

# JBuilder 2008 com Applic

## O novo paradigma em desenvolvimento

Um dos problemas mais persistentes enfrentados pelos desenvolvedores diz respeito a como transferir conhecimento entre os membros da sua equipe. Alguns exemplos: como um arquiteto indica no código a sua intenção para que os próximos desenvolvedores possam usar como referência? Como os membros especializados na implementação da interface web de uma aplicação sabem o local exato do erro que deve ser solucionado na camada de persistência? E onde – se é que este local existe – ficam os repositórios de código e os componentes reutilizáveis que este membro acredita que existam? As Application Factories permitirão que os desenvolvedores seniores comuniquem a intenção, capturem as instruções e recomendações, e apontem os recursos por meio de uma única ferramenta bastante simples.

### *Capturando a intenção do desenvolvedor através das Application Factories*

O cerne da questão é que à medida que as aplicações se tornam mais complexas e passam a envolver um número maior de camadas e frameworks, os desenvolve-

dores são obrigados a especializarem-se, adquirindo conhecimento detalhado – e altamente restritivo – sobre uma pequena seleção de tecnologias. Isso inibe e, por vezes, até elimina a exposição do desenvolvedor a tecnologias de outras partes da mesma aplicação. Como resultado, o conhecimento passa a ser cada vez mais enclausurado em “guetos” e a transferência de conhecimento passa a ser difícil ou quase impossível. Mesmo dentro de um único grupo, os membros podem achar difícil deixar “migalhas de pão” para servir de indicação para os outros desenvolvedores; do mesmo modo, para eles é difícil ir atrás do conhecimento de outros membros da equipe, a menos que trabalhem geograficamente próximos o suficiente para formular a pergunta cara-a-cara.

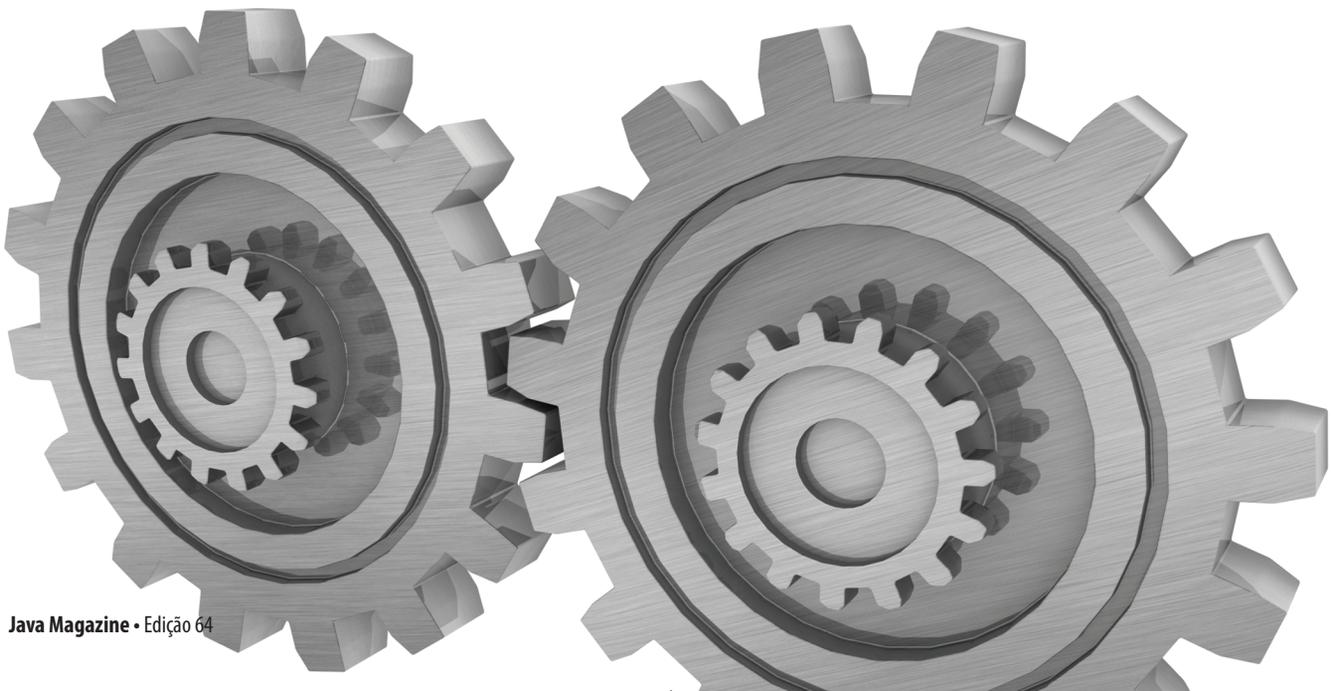
À medida que os projetos tornam-se maiores, a complexidade da transferência de conhecimento surge como um obstáculo ao cumprimento dos prazos, ao estabelecimento da funcionalidade apropriada, da capacidade de resposta às mudanças de requisitos, e especialmente à capacidade de resolver os problemas de modo ágil e

adequado. Apesar de sua universalidade, esta questão é raramente discutida e, até agora, não foi diretamente enfrentada pelas ferramentas.

O JBuilder 2008 traz uma nova tecnologia para IDE chamada Application Factories, que permite que os desenvolvedores seniores comuniquem a intenção, capturem as instruções e recomendações, e apontem os recursos. A primeira implementação das Application Factories está disponível no JBuilder 2008, IDE baseado no Eclipse. Este artigo apresenta brevemente como esta nova tecnologia funciona e como ela pode ser usada pelas equipes para que possam tirar o máximo proveito de suas experiências internas.

### **O que é Application Factories?**

Application Factories são grupos de ferramentas e templates especializados para o desenvolvimento e reutilização de aplicações. Elas contêm uma descrição da aplicação, sua lógica de desenvolvimento, informações sobre a configuração, assim como dicas indicativas sobre como usar e construir cada um dos componentes.



# ation Factories

## Java

ANDREANO LANUSSE

Elas são definidas pelos arquitetos ou desenvolvedores seniores tanto para programas que se repetem quanto para aplicações que são utilizadas uma única vez. Através delas o designer pode especificar todas as informações que um desenvolvedor menos experiente pode precisar para escrever o produto final.

Para este desenvolvedor não se trata do cenário típico de visualizar diagramas UML abstratos ou cartões CRC, tentando compreender o código que precisa ser escrito. Ao contrário, trata-se de uma experiência mais próxima a de uma navegação por meio de Application Factory. Esta navegação consiste em seguir os passos no template, lendo a lógica e a explicação das restrições, seguindo as dicas indicativas dos vários recursos para então escrever o código.

Uma Application Factory pode conter scripts com indicações tais como: "reutilize o módulo de interface gráfica das aplicações derivativas em /code/derivapp/mainUI.java, e conecte-a ao banco de dados de títulos de dívidas em vez de usar o sistema de opções". Ou pode ainda incluir outros scripts e templates a serem preenchidos pelo desenvolvedor. Usadas deste modo, as Application Factories transformam o desenvolvimento em uma série de ações que convertem, propositalmente, artefatos e templates específicos em seqüências de código.

Tais artefatos também podem ser inseridos no código pelos desenvolvedores quando eles indicarem as razões por trás das decisões de implementação, com sugestões embutidas sobre como modificar o código caso alterações futuras sejam necessárias. Ao incluir estes artefatos durante a tomada de decisão, os desenvolvedores dão suporte aos futuros responsáveis pela manutenção, fornecendo-lhes uma sinalização bastante adequada para que naveguem pelo código sem grandes problemas.

Application Factories podem conter vários outros artefatos úteis, tais como diagramas de arquitetura, metadados de complementação de código e várias tecnologias para estender a aplicação.

### Como Application Factories funcionam

A Figura 1 mostra o aspecto deste processo dentro do JBuilder.

No painel central, o cursor localiza-se em uma linha que precisa ser modificada. No painel direito, o script correspondente é destacado (neste caso, um arquivo JavaScript). No painel central inferior fica o script com os nomes das variáveis definidas no template, destacados para que o desenvolvedor lhes dê andamento. No painel central superior fica o recurso associado à linha destacada, que neste caso é parte de um stub auto-gerado.

O exemplo acima compreende a principal aplicação das Application Factories: um corpo de código é auto-gerado, digamos, a partir de uma ferramenta UML. O arquiteto segue pelo código gerado,

inserindo scripts e tags para que os desenvolvedores os utilizem como um template básico para instâncias futuras daquela aplicação específica ou para variantes de ocorrência única. Um site pode facilmente manter uma coleção destes templates para reutilização quando necessário, sem correr o risco de atrasar o trabalho quando um designer ou um programador que trabalhem freqüentemente nas aplicações precisem se ausentar. Estes artefatos e dicas indicativas possibilitam que qualquer desenvolvedor exerça o papel de especialista e faça a tarefa progredir.

Application Factories permitem adicionar tags e controlar a evolução da aplicação, executar scripts para gerar partes do código, assim como capturar artefatos alterados como parte da Arqueologia de script. Em um cenário típico, um desenvolvedor ou um arquiteto abre uma Application Factory e então:

- Passa a navegar conceitualmente através da aplicação, usando os metadados da Application Factory;

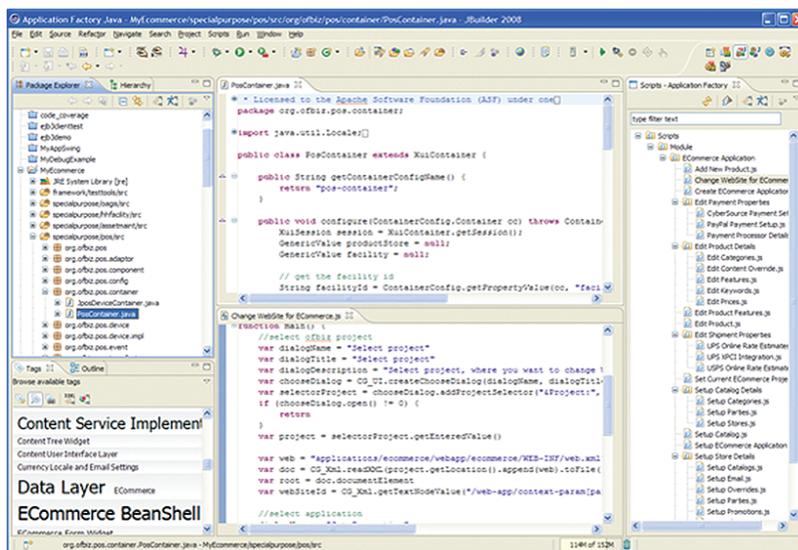


Figura 1. Uma Application Factory no Eclipse, na qual um template foi implementado como um arquivo JavaScript (painéis direito e central inferior).

- Executa um script para gerar código de template ou rastros;
- Muda para uma fase de resolução das alterações realizadas pelo script. Esta fase inclui:
  - Revisar cada alteração e o motivo pelo qual foi realizada;
  - Efetuar quaisquer edições locais nas alterações realizadas pelo script;
  - Adicionar tags às alterações, conforme necessário;
  - Rolar até a linha de script que produziu a alteração, customizando-a ou modificando-a conforme o caso;
  - Efetivar as alterações no *changeset*.
- Repetir, se necessário.

A **Figura 2** mostra uma janela do JBuilder 2008 que permite que o desenvolvedor gerencie as tags de diferentes partes de uma aplicação.

Application Factories também são providas de funções de navegação de foco, que permitem que os desenvolvedores caminhem adiante ou para trás entre as tags da Factory e a aplicação, tornando a navegação verdadeiramente coesa. Esta capacidade evita que os desenvolvedores tenham que pesquisar quantidades enormes de código para encontrar os itens que precisam alterar – em um projeto novo, o que se torna definitivamente um auxílio significativo.

## Utilizando Application Factories com Frameworks

Os benefícios desta abordagem são claros para fábricas que se baseiam em

frameworks, tais como Struts ou Spring, ou em software corporativo como Tomcat. Os designers seniores podem criar artefatos de Application Factory enquanto codificam, preservando assim o percurso mental que transcorreram, disponibilizando tal trilha para o uso de outros desenvolvedores. Como já foi dito, esta abordagem também permite que os membros seniores funcionem como tutores ou mentores.

O JBuilder 2008 traz diversas aplicações estruturadas, chamadas de Módulos, que se encaixam aos principais frameworks Web e de persistência em Java. As aplicações pré-prontas existentes são derivadas do trabalho com a AppFuse ([www.appfuse.org](http://www.appfuse.org)), uma ferramenta especializada na criação de estruturas de aplicação baseada em framework através do uso de Struts2, SpringMVC, JSF, Hibernate, JPA, Spring, dentre outros. O JBuilder 2008 inclui módulos baseados em outros frameworks, assim como uma funcionalidade significativamente expandida para Application Factories.

Application Factories oferecem aos arquitetos várias ferramentas que lhes possibilitam construir os seus próprios módulos de aplicação, tão ricos quanto necessário. O design básico do Módulo pode ser realizado com as ferramentas.

As ferramentas são particularmente úteis para aplicações que possuam funcionalidades claramente definidas e restritas. Os scripts do módulo podem ser escritos manualmente ou gerados automaticamente.

Application Factories capacitam os arquitetos a buscar nas aplicações as funções desejadas, de modo que a reutilização seja maximizada. Além disso, várias porções dos módulos podem ser transformadas em “somente leitura” ou “read/write”, depen-

dendo puramente da permissão do arquiteto para que os desenvolvedores alterem e atualizem o design do módulo.

## Um Novo Modelo para o Design de Software

Esta abordagem é voltada para a criação facilitada de templates e para a captura da intenção e do conhecimento dos desenvolvedores que implementam uma aplicação e fornece um novo modelo para o design dos softwares. Ela possibilita aplica-

ções de rápida distribuição e que consistem em dados de configuração, templates e de algum código. Iniciando com esta base, o desenvolvedor pode customizar e construir a aplicação, deixando-a pronta para execução em questão de horas, reduzindo drasticamente o custo e as despesas gerais associadas à complexidade das aplicações Java atuais.

Para saber mais sobre Application Factories, acesse a Rede de Desenvolvedores da CodeGear – <http://dn.codegear.com/br> e assista os vídeos disponíveis sobre Application Factories. Você pode também baixar a versão trial através do link <http://www.codegear.com/br/products/jbuilder> e começar a utilizar imediatamente as Application Factories.

### **Blogs.codegear.com/andreaslanusse**

Este é o link para o meu blog, onde disponibilizo diversas informações.

### **Fontes gratuitos - <http://cc.codegear.com/Author/38483>**

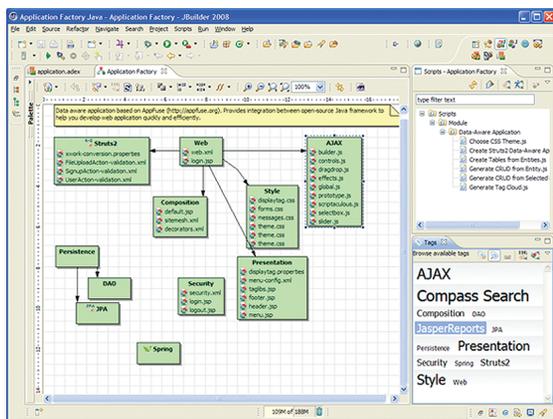
Este é o link para acesso ao CodeCentral, onde disponibilizo diversos exemplos com fontes.

### **[dn.codegear.com/br](http://dn.codegear.com/br)**

Site de relacionamento da CodeGear com os desenvolvedores, possui diversos artigos e tutoriais em português.

### **[www.codegear.com](http://www.codegear.com)**

Site oficial da CodeGear, contém todas as informações sobre os produtos da CodeGear.



**Figura 2.** As tags de uma aplicação podem ser agrupadas de modo lógico e gerenciadas de maneira inteligente.



### **Andreano Lanusse**

[andreano.lanusse@codegear.com](mailto:andreano.lanusse@codegear.com) é Product Line Manager da CodeGear & Evangelista Líder para a América Latina, sediado em Scotts Valley – Califórnia, tem como missão

atuar junto à equipe de desenvolvimento dos produtos CodeGear no direcionamento estratégico dos produtos e definição de novos recursos de acordo com as solicitações da comunidade de desenvolvedores. Possui vasta experiência em desenvolvimento de aplicações atuando em diversos projetos de missão crítica, além de estar certificado em todos os produtos Borland/CodeGear e MCP em SQL Server e Windows 2000.

### **Dê seu feedback sobre esta edição!**

A Java Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

**[www.devmedia.com.br/javamagazine/feedback](http://www.devmedia.com.br/javamagazine/feedback)**