JArchitecture: Ferramenta de Apoio ao Desenvolvimento de Software

Jean Lucas de Sousa[1], Eduardo Henrique Silva[2], Dalmi Alves Ferreira[3],José Rodrigues Nunes Neto[4], Fernando Corrêa de Mello Júnior[5]
[1, 2, 3, 4, 5] Centro Universitário de Patos de Minas - UNIPAM
jeanlucasdesousa@hotmail.com, fernandocmjr@unipam.edu.br

1 - Introdução

Empresas de desenvolvimento de software têm como principal dificuldade entregar seus produtos dentro dos prazos e dos custos planejados. O não cumprimento desses prazos e custos pode acarretar na insatisfação dos clientes, perda da confiabilidade da organização, aumento dos custos pela correção de erros de desenvolvimento e prejuízos, causados principalmente por falta de planejamento.

O desenvolvimento de software é uma atividade cooperativa, que depende da qualidade de comunicação entre os desenvolvedores, da qualidade dos processos de software utilizados em seu desenvolvimento e da agilidade da manutenção dos produtos, etapa que consome maior parcela do tempo e esforço gastos no processo de desenvolvimento. Embora existam diversas práticas e metodologias que reduzam o esforço despendido na manutenção, desenvolver um sistema que não sofrerá mudanças é praticamente impossível.

Baseado nas dificuldades encontradas pelas empresas de desenvolvimento foi proposto o desenvolvimento de uma ferramenta que estabelece uma arquitetura baseada em camadas e aplica as práticas e as tecnologias de desenvolvimento consolidadas no mercado, a *JArchitecture*.

Os objetivos principais da ferramenta são: padronizar o desenvolvimento, agilizar o processo de implementação, facilitar e apoiar o gerenciamento de mudanças e a realização de manutenções nos sistemas desenvolvidos e melhorar a qualidade dos testes e do código fonte. Além disso, pretende apoiar as atividades de gestão do processo de desenvolvimento a partir da definição das etapas e das atividades da metodologia especificada, automatizando atividades e fornecendo informações sobre o volume de alterações e de código gerado, o que contribui para melhorar a documentação e a determinação da previsão do tempo de desenvolvimento de futuras alterações.

2 - Materiais e métodos

Para o desenvolvimento da ferramenta, o primeiro passo realizado foi a definição dos padrões de projeto e suas especificações, o resumo da arquitetura a ser trabalhada pela ferramenta e a forma que a arquitetura será implantada nos softwares a serem gerados. Após essa definição, foi desenvolvido um software genérico com todos os padrões de projeto, utilizando a arquitetura em camadas. Durante o desenvolvimento, foi registrado o tempo de execução de algumas atividades. Após o desenvolvimento do software, partiu-se para a implementação da ferramenta *JArchitecture*. Em seguida, desenvolveu-se novamente o software genérico, utilizando a *JArchitecture* como apoio.

3 - Resultados e discussão

A ferramenta *JArchitecture* possui cinco funcionalidades implementadas: criação e gerenciamento de projetos de software, criação e manipulação dos diagramas XML, gerenciamento dos validadores, geração de código-fonte e comparação de códigos gerados.

A função de gerenciamento dos projetos é responsável por: criar o projeto se ele não existir, criar a disposição das camadas que serão utilizadas pela ferramenta, realizar as importações de bibliotecas, criar o repositório genérico, criar as configurações de conexão com o banco de dados que será utilizado pelo software e armazenar as informações do projeto.

Cada camada é criada em um projeto diferente, para possibilitar que as camadas de aplicação, de negócios e de dados sejam utilizadas em qualquer projeto. Por meio da disposição das camadas em projetos diferentes é possível criar aplicações Desktop e Web utilizando as mesmas camadas responsáveis pela persistência e pelas regras de negócio. As únicas modificações seriam realizadas na camada de apresentação, onde cada uma das plataformas tem suas particularidades.

Para realizar a geração do código-fonte, a ferramenta *JArchitecture* utiliza o diagrama de entidade relacionamento (DER) e o diagrama de caso de uso, ambos criados dentro da própria ferramenta e salvos em formato XML.

A ferramenta *JArchitecture* permite ao engenheiro de software ou desenvolvedor criar as próprias validações a partir das regras de negócio encontradas no software desenvolvido. Além disso, utiliza as informações do DER para obter validadores, como por exemplo: quantidade de caracteres, casas decimais. A grande vantagem é que quando uma validação particular é criada,

esta regra de negócio é mantida mesmo após realizar uma nova importação do DER.

A partir das informações coletadas no diagrama DER, a ferramenta gera o código fonte nas camadas de dados, negócios e aplicação. A implementação manual destas camadas se resume a códigos repetitivos, mas necessários para o funcionamento do software. Com a geração automática dessas camadas, o desenvolvedor pode se concentrar em desenvolver apenas as regras de negócio e a interface de comunicação entre o usuário e o software, acarretando ganhos de produtividade e tempo.

Além disso, a ferramenta gera a partir do diagrama de caso de uso a camada de visão para Web, usando o paradigma MVC e gerando as interfaces a partir de verbos pré-definidos na ferramenta. A lista dos verbos está definida na Tabela 1 – Verbos utilizados pela ferramenta:

Tabela 1 – Verbos utilizados pela ferramenta.

Verbo	Descrição
Controlar	A ferramenta irá criar uma interface com pesquisa, com a possibilidade de alterar ou criar um novo registro.
Gerenciar	A ferramenta irá criar uma interface listando todos os registros, com a possibilidade de alterar, excluir ou criar um novo registro.
Manter	A ferramenta irá criar uma interface com abas, permitindo o usuário adicionar e remover seus dados com outros relacionamentos.
Pesquisar	A ferramenta irá criar uma interface com um campo de texto (*textbox*) vinculado a um botão, que é preenchido por uma caixa de diálogo.
Selecionar	Indica que a ferramenta cria uma caixa de marcação (*checkbox*).
Marcar	Indica que a ferramenta cria uma caixa de seleção múltipla (*listbox*).
Escolher	Indica que a ferramenta cria botões de seleção (*radiobutton*).
Assinalar	Indica que a ferramenta cria uma caixa de combinação única (*combobox*).

Na comparação de códigos, a ferramenta exibe os códigos que foram incluídos, alterados ou excluídos a partir da leitura dos novos diagramas. O foco principal é permitir que o desenvolvedor tenha conhecimento das mudanças ocorridas e controle quais serão processadas.

A ferramenta exibe as classes geradas que não existiam no projeto, as classes que sofreram modificações e as classes excluídas no projeto. A partir da listagem de classes, o desenvolvedor pode escolher quais serão processadas na execução da ferramenta, selecionando as classes e clicando no botão processar codificação.

Ao efetuar um clique duplo em uma das classes, o software exibe uma nova tela conforme a Figura 1 – Alterações realizadas no código fonte, onde são mostradas as alterações, exclusões ou inclusões de atributos, associações e métodos, que podem ser comparados. A representação das mudanças no código auxilia o desenvolvedor a escolher apenas as modificações que ele considera importantes, não o obrigando a aceitar o código gerado pela ferramenta.

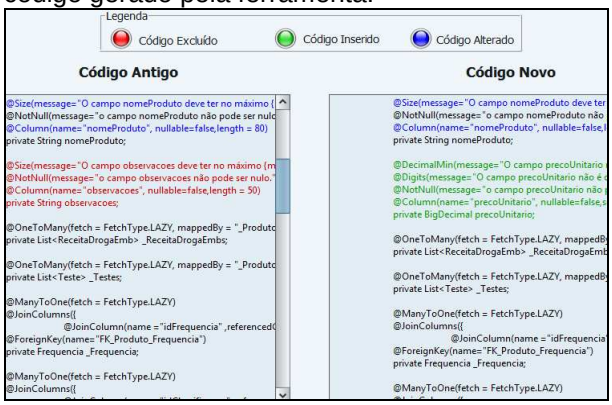


Figura 1 – Alterações realizadas no código fonte

A ferramenta também mantém todas as alterações de código realizadas manualmente nas camadas de aplicação, dados e objetos de negócio.

Uma característica da utilização da ferramenta *JArchitecture* no apoio da geração do código fonte é o ganho de produtividade no desenvolvimento. Para a comprovação da capacidade de aumentar a produtividade do processo de desenvolvimento, foi proposta a implementação de um software com e sem a utilização da ferramenta possibilitando a medição de tempo e custos envolvidos.

O software genérico inicialmente possuía 16 entidades e 5 grandes módulos. Após o desenvolvimento inicial, foi necessário adicionar novos módulos e entidades, totalizando 42 entidades e 11 grandes módulos,

A Tabela 2 – Mensuração do tempo de desenvolvimento sem a utilização da ferramenta apresenta valores médios em minutos para a realização de atividades geradas automaticamente pela ferramenta no desenvolvimento do software, baseada na mesma arquitetura, porém sem o apoio da *JArchitecture*. Em algumas atividades o tempo pode variar de acordo com a complexidade da entidade no modelo, como o mapeamento dos objetos de negócio. Já em outras atividades o tempo foi fixado, pois o modelo não influencia na complexidade das classes.

Tabela 2 – Mensuração do tempo de desenvolvimento sem a utilização da ferramenta.

Atividades realizadas por projeto	**Tempo**
Criar a disposição das camadas e importar suas dependências	30
Criar o repositório genérico	10
Criar a configuração do banco de dados	10
Total / atividades gerais	**50**
Atividades realizadas por entidade	**Tempo**
Criar interfaces e as implementações dos repositórios de cada entidade.	5
Criar interfaces e as implementações da *Application* de cada entidade.	15
Criar e mapear as entidades.	10 ~ 50
Total / atividades por entidade	30 ~ 70

Os valores são baseados em um desenvolvedor que possui conhecimento na arquitetura e na linguagem utilizada, sem a utilização de nenhuma ferramenta, e ignorando eventuais erros. Esses valores podem variar de desenvolvedor para desenvolvedor.

Posteriormente, utilizando a ferramenta, todas as etapas citadas acima foram criadas automaticamente. Para realizar o mesmo processo com a ferramenta, as etapas de cadastro de projeto, criação do DER e do Caso de Uso, aplicação de validações e processamento duram aproximadamente uma hora.

Com a utilização da ferramenta, houve um ganho de produtividade que foi de aproximadamente 40 minutos por entidade desenvolvida. No sistema criado, que possui quarenta e duas entidades distintas, foi possível obter uma economia de 27 horas, tempo expressivo que representa aproximadamente uma semana de trabalho de um programador.

A Tabela 3 – Cargos e salários em reais apresenta uma pesquisa de cargos e salários na área da tecnologia da informação, foram utilizados apenas os cargos que atuam com o desenvolvimento de software e que poderiam utilizar a ferramenta.

Tabela 3 – Cargos e salários em reais.

Cargo	**Júnior**	**Pleno**	**Sênior**
Analista Java	3.000,00	4.500,00	6.000,00
Programador Java	2.500,00	3.800,00	5.500,00
Webmaster	6.139,00	6.798,00	8.121,00

Realizando uma média a partir dos salários apresentados, obteve-se um valor de R$ 5150,89 mensais para o pagamento de um profissional. Supondo que o ele trabalhe 160 horas mensais, o valor / hora será R$ 32,19. O valor gasto para a realização das atividades automatizadas pela *JArchitecture* é de aproximadamente R$ 869,13.

Além disso, o custo para aplicar as alterações (manutenções e inclusões de novos requisitos) de um projeto depois de seu desenvolvimento para outro é praticamente nulo, pois a ferramenta fornece total integração entre módulos criados internamente e mantém as alterações realizadas durante a primeira etapa, permitindo que somente os códigos necessários para manter a integridade do sistema sejam adicionados ou alterados.

4 - Considerações finais

A partir das métricas obtidas após o desenvolvimento de um sistema, da geração da documentação e do código padronizado foi possível determinar que a ferramenta *JArchitecture* tem a capacidade de apoiar o desenvolvimento de sistemas e, ao automatizar tarefas de programação, é possível reduzir o tempo de desenvolvimento e de manutenção, permitindo que o desenvolvedor se preocupe principalmente com as regras de negócio específicas de uma aplicação.

A geração do código fonte padronizado, o controle das alterações existentes e a utilização da arquitetura em camadas tornam o código mais legível, desacoplado e reutilizável, além de flexível e aberto a mudanças.

Graças ao gerenciamento das atividades de desenvolvimento, a obrigatoriedade de atualizar as mudanças do banco de dados na ferramenta e a documentação das regras de negócio, é possível manter uma documentação mínima para o projeto mesmo durante a fase de manutenção do produto, garantindo ainda que o modelo de dados fique sincronizado com a aplicação, o que dificilmente ocorre no desenvolvimento convencional.

Em continuidade ao desenvolvimento da ferramenta, são levantados os tópicos para desenvolvimento futuro: melhoria da capacidade de gestão das atividades, realização de relacionamentos “virtuais” entre entidades que não possuem relacionamento no banco de dados, criação de código fonte em outras linguagens de programação, controle de usuários dentro da ferramenta, gerenciamento e suporte a logs de alteração e criação da documentação automática.

5 - Referências

INFO, Notícias. Quadro de Salários. Disponível em: <http://info.abril.com.br/professional/salarios/>. Acesso em: 12 out. 2011.