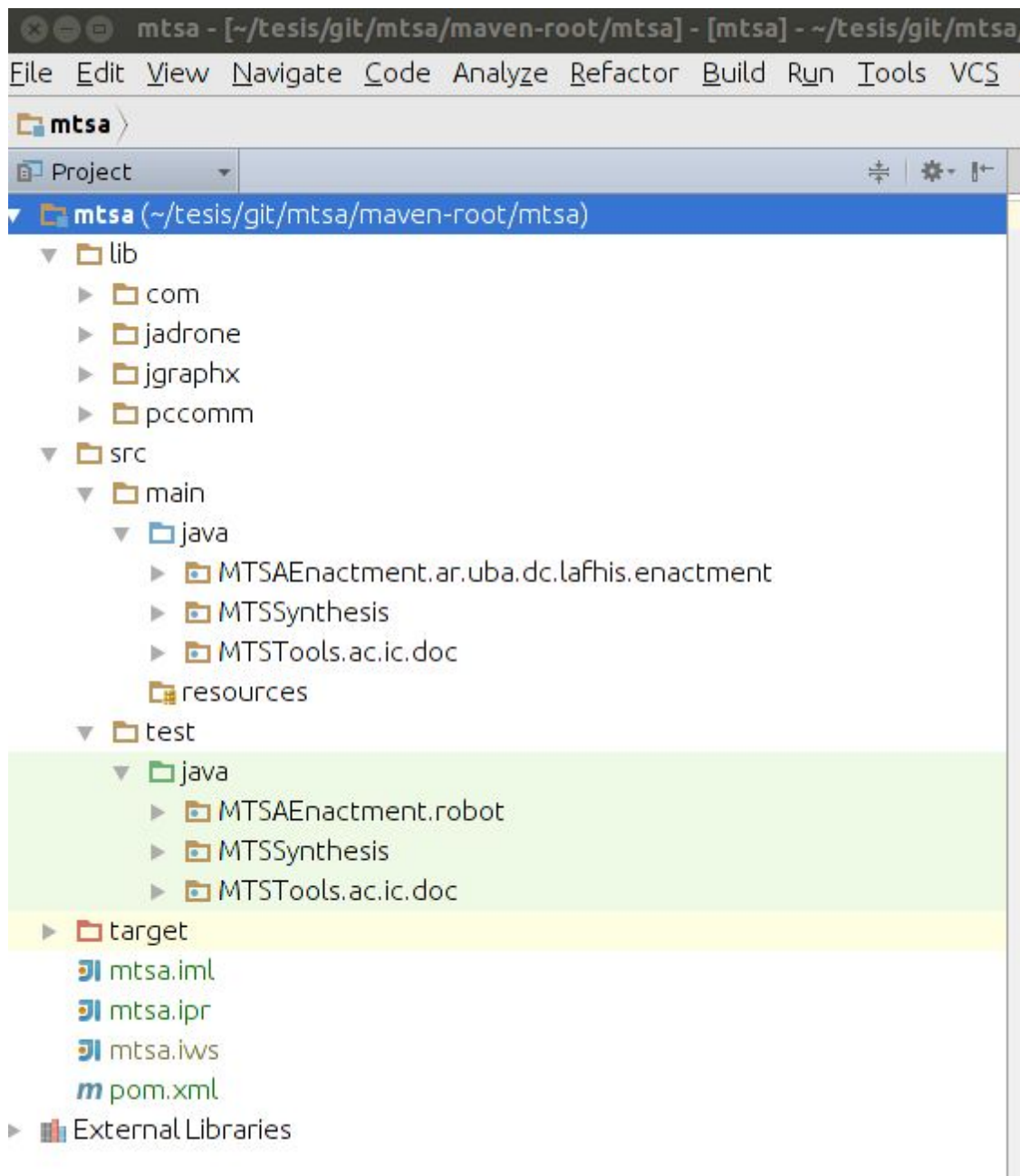


Migración a Maven:

La migración a Maven requiere que el proyecto de MTSA se adecue a la estructura esperada por Maven. De lo contrario la configuración se hace tan engorrosa que es casi imposible utilizarlo.

Maven no es bueno para manejar proyectos múltiples con dependencias entre sí, por lo cual la migración implica generar un solo proyecto y transformar en módulos del mismo a los viejos proyectos (Itsa, MTSTools, MTSAEnactment, etc)

La estructura de directorios del nuevo proyecto es como sigue: (esto es parcial porque todavía faltan migrar módulos, pero se va a respetar la misma estructura mostrada aquí)



maven-root: es el root del repositorio.

mtsa: es el root del proyecto. Contiene src, lib, target y el pom.xml

Nota: este pom.xml es el único que existe y en él se configura todo lo referido a maven.

lib: contiene las librerías de terceras partes. Se instalan copiando el jar en “maven-root/locallib”, ejecutando el comando siguiente y modificando el pom.xml:

Comando para jgraphx.jar:

```
mvn install:install-file -DlocalRepositoryPath=lib -DcreateChecksum=true -Dpackaging=jar -Dfile=../locallib/jgraphx.jar -DgroupId=jgraphx -DartifactId=jgraphx -Dversion=0.1
```

Se agrega la dependencia en el pom.xml:



```
<!-- Local dependencies -->
<dependency>
  <groupId>jadrone</groupId>
  <artifactId>jadrone</artifactId>
  <version>0.32</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/jadrone/jadrone/0.32/jadrone-0.32.jar</systemPath>
</dependency>
<dependency>
  <groupId>pccomm</groupId>
  <artifactId>pccomm</artifactId>
  <version>0.1</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/pccomm/pccomm/0.1/pccomm-0.1.jar</systemPath>
</dependency>
<dependency>
  <groupId>jgraphx</groupId>
  <artifactId>jgraphx</artifactId>
  <version>0.1</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/jgraphx/jgraphx/0.1/jgraphx-0.1.jar</systemPath>
</dependency>
<dependency>
  <groupId>com.microsoft</groupId>
  <artifactId>z3</artifactId>
  <version>1.0</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/com/microsoft/z3/1.0/z3-1.0.jar</systemPath>
</dependency>
</dependencies>
</project>
```

target: [Este directorio no se comitea]

Contiene todo el output de la compilación, ejecución de tests, generación de jars, reporte de tests, etc.

los comandos de maven deben ejecutarse en el root del proyecto, o sea, en “maven-root/mtsa”:

- Compilación de código: General los .class del código y queda en una estructura de directorios idéntica a src dentro de target.
comando: “mvn compile”

- Ejecución de tests y generación de “mtsa-1.0-SNAPSHOT.jar”
comando: “mvn package”
- Generación del reporte de tests: (“mtsa/target/newsite/surefire-report.html”)
comando: “mvn surefire-report:report”



Last Published: 2016-02-19 | Version: 1.0-SNAPSHOT

[Built by Maven](#)

Surefire Report

Summary

[\[Summary\]](#) [\[Package List\]](#) [\[Test Cases\]](#)

Tests	Errors	Failures	Skipped	Success Rate	Time
77	0	0	2	97.403%	0.283

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Package List

[\[Summary\]](#) [\[Package List\]](#) [\[Test Cases\]](#)

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
MTSSynthesis.ar.dc.uba.model.structure	1	0	0	0	100%	0
MTSSynthesis.ar.dc.uba.model.language	15	0	0	0	100%	0.004
MTSSynthesis.ar.dc.uba.parser	12	0	0	0	100%	0.013
MTSTools.ac.ic.doc.common.collections	15	0	0	0	100%	0.034
MTSSynthesis.ar.dc.uba.lsc	12	0	0	2	83.333%	0.059
MTSSynthesis.controller.game.bgr	5	0	0	0	100%	0.046
MTSAEnactment.robot	4	0	0	0	100%	0.031
MTSTools.ac.ic.doc.common.relations	4	0	0	0	100%	0.016
MTSSynthesis.ar.dc.uba.model.condition	3	0	0	0	100%	0.003
MTSTools.ac.ic.doc.mtstools.model.operations.impl	1	0	0	0	100%	0.014
MTSSynthesis.ar.dc.uba.util	5	0	0	0	100%	0.063

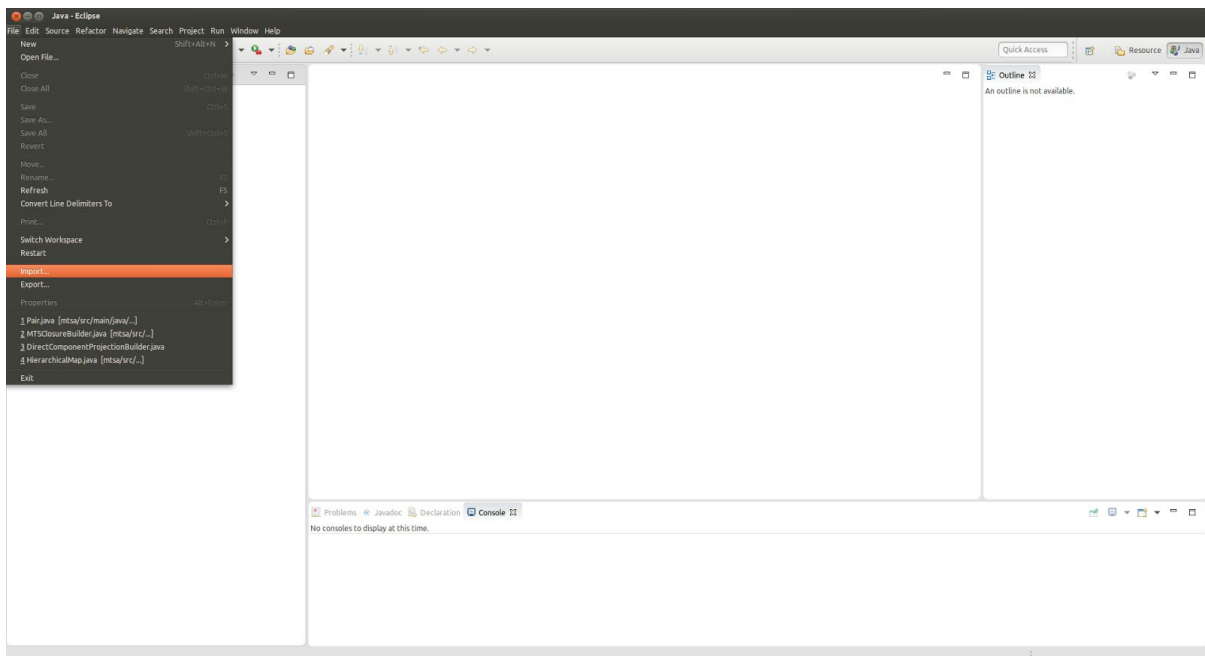
Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

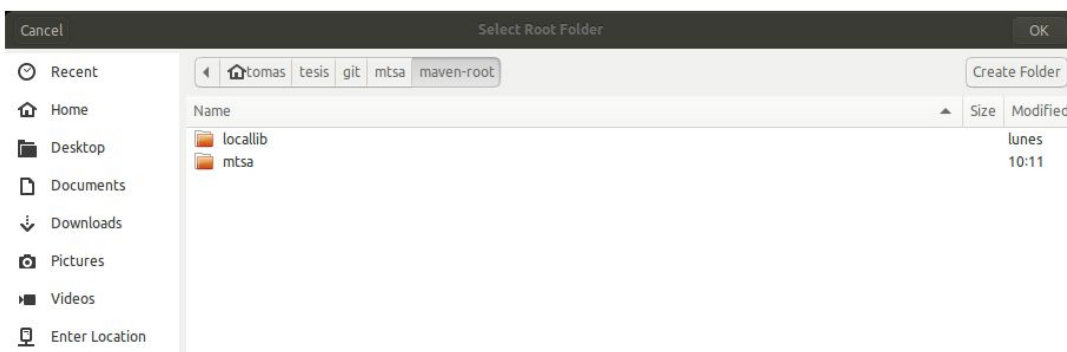
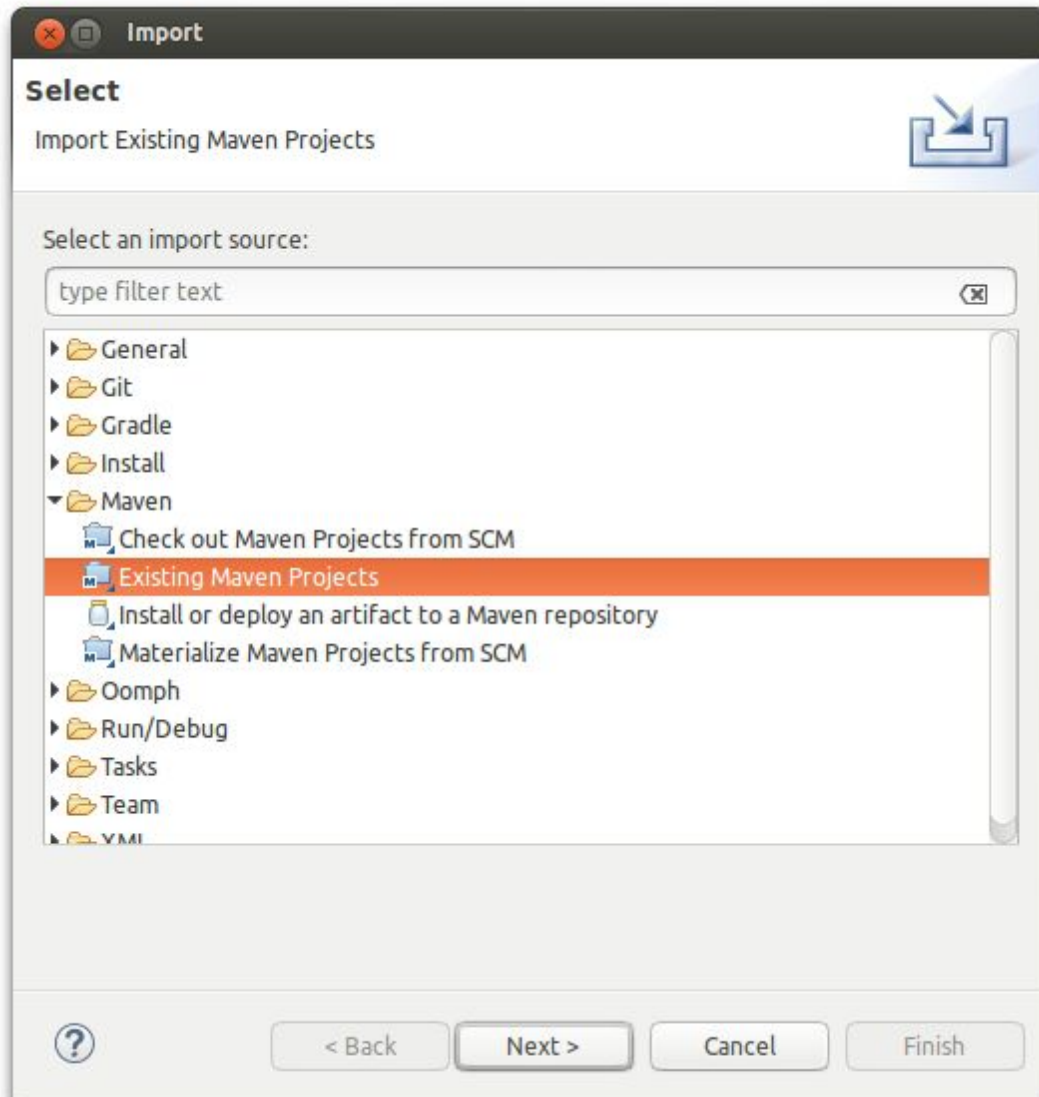
Trabajando con Eclipse

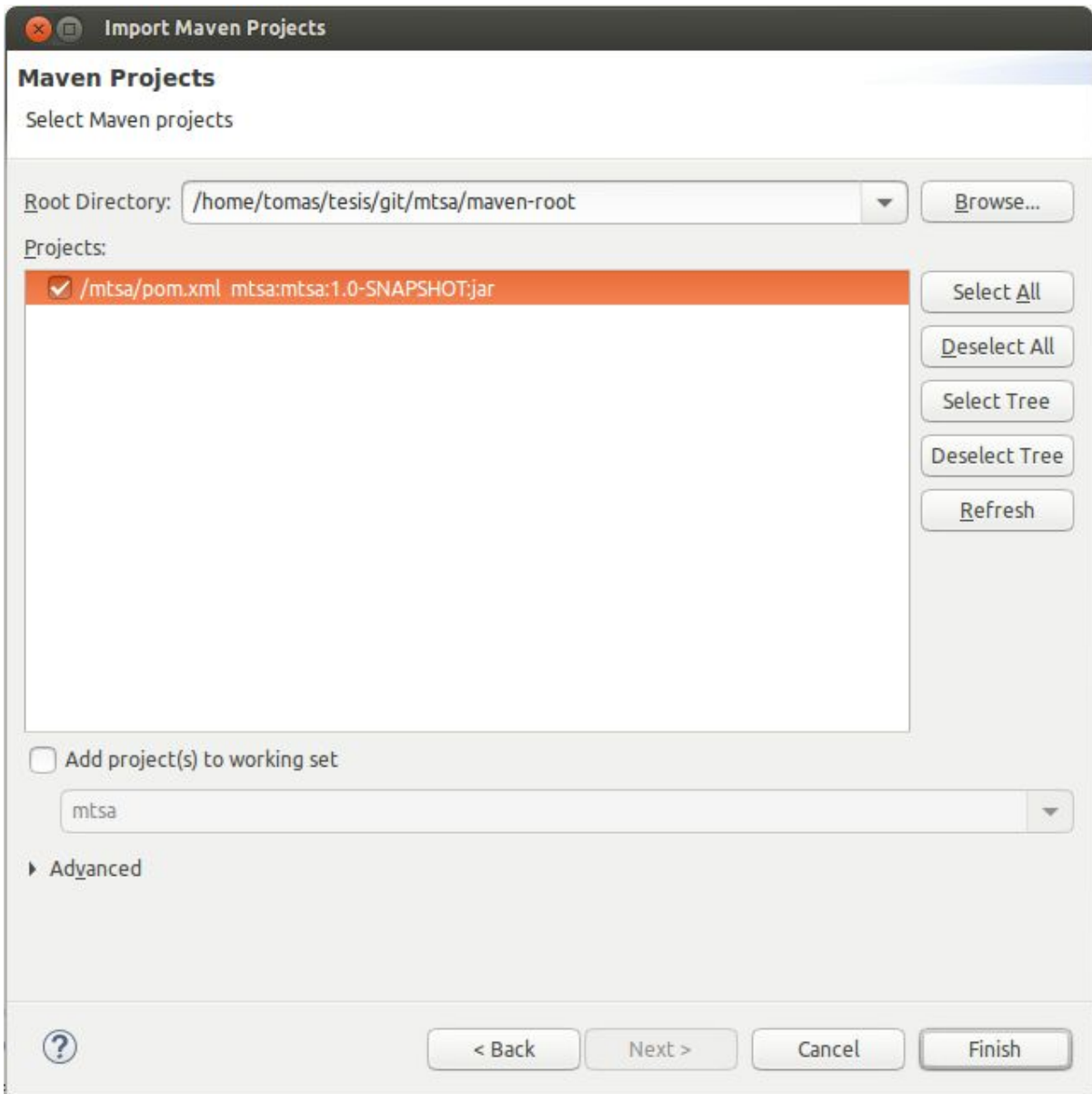
Versión:

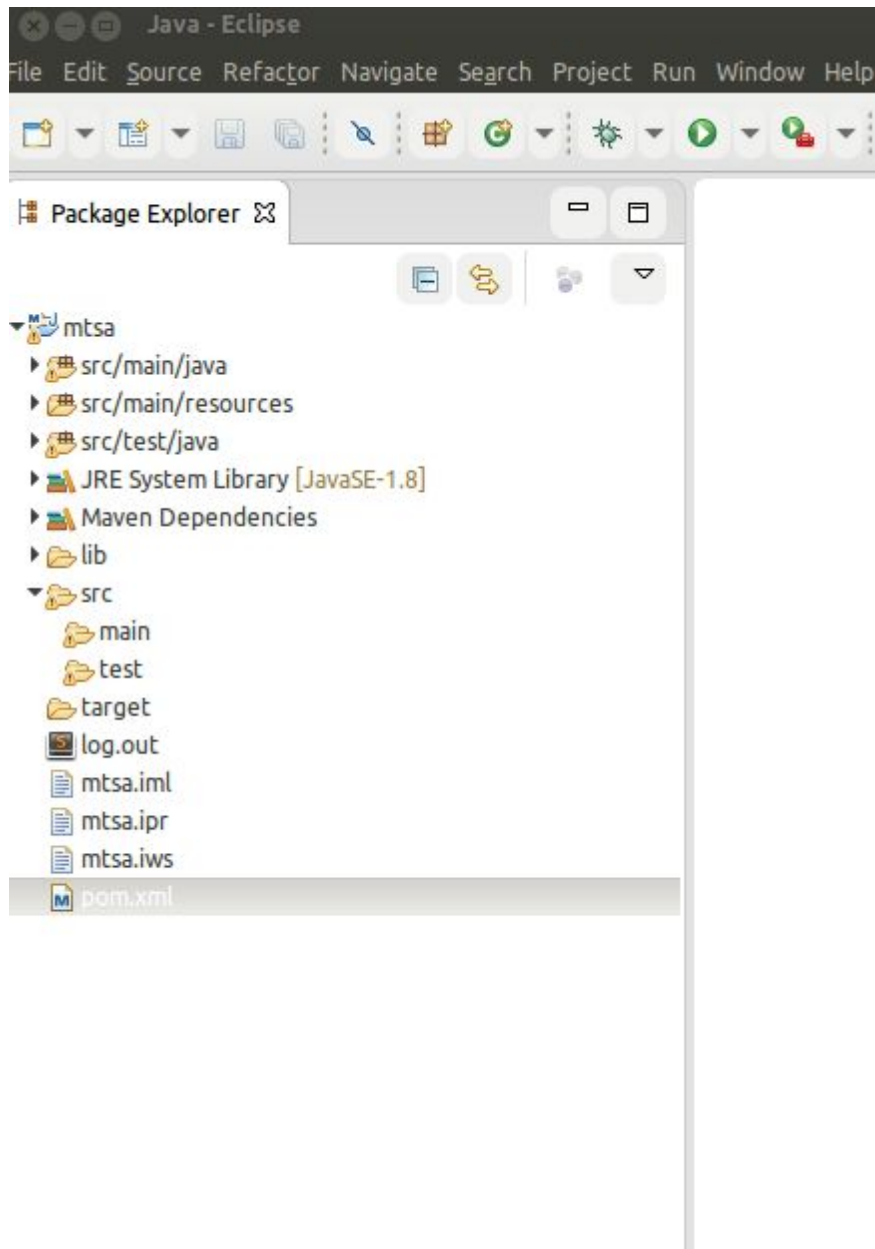


Cómo importar el proyecto desde eclipse:

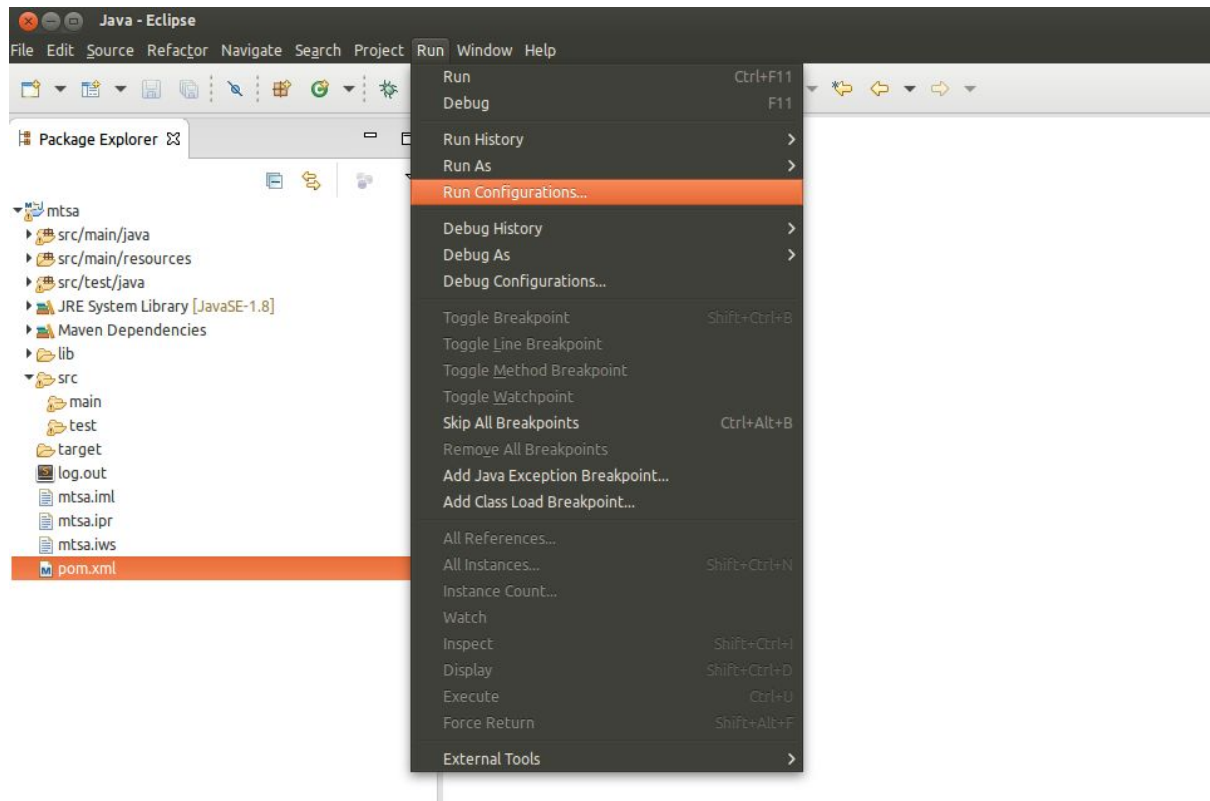








Cómo ejecutar tareas de maven desde Eclipse.



Se deben configurar las tareas a mano por unica vez, especificando el Base directory y asegurando de que eclipse utilice java8 para la ejecucion:

Run Configurations

Create, manage, and run configurations

Name: **compile**

Main JRE Refresh Source Environment Common

Base directory: /home/tomas/tesis/git/mtsa/maven-root/mtsa

Goals: compile

Profiles:

User settings: File...

Offline Update Snapshots

Debug Output Skip Tests Non-recursive

Resolve Workspace artifacts

1 Threads

Parameter Name	Value
----------------	-------

Filter matched 9 of 9 items

Revert Apply

Close Run

Run Configurations

Create, manage, and run configurations

Name: compile

Main JRE Refresh Source Environment Common

Runtime JRE:

- Workspace default JRE (jdk1.8.0_45)
- Execution environment: [] Environments...
- Alternate JRE: jdk1.8.0_45 Installed JREs...

VM arguments:

[] Variables...

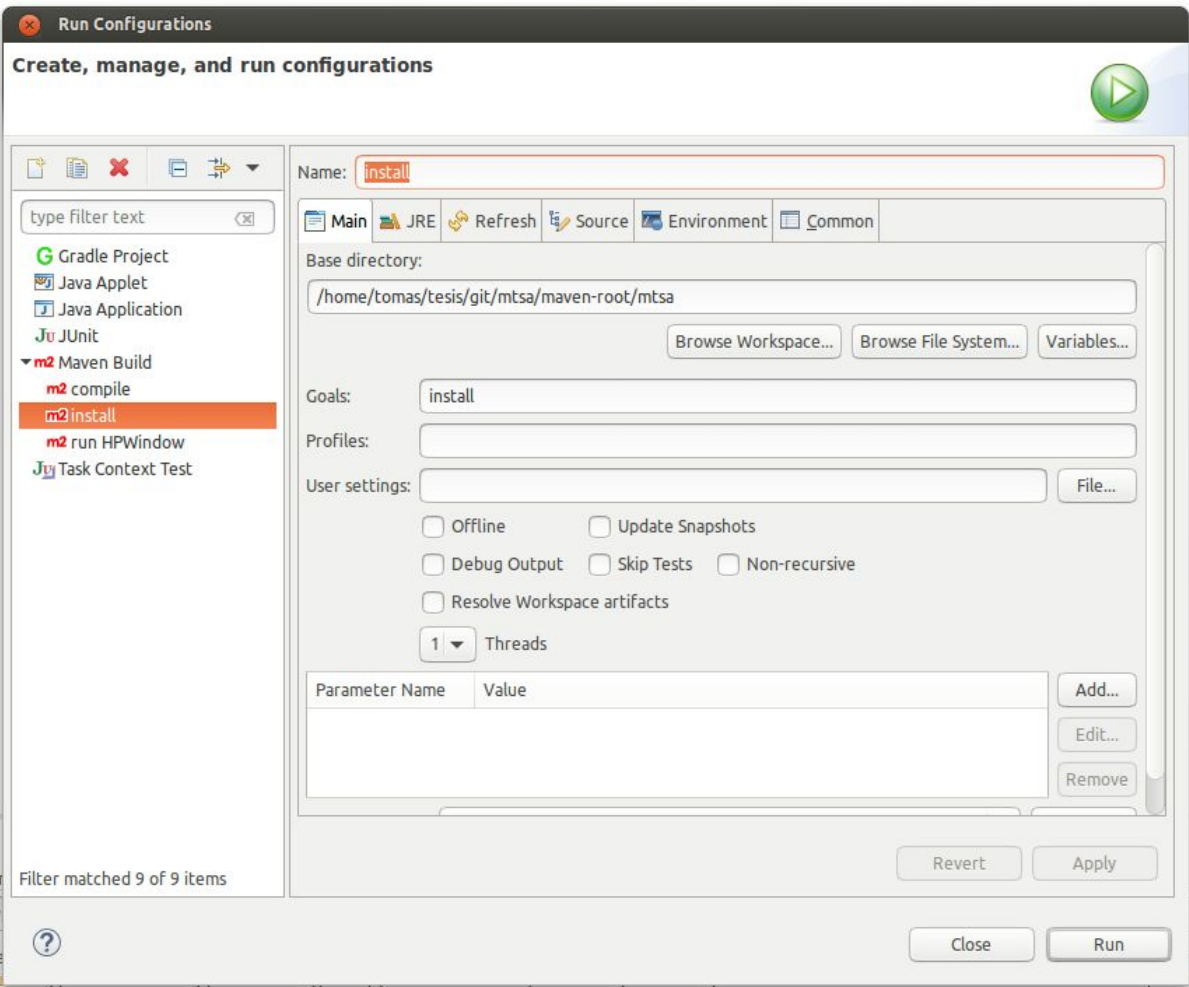
Java executable:

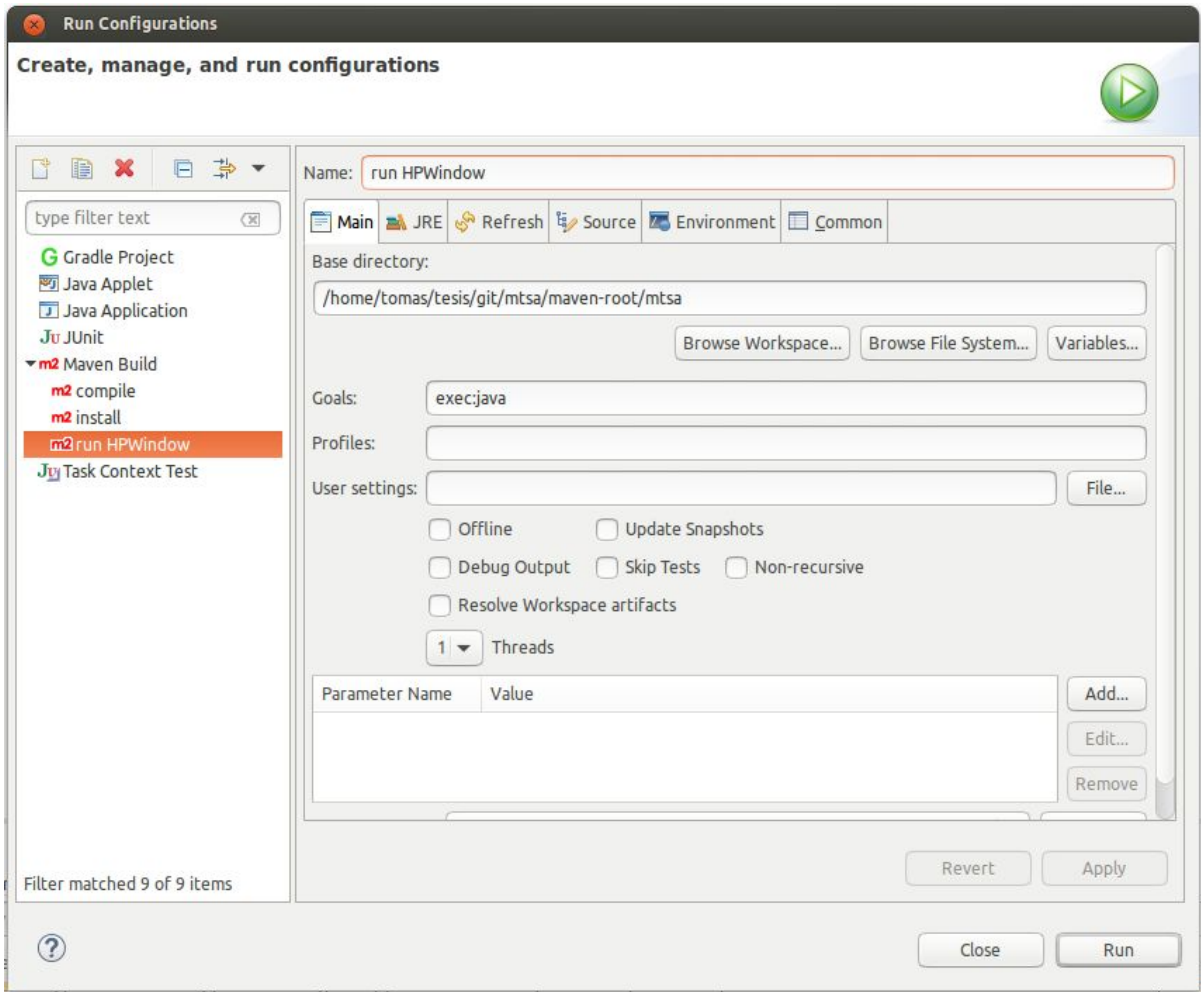
- Default (javaw)
- Alternate [javaw]

Filter matched 9 of 9 items

Revert Apply

Close Run





Run Configurations

Create, manage, and run configurations

Name:

Base directory:

Goals:

Profiles:

User settings: File...

Offline Update Snapshots

Debug Output Skip Tests Non-recursive

Resolve Workspace artifacts

1 Threads

Parameter Name	Value

Revert Apply

Close Run

Filter matched 10 of 10 items

- Gradle Project
- Java Applet
- Java Application
- JUnit
- ▼ m2 Maven Build
 - m2 compile
 - m2 install
 - m2 run HPWindow
 - m2 test report**
 - JUnit Task Context Test