

DESENVOLVIMENTO DO FRAMEWORK GERA-JAVA

Entrega do Trabalho de
Conclusão da Pesquisa do Programa do PIBITI/CNPq
FEMA/IMESA 2013.

Autores:

Lucas Pompeo Pontes Spinelli – Aluno

Dr. Luiz Carlos Begosso – Orientador

SUMÁRIO

1. INTRODUÇÃO	3
1.1 OBJETIVO GERAL.....	5
1.2 OBJETIVO ESPECÍFICO.....	5
2. ENGENHARIA DE SOFTWARE	6
2.1 DESAFIOS DA ENGENHARIA.....	6
3. DESENVOLVIMENTO DO <i>FRAMEWORK</i> GERA-JAVA	7
3.1 GERADOR DE CÓDIGO.....	7
3.2 FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO DO <i>FRAMEWORK</i> GERA-JAVA.....	7
3.2.1 JSF.....	7
3.2.2 HIBERNATE.....	8
3.2.3 HTML.....	9
3.2.4 NETBEANS.....	9
3.2.5 JAVA.....	9
3.3 <i>FRAMEWORK</i> GERA-JAVA.....	10
3.3.1 IMPORTÂNCIA.....	10
3.3.2 CLASSES UTILIZADAS.....	11
3.3.3 FUNCIONAMENTO.....	11
3.3.4 DIAGRAMA DE CASO DE USO.....	12
4. CONCLUSÃO	13
REFERÊNCIAS.....	14

1. INTRODUÇÃO

Pode-se dizer que nos dias de hoje existe um desafio constante na área de Engenharia de Software, que é o de melhorar o processo de desenvolvimento. A cada dia as empresas de diversos ramos de atividades procuram soluções que atendam especificamente os seus requisitos em um tempo muito rápido.

Mesmo com a constante evolução de métodos, técnicas e ferramentas, a entrega de software em prazos e custos estabelecidos nem sempre é alcançada. A consequência dessa situação é o desenvolvimento de códigos desnecessários, falta de padronização nos métodos e nos atributos, e também repetição de tarefas com a mesma finalidade na programação. Em decorrência disso, aumentam-se os custos da produção diminuindo o lucro final, e muitas vezes chega-se a ter prejuízo.

Uma das soluções para resgatar o tempo perdido durante o processo de desenvolvimento do software, é renegociar o contrato com o cliente de forma a obter um aumento no prazo e no custo do produto final. Esta situação é desgastante para ambas as partes e já foi amplamente discutida em Pressman (1995) e Sommerville (2007).

Alternativamente, as empresas podem utilizar de ferramentas que as auxiliam na automatização e padronização do gerenciamento do projeto e na fase de implementação, o processo de desenvolvimento de software pode ser beneficiado por ferramentas geradoras de código. Um bom exemplo dessas ferramentas é o uso dos *frameworks*.

De acordo com França (2000), um *framework* é uma estrutura de classes inter-relacionadas, que constitui uma implementação inacabada, para um conjunto de aplicações de um domínio. Além de permitir a reutilização de um conjunto de classes, um *framework* também minimiza o esforço de desenvolvimento de novas aplicações, pois já contém a definição de arquitetura gerada a partir dele bem como, tem predefinido o fluxo de controle da aplicação.

Dessa forma, a partir de um *framework* pode-se compor um gerador de código que é uma ferramenta de software desenvolvida para gerar o código fonte de uma aplicação. Este código pode servir como base do desenvolvimento de uma

aplicação ou simplesmente se tornar um protótipo, conforme destaca Leal (2005).

Para Zemel (2005), podem-se citar cinco vantagens do uso de um framework:

Utilidade: - O objetivo primeiro dos frameworks é auxiliar no desenvolvimento de aplicações e softwares. Para tal, eles têm funcionalidades nativas das mais variadas, que ajudam você a resolver as questões sobre programação do dia-a-dia com muito mais qualidade e eficiência.

Segurança: - Os bons frameworks são projetados de modo a garantir a segurança de quem programa e, principalmente, de quem usa o que foi feito a partir dele. Não se preocupe mais com aquelas intermináveis linhas de código para evitar um SQL Injection, por exemplo; com frameworks, a parte de segurança já “vem de fábrica”.

Extensibilidade: - Os frameworks permitem que você estenda suas funcionalidades nativas. Se aquela biblioteca de envio de e-mails por SMTP não contempla todas as possibilidades que você gostaria, simplesmente estenda suas funcionalidades e as use como se fossem parte do framework (na verdade, elas serão).

Economia de tempo: - O que você demoraria algumas horas ou alguns dias para fazer, você encontra pronto em um framework. Pense nas quão trabalhosas aquelas funções de manipulação de imagens são; usando um framework que tenha isso, você só usa, e pronto.

Ajuda fácil: - Os que desenvolvem frameworks geralmente disponibilizam material de qualidade nos web sites ou repositórios oficiais, com uma vasta documentação a respeito. Além disso, os bons frameworks sempre têm uma comunidade de desenvolvedores dispostos a se ajudarem entre si. É um prazer para os que já sabem mexer ajudar os que ainda não sabem (embora a falta de tempo também seja uma realidade)

1.1 OBJETIVO GERAL

O presente projeto tem por objetivo o desenvolvimento de um *framework* denominado *Gera-Java*.

1.2 OBJETIVO ESPECÍFICO

O *Gerador Automático de Código*, agilizará o processo de desenvolvimento de software, no tocante à criação de módulos CRUD (*Create, Read, Update e Delete*), relatórios, consultas simples, consultas avançadas, menu principal, tabelas no banco de dados.

O *Gerador Automático de Código* fornecerá classes que podem ser usadas por todo o projeto agilizando a construção da interface gráfica e também no acesso ao banco de dados. O código será gerado na linguagem Java.

2.ENGENHARIA DE SOFTWARE

Sommerville (2007), destaca que a engenharia de software é uma disciplina que envolve aspectos desde o início, até a manutenção do sistema. Para o autor existem duas frases importantes a serem consideradas:

Engenharia: Aplicações de teorias, métodos e ferramentas, fazendo as coisas funcionarem. Também sempre procurando soluções para os problemas.

Produção de Software: Gerenciamento de projeto de software e o desenvolvimento de ferramentas, métodos e teorias que produzem o software também estão relacionados a esse processo.

A Maneira mais eficaz de produzir software de alta qualidade, é adotando um trabalho organizado e sistemático.

2.1 DESAFIOS DA ENGENHARIA

Para Sommerville (2007), a engenharia de software enfrenta três desafios:

Desafio de heterogeneidade: O sistema é distribuído em rede, ou seja, em diversos computadores com sistemas operacionais diferentes. Esse desafio se baseia em desenvolver técnicas mais flexíveis para se adaptar a essa heterogeneidade.

Desafio da entrega: O tempo necessitado é necessário para a qualidade do software, mas esse desafio consiste em diminuir o tempo de entrega, sem diminuir a qualidade.

Desafio da confiança: Necessitamos confiar no software. Esse desafio consiste desenvolver técnicas para mostrar que o software pode ter a confiança do usuário.

3. DESENVOLVIMENTO DO FRAMEWORK GERA-JAVA

3.1 GERADOR DE CÓDIGO

O gerador agiliza o processo de desenvolvimento de software, eliminando assim tempo gasto com funções repetidas, códigos desnecessários, falta de padronização, etc. Não começando do zero, gerando um código fonte básico para dar início ao seu trabalho.

Para o Codegeneration (2006), um gerador de código apresenta as seguintes vantagens:

- qualidade do código: códigos gerados manualmente variam ao longo do projeto. É possível iniciar com alta qualidade e terminar em baixa ou vice-versa. Geradores de código padronizam a codificação minimizando pontos de falha, tendo em vista que bugs e não conformidades conhecidos são corrigidos e aplicados ao longo do sistema;
- consistência: códigos gerados seguem padrões de nomenclatura. Isso é benéfico para outros processos automáticos que usam este código (ex: Programação orientada a aspectos, AOP);
- produtividade: códigos automáticos são gerados em frações de segundos, comparados com o tempo que um programador levaria para gerar o mesmo código. A partir de outra perspectiva, os geradores possibilitam que o programador concentre-se em processos que exigem conhecimento no negócio, deixando um tempo maior para a programação que tem um tempo crítico;

3.2 FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO DO GERA-JAVA

3.2.1 JSF

Basicamente JavaServer Faces ou simplesmente JSF, é um *framework* que permite a elaboração de interfaces de usuário web colocando componentes em um formulário e ligando-os a objetos Java permitindo a separação entre lógica e regras de negócio, navegação, conexões com serviços externos e

gerenciamento de configurações. Seu ponto forte é um grande número de componentes e um design muito flexível o que permitiu que este framework crescesse muito acomodando novas tecnologias.

O JSF possui as seguintes partes: Um conjunto de componentes pré-fabricados de IU (interface de usuário), um modelo de programação orientado a eventos e um modelo de componentes que permite a desenvolvedores independentes fornecerem componentes adicionais.

O JSF possui componentes simples como input e botões e outros componentes sofisticados como tabelas de dados e árvores, porém o mais importante talvez seja o fato de integrar o padrão Java EE e estar incluído em cada servidor de aplicação Java EE, podendo facilmente ser adicionado a um container web. (Godoy, 2011)

3.2.2 HIBERNATE

Hibernate é um *framework* que permite ao desenvolvedor libertar-se de preocupações com a persistência de seus objetos. Com ele, pode-se fazer o mapeamento de objetos para bancos de dados relacionais, persistindo os objetos em tabelas de um banco de dados relacional, permitindo a convivência dos dois paradigmas (relacional e objetos) .

A relação entre o banco de dados e os objetos pode ser feita através de anotações ou com XML.

Oferece também métodos que podem ser utilizado para a criação das tabelas no banco de dados, com base nos objetos que são configurados como entidades no Hibernate.

As operações de criação, atualização, remoção e seleção de objetos são feitas através da API do Hibernate que, consultando seus arquivos de mapeamento e configuração, envia os comandos corretos ao banco de dados configurado.

Assim, a aplicação fica portátil entre bancos de dados. (Sodré, 2011)

3.2.3 HTML

HTML é uma linguagem de marcação utilizada para produzir páginas na Web. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto do "casamento" dos padrões HyTime e SGML.

HyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), conectados por hiperligações. O padrão é independente de outros padrões de processamento de texto em geral.

3.2.4 NETBEANS

O NetBeans é um projeto open source de sucesso, com uma grande base de utilizadores, uma crescente comunidade e perto de 100 (e a aumentar!) parceiros mundiais. A Sun Microsystems fundou o projeto NetBeans em junho de 2000 e continua a ser o seu principal patrocinador.

Atualmente existem dois produtos: o IDE NetBeans (NetBeans IDE) e a Plataforma NetBeans (NetBeans Platform).

O NetBeans IDE é um ambiente de desenvolvimento - uma ferramenta para programadores, que permite escrever, compilar, depurar e instalar programas. O IDE é completamente escrito em Java, mas pode suportar qualquer linguagem de programação. Existe também um grande número de módulos para estender as funcionalidades do IDE NetBeans. O NetBeans IDE é um produto livre, sem restrições à sua forma de utilização. (NetBeans, 2013)

3.2.5 JAVA

Java é uma linguagem de programação e uma plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. É a tecnologia que capacita muitos programas da mais alta qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros, por exemplo. O Java é executado em mais de 850 milhões de computadores pessoais e em bilhões de

dispositivos em todo o mundo, inclusive telefones celulares e dispositivos de televisão.

3.3 FRAMEWORK GERA-JAVA

Como citado anteriormente, o *framework* Gera-Java é um gerador automático de código. Tem por finalidade ajudar, e muito, o desenvolvimento de um sistema web. Esse gerador de código é responsável por fornecer classes que podem ser usadas por todo o projeto agilizando a construção da interface gráfica e também no acesso ao banco de dados, economizando tempo e contribuindo com a padronização do projeto. O código será gerado na linguagem Java.

3.3.1 IMPORTÂNCIA

O tempo dedicado, pelo desenvolvedor de software, à criação de aplicações básicas, porém de extrema importância, tais como, telas, páginas e objetos para conexão com banco de dados é elevado se comparado com o tempo crítico do desenvolvimento, qual seja o processo de construção de regras de negócio, numa dada linguagem de programação.

O desenvolvimento do *Gerador Automático de Código* se justifica pela tentativa de agilizar o processo de desenvolvimento do software, favorecendo o *feedback* dado pelo cliente sobre pequenas partes de produto entregue e acabado.

Uma das motivações para este trabalho é que o *Gerador* poderá produzir código para o ambiente WEB proporcionando agilidade de fato na criação de código e padronização, sobrando tempo para outras especificações do sistema.

3.3.2 CLASSES UTILIZADAS

Para a criação do framework Gera-Java, foram geradas as classes: CriarJava, Projeto, Classe, AtributosClasse, PáginaGeradora e PáginaGeradoraBean. A seguir, descreve-se cada uma das referidas classes.

CriarJava – Onde fica praticamente tudo a ser gerado. Essa classe fica responsável por pegar as classes a serem geradas, e setar o Bean (Classe de relação com a tela), Dao(Classe de relação com o banco de dados), e o Modelo(Objeto).

Projeto – Onde fica o caminhos da aplicações, pacotes, etc...

Classe – Nome da classe, nome do banco, etc...

AtributosClasse – Tipo, nome, relacionamento, etc (Tudo relacionado aos atributos de uma classe)

PáginaGeradora – Página responsável por gerar as classes e seus atributos, e também o menu.

PáginaGeradoraBean – Configuração da página geradora, onde chama todas as classes já citadas.

3.3.3 FUNCIONAMENTO

O Gera-Java apresenta o seguinte funcionamento: Abrirá a tela geradora, onde, o usuário colocará o nome da classe desejada. Quando cadastrar a classe, uma datatable abrirá com um atributo padrão (código). Na tela há uma opção (+) para cadastrar novos atributos para essa classe - (existe a possibilidade de alterá-la e removê-la). Em relação ao cadastro de atributos, estes possuem as seguintes características: Nome no banco, nome, tipo, tamanho, obrigatório, etc... Assim que gerada a classe com atributos... o usuário será redirecionado para uma página de menu contendo as classes e atributos já cadastrados para aquele projeto. Em seguida o usuário poderá cadastrar mais classes e objetos, (construindo o relacionamento,) e assim, gerando a tela JSF.

3.3.4 DIAGRAMA DE CASO DE USO

A Figura 1 ilustra o diagrama de Casos de Uso para o *Framework Gera-Java*.

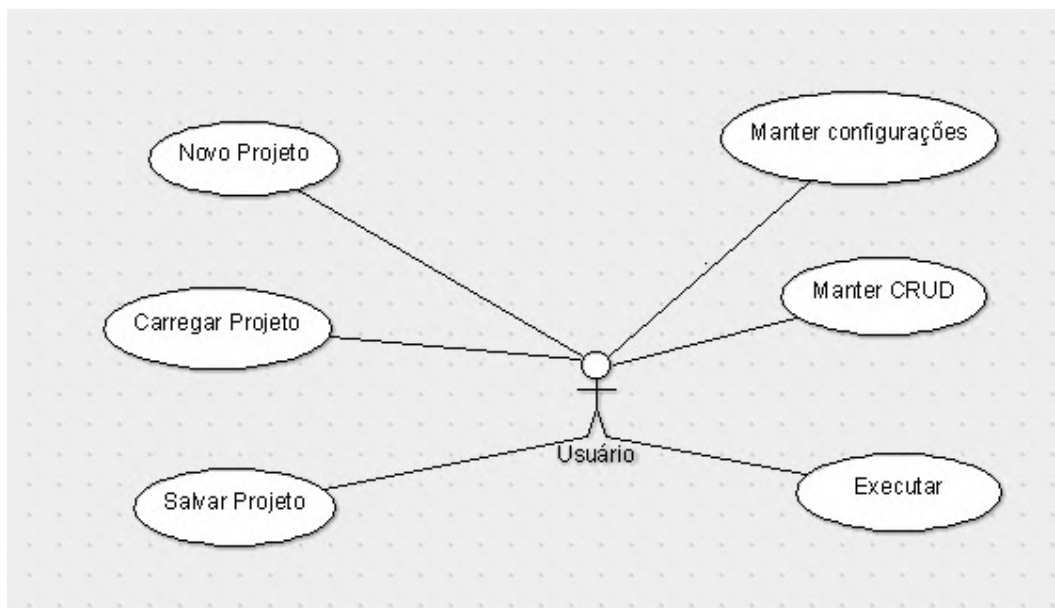


Figura 1 – Caso de Uso do *Framework Gera-Java*

Caso de uso – Carregar Projeto: Tem por finalidade carregar um projeto já feito, ou seja, como pré-condição, deve existir um projeto salvo. Caso não haja projeto salvo, criar um novo projeto. Ator: Usuário.

Caso de uso – Salvar Projeto: Tem por finalidade é salvar o projeto. Ator: Usuário.

Caso de uso – Manter CRUD: Tem por finalidade é gerar classes que executem as funções: inserir, alterar, excluir e busca. Ator: Usuário.

Caso se uso – Executar: Tem por finalidade gerar as classes, dao, tela, bean, banco de dados e menu principal. Deve haver uma classe já cadastrada. Ator: usuário.

4. CONCLUSÃO

O desenvolvimento do *framework* visa atender o ambiente Web na linguagem Java, facilitando a vida do usuário, gerando códigos-fonte, proporcionando diminuição do tempo de programação, padronização de códigos.

Esse projeto proporcionou ao autor o conhecimento de novas tecnologias e certamente levará esse aprendizado para o resto de seu curso e de novos desafios a serem enfrentados.

REFERÊNCIAS

GODOY, Fernando. Disponível em

<<http://fernandogodoy.wordpress.com/2011/02/12/o-que-e-jsf/>>, acesso em 28/07/2013.

Java,

http://www.java.com/pt_BR/download/faq/whatis_java.xml?os..=&printFriendly=true, acesso em 30/07/2013.

LEAL, Marcelio D'Oliveira. CLASSGENERATOR: Um gerador de artefatos multiplataforma. Trabalho de Conclusão de Curso – Departamento de Informática – UFPA, 2005.

NetBeans, https://netbeans.org/index_pt_PT.html, acesso em 30/07/2013.

SODRÉ, Eduardo. DESENVOLVIMENTO DO FRAMEWORK JAVA-FACIL.

Trabalho de Conclusão de Curso – Análise e Desenvolvimento de Sistemas – FEMA, 2011.

Wikipedia, http://pt.wikipedia.org/wiki/Gerador_de_c%C3%B3digo, acesso em 29/07/2013.

