

Eclipse: Criando e disponibilizando plugins

Heitor Luis Belloni Silva Rodrigues

Raphael Ernani Rodrigues

Tarcísio Guerra Savino Filó

Resumo

O Eclipse é uma poderosa plataforma para desenvolvimento de software, que é livre e permite ter suas funcionalidades ampliadas com facilidade, por meio de complementos, conhecidos como plugins. Este trabalho tem como objetivo mostrar o caminho a ser seguido por quem tem interesse de criar e disponibilizar para a comunidade seus próprios complementos.

Palavras-chave: Software Livre, Eclipse, Plugin.

Introdução

Desde o surgimento do computador, o homem tem utilizado essa ferramenta para realizar tarefas que a mente humana não pode realizar sem auxílio. Muitas dessas atividades são atividades que exigem memorização de grandes volumes de informações ou atividades que exigem muitas repetições de atividades simples. Com base nesses princípios, o homem começou a automatizar uma série de tarefas por meio de programas, onde os usuários apenas informavam certos parâmetros e os programas faziam todo o trabalho de processamento de dados. Com o passar do tempo, os problemas foram se tornando cada vez mais complexos, assim como os programas que os resolviam. Hoje, temos uma infinidade de plataformas, linguagens, padrões e protocolos a serem atendidos, de forma que escrever um programa para resolver um problema ,é por si só, um problema cada vez mais complexo. Com o esse problema, vieram soluções das mais diversas naturezas. Uma delas é a construção de pequenos programas para auxiliar tarefas rotineiras de desenvolvimento. Esse tema será abordado de forma mais detalhada neste trabalho. Algumas plataformas, como o Eclipse, permitem ao programador desenvolver pequenos programas conhecidos como plugins, que complementam as funcionalidades da ferramenta de desenvolvimento, completando partes do código fonte, fazendo integração com controladores de versão, emuladores de dispositivos móveis, dando suporte a mais linguagens de programação, etc. Não há limites para esses complementos, pois é a necessidade e a criatividade dos desenvolvedores que são aplicados na hora de criar os complementos.

II – Java, Eclipse e Software Livre

Vindo para o contexto do software livre, encontramos com facilidade uma infinidade de sistemas livres construídos no eclipse, utilizando a linguagem Java. O Java se tornou popular por ser uma plataforma livre, que em 15 de novembro de 2006 passou a ser distribuída sob a licença GPL (General Public License, a licença de software livre mais utilizada no mundo) [2] e recebeu publicamente as boas vindas da Free Software Foundation, uma das instituições mais respeitadas no que diz respeito à luta pelo software livre [3].

Desenvolvedores não costumam gostar de escrever seus programas em blocos de notas e chamar o compilador por meio de linhas de comando complicadas. Para isso, existem IDEs, as interfaces utilizadas pelos programadores para escrever programas, executar testes, depurar a execução em busca de problemas, etc. Para o desenvolvimento de aplicações Java, o Eclipse é a IDE mais utilizada. Ela também é open source e permite que o desenvolvedor construa complementos, sem ter que alterar o código fonte do próprio Eclipse, que está disponível, caso o programador quiser fazer algo que não seja possível fazer apenas com complementos.

III – Preparando o Terreno

Feitas as apresentações, mãos à obra! O objetivo é desmistificar o que há por trás dos plugins e mostrar que com bem pouco é possível criar e distribuir plugins para a comunidade.

Um plugin para o Eclipse nada mais é que um programa Java compilado em um arquivo com extensão JAR e que segue certos padrões.

O próprio Eclipse pode ser preparado para auxiliar o desenvolvedor a criar seus próprios plugins. Para isso, basta instalar o plugin PDE (Plug-in Development Environment). Ele já vem instalado na versão SDK do Eclipse, pronto para ser utilizado.

Com o PDE instalado, é possível criar um projeto para o novo plugin. No Eclipse, basta ir em *File -> New -> Project -> Plug-in Development -> "Plug-in Project"*. A partir daí, basta dar um nome para o plugin e finalizar a criação do projeto. É possível criar o projeto a partir de um template, para facilitar o desenvolvimento, a partir de um modelo.

IV – Os elementos do Plugin

Um plugin para o Eclipse segue a regra básica de qualquer programa Java. Todo o código fonte fica encapsulado em pacotes e precisa implementar certas classes de forma que o programa do plugin possa ser chamado pela IDE.

O principal elemento do plugin é o arquivo plugin.xml. É esse arquivo que é responsável por descrever a estrutura do plugin. Esse arquivo é responsável por informar ao eclipse os pontos de extensão que o plugin adiciona, as classes desses pontos de extensão e todas as demais características do plugin.

Entre as principais tags do arquivo XML, se destacam as seguintes:

`<runtime>` : Define as configurações de tempo de execução do plugin. Pode definir, por exemplo, o nome do arquivo JAR em que o plugin será compilado.

`<extension>`: Define cada uma das extensões implementadas pelo plugin. O parâmetro `point`, dessa tag, define o tipo de extensão. Uma analogia que pode ser feita é que o `extension` é uma lâmpada e o `extension point` é o soquete adequado para conectar esse tipo de lâmpada [1].

Por exemplo: se um desenvolvedor deseja criar um plugin que tenha dentre as suas funcionalidades a inclusão de um tipo novo de wizard para ser escolhido quando se entra no menu *File -> New*, deve ser utilizada a seguinte tag:

```
<extension point="org.eclipse.ui.newWizards">
```

Dentro da tag `extension`, o desenvolvedor pode colocar outras tags, para informar qual o pacote e a classe que contém a implementação daquela extensão, em quais circunstâncias aquela extensão estará habilitada, etc. Cada tipo de `extension` deve implementar uma classe diferente. No exemplo do wizard de criação de arquivo, por exemplo, a classe a ser implementada seria uma classe que herda os métodos e atributos da classe *org.eclipse.jface.wizard.Wizard*.

`<requires>` Indica uma lista de plugins necessários para o seu plugin funcionar.

Essa tag define quais os plugins são necessários para que o plugin que está sendo desenvolvido funcione corretamente. No exemplo a seguir, o trecho XML indica que 4 plugins são necessários para o correto funcionamento do plugin a ser construído:

```
<requires>
  <import plugin="org.eclipse.ui"/>
  <import plugin="org.eclipse.core.resources"/>
  <import plugin="org.eclipse.jdt.core"/>
  <import plugin="org.eclipse.jdt.ui"/>
</requires>
```

Informações mais detalhadas a respeito das regras para a criação de plugins podem ser encontradas na documentação oficial do eclipse, na internet, através do site <http://help.eclipse.org>. Lá, basta procurar por "Platform Plug-in Developer Guide".

V – Distribuindo o plugin para a comunidade

Após criar e testar o plugin, há diversas maneiras de disponibilizar o plugin para a comunidade. Uma maneira bem fácil de fazer isso é submetendo o programa com os fontes para o site www.sourceforge.net. O site armazena os programas de forma gratuita e os disponibiliza para qualquer pessoa no mundo, permitindo que o software receba comentários dos usuários. A divulgação pode ser feita em redes sociais especializadas em software livre, como a Software Livre Brasil, acessível por meio do site <http://www.softwarelivre.org>.

VI – Considerações finais

Hoje em dia, criar e distribuir software livre é muito fácil. Há cada vez mais recursos acessíveis de forma simples, à mão do desenvolvedor. As plataformas são livres, a documentação está aberta e sempre há pessoas em fóruns e redes sociais querendo ajudar quando aparece alguma dúvida. Além disso, publicar esses programas também é uma tarefa bem simples, o que favorece o surgimento de cada vez mais sistemas ou complementos livres.

Referências Bibliográficas

- [1] Autor Desconhecido. Extensions and Extension Points. Disponível em: <<http://help.eclipse.org/galileo/topic/org.eclipse.pde.doc.user/concepts/extension.htm>> Data de acesso: 28/10/2011
- [2] – Free Software Foundation. GNU GENERAL PUBLIC LICENSE. Disponível em: <<http://www.gnu.org/licenses/gpl.html>>. Data de acesso: 27/10/2011
- [3] Matt Lee. Sun begins releasing Java under the GPL. Disponível em: <<http://www.fsf.org/news/fsf-welcomes-gpl-java.html>>. Data de acesso: 27/10/2011