

# Tutorial Eclipse (IDE) (Start)



# Sumário

- **Introdução**
  - **O que é o Eclipse?**
  - **Característica e Ferramentas**
  - **Download**
  - **Preparando Ambiente de Trabalho**
  - **Iniciando o Eclipse**
  - **Criando um Projeto**
  - **Criando um Programa**
  - **Primeiro \*.Class**
  - **Compilando \*.Class**
  - **Syntax Error**
  - **Refêrencias**
-

# Introdução

- Este tutorial foi desenvolvido com o intuito de ajudar estudantes como desenvolvedores iniciantes e aqueles que estão iniciando em nosso grupo de estudo a conhecer um pouco mais sobre Eclipse(IDE), utilizando como linguagem de desenvolvimento JAVA em ambiente Linux.
- Escolhemos o Open Source por ser de alta qualidade e por possuir uma comunidade fantástica. Sim nós amamos a Liberdade. =)
- Por favor não se prendam a este TUTORIAL, ele contém uma simples orientação.

“Um **eclipse** (do grego *ekleipsis*, "desvanecer") é um evento celeste que mais tem atraído a curiosidade humana com respeito a mecânica celeste.”

# O que é o Eclipse?

- O Eclipse que iremos falar aqui não é o evento celeste mas sim a IDE (**Integrated Development Environment**) ou seja Ambiente Integrado de Desenvolvimento.
- Eclipse é uma comunidade open source, cujos projetos estão focados em construir uma plataforma aberta desenvolvimento composta por ferramentas e runtimes para a construção, implantação e gestão de software em todo o ciclo de vida. O Eclipse é uma fundação sem fins lucrativos.
- O projeto Eclipse foi originalmente criado pela IBM em novembro de 2001 e apoiado por um consórcio de vendedores de software.
- Hoje o Eclipse é a principal IDE java e uma das suas principais característica é o desenvolvimento baseado em plug-ins auxiliando assim os desenvolvedores, tornando assim um ambiente prático de desenvolvimento e resolvendo a suas necessidades.

# Característica e Ferramentas

Uma IDE é um programa que reúne ferramentas de apoio ao desenvolvimento de software possibilitando uma agilidade maior no processo de construção de um software.

- **Editor** – edita o código-fonte do programa escrito
- **Compiler** – compila o código-fonte transformando em código de máquina.
- **Linker** – são os vários pedaços de códigos reunidos transformando em um executável.
- **Debugger** – processo de se localizar erros(famosos bugs) .
- **Modelling** - criação do modelo de [classes](#), [objetos](#), [interfaces](#), [associações](#) e [interações](#) dos artefatos envolvidos no [software](#).
- **Geração de código** - A IDE gera códigos(templates) comumente utilizados possibilitando algumas soluções.
- **Deploy** - auxilia no processo de criação do instalador do [software](#). (no nosso caso os famosos \*.jar).
- **Automated tests** - realiza testes no software de forma automatizada,
- **Refactoring** - consiste na melhoria constante do código-fonte do software, seja na construção de código mais otimizado,

# Característica e Ferramentas

Preciso pagar pela Ferramenta?

- **Não**, é uma ferramenta OPEN SOURCE conforme mencionado acima é um software livre.

Por que o Eclipse faz tanto sucesso?

- *“Na prática ele cria um modelo de Open Innovation Network, onde empresas concorrentes podem criar redes de inovação, cooperando no desenvolvimento de softwares Open Source...” - Cezar Taurion(IBM)*

## Download

- Para download acesse: <http://www.eclipse.org/downloads/>
- Escolha a opção Linux, conforme (fig. 1).
- Aparecerá varias opções em nosso caso utilizaremos
- **Eclipse IDE for Java Developers**, caso trabalhe com web escolha a opção **Eclipse IDE for java EE Developers**. Existe uma descrição abaixo do subtópico sobre a função de cada um.

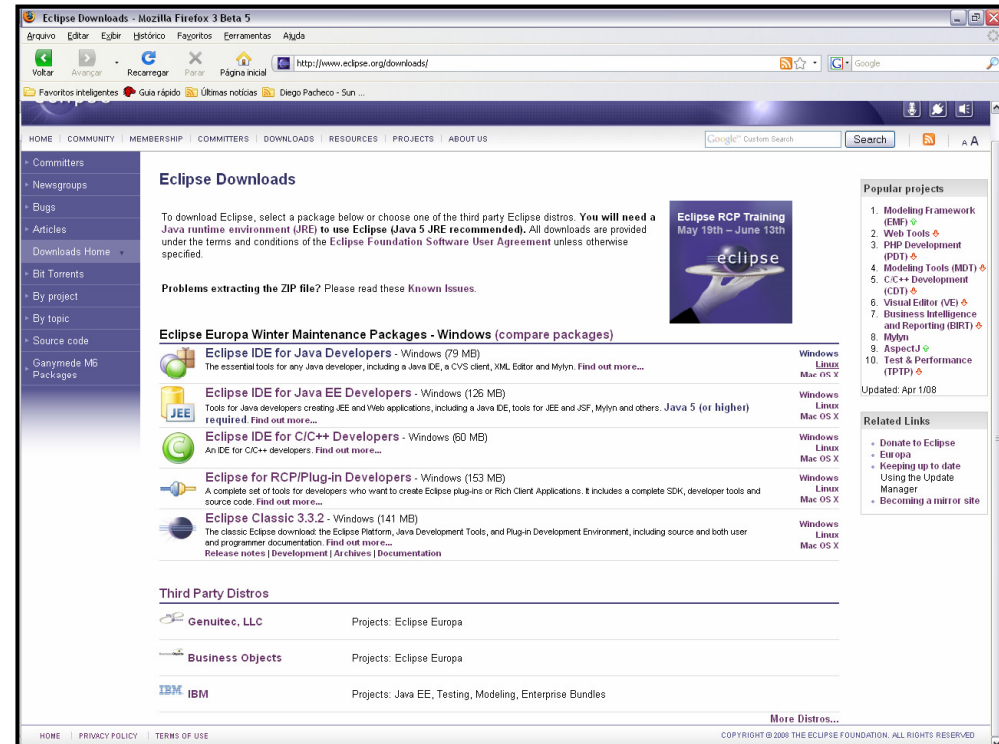


Fig. 1 – Download - <http://www.eclipse.org/downloads/>

## Preparando Ambiente de Trabalho

- Abra o terminal de sua Distribuição Linux. E descompactaremos dentro de uma pasta especifica. Copie o arquivo para à pasta: */opt/java*

-----*Todo Comando executado dentro do terminal*-----

▶ Criaremos a pasta java

```
# mkdir /opt/java
```

▶ Copiaremos o arquivo para pasta /opt/java

```
# cp eclipse-java-europa-winter-linux-gtk.tar.gz /opt/java/
```

▶ Acessaremos a pasta java

```
# cd /opt/java/
```

▶ Descompactaremos o arquivo

```
# tar -zxvf eclipse-java-europa-winter-linux-gtk.tar.gz
```

▶ Entre dentro da pasta Descompactada

```
# cd /eclipse-java-europa-winter-linux-gtk
```

▶ Para iniciar o eclipse pelo terminal

```
# ./eclipse
```



# Iniciando o Eclipse

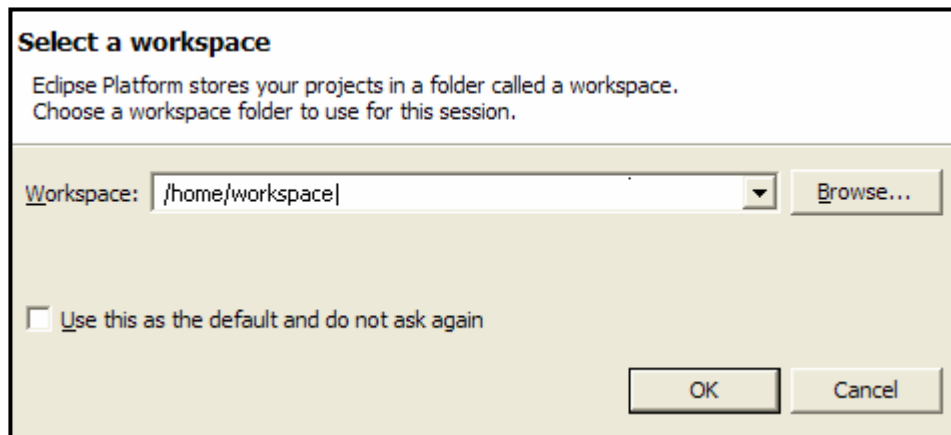


Fig.2 – Workspace Launcher

- Com o Workspace Launcher (*fig.2*) nós escolheremos o diretório que será armazenado os projetos.

- Assim que clicarmos em OK aparecerá o menu principal do *ECLIPSE*(*fig.3*).



Fig.3 – Menu Principal

## Criando um Projeto

- Para criar um projeto, escolha [ File ] – [ New ] – [ Java Project ] o seguinte menu surgirá (*fig. 4*).
- Defina o nome do projeto no nosso caso utilizaremos: dejavux
- **Project name: *dejavux***
- Click em NEXT >
- Click FINISH

Projeto Criado com Sucesso =)

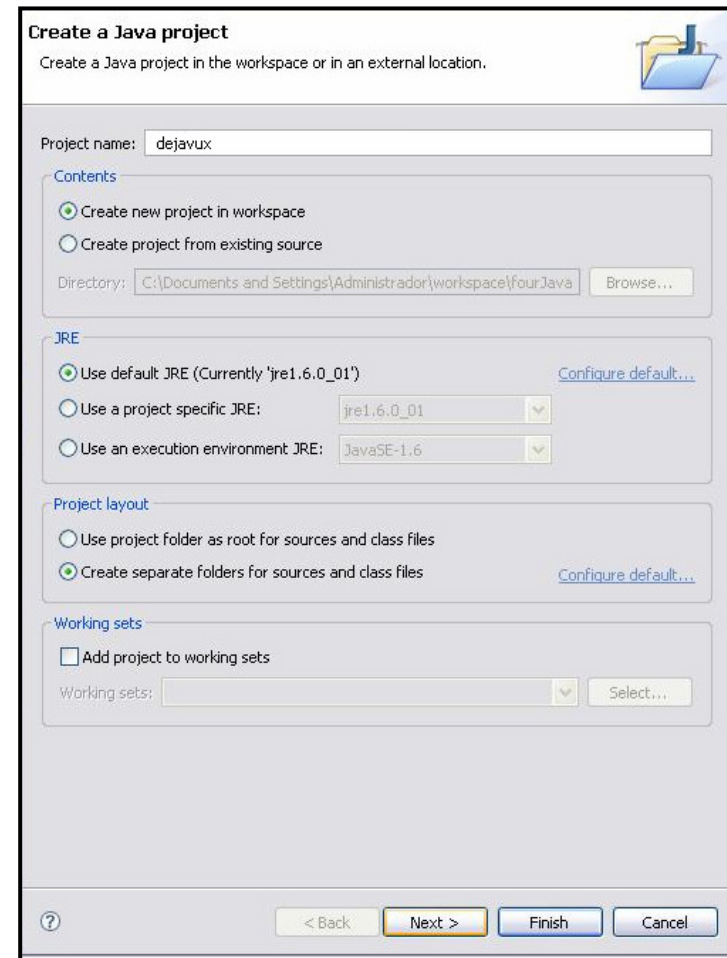


Fig 4 – Criando um projeto Java

# Criando um Programa

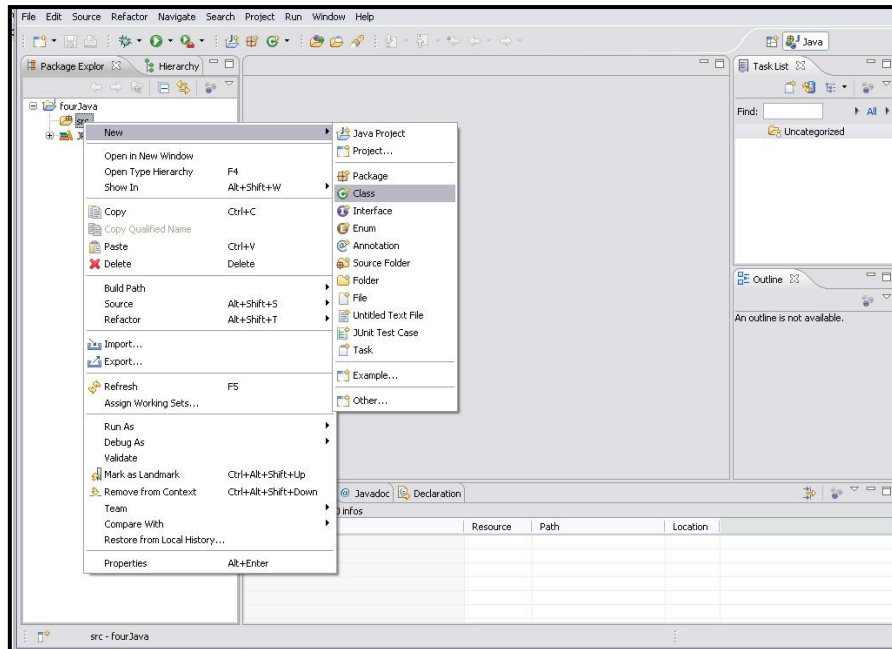


Fig. 5 – Menu : New ► Class

- A partir de agora criaremos o nosso primeiro class.
- Com o botão direito click sobre **src**
- Escolha a opção **New ► Class** (fig. 5).

# Primeiro \*.Class

Figura (fig. 6) mostra de forma detalhada a Criação de uma Classe Java.

- ▶ Define o nome do pacote que desejar.
  - **Package:** world
- ▶ Nome da Classe
  - **Name:** HelloWorld
- ▶ Marque esta opção para que possamos executar o programa
  - **[V]public static void main(String[] args)**

Para finalizar click em FINISH

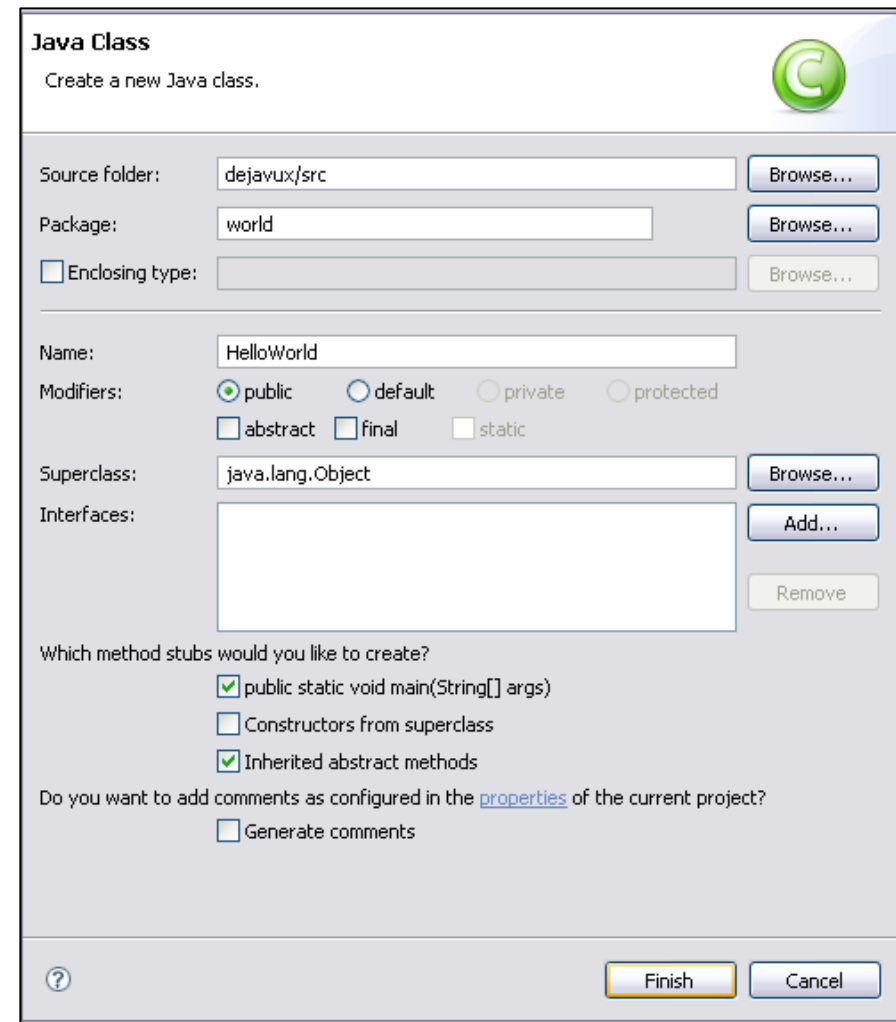


Fig.6 – Primeiro \*.class

# Primeiro \*.Class

- Ao lado observamos no Eclipse um Código Template(criado pela IDE) (fig. 7) e um definido por nós:

```
{
```

```
System.out.println("www.dejavuxteam.wordpress.com");  
System.out.println("Hello World");
```

```
}
```

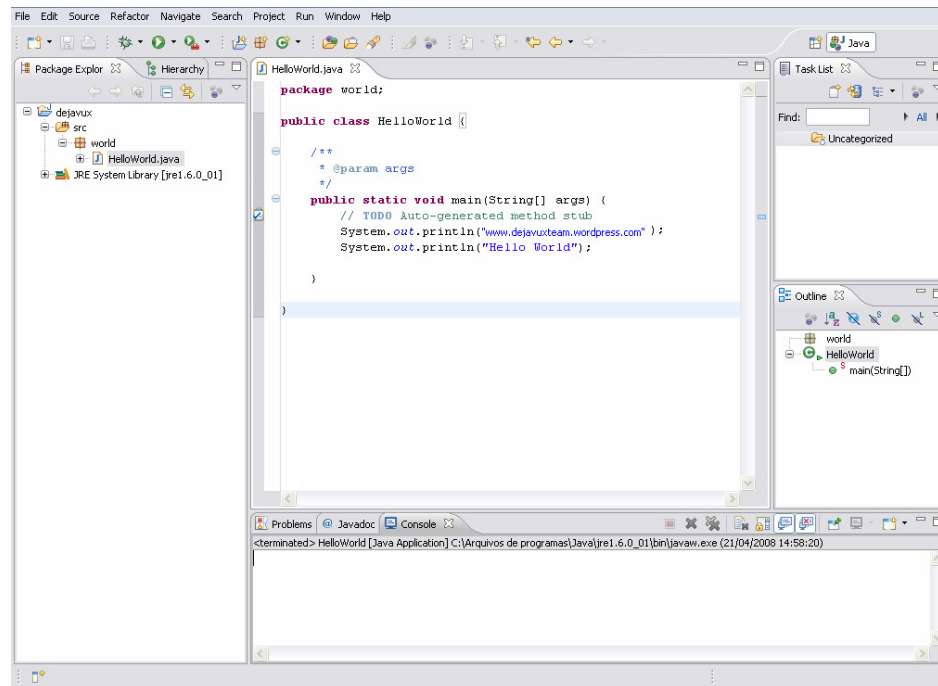


Fig. 7 – Código Template

## Compilando \*.Class

- Para compilar a class click:
- **Run ► Atalho (CTRL + F11).**

*Sua saída será mostrada no Console Pane.*

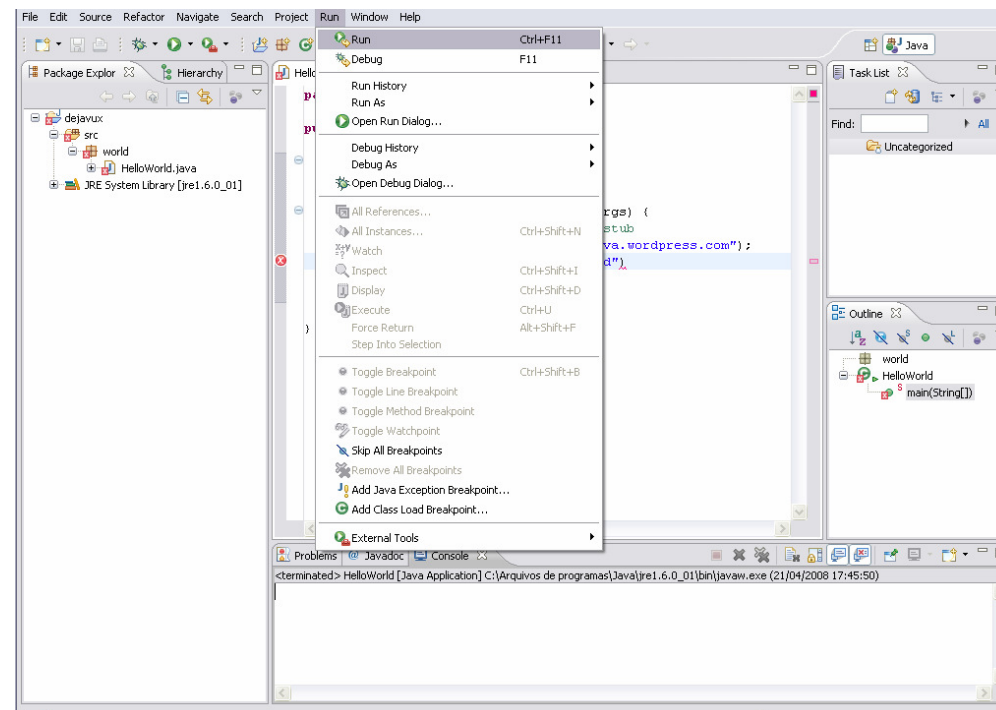


Fig. 8 - Run - Compilando Programa

# Syntax Error

- Por padrão, seu código é compilado dinamicamente .
- Por exemplo, se esquecer de colocar um *ponto e virgula (;)* no final do código.
- A IDE destacará em vermelho que ali existe um erro(*syntax error*).

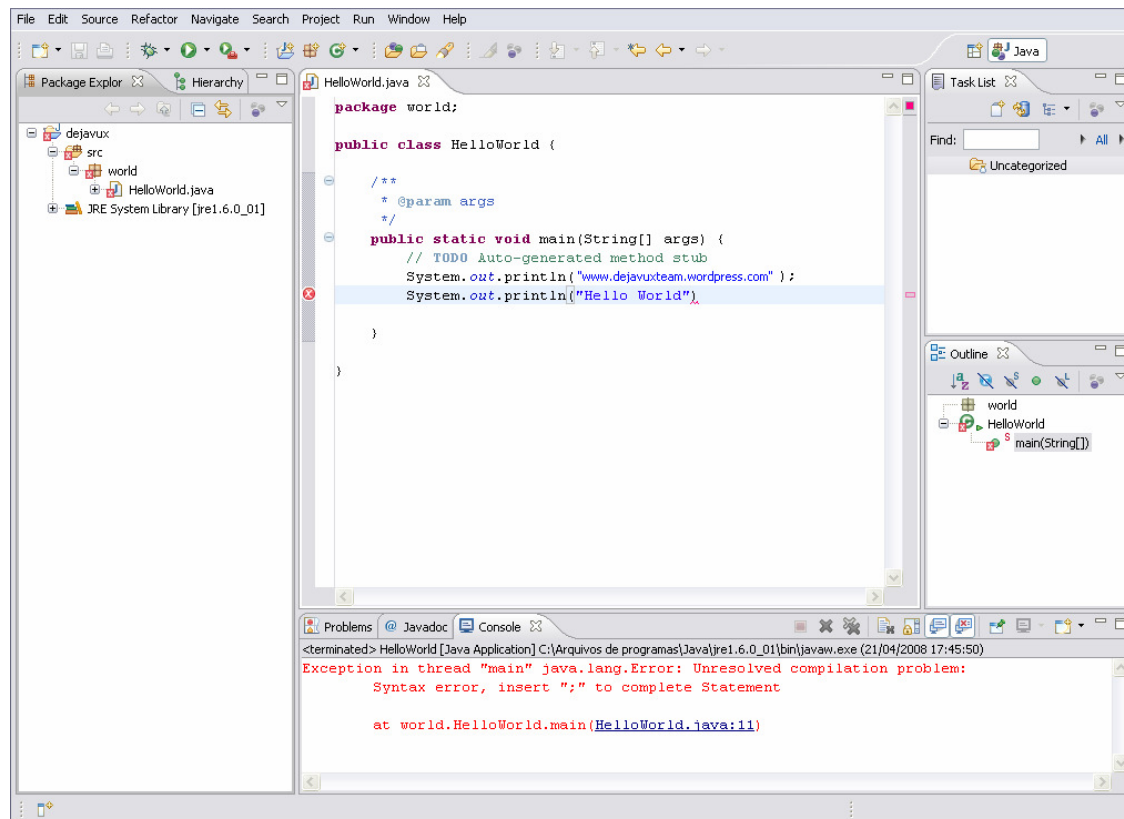


Fig. 9 – Eclipse checka dinamicamente Sintaxe que contém erro.

# Refêrencias

- Liang, Daniel, Eclipse Tutorial: a Supplement J, 2005
- Eclipse, In: Eclipse Foundation: Disponível em: <http://www.eclipse.org/> Acesso em 21 de abril 2008.
- Eclipse(IDE), In: Wikipédia: a enciclopédia livre. Disponível em: [http://pt.wikipedia.org/wiki/Eclipse\\_\(IDE\)](http://pt.wikipedia.org/wiki/Eclipse_(IDE)) Acesso em 20 de abril 2008.
- Eclipse, In: Wikipédia: a enciclopédia livre. Disponível em: <http://pt.wikipedia.org/wiki/Eclipse> Acesso em 20 de abril 2008.
- Erickson, Marc (IBM), What is Eclipse, and how do I use it? Disponível em: <http://www-128.ibm.com/developerworks/opensource/library/os-eclipse.html> Acesso 19 de abril 2008



**Dúvidas???**

Thank you! !!

Razec  
mrazec@gmail.com

