

APLICAÇÃO DE PADRÕES DE PROJETO NO DESENVOLVIMENTO DE SOFTWARE

Sérgio Akio Tanaka¹
Marcos H. Pansanato²

RESUMO

A aplicação de padrões de projeto e a utilização de um estilo arquitetural visa facilitar e reduzir o tempo de desenvolvimento de um projeto, bem como sua organização. O presente trabalho apresenta aplicações de padrões de projeto da plataforma J2EE (Java 2 Enterprise Edition) no desenvolvimento de software, utilizando como estudo de caso um projeto de advocacia. Tal projeto utilizou o estilo arquitetural de camadas, baseado no MVC (Model View Control), que é composto por três camadas: apresentação, controle e integração. Os padrões aplicados foram: a VO (Value Object) para encapsular os dados de negócios e a DAO (Data Access Object) para encapsular os acessos à origem dos dados.

PALAVRAS-CHAVES: Aplicação de Patentes, Projetos

ABSTRACT

The application of project standards and the use of an architectural style aim at facilitating and reducing the development time of a project, as well as its organization. The present article shoes applications of project standards of the J2EE platform (Java 2 Enterprise Edition) in the software development, using a law project as case study. This law project used the multi-layered architectural style based on the MVC (Model View Control), which is made up of three layers: presentation, control and integration. The standards applied were: the VO (Value Object) to encapsulate the business-oriented data and the DAO (Data Access Object), to encapsulate the accesses to the origin of the data.

47

KEYWORDS: Pattern applications, project.

1. Fundamentação Teórica

Nesta seção é abordada a conceituação necessária para a aplicação dos padrões de projeto no sistema de advocacia, com base na plataforma J2EE.

1.1. Melhores Práticas de Desenvolvimento de Software

Segundo [KRUC, 2003], baseando-se nas experiências comprovadas foram destacadas as melhores práticas de desenvolvimento de software: desenvolver o software iterativamente; gerenciar as exigências; usar arquiteturas baseadas em componente; modelar visualmente o software; verificar continuamente a qualidade do software; controlar mudanças do software.

¹ Docente da UniFil - Colegiado de Tecnologia em Processamento de Dados. E-mail: tanaka@dc.unifil.br.

² Docente da Universidade Estadual de Londrina - UEL. E-mail: {mpansanato}@gmail.com.

1.2. RUP

O RUP é um processo de desenvolvimento de software baseado nas melhores práticas. Segundo [KRUC, 2003] o RUP faz uma abordagem disciplinada para assumir tarefas e responsabilidades dentro de uma organização e desenvolver com alta qualidade, satisfazendo o cliente e cumprindo os compromissos de prazo e orçamento. A Figura 1 apresenta a estrutura do RUP através das suas fases e disciplinas.

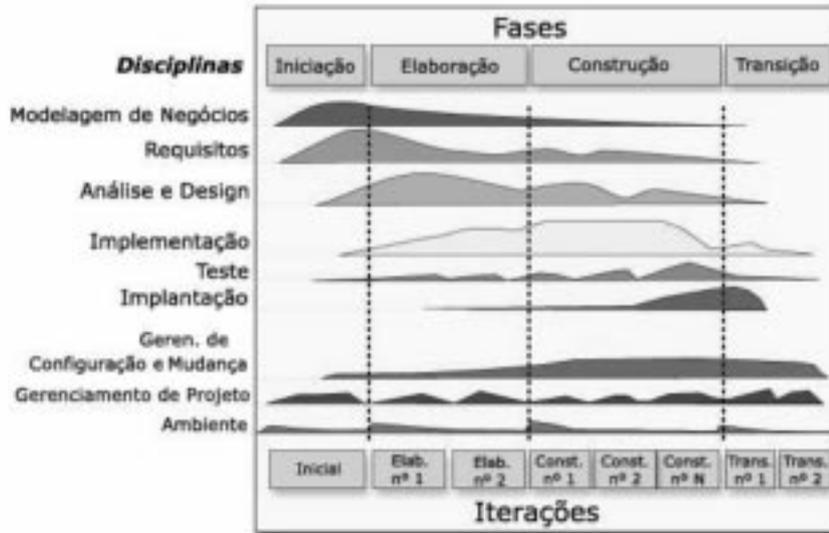


Figura 1: Visão geral da arquitetura do RUP.

48

1.3. Arquitetura de Software

Baseado em (MEND, 2002), os sistemas têm se tornado cada vez maiores e mais complexos. Com isso, engenheiros de software têm adotado princípios de projeto, como a ocultação de informações para simplificar. O uso de uma disciplina é essencial para se ter sistemas com qualidade e com um custo baixo.

1.4. Padrões de Projetos

Baseado em (GAMA, 2000), os padrões de projetos surgiram de experimentos exaustivos para se resolver problemas comuns encontrados em projetos. Cada padrão descreve um problema e sua solução. Solução essa que, muitas vezes, é utilizada de maneira diferente, mas atingindo o mesmo objetivo.

Em geral, um padrão tem quatro elementos essenciais: **nome do padrão**: é uma referência que podemos usar para descrever um problema de projeto, suas soluções e conseqüências em uma ou duas palavras; **problema**: descreve quando aplicar o padrão. Ele explica o problema e seu contexto; **solução**: descreve os elementos que compõem o projeto, seus relacionamentos, suas responsabilidades e colaborações; **conseqüências**: são os resultados e análises das vantagens e desvantagens da aplicação do padrão.

1.5. Padrões de Projeto no MVC (Model-View-Control)

Segundo GAMA, (2000), o MVC é composto por três tipos de objetos. O Model é o objeto de aplicação, a View é a apresentação na tela e o Control define a maneira como a interface do usuário reage às entradas do mesmo. Antes do MVC, os projetos de interface para o usuário tendiam a agrupar esses objetos. O MVC separa esses objetos para aumentar a flexibilidade e a reutilização.

1.6. Plataforma J2EE

De acordo com CORE (2002), a linguagem Java, desde sua criação, teve uma ampla aceitação e um grande crescimento. Novas APIs (application programming interface) e padrões foram surgindo para solucionar problemas no desenvolvimento de software. Com isso, a SUN e um grupo de líderes da indústria resolveram juntar todas APIs e padrões relativos da empresa em uma plataforma para o desenvolvimento de aplicativos empresariais e distribuídos, a Plataforma J2EE.

1.7. Padrões J2EE

Segundo CORE (2002), cada padrão se encontra em algum lugar, entre um padrão de design e um padrão de arquitetura, embora as estratégias documentem partes de cada padrão no nível mais baixo de abstração. O único esquema que é explicitado consiste em classificar cada padrão dentro de uma das três seguintes camadas de arquitetura lógica: camada de apresentação; camada de negócios; camada de integração.

Os padrões J2EE descritos no presente trabalho têm por base experiências coletivas de trabalho na plataforma J2EE com clientes da SJC (Sun Java Center) em todo o mundo. A SJC, uma parte da SPS (Sun Professional Services), é uma empresa de consultoria com foco na construção de soluções baseadas em tecnologia Java para clientes.

2. Estudo de Caso – Projeto de Advocacia

Esta seção apresenta o estudo de caso do projeto de advocacia.

2.1. Funcionalidades Básicas

Um projeto de advocacia visa a oferecer apoio informatizado ao controle de processos, andamentos e arquivamentos de um escritório de advocacia.

2.2. Protótipo da Principal Interface Gráfica

A Figura 2 representa a principal tela do projeto de advocacia, o controle de processo, correspondente a um caso em uso para controlar um processo, conforme apresentado na Figura 3.

Dados do Processo

Número: Data Ab.: Data Fec.:

Cliente:

Contrá:

Adv.:

Tipo P.: Ação:

Comarca: Vara:

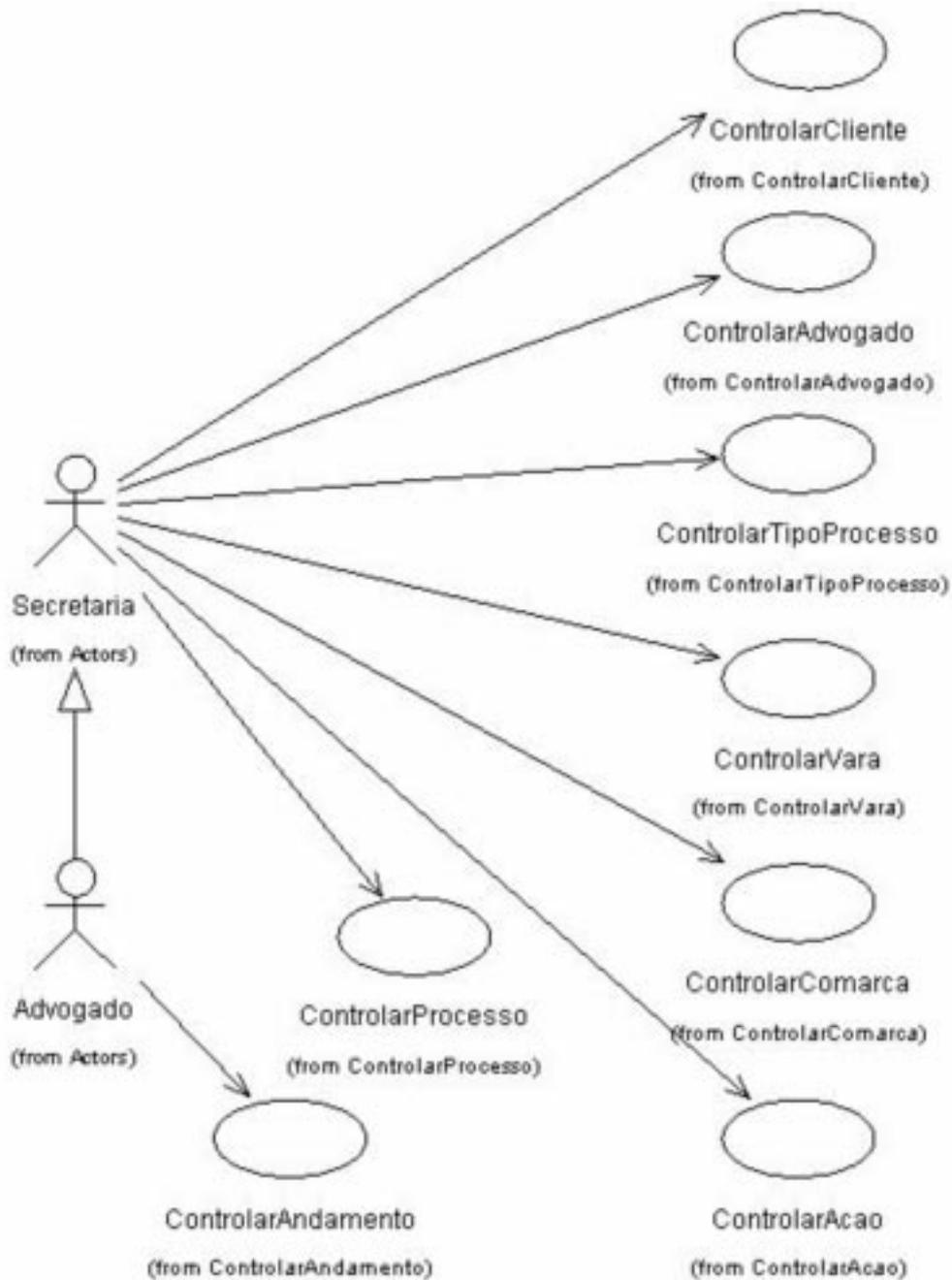
Aberto Fechado Pasta:

Figura 2: Interface Gráfica do Controle de Andamentos.

Todas as outras interfaces gráficas do projeto podem ser visualizadas em: www.dc.uel.br/~mhpananato.

2.3. Diagrama de Caso de Uso

A Figura 3 representa o estudo e a modelagem da interface externa do sistema, ou seja, descreve os requerimentos dos usuários. O sistema é formado por atores, alguém ou algo que utiliza funcionalidades do sistema, e os casos de uso, que representam serviços do sistema.



50

Figura 3: Diagrama de caso de uso do projeto de advocacia.

3. Aplicação de Projeto no Projeto de Advocacia

O Projeto de Advocacia foi estruturado para ter um melhor desempenho e uma melhor organização. Nesta seção é abordada a arquitetura de camadas do projeto, bem como os padrões J2EE utilizados.

3.1. Arquitetura de Camadas

O Projeto Advocacia é baseado no MVC, consistindo em um sistema dividido em 3 camadas.

As camadas do Projeto de Advocacia são: **camada de apresentação**: esta camada encapsula toda a lógica de apresentação para servir os clientes que acessam o sistema. Ela é composta pelas telas do sistema; **camada de negócios**: esta camada fornece os serviços de negócios necessários aos clientes das aplicações. Ela é composta pelas regras de distribuição de funcionalidades; **camada de integração**: esta camada é responsável pela comunicação e recursos externos. Ela é composta pelas classes que fazem a persistência dos dados no banco de dados.

3.2. Aplicação de Padrões J2EE

O sistema contém 2 padrões J2EE: a DAO (Data Access Object) na camada de integração e a VO (Value Object) na camada de negócios.

A DAO é utilizada para extrair e encapsular todos os acessos à origem de dados e gerenciar a conexão com a origem de dados para obter e armazenar esses dados.

A VO é utilizada para encapsular os dados de negócios. Uma única camada de método é utilizada para enviar e recuperar o objeto de dados. Quando o cliente solicita o enterprise bean para os dados de negócios, o enterprise bean pode criar o objeto de dados, preenchê-lo com seus valores de atributos e passá-lo por valor para o cliente. Essa alternativa diminui a sobrecarga na rede.

3.3. Modelagem dos Casos de Uso do Sistema Advocacia

É demonstrada a estrutura dos casos de uso, utilizando a arquitetura de camadas e o uso de padrões J2EE, através de diagramas de classes.

A Figura 4 representa as classes do caso de uso: controlar processo. O caso de uso controlar processo funciona como o caso de uso controlar cliente. A classe FR_Processo é a interface gráfica. A RN_Processo distribui seus métodos a outras regras. A VO_Processo estende a classe VO para utilizar e implementar seus métodos e armazena os dados dos atributos referentes ao processo. A DAO_Processo estende a classe DAO para utilizar e implementar seus métodos, e é também encarregada de fazer a persistência dos dados.

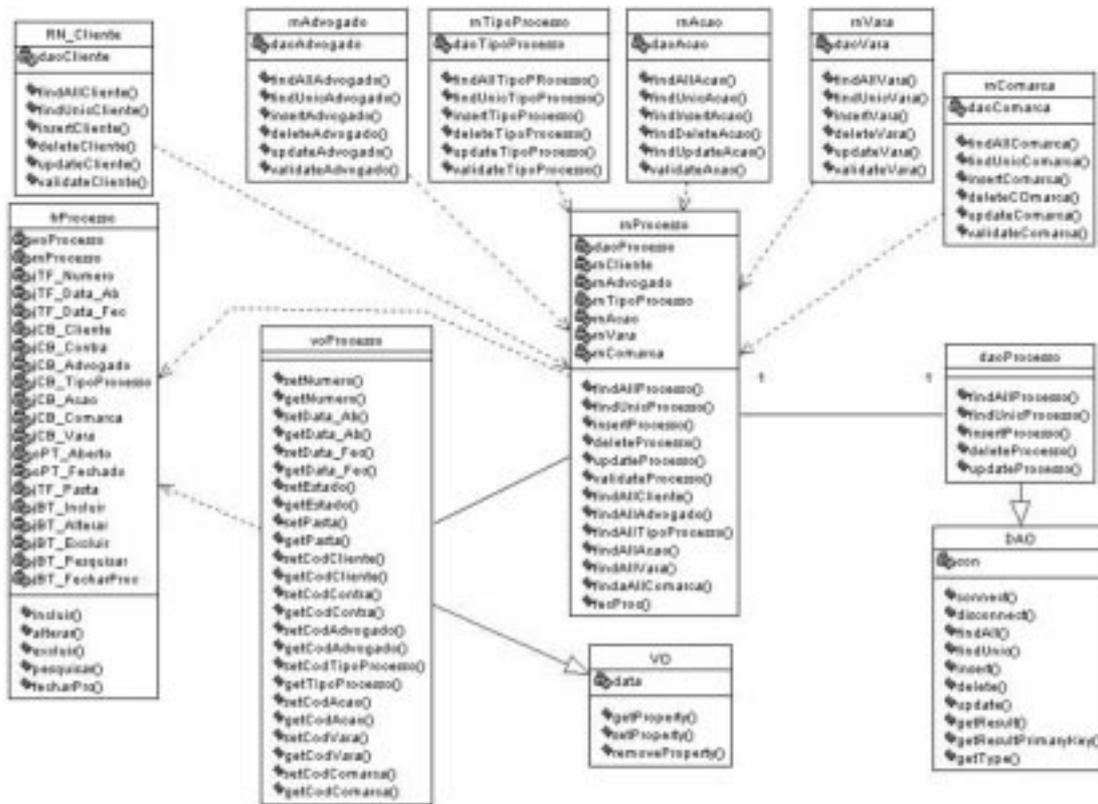


Figura 4: Diagrama de classe do caso de uso controlar processo.

52

REFERÊNCIAS

- ALUR, Deepak; CRUPI, John; MALKS, Dan Core. *J2EE Patterns: as melhores práticas e estratégias de design*. Rio de Janeiro: Campus, 2002.
- GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. *Padrões de projeto: soluções reutilizáveis de software O.O*. Porto Alegre: Bookman, 2000.
- HUSTED, Ted. *Struts em Ação*. Rio de Janeiro: Ciência Moderna, 2004.
- KRUCHTEN, Philippe. *Introdução ao RUP: Rational Unified Process*. Rio de Janeiro: Ciência Moderna, 2003.
- MENDES, Antônio. *Arquitetura de software: desenvolvimento orientado para arquitetura*. Rio de Janeiro: Ed. Campos, 2002.