



DevTools



Instalación automática de un puesto de desarrollo

Agenda

- ▶ Objetivos «DevTools»
- ▶ Etapas en la instalación con DevTools
- ▶ Demo live
- ▶ Administrar «DevTools»



Objetivos «DevTools» 1 / 2

- ▶ Instalar un entorno de desarrollo en la estación de trabajo local de manera :
 - ▶ Automática
 - ▶ Pre-configurada
 - ▶ Idea : Definir la lista de las herramientas necesarias para el desarrollo del proyecto
 - ▶ Eclipse Galileo
 - ▶ JDK 1.6
 - ▶ Apache Tomcat 6
 - ▶ Oracle Sql Developer
 - ▶ Plug-ins Eclipse
 - ▶ Subclipse
 - ▶ M2Eclipse
 - ▶ CheckStyle
 - ▶ PMD
 - ▶ ...
 - ▶ ...
-



Objetivos «DevTools» 2/2

- ▶ **Idea : pre-configurar las herramientas definidas anteriormente.**
 - ▶ Parámetros proxy dentro Eclipse Galileo
 - ▶ Carpetas de base para TOMCAT
 - ▶ URLs conexiones bases de datos dentro SQL Developer
 - ▶ Parámetros de los plug-ins
 - ▶ URL's SVN para subclipse
 - ▶ Fichero «input» para checkstyle
 - ▶ ...



Etapas en la instalación con DevTools

► Etapa I : el bootstrap



- Descarga del bootstrap desde un servidor web remoto
- Distribución compuesta de:
 - Un JDK
 - Una distribución de MAVEN
 - Un cliente SVN
- Se descomprime
 - Dentro una carpeta nombrada e:/bootstrap-myproject
 - Con un log e:/bootstrap-myproject.log



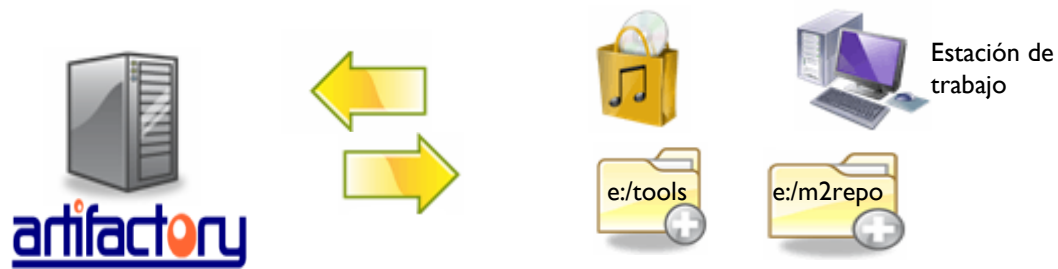
Etapas en la instalación con DevTools

► Etapa 2 : Ejecución de «DevTools»

► Fase de recuperación de la información para la instalación



► Fase de descarga e instalación

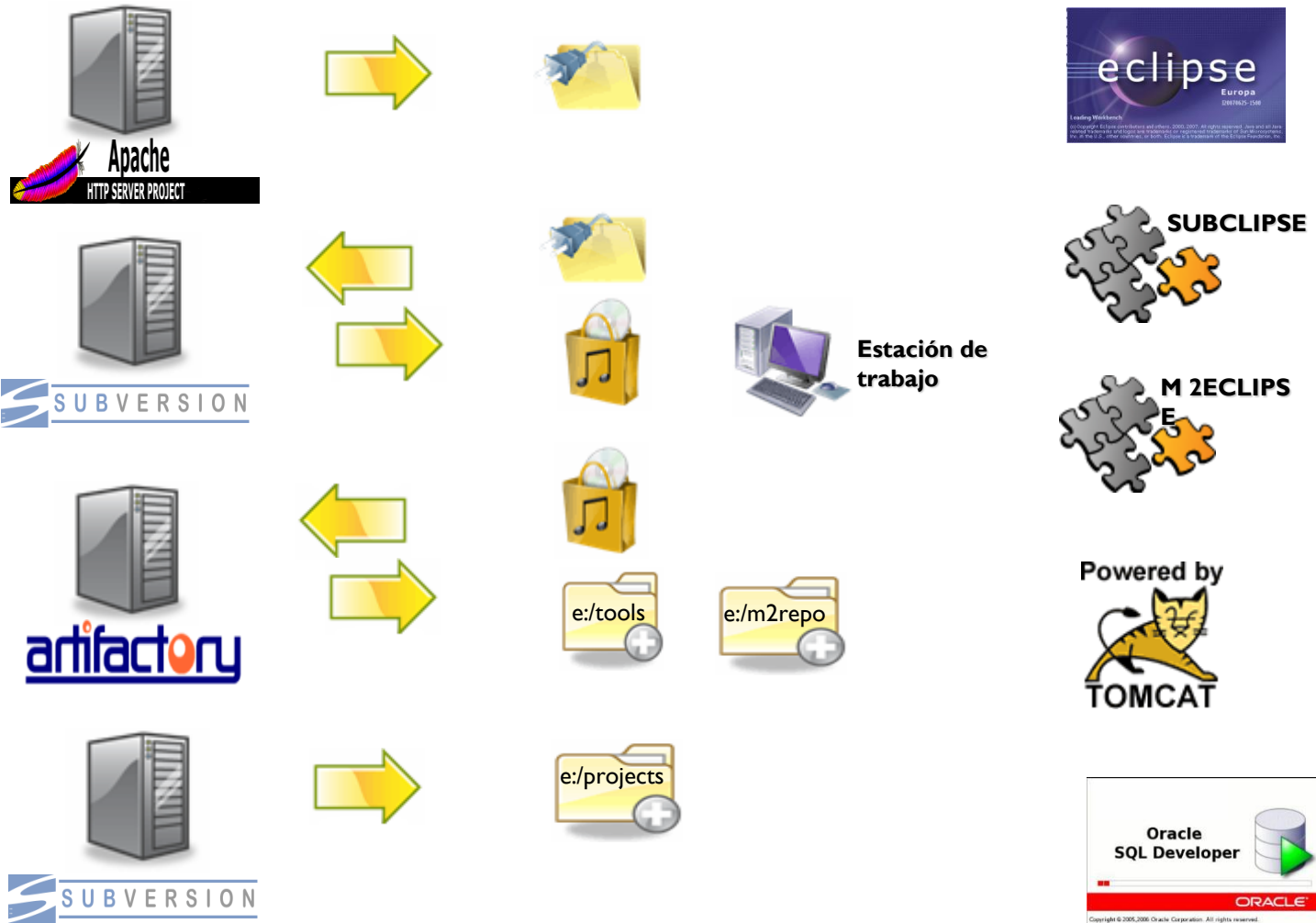


Etapas en la instalación con DevTools

- Etapa 3 : check-out de los proyectos de desarrollo



Etapas en la instalación con DevTools





DevTools



Demo live

Administrar «DevTools» : bootstrap

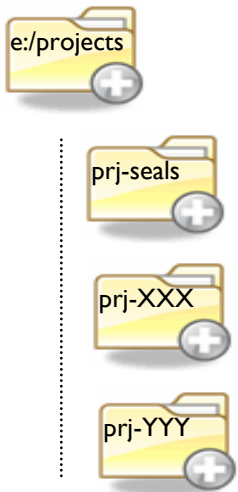
- ▶ Descarga posible vía WEB
- ▶ Auto-extraíble
 - ▶ Compuesto de un JDK – MAVEN – CLIENTE SVN
 - ▶ Carpeta d:/tools y convención de nombre «maven-like»
- ▶ Ejecución de un batch shell en 3 etapas(después de la descompresión)
 - ▶ Etapa 1 : recuperación de SVN config installation
 - ▶ Etapa 2 : ejecución del plugin DevTools
 - ▶ Etapa 3 : check-out de los proyectos de desarrollo



Administrar «DevTools» : bootstrap

Ejecución de un batch shell en 3 etapas (después de la descompresión)

- Etapa I : recuperación, vía SVN, de la configuración



```
REM == =====
REM == CHECKOUT OF PROJECTS SVN CONFIGURATION ==
REM == =====

REM Variable MAVEN_COMMAND is useful only for errors messages
REM
SET MAVEN_COMMAND=STEP - SVN checkout/update DevTools configuration to
    %PROJECT_HOME%
ECHO [BOOTSTRAP] %MAVEN_COMMAND% >> %LOG_FILE%
ECHO [BOOTSTRAP] %MAVEN_COMMAND%

IF NOT EXIST "%PROJECT_HOME%\svn" GOTO svncheckout

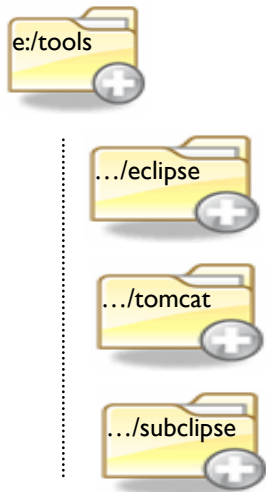
:svnupdate
CD /D %PROJECT_HOME%
CALL svn.exe update --username=%SVNUSERNAME% --password=%SVNPASSWORD%
GOTO :svnend

:svncheckout
CALL svn.exe checkout %CONNECTION_URL% %PROJECT_HOME% --username=%SVNUSERNAME%
    --password=%SVNPASSWORD%
```

Administrar «DevTools» : bootstrap

Ejecución de un batch shell en 3 etapas (después descompresión)

- Etapa 2 : ejecución del plugin DevTools

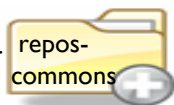


```
REM == =====  
REM == TOOLS INSTALLATION ==  
REM == =====  
  
SET MAVEN_COMMAND=STEP - tools installation with devtools (download  
and extract)  
ECHO [BOOTSTRAP] %MAVEN_COMMAND% >> %LOG_FILE%  
ECHO [BOOTSTRAP] %MAVEN_COMMAND%  
CD /D %PROJECT_HOME%  
CALL mvn.bat -U -e devtools:update  
IF NOT %ERRORLEVEL% EQU 0 GOTO :finErreur
```

Administrar « DevTools » : bootstrap

Ejecución de un batch shell en 3 etapas (después descompresión)

■ Etapa 3 : check-out de los proyectos de desarrollo



```
REM == =====
REM == Projects
REM == =====

SET MODULE=/
SET PROJECT_NAME=repos-commons
SET FOLDER=trunk
SET REVISION=HEAD

IF EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" SET SCM_CMD=update
IF NOT EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" SET SCM_CMD=checkout
IF EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" SET CONNECTION_URL=
IF NOT EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" SET
CONNECTION_URL=%SVN_URL_BASE%\MODULE%\%PROJECT_NAME%\%FOLDER%\%REVISION%
IF EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" CD /D
"%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%"
IF NOT EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" CD /D %PROJECT_HOME%
IF EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" SET CHECKOUT_DIR=
IF NOT EXIST "%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%\svn" SET
CHECKOUT_DIR=%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME%
SET MAVEN_COMMAND=ETAPE - %SCM_CMD% du project (%PROJECT_NAME%)
ECHO [BOOTSTRAP] %MAVEN_COMMAND%
ECHO [BOOTSTRAP] %MAVEN_COMMAND% >> %LOG_FILE%
CALL svn.exe %SCM_CMD% %CONNECTION_URL% -r %REVISION% --depth infinity
%PROJECT_WORKSPACE_TMP%\%PROJECT_NAME% --username=%SVNUSERNAME% --
password=%SVNPASSWORD%
IF NOT %ERRORLEVEL% EQU 0 GOTO :finErreur
```

Administrar « DevTools » : Herramientas

Proyecto POM – almacenado en SVN (configuración de las herramientas)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

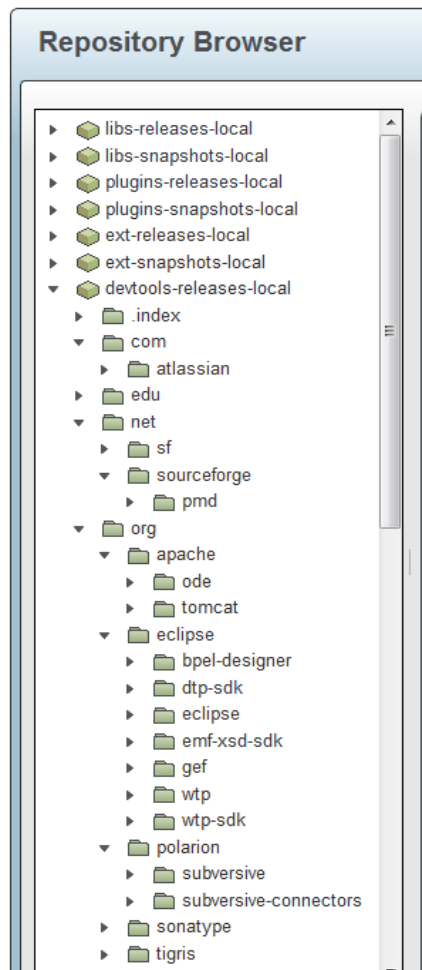
  <modelVersion>4.0.0</modelVersion>
  <groupId>eu.sealsproject</groupId>
  <artifactId>seals-dev-env</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>pom</packaging>
  <name>SEALS development environment</name>

  <pluginRepositories>
    <pluginRepository>
      <id>devtools</id>
      <url>http://aldebaran.dia.fi.upm.es/artifactory/devtools-releases-local</url>
    </pluginRepository>
  </pluginRepositories>

  <build>
    <plugins>
      <plugin>
        <groupId>org.seals</groupId>
        <artifactId>maven-devtools-plugin</artifactId>
        <configuration>
          <toolsDirectory>E:/tools</toolsDirectory>
        </configuration>
      </plugin>
    </plugins>
  </build>
```

Administrar « DevTools » : Herramientas

Proyecto POM – elección de las herramientas y plugins



```
<dependencies>

<!-- ECLIPSE GALILEO JEE EDITION -->
<dependency>
  <groupId>org.eclipse</groupId>
  <artifactId>eclipse</artifactId>
  <version>jee-galileo-SR1-win32</version>
  <type>zip</type>
</dependency>
<!-- -->
<!-- MAVEN ECLIPSE PLUG-IN M2ECLIPSE -->
<dependency>
  <groupId>org.sonatype</groupId>
  <artifactId>m2eclipse</artifactId>
  <version>0.10.0.20100209.0800</version>
  <type>zip</type>
</dependency>
<!-- -->
<!-- SVN ECLIPSE PLUG-IN SUBCLIPSE -->
<dependency>
  <groupId>org.tigris.subversion</groupId>
  <artifactId>subclipse</artifactId>
  <version>1.6.5</version>
  <type>zip</type>
</dependency>

....
```

Administrar « DevTools » : Herramientas

Proyecto POM – instalación personalizada

Más que una simple descompresión

Script ANT.

```
<project name="install-tomcat-6.0.16">  
  <target name="install">  
    <unzip src="${net.sf.mojo.DevTools.installFile}"  
      dest="${net.sf.mojo.DevTools.installTo}" >  
      ...  
    </unzip>  
  </target>  
</project>
```



Lista de los binarios existentes

- ▶ Eclipse Galileo JEE Edition – win 32
- ▶ Eclipse Galileo SOA Edition – win 32
- ▶ Apache Tomcat 6.0.18
- ▶ PMD (plugin Eclipse)
- ▶ CheckStyle (plugin Eclipse)
- ▶ Findbugs (plugin Eclipse)
- ▶ M2Eclipse (plugin Eclipse)
- ▶ Subclipse (plugin Eclipse)
- ▶ Subversive(plugin Eclipse)

....



Administrar « DevTools » : configuración de las herramientas

- ▶ Los plug-ins eclipse
 - ▶ Carpeta /my-project/reference
 - ▶ Fichero platform.xml

```
<!-- MAVEN M2ECLIPSE PLUGIN -->
<site enabled="true" updateable="true" policy="USER-EXCLUDE" url="file:/E:/tools/org/sonatype/m2eclipse/0.10.0.20100209.0800/">
  <feature id="org.maven.ide.eclipse.feature" plugin-identifier="org.maven.ide.eclipse" url="features/org.maven.ide.eclipse.feature_0.10.0.20100209-0800/"
  version="0.10.0.20100209-0800">
    </feature>
  </site>

<!-- CHECKSTYLE ECLIPSE PLUG-IN -->
<site enabled="true" updateable="true" policy="USER-EXCLUDE" url="file:/E:/tools/net/sf/checkstyle/5.0.3.200911042225" >
  <feature url="features/net.sf.eclipsecs_5.0.3.200911042225.jar" id="net.sf.eclipsecs" version="5.0.3.200911042225">
    </feature>
  <feature url="features/net.sf.eclipsecs.migration_5.0.3.200911042225.jar" id="net.sf.eclipsecs.migration" version="5.0.3.200911042225">
    </feature>
  </site>
```

Administrar « DevTools » : configuración de las herramientas

▶ Los plug-ins eclipse

- ▶ Carpeta /my-project/reference
- ▶ Ficheros de preferences

```
#Fichero org.eclipse.team.svn.core  
  
eclipse.preferences.version=1  
  
repositories/c0044acf4cf4001e157aba4f34f0533d=c0044acf4cf4001e157aba4f34f0533d;ht  
tps://svn.seals-project.eu/seals-dev;https://svn.seals-project.eu/seals-  
dev;branches;tags;trunk;true;0be4c893-6316-5341-9597-  
c3b3b8aa611f;https://svn.seals-project.eu/seals-dev;;false;;;22
```

- ▶ Al arranque de eclipse, simple copia de estos ficheros.
-









Más información

http://delicias.dia.fi.upm.es/wiki/index.php/DevTools%2C_a_plug-in_installer



Conclusiones

-  Instalación automática (4 minutos vs unos días)
-  Primero instalación y actualizaciones
-  Multi-proyecto
-  Centralización de los binarios, reutilización y validación
-  Configuración automática por defecto
-  Reversibilidad hacia configuración de referencia

