

Eclipse – Quem es tu ???

Rogério Peres Gomes

A idéia deste artigo é apresentar uma das mais completas ferramentas de desenvolvimento Java e o poder que essa ferramenta tem com a adição de plugins de primeira linha.

Introdução

O projeto Eclipse.org nasceu da iniciativa de grandes empresas líderes em seus seguimentos, tais como Borland, IBM, QNX Software, Rational Software, Red Hat, Suse, Toghether Software (comprada pela Borland), Sybase e Fujitsu. Em fevereiro de 2004 o Eclipse.org tornou-se independente, ficando “livre” da IBM.

Atualmente o Eclipse.org mantém 4 grandes projetos e 19 subprojetos, dentre esses grandes projetos podemos destacar a IDE Eclipse para desenvolvimento Java.

Esta ferramenta permite personalizar totalmente o ambiente de acordo com o projeto que está sendo desenvolvido, seja ele desenvolvimento para plataforma WEB ou Desktop (Standard), com ou sem EJBs, J2ME, etc. Além disto, permite a instalação de plugins que trabalham integrado com a ferramenta. Em sua versão básica, o Eclipse já traz alguns plugins (que podemos chamar de plugins básicos), dentre eles podemos destacar a integração com o Ant e com o CVS.

Apesar de ser uma IDE free e opensource, o Eclipse é bem completo, permitindo quê seja feito nele tudo o que pode ser feito em uma IDE paga, claro, que em alguns casos com um grau de dificuldade maior, mas para atividades de desenvolvimento básicas como desenvolvimento em Swing, Jsp, Servlets ele atende muito bem, tem grandes facilidades para desenvolvimento de EJBs, mas nada que se compare ao Jbuilder da Borland neste quesito.

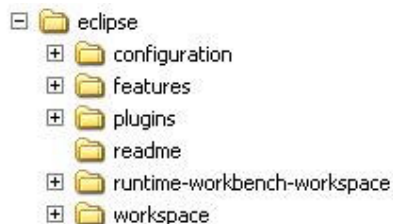
Será que vale a pena comprar uma IDE de desenvolvimento Java ?

Essa pergunta me perseguiu durante um tempo e não consegui me conformar em ter que pagar preços salgados por IDEs Java se tínhamos o Eclipse e o Netbeans que poderiam atender muito bem a todas minhas necessidades. Em um projeto que trabalhei necessitei utilizar uma IDE já comprada pelo cliente e percebi que as vezes vale a pena pagar. Como por exemplo criar EJBs no Eclipse pode ser um trabalho um pouquinho chato, mas no Jbuilder pode-se fazer isso totalmente de forma visual.

O início – Instalando no Windows

O download do Eclipse pode ser feito no endereço <http://www.eclipse.org/downloads/index.php>, sendo que hoje a versão estável é a 3.0.1.

Descompacte o arquivo baixo para a raiz do drive C:\ da máquina, tendo a seguinte raiz.



Pode-se criar um atalho do arquivo eclipse.exe em sua área de trabalho (Desktop), pois o eclipse não faz isso automaticamente.

Eclipse.org

Pode-se conseguir muita informação interessante sobre o Eclipse, inclusive sua documentação no site www.eclipse.org.

O início – Instalando no Linux

O download do Eclipse pode ser feito no endereço <http://www.eclipse.org/downloads/index.php>, sendo que hoje a versão estável é a 3.0.1.

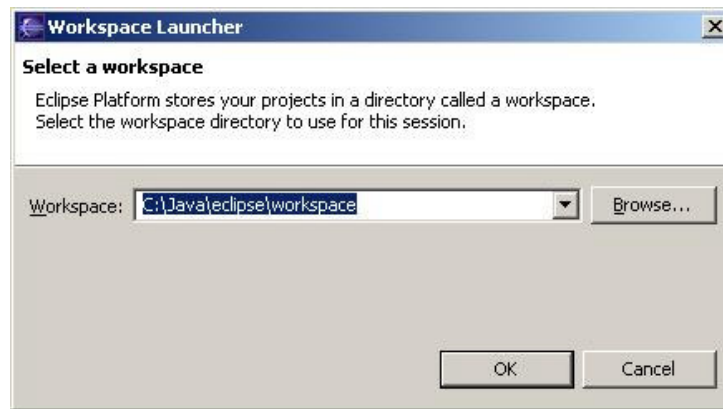
Após o download, descompacte o arquivo executando o seguinte comando:
unzip /download / eclipse-SDK-3.0.1-linux-motif.zip

Não esqueça de dar permissão para o usuário no diretório criado, para isto basta utilizar o comando:
chmod +x /usr/local/bin/eclipse

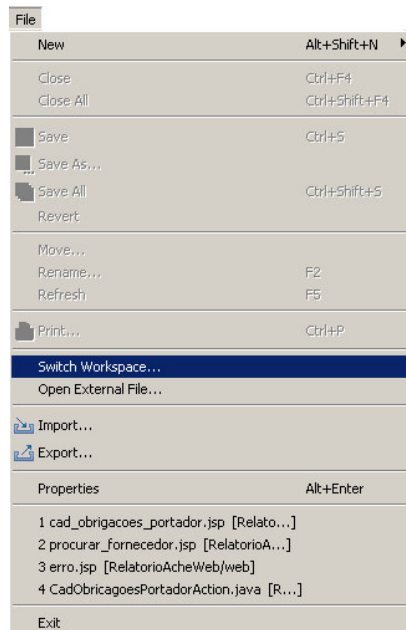
Depois basta executar o binário do eclipse. Basta digitar **eclipse** no diretório onde foi descompactada a aplicação.

Inicialização

Durante a inicialização do eclipse deve-se configurar o diretório padrão das Workspaces, que será o local onde ficarão armazenados os projetos.



Esta configuração pode ser alterada em File > Switch workspace

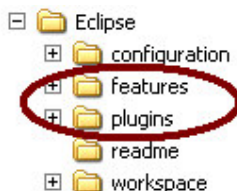


Instalação de plugins

O Eclipse traz na sua instalação, alguns plugins básicos para desenvolvimento, dentre eles o plugin do CVS e do Ant, mas, existem plugins que facilitam muito o trabalho do desenvolvedor, por exemplo, o VEP para criação de interfaces em Swing, o Easy Struts para se trabalhar com o Struts, o Jboss-Ide ou o Lomboz, para se trabalhar com JSP, Servlets e EJBs.

Dentro deste contexto, iremos exemplificar a instalação do VEP, do Jboss-Ide e do OMONDO, demonstrando também, como usar essa turma do barulho, ou melhor, do Eclipse.

Observe que dentro da estrutura de pastas/diretórios do Eclipse existe uma pasta chamada plugins, essa pasta/diretório contém as classes dos plugins. Há uma outra pasta/diretório interessante chamado features que faz uma espécie de gerenciamento dos plugins.



A importância dessas pastas para o mundo do plugins é extrema, na pasta plugins devem ser descompactados ou instalados todos os plugins. Alguns desses plugins podem ter algumas classes que serão instaladas na pasta features também.

Iremos começar realizando a instalação do plugin do Jboss-Ide, este plugin permite a criação de EJB's e Jsp facilmente, além de permitir uma integração básica com o Jboss, onde, pode-se realizar deploys das aplicações de forma "amigável" (bastam dois cliques - ☺, quer mais fácil que isso????).

JBOSS-IDE

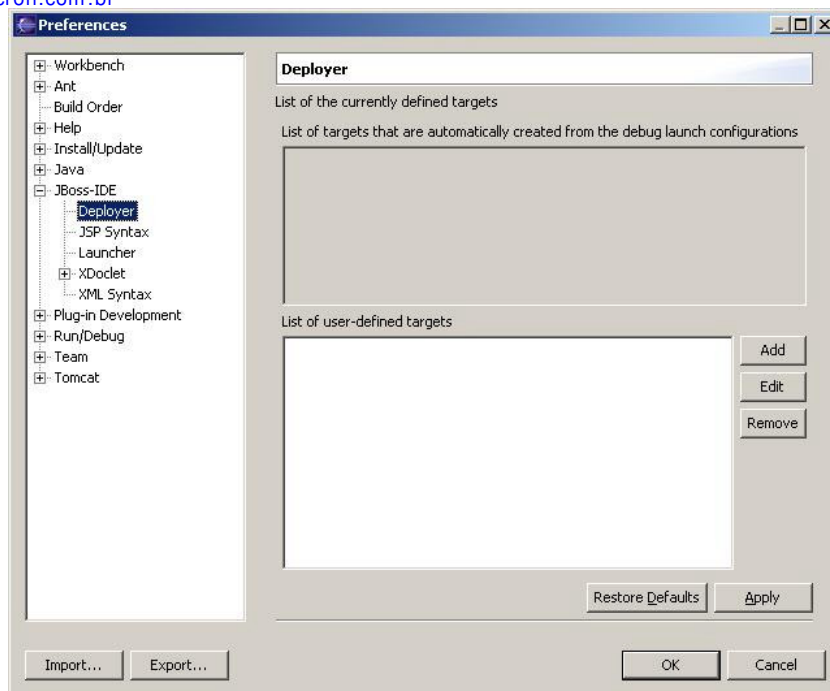
Iniciaremos o procedimento indo até a página do Jboss-Ide, <http://www.jboss.org/products/jbosside>, e fazendo o download da versão correspondente ao seu Eclipse, estaremos aqui utilizando a versão 1.4 que é compatível com o Eclipse 3.0.x. O arquivo da versão 1.4 é um tanto grande, cerca de 16 Mb.

O arquivo .zip que foi devidamente trazido do servido para o seu computador possui, após sua descompactação, duas pastas, uma chamada plugins onde o seu conteúdo deve ser colocado dentro da pasta plugins do Eclipse e, obviamente a pasta features que deverá ter seu conteúdo colocado dentro da pasta features do Eclipse.

Iremos agora configurar o Jboss-Ide, levando em conta que já se tem um Jboss instalado no computador, esta instalação também é bem simples, basta ir até o site do Jboss (www.jboss.org) e obter a ultima versão do Jboss, deve-se, após o download, descompactar o arquivo. Para se configurar o Jboss-Ide, iremos iniciar o Eclipse.

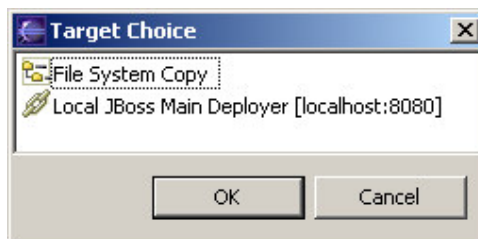
Após essa iniciação iremos até o menu Window / Preferences onde será iniciada uma janela de configuração do Eclipse, dentre essas configurações temos a possibilidade de configurar além do Jboss-Ide, opções do Eclipse, como indentação, configurações do plugin do Ant, parâmetros de compilação e da máquina virtual, configuração do CVS, etc.

Para configurarmos o Jboss-Ide iremos acionar o item Jboss-Ide / Deployer, nesta opção iremos configurar o local de deploy dos arquivos, apresentado na figura abaixo:

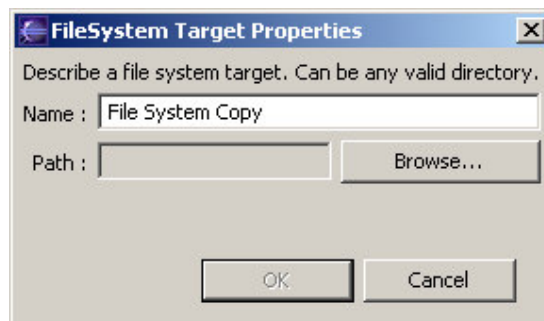


Interessante é notar que podemos criar vários lugares para deploy, escolhendo entre os arquivos na hora do deploy para onde cada um irá, desta forma podemos trabalhar em parceria com o Plugin do Tomcat para iniciar o Tomcat de dentro do Eclipse e realizar o deploy dos arquivos war (jsp) para dentro do Tomcat, iremos realizar mais adiante um exemplo desta integração.

Clique em Add, e selecione File System Copy e clique em OK.



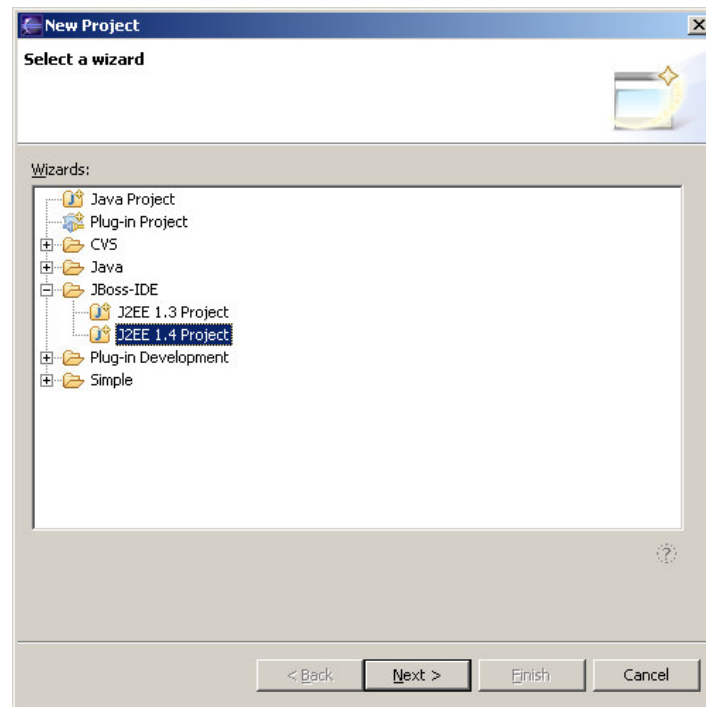
Clique em Browse, e selecione o caminho de deploy do Jboss, por exemplo, C:/jboss-3.2.5/server/default/deploy/, pode-se no Label Name, associar uma descrição a este tipo de deploy



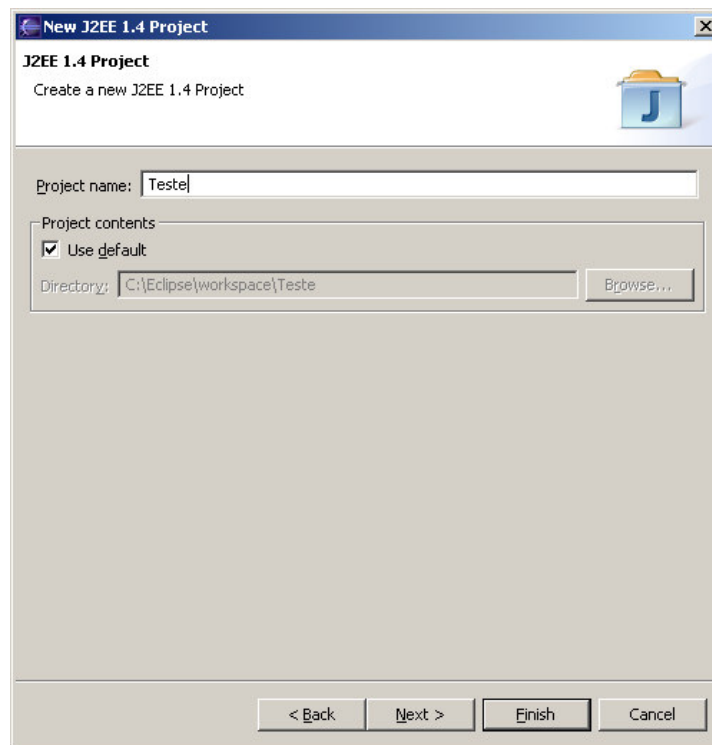
Pode-se desta forma criar vários locais de deploy, por exemplo, um para Ejb's e outro para JSP e/ou Servlets.

O Jboss-Ide permite a criação de EJBs, Servlets e JSP, onde para o JSP e HTML ele permite a exibição dos códigos com cores destacadas. Iremos construir um exemplo simples utilizando apenas um JSP, mas que dará uma visão inicial sobre o Plugin.

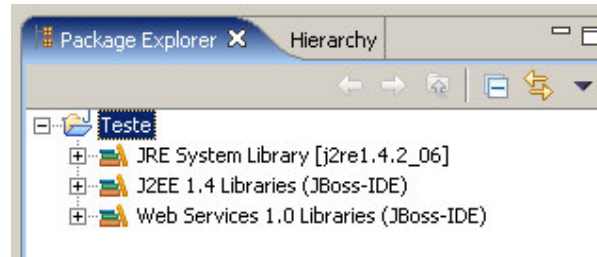
Inicialmente criaremos um novo projeto no Eclipse, para isto basta clicar em File / New / Project, nas opções de projeto selecionar Jboss-Ide e J2EE 1.4 Project (Lembre-se que projetos J2EE têm um poder muito maior do que o demonstrado neste exemplo, a idéia aqui é validar apenas o JSP).



Clique em Next e insira o nome do projeto, para este exemplo iremos nomear de Teste.



Clique agora em Finish, será criada a seguinte estrutura no Eclipse.



Iremos agora criar um arquivo JSP, para isto, clique com o botão direito sobre o projeto e selecione New / File. Nomeie o arquivo, neste caso iremos chamar de index.jsp e clique em Finish.



Iremos inserir código a página agora, para isto, clique duas vezes sobre o arquivo Teste.jsp e inclua o seguinte código:

```
<HTML>
<HEAD>
<TITLE>Um Exemplo em JSP</TITLE>
</HEAD>
<BODY>
<%! private String mensagem; %>
<%mensagem = new String("JBoss – IDE !");%>
<H1>Este eh um teste com o <%= mensagem %> </H1>
</BODY>
</HTML>
```

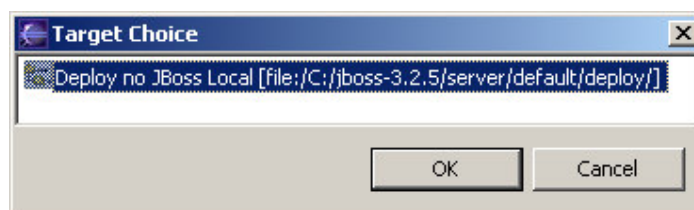
Salve o arquivo. Iremos configurar o empacotamento do arquivo para o deploy no Web Container. Para isto clique no projeto e tecle Alt + Enter, este procedimento acionará a janela de propriedades do projeto, selecione a opção Packaging Configurations, clique com o botão direito e selecione Add Archive, insira no nome do Arquivo Teste.war e clique Ok.



Clique agora com o botão direito do mouse sobre o Arquivo Teste.war, clique em Project Files e selecione Add File, e selecione o arquivo index.jsp. Clique agora com o botão direito do mouse sobre o projeto e selecione Run Packaging, aguarde o fim do processo.

Iniciaremos agora o Jboss, entre no diretório do Jboss / bin e acione o arquivo run.bat ou run.sh, aguarde o fim da inicialização.

Agora clique com o botão direito sobre o projeto e selecione Deployment / Deploy to, selecione a opção cadastrada e clique em OK.



Aguarde a finalização do deploy, abra o seu Browser e digite na URL <http://127.0.0.1/Teste>.

A idéia aqui é mostrar apenas um pequeno exemplo com o Jboss-Ide, que facilita ainda a criação de EJBs e servlets.

OMONDO PARA UML

O Omondo é um plugin free para o Eclipse que permite a construção de diagramas UML, desde diagramas de sequência até diagramas de classes, iremos demonstrar todo o processo de instalação deste plugin bem como a criação de um diagrama de casos de uso como um diagrama de classes.

Para realizar a instalação do Omondo devemos obter os arquivos para instalação do site <http://www.omondo.com/download/free/index.html>, devemos realizar o download dos seguintes arquivos disponíveis nesta página :

- Eclipse 3.0.1 Build R-3.0.1-200409161125 – Este é a instalação do Eclipse, caso você já tenha o Eclipse instalado não é necessário obter este arquivo
- EMF 2.0.1 Build 20040917 1617 – Este é o plugin de Framework do Eclipse, é necessário baixar este arquivo mesmo que você já tenha o Eclipse instalado.
- GEF 3.0.1 Build 20040831 1615 – Este é o plugin que permite manipulações gráficas dentro do Eclipse
- UML2 1.0.1 Build 20040721 – Este é o plugin para manipulações de arquivos no padrão UML 2
- E finalmente o arquivo Eclipse Uml que é um jar com a instalação do Omondo.

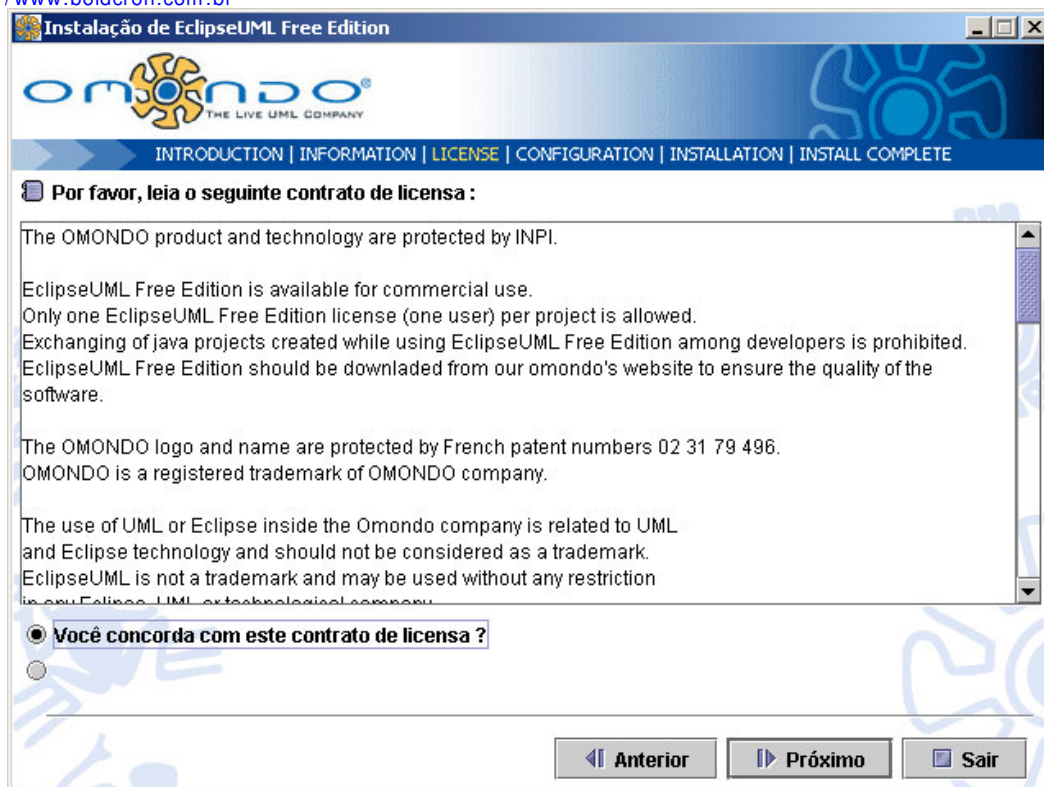
A instalação é novamente simples, descompacte os arquivos Eclipse 3.0.1 Build R-3.0.1-200409161125, EMF 2.0.1 Build 20040917 1617, GEF 3.0.1 Build 20040831 1615, UML2 1.0.1 Build 20040721 para os diretórios de plugin e features, de acordo com a configuração de cada um deles. Após isso, execute o arquivo do Eclipse UML, bastando para isso 2 cliques ou em linha de comando:

```
java -jar eclipseUML_E301_freeEdition_2.0.0.beta.20041026.jar
```

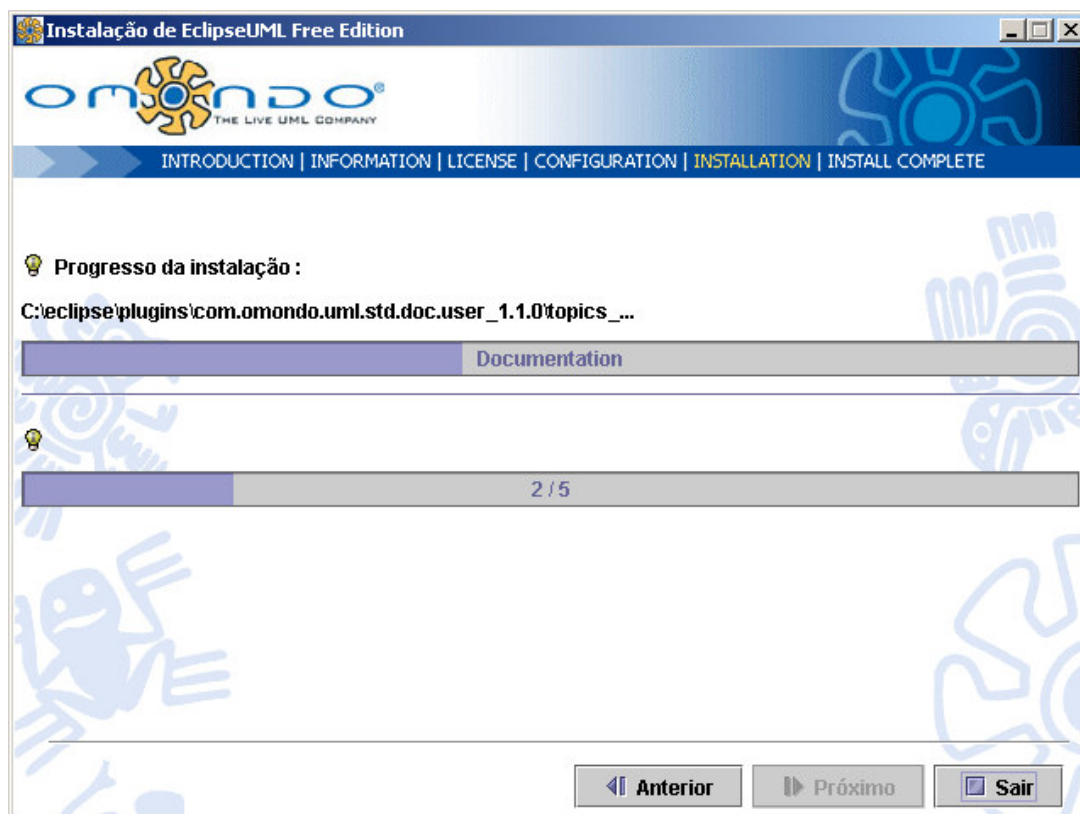
Será apresentada a tela abaixo, selecione o idioma e clique em OK



Clique em Próximo na tela de Introdução, Próximo na tela de informação e aceite o contrato e clique em Próximo novamente.



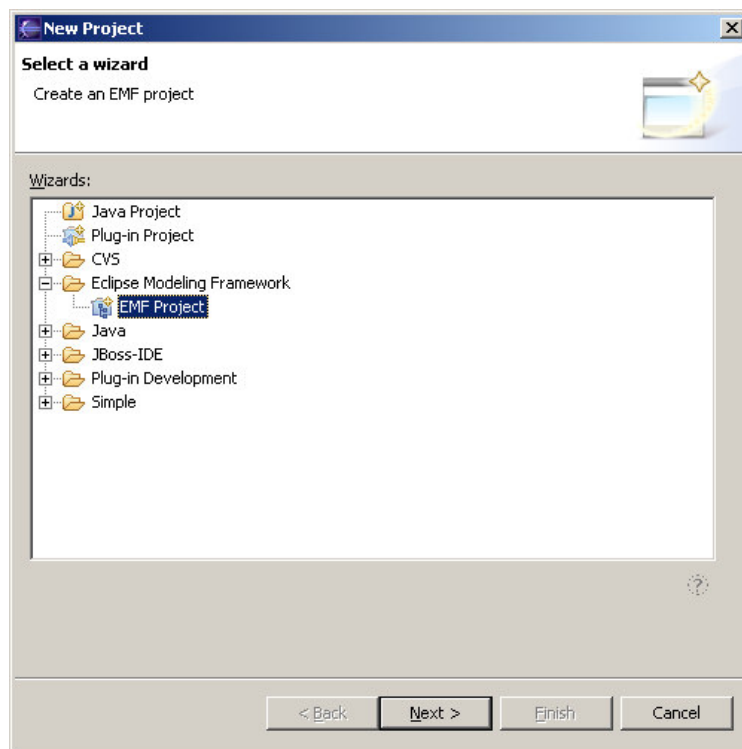
Selecione agora o diretório onde o Eclipse está instalado, clique em Próximo e aguarde a instalação.



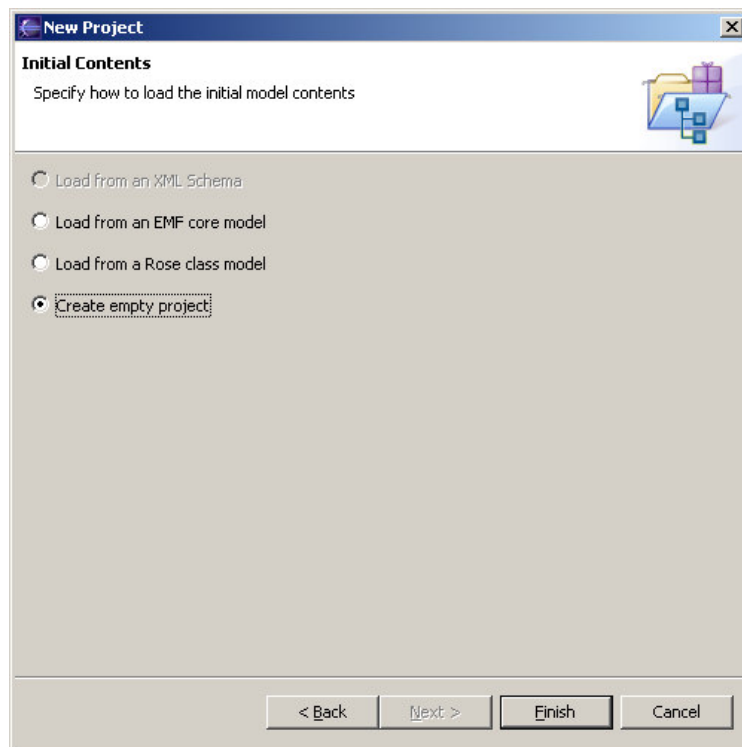
Após a instalação, clique em Próximo e Sair.

Iremos agora iniciar a criação de alguns diagramas no Omondo:

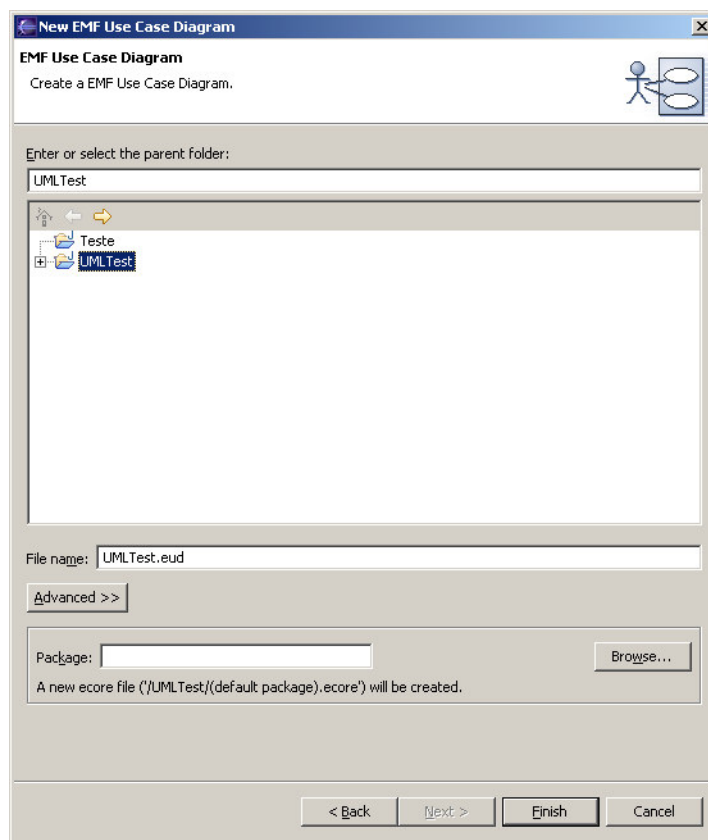
Clique em File / New Project e selecione Eclipse Modeling Framework e EMF Project.



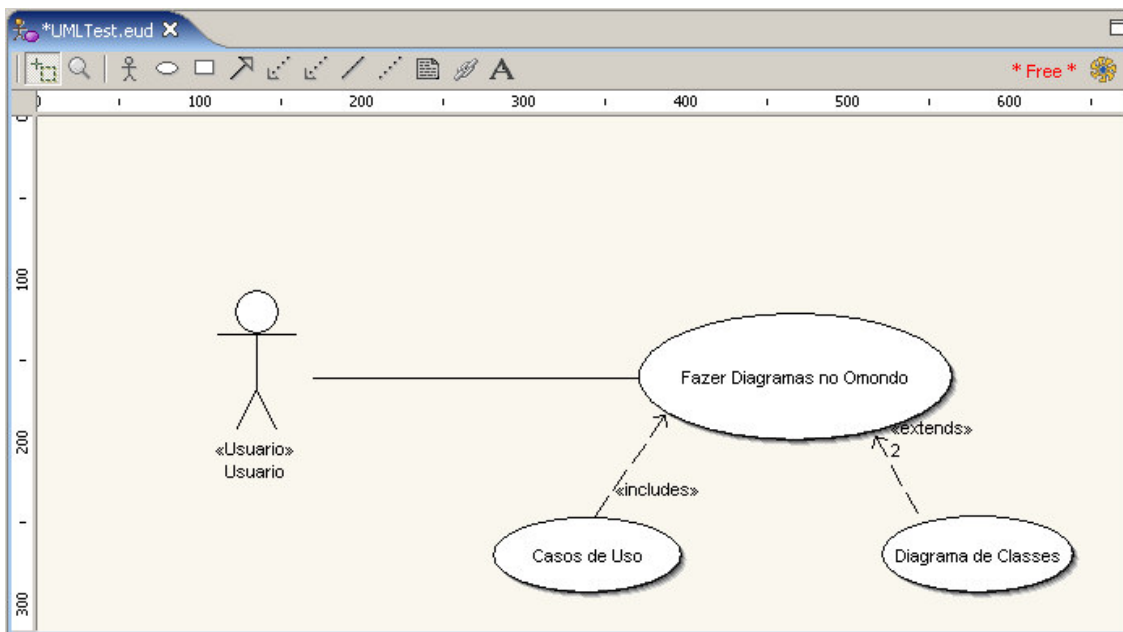
Coloque um nome no projeto como UMLTest , clique em Next, agora clique em Create a empty project e em Finish.



Clique com o botão direito do mouse sobre o projeto e selecione New / Other, selecione EMF Diagrams e EMF Use Case Diagram. Selecione o local onde o arquivo ficará e o nome para o Diagrama e clique em Finish.



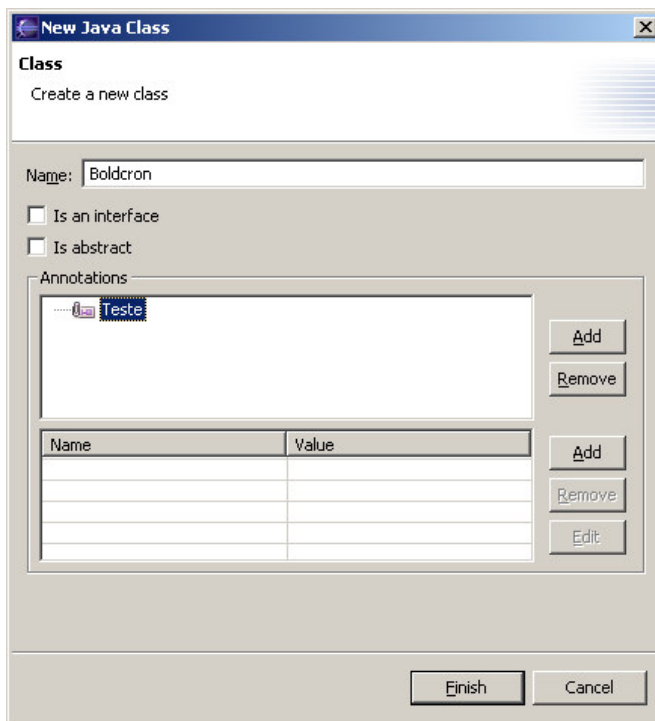
Podemos agora modelar um diagrama de casos de uso como o apresentado abaixo, para gerar, basta arrastar e soltar os elementos que estão na barra do Eclipse, veja o exemplo abaixo, este modelo é apenas para exemplo, não estando correto as coligações entre os sistemas.



Vamos criar agora um diagrama de classes, para isto clique com o botão direito do mouse sobre o projeto e selecione New / Other, selecione EMF Diagrams e EMF Class Diagram. Selecione o local onde o arquivo ficará e o nome para o Diagrama e clique em Finish..

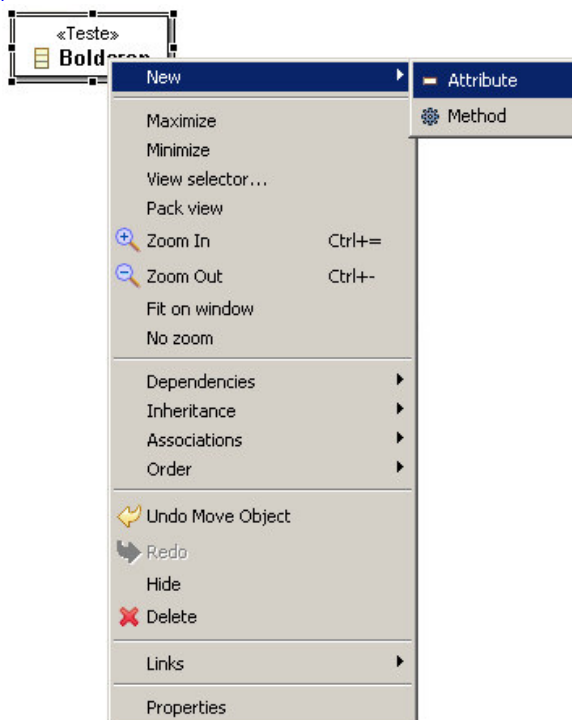
Devemos proceder da mesma forma do Diagrama de casos de uso, apenas arrastando os objetos para a área de diagramas.

A diferença é que quando incluímos uma classe no diagrama temos uma janela para configurações da classe, como se ela é na verdade uma interface ou é abstrata, como representado abaixo:



Name	Value

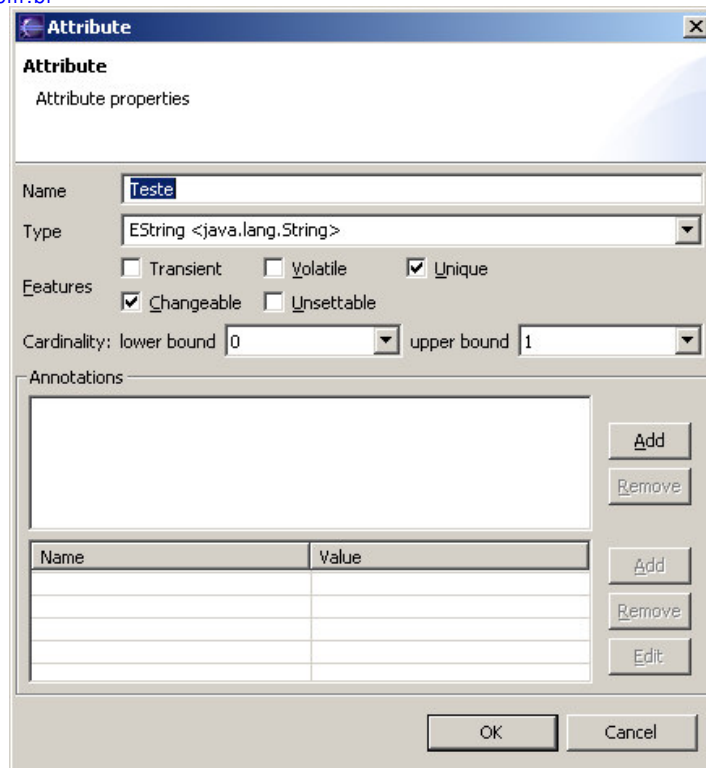
Para incluir um método ou um atributo, clique com o botão direito na classe e selecione New / Method ou New / Attribute



Selecione Method, preencha as opções do seu método, como descrito abaixo:

Name	Value

Pode-se também, incluir um atributo da classe, conforme tela abaixo:



Pode-se ainda, gerar engenharia reversa de projetos já prontos, para isto, clique com o botão direito sobre o projeto, selecione UML / Reverse Engineering, selecione os pacotes que se deseja fazer a engenharia reversa e clique em Finish que o Omondo fará o trabalho sujo por você.

Esse é uma rápida visão sobre o que o Omondo pode fazer, mas ele é muito mais poderoso que essa pequena introdução. Além desses dois diagramas é permitido criar diagramas de sequência, colaboração, componente, deployment, objetos, etc, etc, etc.

VEP – Visual Eclipse Project

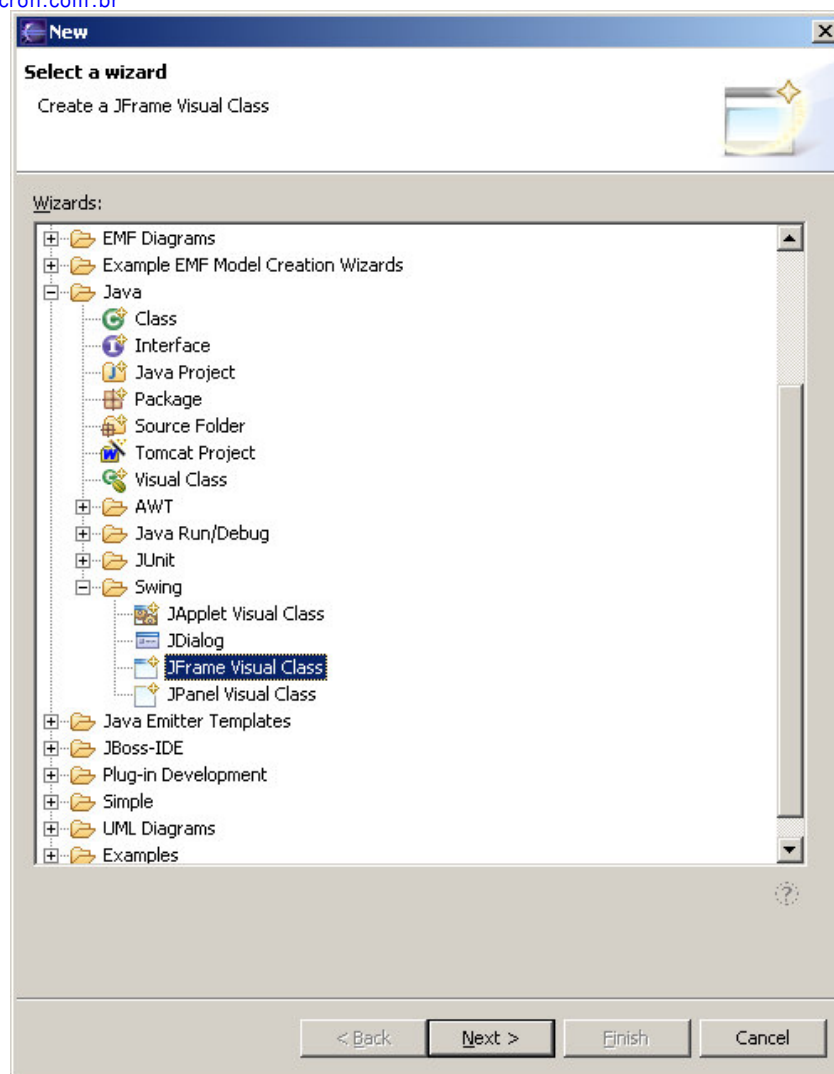
Este plugin, permite a criação de interfaces gráficas em Swing e SWT no mais fácil modo arrastar soltar, este projeto pertence ao grupo Eclipse.org e seu arquivo deve ser baixado em <http://www.eclipse.org/vep/>.

Para a instalação do VEP, ainda deve-se ter instalado o GEF e o EMF que podem ser encontrados na mesma página do GEF.

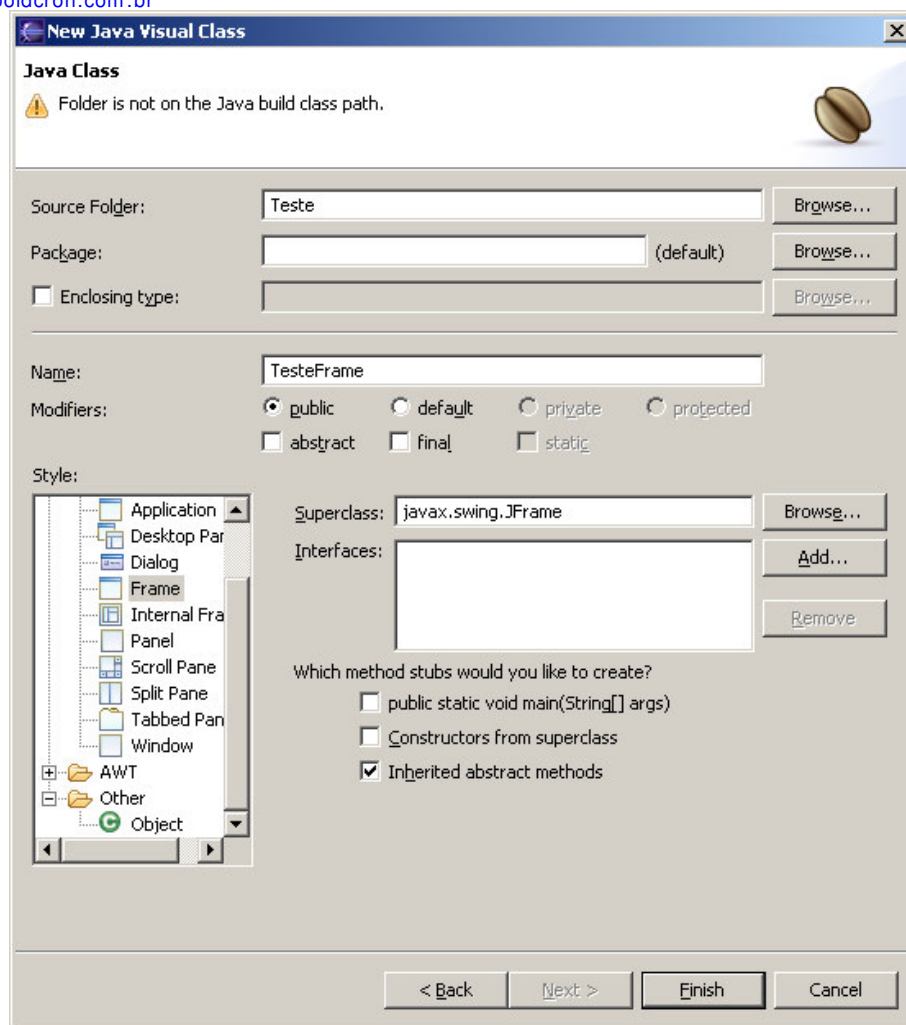
Para instalar o VEP, basta descompactar seus arquivos no diretório de plugins e de features do Eclipse.

Inicie o Eclipse e clique em File / New / Project e selecione Java Project, insira um nome para o projeto, tal como SwingTest.

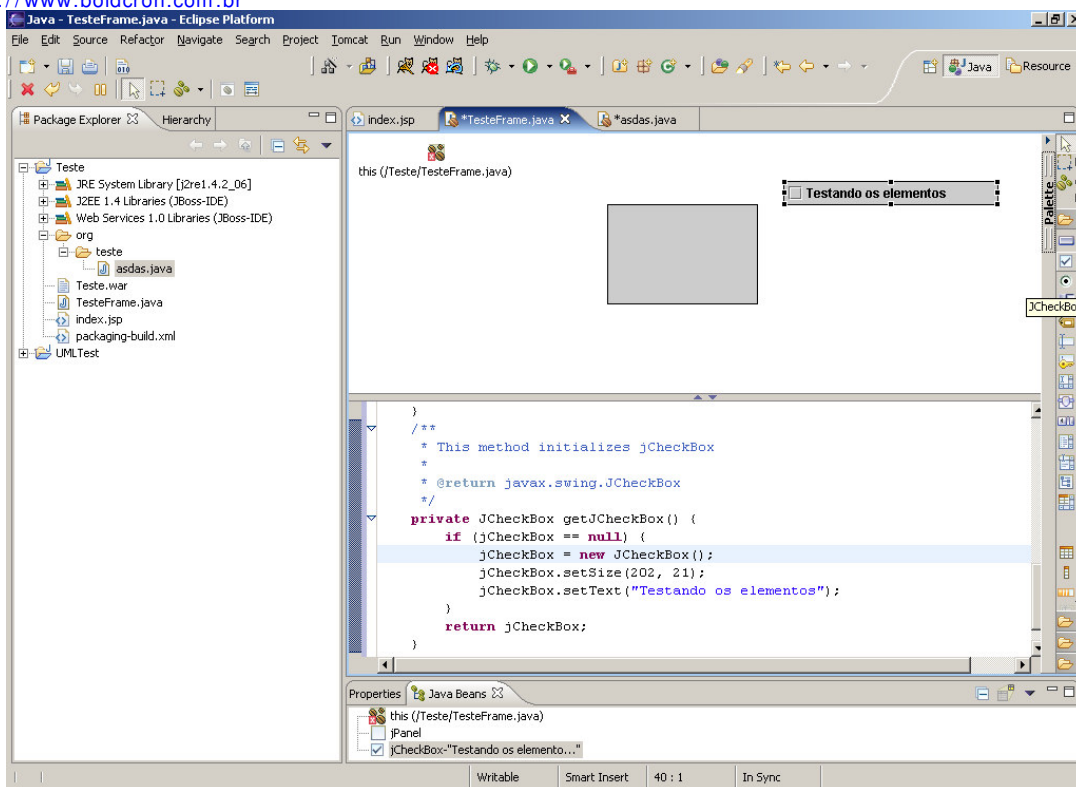
Crie uma nova classe neste projeto, clicando com o botão direito do mouse sobre a pasta e clique em New / Other, selecione o JFrame Visual Class:



Devemos agora configurar a classe visual com a janela abaixo:

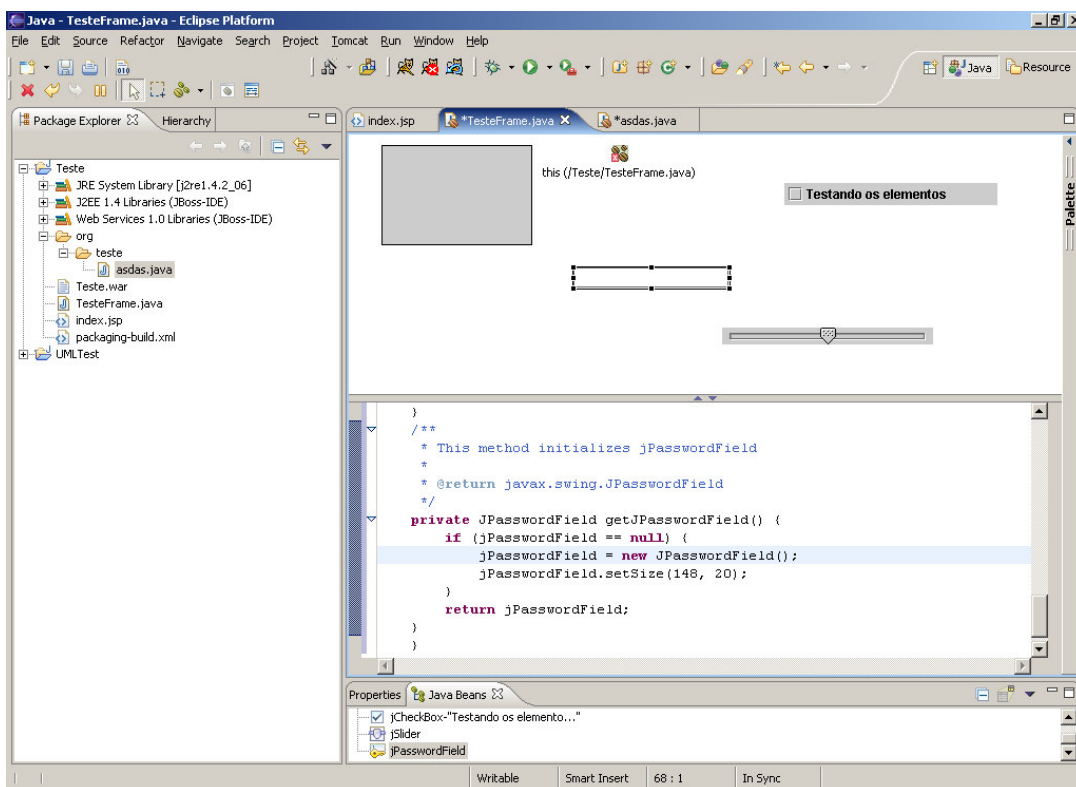


Agora podemos arrastar elementos do Swing para o nosso Frame, conforme figura abaixo:



Devemos dar destaque a uma informação importante, mesmo utilizando esse plugin deve-se conhecer bem Swing/AWT/Swt pois a utilização deste plugin não dispensa o conhecimento destas tecnologias.

Exemplo de um Frame montado com elementos Swings em DragDrop.



Conclusão

Através deste pequeno tutorial, podemos verificar que o Eclipse permite facilmente a instalação de plugins, devendo-se tomar especial cuidado com as versões do Eclipse e dos plugins, pois muitos plugins da versão 2.x não funcionam na versão 3.x.

Além disto, podemos verificar que o Eclipse não perde em nada para IDE pagas e, se bem configurado e utilizando os plugins corretos, pode ser uma ótima alternativa para equipes que pretendem migrar de IDEs.

Alguns outros itens que não foram abordados, mas que serão descritos em outro artigo é a praticidade de utilizar as perspectivas de projeto, permitindo assim que usuários de outras IDE customizem a visualização de seus projetos de acordo com o seu interesse, tornando suave a transição entre uma IDE e outra.

Rogério Peres Gomes (rogerio@boldcron.com.br) é Bacharel em Análise de Sistemas pela Universidade Paulista – UNIP e cursa matérias de especialização para o mestrado no Instituto Tecnológico Aeroespacial – ITA. Trabalha com desenvolvimento de software há 9 anos e tem como especialidades desenvolvimento e análise de sistemas orientados a objeto. Estuda Java há 2 anos. É integrante ativo do Grupo de usuários Java de São José dos Campos (www.sjcjug.org). Trabalha atualmente como desenvolvedor na Boldcron (www.boldcron.com.br), empresa de tecnologia especializada em aplicações Java.