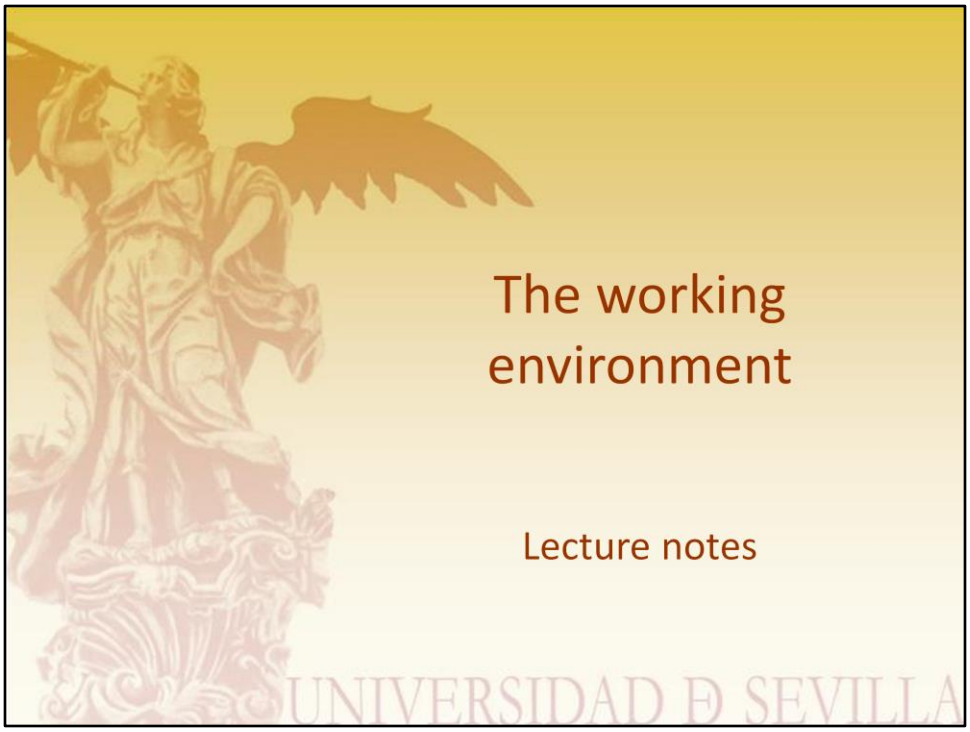


Welcome to this lesson in which our goal's to elaborate on the working environment that we're going to use in this subject.

--

What's a working environment?

UNIVERSIDAD Ð SEVILLA

2

Then, it makes sense to start with this question: what's a working environment? Please, make a point of answering this question before you go on.

This is a good definition

It's a piece of hardware and software that provides a workbench, a workspace template, and a project template

UNIVERSIDAD Ð SEVILLA

3

This is our definition: it's a piece of hardware and software that provides a workbench, a workspace template, and a project template. Think of your working environment as an artefact that provides you with a configured computer, development tools, database and application servers, software components, and so on… everything you need to be productive as a developer.

How are they devised?

UNIVERSIDAD Ð SEVILLA

4

How do you think they are devised? Please, make a point of answering this question before you go on.

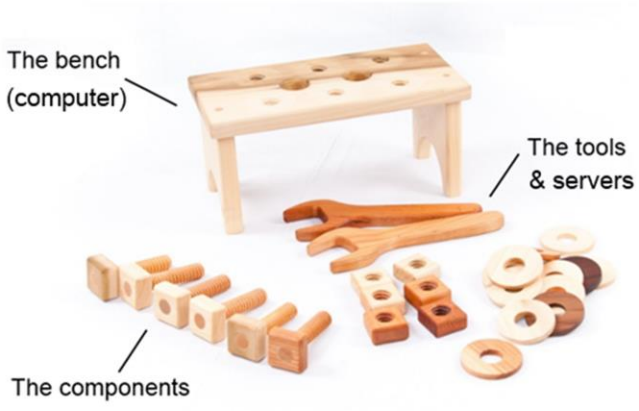
Starting point: local experts

UNIVERSIDAD Ð SEVILLA

5

The usual starting point's commonly provided by your supervisor, your support technicians, your local gurus, … that is, your local experts. Typically, software development companies have worked on a number of projects and have already devised a number of working environment templates that their developers can instantiate. Since you're not working for an actual company, we, the lecturers, will be your local experts.

Step 1: set up your workbench

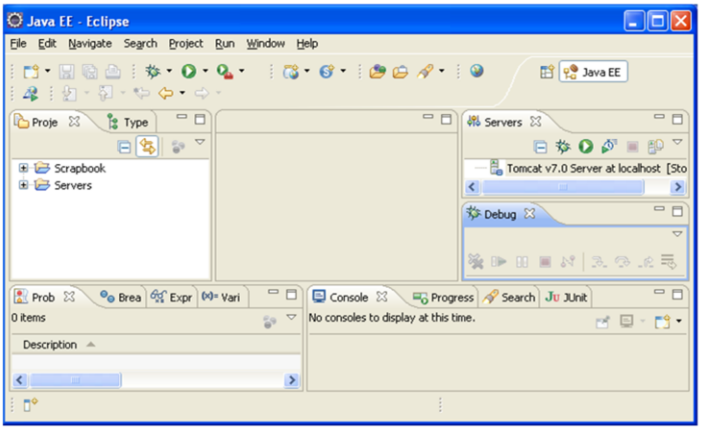


UNIVERSIDAD Ð SEVILLA

6

The first step is to set up a workbench, which is the term that we use to refer to a computer and a number of tools, servers, and components that you require to work on your own projects.

Step 2: instantiate a workspace

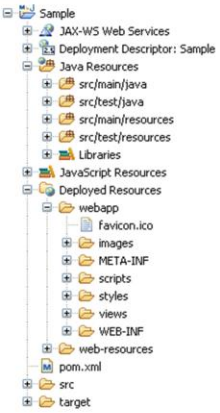


UNIVERSIDAD D SEVILLA

7

Unfortunately, the workbench is of little interest if you don't have an appropriate workspace. This term refers to a configuration of your integrated development environment in which every view you may require is at your fingertips. Typically, your local experts will provide you with a workspace template that you just have to instantiate to start working. Ours will be introduced in this lecture.

Step 3: instantiate/customise a project



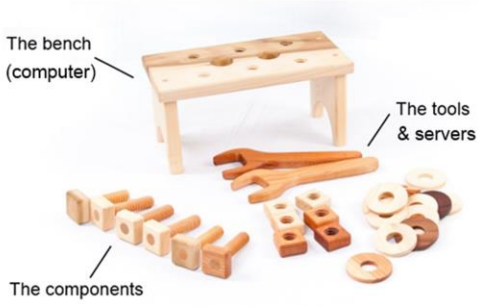
You also need to instantiate and customise a project, that is, a predefined folder structure and a number of files that provide a harness to start working on your own projects. Think of the template project as if it was a complex hello-world project, something you can use to get started quickly. Ours will be presented in this lecture.

And, finally, once you have your own workbench, your workspace, and your project templates, please, do not forget to back them up. Please, do not underestimate the power of a good backup.

This is our roadmap for today's lecture. First, we'll explore workbenches, then a workspace template, and, finally, a project template. We don't report on how to create a backup; you should be able to do that alone!

Let’s start with workbenches.

What's a workbench?



It's a computer and a number of tools, servers, and components that you require to work on your own projects

Please, recall from the introduction to this lecture that we use term workbench to refer to a computer and a number of tools, servers, and components that you require to work on your own projects.

Then, this is our roadmap in this section.

Let’s start with a quick look at your computer.

What's your computer?

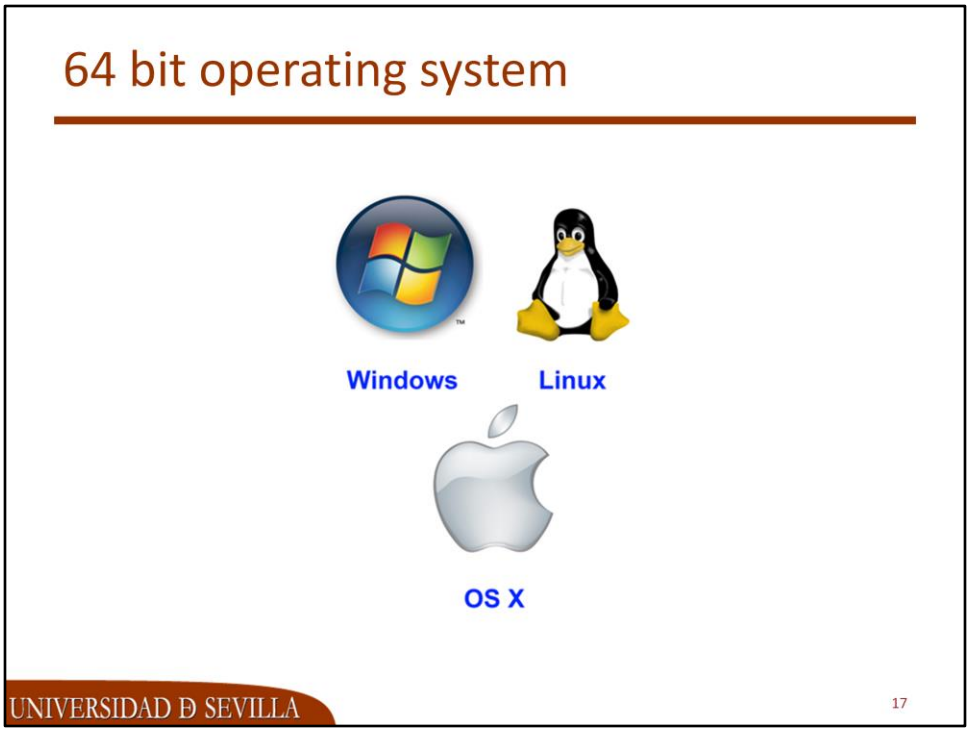
It's a piece of hardware with a developer's operating system

Your computer is your bench, a piece of hardware with a developer's operating system. It's, we're in no doubt, the most important tool in your career as a professional software engineer.

Requirements for this subject

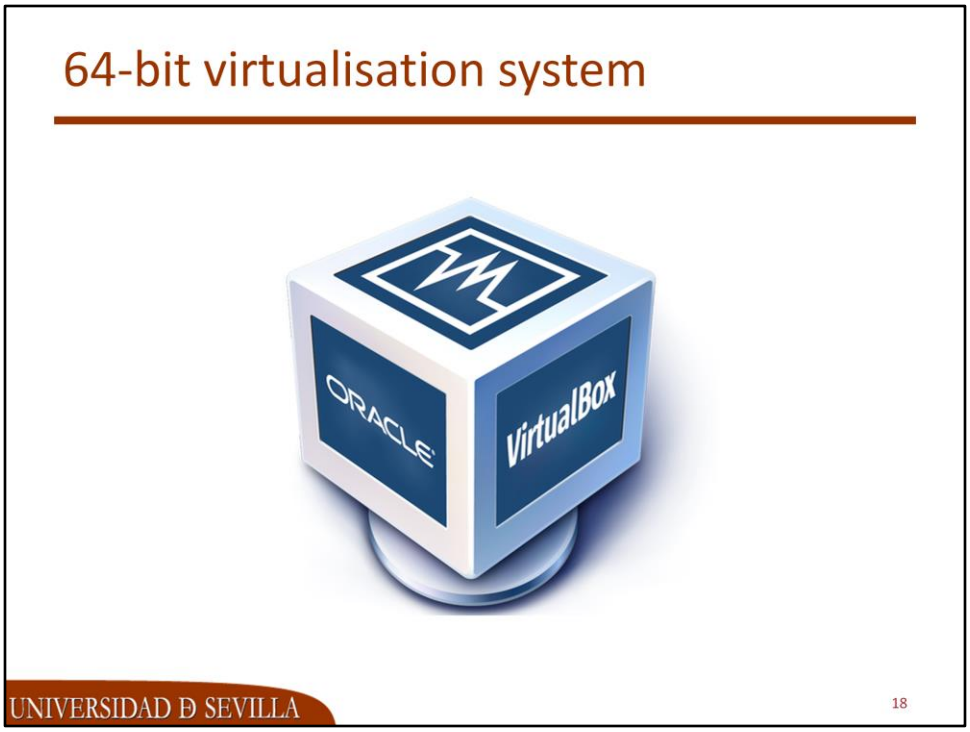
- 15” monitor
- 4 GiB of RAM
- 150 GiB/5700 RPM HD
- Intel i5 processor (5G)
- An 802.11g Wi-Fi card
- An Intel graphic card

These are the minimum requirements for a computer that you can use in this subject: a laptop with a 15.6” monitor, 4 GiB of RAM memory, a 150 GiB hard disk at 5700 RPM, a 5th generation Intel i5 processor, an 802.11g Wi-Fi card, and an Intel graphic card. Note that these are the minimum requirements; the more powerful your laptop, the easier to work.

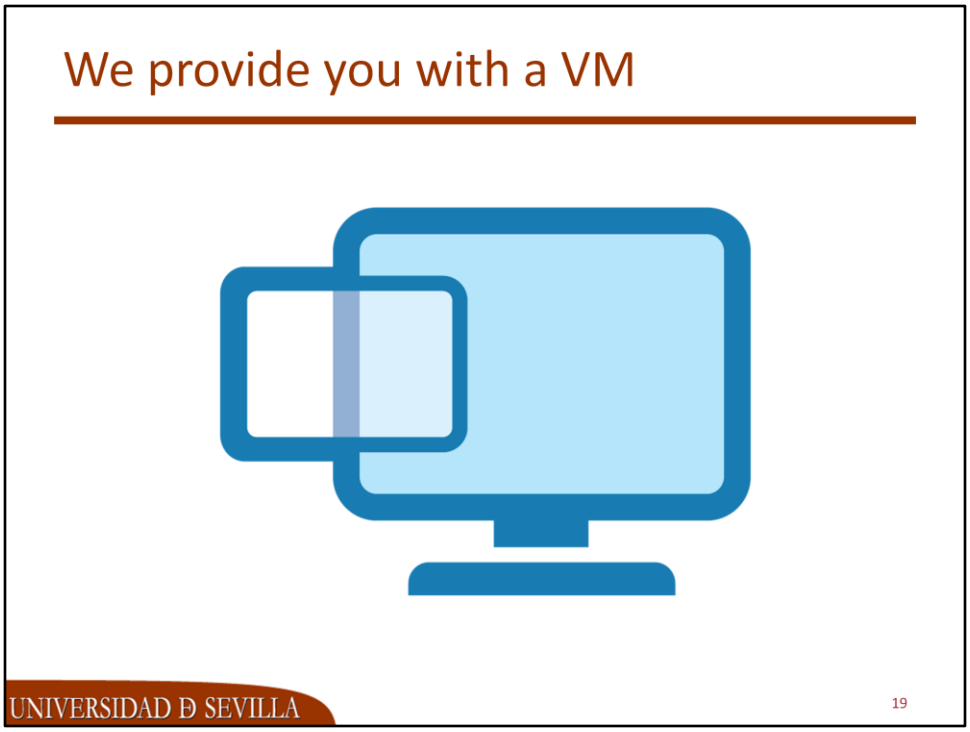


Regarding the operating system, please, choose a developer version of Windows 64-bit, Linux 64-bit, or OS X 64-bit. What matters now is that you choose a developer version: windows XP Home is fun, Sugar Linux is amazing, Mac Kodiak is stunning, and BeOS is very appealing. Unfortunately, they are not appropriate for development purposes, which renders them totally useless in this subject.

NOTE: Windows is available free of charge from DreamSpark; please, consult our laboratory technicians for details. Linux is also available free of charge from many open-source repositories. The default OS X version that is available in Mac computers is a user version; the developer updates can be installed from Apple's app store and require paying a fee. Users of Linux and OS X might well be interested in giving a try to the many pre-configured stacks that are available at http://www.bitnami.com.



Typically, in professional environments, the workbench will be supported by a virtual computer. If you wonder why we recommend using virtual computers instead of physical computers, the answer is simple: it's very simple to expand the memory, the number of processors, or the storage capacity of a virtual machine; it's also very easy to migrate it to a more powerful physical machine if it is necessary; it's also very simple to make a snapshot, make changes to the configuration, and revert them if something goes wrong; it's very easy to move a virtual machine from your desktop computer to a pen drive and then to your laptop or our laboratories. Virtual machines are nowadays very common in the industry, so it makes sense that you get accustomed to working with them. Typically, your local experts will be able to provide you with a ready-to-use virtual machine with a built-in workbench. Regarding the virtualisation system, we recommend that you should use VirtualBox 4.2 or higher. Virtual PC or VM Player are also good choices.



We provide you with a virtual machine that has almost everything you need to work on this subject. Please, download it from our repository.

Can’t you run the virtual machine?

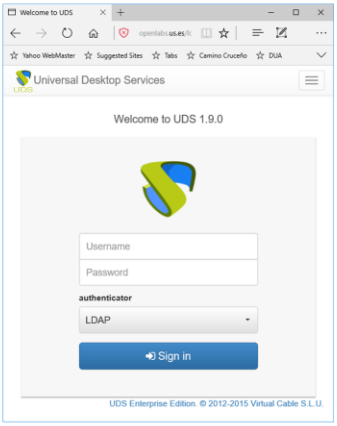
UNIVERSIDAD Ð SEVILLA

20

Can’t you run our virtual machine?

Does it run too slowly on your computer. If this is your case, please, work on your bare metal, that is: install the tools, servers, and components on your physical computer. Recall that there's a repository available in which we provide versions of every tool you need. Install them; they'll work a lot faster on your physical computer than on our virtual machine.

Or a virtual computer from OpenLabs



UNIVERSIDAD D SEVILLA

22

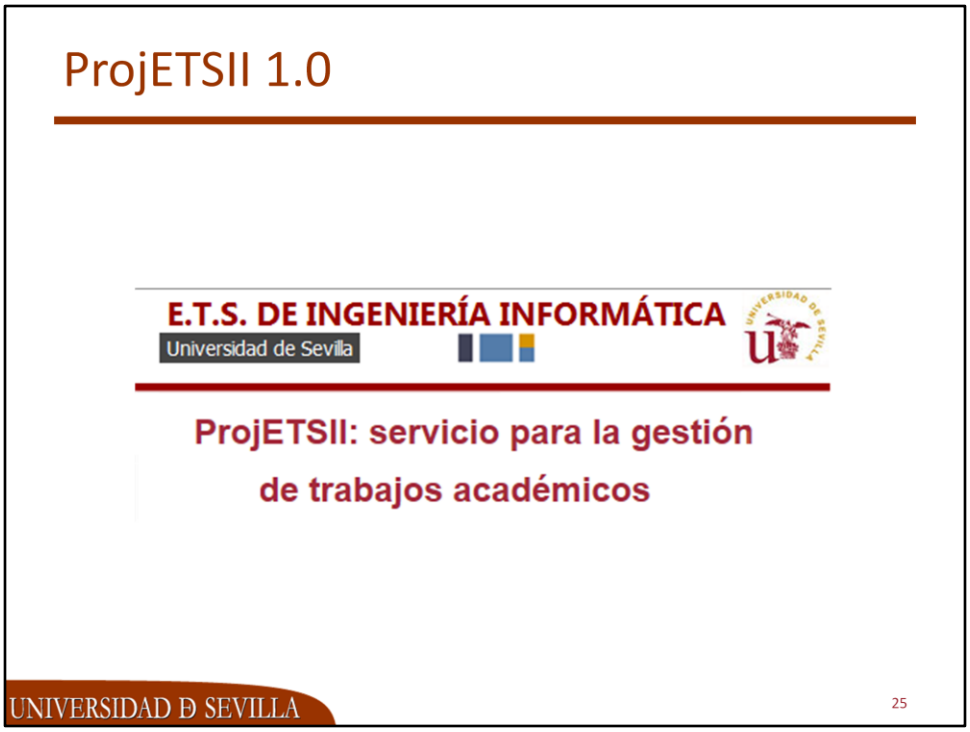
If your physical computer is just not powerful enough to support the tools, servers, and components that we require in this subject, then you can resort to OpenLabs. This is a cloud infrastructure in which the USE provides virtual machines that are set up with everything you need to work in this subject. The virtual machine is called "WXP64-DT". Please, do contact the technicians at OpenLabs if you need to use that infrastructure.

That's enough regarding your computer. Let's go on and delve into the tools that we're going to use.

What's a tool?

It's an artefact that helps you perform a task that is related to working on a project

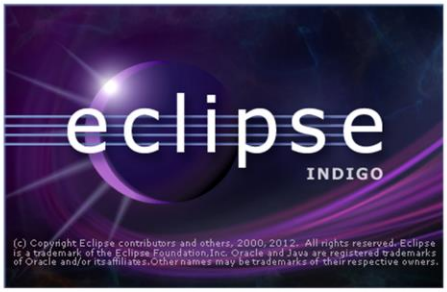
A tool is kind of a gadget. It's an artefact that helps you perform a task that is related to working on a project, where tasks range from managing personnel to compiling, debugging, or deploying your system. In the next slides, we overview the tools that we're going to use in this subject.



One of the most important, although our students typically underestimate it, is ProjETSII. It's a project management tool that is provided by our school. Please, use it to manage your work in this subject. Get accustomed to working with project management tools like ProjETSII since they are the standard in the industry.

JDK 1.7 SE provides the Java compiler, the standard Java library, and the standard Java runtime engine that we'll use to compile and run our applications and servers.

Eclipse Indigo EE SR2

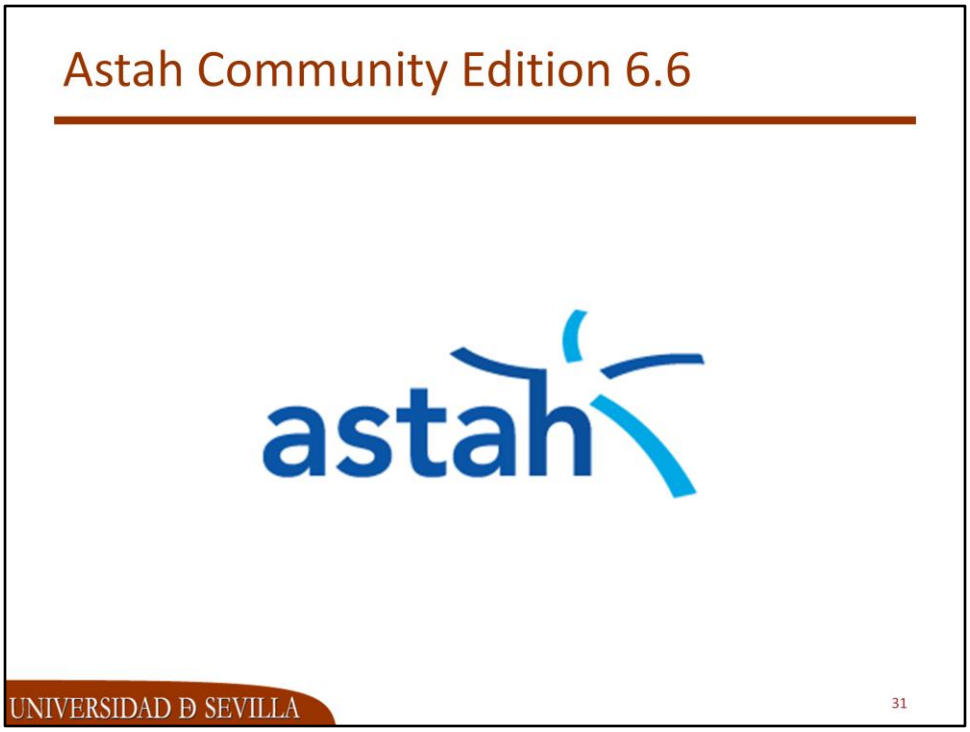


We recommend using Eclipse Indigo EE SR2 as your integrated development environment. Please, make sure that you install exactly this version: Eclipse Indigo (without EE SR2) will just not be enough for this subject.

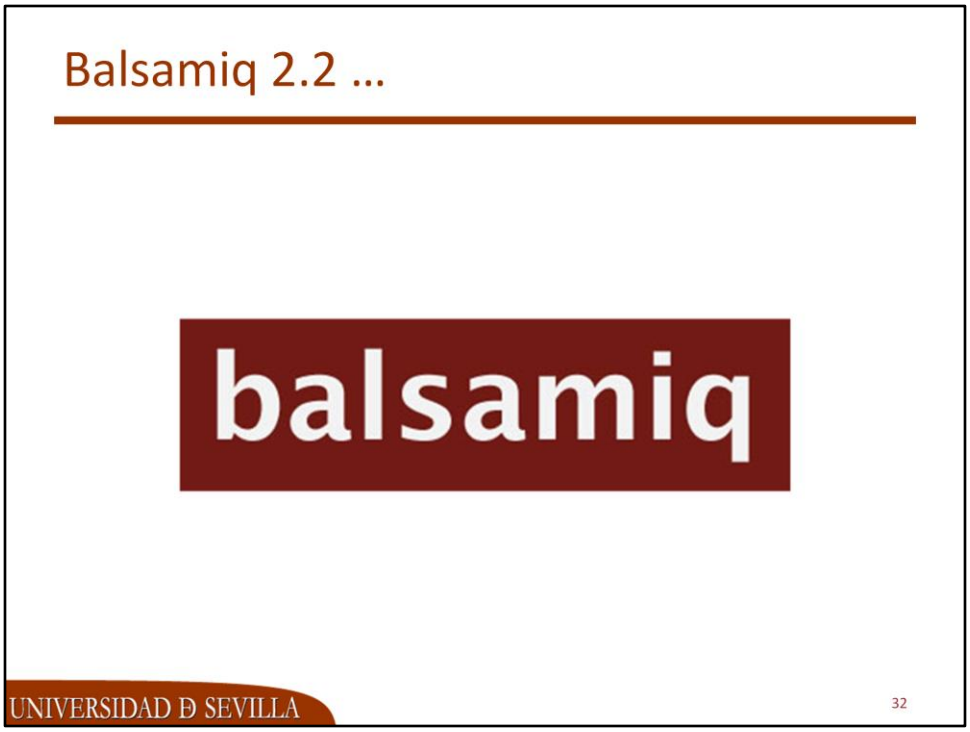
We also recommend to install a plugin called Maven Integration for Eclipse WTP, which is available through the Eclipse Market Place. This plugin is mandatory or, otherwise you won't be able to create a single project.

As you should know, ProjETSII provides a Subversion repository that you should use to store the many different versions of the artefacts that you'll produce during the course. We recommend that you should install a plugin to get connected to your repository, and Subversive is a good choice.

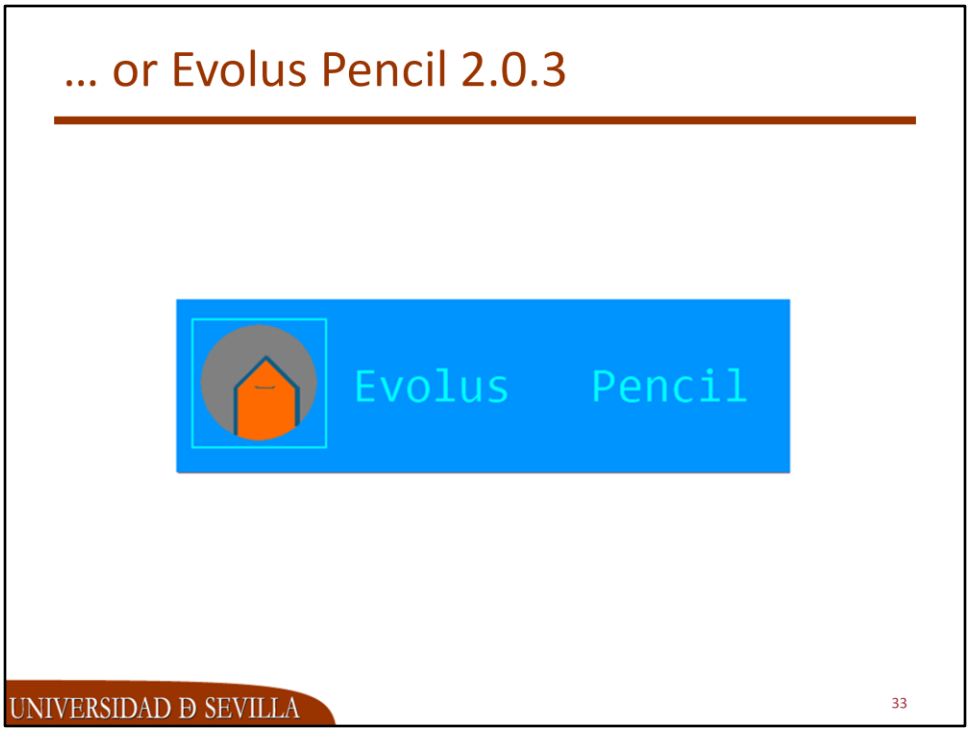
Maven is a tool that provides a command-line interface to manage your projects, that is, cleaning them, compiling them, executing tests, running them, and so on. We won't use it a lot since most of the work will be carried out using Eclipse and its integrated Maven plugin. You however, must get accustomed to this simple tool since it is the one that your lecturers will use to evaluate your deliverables. There's a document called "On your deliverables.pdf" that provides a full explanation of the procedure that your lecturers will use to evaluate your deliverables using Maven. Please, locate this document at the USE's e-learning platform and read it carefully.



Astah is a good tool to create UML diagrams. We'll use it a lot in a few days.



Balsamiq is an excellent tool to create so-called mock-ups, that is, sketches of user interfaces. We have a university license for you; please, contact your lecturer and ask him or her to create an account for you. Please, make sure that you contact him from your USE's email account and that you let him or her know of the names of your partners and their email addresses.



Balsamiq's an excellent tool, but it's a cloud application. That means that you have to be connected to the Internet every time you use it. There's an alternative: Evolus Pencil.

Please, mind the versions! The versions that we recommend or the versions that are available in the virtual machine or in our repository are guaranteed to work well. Shifting to other versions might not work at all.

Let’s now take a look at the servers.

What's a server?

It's a host for your databases or your applications

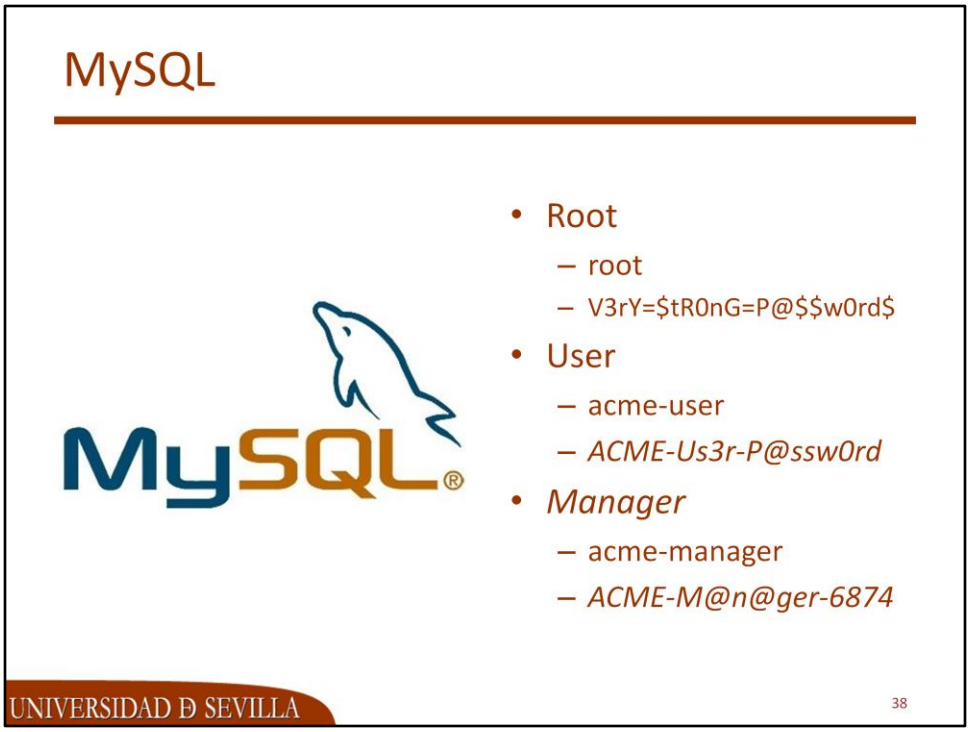
UNIVERSIDAD Ð SEVILLA

36

Simply put: a server's a host for your databases or your applications. Being a host means that the server provides an environment to which you can deploy your databases or your applications; hosting also means that it's the server that cares of handling resources, managing security, user accounts, and so on.

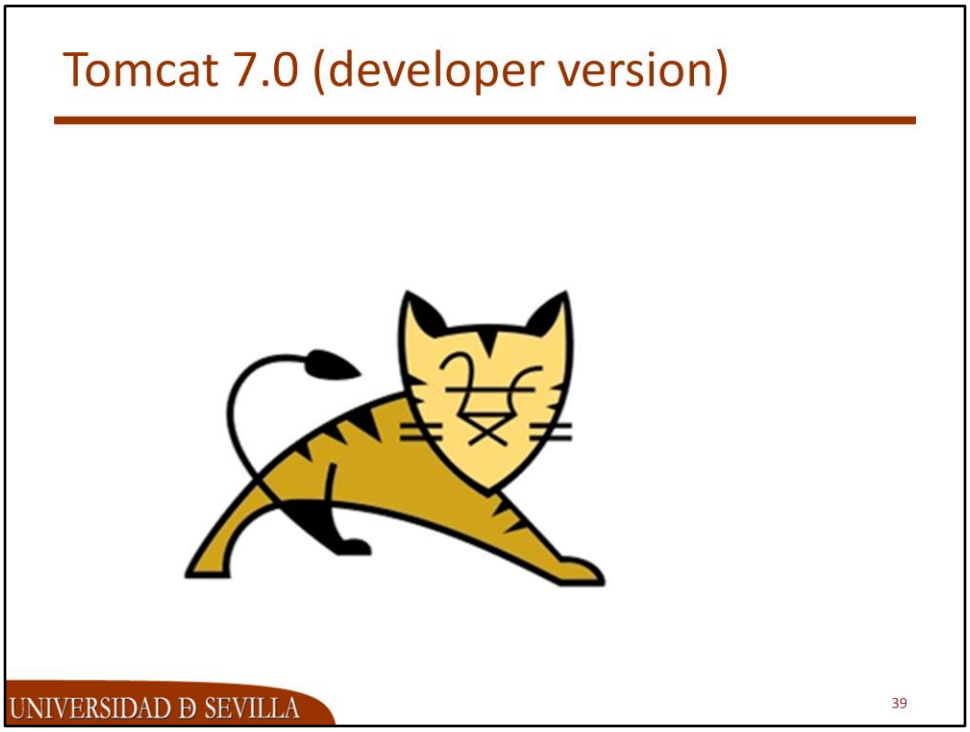
NOTE: please, note that we couldn't find a good visual metaphor to represent servers. We used the same metaphor as we used for tools. To some extent, a server is kind of a tool since you have to use it to perform some tasks in your projects; the difference's that your customers must have these tools in their environments so that they can run your systems. If you happen to find a good metaphor that can be easily integrated into ours, please, do not hesitate to contact us. (If you contact us by email, please, recall that you must use your USE's email account.)

Our database server will be MySQL, which is a well-known database management system. We'll use it to store the data our web information systems manage. The version that we recommend includes several Java connectors, that is, components that will help us connect our applications to the server, and a workbench that will help us manage our databases.

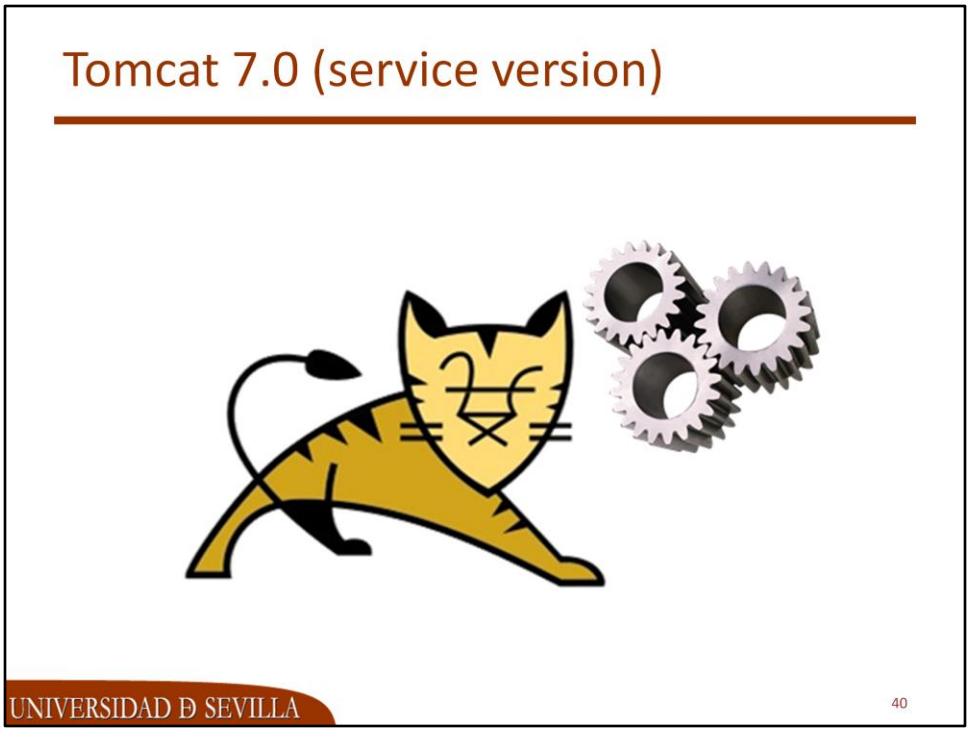


Regarding MySQL, there's not a lot of configuration you must make: simply create an instance called "localhost" and create an administrator account with username "root" and password "V3rY=$tR0nG=P@$$w0rd$". Use your administrator account to create a user account called "acme-user" with password "ACME-Us3r-P@ssw0rd" and a manager account with username "acme-manager" and password "ACME-M@n@ger-6874".

WARNING: please, ***do use exactly*** the usernames and passwords in this slide or, otherwise, you won't be able to use the project templates that we'll provide to you during the course. If you don't know how to create MySQL users, then consult a script called "create-user.sql", which is available in the workspace template that we provide to you.



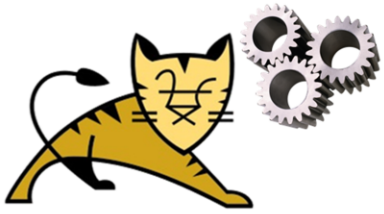
Tomcat's an application server that we'll use to deploy our applications. Most of the time, we'll use the developer version, which is quite well integrated with Eclipse.



The service version of Tomcat is functionally the same as the developer version, but it's intended for production purposes. It's the version that we'll use to deploy the our projects to our pre-production configuration.

NOTE: in the virtual machine that we handed to you, both the developer and the service version of Tomcat are installed in the same folder. That's quite a common practice to avoid update problems. If you wish to keep both versions apart in your own workbench, please, go ahead.

Tomcat 7.0 service



- Use port 80
- Administrator account
  - admin
  - T0mC@t=Adm1n1$trat0R

UNIVERSIDAD Ð SEVILLA

41

The Tomcat service is very easy to install. Just configure the service to use port 80, set the administrator user to “admin” and the password to “T0mC@t=Adm1n1$trat0R”.

WARNING: please, ***do use exactly*** the username and password in this slide or, otherwise, you won’t be able to use the project templates that we’ll provide to you during the course.

Please, mind the versions! The versions that we recommend or the versions that are available in the virtual machine that we handed to you are guaranteed to work well. Shifting to other versions might not work at all.

An additional warning: run your servers securely. Make sure you that you run the MySQL and Tomcat services in a user account with minimum privileges. In Windows, that means running the servers in the “Local System” user account or the “Network Service” account; regarding Linux or OS X, please, consult the documentation of your operating system to find out which the most appropriate user account is.

Finally, let's take a look at the components that we're going to use.

What's a component?

It's a piece of software that you require to assemble and deploy a working system

The components are your nuts and bolts. Simply put: they are pieces of software that you require to assemble and deploy a working system; they are the parts of which a working system is composed, excluding the servers, obviously. We're going to use an array of components in this subject. In the following slides, we provide an overall picture.

Spring components

- spring-context 4.0.0.RELEASE
- spring-webmvc 4.0.0.RELEASE
- spring-orm 4.0.0.RELEASE
- spring-tx 4.0.0.RELEASE
- spring-aspects 4.0.0.RELEASE
- spring-data-jpa 1.4.3.RELEASE
- spring-security-web 3.2.0.RELEASE
- spring-security-config 3.2.0.RELEASE
- spring-security-taglibs 3.2.0.RELEASE
- spring-test 3.2.4.RELEASE

We'll use many Spring components to which we collectively refer to as the Spring Framework.

Hibernate components

- hibernate-c3p0 4.2.3.Final
- hibernate-validator 4.3.1.Final
- hibernate-entitymanager 4.3.0.Final

Hibernate provides components to manage pools of database connections, to validate our data, and to persist them.

MySQL Connector /J

- mysql-connector-java 5.1.26

MySQL provides a component called Connector /J that allows our applications to get connected to a MySQL database server.

NOTE: the virtual machine that we've handed to you includes Connector /J version 5.0.8; this version is used by Eclipse plugins only. The version that our applications will use is 5.1.26.

Apache components

- tiles-core 2.2.2
- tiles-servlet 2.2.2
- tiles-jsp 2.2.2

Apache is a foundation that provides hundreds of components. We'll only use a couple of them that are related to the rendering web user interfaces.

Javax components

- servlet-api 2.5
- jsp-api 2.0
- jstl 1.2

We'll also use some Javax components that are provided by Tomcat…

Miscellaneous components

- jQuery 1.8.3
- jQuery UI 1.9.2
- jMenu 1.9
- taglibs 1.1.2
- displaytag 1.2
- joda-time 2.2
- log4j12 1.7.5
- aspectjweaver 1.7.4

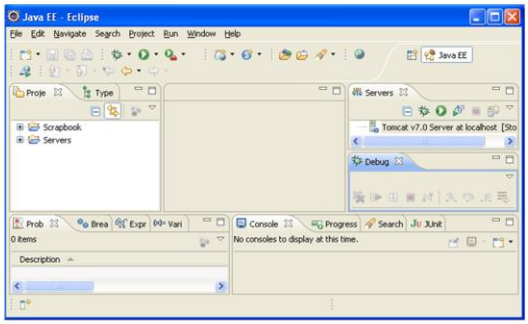
… and a variety of other components, including jQuery, jQueryUI, jMenu, the standard tag libraries, the DisplayTag tag library, the Joda library, the Log4J library, and Aspect Weaver.

Unfortunately, finding a combination of components that work well is not easy at all. We can guarantee that the combination that we recommend works well. Shifting to other versions might not work at all.

These components are available in our repository just for archival purposes. You don't have to download and install them individually. Can you remember Maven? We talked about this tool a few slides ago. Fortunately, Maven will care of downloading and installing the previous components for you. You don't actually have to care about this; you only need to care about using them appropriately.

The previous section reported on everything you need to set your workbench up. It's now time to report on the workspace template that we provide to you.

What's a workspace template?



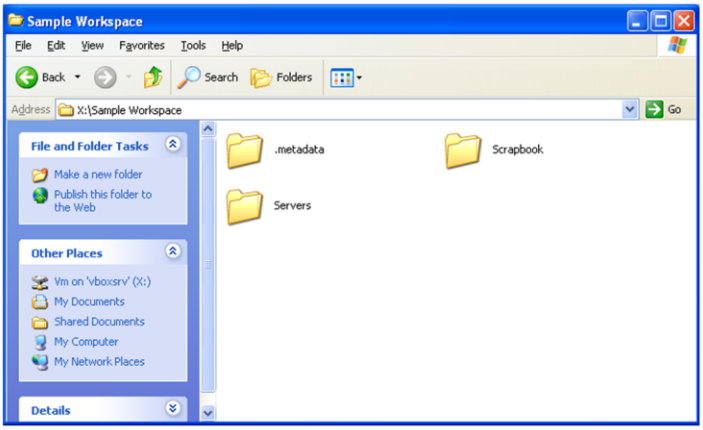
It's a configuration of your integrated development environment in which every view you may require is at your fingertips

Please, recall from the introduction to this lecture that a workspace is a configuration of your integrated development environment in which every view you may require is at your fingertips.

In this section, we'll provide an insight into the workspace template that we provide to you: we'll report on how to instantiate it, how to perform a reset, and how to purge it.

Let's start with how to instantiate our template workspace. It's very easy, we promise.

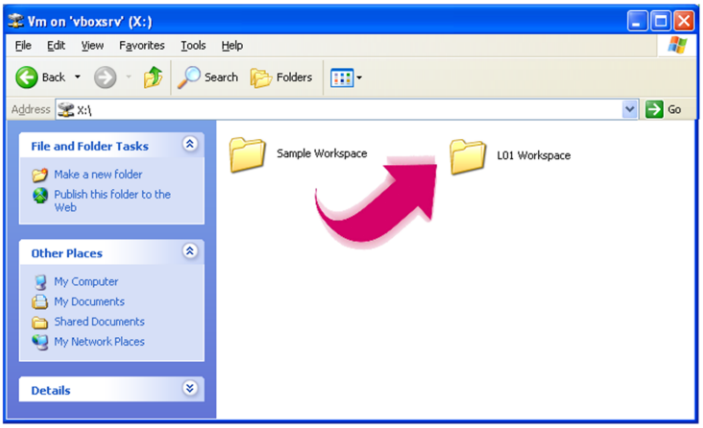
Our sample workspace



There should be a folder called “Sample Workspace” in the materials that accompany this lecture. It should look like the folder in this slide. There’s a folder called “.metadata” in which the different plugins of Eclipse store data. There’s an additional folder called “Servers” that stores data about Tomcat, our application server. The “Scrapbook” folder is intended to store useful documents that you need to use very frequently; in our template, it provides a number of SQL scripts to create databases, users, to grant privileges, and the like.

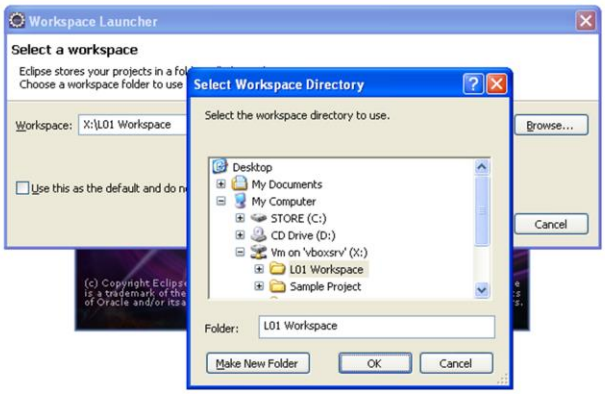
NOTE: the folder that we provide to you might include a version number since these resources are updated regularly.

Copy the template



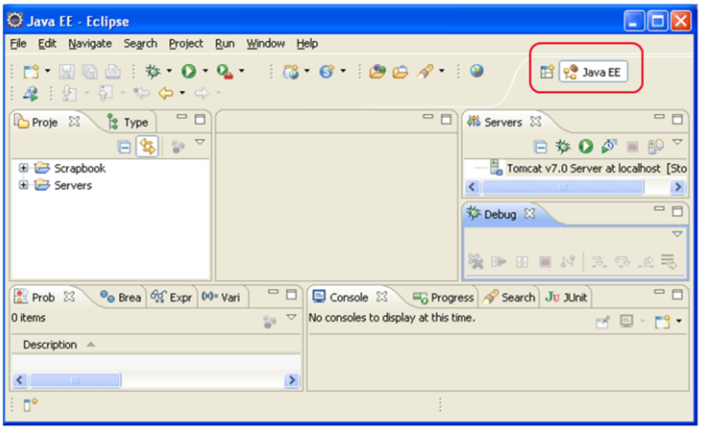
To instantiate the template, you just need to copy folder “Sample Workspace” and paste it using a different name. We recommend that you should instantiate a workspace per lesson, so you should paste it as “L01 Workspace”.

Load it into Eclipse



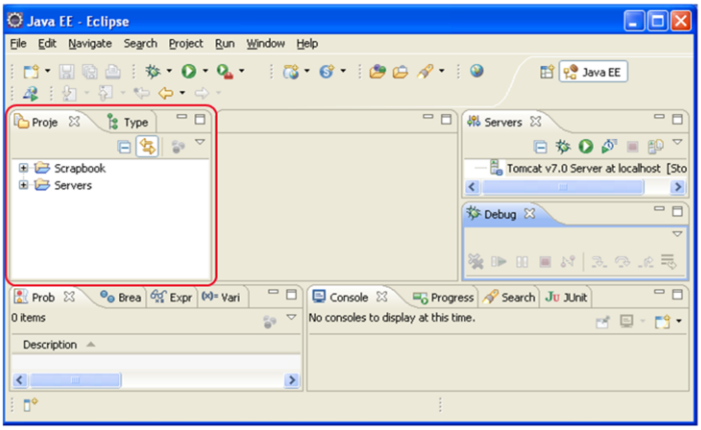
Let's load your first workspace into Eclipse. Launch Eclipse, and select folder "L01 Workspace" when prompted for a workspace.

Default perspective



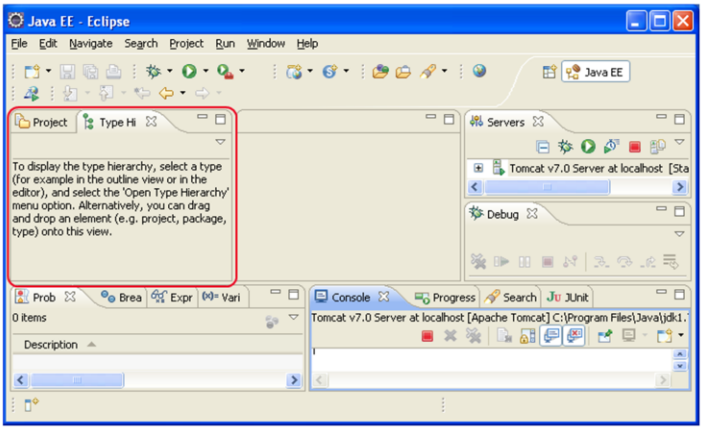
Eclipse should look like in this slide after loading your workspace. This is the default perspective: Java EE. A perspective is a collection of views; you can think of views as if they were specialised toolboxes. In the following slides, we overview the views that we've configured for you in the Java EE perspective.

Default views – Project Explorer



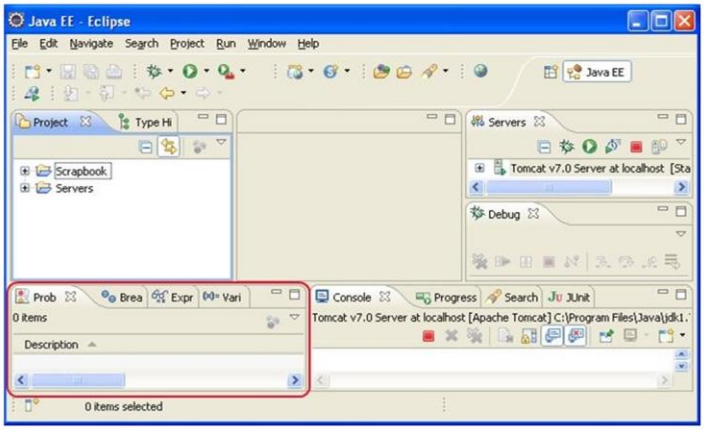
This is the Project Explorer, which you can use to browse the folders in your workspace.

Default views – Type hierarchy



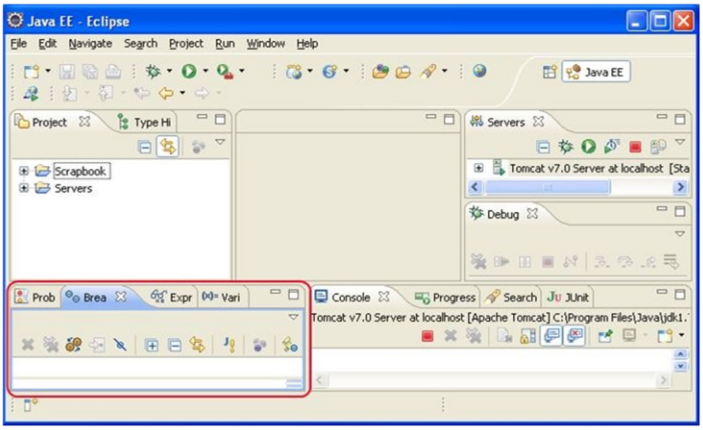
This is the Type Hierarchy view. You can use it to explore your Java classes, their ancestors, and their descendants.

Default views – Problems



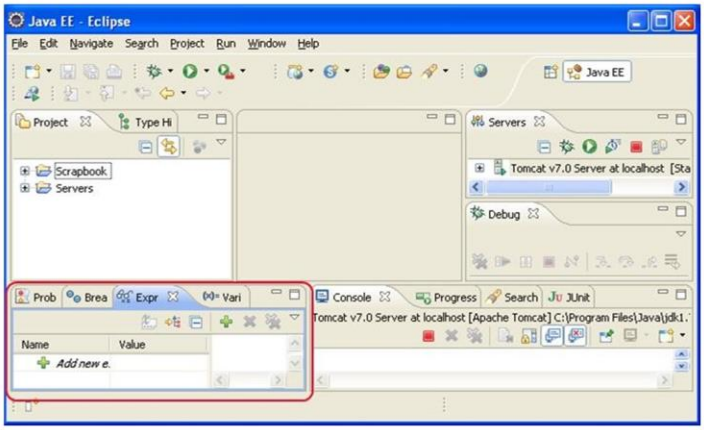
This is the Problems view. If you make a mistake in you Java code, for instance, it'll be shown in this view.

Default views – Breakpoints



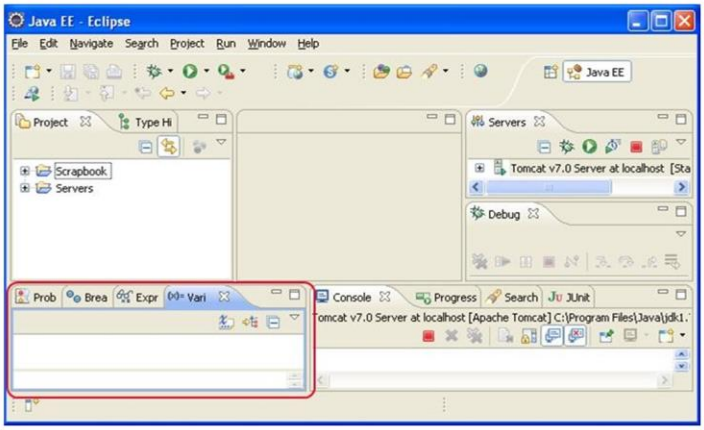
This is the Breakpoints view. You may use a breakpoint to stop the execution of a system when it reaches a particular line of code. This view lists all of your breakpoints and allows you to disable, enable, or delete them.

Default views – Expressions



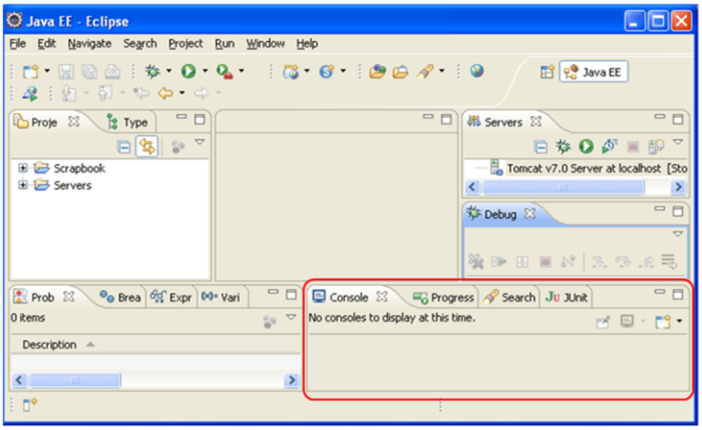
This is the Expressions view, in which you can see and modify the values of your variables (or arbitrary expressions) while debugging a system.

Default views – Variables



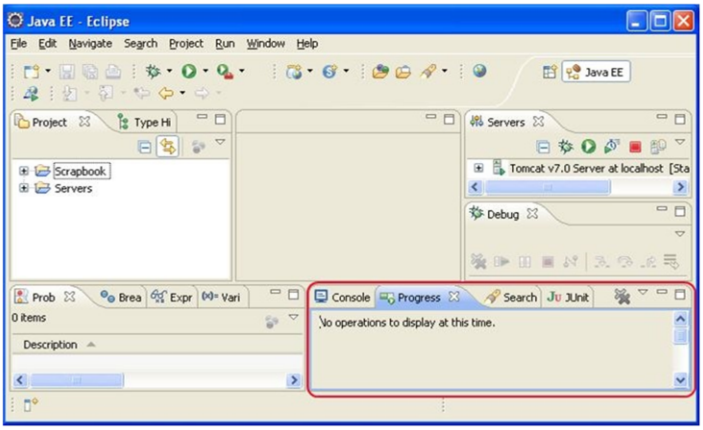
This is the Variables view, which is a restricted Expressions view in which you can only see/modify the variables that are available at the exact line of code that you're debugging.

Default views – Console



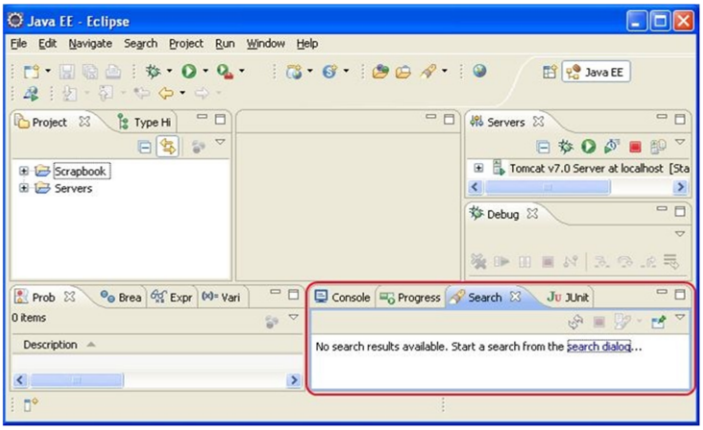
This is the Console view. It's used to show logging messages and to read input from your keyboard.

Default views – Progress



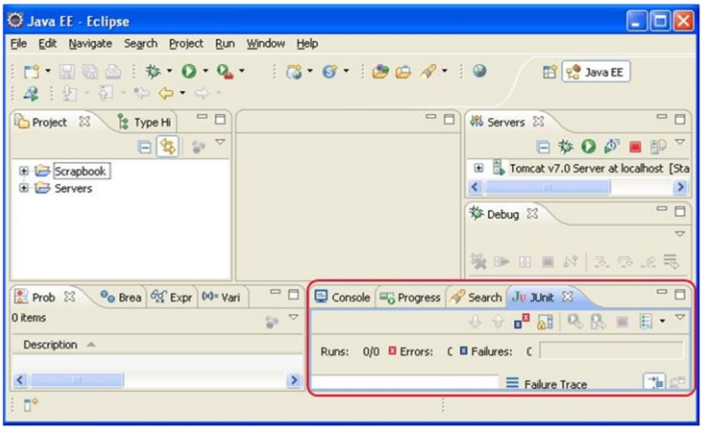
This is the Progress view. It shows information about the tasks that Eclipse's performing in the background, e.g., compiling a project, starting the application server, and the like.

Default views – Search



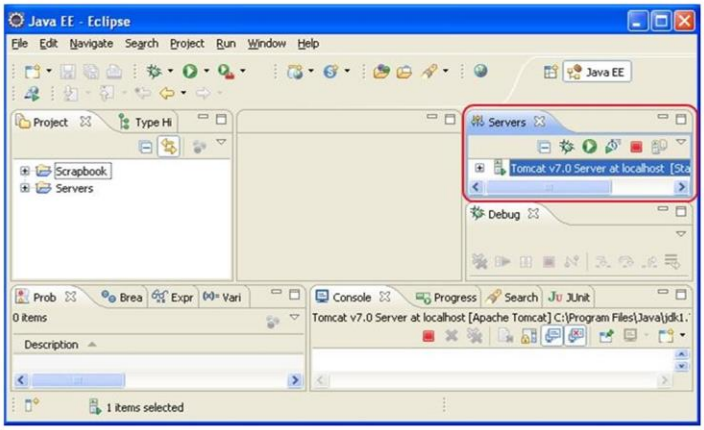
This is the Search view, which shows the results of your searches.

Default views – JUnit



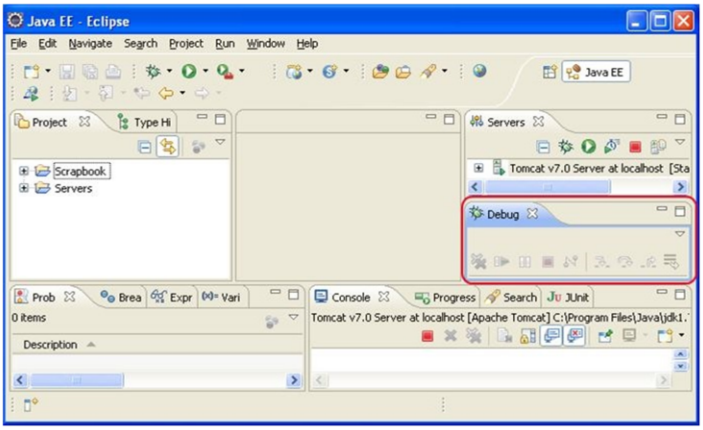
This is the JUnit view. It has to do with testing, so we won't explore it until the Spring semester.

Default views – Servers



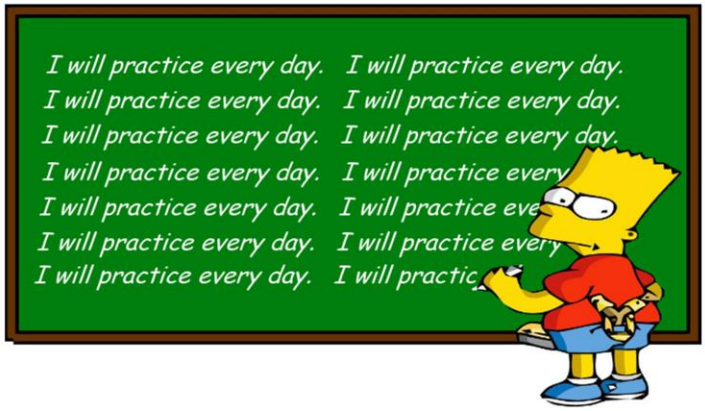
This is the Servers view. It allows to interact with Tomcat, our application server.

Default views – Debug



And, finally, this is the Debug view, by means of which you can debug your systems, that is, execute the code line by line, stop threads, resume the execution, and so on; everything you need to do until you find and correct your bugs.

Practice as you go!



UNIVERSIDAD Ð SEVILLA

74

We know that the majority of our students are not familiar with the previous views, although they are of uttermost importance to work with typical projects. We'll allow you to practice with them as you go. Please, do not underestimate them; try them; get familiar with them.

Now, we'll report on how to reset your workspace.

Great! No problems!

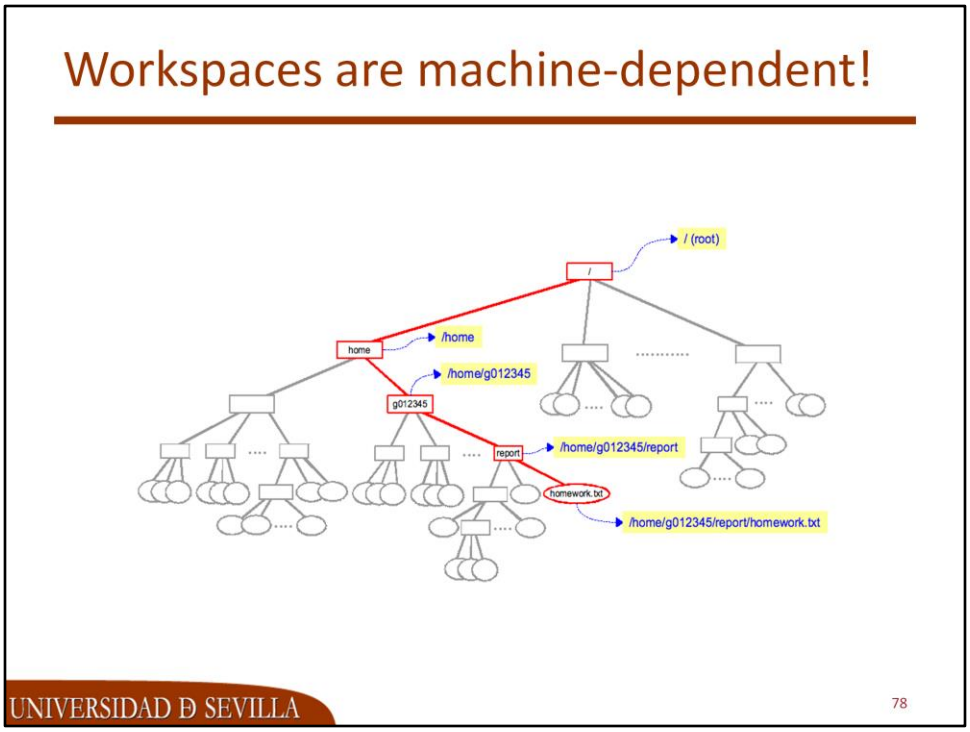
UNIVERSIDAD Ð SEVILLA

76

We're pretty sure that you didn't run in trouble to instantiate and load your workspace. You might have though: Great! No problems!

Look out!

Unfortunately, there might be problems, even though Eclipse doesn't report any such problem now.



The reason's that the configuration in the workspaces is machine-dependent. Eclipse configuration files store many absolute paths that depend on the machine you're using. If you're running the virtual machine that we provided to you, then you don't have to reset the workspace; it's ready to work on that machine. Otherwise, please, keep reading.

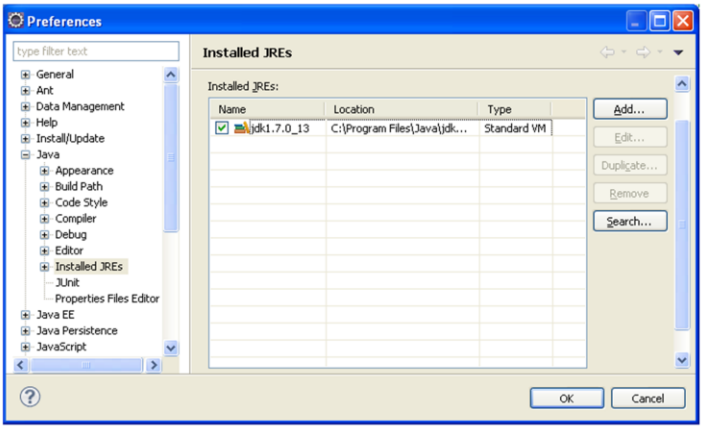
Resetting a workspace

UNIVERSIDAD Ð SEVILLA

79

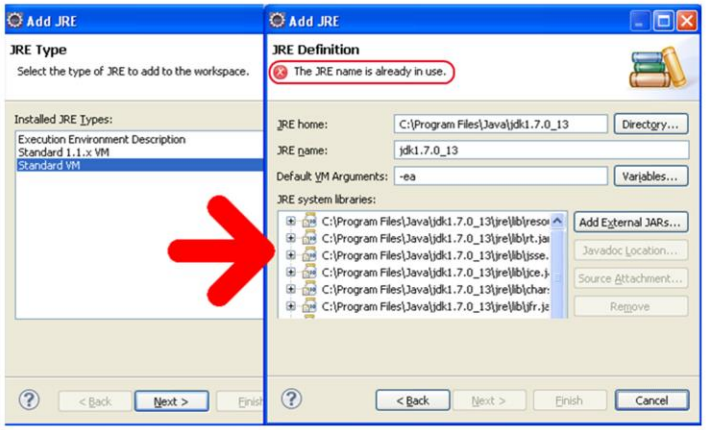
Resetting a workspace is not difficult at all, but it's cumbersome if you're not familiar with Eclipse and its intricacies. Please, pay a lot of attention to the following procedure and repeat it several times until you get very familiar with it.

Configure your JRE (I)



The first step is to configure your Java Runtime Environment, or JRE for short. Please, launch the “Window > Preferences” dialog box, and search for “Installed JREs”. This will bring this dialog box onto your screen. Please, note that this slide shows the dialog box as it appears in our virtual machine; yours may be completely different. If you have the JDK version 7.0 installed (7.0.13 in this case), select it as your default JRE; otherwise, install it and go ahead by clicking on the “Add…” button.

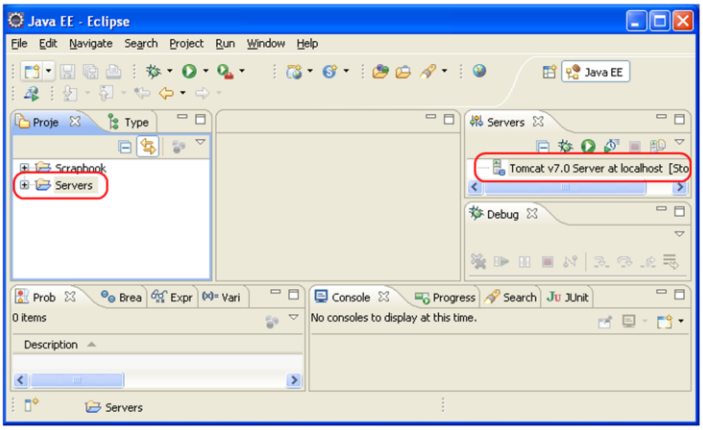
Configure your JRE (II)



Select that you're going to use a "Standard VM" and then click on the "Directory…" button and select the folder where your JDK's installed. Note that the majority of fields are filled in automatically when you select the folder. You only need to add the following parameter: "-ea". This instructs the virtual machine to execute "assert" instructions.

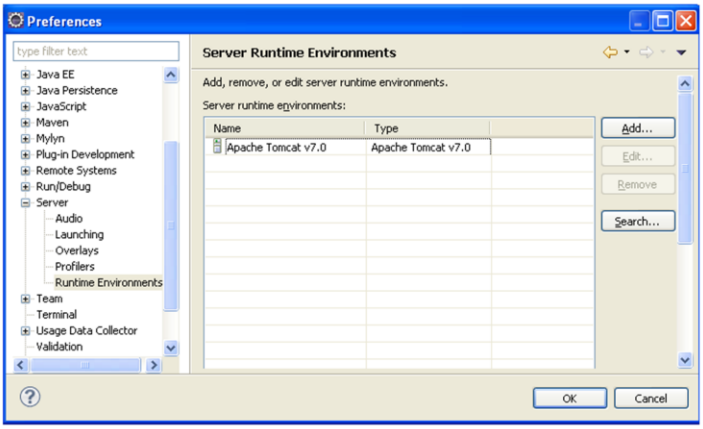
NOTE: note that this screenshot was taken on our virtual machine. It then shows an error message (see the red rounded box) to inform that we cannot configure the JDK twice, and this is the reason why the "Finish button" is disabled.

Delete Tomcat server



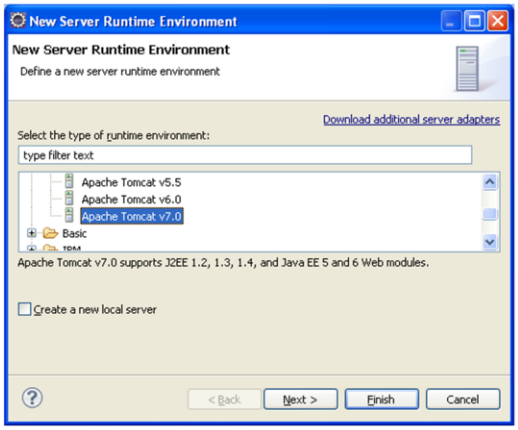
Once the JRE is configured, remove the Tomcat server. That requires you to remove both the folder called “Servers” in the Project Explorer and the Tomcat 7.0 Server from the Servers view.

Configure Tomcat server (I)



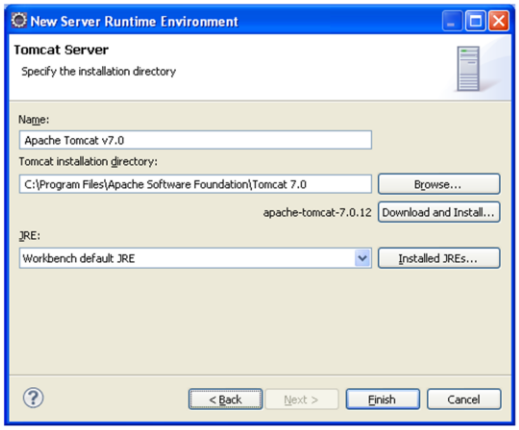
To add a new Tomcat server, you need to launch the dialog box at “Window > Preferences” and then search for “Runtime Environments”. This will show a screen that is similar to the one in this slide. If there are any servers available, like in this slide, please, remove them. Once you’ve removed the available servers, click on the “Add…” button and go ahead.

Configure Tomcat server (II)



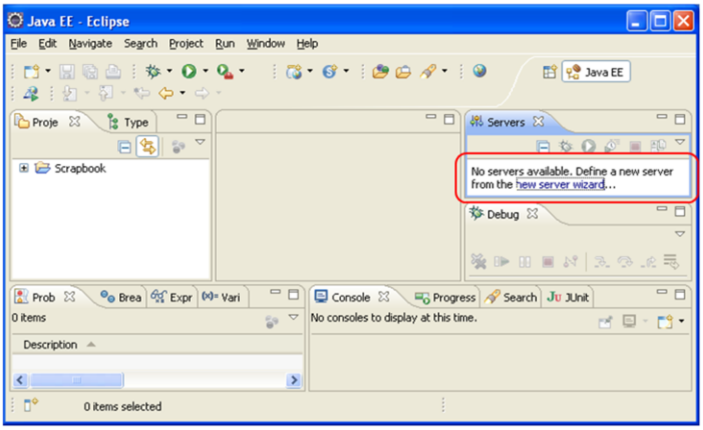
Select that you wish an “Apache Tomcat v7.0” server and click on the “Next >” button.

Configure Tomcat server (III)



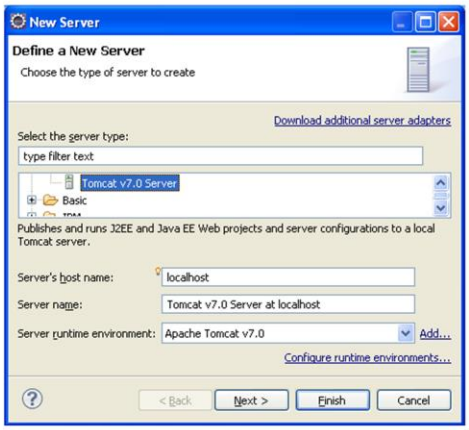
Click on the “Browse…” button and select the folder where Tomcat was installed. Make sure that you’ve selected the “Workbench default JRE” as your JRE, and click on the “Finish” button.

Instantiate Tomcat server (I)



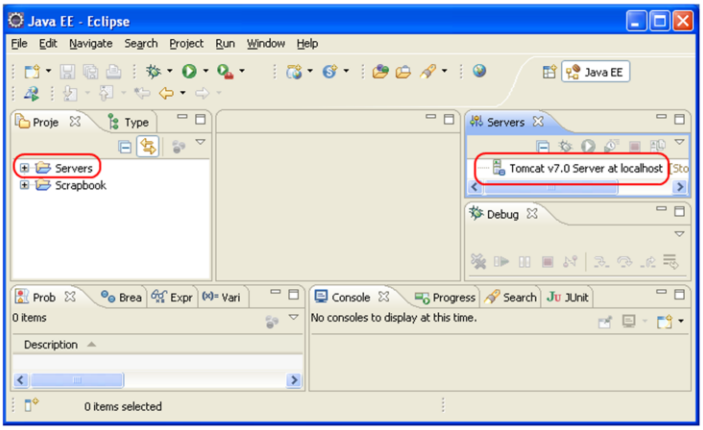
The previous steps have registered a new server in Eclipse. You now have to instantiate it. To perform this task, please click on the “new server wizard” link in the Servers view.

Instantiate Tomcat server (II)



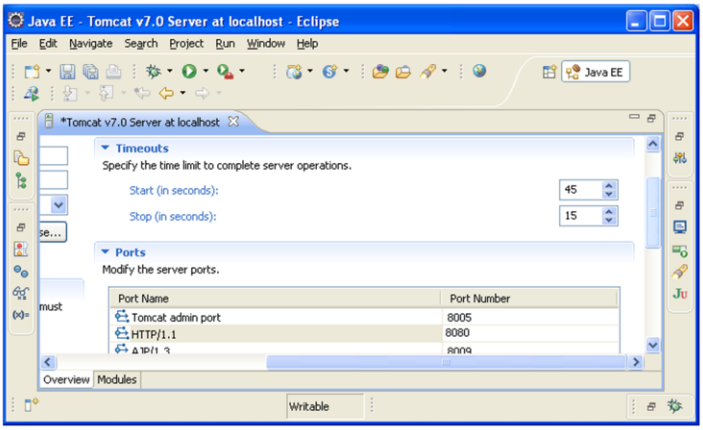
Unfortunately, this dialog box is a little confusing. Please, take a look at this slide and make changes in your dialog box so that it looks the same. When finished, please, click on the “Finish” button. You can click on the “Next >” button, but there’s nothing else you can configure now.

Instantiate Tomcat server (III)



This is how your workspace should look now. Eclipse's instantiated Tomcat and it's created a folder called "Servers" and it has also added the new instance to the Servers view.

Instantiate Tomcat server (IV)

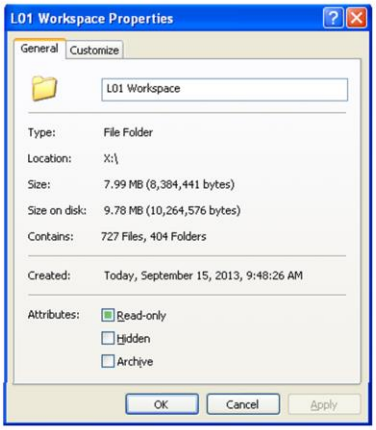


Please, double click on the "Tomcat v7.0 Server" in the Servers view. This will display a dialog box in which you can configure many parameters of Tomcat. Regarding this subject, you only have to make sure that the timeouts are long enough for your computer and that the ports are configured as shown in this slide. If your computer barely meets the minimum requirements that we presented at the beginning of this lecture, please, change the timeouts to 120 seconds or 180 seconds. That should be enough for very slow computers.

NOTE: it might be the case that the default HTTP/1.1 port is "80" instead of "8080". In such cases, please, make the appropriate changes. It's strongly recommended that your default port be "8080" in a developer's configuration.

The final topic regarding workspaces is purging them.

Check the size of your workspace



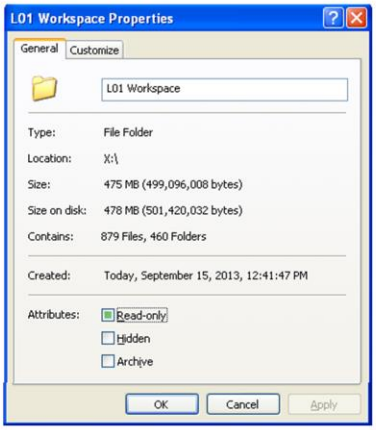
UNIVERSIDAD D SEVILLA

91

Please, check the size of your workspace. It should be about 8 MiB, as shown in this slide.

NOTE: the actual size may not be the same as shown in this slide, but it must be very similar.

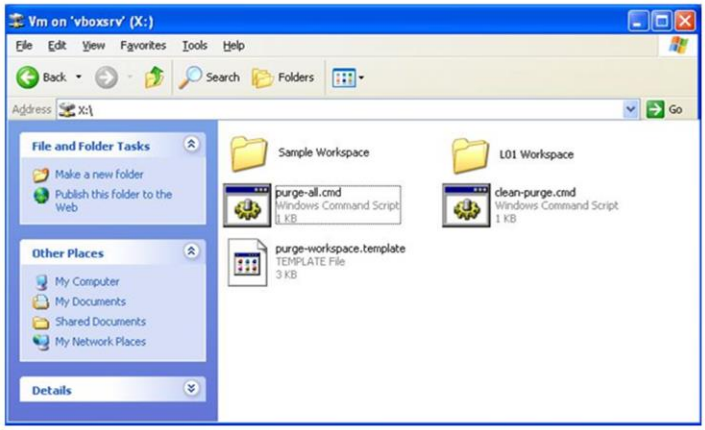
Check it later!



Check it later, when we import the first project. The size of your workspace will increase to about 475 MiB! Even if the project's as small as 900 KiB. That's common and there's no way to avoid it. That means that the average size of your workspaces will be roughly 500 MiB.

NOTE: the actual size may not be the same as shown in this slide, but it must be very similar.

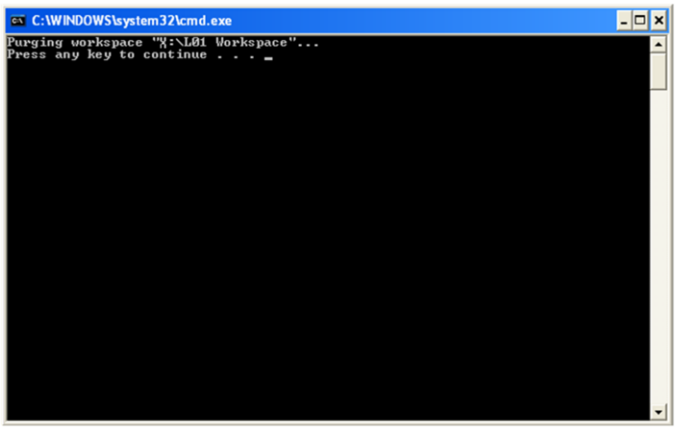
The purging scripts



We assume that you will store all of your workspaces in the same folder. (That's a good recommendation.) We've provided you with three scripts that allow to purge your workspaces, that is, to remove useless data from them. Don't worry, those data will be re-created the next time you load the workspace into Eclipse.

NOTE: only Windows versions of the scripts are provided. If you produce a Linux or an OS X version, please, let us know.

The purge-all.cmd script

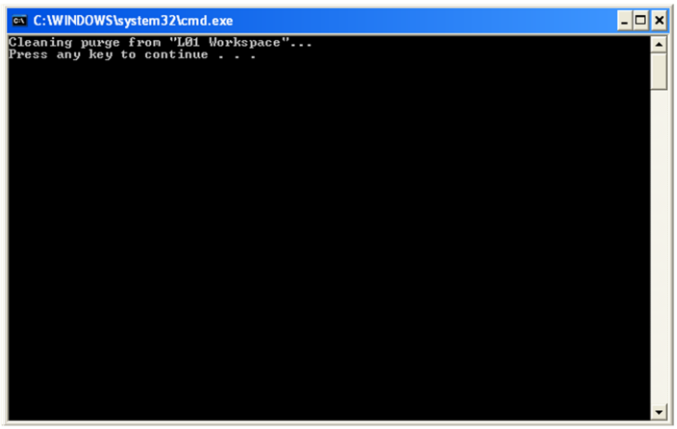


UNIVERSIDAD Ð SEVILLA

Double click on the “purge-all.cmd” script, and you’ll get a screen like the one in this slide. This script walks through every folder called “*Workspace*” and purges it.

WARNING: the lecturers have made a point of polishing this script to ensure that it works well. We, however, cannot guarantee that it works as expected in every possible situation, chiefly if you’re not using Eclipse Indigo EE SR2. Please, use these script at your own risk.

The clean-purge.cmd script



UNIVERSIDAD D SEVILLA

95

The “purge-all.cmd” script creates an additional script and a log file in every folder whose name matches the following pattern: “*Workspace*”. You can clean them all by executing the “clean-purge.cmd” script. This is how executing the script looks like.

When to purge?

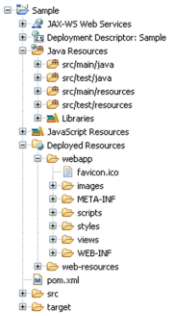
UNIVERSIDAD Ð SEVILLA

96

Please, try purging your workspaces later and check that our scripts effectively help reduce the amount of storage they require. Also make sure that you understand that you won't usually have to purge your workspaces. Purge them only if you're running out of space, if you wish to back them up, if you need to copy them to a pen drive that doesn't have enough space, if you wish to upload them to a cloud storage service… don't purge them systematically and, obviously, don't purge a workspace while it's loaded in Eclipse!

Let's move on! Let's talk a little about our project template.

What's a project template?



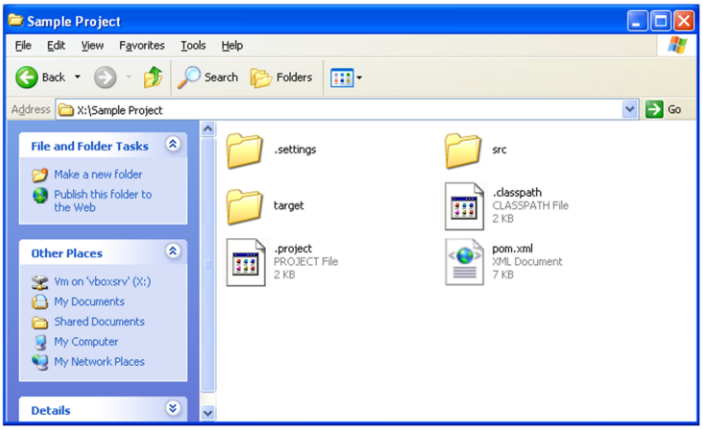
It's a sample project that we'll use as a starting point to develop new projects

Please, recall from the introduction to this lecture, that a project template is a sample project that we'll use as a starting point to develop new projects. In the previous lecture, we introduced the architecture that we're going to use in this subject and we also introduced a framework to instantiate it. The project template that we provide in this lecture is just an instantiation of that framework.

We'll talk about the following topics in this section: first, we'll talk about instantiating our project template, then about a file called "pom.xml", and then about creating the accompanying database (it's a project template for a web information system; it shouldn't be surprising then that it requires a database); finally, we'll get a sample project up and running so that you can get familiar with the look and feel of the projects that we're going to develop during the course.

Let's first provide an insight into how to instantiate the project template.

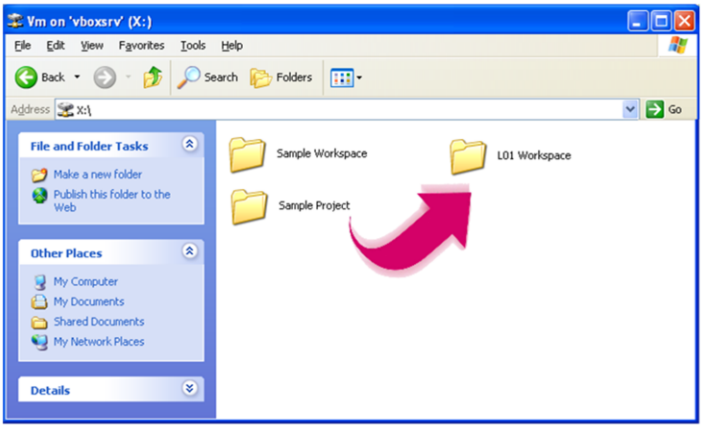
Our sample project



There's a folder called "Sample Project" that accompanies this lecture. It should look like in this slide. There's a folder called ".settings" and two files called ".project" and ".classpath" that have data required by Eclipse; please, never make changes to this folder or these files since this might render your project unusable. There's another folder called "target" to which Eclipse (actually Maven) will output the results of compiling, testing, or packaging the project; you may freely delete this folder if necessary since it's reconstructed on the fly. Folder "src" provides the source code to your Java classes.

NOTE: please, note that these resources are updated regularly. So the actual folder that you've got with this lesson might include a version number.

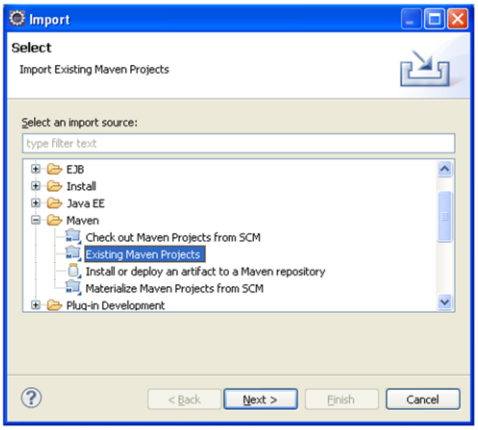
Copy the template



UNIVERSIDAD Đ SEVILLA

Instantiating the project template is very simple: just copy its entire folder into your workspace and rename it appropriately. Please, recall that we strongly recommend that you should have an independent workspace for each lesson. So you should copy folder “Sample Project” into folder “L01 Workspace” and rename it as “Sample”.

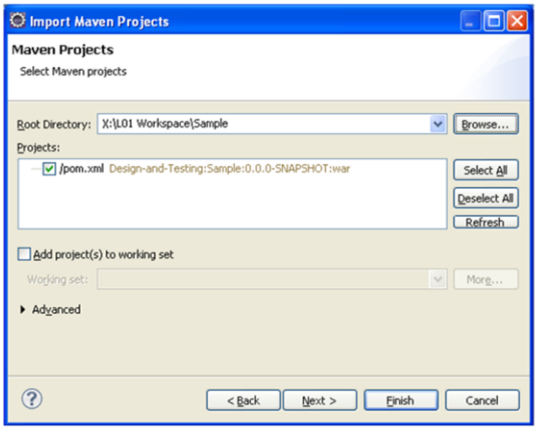
Import the template (I)



UNIVERSIDAD D SEVILLA

Once the folder's been copied, you have to import the project that it contains into your workspace. To do so, launch the "File > Import…" dialog box in Eclipse. Search this dialog box for "Existing Maven Projects" and then click the "Next >" button.

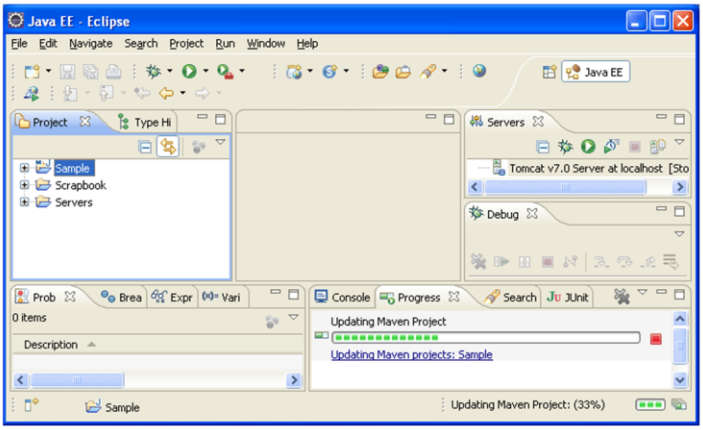
Import the template (II)



UNIVERSIDAD D SEVILLA

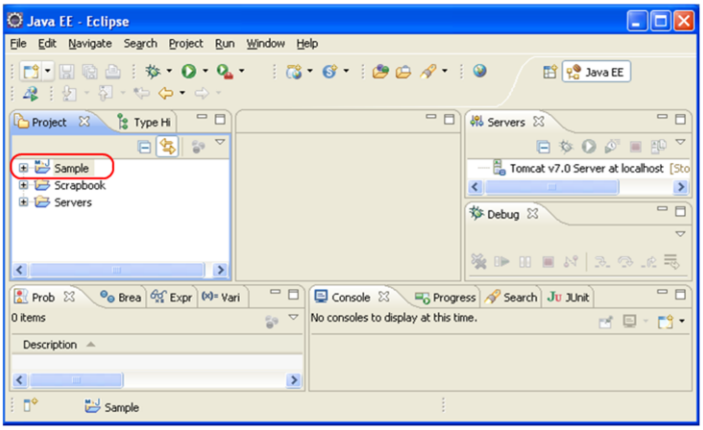
Click on the “Browse…” button and select folder “Sample”. Please, make sure that you select the “Sample” folder in your workspace, not the original “Sample Project” folder, which should now be read-only or reside somewhere else. Check the “pom.xml” file if it is not checked by default, and click on the “Finish” button. If you click on the “Next >” button, you’ll get a new dialog box to select Maven goals; just click the “Finish” button and go ahead.

Please, wait



Please, wait. Maven will take a little to import the project into your workspace. Switch to the Progress view to see what's happening: Maven's downloading the required components, it's installing them to a local repository, and it's compiling your project.

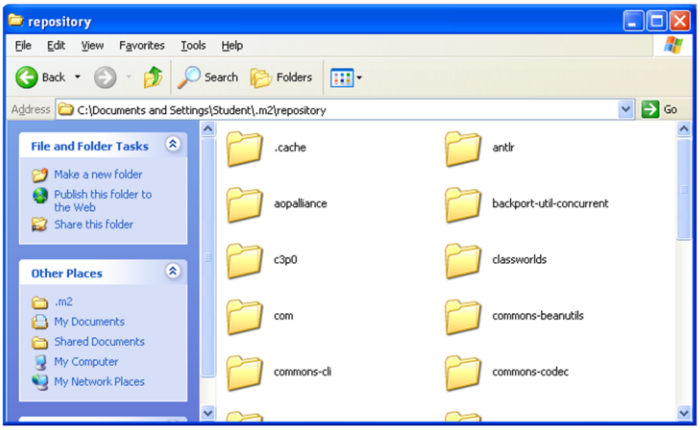
And here it is!



If no problem has occurred, then Eclipse should now look like in this slide. Note that there's an additional folder in the Project Explorer view that corresponds to the project that we have just instantiated.

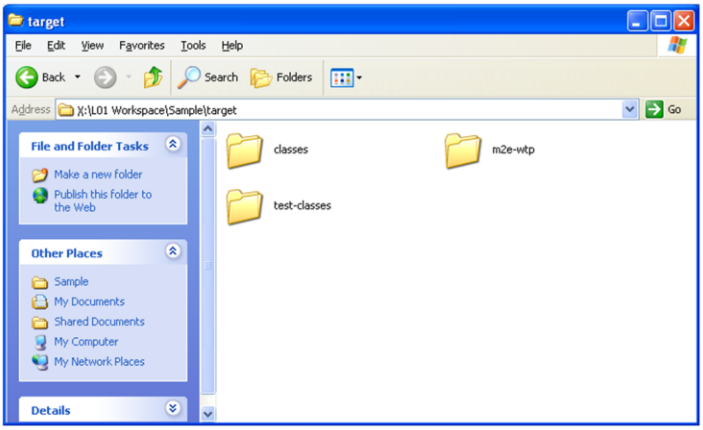
WARNING: this step should take a few seconds in our virtual machine since the repository of components was downloaded prior to releasing the machine to you. If you're not working on our virtual machine, this step may take 20-30-40 minutes depending on your computer and your Internet connection. Note that you have to be connected to the Internet the first time that you instantiate a project since, otherwise, Maven won't be able to download the components that it requires.

The Maven repository



Please, peek at Maven's repository once it's finished working. It's a folder called ".m2" in your home folder. It should look, more or less like in this slide.

The target folder



Take also a look at the "target" folder within your project's folder. It stores intermediate compilation files, which are re-generated automatically.

Let's now delve into how to customise a project that has just been instantiated.

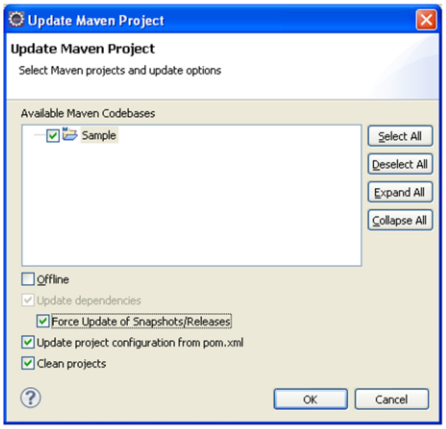
Change your pom.xml file

```
<groupId>Sample</groupId>
<artifactId>Sample</artifactId>
<version>0.0.0-SNAPSHOT</version>

<name>Sample</name>
<url>http://www.example.com</url>
<description>This is a sample template project</description>
```


The first thing you have to do is to locate a file called “pom.xml” and change the information in the elements shown in this slide to adapt them to your project. “artifactId” and “version” refer to the name of your project and its version number, respectively. “groupId” refers to a name that you give to a number of related projects; we recommend that you should use the name of the subject or the name of the lesson. “name”, “url”, and “description” are free-text fields that you can use to describe your project.

Let Maven update your project



It's necessary that Maven updates your project configuration after you change your "`pom.xml`" file; to do so, please, right-click your project, select option "Maven", and then option "Update project". This opens a dialog box in which you just have to select your project, check "Force updates of Snapshots/Releases", and click the OK button. Maven may take a little to reconfigure your project; please, wait until the re-configuration's finished.

Change your `web.xml` file

```
<display-name>Sample</display-name>

<servlet>
    <servlet-name>SampleServlet</servlet-name>
    …
</servlet>

<servlet-mapping>
    <servlet-name>SampleServlet</servlet-name>
    …
</servlet-mapping>
```


Locate file “web.xml” file, and change the elements in this slide to reflect the features of your project. “display-name” refers to the name of your project within the application server, and “servlet-name” refers to the name of your servlet. We’ll explain what an application server and a servlet is in the next session; so far, it’s enough to know that we recommend that the servlet name should be the name of your project with suffix “Servlet”.

Change your `persistence.xml` file

```
<persistence-unit name="Sample">

<property
    name="javax.persistence.jdbc.url"
    value="jdbc:mysql://localhost:3306/Sample" />
```

UNIVERSIDAD Ð SEVILLA

113

In the “persistence.xml” file, there are two elements that you must change to adapt them to your new project. First, you must change the name of your persistence unit to the name of your project; then seek for property “javax.persistence.jdbc.url” and change it accordingly. We strongly recommend that each project should have a companion database with the same name.

Change DatabaseConfig.java

```
public final String PersistenceUnit = "Sample";

public final String entitySpecificationFilename =
               "./src/main/resources/PopulateDatabase.xml";
public final String entityMapFilename =
               "./src/main/resources/Entities.map";
```


"DatabaseConfig.java" is a configuration file that we need to populate and query the database of your project. You have to change this line to change the name of your persistence unit, which must be exactly the same name that you used in file "persistence.xml". This configuration file also allows to change the name of the default XML file in which the entities used to populate the database are specified ("PopulateDatabase.xml" by default) and the name of a map that associates their bean names with their identifiers in the database ("Entities.map" by default). It's unlikely that you need to modify those filenames, so keep them to their default values. We'll learn more about them in forthcoming lessons.

Change your `data.xml` file

```
<bean  id="dataSource"
       class="com.mchange.v2.c3p0.ComboPooledDataSource"
       destroy-method="close">
   <property name="driverClass"
       value="com.mysql.jdbc.Driver" />
   <property name="jdbcUrl"
       value="jdbc:mysql://localhost:3306/Sample" />
   <property name="user" value="acme-user" />
   <property name="password" value="ACME-Us3r-P@ssw0rd" />
</bean>
…
<bean id="persistenceUnit" class="java.lang.String">
    <constructor-arg value="Sample" />
</bean>
```


The “data.xml” file provides additional information about the database of your project. You must first configure a datasource, which is an object that manages several concurrent connections to the database, and then reference the persistence unit defined in “persistence.xml”.

Change your `footer.jsp` file

```
<b>Copyright &copy;
    <fmt:formatDate value="${date}" pattern="yyyy" />
    Sample Co., Inc.</b>
```


This file contains the footer of your application. By default it simply shows a copyright message. You should adapt it to your project.

Change your header.jsp file (I)

```
<div>
    <img src="images/logo.png" alt="Sample Co., Inc." />
</div>
```


This file contains the header of your application, which basically consists of a logo and a menu. Change the alternate text that is displayed on the logo to adapt it to your new project, change file “logo.png”, and “favicon.ico” accordingly.

Change your header.jsp file (II)

```
<ul id="jMenu">
    <li><a class="fNiv">Option 1</a>
        <ul>
            <li class="arrow"></li>
            <li><a href="…">Option 1.1</a></li>
            <li><a href="…">Option 1.2</a></li>
        </ul>
    </li>
    <li><a class="fNiv">Option 2</a>
        <ul>
            <li class="arrow"></li>
            <li><a href="…">Option 2.1</a></li>
            <li><a href="…">Option 2.2</a></li>
        </ul>
    </li>
    …
</ul>
```


Regarding the menu, this slide shows its structure. Please, analyse it and the template we provide so that you can adapt the menu to your new project.

Change your header.jsp file (III)

```
<security:authorize access="hasRole('ADMIN')">
    <li><a class="fNiv">Administrator's options</a>
        <ul>
            <li class="arrow"></li>
            <li><a href="…">Administrator's Action 1</a></li>
            <li><a href="…">Administrator's Action 2</a></li>
        </ul>
    </li>
</security:authorize>

<security:authorize access="hasRole('CUSTOMER')">
    <li><a class="fNiv">Customer's options</a>
        <ul>
            <li class="arrow"></li>
            <li><a href="…">Customer's Action 1</a></li>
            <li><a href="…">Customer's Action 2</a></li>
        </ul>
    </li>
</security:authorize>
```

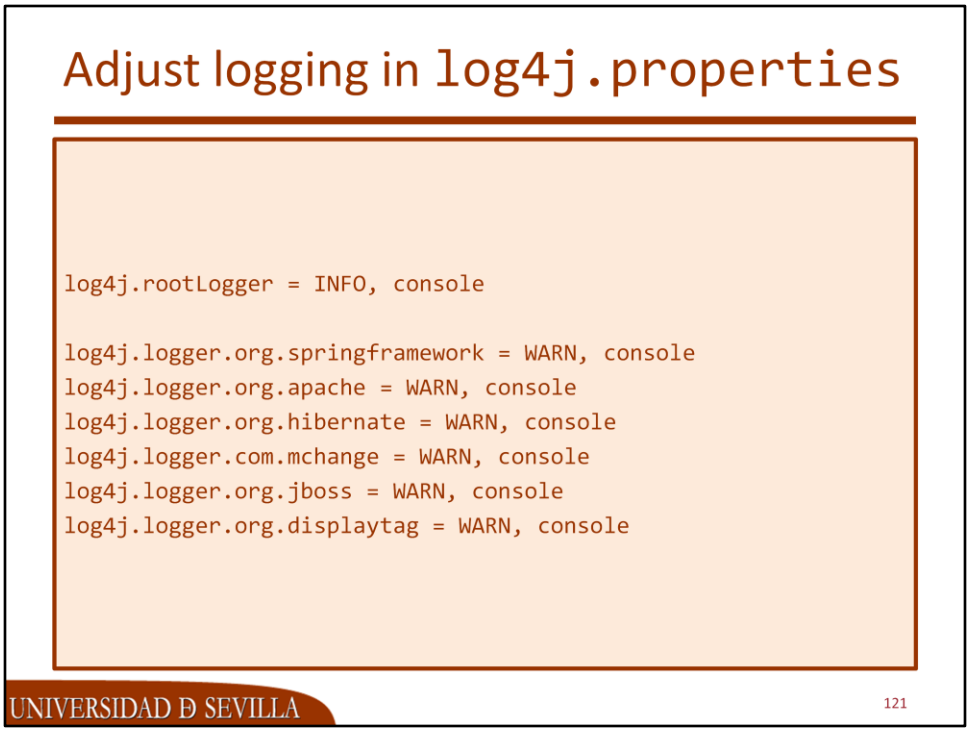

This slide shows how to restrict the access to some sections of the menu to users who have a given role/authority. Please, study the template that we provide, but bear in mind that this is just one side of the coin; later, we'll learn how to actually prevent an unauthorised user from typing in a URL and having access to unauthorised documents. Note that restricting an option from appearing in a menu does not prevent the user from typing in the corresponding URL in the address bar.

Change your header.jsp file (IV)

```
<li>
  <a href="…">
    <spring:message code="master.page.administrator.action.1" />
  </a>
</li>

<li>
  <a href="…">
    <spring:message code="master.page.administrator.action.2" />
  </a>
</li>
```


Take a look at how i18n & l10n works. Note that we never write a message in our ".jsp" files (the views). We always use instructions of the form <spring:message code="..." />. At runtime, these instructions result in messages that are translated into Spanish or English. Take a look at files "messages.properties" and "messages_es.properties" to learn more.

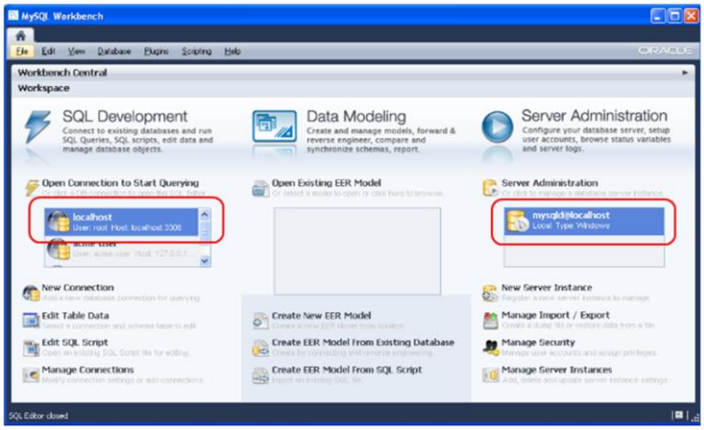


There's a final file you might have to change: "log4j.properties". This file controls the level of logging of your system. By default, only general informational messages and warning messages from our components are logged to the console. This is great as long as you don't get in trouble; if you have problems, you might try to adjust the logging level so that the console shows more messages regarding what's going on behind the scenes. For instance, if you suspect that the problem might be with the Spring Framework, you might try to change key "log4j.logger.org.springframework" to "INFO, console" or even "debug, console"; if you suspect that it's a problem with Apache, then change key "log4j.logger.org.apache" accordingly. In general, we recommend that you should first review our guidelines very carefully whenever you get in trouble; if you can't find an answer there, please resort to the Web; there's usually a lot of information available on almost every problem you may find.

NOTE: there are a few more configuration lines in this file. You should not change them unless you're a maven regarding the Log4J component.

Let's now analyse how to create the database that is associated with every project.

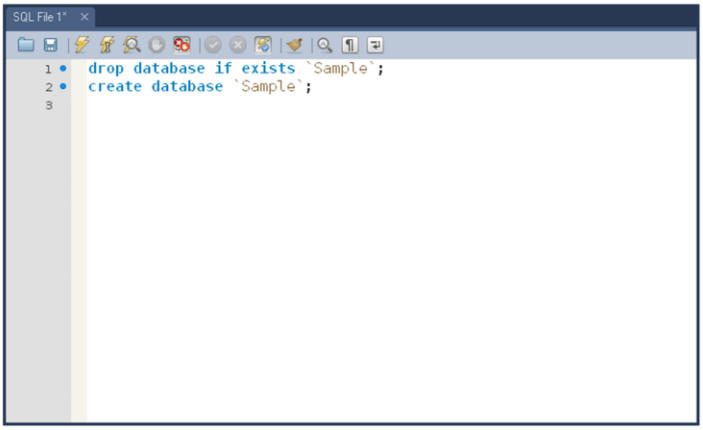
Open a root shell



UNIVERSIDAD D SEVILLA

First, open a root shell in MySQL Workbench. If you're using our virtual machine, just launch MySQL Workbench, click on "mysqld@localhost" to start the server, and then on the preconfigured root connection to open a root shell. In MySQL's parlance, a root's a database administrator.

Create the database

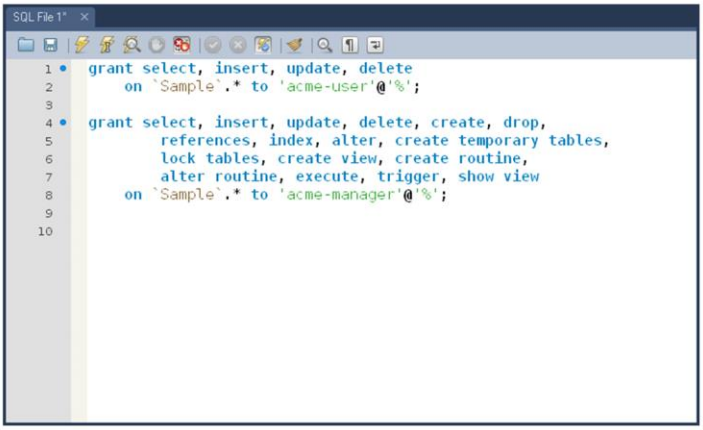


UNIVERSIDAD D SEVILLA

To create the database, you just have to execute the following script:

```
drop database if exists `Sample`;
create database `Sample`;
```

Grant privileges



UNIVERSIDAD D SEVILLA

Creating the database is not enough; we also have to grant privileges to users “acme-user” and “acme-manager”. “acme-user” is a restricted user that our web information systems will use to select, insert, update, or delete data from our databases. It should then be granted exactly these privileges on database “Sample”, which can be accomplished by means of the following script:

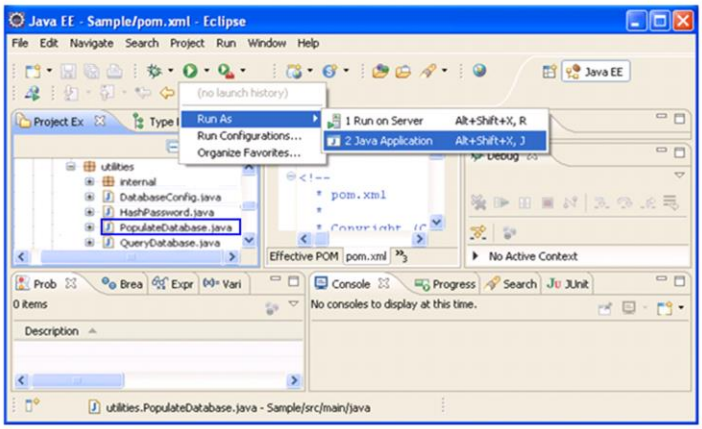
```
grant select, insert, update, delete
    on `Sample`.* to 'acme-user'@'%';
```

“acme-manager” is a user that can manage the “Sample” database; that is, it can drop it, re-create it, define views, lock tables, and the like. This user will obviously be used in a limited number of cases. To grant management privileges to “acme-manager”, you have to execute the following script:

```
grant select, insert, update, delete, create, drop, references, index, alter,
    create temporary tables, lock tables, create view, create routine,
    alter routine, execute, trigger, show view
    on `Sample`.* to 'acme-manager'@'%';
```

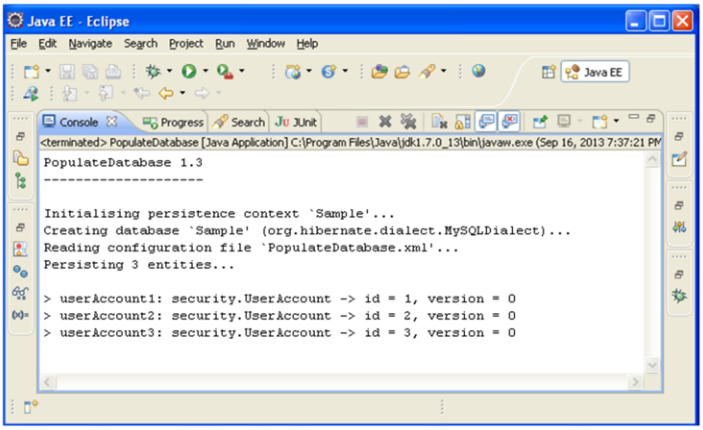
Pretty easy, isn't it? Yes it is. But before concluding, we'd like to warn you. Please, note that you need to have access to your MySQL server as an administrator in order to create a database and grant privileges to your users. After that, there are very few cases in which you need to have access to the administrator's account.

Just one more step: populating



There's one more step before concluding this section: the database you've just created is empty. We have to populate it with some data. To do so, search for a file called "PopulateDatabase.java", which should be in the following folder: "Java Resources/src/main/java/utilities". Right-click it and select "Run as > Java Application".

Check the results



After a little activity, your Console view should look more or less like this. The information on the screen shows that the database's been populated with three sample user accounts. By default, the database will be populated with information about three user accounts: "super"/"super", "admin"/"admin", and "customer"/"customer". We'll provide additional details later.

Great, we're almost done! It's time to try our sample project.

Are you nervous?

UNIVERSIDAD Ð SEVILLA

Are you nervous?

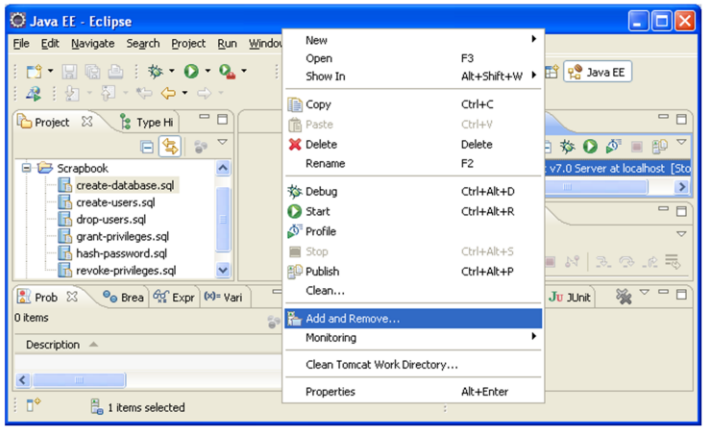
We too!

UNIVERSIDAD Ð SEVILLA

131

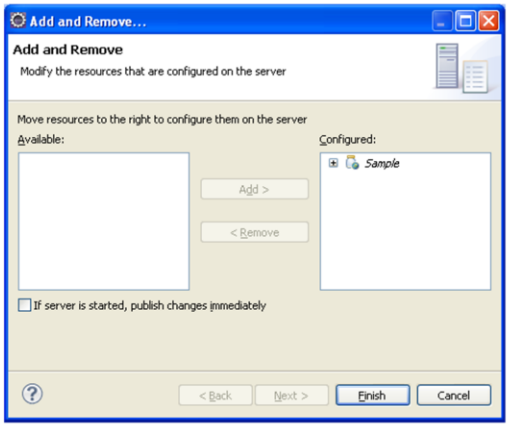
We too! We hope you've followed every guideline in this lecture so that you can start up the "Sample" project and get it running.

Let's deploy the project (I)



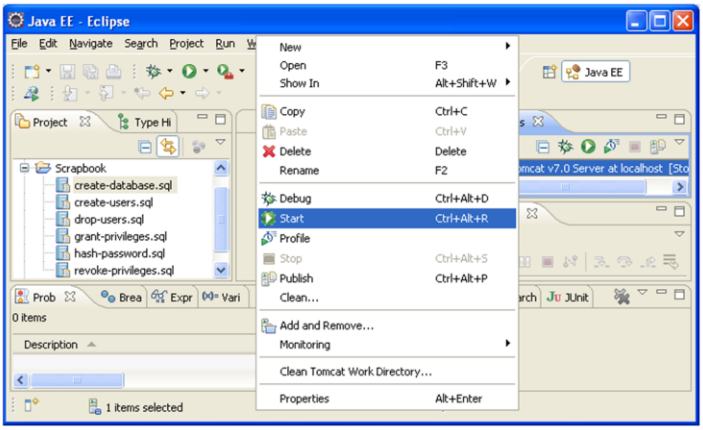
First of all, let's deploy your project to Tomcat. This is very simple: right-click "Tomcat v7.0 Server" in the Server view, and select "Add and Remove…".

Let's deploy the project (II)



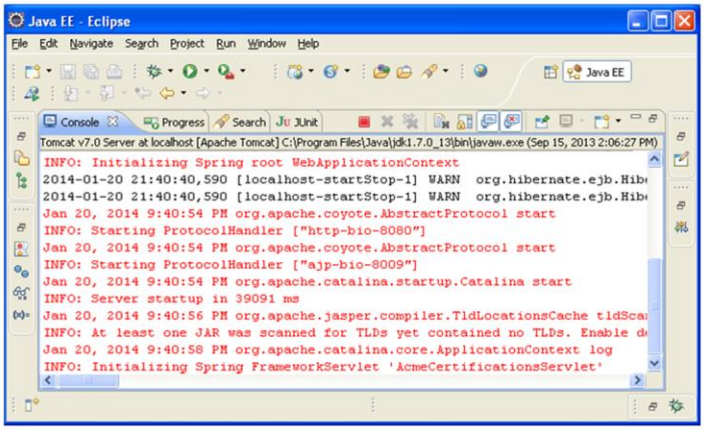
In this dialog box, move the "Sample" project from the "Available" column to the "Configured" column. Then, click the "Finish" button.

Let's start Tomcat



Once the project's been deployed, you have to start Tomcat. It's as easy as right-clicking the "Tomcat v7.0 Server" in the Servers view, and then selecting "Start".

Check the console



Starting Tomcat may take up to a minute depending on how powerful your computer is. Switch to the Console view to see what's happening. You'll see a lot of red messages, but don't worry, they are not error messages, just info messages. You should pay attention to the messages that are logged to the console. If everything's OK (everything should be OK if you've followed the guidelines in this lecture very carefully), then the following message should appear on the screen "INFO: Server startup in XXX seconds". That means that Tomcat is up and running, and that it's willing to accept requests to your web information system.

NOTE: unfortunately, there seems to be a problem with Spring Data 1.4.3: it displays the following warning every time your code has access to the database:

> WARN org.hibernate.ejb.HibernatePersistence - HHH015016: Encountered a deprecated javax.persistence.spi.PersistenceProvider [org.hibernate.ejb.HibernatePersistence]; use [org.hibernate.jpa.HibernatePersistenceProvider] instead.

Don't worry about this warning. Spring uses the correct class internally, but shows this warning mistakenly.

Open your browser and enjoy

Open Chrome and make it for “http://localhost:8080/Sample”. After a little activity in your Tomcat server, Chrome must display this screen. You can login using credentials “super”/”super”, “admin”/”admin”, or “customer”/”customer”. Please, give a try to the application and get familiar with it.

Is this a picture of you?

UNIVERSIDAD Ð SEVILLA

137

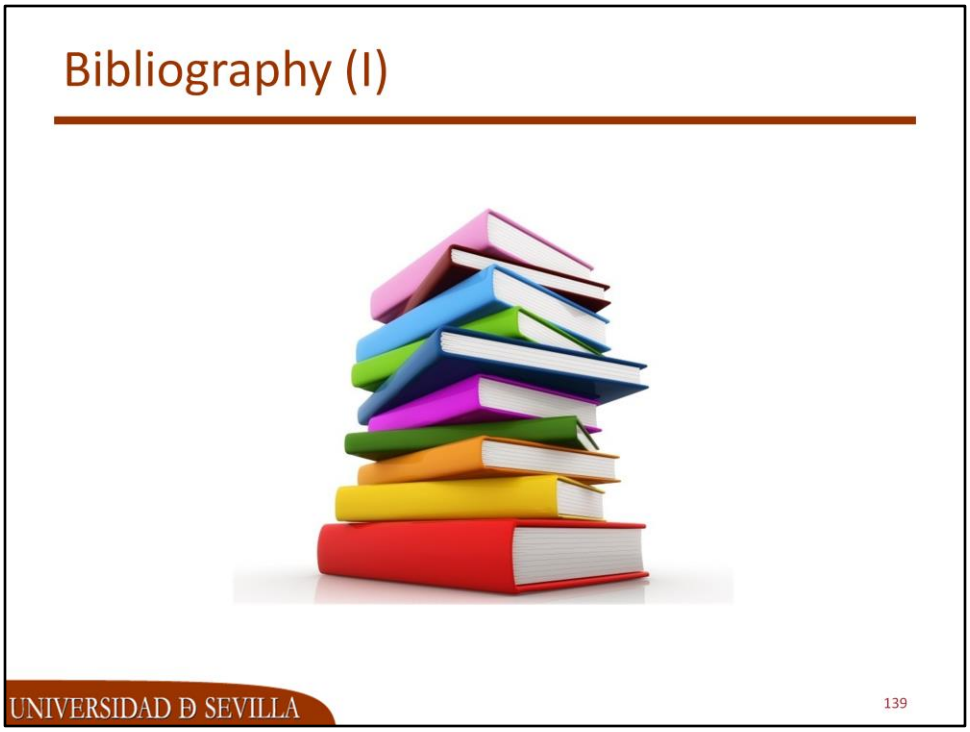
We sincerely expect that this is a picture of you. We hope you've followed every step and that the sample project's up and running now.

Or is this?

UNIVERSIDAD Ð SEVILLA

138

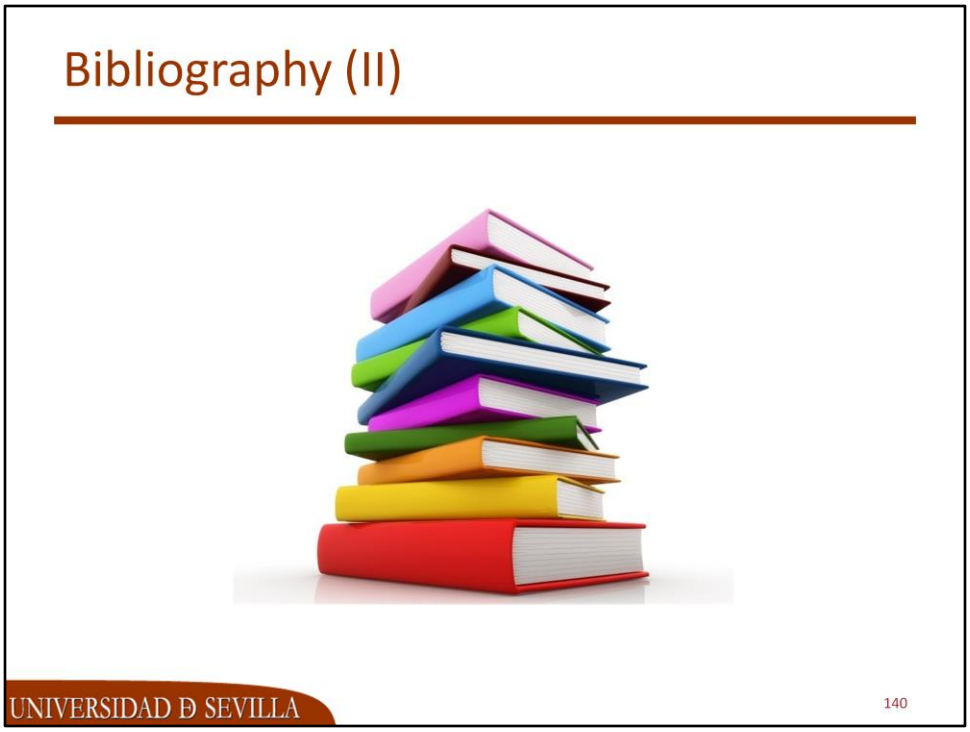
If not, please, get back, find what you didn't do the right way, and repeat it several times until your get familiar with the process. It's very important that you command the procedure to instantiate a workspace and a project template the sooner as possible. Please, bear in mind, that we, the lecturers, cannot help you with your computer or your project; we guarantee that the guidelines we provide work well; it's your duty to follow them very carefully.



Should you need more information on this lesson, please take a look at the following bibliography:

Pro Spring MVC
Marten Deinum, Koen Serneels, Colin Yates, Seth Ladd, Christophe Vanfletere
Apress, 2012

This bibliography is available in electronic format for our students at the USE's virtual library. If you don't know how to have access to the USE's virtual library, please, ask our librarians for help.



Should you need more information on the tools we're going to use, please, take a look at the following bibliography:

- ProjETSII: https://projetsii.informatica.us.es > Help
- Java 1.7: http://docs.oracle.com/javase/7/docs/
- Eclipse: http://help.eclipse.org/indigo/index.jsp
- Maven: http://maven.apache.org/guides/getting-started/index.html
- Subversive: http://www.polarion.com/products/svn/subversive/documentation.php
- Astah: http://astah.net/resources/documents/astah-basic-operation.pdf
- Balsamiq: https://support.balsamiq.com/tutorials/
- E. Pencil: http://pencil.evolus.vn/wiki/devguide/Introduction.html
- MySQL: http://dev.mysql.com/doc/
- Tomcat: http://tomcat.apache.org/tomcat-7.0-doc/index.html

Thanks for reading this materials!