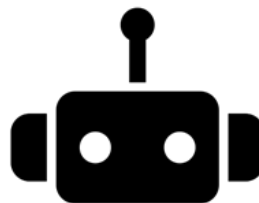


SESIÓN DE LABORATORIO

Teoria de Aplicaciones Web



HORARIO 10M1

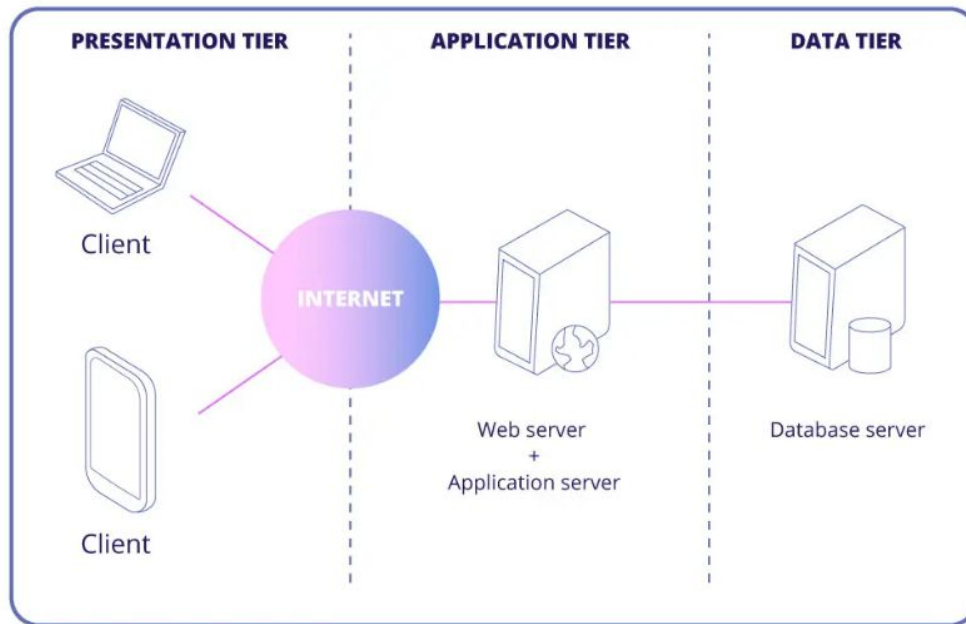
Bienvenidos !

Empezaremos a las 6:10 p.m

Gracias! c:



Teoría de aplicaciones web



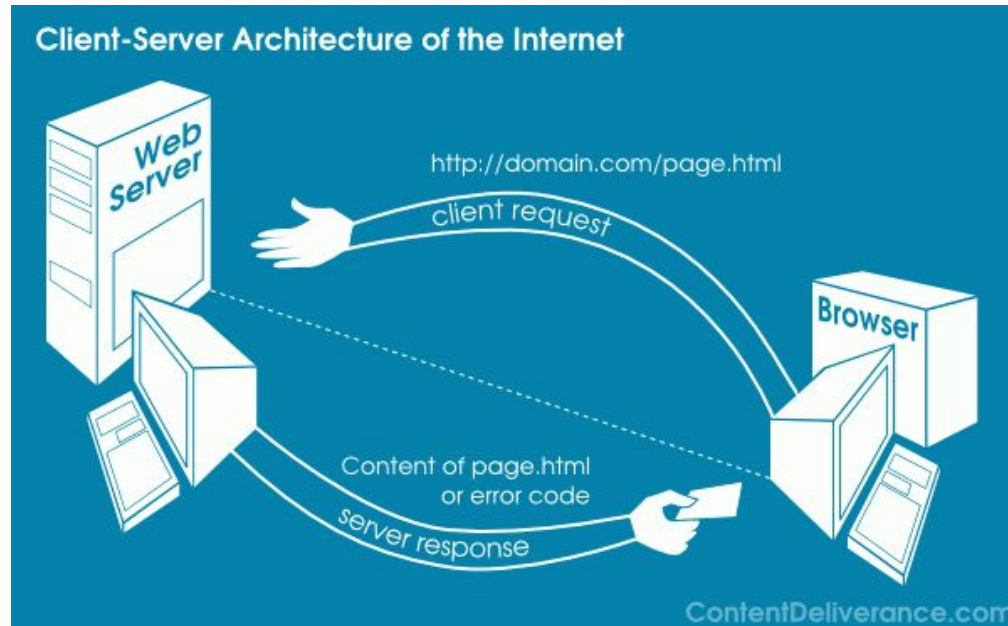
Una aplicación web , es un software que correr en un servidor web, donde la data necesaria es procesada antes de ser mostrada

Ejemplos

- Google
- Correo electrónico
- Intranet Pucp
- Redes sociales
- Wikipedia
- etc...

Teoría de aplicaciones web

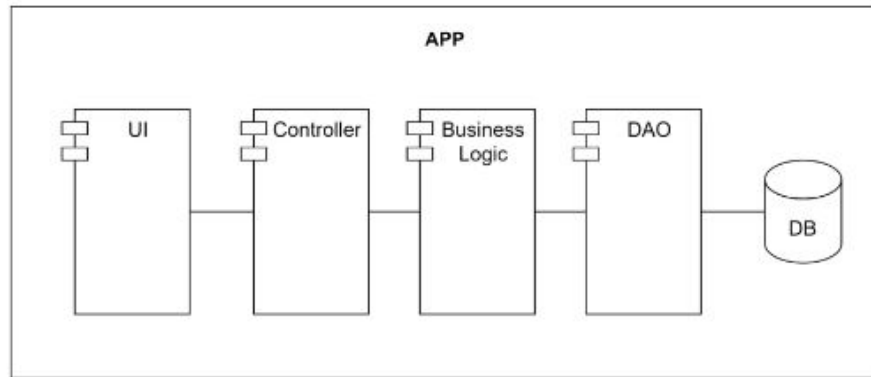
Arquitectura Cliente - Servidor



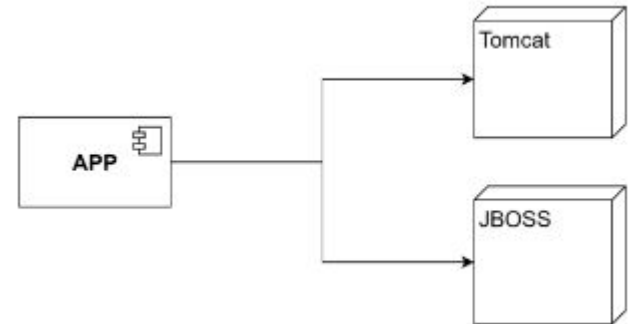
Teoría de aplicaciones web

Arquitectura monolítica

Aplicación :



Desarrollo :



Desarrollo de nuestra aplicación

Sistema de ingesta de productos

Funcionalidades :

Listado de productos



Creación de productos

Modificación de productos

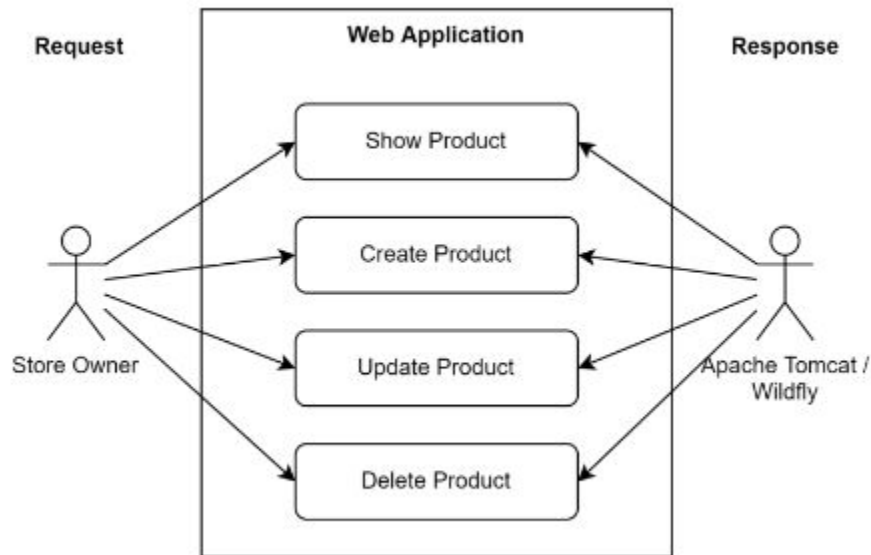
Eliminación de productos

Productos

Nuevo +

ID	Código	Descripción	Precio de compra	Precio de venta	Existencia	Editar	Eliminar
1	1	Galletas chokis	10.00	15.00	2.00		
2	2	Mermelada de fresa	65.00	80.00	97.00		
3	3	Aceite	18.00	20.00	100.00		
4	4	Palomitas de maíz	12.00	15.00	98.00		
5	5	Doritos	5.00	8.00	99.00		

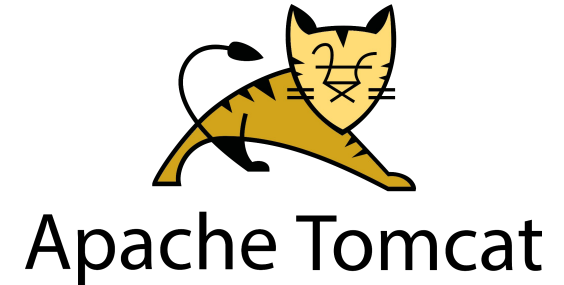
Desarrollo de nuestra aplicación CRUD



Desarrollo de nuestra aplicación

Requisitos

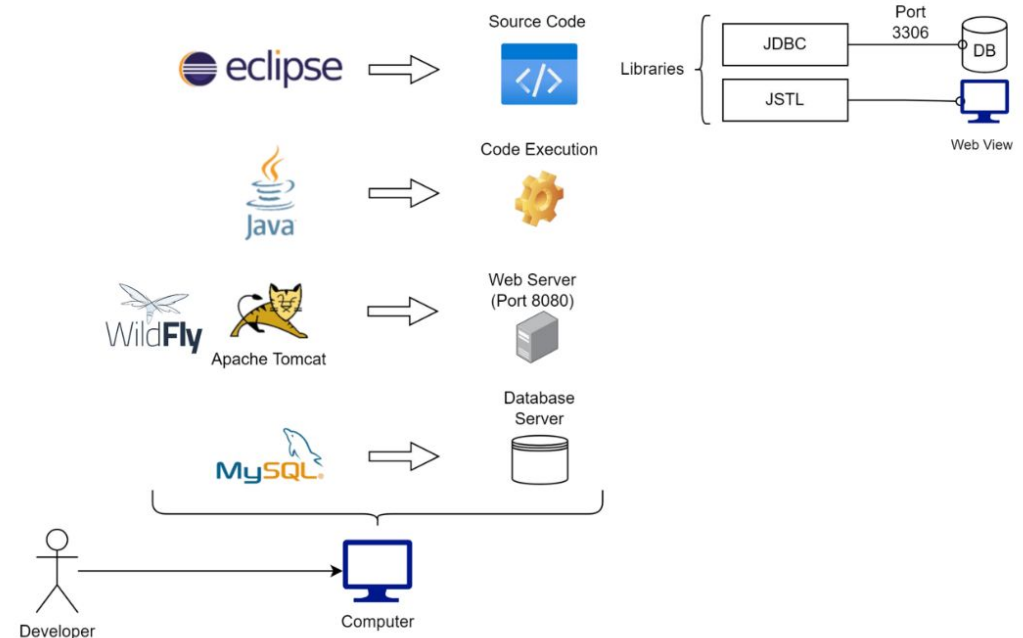
- Java JDK 8
- Eclipse IDE
- Apache Tomcat 9
- MySQL Server
- MySQL Connector for Java 8



Desarrollo de nuestra aplicación

Requisitos

- Java JDK 8
- Eclipse IDE
- Apache Tomcat 9
- WildFly (JBoss Community Version)
- MySQL Server
- MySQL Connector for Java 8



- Todo lo necesario puede ser descargado desde este link :

<https://drive.google.com/drive/folders/18stJmTYo3SKm4TszNuNUS4kvEioZvHd6>

Instalaciones

Java JDK 8 Instalación (1/4)

- (Nota: Para poder descargar desde Oracle es necesario tener una cuenta)
- Deben ir al siguiente url:
<https://www.oracle.com/java/technologies/downloads/archive/>
- Luego deben bajar hasta la sección de java SE 8 y seleccionar el instalador para windows que debe ser del tipo “jdk-8u371...” en el apartado de **Java SE Development Kit 8u371**

Java SE Development Kit 8u371

This software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#)

JDK 8u371 checksum

Product / File Description	File Size
Linux ARM64 RPM Package	72.14 MB
Linux ARM64 Compressed Archive	71.16 MB
Linux ARM32 Hard Float ABI	73.85 MB

Instalaciones


Java JDK 8 Instalación (2/4)

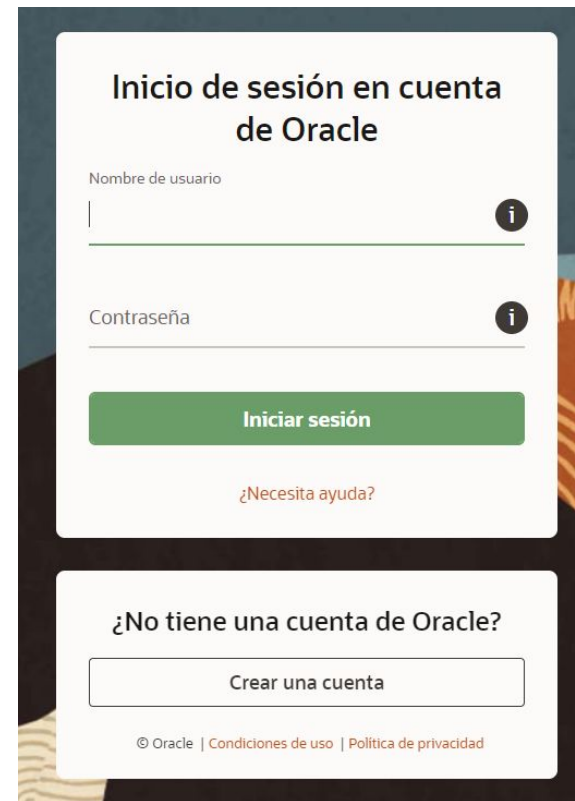
- Deben aceptar los términos de licencia
- Una vez se logeen , la descarga empezará

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

☒ I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE
Required

You will be redirected to the login screen in order to download the file.

Download `jdk-8u321-windows-x64.exe` 

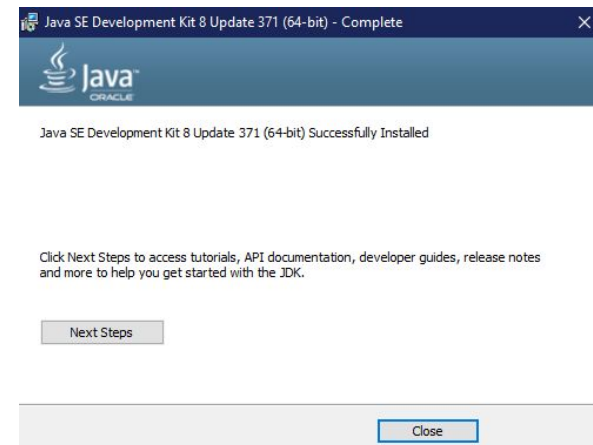
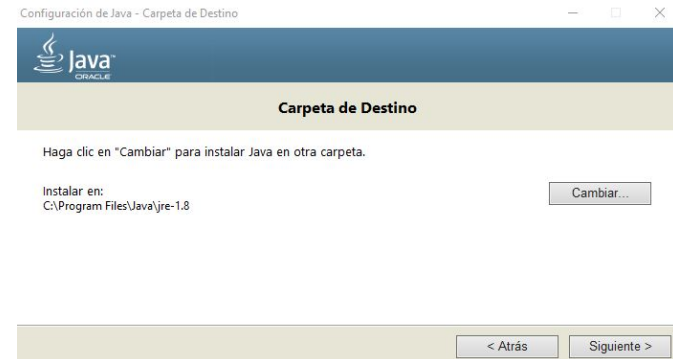
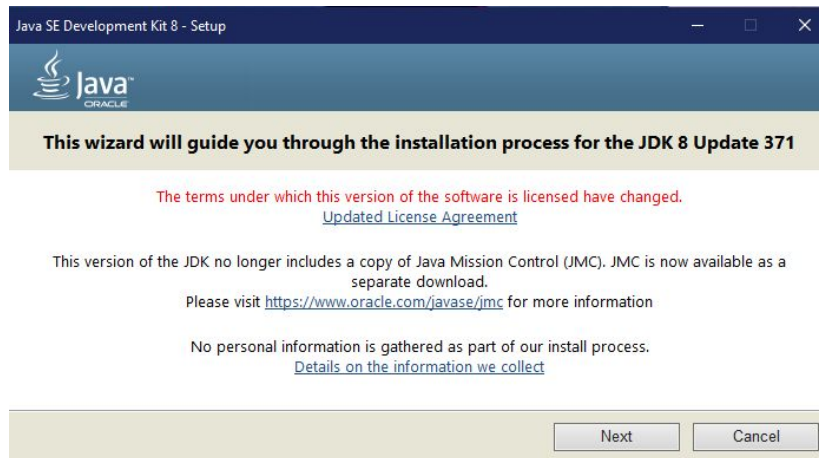


The screenshot shows the Oracle login interface. At the top, it says 'Inicio de sesión en cuenta de Oracle'. Below this are two input fields: 'Nombre de usuario' and 'Contraseña', each with an information icon (i) to its right. A green 'Iniciar sesión' button is positioned below the password field. Underneath the button is a link that says '¿Necesita ayuda?'. At the bottom of the login section is a box titled '¿No tiene una cuenta de Oracle?' containing a 'Crear una cuenta' button. The footer of the page includes the text '© Oracle | Condiciones de uso | Política de privacidad'.

Instalaciones

Java JDK 8 Instalación (3/4)

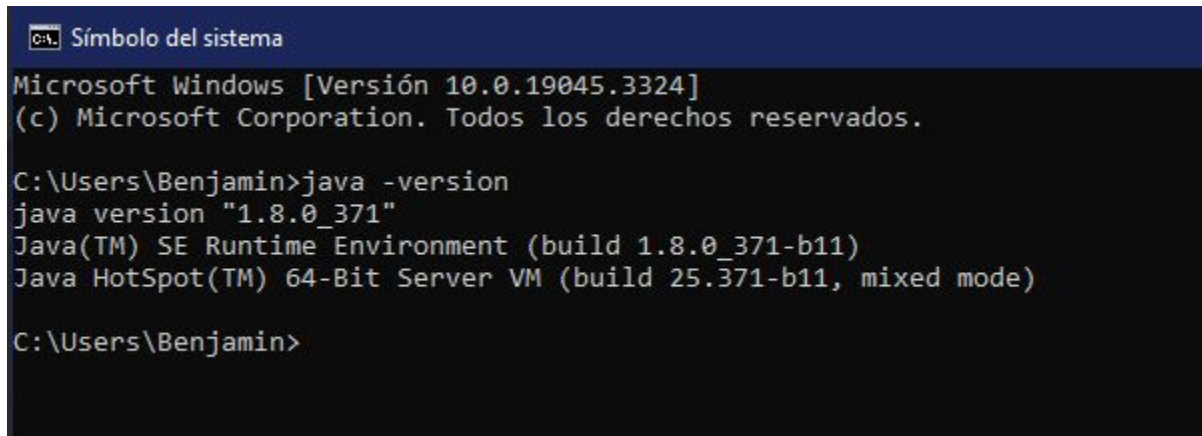
- Luego deben de ejecutar el instalador
- Deberán hacer click en siguiente en cada pantalla



Instalaciones

Java JDK 8 Instalación (4/4)

- Para poder verificar que todo se ha instalado correctamente
- Deben hacer Win + R → CMD → java -version



```
C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19045.3324]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Benjamin>java -version
java version "1.8.0_371"
Java(TM) SE Runtime Environment (build 1.8.0_371-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.371-b11, mixed mode)

C:\Users\Benjamin>
```

Instalaciones

Eclipse IDE Instalación (1/2)

Deben ir al siguiente url :

<https://www.eclipse.org/downloads/>

- Click “Download x86_64” -> Download

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

 **Download**

Download from: Canada - Rafal Rzczkowski (<https>)

File: [eclipse-inst-jre-win64.exe](#) SHA-512

[>> Select Another Mirror](#)

The Eclipse Installer 2023-06 R now includes a JRE for macOS, Windows and Linux.



Get **Eclipse IDE 2023-06**

Install your favorite desktop IDE packages.

Download x86_64

[Download Packages](#) | [Need Help?](#)

Instalaciones

Eclipse IDE Instalación (2/2)

Al abrir el instalador , deberán seleccionar Eclipse IDE for Enterprise Java and Web Developers y seleccionar en instalar

Es importante aclarar que el uso de Java 17 será cambiado luego por Java 8

**Eclipse IDE for Enterprise Java and Web Developers**[details](#)

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.

Java 17+ VM

JRE 17.0.8 - <https://download.eclipse.org/justj/jres/17/updates/release/latest>

Installation Folder

C:\Users\Benjamin\workspace\jee-2023-06

☒ create start menu entry

☒ create desktop shortcut



**Eclipse IDE for Enterprise Java and Web Developers**[details](#)

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.

Java 17+ VM

JRE 17.0.8 - <https://download.eclipse.org/justj/jres/17/updates/release/latest>

Installation Folder

C:\Users\Benjamin\workspace\jee-2023-06

☒ create start menu entry

☒ create desktop shortcut



[show readme file](#)

[open in system explorer](#)

[keep installer](#)

Instalaciones

Apache Tomcat Instalación (1/1)

Descargarlo desde :

<https://tomcat.apache.org/download-90.cgi>

utilizando la opción Zip

Extraerlo en un folder

9.0.79

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

Source Code Distributions

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

Instalaciones

JSTL Instalación (1/1)

Es una librería para manejar tags en programas java

Descargarlo desde :

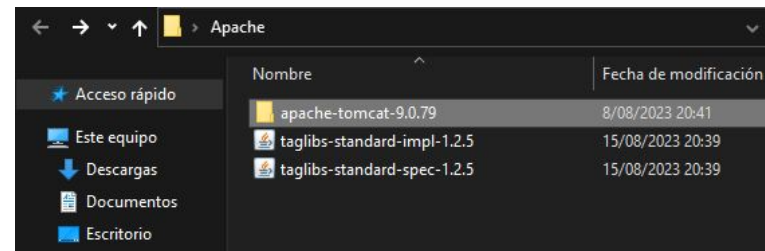
<https://tomcat.apache.org/download-taglibs.cgi>

Donde se deberán escoger las opciones que contienen Impl y Spec

Finalmente se deberán poner en el mismo folder que Apache

Jar Files

- [Binary README](#)
- Impl:
 - [taglibs-standard-impl-1.2.5.jar](#) (pgp, sha512)
- Spec:
 - [taglibs-standard-spec-1.2.5.jar](#) (pgp, sha512)
- EL:
 - [taglibs-standard-jstlel-1.2.5.jar](#) (pgp, sha512)
- Compat:
 - [taglibs-standard-compatible-1.2.5.jar](#) (pgp, sha512)



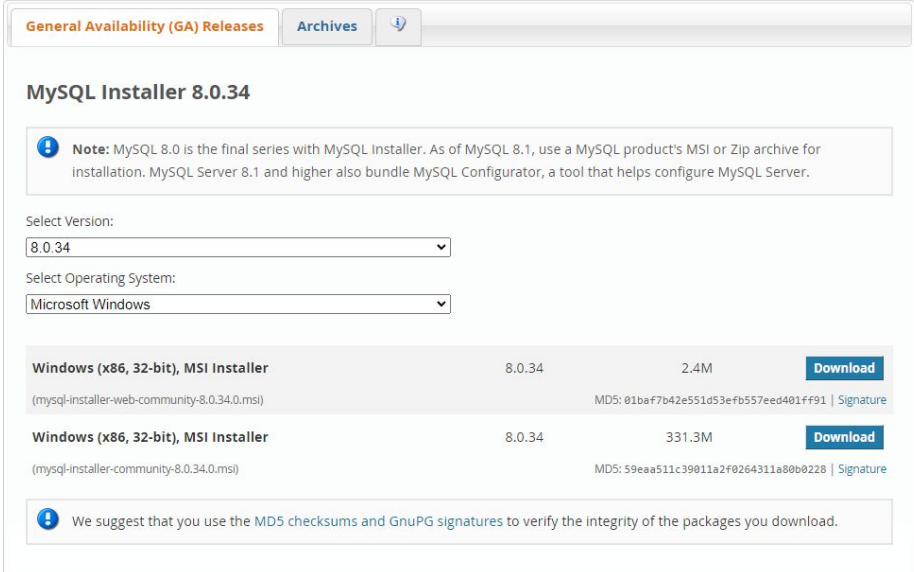
Instalaciones

MySQL Server Instalación (1/12)

Descargarlo desde :

<https://dev.mysql.com/downloads/windows/installer/8.0.html>

Donde se usará la versión 8.0.34 y la versión community



The screenshot shows the MySQL Installer 8.0.34 download page. It features a navigation bar with 'General Availability (GA) Releases', 'Archives', and a download icon. The main heading is 'MySQL Installer 8.0.34'. A note states: 'Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.' Below this, there are two dropdown menus: 'Select Version:' with '8.0.34' selected, and 'Select Operating System:' with 'Microsoft Windows' selected. A table lists two download options:

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.34.0.msi)	8.0.34	2.4M	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.34.0.msi)	8.0.34	331.3M	Download

Each row also includes an MD5 checksum and a link to the signature. At the bottom, a note suggests using MD5 checksums and GnuPG signatures to verify the integrity of the packages.

Instalaciones

MySQL Server Instalación (2/12)

Una vez se inicie la instalación
haremos click en Full

Luego en siguiente y en ejecutar y
esperaremos a que culmine la
instalación

Choosing a Setup Type

Please select the Setup Type that suits your use case.

☐ **Server only**

Installs only the MySQL Server product.

☐ **Client only**

Installs only the MySQL Client products, without a server.

☒ **Full**

Installs all included MySQL products and features.

☐ **Custom**


Manually select the products that should be installed on the system.

Setup Type Description

Installs all of the products available in this catalog including MySQL Server, MySQL Shell, MySQL Router, MySQL Workbench, documentation, samples and examples and more.

Installation

The following products will be installed.

Product	Arch	Status	Progress	Note
 MySQL Server 8.0.34	X64	Installing	60%	
 MySQL Workbench 8.0.34	X64	Ready to Install		
 MySQL Shell 8.0.34	X64	Ready to Install		
 MySQL Router 8.0.34	X64	Ready to Install		
 MySQL Documentation 8.0.34	X86	Ready to Install		
 Samples and Examples 8.0.34	X86	Ready to Install		

Show Details >

< Back

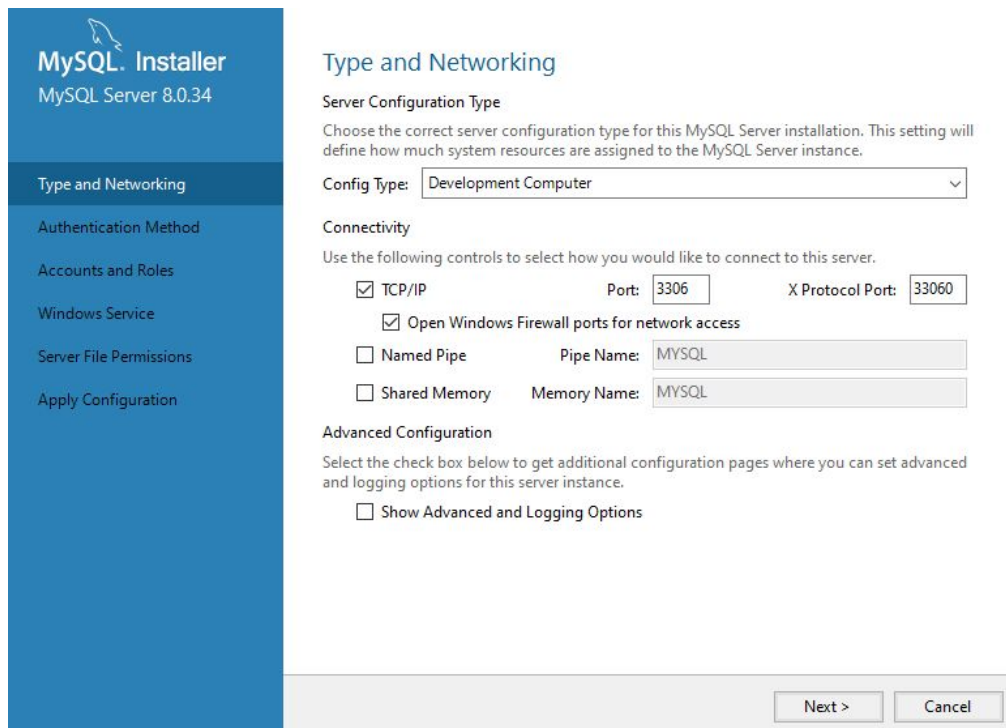
Execute

Cancel

Instalaciones

MySQL Server Instalación (3/12)

Una vez terminada la instalación en el apartado de Type and Networking debemos asegurarnos que el config Type está en Development Computer y apuntar los valores de Port : 3306 y X Protocol Port 33060



The screenshot shows the 'MySQL Installer' window for 'MySQL Server 8.0.34'. The left sidebar lists several configuration steps: 'Type and Networking' (selected), 'Authentication Method', 'Accounts and Roles', 'Windows Service', 'Server File Permissions', and 'Apply Configuration'. The main area is titled 'Type and Networking' and contains the following sections:

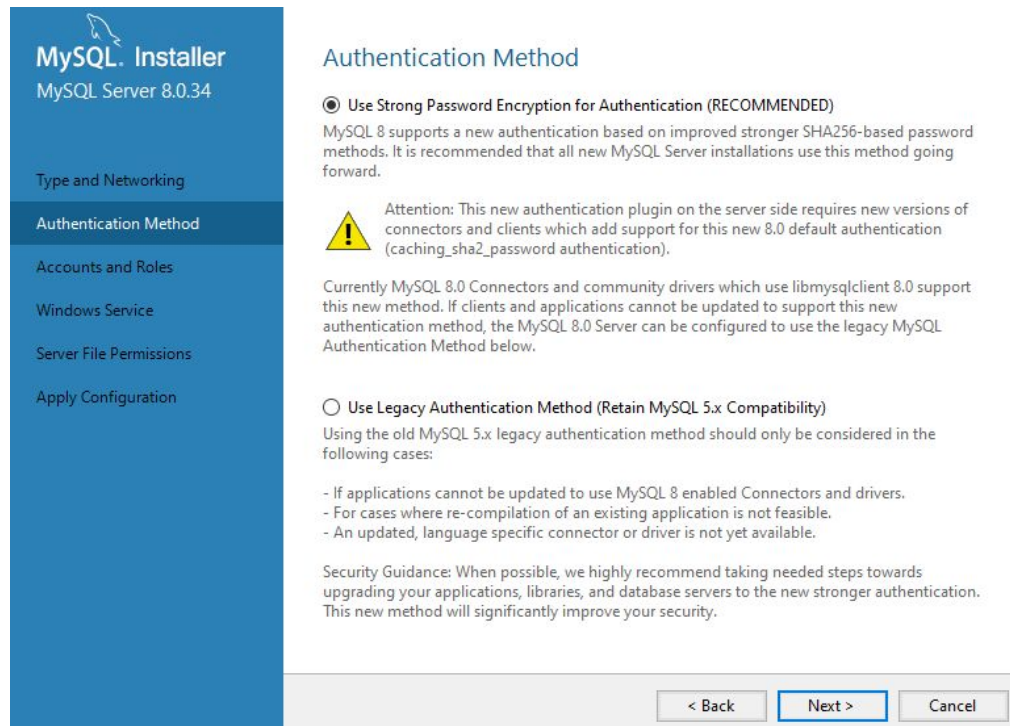
- Server Configuration Type**: A dropdown menu set to 'Development Computer'. Below it, a note states: 'Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.'
- Connectivity**: A section with the instruction 'Use the following controls to select how you would like to connect to this server.' It includes:
 - ☒ TCP/IP: Port: 3306, X Protocol Port: 33060
 - ☒ Open Windows Firewall ports for network access
 - ☐ Named Pipe: Pipe Name: MYSQL
 - ☐ Shared Memory: Memory Name: MYSQL
- Advanced Configuration**: A section with the instruction 'Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.' It includes:
 - ☐ Show Advanced and Logging Options

At the bottom right, there are 'Next >' and 'Cancel' buttons.

Instalaciones

MySQL Server Instalación (4/12)

En el siguiente apartado se usará la opción recomendada para encriptación.



MySQL. Installer
MySQL Server 8.0.34

Type and Networking

Authentication Method

Accounts and Roles

Windows Service


Server File Permissions

Apply Configuration

Authentication Method

☒ **Use Strong Password Encryption for Authentication (RECOMMENDED)**

MySQL 8 supports a new authentication based on improved stronger SHA256-based password methods. It is recommended that all new MySQL Server installations use this method going forward.

 **Attention:** This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (caching_sha2_password authentication).

Currently MySQL 8.0 Connectors and community drivers which use libmysqlclient 8.0 support this new method. If clients and applications cannot be updated to support this new authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below.

☐ **Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)**

Using the old MySQL 5.x legacy authentication method should only be considered in the following cases:

- If applications cannot be updated to use MySQL 8 enabled Connectors and drivers.
- For cases where re-compilation of an existing application is not feasible.
- An updated, language specific connector or driver is not yet available.

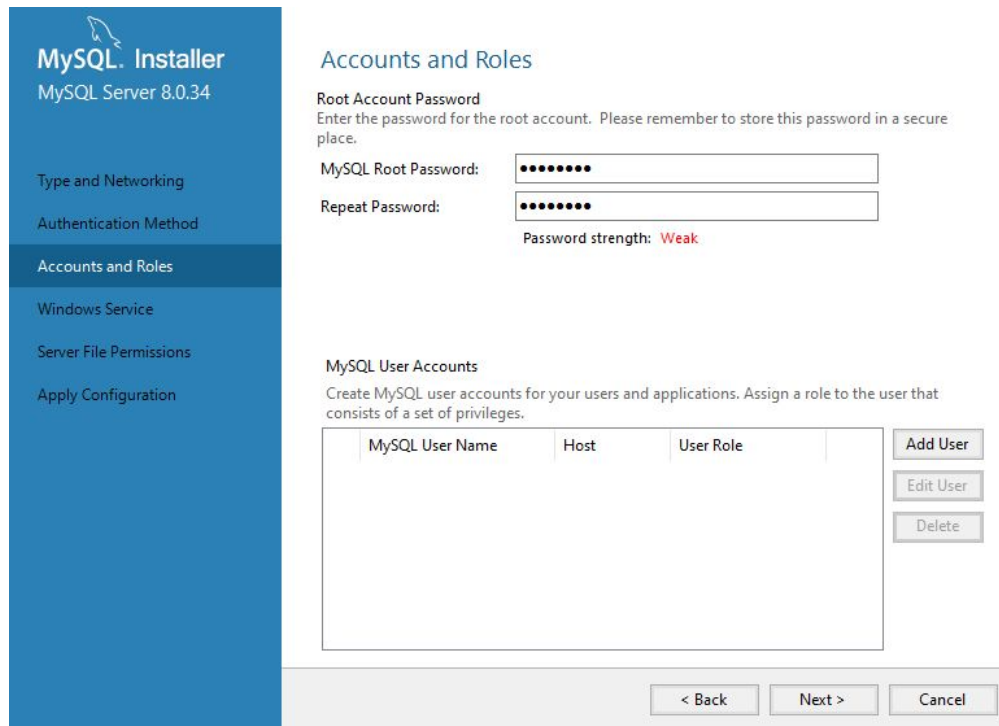
Security Guidance: When possible, we highly recommend taking needed steps towards upgrading your applications, libraries, and database servers to the new stronger authentication. This new method will significantly improve your security.

< Back **Next >** Cancel

Instalaciones

MySQL Server Instalación (5/12)

En el siguiente apartado debemos escoger una contraseña la cual debemos recordar dado que la usaremos para acceder a la base de datos



The screenshot shows the 'Accounts and Roles' step of the MySQL 8.0.34 installer. On the left is a blue sidebar with navigation options: 'Type and Networking', 'Authentication Method', 'Accounts and Roles' (selected), 'Windows Service', 'Server File Permissions', and 'Apply Configuration'. The main area is titled 'Accounts and Roles' and contains two sections. The first section, 'Root Account Password', prompts the user to enter a password for the root account, with a note to store it securely. It includes input fields for 'MySQL Root Password' and 'Repeat Password', and a 'Password strength' indicator showing 'Weak'. The second section, 'MySQL User Accounts', instructs the user to create MySQL user accounts and assign roles. It features a table with columns for 'MySQL User Name', 'Host', and 'User Role'. To the right of the table are buttons for 'Add User', 'Edit User', and 'Delete'. At the bottom of the window are '< Back', 'Next >', and 'Cancel' buttons.

MySQL. Installer
MySQL Server 8.0.34

Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Server File Permissions
Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

Password strength: **Weak**

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL User Name	Host	User Role
-----------------	------	-----------

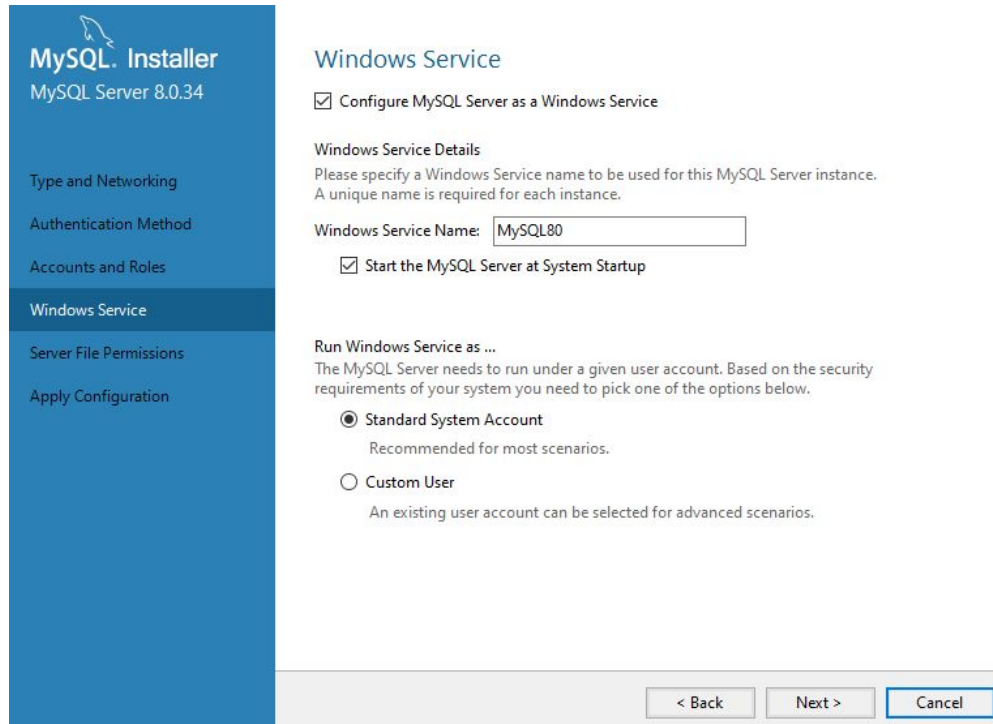
Add User
Edit User
Delete

< Back Next > Cancel

Instalaciones

MySQL Server Instalación (6/12)

Luego deberemos aceptar la configuración por default de MySQL como servicio de windows y aplicarlas



The screenshot shows the 'MySQL. Installer' window for 'MySQL Server 8.0.34'. The left sidebar contains the following steps: 'Type and Networking', 'Authentication Method', 'Accounts and Roles', 'Windows Service' (which is the active step), 'Server File Permissions', and 'Apply Configuration'. The main area is titled 'Windows Service' and contains the following options:

- ☒ **Configure MySQL Server as a Windows Service**

Windows Service Details
Please specify a Windows Service name to be used for this MySQL Server instance. A unique name is required for each instance.

Windows Service Name:

☒ **Start the MySQL Server at System Startup**

Run Windows Service as ...
The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.

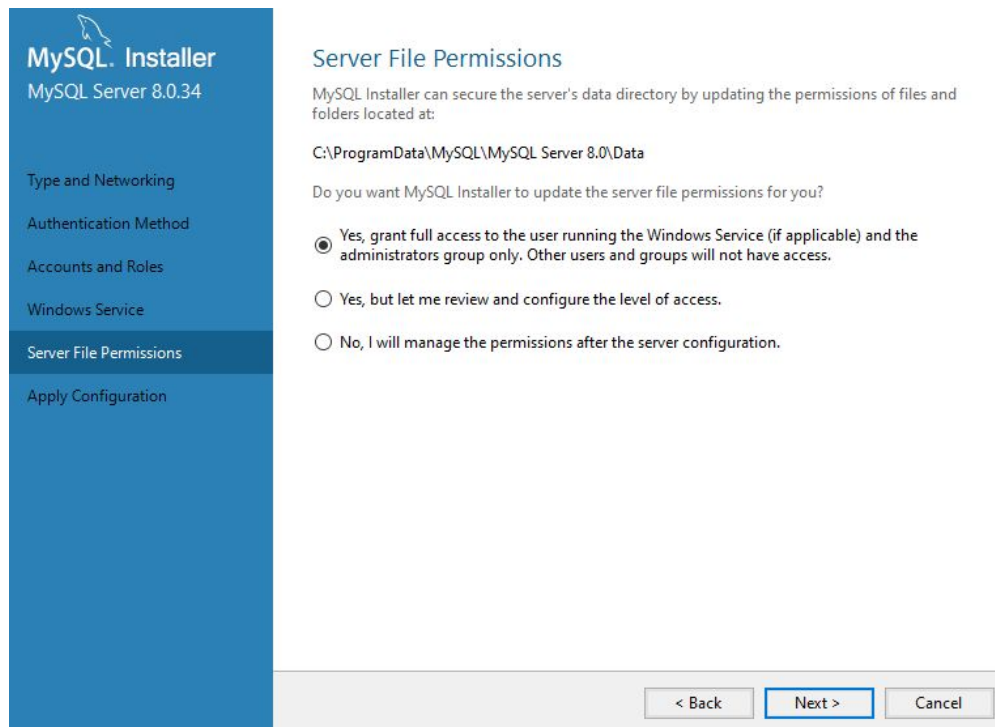
- ☒ **Standard System Account**
Recommended for most scenarios.
- ☐ **Custom User**
An existing user account can be selected for advanced scenarios.

At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Instalaciones

MySQL Server Instalación (7/12)

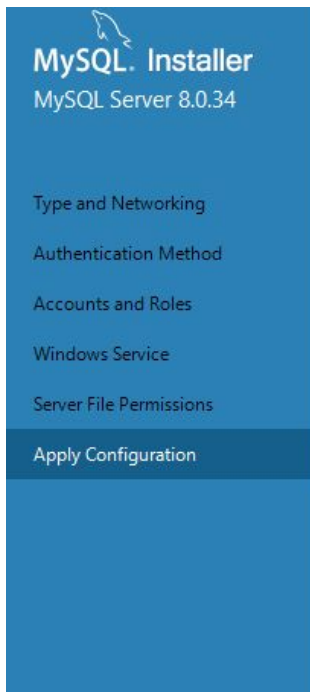
Luego debemos asignar permisos totales



Instalaciones

MySQL Server Instalación (8/12)

Finalmente se aplican las configuraciones dándole click en execute



Apply Configuration

Click [Execute] to apply the changes

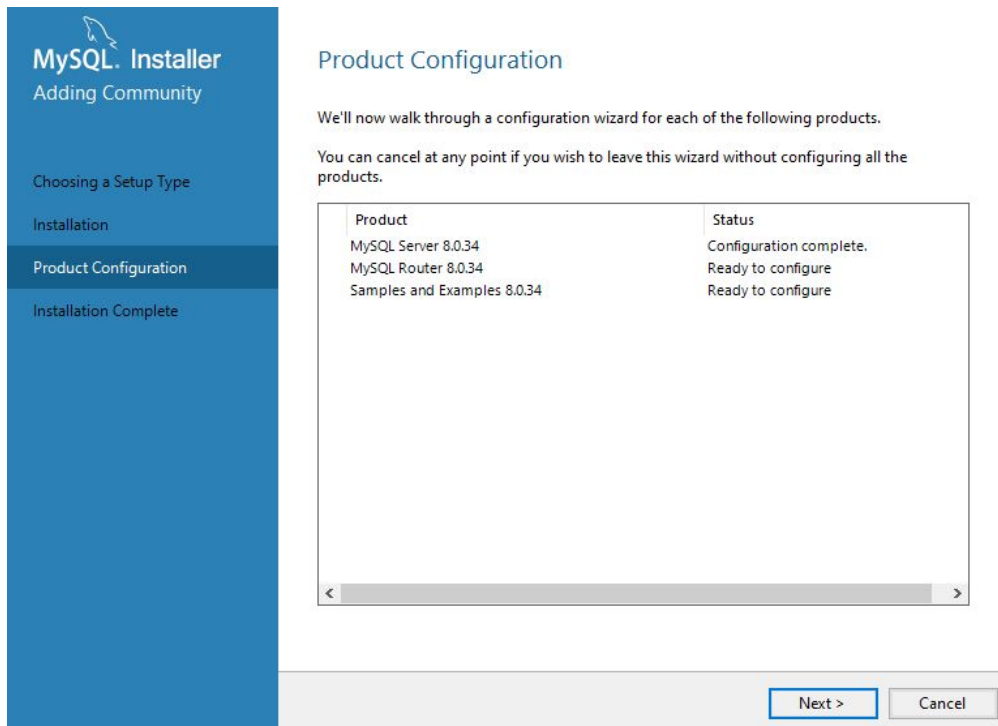
Configuration Steps Log

- ☐ Writing configuration file
- ☐ Updating Windows Firewall rules
- ☐ Adjusting Windows service
- ☐ Initializing database (may take a long time)
- ☐ Updating permissions for the data folder and related server files
- ☐ Starting the server
- ☐ Applying security settings
- ☐ Updating the Start menu link

Instalaciones

MySQL Server Instalación (9/12)

Continuando se deberá hacer la configuración de los otros productos



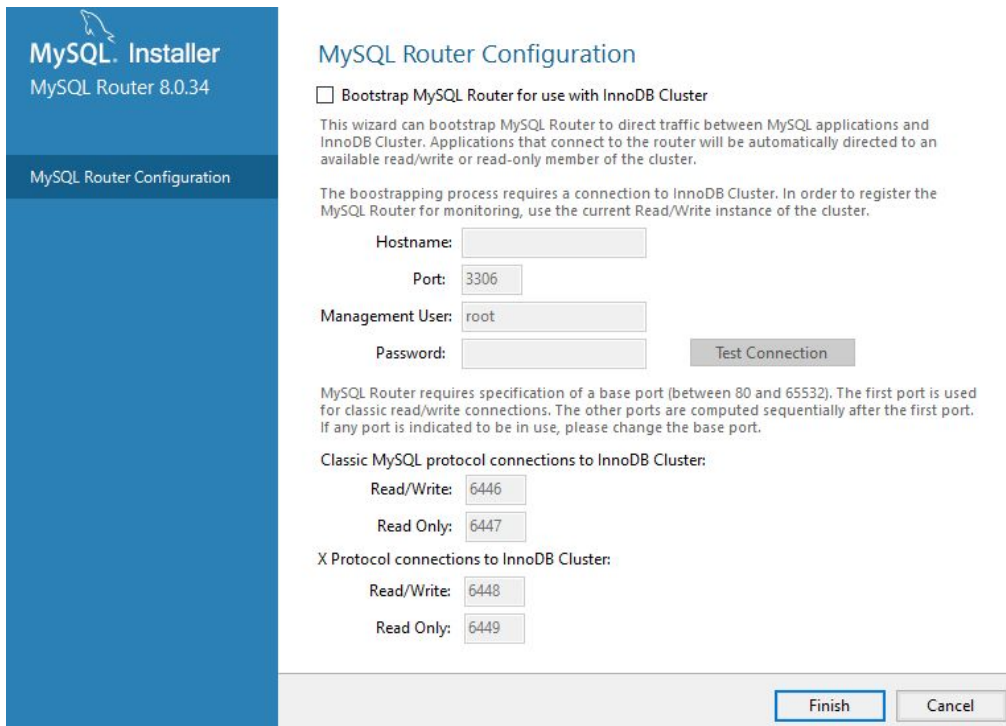
The image shows the MySQL Installer 'Product Configuration' window. On the left is a blue sidebar with the MySQL logo and the text 'MySQL. Installer Adding Community'. Below this are four menu items: 'Choosing a Setup Type', 'Installation', 'Product Configuration' (which is highlighted), and 'Installation Complete'. The main area on the right is titled 'Product Configuration' and contains the following text: 'We'll now walk through a configuration wizard for each of the following products.' and 'You can cancel at any point if you wish to leave this wizard without configuring all the products.' Below this text is a table with two columns: 'Product' and 'Status'. The table lists three items: 'MySQL Server 8.0.34' with status 'Configuration complete.', 'MySQL Router 8.0.34' with status 'Ready to configure', and 'Samples and Examples 8.0.34' with status 'Ready to configure'. At the bottom right of the window are two buttons: 'Next >' and 'Cancel'.

Product	Status
MySQL Server 8.0.34	Configuration complete.
MySQL Router 8.0.34	Ready to configure
Samples and Examples 8.0.34	Ready to configure

Instalaciones

MySQL Server Instalación (10/12)

Respecto a la configuración de Router , se debe mantener la configuración



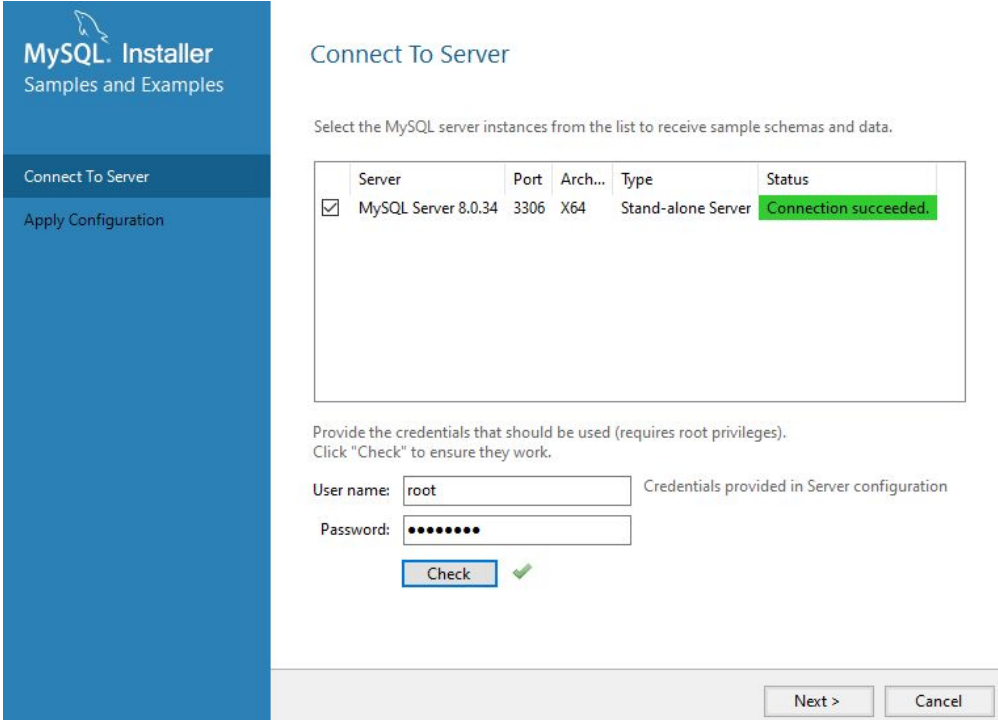
The screenshot shows the 'MySQL Router Configuration' window from the MySQL Installer. The window has a blue header with the MySQL logo and 'MySQL Router 8.0.34'. Below the header, the title 'MySQL Router Configuration' is displayed. The main content area is white and contains the following elements:

- ☐ Bootstrap MySQL Router for use with InnoDB Cluster
- A paragraph explaining that the wizard can bootstrap MySQL Router to direct traffic between MySQL applications and InnoDB Cluster, and that applications connecting to the router will be automatically directed to an available read/write or read-only member of the cluster.
- A paragraph stating that the bootstrapping process requires a connection to InnoDB Cluster and that the MySQL Router for monitoring should use the current Read/Write instance of the cluster.
- Fields for 'Hostname:', 'Port:' (set to 3306), 'Management User:' (set to root), and 'Password:'.
- A 'Test Connection' button.
- A paragraph explaining that MySQL Router requires specification of a base port (between 80 and 65532) and that the first port is used for classic read/write connections, while other ports are computed sequentially.
- A section titled 'Classic MySQL protocol connections to InnoDB Cluster:' with fields for 'Read/Write:' (6446) and 'Read Only:' (6447).
- A section titled 'X Protocol connections to InnoDB Cluster:' with fields for 'Read/Write:' (6448) and 'Read Only:' (6449).
- 'Finish' and 'Cancel' buttons at the bottom right.

Instalaciones

MySQL Server Instalación (11/12)

Finalmente se debe conectar al servidor y aplicar la configuración



The screenshot shows the MySQL Installer window with the 'Connect To Server' tab selected. It displays a table of server instances with one instance selected and a 'Check' button to verify credentials.

MySQL Installer
Samples and Examples

Connect To Server

Apply Configuration

Connect To Server


Select the MySQL server instances from the list to receive sample schemas and data.

	Server	Port	Arch...	Type	Status
<input checked="" type="checkbox"/>	MySQL Server 8.0.34	3306	X64	Stand-alone Server	Connection succeeded.

Provide the credentials that should be used (requires root privileges).
Click "Check" to ensure they work.

User name: Credentials provided in Server configuration

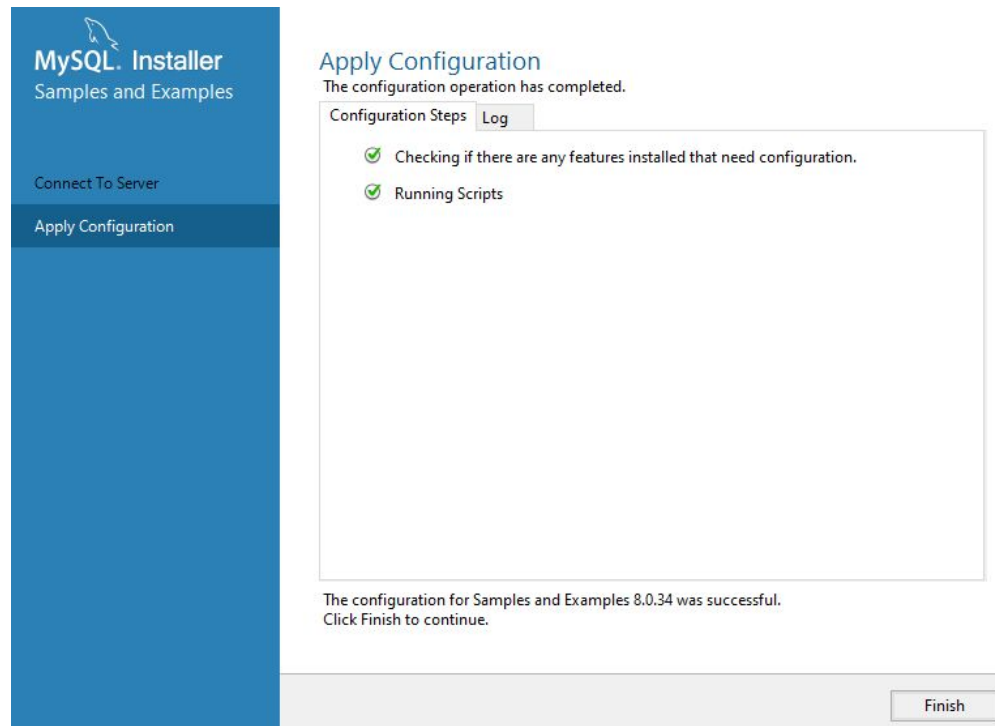
Password:



Instalaciones

MySQL Server Instalación (12/12)

Finalmente se debe completar la instalación



Instalaciones

MySQL Connector Download

Se debe descargar desde :

<https://dev.mysql.com/downloads/connector/j/>

Aquí se debe seleccionar
Plataforma Independiente y
descargar.

Luego se debe extraer el archivo
ZIP y mantener solo el archivo jar

Connector/J 8.1.0

Select Operating System:

Platform Independent

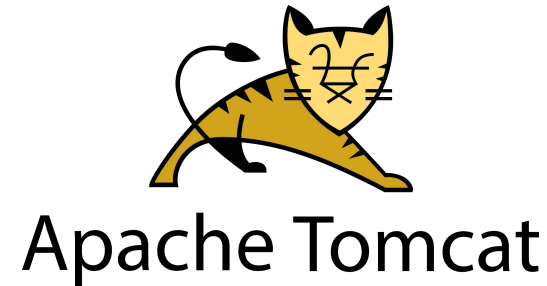
Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-j-8.1.0.tar.gz)	8.1.0	4.0M	Download
MD5: 4a95d62b0cfbad68b92ffc62ae7ee266 Signature			
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-j-8.1.0.zip)	8.1.0	4.8M	Download
MD5: d745362823ec4d37fa0607746d40a1b9 Signature			



We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

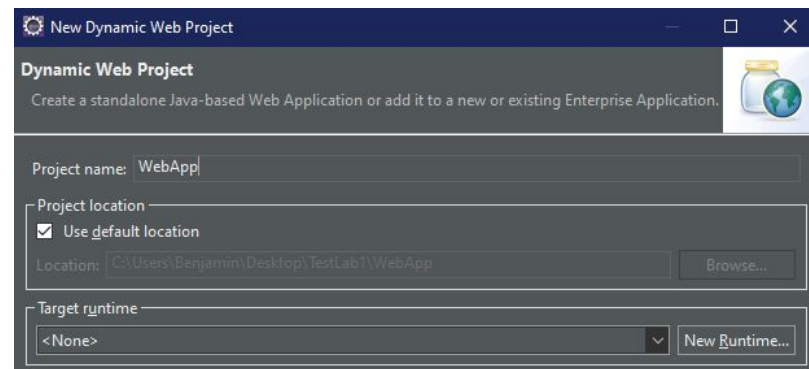
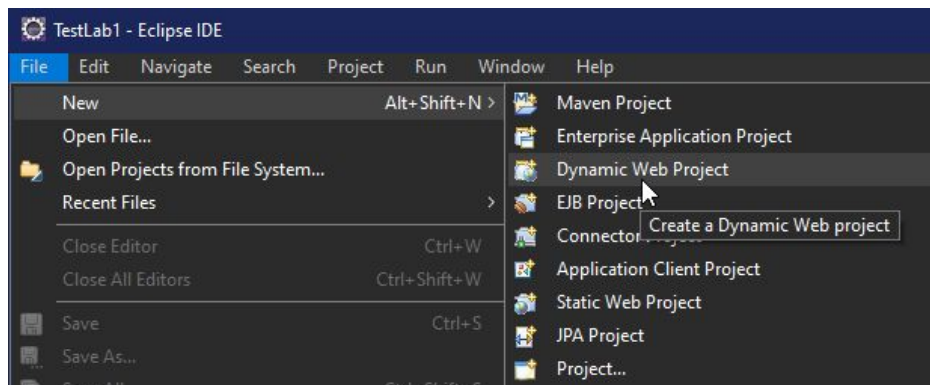
			Carpeta de archivos		
..			Carpeta de archivos	26/06/2023 12:10	
src	13,619,451	2,362,512	Carpeta de archivos	26/06/2023 12:10	
mysql-connector-j-8.1.0.jar	2,485,905	2,346,806	Executable Jar File	26/06/2023 12:10	403006D7
build.xml	96,999	15,524	Documento XML	26/06/2023 12:10	4E497814
CHANGES	280,612	94,069	Archivo	26/06/2023 12:10	594EE7B5
INFO_BIN	186	155	Archivo	26/06/2023 12:10	41DA8EEE
INFO_SRC	134	109	Archivo	26/06/2023 12:10	ABFF824E
LICENSE	71,046	22,653	Archivo	26/06/2023 12:10	E1A96E11
README	1,624	732	Archivo	26/06/2023 12:10	C7F0220A

Configuración y creación del proyecto



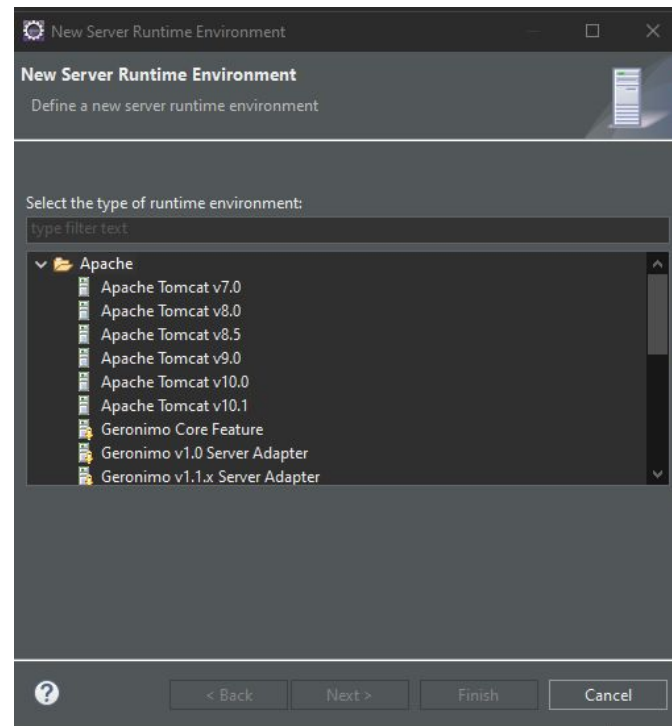
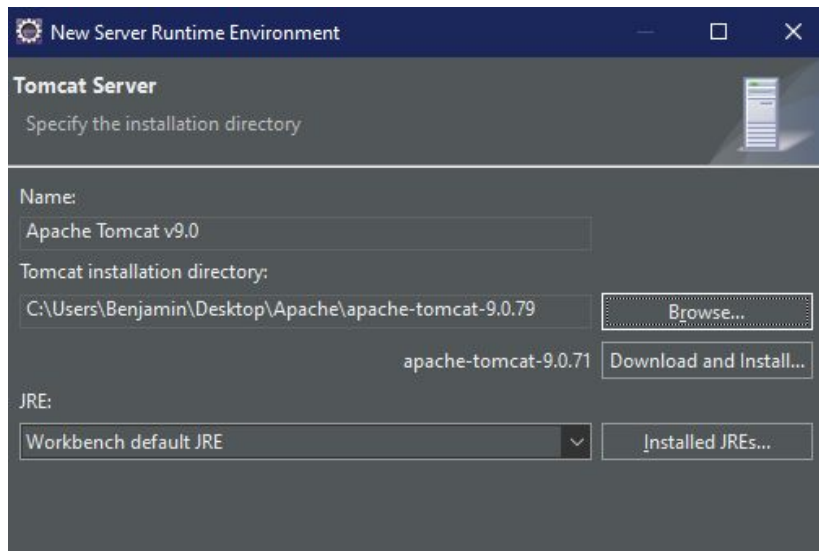
Configurando Eclipse IDE con Tomcat y Java 8 (1/5)

- File → New → Dynamic Web Project
- Se debe dar un nombre al proyecto (WebApp)
- Se da click a New Runtime



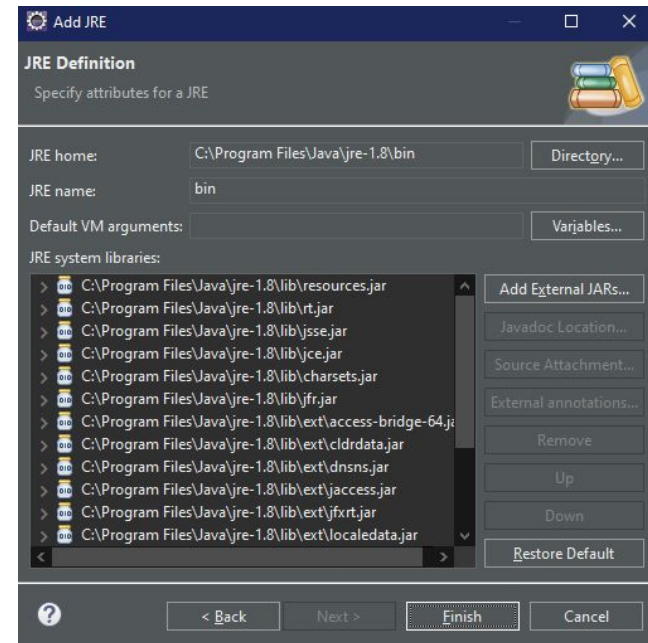
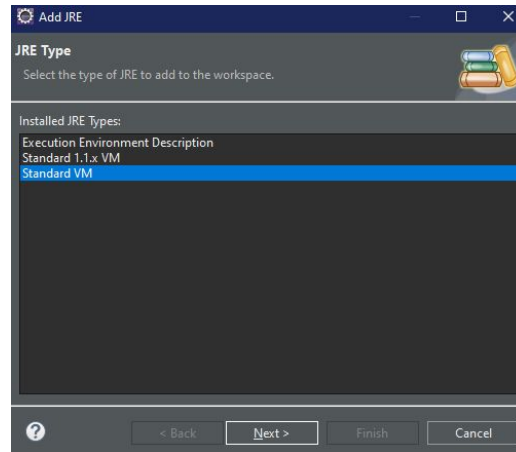
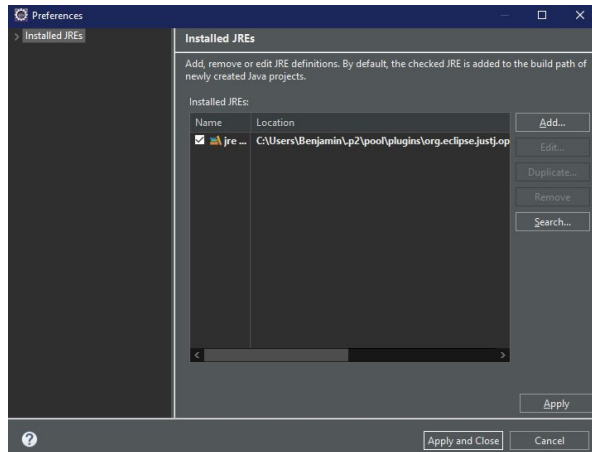
Configurando Eclipse IDE con Tomcat y Java 8 (2/5)

Aquí vamos a seleccionar Apache TomCat 9.0 y seleccionamos la carpeta donde lo descomprimos y luego le damos click en Finish



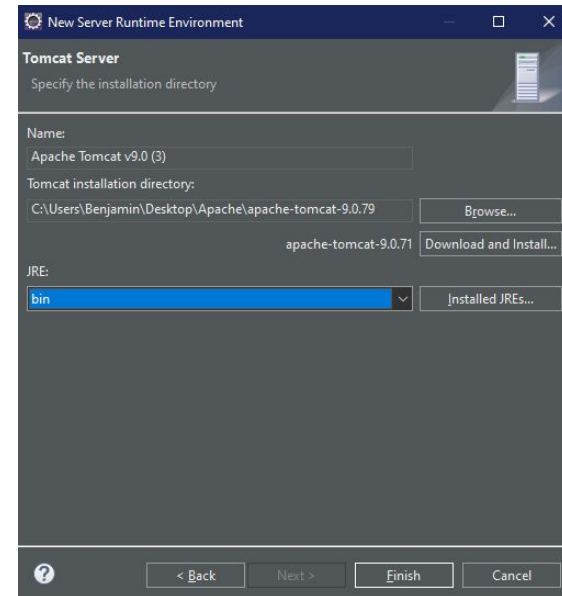
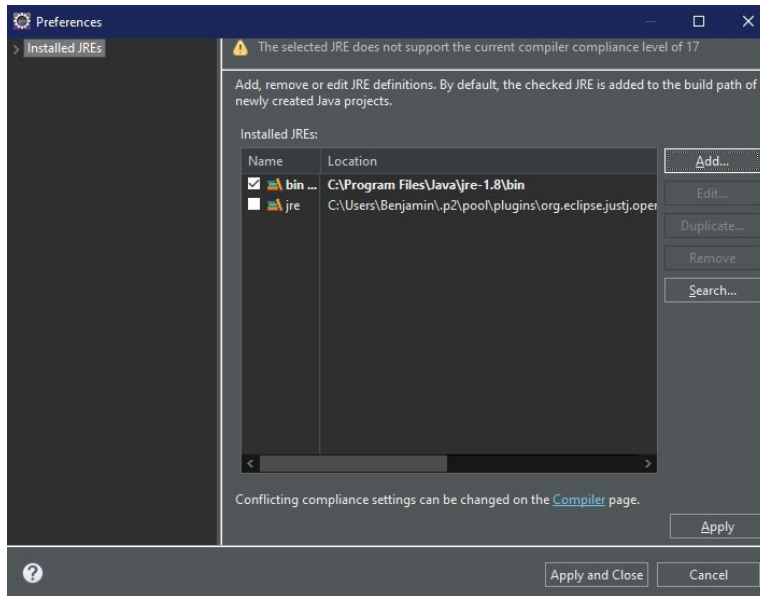
Configurando Eclipse IDE con Tomcat y Java 8 (3/5)

Luego debemos hacer click en Installed JREs y le damos en Add.
Aquí haremos click en Standard VM y para JRE Home seleccionaremos “C:\Program Files\Java\jre-1.8\bin” y le damos a finish



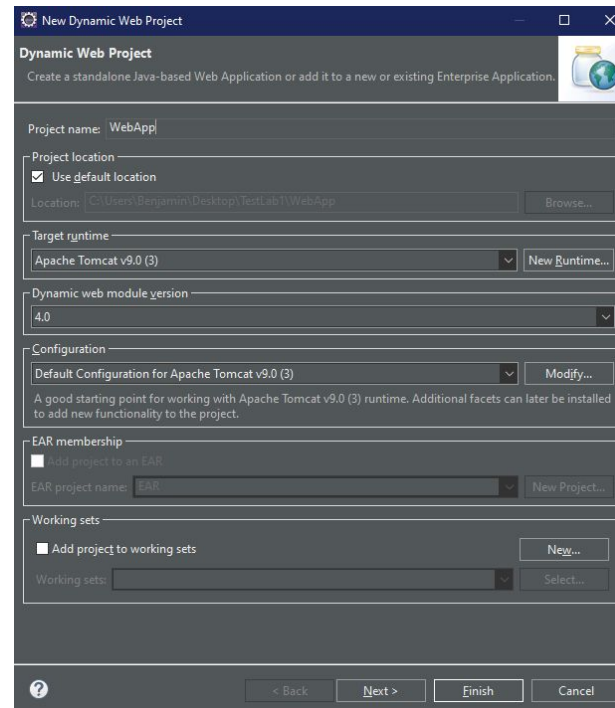
Configurando Eclipse IDE con Tomcat y Java 8 (4/5)

Una vez tengamos el nuevo bin, lo seleccionamos y le damos click en Apply and Close, Finalmente en JRE , Seleccionamos bin y le damos en Finish.



Configurando Eclipse IDE con Tomcat y Java 8 (5/5)

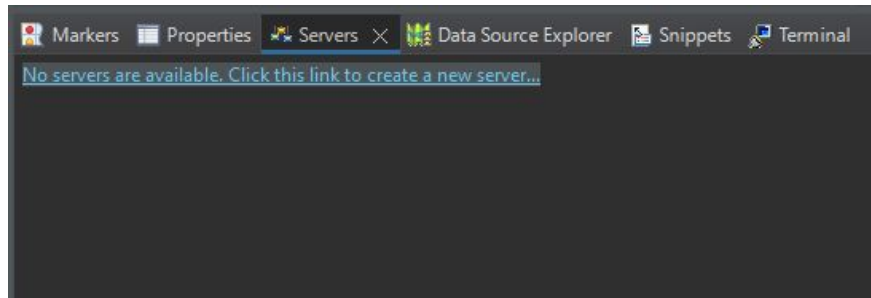
Para finalizar con la configuración le damos click en Finish



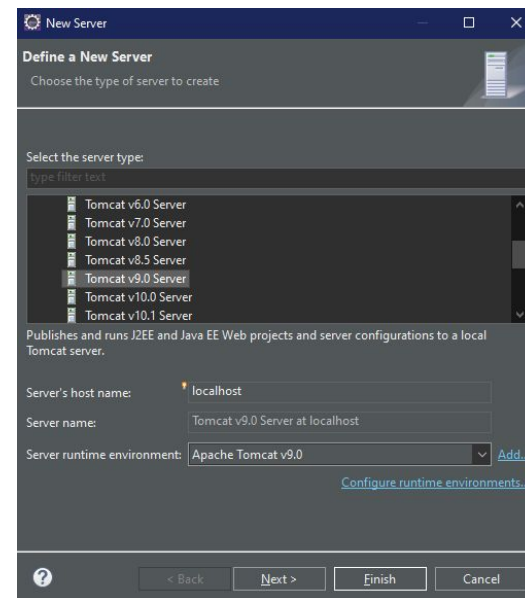
Configuración del servidor (1/2)

Para activar la vista de servidores, debemos ir a window, show view, servers

Una vez hecho esto, la pestaña servidores será visible en la parte inferior del programa

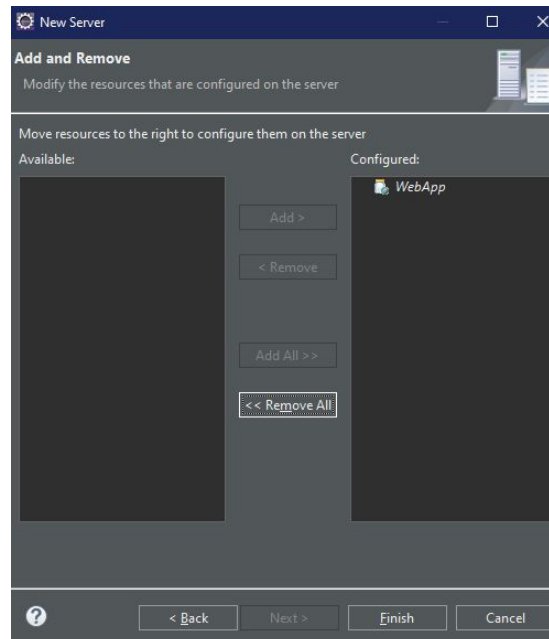
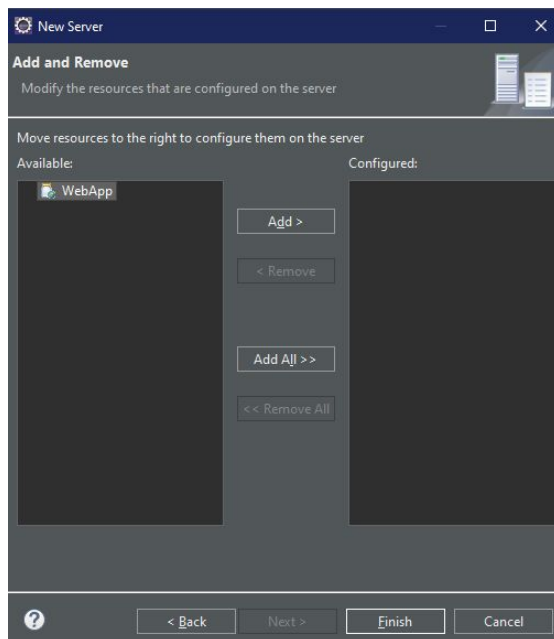


Una vez hecho esto daremos clic en crear nuevo servidor, Apache, Tomcat v9.0 y daremos en next



Configuración del servidor (2/2)

Aquí vamos a mover el proyecto a Configured y le daremos click en Finish

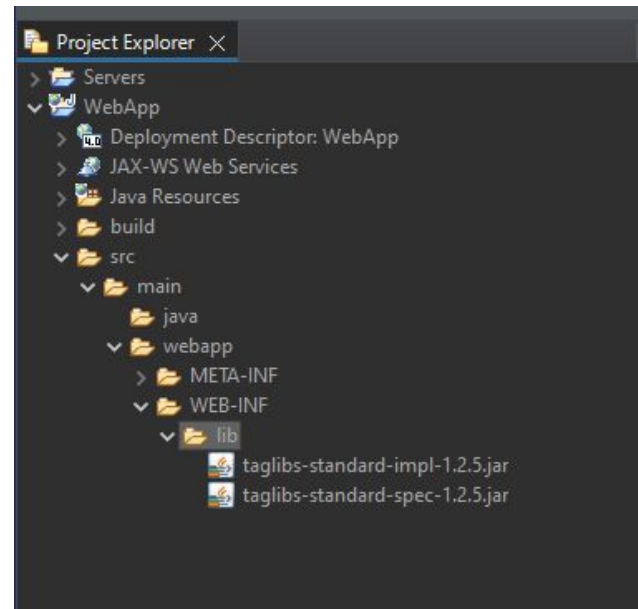


Configuración JSTL

En el folder del proyecto vamos a ubicar la carpeta SRC

Aquí vamos a expandir de la siguiente forma :
“src/main/webapp/WEB-INF/lib”.

Aquí vamos a copiar y a pegar los archivos
“taglibs-standard-impl-1.2.5.jar” y
“taglibs-standard-spec-1.2.5.jar”



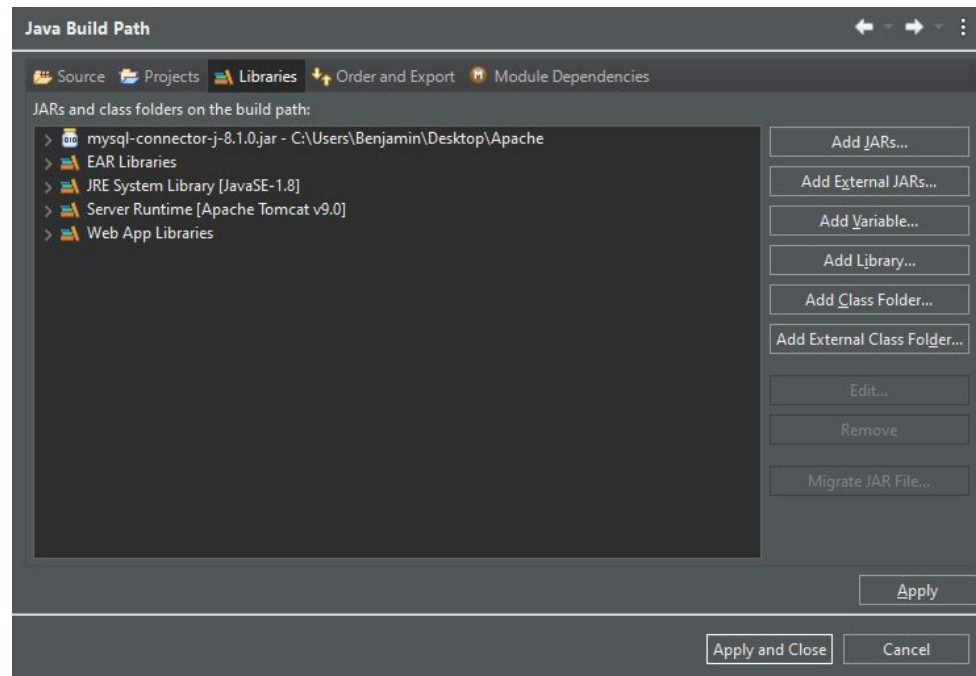
Configuración de la base de datos con MySQL Connector

Para esto vamos darle click derecho al proyecto y darle a properties.

Aquí iremos a Java Build Path y seleccionaremos la pestaña libraries.

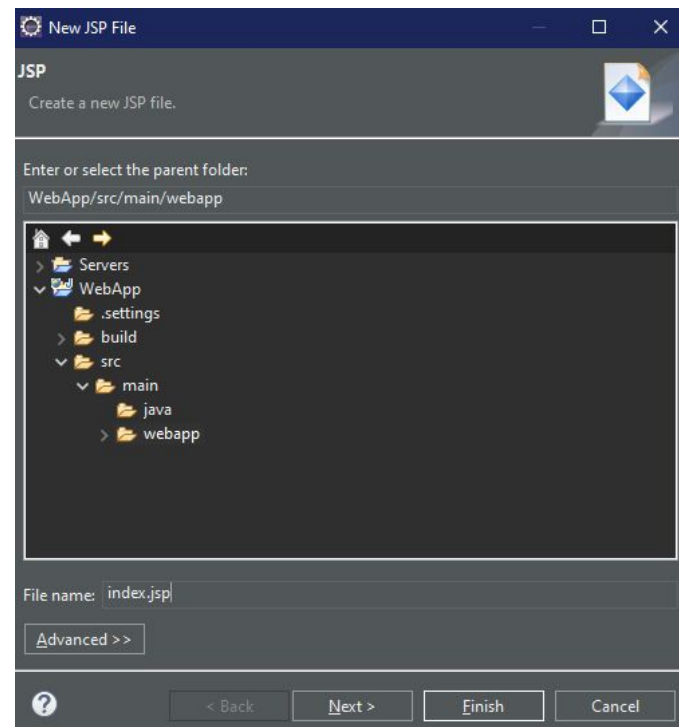
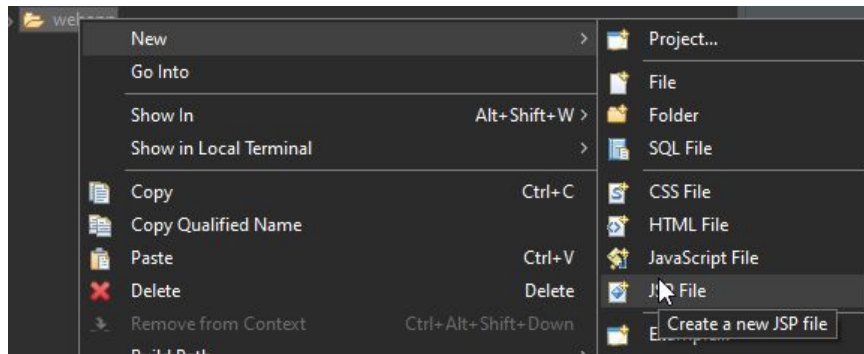
Finalmente le daremos click en Add External JARs y seleccionaremos el archivo previamente descargado.

Finalmente le daremos en Apply and Close



Test de nuestro proyecto (1/2)

Para poder hacer el test , lo que vamos a hacer es crear un nuevo archivo JSP en src/main/webapp/ y lo nombraremos como index.jsp



Test de nuestro proyecto (1/2)

Esto nos abrirá un nuevo documento donde podremos hacer los cambios respectivos para que sea visible en nuestro test, en este caso:

`<h1>Esta es nuestra página principal</h1>`

Además de un poco de color con :

`<style>`

`h1 { color: #00A8F7 ; }`

`</style>`

Y reemplazaremos el título por “Primera Prueba”

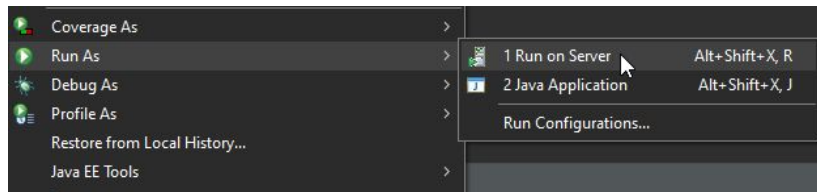
Finalmente grabaremos con Ctrl + S



```
index.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <style>
8   h1 { color: #00A8F7 ; }
9 </style>
10 <title>Primera prueba</title>
11 </head>
12 <body>
13 <h1> Esta es nuestra página principal</h1>
14 </body>
15 </html>
```

Test de nuestro proyecto (2/2)

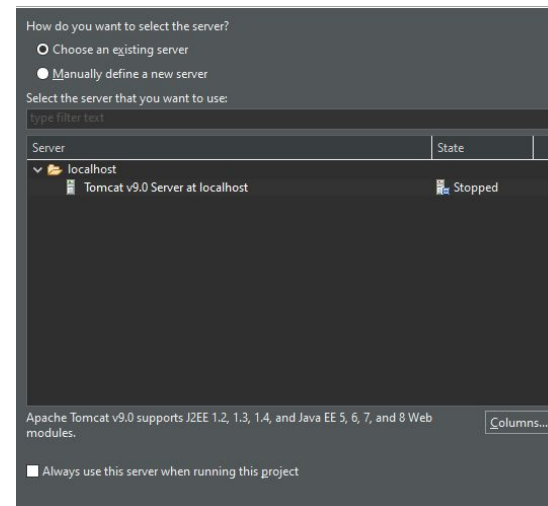
Para poder visualizar nuestro test le daremos click derecho a nuestro aplicativo y le daremos a Run As , Run on Server



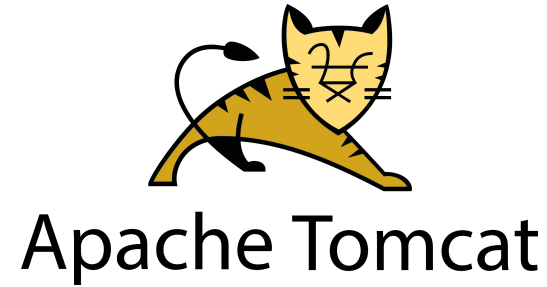
En caso de que nos salga una alerta de firewall , debemos dar click en permitir acceso.

Para poder acceder a nuestro test deberemos abrir una página en nuestro navegador y dirigirnos a "localhost:8080/WebServer/"

Aquí seleccionaremos nuestro servidor Tomcat 9.0 , Next y Finish



Implementación del proyecto

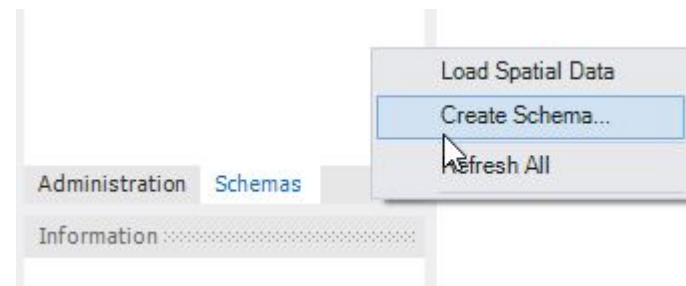
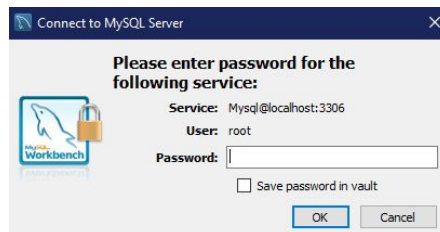
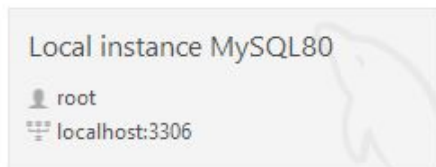


Inicialización de la base de datos (1/3)

Debemos abrir MySQL WorkBench
Luego hacemos click en Local Instance
MySQL80 y ingresamos nuestras credenciales

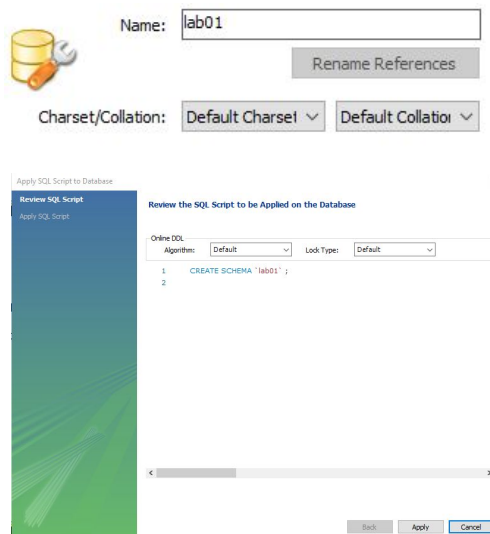
Luego en el apartado Schemas le damos click
derecho y seleccionamos create schema

MySQL Connections

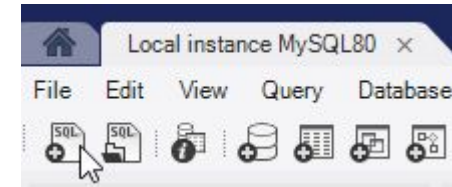


Inicialización de la base de datos (2/3)

Aquí generamos un nuevo schema al cual llamaremos lab 01



Una vez se ha creado el schema generaremos un nuevo query con el boton +SQL y añadiremos el siguiente query



```
CREATE TABLE lab01.product(  
id INT NOT NULL AUTO_INCREMENT,  
name VARCHAR(45) NOT NULL,  
stock INT NOT NULL,  
PRIMARY KEY(id)  
);
```

```
INSERT INTO lab01.product(name, stock)  
VALUES ('Laptop', 1);
```

Inicialización de la base de datos (3/3)

Para poder correr cualquier query , es necesario darle click al rayito



Este query nos generará una nueva tabla llamada productos con un nuevo row

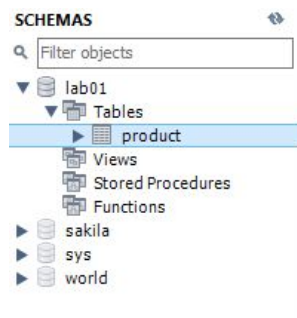
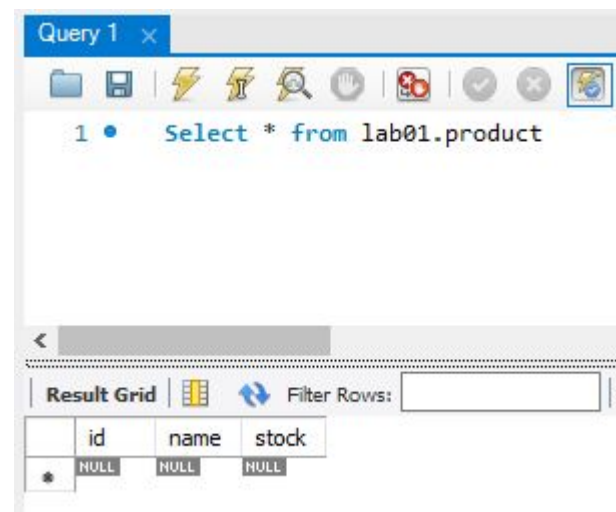


Table: **product**

Columns:

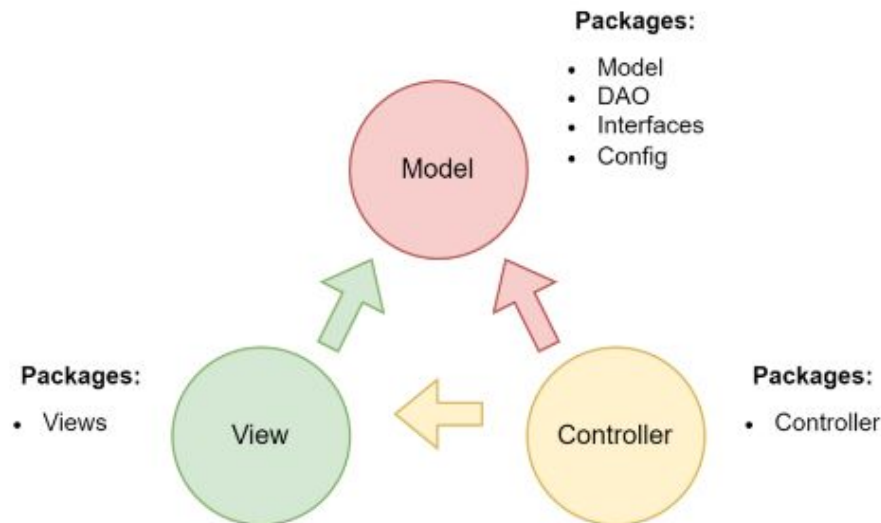
id int AI PK
name varchar(45)
stock int

Para poder ver los valores de nuestra nueva tabla solo debemos hacer `Select * from lab01.productos`



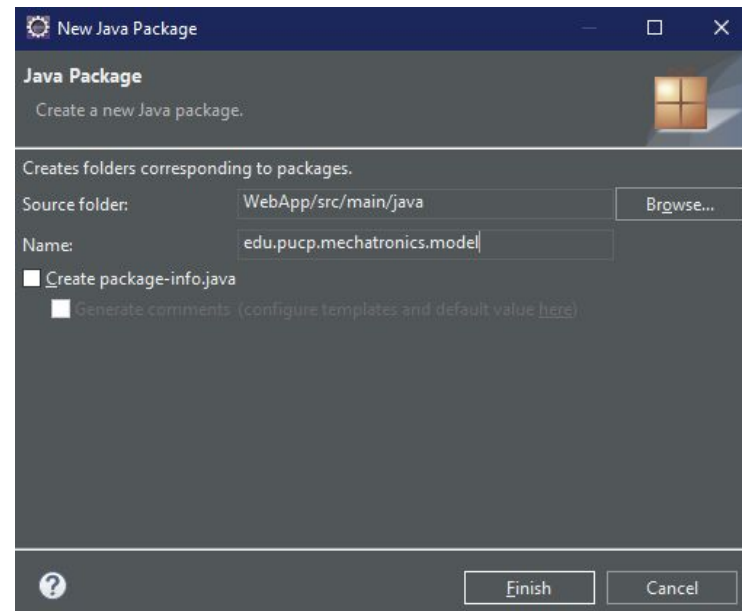
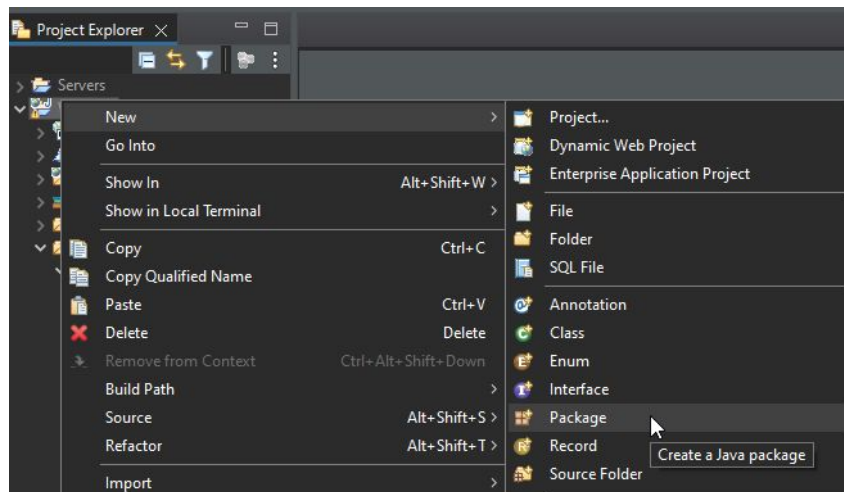
Arquitectura en Capas (1/4)

Para seguir un patrón MVC(Modelo-Vista-Controlador), utilizaremos paquetes para aislar las capas de nuestra aplicación.



Arquitectura en Capas (2/4)

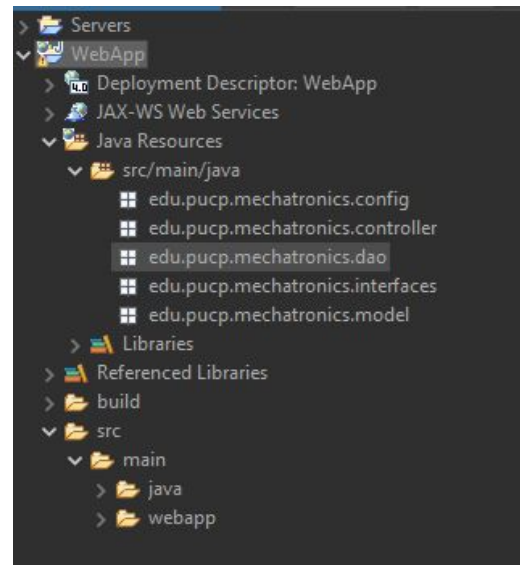
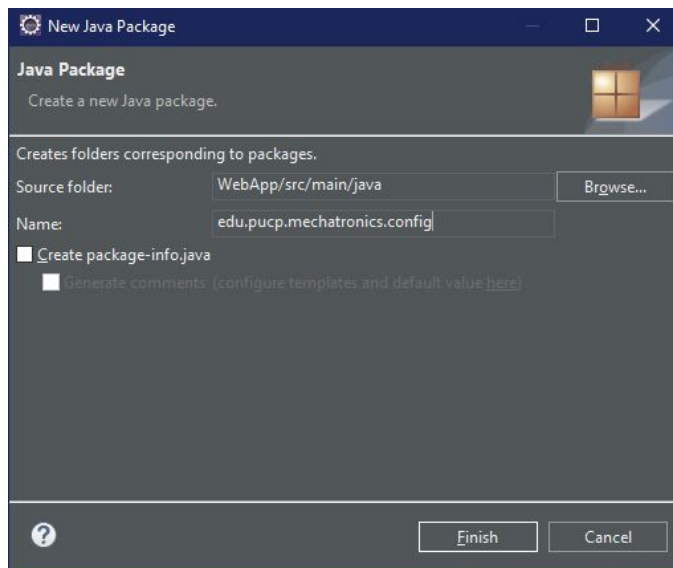
Para poder aplicar esto, en nuestro proyecto ya creado le daremos click derecho → new → paquete
El cual se llamara : edu.pucp.mechatronics.model



Arquitectura en Capas (3/4)

Una vez creado el nuevo paquete, crearemos otros nuevos paquetes añadiendo lo que buscamos luego del nombre como : **edu.pucp.mechatronics.model**

Los paquetes a crear seran: **controller,dao,interfaces y config**

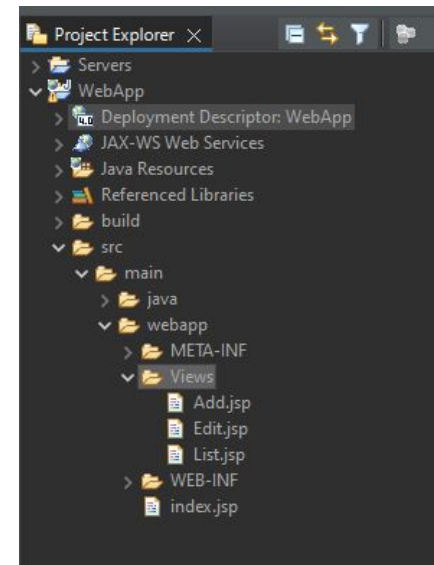
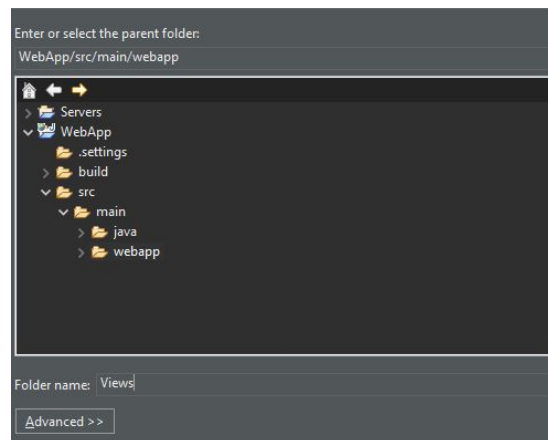
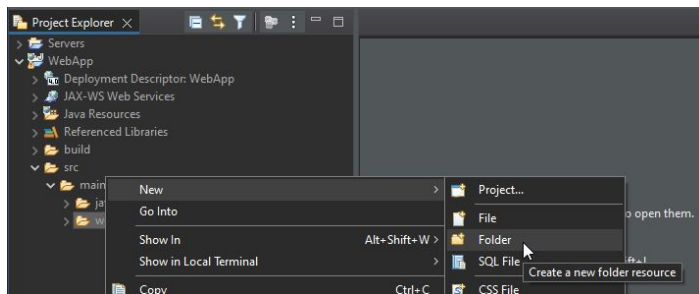


Arquitectura en Capas (4/4)

Nuestra Capa Vista , estará compuesta de archivos .jsp

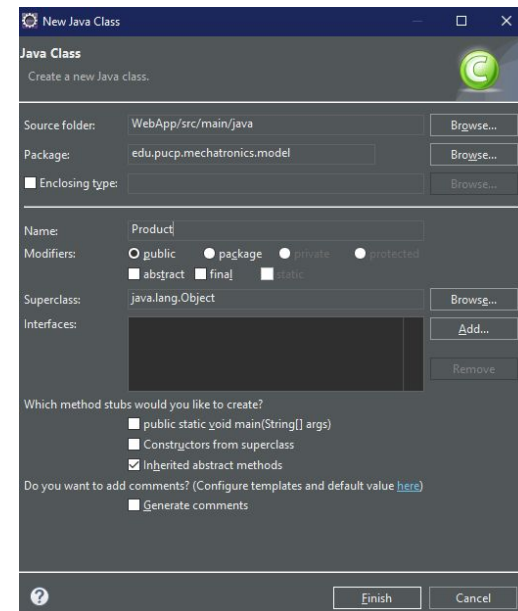
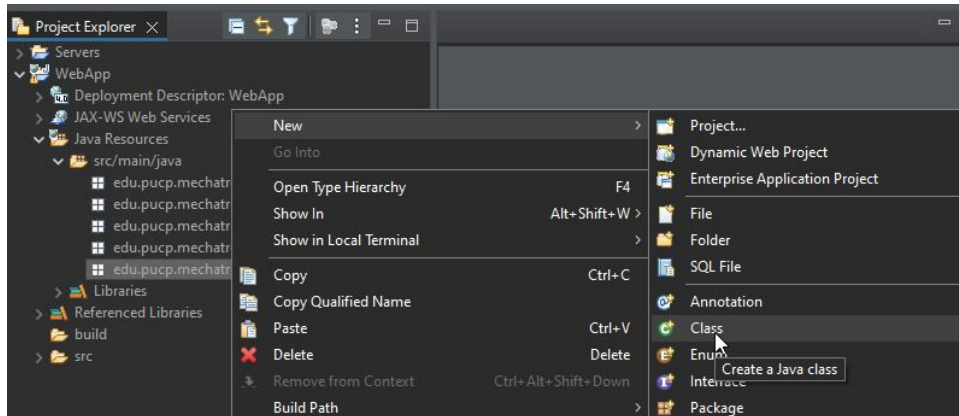
Por lo cual dentro de nuestro WebApp Folder, crearemos un nuevo folder llamado views.

Dentro de este folder crearemos las vistas, add, edit y list









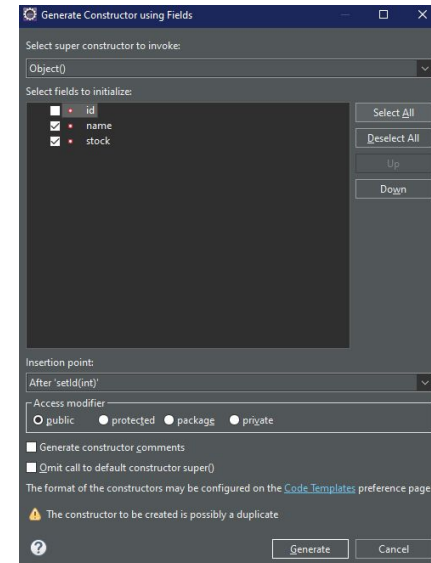
Paquete Model (1/4)

Aquí definiremos nuestra clase Producto, para eso crearemos una nueva clase dentro de nuestro paquete modelo , el cual se llamara Product



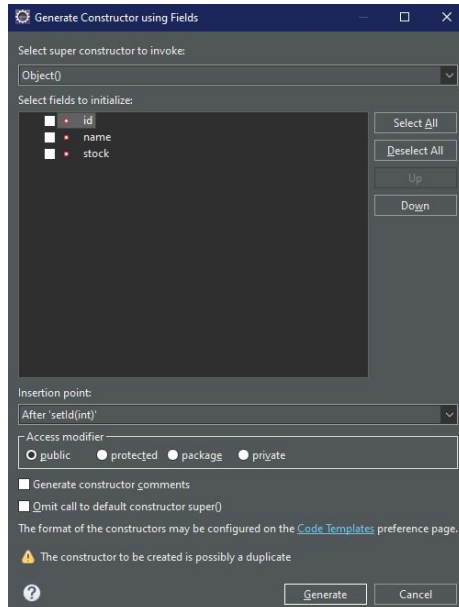
Una vez creado definiremos nuestra clase , la cual contendrá : id, name y stock
Una vez tengamos esto, podemos añadir los constructores dándole click derecho a nuestra clase → Source → Generate Constructor using Fields

Source	Alt+Shift+S >	Toggle Comment	Ctrl+7
Refactor	Alt+Shift+T >	Remove Block Comment	Ctrl+Shift+V
Local History		Generate Element Comment	Alt+Shift+J
References	>	Correct Indentation	Ctrl+I
Declarations	>	Format	Ctrl+Shift+F
 Add to Snippets...		Format Element	
 Coverage As	>	Add Import	Ctrl+Shift+M
 Run As		Organize Imports	Ctrl+Shift+O
 Debug As	>	Sort Members...	
 Profile As		Clean Up...	
 GitHub	>	Override/Implement Methods...	
Team		Generate Getters and Setters...	
Compare With		Generate Delegate Methods...	
Replace With	>	Generate hashCode() and equals()...	
<input checked="" type="checkbox"/> Validate		Generate toString()...	
Preferences...		Generate Constructor using Fields...	
		Generate Constructors from Fields...	
		Generate Constructor Using Fields	



Paquete Model (3/4)

Los constructores que crearemos serán dos , un Constructor Producto que solo contenga Nombre y Stock , y otro que no contenga ninguno



```
package edu.pucp.mechatronics.model;

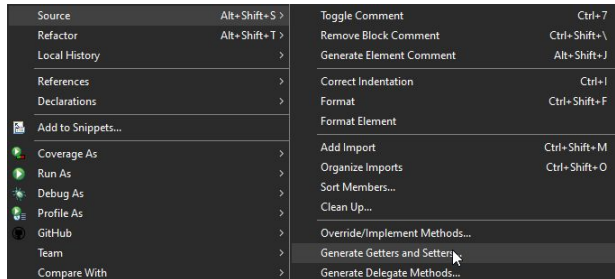
public class Product {
    private int id;
    private String name;
    private int stock;

    public Product() {
    }

    public Product(String name, int stock) {
        this.name = name;
        this.stock = stock;
    }
}
```

Paquete Model (4/4)

Finalmente Implementaremos los getters and setters de nuestra clase :



```
package edu.pucp.mechatronics.model;

public class Product {
    private int id;
    private String name;
    private int stock;

    public Product() {
    }

    public Product(String name, int stock) {
        this.name = name;
        this.stock = stock;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

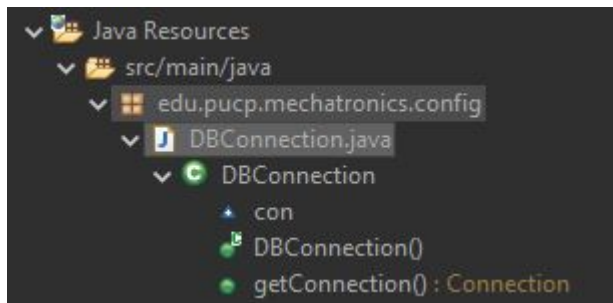
    public int getStock() {
        return stock;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }
}
```

Paquete Config

Este nos permite definir las conexiones con la base de datos así como también su interacción con este.

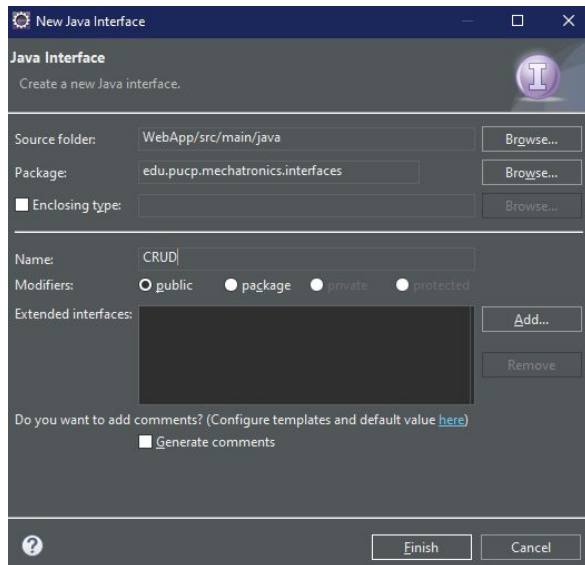
Para ello generamos una clase llamada DBConnection y insertamos el siguiente código



```
1 package edu.pucp.mechatronics.config;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6 public class DBConnection {
7     Connection con;
8     public DBConnection() {
9         try {
10             Class.forName("com.mysql.jdbc.Driver");
11
12             con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lab01", "root", "pucp2023");
13         }
14         catch (Exception e) {
15             System.err.println("Error:" + e);
16         }
17     }
18     public Connection getConnection() {
19         return con;
20     }
21 }
22
23 }
```

Paquete Interface

Este paquete define las actividades CRUD, las cuales son funciones que deben ser implementadas para realizar procedimientos como buscar, guardar, updatear o eliminar. Para ello generaremos una Interfaz llamada CRUD e insertamos el siguiente código :



```
1 package edu.pucp.mechatronics.interfaces;
2
3 import java.util.List;
4
5
6
7 public interface CRUD<E> {
8     public List<E> findAll();
9     public Product find(int id);
10    public boolean save(Product p);
11    public boolean update(Product p);
12    public boolean delete(int id);
13 }
```


Paquete DAO

Este paquete define como los objetos son leídos y escritos dentro de la base de datos. Para esto generamos una clase llamada ProductDao la cual implementara la interfaz CRUD y ingresamos el siguiente código :

```
1 package edu.pucp.mechatronics.dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.util.ArrayList;
7 import java.util.List;
8
9
10 import edu.pucp.mechatronics.config.DBConnection;
11 import edu.pucp.mechatronics.interfaces.CRUD;
12 import edu.pucp.mechatronics.model.Product;
13
14
15 public class ProductDAO implements CRUD<Product>{
16     DBConnection connDB = new DBConnection();
17     Product currentProduct = new Product(); //product for details, updates or deletion
18
19     @Override
20     public List<Product> findAll() {
21
22         List<Product> products = new ArrayList<>();
23         String sql = "select * from product;";
24         try {
25             Connection con = connDB.getConnection();
26             PreparedStatement statement = con.prepareStatement(sql);
27             ResultSet rs = statement.executeQuery();
28             while(rs.next()) {
29                 Product prod = new Product();
30                 prod.setId(rs.getInt("id"));
31                 prod.setName(rs.getString("name"));
32                 prod.setStock(rs.getInt("stock"));
33                 products.add(prod);
34             }
35         } catch (Exception e) {
36             System.err.println("Error: " + e.getMessage());
37         }
38     }
39
40     return products;
41 }
42
```

```
43 @Override
44 public Product find(int id) {
45     String sql = "select * from product where id=?";
46     try {
47         Connection con = connDB.getConnection();
48         PreparedStatement statement = con.prepareStatement(sql);
49         statement.setInt(1, id);
50         ResultSet rs = statement.executeQuery();
51         while(rs.next()) {
52             currentProduct.setId(rs.getInt("id"));
53             currentProduct.setName(rs.getString("name"));
54             currentProduct.setStock(rs.getInt("stock"));
55         }
56     } catch (Exception e) {
57         System.err.println("Error: " + e.getMessage());
58     }
59
60     return currentProduct;
61 }
62
63 @Override
64 public boolean save(Product p) {
65     String sql = "insert into product(name, stock) values(?,?)";
66     try {
67         Connection con = connDB.getConnection();
68         PreparedStatement statement = con.prepareStatement(sql);
69         statement.setString(1, p.getName());
70         statement.setInt(2, p.getStock());
71         statement.executeUpdate();
72     } catch (Exception e) {
73         System.out.println("Error: " + e.getMessage());
74     }
75
76     return false;
77 }
78
79
80
81 }
82
```

```
83 @Override
84 public boolean update(Product p) {
85     String sql = "update product set name=?,stock=? where id=?";
86     try {
87         Connection con = connDB.getConnection();
88         PreparedStatement statement = con.prepareStatement(sql);
89         statement.setString(1, p.getName());
90         statement.setInt(2, p.getStock());
91         statement.setInt(3, p.getId());
92         statement.executeUpdate();
93     } catch (Exception e) {
94         System.err.println("Error: " + e.getMessage());
95     }
96     return false;
97 }
98
99
100 @Override
101 public boolean delete(int id) {
102     String sql = "delete from product where id=?";
103     try {
104         Connection con = connDB.getConnection();
105         PreparedStatement statement = con.prepareStatement(sql);
106         statement.setInt(1, id);
107         statement.executeUpdate();
108     } catch (Exception e) {
109         System.err.println("Error: " + e.getMessage());
110     }
111     return false;
112 }
113
114 }
115
```

Paquete Controlador

Este paquete define la clase controlador quien es el encargado de tomar acciones mandadas a través de las vistas como transiciones, modificaciones , etc .

Para ello crearemos la clase Controller y añadiremos el siguiente código :

```
1 package edu.pucp.mechatronics.controller;
2
3 import java.io.IOException;
4 import java.util.List;
5
6 import javax.servlet.RequestDispatcher;
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13
14 import edu.pucp.mechatronics.model.Product;
15 import edu.pucp.mechatronics.dao.ProductDAO;
16
17 @WebServlet("/Controller")
18 public class Controller extends HttpServlet {
19     private static final long serialVersionUID = 1L;
20
21     String list = "views/list.jsp";
22     String edit = "views/edit.jsp";
23     String add = "views/add.jsp";
24     ProductDAO dao = new ProductDAO();
25
26     public Controller() {
27         super();
28     }
29
30     protected void doGet(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32
33         String currentAcces = "";
34         String action = request.getParameter("action"); //parameter from "index.jsp"
35         if(action.equalsIgnoreCase("list")) {
36
37             currentAcces = list;
38         }
39         else if(action.equalsIgnoreCase("goAdd")){ //from index.jsp
40             currentAcces = add;
41         }
42         else if(action.equalsIgnoreCase("add")){ //from add.jsp, submit button
43             String name = request.getParameter("txtName");
44             int stock = Integer.parseInt(request.getParameter("txtStock"));
45             Product prod = new Product(name, stock);
```

```

            Product prod = new Product(name, stock);
            dao.save(prod);
            currentAcces = list;
        }
        else if(action.equalsIgnoreCase("goEdit")) {
            request.setAttribute("idProd", request.getParameter("id"));
            currentAcces = edit;
        }
        else if(action.equalsIgnoreCase("update")) {
            int id = Integer.parseInt(request.getParameter("txtid"));
            String name = request.getParameter("txtName");
            int stock = Integer.parseInt(request.getParameter("txtStock"));
            Product prod = new Product();
            prod.setId(id);
            prod.setName(name);
            prod.setStock(stock);
            dao.update(prod);
            currentAcces = list;
        }
        else if (action.equalsIgnoreCase("Delete")) {
            int id = Integer.parseInt(request.getParameter("id"));
            dao.delete(id);
            currentAcces = list;
        }
        List<Product> products = dao.findAll();
        request.setAttribute("products", products); //link object for jsp
        RequestDispatcher view = request.getRequestDispatcher(currentAcces);
        view.forward(request, response);
    }
    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // 1000 Auto-generated method stub
        doGet(request, response);
    }
}
```

Vistas

Este apartado contiene todos los archivos JSP que permitirán ver y hacer update de los distintos objetos, este tipo de archivo permite que código java pueda ser insertado dentro de un código HTML , lo cual permite el uso de objetos variables y métodos.

```
1 %<@page import="java.util.Enumeration"%>
2 %<@page import="edu.pucp.mechatronics.model.Product"%>
3 %<@page import="edu.pucp.mechatronics.dao.ProductDAO"%>
4 %<@page language="java" contentType="text/html; charset=ISO-8859-1"
5   pageEncoding="ISO-8859-1"%>
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta charset="ISO-8859-1">
10 <title>Insert title here</title>
11 </head>
12 <body>
13 <div>
14 <h1>Edit Product Information</h1>
15 <%
16   ProductDAO dao = new ProductDAO();
17   int id = Integer.parseInt((String)request.getAttribute("idProd"));
18   Product p = dao.find(id);
19 %>
20 <form action="Controller"
21   Product:<br>
22   <input type="text" name="txtName" value="<%= p.getName() %%"><br>
23   Stock:<br>
24   <input type="text" name="txtStock" value="<%= p.getStock() %%"><br>
25   <input type="hidden" name="txtId" value="<%= p.getId() %%">
26   <input type="submit" name="action" value="Update"><br>
27 </form>
28 </div>
29 </body>
30 </html>
```

```
1 %<@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
2 %<@page language="java" contentType="text/html; charset=ISO-8859-1"
3   pageEncoding="ISO-8859-1"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="ISO-8859-1">
8 <title>Insert title here</title>
9 </head>
10 <body>
11 <div>
12 <h1>Products</h1>
13 <a href="Controller?action=goAdd">Create a new product</a>
14 <table border="1">
15 <thead>
16 <tr>
17 <th>ID</th>
18 <th>Product</th>
19 <th>Stock</th>
20 </tr>
21 </thead>
22 <tbody>
23 <c:forEach var="element" items="${products}">
24 <tr>
25 <td>${element.id}</td>
26 <td>${element.name}</td>
27 <td>${element.stock}</td>
28 <td>
29 <a href="Controller?action=goEdit&id=${element.id}">Edit</a>
30 <a href="Controller?action=delete&id=${element.id}">Delete</a>
31 </td>
32 </tr>
33 </c:forEach>
34 </tbody>
35 </table>
36 </div>
37 </body>
38 </html>
```

```
1 %<@page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 <div>
12 <h1>New Product</h1>
13 <form>
14   Product:<br>
15   <input type="text" name="txtName"><br>
16   Stock:<br>
17   <input type="text" name="txtStock"><br>
18   <input type="submit" name="action" value="Add"><br>
19 </form>
20 </div>
21
22 </body>
23 </html>
```

Index

Asi mismo , es necesario modificar el index , para que tengamos un punto de acceso en nuestra página

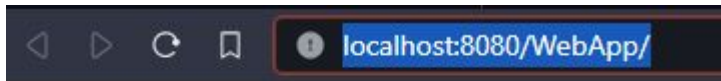
```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="ISO-8859-1">
7   <title>Insert title here</title>
8 </head>
9 <body>
10
11   <div>
12     <a href="Controller?action=list">Show products</a>
13     <a href="Controller?action=goAdd">Create a new product</a>
14   </div>
15
16 </body>
17 </html>
```

Correr nuestro proyecto

Para correr nuestro proyecto , es necesario que le demos click derecho → Run A → Run on Server

Aquí seleccionaremos nuestro servidor Tomcat 9.0 y le daremos en Finish

Nuestro Navegador nos dirigirá al url: "localhost:8080/WebServer/"



[Show products](#) [Create a new product](#)

Products

[Create a new product](#)

ID	Product	Stock	
3	Laptop	1	Edit Delete

New Product

Product:

Stock:

Add

En caso de que no se reconozca el driver en la ejecución

En caso de que esto suceda deberemos dar click derecho en nuestro proyecto → Run As → Run Configurations

Aquí seleccionaremos nuestro servidor TomCat y le daremos click on Classpath → Bootstrap Entries → Add External Jars

Aquí ingresamos nuestro connector Mysql en formato jar y le daremos en Apply

