ProjectNCode - Anotações

*Esse documento é uma compilação de idéias e links importantes que serão usados como referência para a construção da ferramenta.*

*ORIGINAL E ATUALIZADO ESTÁ NO DOCS:*

*https://docs.google.com/document/d/1\_pOl5T5UncNbT0ZCVcjLAiIG8ZYQ3DTdzBHmbRlfFL0*

[Histórico de revisão e implementação](#_ci5hxzcoelh0)

[[28/11/2010]](#_2pktp8vo6byo)

[Sobre a documentação](#_xgr1qz49alt4)

[Concorrentes](#_l8me6y9w9yj9)

[Diferencial dos Concorrente](#_7io5mk3p8gjw)

[Links Com Codigos que podem Ajudar](#_xo1eqeemzvff)

# Histórico de revisão e implementação

## [28/11/2010]

- O projeto possuirá um determinada configuração, que nada mais é do que uma pasta/jar que comtem, os templates, builders, css/html, e possue um arquivo

xml para descrever "o que" essa configuração irá prover para o plugin.

A nova ideia ao invés de gerar artefatos separados é que será possivel gerar "casos de uso", que serão descritos nesse XML da configuração e que irá apontar todos os templates usados nesse caso de uso, e seus respectivos builders e wizards.

[] Migração para plataforma Eclipse E4

- tutorial: http://www.vogella.de/articles/EclipseE4/article.html

# 

# 

# Sobre a documentação

- Deve dizer que a ferramenta é focada no desenvolvimento MVC e MDD.

- Deve ser um tutorial explicando como se criar novos builders, como por exemplo um DAO ou outro recurso

de outro framework e que seja chato de se manter, como arquivos XML.

- Fazer um planejamento sobre os videos, mostrando o build incremental, e lidando com as mudanças,

começando com um projeto simples e extendendo-o.

- Modelo de Documentação de Requisitos interessante

http://www.eclipse.org/datatools/project\_sqldevtools/plans/SQL%20Dev%20Tools%20Plan.html

- Fazer comparações ou comentarios sobre EGF "(Eclipse Generation Factories)", que ainda esta na fase PROPOSAL.

Tambem montar um diagrama de alto nivel de como funcina a geração, assim como no EGF

http://www.eclipse.org/proposals/egf/

- Algumas refencias sobre MDA e padrões podem ser tirados da MOF Model To Text Transformation Language

http://www.omg.org/spec/MOFM2T/1.0/

MOF - Tambem define padroes de scripts para geracao de codigo

- Várias dicas de Geração de Código presentes no Eclipse (pode ir para o site)

<http://eclipse.dzone.com/news/effective-eclipse-dont-write-c>

# Padrões e Especificações:

- ATL - a model transformation technology

<http://www.eclipse.org/atl/>

<http://www.slideshare.net/ejuliot/2010-06-22-omg-obeo> (pagina 39)

Caracteristicas:

- Editor / Debug / Profiler

[] MetaModelo, XMI, UML, etc...

- Meta-Object Facility : http://www.service-architecture.com/web-services/articles/meta-object\_facility\_mof.html

- http://www.service-architecture.com/web-services/articles/common\_warehouse\_meta-model\_cwm.html

- MDA: http://www.service-architecture.com/web-services/articles/model\_driven\_architecture\_mda.html

[] Para importação/exportação de XMI pode ser usado o Eclipse Modelling Framework.

- http://wiki.eclipse.org/MDT-UML2

- Tutorial: Getting Started with UML2 : http://www.eclipse.org/modeling/mdt/uml2/docs/articles/Getting\_Started\_with\_UML2/article.html

[] Especificação para Interfaces graficas, e possue bons textos para a documentaçã do projeto.

http://www.eclipse.org/proposals/pmf/

http://wiki.eclipse.org/PMF

- Implementação:

http://wiki.eclipse.org/E4/XWT

http://wiki.eclipse.org/E4/XWTDesigner

http://wiki.eclipse.org/E4/XWT/Running\_the\_demos

http://www.soyatec.org/eclipse/xwt/getting-started/getting-started.html (video)

# Concorrentes

http://akrogen.sourceforge.net/

http://eclipsework.sourceforge.net/ (brasileiro)

http://fornax-sculptor.blogspot.com/2009/08/introduction-to-sculptor.html

- tutorial de um projeto exemplo: http://fornax.itemis.de/confluence/display/fornax/3.+Advanced+Tutorial+%28CSC%29

-wavemaker (open-source) - IDE WEB

http://www.wavemaker.com/product/screencasts.html

https://github.com/SpringSource/wavemaker

https://github.com/cloudjee/wavemaker (mais atual.)

- jwork

Framework for Java code generation of web applications is for the above framework mentawai, done to expedite the process of creating a project in Java technologies with the JPA, Hibernate, Mentawai, AJAX with DWR among others.

https://sourceforge.net/projects/jwork/

- Acceleo (open-source)

Seque as especificações da OMG o: MOF. Parece que está se tornando um projeto do Eclipse.

Tem uma proposta parecida com a da minha ferramenta, então da para tirar referências

http://www.acceleo.org/pages/home/en

http://www.acceleo.org/pages/demo-flash/en [VIDEOS]

http://www.acceleo.org/pages/uml2-to-jee-java-struts-hibernate-generator/en (boa referencia para a documentação)

<http://www.slideshare.net/ejuliot/2010-06-22-omg-obeo> (

- Interessante:

- Templates com suporte a autocomplete.

- WaveMaker (open-source)

Ferramenta completa, WEB, para criação de projetos, aborda desde a manipulação do banco de dados e geração das tabelas

a criação VISUAL dos formulários. (da VMWARE/SpringSource)

- Fast-code (parece um pouco com o meu)

<http://www.3pintech.com/products/fast-code/templates.htm>

- Tem uma documentação bem boa de como gerar os snippets

- blocos ordenados de inserção de codigo (insert, fields, method)

Exemplos:

- <http://fast-code.sourceforge.net/getter-setter.html>

- <http://fast-code.sourceforge.net/templates-config.xml>

- http://appfuse.org/display/APF/Home

# Diferencial dos Concorrente

- Processo envolendo MDD (mais rápido e que as mudanças envolvendo o modelo, reflitam na aplicação)

- Suporte a Modulos (separados em projetos ou não)

- Focado em GUI e independente da plataforma, pode ser Flex, Swing, J2ME, JSP, etc ...

- A adição de NOVAS funcionalidades, como analisers e builders se torne uma tarefa simples.

- Suporte a arquiteturas de projetos diferentes.

[] Permitir salvar qualquer arquivo como tempalte, só botão direiro > salvar como template.

Ai será aberto o editor de templates.

[] Geração de um MetaModelo para consultas/criteria, usando o Hibernate Metamodel Generator

- http://www.hibernate.org/subprojects/jpamodelgen.html

[] Definirção de Formulários usando DSL.

EMFText - já cria os editores baseado no EMF.

[] Integração com o Visual paradigm

- Ele possue uma API chamada OpenAPI, e a lib dele esta da pasta 'lib' do Visual Paradigm

- Referencias:

http://www.visual-paradigm.com/support/documents/#extend

http://www.visual-paradigm.com/support/documents/vpumluserguide/276/330/7416\_customizingc.html (Customizando a Geração de Código)

Pode ser usado para fazer com que minha estrutura de templates seja parecida.

- Links Interessantes:

http://knowhow.visual-paradigm.com/openapi/reload-your-plugin-without-restart-the-application/

http://knowhow.visual-paradigm.com/openapi/retrieve-relationships-from-models-api/

[] Seguir a OSGI para a modularização.

Tutorial Simples: http://infoblogs.com.br/view.action?contentId=194687&OSGI-Modularizando-sua-aplicacao.html

http://www.infoq.com/articles/modular-java-dynamic-modularity

[] Hibernate Tools API para mapeamento com Hibernate, o JEESpider usa ele.

[] Pode gerar um documento Jasper pré-configurado com os fields ou parametros necessários.

NOTA: Não sei se isso é viavel, pois o jasper ja tem isso e muito bem.

[] Criar template para Classes, assim como Pessoa/TipoPessoa/Endereço que precisa de um formulário já organizado.

[] Criar um Builder de Proposito geral (GenericBuilder) que será usado pelos para criar novos templates de geração.

na configuração do Template deve ser informado qual builder sera usado.

Esse builder coloca uma lista de Classes e para cada classe seus atributos, bem como outras variáveis de ambiente.

[] Criar uma ferramenta ou Wizards para criar os templates. Onde o usuário informa para cada template.

A plataforma, os Frameworks , e a categoria (GUI, Model, Configuração)

[] Criar uma QueryBuilder ou SearchBuilder, que monta os critérios de busca do Hibernate, ainda não ligado diretamente ao formulário.

IDEIA: Isso pode ser trasformado em um modulo a parte que pode ser vendido...

- Se basear no QueryBuilder do SqlYog

- Poder gerar as Consultas em: HQL, Criteria, Criteria+Projection, Criteria+BeanProjection, e se possivel até em SQL

- Criar sistema para analizar uma consulta SQL e trasformar no diagrama de consulta, que dai pode gerar qualquer coisa...

-Referencias:

http://fornax-sculptor.blogspot.com/2009/11/internal-dsl-for-criteria-part-13.html (pode ajudar com algumas ideias)

http://www.eclipse.org/datatools/project\_sqldevtools/ (tem um parser...)

http://jsqlparser.sourceforge.net/home.php (parece bom - e bem maduro)

[] sqldesigner - editor de diagramas in JS.

http://code.google.com/p/wwwsqldesigner/

[] Sistema de geração de Modelos de relatório (Jasper)

a API DynamicJasper, pode ser usada para isso e atraves o metodo:

DynamicJasperHelper.generateJRXML

[] Os Winzards de Geração (menu outros) deve ser mais ou menos como o "Novo" do Eclipse, dividido em Categorias tipo: WEB, Testes, Mentawai, Hibernate

Os menus diretos exibidos pela ferramente pode ter os principais, deacordo com o framework usado, e pode ter tambem os mais usados, e

a opção "outros".

[] Winzards para Criação de Actions, Services (que gera o service e sua implementação).

- Possivelmente pode ter a opção para criar a partir de um Bean a Action já o Service e o Formulario

[] Versão WEB

IDE Opensource para desenvolvimento web: https://github.com/ajaxorg/cloud9

outro: http://collabedit.com/

[] Fazer o parser de classes Java, em projeto web (utiliza as libs do eclipse)

- http://www.programcreek.com/2011/01/a-complete-standalone-example-of-astparser/

[] Sistema de merge WEB - opensource

http://www.mergely.com/

[] Modeloar e Criar diagrmas em JavaScript

http://www.jointjs.com/demos.html

http://neyric.github.com/wireit/

[] Ferramenta: Plugin para o eclipse para criação de CodeSnipets/Templates automaticamente baseano nos códigos do Projeto.

Foi premiado pelo Google.

Descrição: http://code-recommenders.blogspot.com.br/2013/05/gsoc-proposal-workspace-local-code.html

Source:http://git.eclipse.org/c/recommenders.incubator/org.eclipse.recommenders.codesearch.git/

Ideias para o ApplicationManager

====================================

Possibilidades de Edição

- De forma gráfica, assim como no Struts usando o plugin do JBossTools.

- Ou como a configuração do Hibernete usando um formulário.

Implementação:

- http://www.eclipse.org/jwt/ (Java Workflow Tooling) - olhar scremmcast , muito interessante..., vario modos de edição e visualização.

- http://wiki.eclipse.org/JWT\_Metamodel\_Extension\_Specifications/Prototype (Como criar extenções do MetaModelo)

- É possivel tambem estabelecer um Controle de Acesso nas Actions do Application Manager

ou configura, isso através do Editor gŕafico de workflow.

Ex: http://www.slideshare.net/mdutoo/eclipse-summit-2009-aspect-oriented-modeling (slide 11/12)

- Sincronizar o JavaDoc das Actions com a configuração do ApplicationManager, isso pode aumentar e muito o entendimento das actions e facilida a

documentação.

Como configurar as InnerAction.

- Pelo editor gŕafico.

- Pensar em outra solução para quando ele criar a innnerAction já existir um modo de ele poder configurar o ApplicationManager de uma forma rapida e eficiente.

Talvez um suporte para o autocomplete (no app manager e nas jsps) com o nome das actions.

<http://stackoverflow.com/questions/2078971/my-proposals-are-hidden-when-using-java-proposals-task-focused>

http://help.eclipse.org/indigo/index.jsp?topic=%2Forg.eclipse.jdt.doc.isv%2Freference%2Fextension-points%2Forg\_eclipse\_jdt\_ui\_javaCompletionProposalComputer.html

http://stackoverflow.com/questions/2078760/eclipse-extension-point-for-attribute-content-in-jsp-editor-webtools?lq=1

- Prover um Painel chamado ActionView, que o cara pode ver as configurações das Actions quando exiver com alguma selecionada.

isso deser produzido de uma maneira genérica para funcionar para qualquer framework,

. Quando se estiver editando uma Action, criar uma extensão do Outline, para mostrar as actions

existentes, e sua configuração no ApplicationManager, pra onde ela da o forward.

Ou seja: Seria um navegador de Actions.

Exemplos que podem ajudar

http://help.eclipse.org/help33/topic/org.eclipse.jdt.doc.isv/guide/jdt\_api\_open\_editor.htm

- Sistema para detectar erros se a action esta dando forward para um JSP ou View que n existe, e na solução, criar uma opção

para criar a JSP.

================================================

Estrutura dos Templates

[] Mentawai

[] View

[] JSP

- register.jsp.groovy

`-->{

Deve ter um tipo: no caso, é um ViewTemplate, isso será usado para organizar os artefatos que podem ser gerados.

Deve saber qual o Builder: Pois os templates genericos\* podem usar Builders customizados.

O Nome final deve ser Internacionalizado: Só o nome ? e o conteudo ? No caso do conteudo pode -se ter tags IF dentro do template..

}

- list.jsp.groovy

- search.jsp.groovy

[] Flex

[] ExtJS

[] ConfigFiles

- displaytag

- ApplicationManager.java.groovy

[] Shared

[] View (JSP,Flex,Etc)

[] ConfigFiles (diplaytag, hibernate, etc..)

[] JavaFiles

[] Libraries

[] MyTemplates

# Sera ondes os templates novos ou editados serão colocados.

A diretiva CrudConifg: Nela será configurada quais os templates que estão em USO para gerar um Crud.

[] Quando estiver editando uma Action, colocar opções para ele ir para o : Service, AppManager (na configuração), Formulario.

O mesmo pode acontecer com o Inverso: Service para a Action ou Formulario....

Implentação: Isso pode ser feito através de botões no ToolBar.

[] Suporte a geração de testes

- Usando selenium (Base: )

- Usando JUnit (base: JUnitDocklet, que pode ser um plugin)

- Referencias:

http://fornax-sculptor.blogspot.com/2009/08/testing-is-simple.html (geração automática)

http://fornax.itemis.de/confluence/display/fornax/3.+Advanced+Tutorial+%28CSC%29#3.AdvancedTutorial%28CSC%29-JUnit (explicação da metodologia)

http://fornax-sculptor.blogspot.com/2009/11/how-we-do-automated-regression-testing.html (selenium e maven)

[] Geração do Diagrama uml das classes.

- O Sculptor faz isso. exemplo: http://fornax-sculptor.blogspot.com/2009/07/world-is-changing.html (posição: 4:15)

- Procurar uma ferramenta ou API q já faz isso.

> Uma é a do Plugin UML do Eclipse outra: http://www2.research.att.com/~john/Grappa/ (a do Sculptor)

[] Ferramenta para Desenho da Tela / GUI Builder.

https://code.google.com/a/eclipselabs.org/p/swixeditor/

- Contruido com as ferramentas do Eclipse, permite desenhar as telas e salvar a representação em XML.

- Ferramenta criação dragand drop de formulários, usando bootstrap

<https://github.com/minikomi/Bootstrap-Form-Builder>

- Ferramenta de criação de formulários EXT (source está no item about!)

http://tof2k.com/ext/formbuilder/

- Editor Visual de Layout

<http://www.layoutit.com/build> (olhar no github)

# Links Com Codigos que podem Ajudar

- Usando várias partes da API do Eclipse JDT

<http://www.programcreek.com/2011/01/best-java-development-tooling-jdt-and-astparser-tutorials/>

- Plugin do JIRA que abre do Browes um arquivo no Eclipse

<https://bitbucket.org/atlassian/connector-eclipse> ([classe](https://bitbucket.org/atlassian/connector-eclipse/src/b26692bd1c5d8eb7fc4e01cf94d731874c25a49a/com.atlassian.connector.eclipse.directclickthrough.ui/src/com/atlassian/connector/eclipse/internal/directclickthrough/servlet/DirectClickThroughServlet.java?at=default))

- Criar dialogos que exibem uma lista de classes já filtradas, de acordo com a necessidade. O legal é que pode

customizar a aparencia, separando as classes de modelo e tals, e possu histórico.

<http://eclipse-tips.com/tutorials/13-extending-filtereditemsselectiondialog>

﻿Topicos de Leitura do Help do Eclipse

Link: http://help.eclipse.org/ganymede/nav/2

- Platform Plug-in Developer Guide

- Plug-in Development Environment Guide

- JDT Plug-in Developer Guide

- EMF Developer Guide

- Fonte de Referencias de Desenvolvimento (incluindo Livros)

http://tonnymadsen.blogspot.com.br/2013/05/sources-of-eclipse-related-information.html