## Capítulo 9

# Trabajar con el sistema en Eclipse

### Contenido

- 9.1 Introducción
- 9.2 Instalación y configuración
- 9.3 Compilación y ejecución

### 9.1 Introducción

A continuación describiremos los pasos necesarios para poner en funcionamiento el sistema implementado, utilizando el entorno de trabajo de Eclipse. A grandes rasgos, el proceso consistirá en instalar Eclipse y otros componentes necesarios, instalar el código Java dentro de un espacio de trabajo, construir la aplicación, es decir, realizar la compilación y demás operaciones necesarias, y ejecutar el código compilado. Se ha confirmado el funcionamiento de los pasos que se describen a continuación en Windows XP. No puede garantizarse su validez para otras versiones de este sistema operativo.

A lo largo de la enumeración que exponemos seguidamente, denotaremos como "C:" el disco duro de trabajo del sistema, mientras que "D:" será la unidad de CD-ROM, donde habremos insertado el CD del Proyecto.

Los componentes J2SE 5.0, Tomcat 5.5, Eclipse, XMLBuddy y Sysdeo Tomcat Launcher, que se instalan previamente al código del sistema implementado, son gratuitos. Se recomienda acudir a sus respectivas páginas de Internet (véase Bibliografía) para descargarlos. En el momento de la descarga es posible que existan ya versiones de los programas más modernas que las utilizadas en este Proyecto, por lo que los nombres exactos de los archivos de instalación podrían variar con respecto a los que aparecen aquí.

# 9.2 Instalación y configuración

Los pasos a seguir para instalar todo lo necesario para poner el sistema en funcionamiento serían los siguientes:

 Instalar J2SE 5.0 como soporte Java. Para ello, ejecutar el fichero instalador llamado

jdk-1\_5\_0\_05-windows-i586-p.exe

Suponemos que lo instalamos en el siguiente directorio:

C:\Archivos de programa\Java\jdk1.5.0\_05

■ Instalar Tomcat 5.5, descomprimiendo el archivo

jakarta-tomcat-5.5.9.zip

en, por ejemplo,

■ Instalar Eclipse, descomprimiendo el archivo

```
eclipse-SDK-3.1-win32.zip
```

en, por ejemplo,

C:\Archivos de programa\Eclipse

Para poder manejar Tomcat directamente desde Eclipse, debemos añadir un "plug-in" para este programa, llamado Sysdeo Eclipse Tomcat Launcher. Para ello, descomprimir

#### tomcatPluginV31beta.zip

en

#### C:\Archivos de programa\Eclipse\plugins

Otro "plug-in" nos proporcionará una mejora en la edición de archivos XML en Eclipse. El programa se llama, en concreto, XMLBuddy y podemos instalarlo descomprimiendo el archivo

xmlbuddy\_2.0.62.zip

en

#### C:\Archivos de programa\Eclipse\plugins

Colocar en el disco duro todo el código Java de la implementación, incluyendo archivos de configuración, librerías adicionales, etc. Para ello copiamos la carpeta

D:\Código\_Java\_PFC

#### C:\Archivos de programa\Eclipse

Eclipse no tiene instalador y no es necesario llevar a cabo ninguna modificación en el registro de Windows para hacerlo funcionar. Basta con descomprimir el anteriormente mencionado archivo ZIP y ejecutar el programa. Así pues, arrancamos el entorno de trabajo ejecutando

#### C:\Archivos de programa\Eclipse\eclipse.exe

Eclipse utiliza un directorio para almacenar información sobre el proyecto en el que se está trabajando, como, por ejemplo, configuración, valores de variables de entorno, etc. A dicho directorio se le denomina genéricamente espacio de trabajo o "workspace". Al arrancar Eclipse se nos pedirá un nombre para dicho directorio. Nosotros vamos a elegir el siguiente:

#### C:\Archivos de programa\Eclipse\workspace

A continuación, y una vez que ya tenemos Eclipse funcionando, debemos realizar algunas modificaciones en el entorno de trabajo. Es posible que algunas de las opciones que se muestran a continuación aparezcan por defecto con el valor adecuado. En la barra superior de Eclipse seleccionamos el menú "Window" y, dentro de éste, "Preferences...". Aparece una ventana donde realizaremos los siguientes cambios:

■ En el apartado "Java", entramos en "Installed JREs" y añadimos o seleccionamos el jre1.5.0\_05 situado en la ruta

#### C:\Archivos de programa\Java\jdk1.5.0\_05\jre

También en el apartado "Java", entramos en "Build Path" y seguidamente en "Classpath variables" y configuramos la variable TOMCAT\_HOME para que apunte a la carpeta siguiente: C:\Archivos de programa\eclipse\jakarta-tomcat-5.5.9

- En el apartado "Tomcat" hay que realizar varias modificaciones:
  - En "Tomcat version" elegiremos "Version 5.x".
  - Configuramos "Tomcat home" con el valor siguiente:

C:\Archivos de programa\eclipse\jakarta-tomcat-5.5.9

- En "Context declaration mode" elegimos "Server.xml" y establecemos "Configuration file" como:

C:\Archivos de programa\eclipse\jakarta-tomcat-5.5.9\conf\server.xml

 Entrando en "JVM Setting", en el apartado "JRE", debemos elegir el Java Runtime Environment que corresponda con nuestro JDK (es posible que sólo tengamos instalado un JRE, con lo cual no habría necesidad de elección).

A continuación vamos a colocar todo el código de la implementación Java en el espacio de trabajo de Eclipse. Necesitamos importar el código de los dos proyectos Java de los que consta el sistema:

Para el proyecto "shib" seleccionamos, en la barra superior de Eclipse, "File" y después, consecutivamente "Import", "Existing Projects into Workspace", "Select root directory" y, finalmente, elegimos el directorio

C:\Archivos de programa\Eclipse\Código\_Java\_PFC\shib

Para el proyecto "shib-filter", repetimos la operación, eligiendo esta vez como directorio la siguiente ruta:

C:\Archivos de programa\Eclipse\Código\_Java\_PFC\shib-filter

Después de realizar estas dos operaciones deberíamos ver dos carpetas con los nombres "shib" y "shib-filter" en la pestaña "Package Explorer" de Eclipse (ver figura 45).



Figura 45: Los dos proyectos Java en Eclipse

Seguidamente, y usando Eclipse, realizamos algunos cambios en los archivos que Ant usará para compilar y construir la aplicación:

Dentro del proyecto "shib", editamos el archivo build.properties para establecer correctamente la carpeta de Tomcat:

tomcat.home = C:\Archivos de programa\eclipse\jakarta-tomcat-5.5.9

Realizamos la misma operación en el otro proyecto. En esta ocasión modificamos el archivo build.xml dentro de la carpeta shib-filter para que aparezca la siguiente línea:

```
<property name="tomcat.home" value="C:\Archivos de
programa\eclipse\jakarta-tomcat-5.5.9" />
```

El siguiente paso consiste en modificar los archivos de Tomcat para dotarlo de comunicación segura y configurar los usuarios deseados.

 Dentro del proyecto "shib", concretamente en la carpeta src\conf están los archivos de claves para los dos servidores que utilizamos en nuestro sistema. Dichos archivos tienen por nombres:

> idp-example.jks sp-example.jks

Debemos copiarlos a la siguiente ubicación:

C:\Archivos de programa\eclipse\jakarta-tomcat-5.5.9\conf

■ Editamos el archivo de configuración de Tomcat, es decir,

C:\Archivos de programa\eclipse\jakarta-tomcat-5.5.9\conf\server.xml

y añadimos el fragmento que aparecía en la figura 32. Mostramos de nuevo el fragmento en la figura 46 para mayor comodidad.

```
<Connector port="443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="conf/idp-example.jks" keystorePass="exampleorg"
/>
 <Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="conf/idp-example.jks" keystorePass="exampleorg"
/>
<Connector port="9443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="conf/sp-example.jks" keystorePass="exampleorg"
/>
```

Figura 46: Fragmento a añadir al archivo de configuración de Tomcat

A continuación modificamos el archivo de usuarios de Tomcat, cuyo nombre completo es el siguiente:

C:\Archivos de programa\eclipse\jakarta-tomcat-5.5.9\conf\tomcat-users.xml

Añadimos esta línea:

<user username="tomcat" password="tomcat" roles="tomcat"/>

Como último paso de configuración, editamos el archivo de "hosts" de Windows, es decir,

```
C:\{windows}\system32\drivers\etc\hosts
```

donde {windows} representa el nombre del directorio raíz de la instalación de Windows y añadimos las siguientes líneas

127.0.0.1 idp.example.org 127.0.0.1 sp.example.org

Si dicho archivo no existe, tendremos que crearlo con esas dos líneas como único contenido.

# 9.3 Compilación y ejecución

Una vez que tenemos todo nuestro código en el espacio de trabajo y que hemos configurado correctamente las opciones necesarias, podemos lanzar Ant para que compile y construya el sistema Java. Vemos a continuación cómo hacemos esto para cada uno de los dos proyectos Java, "shib" y "shib-filter".

### Compilar "shib"

Dentro de la carpeta del proyecto existirá un archivo llamado build.xml. Nos situamos sobre él y hacemos clic con el botón derecho del ratón. Elegimos entonces la opción "Run as" y seguidamente "Ant build...". Podemos ver esta operación en la figura 47.



Figura 47: Iniciando la compilación del proyecto "shib"

En la ventana que aparecerá a continuación, y que se muestra en la figura 48, nos situamos en la pestaña "Targets". De entre todos los targets que se muestran en la lista, sólo deben tener activada su casilla los siguientes:

- clean-install
- compile
- install.idp.filesystem
- install.sp.filesystem

Es muy importante que se ejecuten en el orden en que los acabamos de enumerar. Para ordenarlos pulsamos sobre el botón "Order..." y modificamos su colocación hasta que la lista que aparece en el espacio "Target execution order" esté correctamente organizada y pulsamos "Run" para lanzar el proceso de compilación y construcción del proyecto Java.

Check targets to execute Name	e: Description	
🔲 🍥 init		
pre-checkin		
	Compile source files	
		-
5 out of 50 selected		
o out of oo beletted		
Sort targets	not colocted for everytion	
Sort targets     Hide internal targets	not selected for execution	
Sort targets     Hide internal targets     Target execution order:	not selected for execution	

Figura 48: Elección de los "targets" a ejecutar

#### ■ Compilar "shib-filter"

De la misma forma, dentro de la carpeta del proyecto "shib-filter" existe también un archivo build.xml. Las operaciones para lanzar el proceso de compilación son las mismas que en el apartado anterior, siendo la única diferencia la lista de "targets" a ejecutar que, en este caso, es la siguiente:

- build
- deploy
- deploy-testapp

Al igual que antes, hacemos hincapié en la importancia de respetar el orden de ejecución mostrado.

Una vez que los procesos de compilación de los dos proyectos Java han finalizado con éxito, nos disponemos a lanzar Tomcat. El "plug-in" que hemos instalado, llamado "Sysdeo Eclipse Tomcat Launcher", añade 3 iconos a la barra superior de Eclipse, para simplificar el manejo del contenedor de aplicaciones. Pulsando sobre el que está situado más a la izquierda arrancaremos Tomcat. Hemos destacado el mencionado icono en la figura 49, rodeándolo con un borde negro.



Figura 49: Icono para lanzar Tomcat

Cuando se arranca Tomcat desde Eclipse, éste corre en modo "debug", lo que generalmente tiene como consecuencia una ralentización de todas sus acciones.

Si todo va bien, Tomcat leerá tres archivos WAR que Ant debe de haberse encargado de copiar previamente en la carpeta "webapps" y desplegará las tres aplicaciones web correspondientes, en concreto las siguientes:

- shibboleth-idp
- shibboleth-sp
- secure

Cuando finalice el "start-up" del contenedor de aplicaciones podremos lanzar el proceso descrito en el apartado 7.4, escribiendo en nuestro navegador la dirección siguiente:

http://sp.example.org:8080/secure/historial.htm