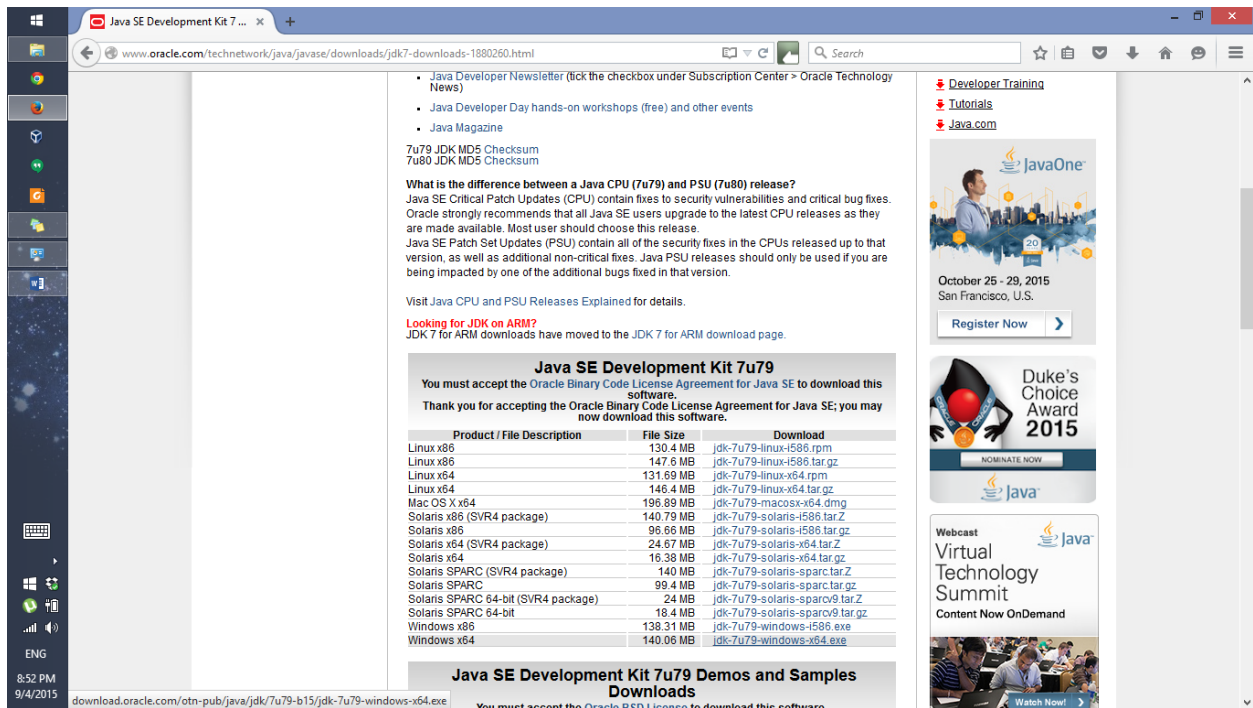


Tutorial de Instalação do Java 7 + Eclipse Mars e importação do projeto.

Primeiramente, dê um pull no repositório do git, pois a estrutura inicial do projeto estará lá. Siga as imagens abaixo.

Baixe a JDK 7 no site da Oracle. Usaremos o Java 7 e não o 8, pois o 8 lança updates a todo mês, e isso pode desestabilizar o projeto.



Java SE Development Kit 7u79

7u79 JDK MD5 Checksum
7u80 JDK MD5 Checksum

What is the difference between a Java CPU (7u79) and PSU (7u80) release?
Java SE Critical Patch Updates (CPU) contain fixes to security vulnerabilities and critical bug fixes. Oracle strongly recommends that all Java SE users upgrade to the latest CPU releases as they are made available. Most users should choose this release.
Java SE Patch Set Updates (PSU) contain all of the security fixes in the CPUs released up to that version, as well as additional non-critical fixes. Java PSU releases should only be used if you are being impacted by one of the additional bugs fixed in that version.

Visit Java CPU and PSU Releases Explained for details.

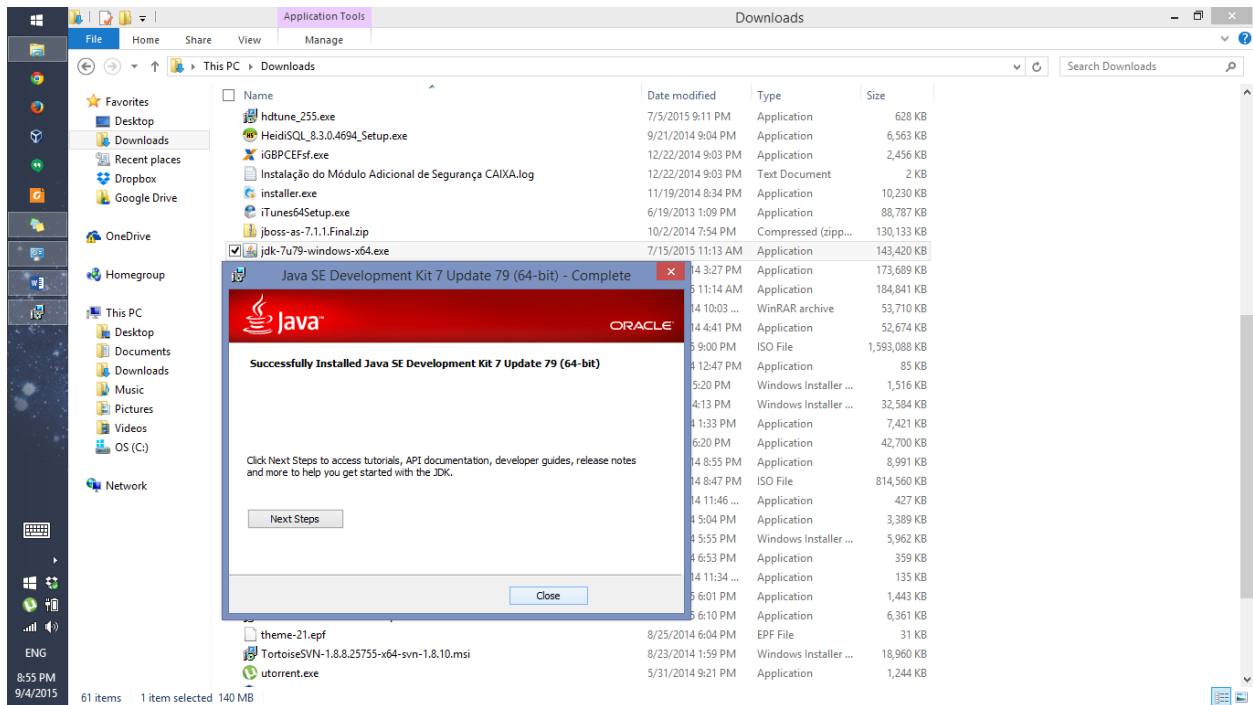
Looking for JDK on ARM?
JDK 7 for ARM downloads have moved to the JDK 7 for ARM download page.

Java SE Development Kit 7u79
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

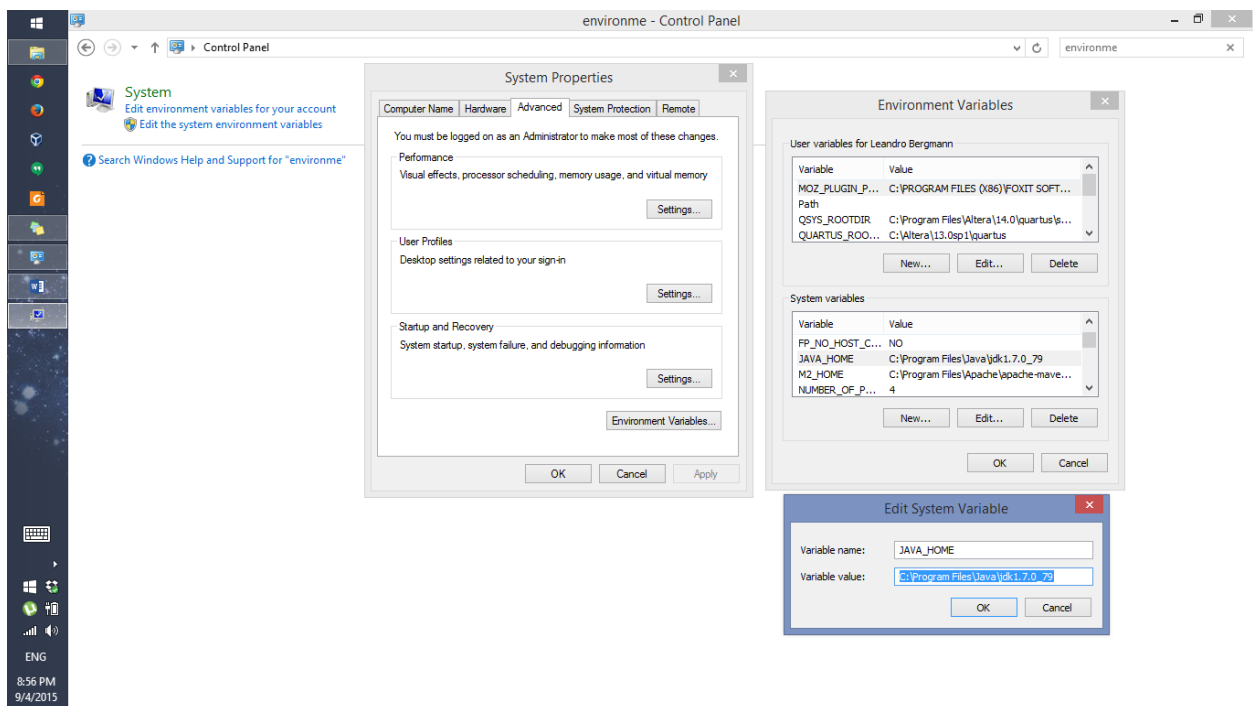
Product / File Description	File Size	Download
Linux x86	130.4 MB	jdk-7u79-linux-i586.rpm
Linux x86	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u79-linux-x64.rpm
Linux x64	146.4 MB	jdk-7u79-linux-x64.tar.gz
Mac OS X x64	196.89 MB	jdk-7u79-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.4 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows x86	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

Java SE Development Kit 7u79 Demos and Samples Downloads
You must accept the Oracle SDK License to download this software.

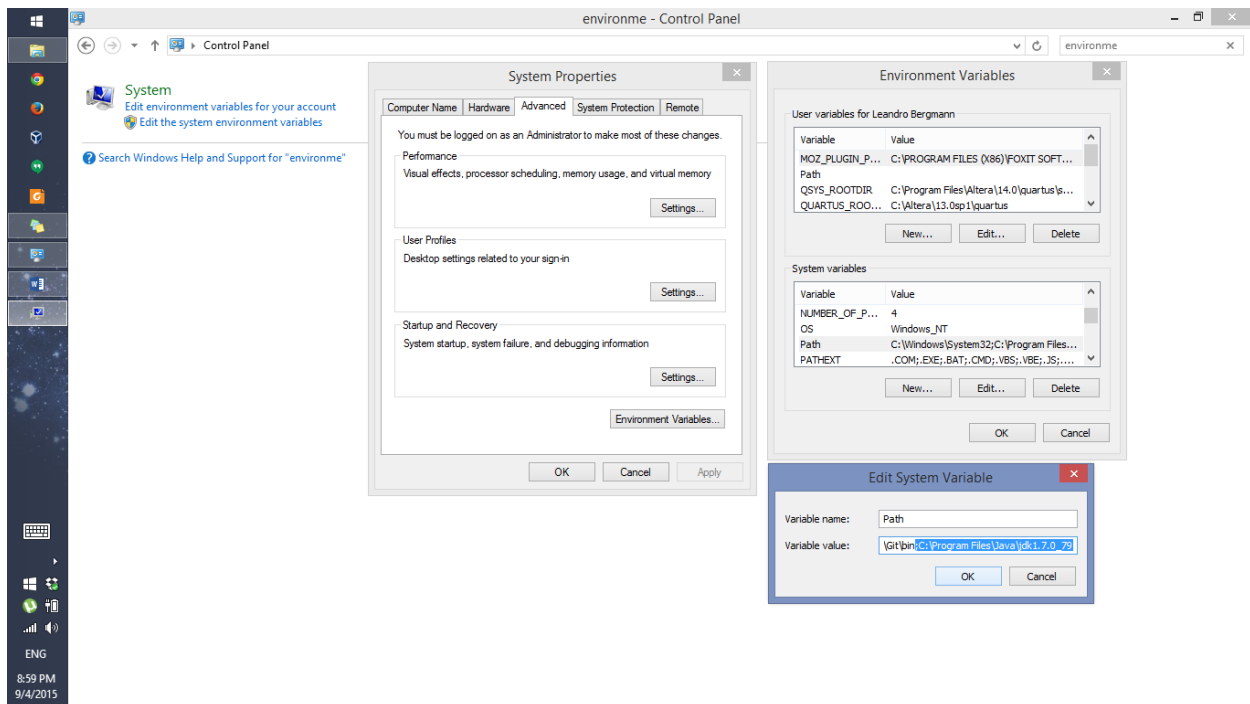
Instale normalmente, basta apertar next até o fim.



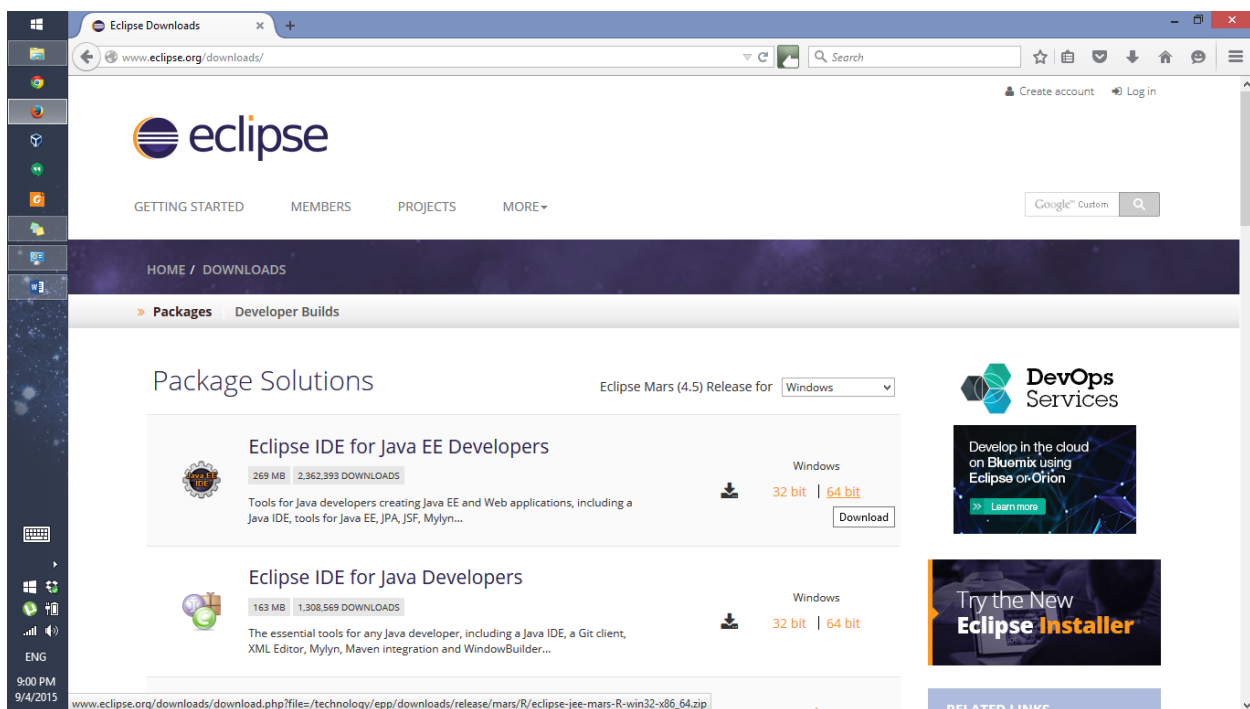
Vá no painel de controle e procure por “Variáveis de Ambiente”. Clique em “Editar variáveis de ambiente”. Clique em “Variáveis de Ambiente”. Clique em “Nova...” e adicione o JAVA_HOME conforme figura. Substituir o endereço com o endereço e que o JDK foi instalado. Lembre-se que não é a JRE, e sim a JDK (haverá uma pasta de cada). Clique em OK.



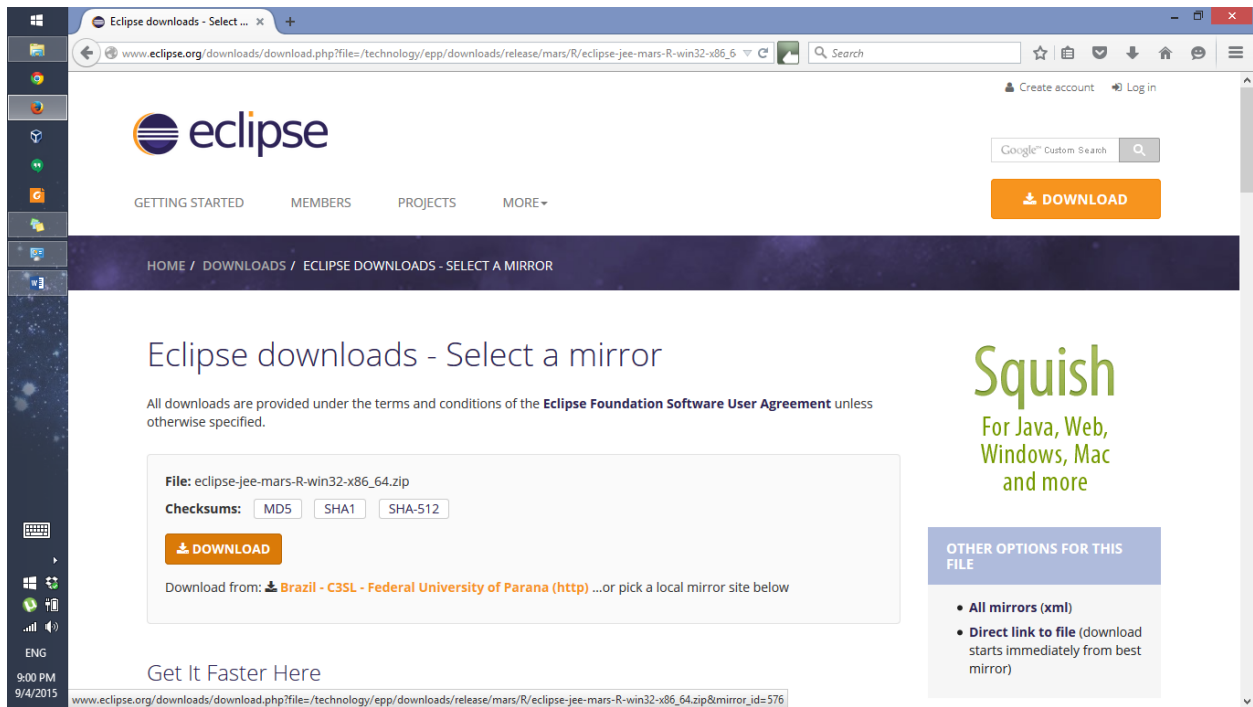
Nas variáveis já criadas, ache uma chamada “Path”. Clique em “Editar...”. Adicione o mesmo endereço adicionado anteriormente, separado do anterior por ponto e vírgula, conforme imagem. Clique em OK.



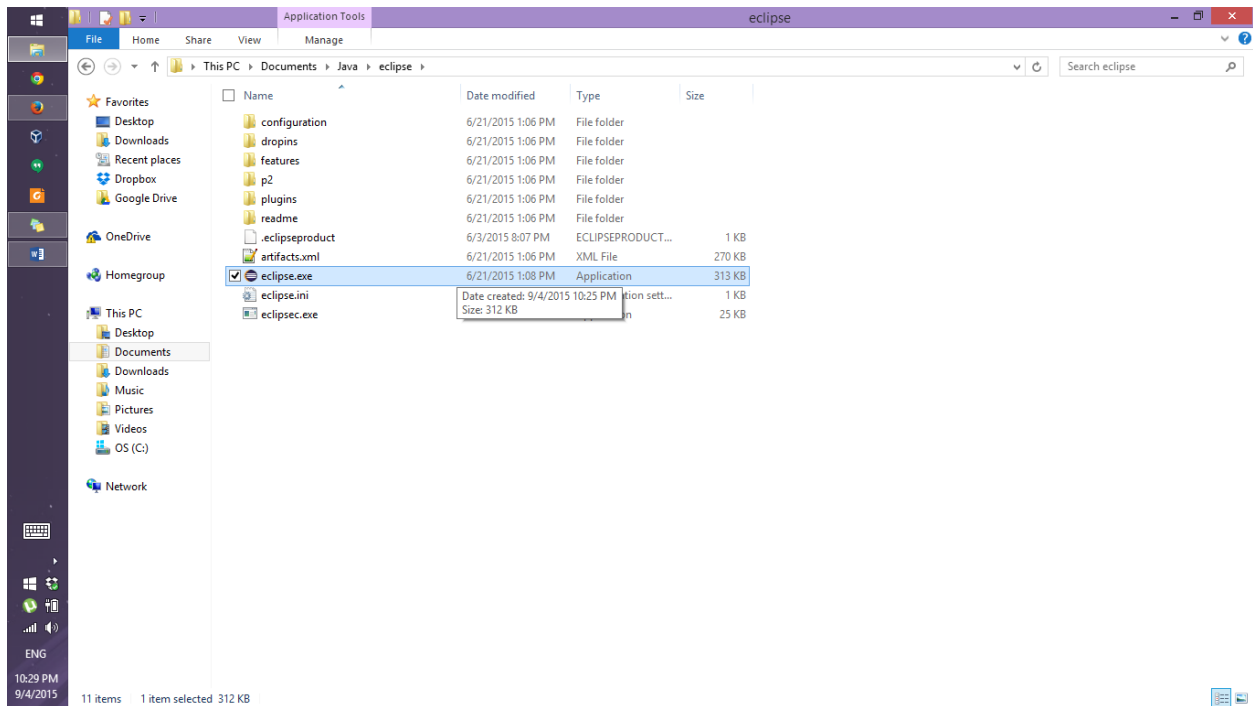
Baixe o Eclipse IDE for Java EE Developers, de acordo com o seu SO.



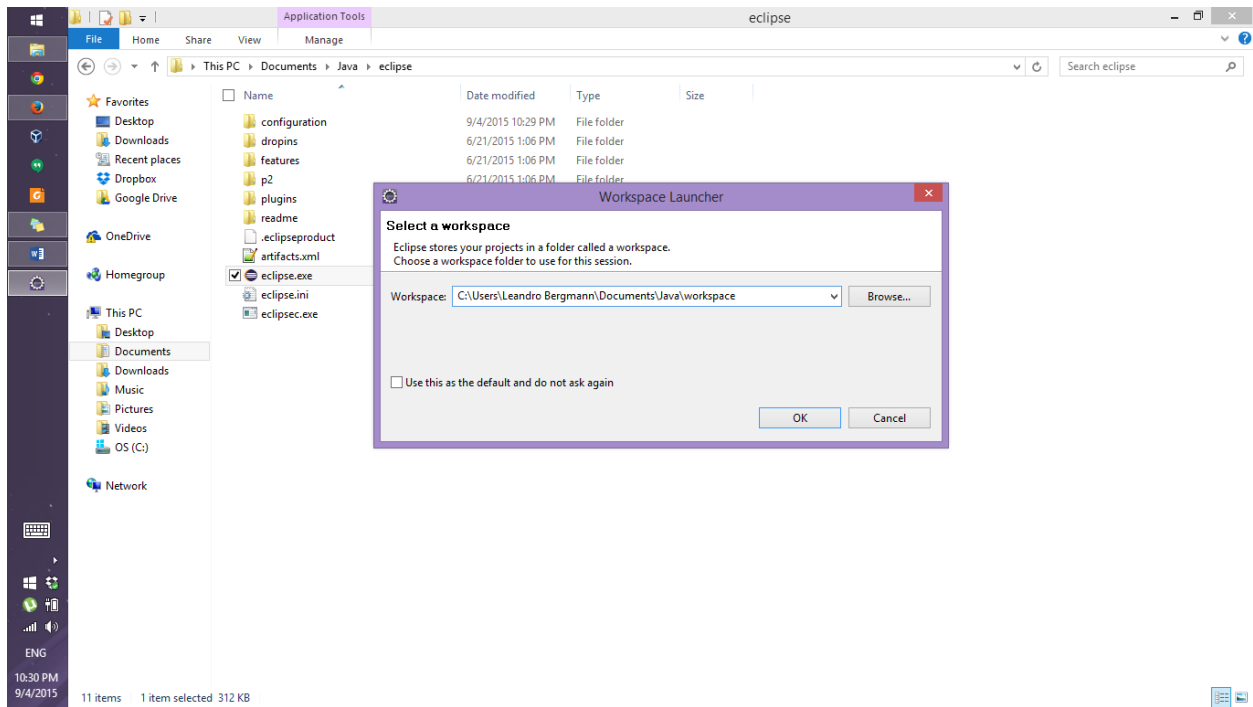
Clique em “Download”.



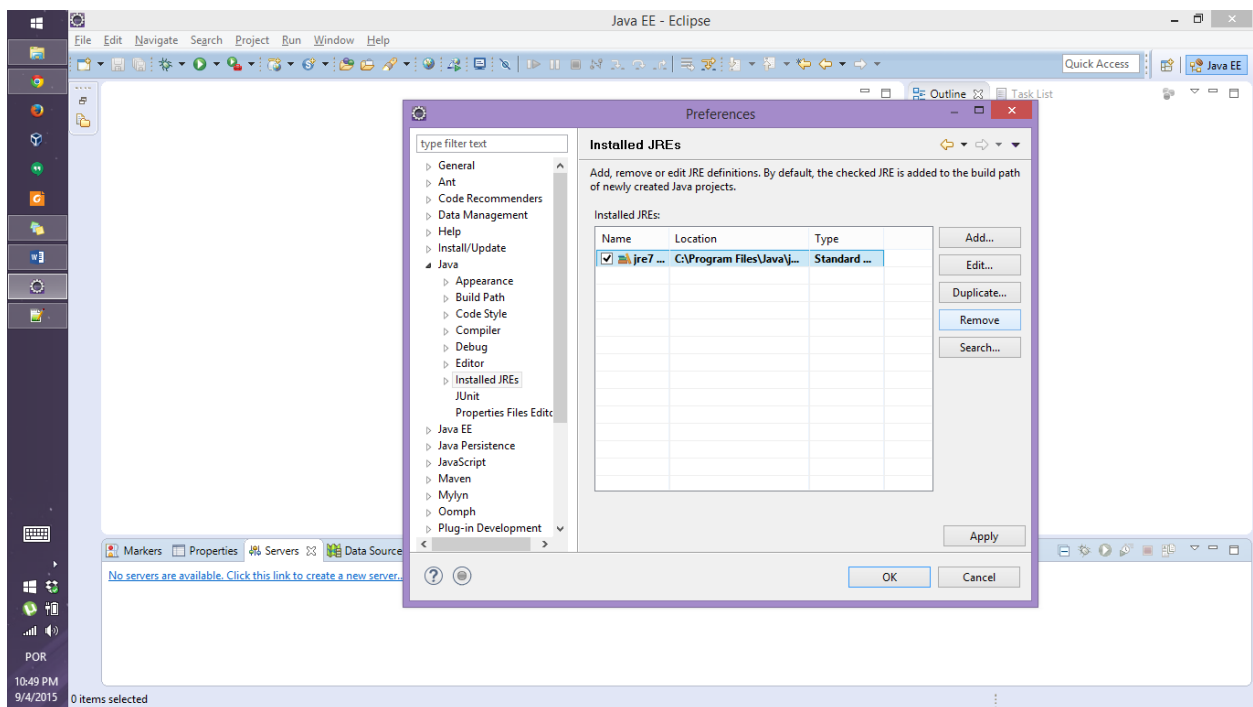
Descompacte o arquivo baixado, e clique em “eclipse.exe”.



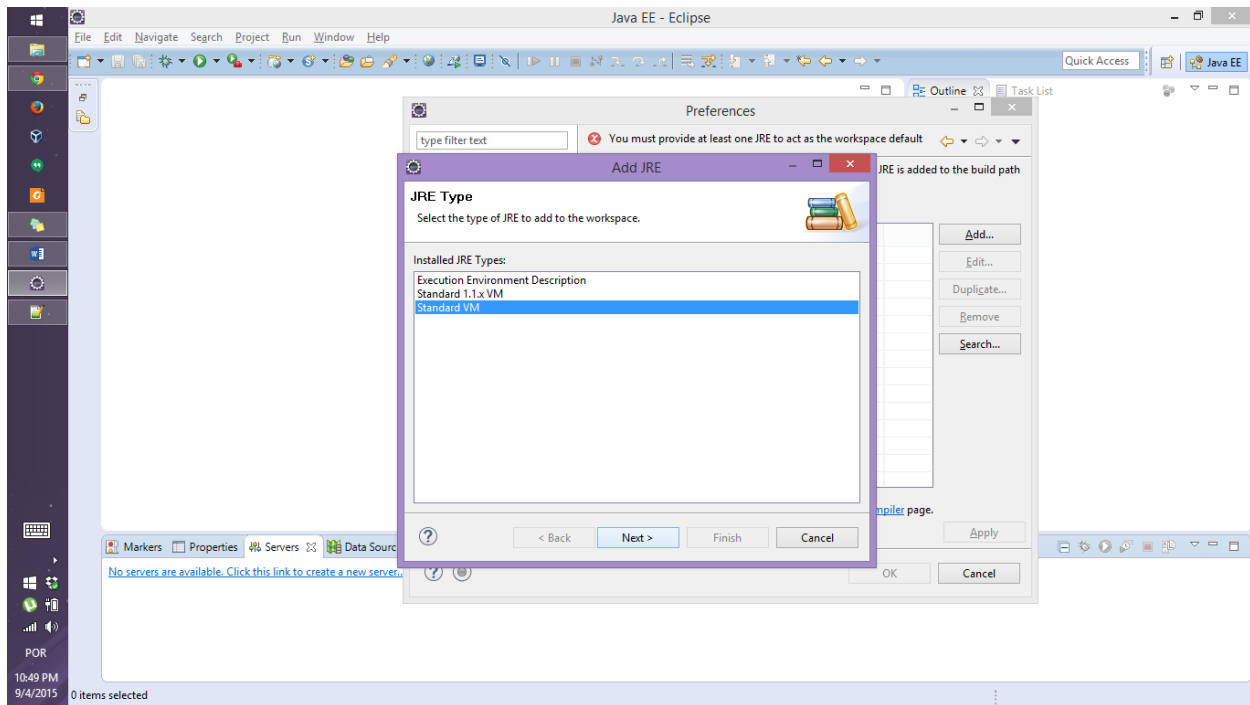
Escolha um local para o seu workspace. O workspace será uma pasta que o eclipse vai usar para colocar seus arquivos de configurações. Esta pasta é descartável e pode ser criada toda vez que o eclipse é iniciado, portanto pode escolher qualquer lugar. Nunca mexa no conteúdo da pasta workspace. Clique em OK.



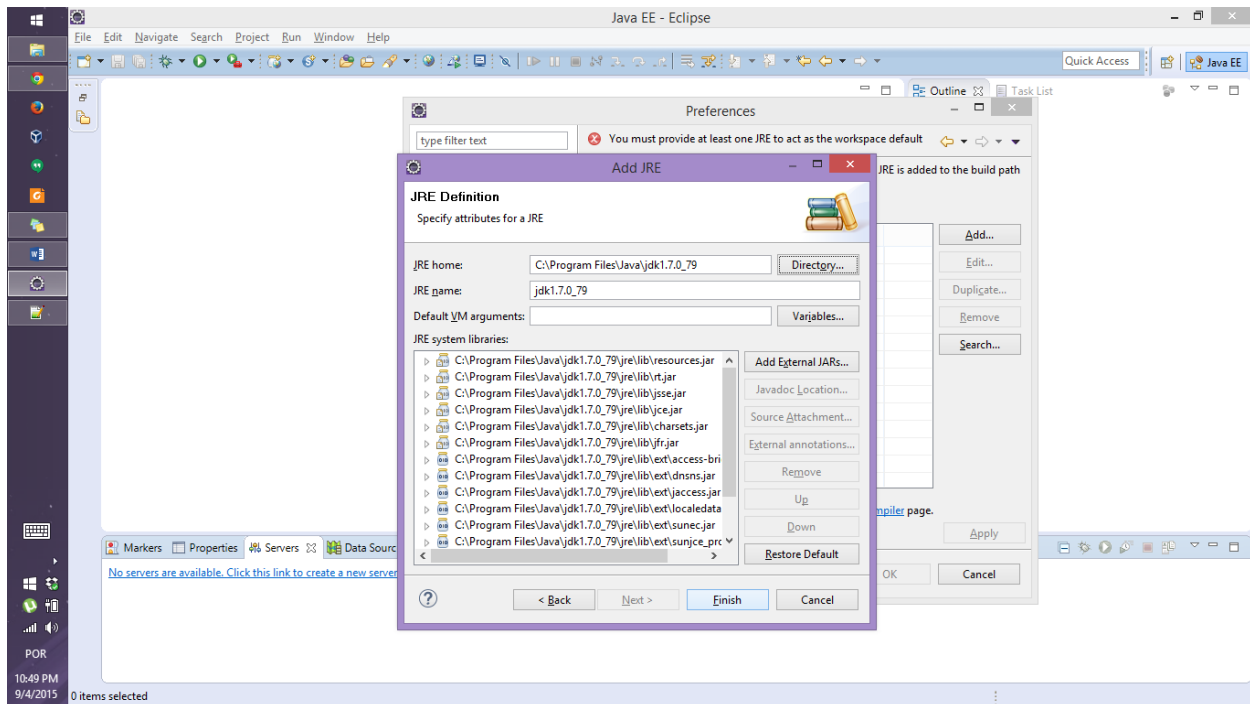
Vá em Windows → Preferences → Java → Installed JREs. Seleccione a jre7 e pressione “Remove”.



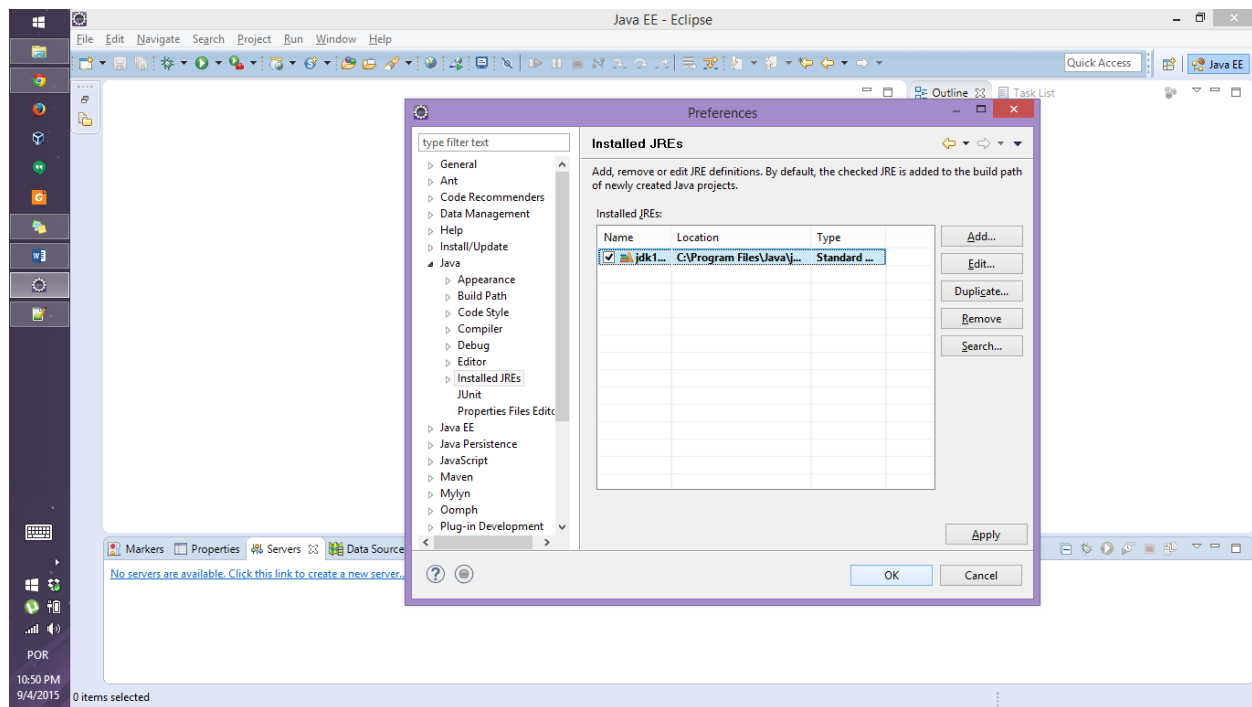
Clique em “Add...” e depois em ‘Standard VM’. Clique em Next.



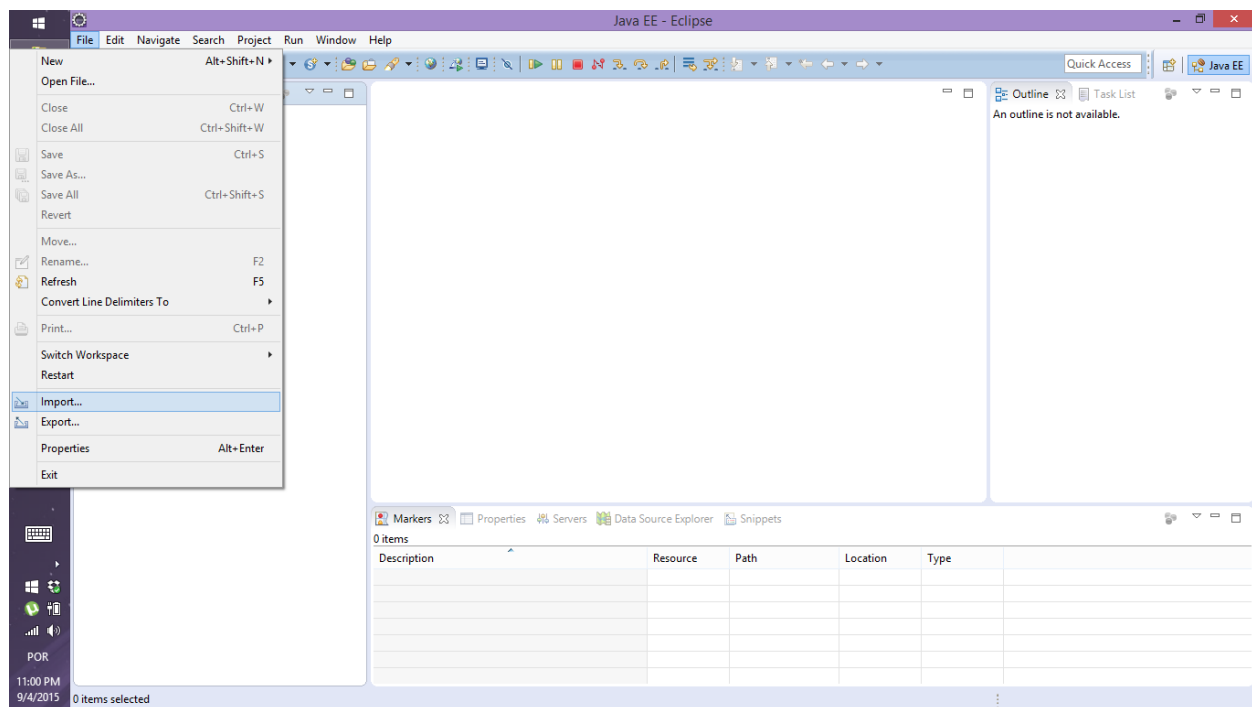
Escolha a pasta onde seu JDK foi instalado. Clique Finish.



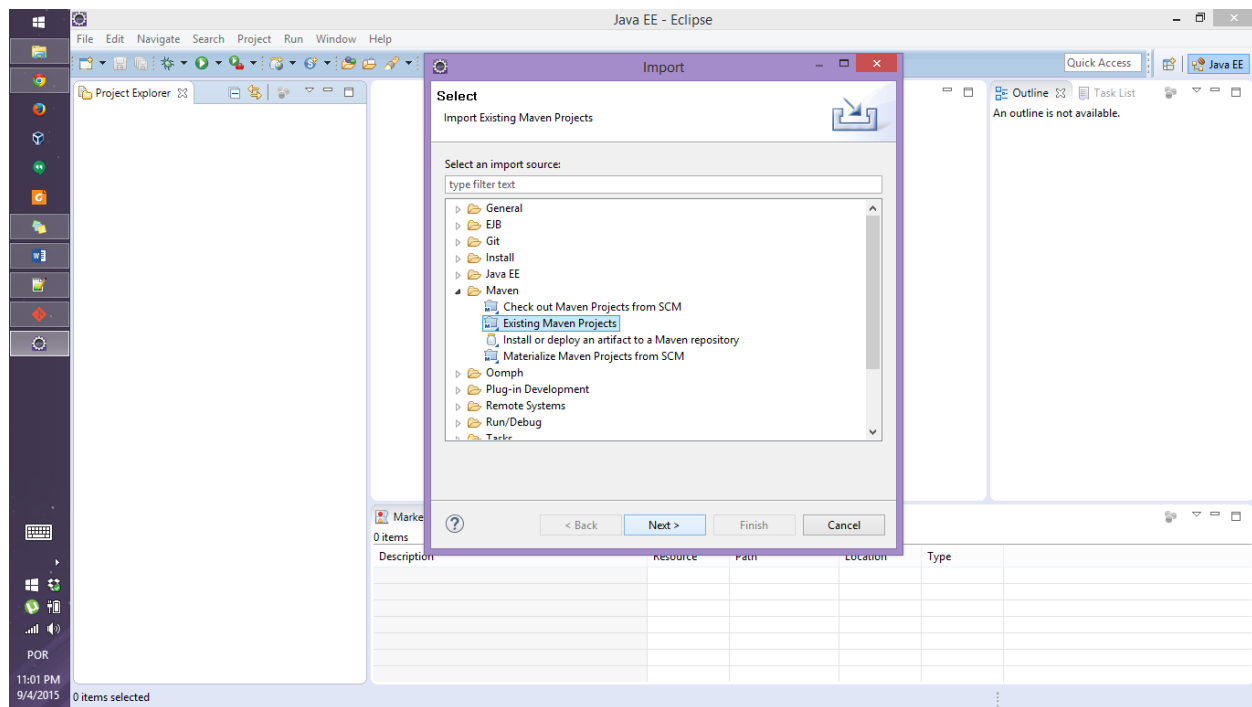
Marque a JDK que foi adicionada, fazendo dela a padrão. Clique em OK.



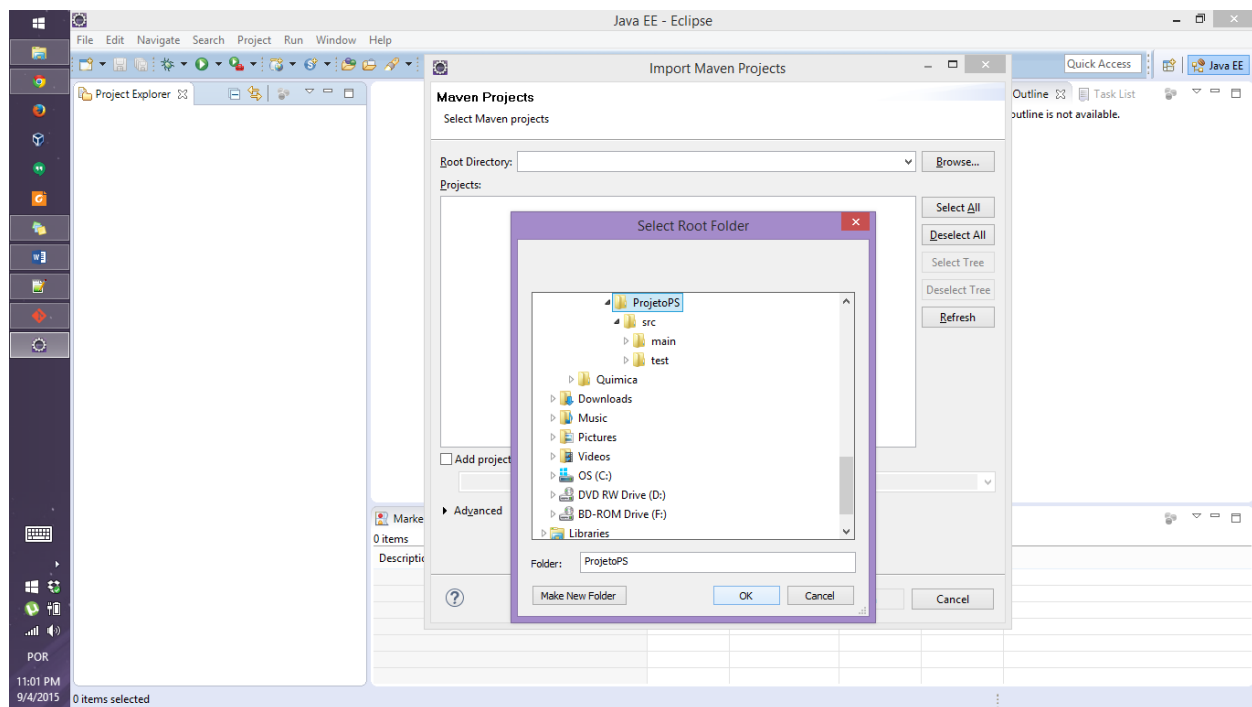
Agora vamos importar o projeto que está no repositório. Clique em File → Import.



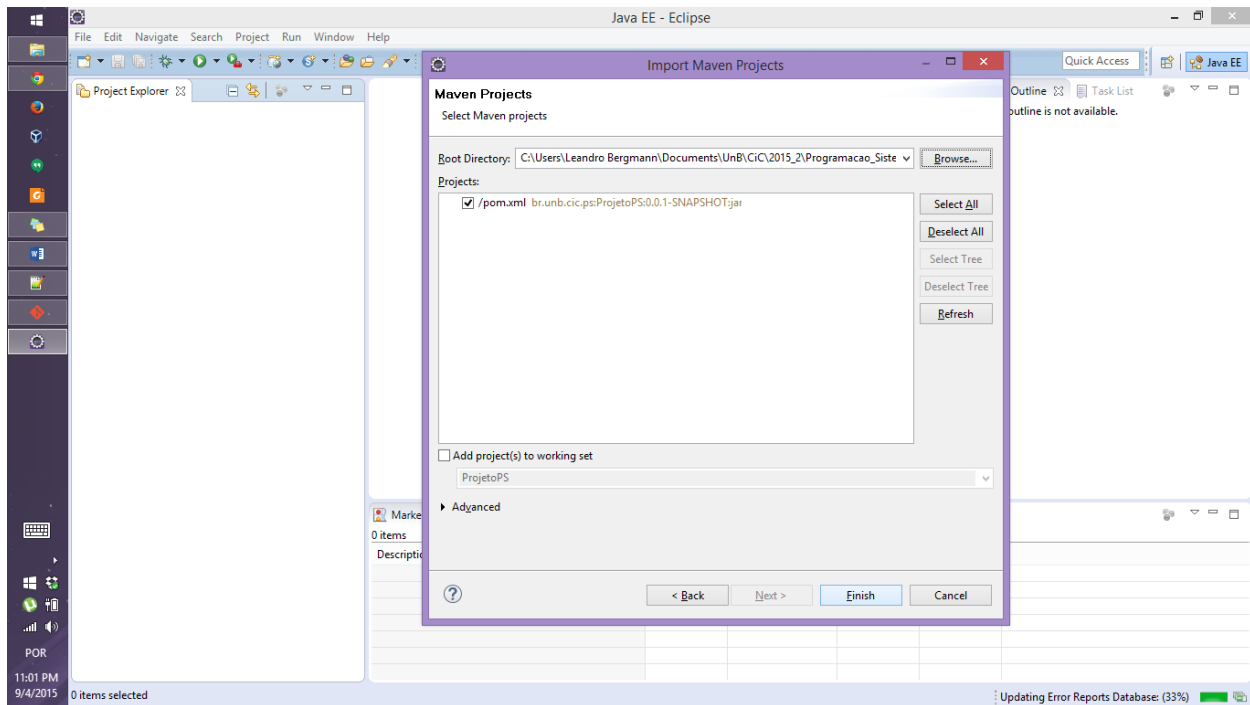
Clique em Maven → Existing Maven Projects.



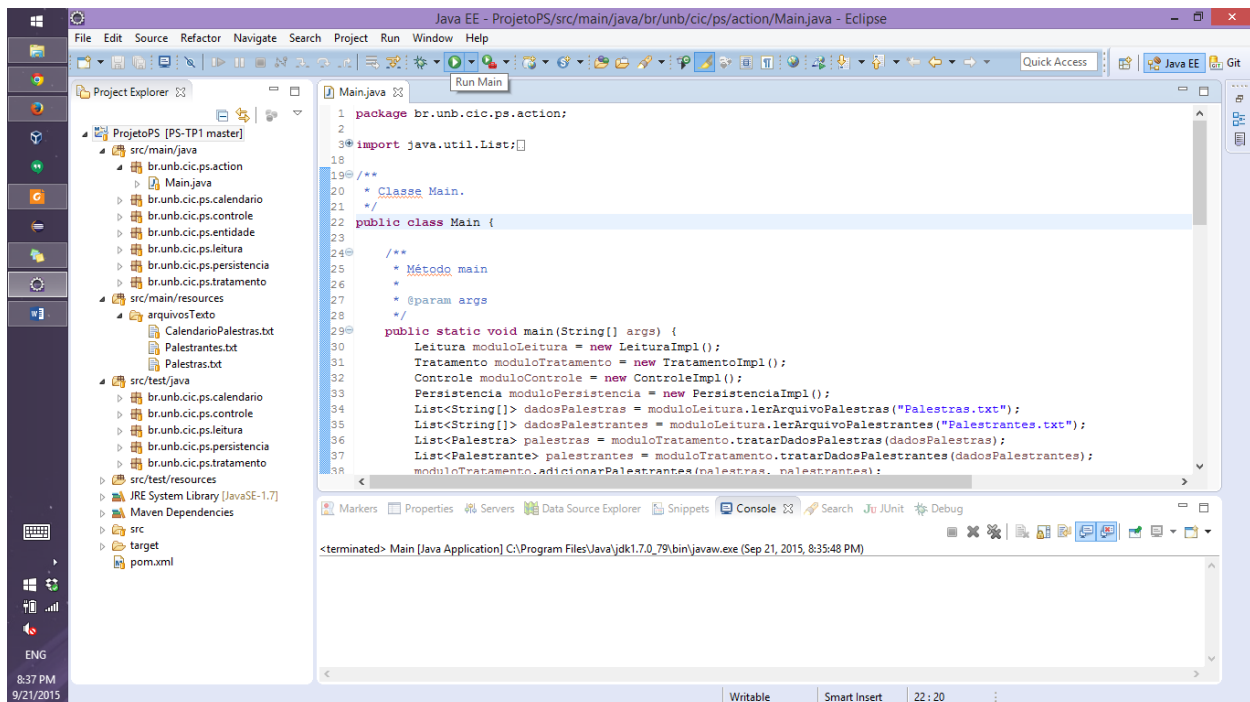
Clique em ProjetoPS, que é o projeto do nosso repositório. Clique em OK.



Clique em Finish.



O eclipse irá importar o projeto e baixar as dependências. Pode-se ver que o JUnit já foi baixado automaticamente, devido à integração com o Maven. O eclipse também já é integrado ao Git, ou seja, podemos dar pull e push sem precisar da linha de comando. O projeto criado para teste é uma classe que faz trata números primos. Para rodar o programa, basta ir no botão de “Play” da figura a seguir.



The screenshot displays the Eclipse IDE with the 'Run As' context menu open for a project named 'ProjetoPS'. The menu options include 'New', 'Go Into', 'Show In', 'Copy', 'Copy Qualified Name', 'Paste', 'Delete', 'Remove from Context', 'Build Path', 'Refactor', 'Import...', 'Export...', 'Refresh', 'Close Project', 'Close Unrelated Projects', 'Validate', 'Show in Remote Systems view', 'Run As', 'Debug As', 'Profile As', 'Replace With', 'Restore from Local History...', 'Maven', 'Team', 'Compare With', 'Configure', 'Source', and 'Properties'. The 'Run As' option is selected, opening a sub-menu with the following items: '1 Java Applet', '2 Java Application', '3 JUnit Test', '4 Maven build', '5 Maven build...', '6 Maven clean', '7 Maven generate-sources', '8 Maven install', '9 Maven test', and 'Run Configurations...'. The background shows the Project Explorer on the left with the 'src/main' directory expanded, and the Java code editor on the right displaying a Java class with a main method.

Segue o resultado esperado.

