

# Evolução e Principais Características do IDE Eclipse

Fernanda B. Faria<sup>1</sup>, Priscila da S. N. Lima<sup>1</sup>, Luiz G. Dias<sup>1</sup>, Andrea A. Silva<sup>1</sup>,  
Mayara P. da Costa<sup>1</sup>, Thiago J. Bittar<sup>2</sup>

<sup>1</sup>Departamento de Ciência da Computação – Universidade Federal de Goiás  
CEP: 75704-020 – Catalão – GO – Brasil

<sup>2</sup>ICMC – Universidade de São Paulo  
Caixa Postal: 668 – São Carlos – SP – Brasil

ferrbontempo@gmail.com, priscila.sn1@gmail.com, gustavodias@msn.com,  
andrea.almeida8000@hotmail.com, mmayarapires@hotmail.com,  
jabur@icmc.usp.br

**Abstract.** *This paper will approach general and individual aspects of the Eclipse. First, has a general boarding and compacts on free software (open source) contends its definition importance, advantages and use. Later, it will have then the general boarding of the eclipse with its evolution, main characteristics, tutorial, advantage and disadvantages. A comparison among other world-wide famous IDE, the Netbeans, highlighting similarities and differences, emphasizing that it is for the developer to choose the platform that suits you.*

**Resumo.** *Este artigo abordará aspectos gerais e individuais do Eclipse. Inicialmente tem-se uma abordagem geral e compacta sobre software livre (open source) contendo sua definição, importância, vantagens e utilização. Posteriormente, haverá então a abordagem geral do Eclipse com sua evolução, principais características, tutoriais, vantagem e desvantagens. E é feita uma comparação com outro IDE mundialmente famoso, o Netbeans, destacando semelhanças e diferenças, enfatizando que cabe ao desenvolvedor escolher a plataforma que mais lhe agrada.*

## 1. Introdução

O Eclipse é um IDE (*Integrated Development Environment*) que foi desenvolvido pela empresa americana IBM (*International Business Machines*), em novembro de 2001. A empresa investiu cerca de 40 milhões de dólares para o desenvolvimento de um IDE de código aberto (*open source*) e de grande usabilidade. Ao longo do tempo, o Eclipse tornou-se uns dos IDEs mais utilizados no mundo, especialmente por utilizar SWT (*The Standard Widget Toolkit*) como biblioteca gráfica. (Eclipse.org, 2010)

Uma das grandes propriedades do Eclipse é o desenvolvimento baseado em plug-ins. O projeto Eclipse foi e continua sendo um projeto tão grandioso que em janeiro de 2004 foi criada uma fundação sem fins lucrativos com membros que fazem o suporte do projeto Eclipse, a *The Eclipse Foundation*, e fornece em geral quatro serviços à comunidade Eclipse: IT (*Information Technology*) *Infrastructure* - que fornece aos usuários e projetistas do Eclipse uma tecnologia evolutiva e segura; *Intellectual*

*Property (IP) Management* - que coordena a propriedade intelectual do software, gerencia os serviços e a comercialização do mesmo; *Development Community Support* - que contribui de maneira direta com melhorias no desenvolvimento do software; e *Ecosystem Development* - onde a Fundação promove reuniões, conferências e afins para promover toda a comunidade Eclipse. (Eclipse.org, 2010)

Este artigo está organizado da seguinte forma: na seção 2 é feita uma breve abordagem sobre software livre, a seção 3 intitulada Eclipse possui subseções que explicam com detalhes o IDE, a 3.1 que trata a evolução, 3.2 que são as principais características do Eclipse, 3.3 sobre como funciona o IDE, na seção 4 é feito um comparativo com o Netbeans, na seção 5 são mostrados os resultados da análise realizada, na seção 6 é feita a conclusão do artigo e na seção 7 são mostradas as referências utilizadas.

## **2. Breve Abordagem sobre Software Livre**

Software Livre (*Free Software*), conforme a definição criada pela *Free Software Foundation* é o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição. A forma usual de um software ser distribuído livremente é sendo acompanhado por uma licença de software livre como a GPL (*General Public License*), e com a disponibilização do seu código-fonte.

Software Livre se refere à existência simultânea de quatro tipos de liberdade para os usuários do software, definidas pela *Free Software Foundation*. Esses quatro tipos de liberdade são: executar o programa, para qualquer propósito; estudar como o programa funciona, e adaptá-lo para as suas necessidades; redistribuir cópias de modo que você possa ajudar ao seu próximo; aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (Campos, 2006).

Um programa é software livre se os usuários possuem todas essas liberdades. Portanto, você deve ser livre para redistribuir cópias, seja com ou sem modificações, seja de graça ou cobrando uma taxa pela distribuição, para qualquer um em qualquer lugar. Ser livre para fazer essas coisas significa (entre outras coisas) que você não tem que pedir ou pagar pela permissão, uma vez que esteja de posse do programa.

A liberdade de utilizar um programa significa que qualquer pessoa física ou jurídica pode utilizar o software em qualquer tipo de sistema computacional, para qualquer trabalho ou atividade, sem que seja necessário comunicar ao desenvolvedor ou a qualquer outra entidade em especial (Alvarenga, 2009).

Ferramentas consideradas "gratuitas" demonstram ser muito eficientes em seus usos. Como exemplo, podemos citar, IDE de desenvolvimento Eclipse, que possui todas as liberdades de um software livre, e é bastante utilizada pelos desenvolvedores de software.

## **3. Eclipse**

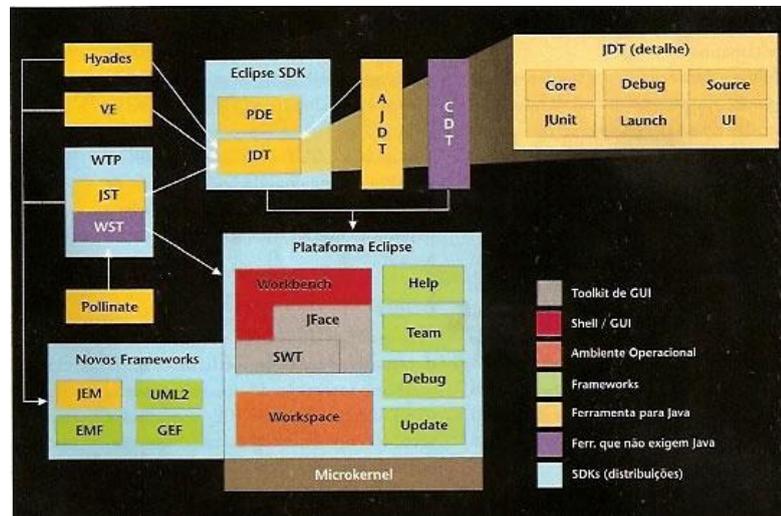
O Eclipse é bastante utilizado, por ser de fácil uso e também por ser software livre. Sua tecnologia baseada em plug-ins fornece um grande suporte aos programadores para realizar diferentes projetos de formas diversas.

### **3.1 Evolução**

O Eclipse começou seu desenvolvimento em 2001, e daí em diante nunca parou de evoluir. Suas versões têm nome de lua, sendo que a última chama-se Helios e está previsto o lançamento da versão Índigos para 2011, que será a versão 3.7 da plataforma.

Outras versões do Eclipse são: Eclipse 3.0 (2004), Eclipse 3.1 (2005), Callisto (2006), Europa (2007), Ganymede (2008), Galileo (2009).

As evoluções do Eclipse são possíveis e contínuas pelo fato de ser um a plataforma baseada em uma arquitetura de plug-ins, que permite ampliar entre outras, as funções gráficas. A arquitetura de plug-ins é mostrada na Figura 1, onde se pode visualizar a divisão dos componentes da plataforma.



**Figura 1: Representação da arquitetura básica de plug-ins, utilizada pelo Eclipse, controlada por um microkernel, ou seja, um único núcleo.**

Na figura 1 são mostrados os pacotes utilizados pelo IDE, como o SDK (*Software Development Kit*), o JDT (*Java Development Toolin*) e a relação existente entre cada componente da plataforma.

Os componentes genéricos do Eclipse, mostrados na figura 1 são: *Workspace*, que é gerenciador de diretórios e arquivos e abrange os projetos do usuário; *SWT* (*The Standard Widget Toolkít*) e *JFace*: são os gerenciadores de interface com os usuários; *WorkBench*: que funcionam como suporte e recursos de interface gráfica; *Team*: que interpreta *Workspace* para controlar versões e histórico dos recursos; *Debug*: que dá suporte a depuração de programas; *Help*: que é recurso de suporte aos usuários e *Update*: que gerencia as atualizações. (Trindade Júnior, Rios, Telles, 2008)

Externo a esses componentes, estão as ferramentas para Java, destacadas na figura 1 pela cor amarela, tais como o pacote JDT, que é um suporte a projetos Java; *PDE* (*Plug-in Development Environment*), que é suporte a plug-ins em Java; *JST*, que amplia o suporte voltado para desenvolvimento de *web*; *Hyades*, que é suporte para testes, entre outros componentes para Java. Na cor roxa, estão as ferramentas que não exigem Java, tais com o *WST* (*Web Standard Tools*), que dá suporte para desenvolvimento voltado para *web* e o *CDT* (*C/C++ development tools*), que dá suporta a projetos C/C++. Destacados pela cor verde estão os *frameworks*, tais como *EMF* (*Eclipse Modeling Framework*), que facilita a construção de geradores de código; *GEF* (*Graphical Editor FrameWork*), que facilita a construção de editores visuais a partir de um modelo de dados e *UML2*, que dá suporte à linguagem de modelagem *UML* (*Unified Modeling Language*). (Trindade Júnior, Rios, Telles, 2008)

Na cor laranja da figura 1 se destaca o ambiente operacional, na cor azul as distribuições do SDK, que é o kit de desenvolvimento de software.

### 3.2 Principais Características do Eclipse

O Eclipse é um IDE de código aberto utilizado para desenvolvimento da linguagem java, mas é também multilinguagem, ou seja, suporta outras linguagens de programação tais como C/C++ instalando-se os devidos plug-ins adicionais para cada linguagem. É um projeto livre de patentes por ser um software livre. E é portátil, ou seja, sua aplicação funciona em vários ambientes.

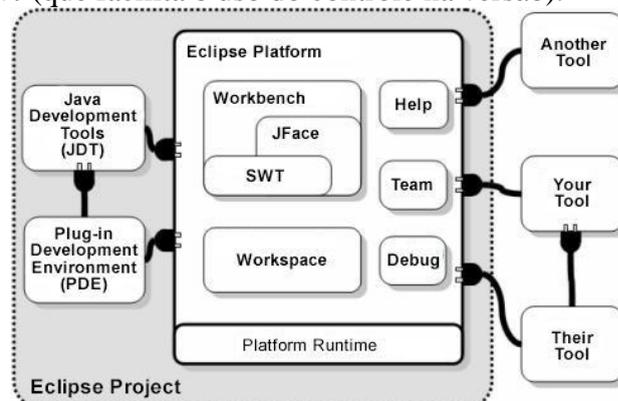
A plataforma do Eclipse fornece vários pacotes de desenvolvimento, tais como Eclipse JDT, que é a base para qualquer plug-in na linguagem java, o Eclipse SDK, que é o pacote de distribuição da IDE java, o Eclipse WTP (*Web Tools Platform*), que é usado para desenvolvimento de linguagem para web, e o compilador do JDT, que é seu próprio compilador java, que é mais rápido e de código aberto.

O Eclipse permite também a refatoração do código, que é uma forma organizada de reestruturar o código para ser melhorado. Um aspecto importante de uma refatoração é que ela melhora o design sem mudar a semântica do design, não adicionando nem removendo sua funcionalidade. Alguns exemplos de refatoração de código é renomear métodos, encapsular campos, extrair classes, introduzir afirmações e especializar os métodos (Santos e Eduardo, 2008).

Todo sistema operacional contém vários componentes que compõe sua interface de usuário. Estes componentes incluem botões, janelas, menus e tudo o que se vê na tela de computador. O Eclipse usa a plataforma SWT, que permite criar aplicações gráficas multiplataforma, como responsividade e aparência de aplicações nativas, sem sacrificar a compatibilidade (Lozano, edição 31). A meta do SWT é dar ao programador Java acesso direto a estes componentes de forma que se possa configurá-los e posicioná-los da forma que desejar. E o que é mais importante, é multiplataforma. A biblioteca SWT foi liberada em 2001, junto com o Ambiente Integrado de Desenvolvimento (IDE) do Eclipse. (Lozano, edição 31)

Algumas desvantagens do Eclipse, é que ele não possui uma versão online, ou seja, totalmente *web*, não é um software tão leve, a instalação de plug-ins pode não ser uma opção muito eficiente e pode causar grandes dificuldades ao usuário no início da utilização do software, algumas implementações podem ser difíceis de ser feitas devida a má instalação dos plug-ins.

A arquitetura do Eclipse, que será mostrada na figura 2, é de pequena *runtime*, e consiste em *Workbench* (que é a interface do Eclipse), *workspace* (responsável por administrar os recursos do projeto), *help* (que é um sistema de documentação extensível) e *team support* (que facilita o uso do controle na versão).



**Figura 2: Arquitetura da plataforma Eclipse.**

Além dos componentes genéricos que já foram mostrados na figura 1, a figura 2 mostra como eles se relacionam com outros componentes, tais como as ferramentas Java (JDT) que se capturam as informações das ferramentas de plug-ins (PDE) e as retorna para a plataforma, as informações das ferramentas de plug-in chegam também ao *Workspace*, pois este administra os recursos do projeto. Então, outras ferramentas

(*Another Tool*), as ferramentas do desenvolvedor (*Your Tool*), e as ferramentas do Eclipse (*Their Tool*), chegam à plataforma, de forma distribuída. As outras ferramentas se dirigem ao sistema de documentação extensível (*help*), as ferramentas do desenvolvedor ao suporte de controle da versão e histórico dos recursos (*Team*) e as ferramentas do Eclipse vão para as do usuário, como no momento de ajuda antes da execução do código e para o *Debug*, para detectar os erros existentes no código, durante a execução (Santos e Eduardo, 2008).

### 3.3 Como Funciona o Eclipse

O Eclipse não é um software que se instala, ele é obtido em forma de arquivo compactado<sup>1</sup>, que será descompactado após o download, e para executá-lo basta clicar no ícone com o nome de Eclipse. Após clicar neste ícone abrirá uma janela que pedirá para escolher o local no qual se deseja armazenar os arquivos do Eclipse, o chamado *workspace*. Após selecionar esse local clique em “ok” e a tela do Eclipse será aberta e daí em diante basta criar códigos-fonte no IDE. Para tal demonstração foi utilizada a versão Eclipse IDE for Java EE Developers (163 MB), que é considerada uma das mais completas (Campos, Leal, Henrique e Borba, 2008).

Depois de instalado, os projetos são criados normalmente, com suas classes, sendo feito os comentários para documentação e uma inovação do Eclipse que permite customizar o ambiente de trabalho.

Ele possui *package explorer* que é um visualizador simples e elegante dos projetos e sua navegação é semelhante ao *Windows explorer*, respeitando a hierarquia do projeto. O editor de texto que destaca as palavras-chave de Java<sup>TM</sup> é trivial em um IDE e facilita a leitura do código (Campos, Leal, Henrique e Borba, 2008).

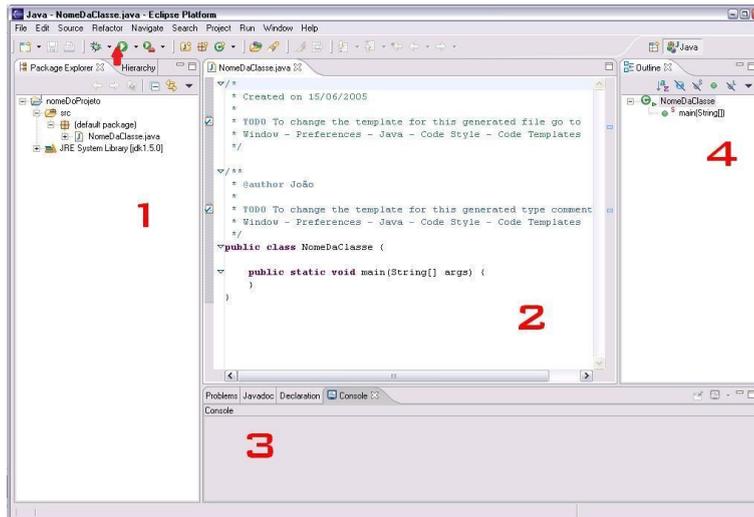
O Eclipse oferece um terminal de console para a entrada de dados do teclado assim como no JDK e possui uma janela *outline* que funciona semelhante ao *package explorer*, mas voltada para a estrutura interna do arquivo. Bibliotecas podem ser adicionadas ao projeto e pode-se importar e exportar arquivos do mesmo, não copiando os arquivos, mas criando um projeto que o referencia. (Campos, Leal, Henrique e Borba, 2008).

O programa é compilado normalmente e pode ser depurado para acompanhar todos os seus passos facilitando o encontro dos erros.

O Eclipse é um IDE de fácil entendimento e manuseio, podendo ser usada por qualquer pessoa que deseja se dedicar à programação. Sua interface é mostrada na figura 3, onde se destaca os principais componentes.

---

<sup>1</sup> <http://www.eclipse.org>



**Figura 3: Interface do Eclipse.**

Mostra-se na figura 3, uma visão geral do ambiente Eclipse, e detalhe de cada componente: 1– *Package Explorer*: que permite visualizar toda a estrutura de arquivos e diretórios contidos no projeto, 2 – Editor de texto: onde se escreve o código com algumas palavras destacadas que são palavras-chave do pacote Java e possui auto completar para as palavras já conhecidas, indica com um “X” vermelho a linha que contém erros e coloca e desenho de uma lâmpada caso o desenvolvedor necessite de ajuda, 3 – Terminal de Console: para entrada de dados do teclado e saída de dados que o programa escreve e 4 – Janela *outline*: visualizador das classes existente no projeto (Carlos, Leal, Henrique e Borba, 2008). A seta em vermelho indica onde se deve clicar para executar os códigos-fonte no Eclipse.

Na figura 4 estão representados os principais ícones encontrados no Eclipse IDE e seus respectivos nomes, a partir deles já se pode entender suas funções.

create, new	compare	forward	jar	plugin
save	debug	backward	WAR	extension
cut	run, execute	previous	EAR	extens'n point
copy	import	next	window	thread
paste	export	project	perspective	process
add	play, resume	open project	property sheet	mapping
remove	suspend	folder	table	error
delete	terminate	open folder	database	warning
erase, clear	stop	file	repository	alert
search	undo	library	class	conflict
find	redo	package	interface	public
help	refresh	session bean	attribute	protected
edit	filter	server	element	private
				default

**Figura 4: Principais ícones do Eclipse.**

#### 4. Eclipse X Netbeans

O Netbeans, assim como o Eclipse é um software livre, ou seja, de código aberto e, também multilinguagem. No Netbeans as aplicações são desenvolvidas com o auxílio do conjunto de componentes modulares.

Quanto à instalação, para o Netbeans é necessário apenas um download, pois ele é obtido em forma de software, e para o Eclipse são necessários seis downloads, para se obter todos os aplicativos (plug-ins) necessários para sua execução. A partir das expansões de cada um é possível perceber as diferenças existentes.

Para a interpretação do relatório de erros, o Eclipse possui uma ferramenta chamada “*manufacturer*”, que ao compilar o código contendo erros, o problema é mostrado com uma marca “X” e com isso pode-se saltar a linha específica no código-fonte onde o erro está marcado. Em contrapartida, o Netbeans não possui esta mesma característica em seu funcionamento e os erros são indicados na janela da saída. Os registros da configuração e os erros da compilação não são visualizados, mas quando o código-fonte é aberto alguns marcadores são mostrados para indicar tais erros (Silva, Coelho, D'el-Rey, Rocha, 2008).

No Eclipse é necessário fazer a instalação e configuração dos componentes que ele necessita, por *default* não pode explorar e editar uma base de dados. Enquanto o Netbeans fornece os módulos que contém esta funcionalidade básica por *default*, além de ser muito útil para as pessoas novatas que querem começar a aprender Java, sem perder tempo procurando os plug-ins necessários (Silva, Coelho, D'el-Rey, Rocha, 2008).

No Netbeans existe o “*Filesystem*”, que controla as informações da configuração através de um repositório que armazena todos os dados internos, enquanto o Eclipse armazena informações e dados usando um diretório situado fora do espaço de trabalho (Silva, Coelho, D'el-Rey, Rocha, 2008).

Esses IDE's possuem algumas semelhanças, tais como compilar, funcionar e debugar seus códigos, pois ambas possuem construtores de interface gráfica do usuário, abreviadamente em inglês, GUI (*grafical user interface*) e suportam as ferramentas ANT, CVS e JUnit, que são ferramentas Java (Silva, Coelho, D'el-Rey, Rocha, 2008).

## **5. Resultados da análise realizada**

Com esse estudo pode-se perceber o quanto um IDE bem estruturado e desenvolvido facilita o trabalho dos desenvolvedores de software. O Eclipse oferece várias facilidades e suporte para vários tipos de projetos o que o torna mais livre para o desenvolvedor usar a criatividade de forma simplificada.

No uso empresarial é de grande valia, pois oferece ferramentas que identificam erros com rapidez e eficiência, destaca as palavras reservadas dos pacotes Java e coloração do código-fonte para verificar se as palavras-chave estão corretas. Tudo isso torna mais fácil programar no IDE do que em editores comuns, auxiliando na entrega rápida e confiável do produto ao cliente.

O Eclipse evolui gradativa e constantemente, e em cada evolução novas ferramentas, plug-ins e pacotes, são adicionados para maior suporte aos desenvolvedores que precisam de mais recursos para desenvolvimentos mais complexos serem de fácil implementação.

O Eclipse IDE é uma excelente ferramenta para o ensino de linguagens possuindo muitos recursos que são fáceis de serem utilizados, pois cada parte do ambiente tem o nome e a usabilidade. É um software que pode ser utilizado por qualquer pessoa que deseja aprender linguagens de programação, visto que possui suporte para outras linguagens além de Java, que é seu grande foco.

## **6. Conclusão**

Neste artigo foi mostrado como é o funcionamento do Eclipse, sua instalação e utilização.

Como se pôde perceber o Eclipse é um IDE muito utilizado, por ser livre e de fácil manuseio, possuindo algumas diversidades, mas mesmo assim ainda é muito utilizado pelos desenvolvedores Java ou de outras linguagens por ser multilinguagem e por ser multiplataforma. Foi desenvolvido baseado na arquitetura de plug-ins, o que o torna mais dinâmico, para desenvolver diferentes tipos de projeto.

O grande concorrente do Eclipse é o Netbeans, no entanto suas características são bastante semelhantes, cabendo ao desenvolvedor escolher o ambiente que mais lhe agrada para desenvolvimento de projetos.

## 7. Referências Bibliográficas:

- ALVARENGA, D. W. “O poder do Software Livre no Desenvolvimento de Sistemas”, [http://www.softwarelivre.goias.gov.br/index.php?option=com\\_content&view=article&id=50:o-poder-do-software-livre-no-desenvolvimento-de-sistemas&catid=9:desenvolvimento&Itemid=2%20](http://www.softwarelivre.goias.gov.br/index.php?option=com_content&view=article&id=50:o-poder-do-software-livre-no-desenvolvimento-de-sistemas&catid=9:desenvolvimento&Itemid=2%20), março 2009.
- CAMPOS, A. “O que é software livre”, <http://br-linux.org/linux/faq-softwarelivre>, março 2006.
- CAMPOS, S., Leal, F., Henrique, J., Borba, P. “Introdução ao Eclipse”, <http://www.cin.ufpe.br/~phmb/ip/MaterialDeEnsino/IntroducaoAoEclipse/IntroducaoAoEclipse.htm>, 2008
- EDGAR, N., Haaland, K., Li, J., Peter, k. “Eclipse User Interface Guidelines” <http://www.eclipse.org/articles/Article-UI-Guidelines/Contents.html>, 2004.
- LOZANO, F. “Eclipse Visual com SWT”, revista Java Magazine, edição 31, <http://www.devmedia.com.br/articles/viewcomp.asp?comp=10157>
- MENS, T., Fernández-Ramil, J., DEGRANDSART, S. “The Evolution of Eclipse”, <http://umh.academia.edu/documents/0011/2632/ICSM2008-MensRamil.pdf>, 2008.
- OLIVEIRA, E. C. M. “Projeto Eclipse for Java”, <http://www.linhadecodigo.com.br/Artigo.aspx?id=677>, Abril, 2005.
- SANTOS, T. S., Eduardo. C, “Eclipse Tarde”, <http://pesquompile.wikidot.com/eclipse-t>, 2008.
- SILVA, M. A., Coelho, A., D'el-Rey, B., Rocha, G. “Eclipse X Netbeans”, <http://pesquompile.wikidot.com/eclipse-x-netbeans>, 2008.
- SILVA, R. “Projeto Eclipse”, <http://pt.shvoong.com/internet-and-technologies/computers/378130-projeto-eclipse/>, julho, 2006.
- TRINDADE Júnior, J. A., Rios, M, Telles, T. “Eclipse Noite”, <http://pesquompile.wikidot.com/eclipse-n>, 2008.
- WERMELINGER, M. e Yu, Y. “Analyzing the Evolution of Eclipse Plugins”, <http://mcs.open.ac.uk/yy66/paper/wermelinger08msr.pdf>, 2008.

[www.eclipse.org](http://www.eclipse.org), acessado em 6 de setembro de 2010.