCE/UFPR*

## 1 INTRODUÇÃO

A Universidade Federal do Paraná (UFPR) possui uma demanda de sistemas muito grande, assim como a maioria das instituições públicas do país, contudo a UFPR não possuía uma estrutura centralizadora, além de apresentar um cenário de ausência da cultura de desenvolvimento de software, equipe reduzida e com alta rotatividade de pessoal em virtude da maioria da equipe ser de estagiários.

Motivados por essa demanda em janeiro de 2008 o CCE/UFPR iniciou a busca por uma arquitetura, esta deveria auxiliar o desenvolvimento de sistemas, e possuir os seguintes requisitos: ser de fácil utilização, utilizar padrões consolidados pelo mercado e fazer uso de tecnologias de código aberto (livres de custos).

## 2 ARQUITETURA E FERRAMENTAS

Algumas decisões já haviam sido tomadas quanto a linguagem de programação JAVA e Sistema de Gerenciamento de Banco de dados (SGBD) PostgreSQL. A escolha das ferramentas era o passo seguinte para o desenvolvimento dos novos sistemas para a Universidade Federal do Paraná (UFPR), visto que a linguagem de programação e o SGBD já haviam sido definidos. Baseado no padrão de arquitetura de software Model View Controller (MVC), em que um modelo de software é dividido em três camadas, ou seja, Modelo (manipula a base de dados), Controladora (controla o fluxo de eventos da aplicação, entre as ações do usuário e camada de Modelo) e Visão (interface da aplicação), surgiram algumas opções de frameworks web de código aberto que atenderiam as necessidades das camadas de controle e visão, entre eles o VRaptor, Struts e Struts 2, sendo o último o escolhido.

Inicialmente foi testado o VRaptor, um framework web de fácil aprendizado, porém, com comunidade mais restrita. Por conseguinte foi testado o Struts 2 (Apache Software Foundation), de grande renome no mercado mundial. Este superou as expectativas quanto ao conjunto de funcionalidades, além de ser um projeto em constante desenvolvimento, possuindo uma ampla comunidade de usuários resultando em um grande compartilhamento de experiências (fóruns, lista de discussões, grupos, etc.)

Para a camada de modelo, era de conhecimento o Java Persistence Api (JPA), que é uma especificação Java, e possibilita um mapeamento objeto relacional baseados em anotações. Visto que o Hibernate é uma implementação JPA bastante conhecida no mercado, além de possuir algumas implementações extras como diferencial, foi o framework de persistência adotado para a camada de Modelo. Com o Hibernate, foi possível desenvolver uma série de operações genéricas de busca, e persistência dos dados, acelerando o processo de desenvolvimento das demais aplicações. Outro ponto importante do hibernate foi o uso de interceptadores que possibilitaram a auditoria de todas as transações de manipulação de dados realizadas no banco.

### 3 ETAPAS DE DESENVOLVIMENTO

Os primeiros passos consistiram paralelamente com o desenvolvimento do sistema de pesquisas acadêmicas chamado ThalesWeb, explorando inicialmente os recursos mais básicos que os frameworks ofereciam. Com o passar do tempo, aumentara a complexidade do sistema, e a necessidade de acrescentar novas funcionalidades para a infra-estrutura do sistema.

Foi iniciado então, o desenvolvimento e testes buscando sempre abstrair as operações para se tirar o máximo de proveito das tecnologias escolhidas. Utilizando o padrão de desenvolvimento Data Access Object (DAO) e o padrão Facade na camada de modelo, foi possível criar uma estrutura genérica para os métodos mais comuns (inserção, atualização, exclusão e buscas com quaisquer parâmetros), mantendo a camada de modelo completamente desacoplada das outras.

Com o aumento da demanda de novos sistemas, surgiu então a necessidade de unificar essa estrutura genérica, para ser reaproveitada nos demais sistemas a serem desenvolvidos, ou seja, foi criado uma biblioteca dessa infra-estrutura contendo todas as funcionalidades genéricas para serem utilizadas em todos os projetos mantendo o mesmo padrão de desenvolvimento. A cada nova funcionalidade adicionada na infra-estrutura, é gerada uma nova versão da biblioteca, mantendo sempre atuais os sistemas.

### 4 CONCLUSÃO

A utilização dessa arquitetura apresentou-se viável por ser flexível e de fácil aprendizagem, representando aumento na produtividade dos desenvolvedores e conseqüentemente economia de recursos. Dessa forma, é possível manter todos os sistemas no mesmo padrão de desenvolvimento, e com seus dados armazenados em uma base de dados unificada, preservando a integridade e acessibilidade dos dados. Atualmente alguns projetos encontram-se em desenvolvimento utilizando essa arquitetura:

- Thales - Sistema de registro de pesquisas academia;
- Sap - sistema de administração patrimonial;
- Sigepe - Sistema integrado de gestão de pessoas;
- Security - sistema de restrição de acesso;
- Portal - Sistema que centraliza o acesso aos sistemas;
- Sigo - Sistema integrado de gestão odontológica;

### 5 REFERENCIAS

Fernandes, Raphaela Galhardo Lima, Gleydson de A. Ferreira (2007) **Hibernate com Anotações**. Disponível em: <[ftp://users.dca.ufrn.br/UnP2007/Hibernate\\_Anotacoes.pdf](ftp://users.dca.ufrn.br/UnP2007/Hibernate_Anotacoes.pdf)> (último acesso em março/2009).

Hibernate. Disponível em: <<http://www.hibernate.org/5.html#A7>> (último acesso em março/2009).

Roughley, Ian (2006) **Starting Struts 2**. Disponível em: <<http://struts.apache.org/2.x/docs/home.html>> (último acesso em março/2009)