

(P7) Aplicación backend con tecnología Java en servidores de aplicaciones

PRODUCTO 1

Instalación, configuración y autenticación

(Conv_0921)

Consultor: Prof. Paco Gómez Arnal

GRUPO - THE BINARIES

Asier Uruñuelas Sánchez
Francisca M. Rodríguez
Vázquez
Pau Egea Cortés
Sofía Figueroa Arocha

ÍNDICE

- 1. Descargar e instalar el entorno de programación JAVA EE (Jakarta EE).**
 - 1.1. JDK 14
 - 1.2. Eclipse IDE for Enterprise Java and Web Developers
 - 1.3. Prueba eclipse “Hola Mundo”
 - 1.4. Nuevo proyecto Maven
 - 1.4.1. Configuración properties archivo “pom.xml”
- 2. Analizar los diversos servidores de aplicaciones web disponibles.**
- 3. Instalar y configurar el servidor de aplicaciones web elegido.**
- 4. Desarrollar un programa que realice la autenticación de usuarios utilizando LDAP y Servlets.**
 - 4.1. Apache Directory Studio
 - 4.2. Instalar en Windows
 - 4.3. Configuración LDAP
 - 4.4. Desarrollar un programa que realice la autenticación de usuarios utilizando LDAP y Servlets.
- 5. Referencias.**

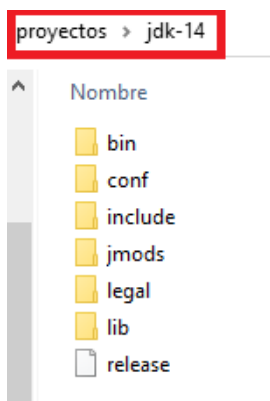
1. Descargar e instalar el entorno de programación JAVA EE (Jakarta EE)

Comenzamos el proyecto con la descarga e instalación de las herramientas necesarias para crear el entorno de servidor de aplicaciones web.

Para crear nuestro entorno Java EE (Jakarta EE), decidimos trabajar con las siguientes opciones:

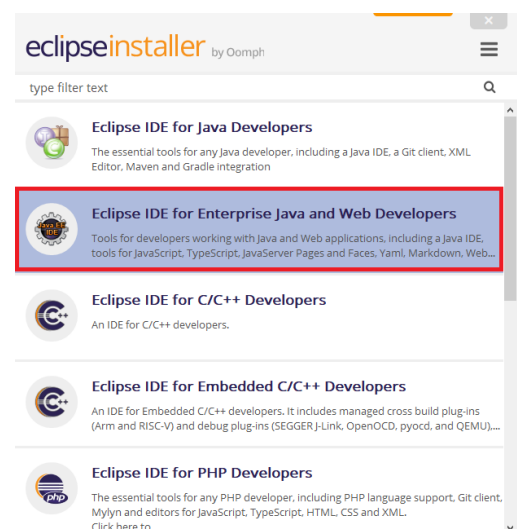
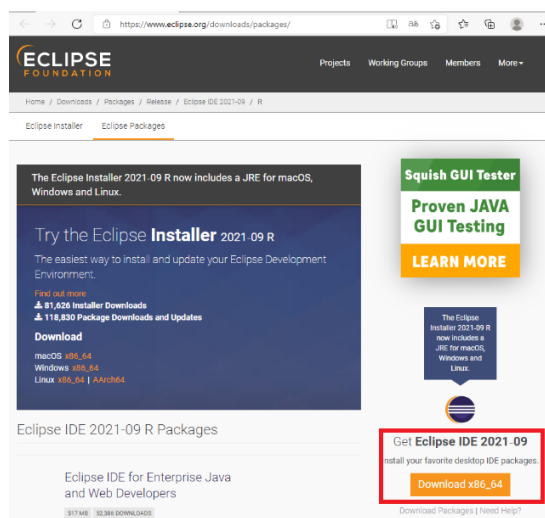
1.1. JDK 14

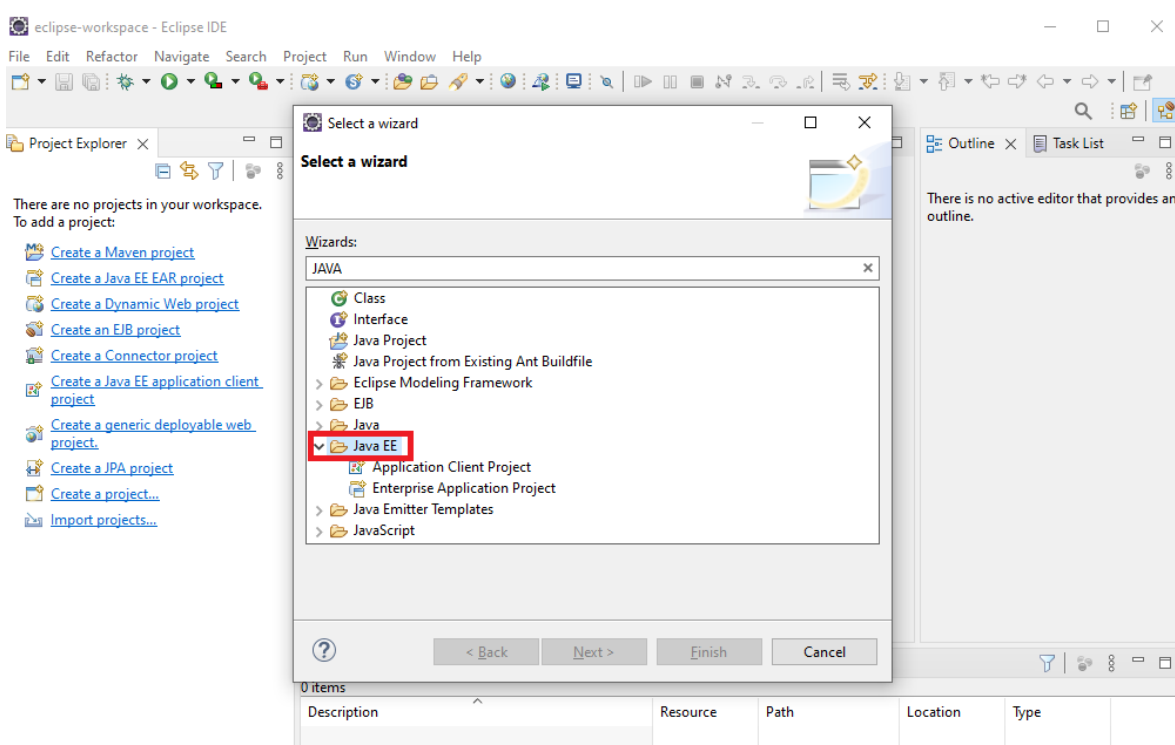
[JDK 14 Releases](#)



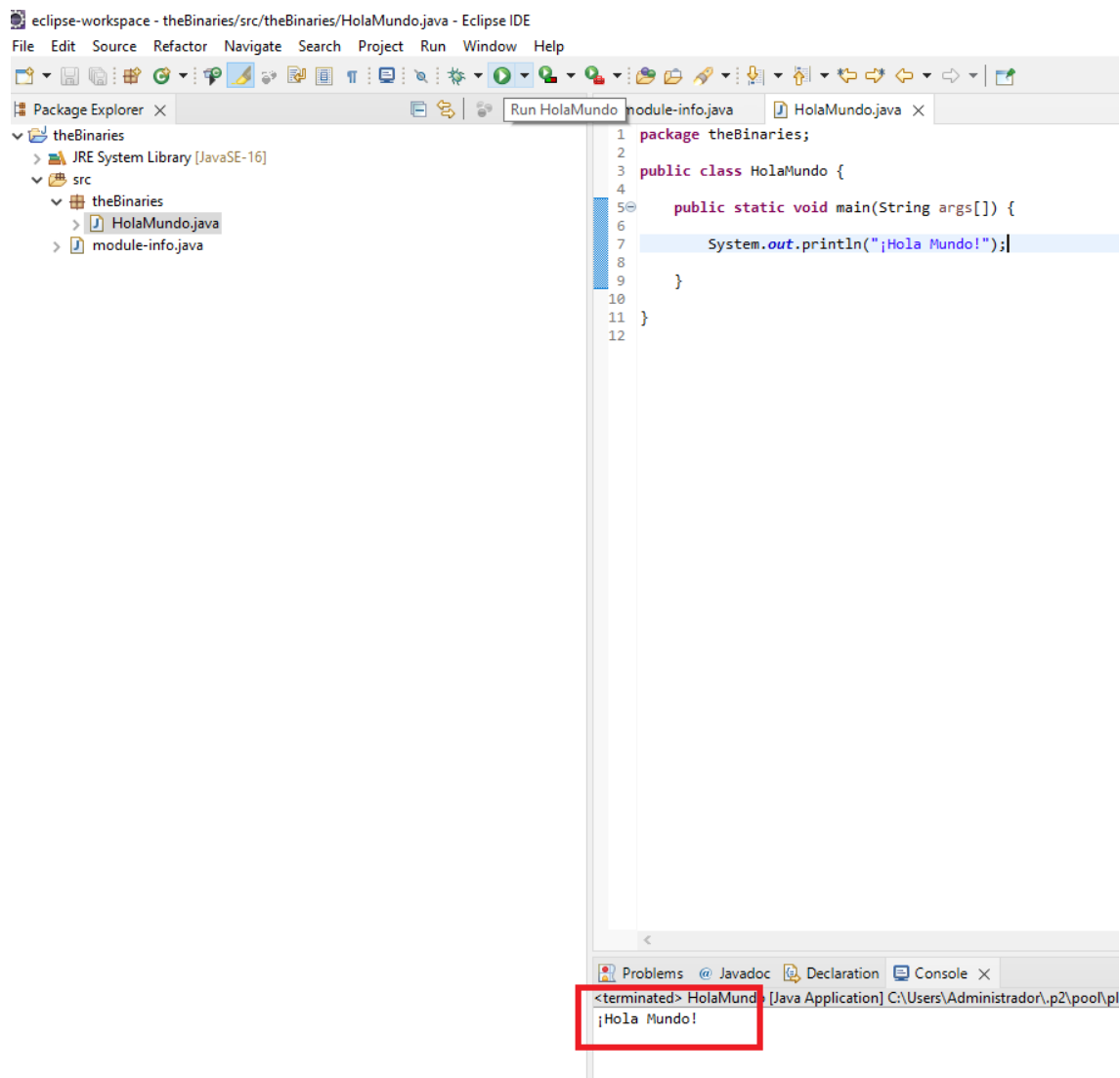
1.2. Eclipse IDE for Enterprise Java and Web Developers

[Eclipse Packages](#)

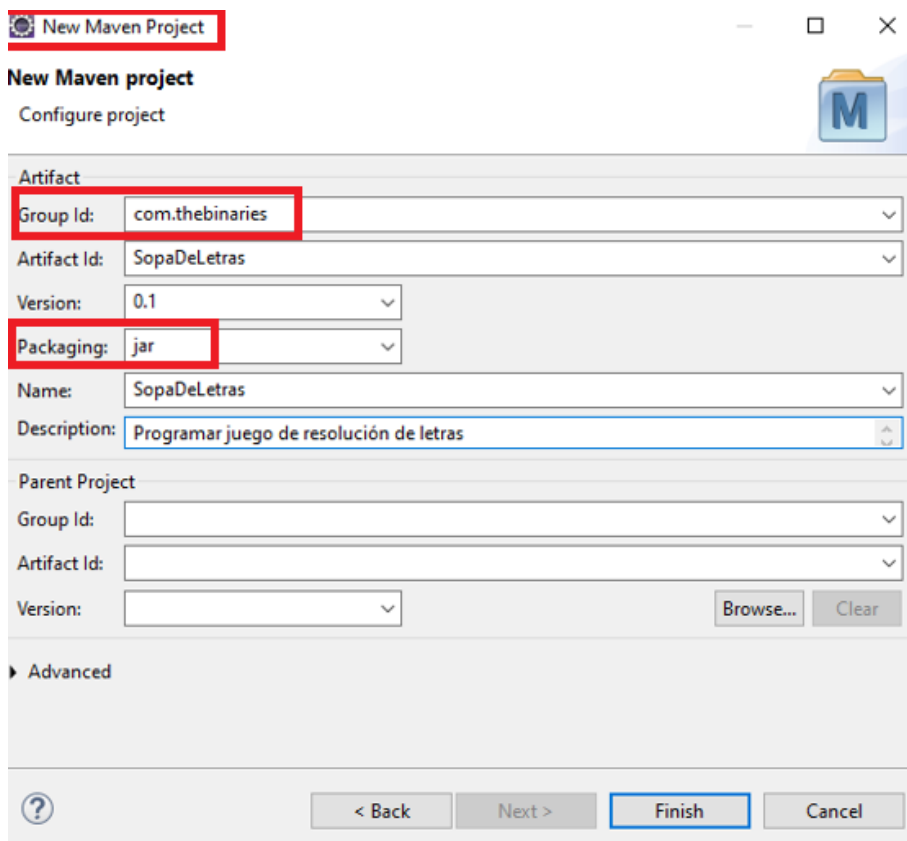




1.3. Prueba eclipse “Hola Mundo”



1.4. Nuevo proyecto Maven



New Maven Project

Configure project

Artifact

Group Id: com.thebinaries

Artifact Id: SopaDeLetras

Version: 0.1

Packaging: jar

Name: SopaDeLetras

Description: Programar juego de resolución de letras

Parent Project

Group Id:

Artifact Id:

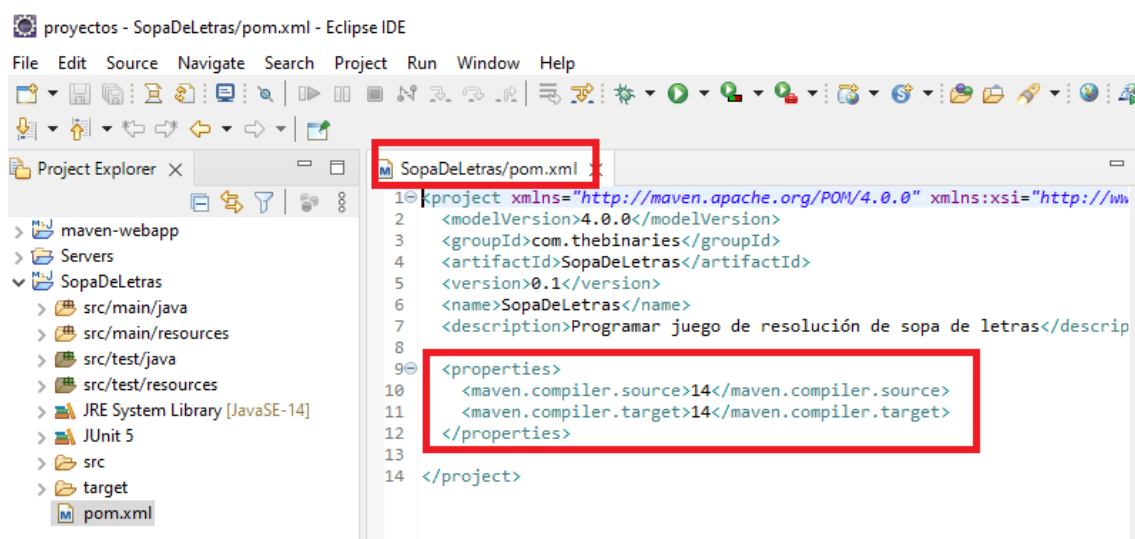
Version:

Browse... Clear

Advanced

< Back Next > Finish Cancel

1.4.1. Configuración properties archivo "pom.xml"



proyectos - SopaDeLetras/pom.xml - Eclipse IDE

File Edit Source Navigate Search Project Run Window Help

Project Explorer

- maven-webapp
- Servers
- SopaDeLetras
 - src/main/java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [JavaSE-14]
 - JUnit 5
 - src
 - target
 - pom.xml

SopaDeLetras/pom.xml

```

1 <?xml encoding="UTF-8" ?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
4   <groupId>com.thebinaries</groupId>
5   <artifactId>SopaDeLetras</artifactId>
6   <version>0.1</version>
7   <name>SopaDeLetras</name>
8   <description>Programar juego de resolución de sopa de letras</description>
9   <properties>
10    <maven.compiler.source>14</maven.compiler.source>
11    <maven.compiler.target>14</maven.compiler.target>
12  </properties>
13
14 </project>
  
```

2. Analizar los diversos servidores de aplicaciones web disponibles

El objetivo del siguiente apartado será analizar diferentes Servidores de Aplicaciones Web (a partir de ahora, SSAW), y para ello, para que el ejercicio no quede como una simple enumeración de SSAW creemos necesario cumplir con un seguido de pasos:

1. Definir qué es un servidor de aplicaciones web
2. Explicar qué es Java EE.
3. Qué es un contenedor de Servlets.
4. Qué es un Servlet.
5. Enumeración de SSAW.

En primer lugar, hemos seleccionado esta excelente definición de IBM para representar qué entendemos como servidor de aplicaciones, y dice así:

«Los servidores de aplicaciones son programas de servidor en una red distribuida que proporciona el entorno de ejecución para un programa de aplicación. Más específicamente, [...] es el componente de tiempo de ejecución principal en todas las configuraciones y donde una aplicación se ejecuta realmente. El servidor de aplicaciones colabora con el servidor web para ofrecer una respuesta dinámica y personalizada a una solicitud de cliente (IBM 2014)».

Esta definición es bastante completa y nos dice muchas cosas. Pero la idea más destacable es que los SSAW son los últimos responsables de velar por la ejecución de una aplicación. El otro elemento a destacar es que el SAW y el Servidor Web son dos tipos de servidores distintos. Mientras que el primero ejecuta la aplicación, el Servidor Web simplemente es un nodo (un dispositivo o una máquina) que se encarga de transferir la ejecución al cliente. Para entender los servidores de aplicaciones web, uno de los conceptos más importantes a saber es qué papel juega Java Enterprise Edition, conocido comúnmente como Java EE, en el desarrollo del lado servidor.

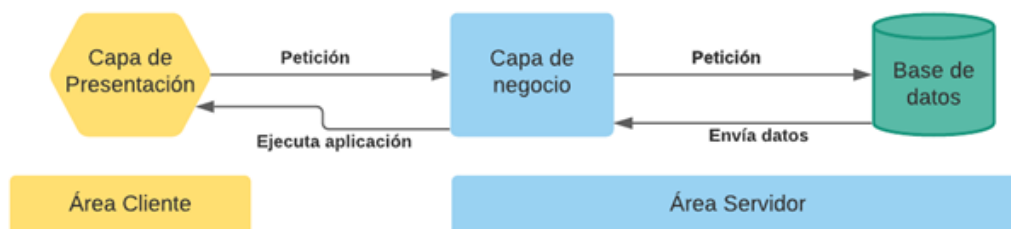
Java EE es una plataforma de programación, cuyo objetivo es desarrollar aplicaciones en Java. Pero a su vez, Java EE es considerado un estándar, o mejor dicho, un conjunto de normas y procedimientos, que dictan cómo desarrollar aplicaciones Java en el lado del servidor. Es decir, que la mayoría de los servidores de aplicaciones importantes están basados en los procedimientos de desarrollo de Java . Por eso, el conocimiento de Java EE es un punto de referencia importante para entender el funcionamiento de los SSAW.

Pero antes de entender la arquitectura de Java EE, hay que tener en cuenta el concepto programación por capas, que es un modelo estructurado dentro de la Arquitectura Cliente-Servidor (NO ES UN MODELO EXCLUSIVO DE JAVA). La idea es que el sistema esté dividido en capas independientes entre sí (no confundir con niveles, porque eso ya sería a nivel físico), por si ocurre algún tipo de error o fallo poder salvar el resto de estructuras del sistema. De este modo, el desarrollador podrá intervenir con más precisión en caso de que ocurra algún fallo en alguna capa. Lo habitual es que el sistema se divida, por lo menos, en tres capas – aunque es más complejo - adyacentes (Mírese Imagen 1). ¿Pero qué tiene que ver la programación con capas con la arquitectura de Java? La respuesta está en estas tres patas de Java EE :

1. **Servicios:** Estos pueden ser múltiples, y son proporcionados por un contenedor, que pueden comprender desde funciones de seguridad hasta funciones de registro. La idea de fondo es que el desarrollador pueda centrar su atención en programar en la lógica de negocio.
2. **Contenedor:** Es un Runtime Environment (RTE), como por ejemplo, Node.js. En pocas palabras, permite la portabilidad de una aplicación fuera del navegador (en el escritorio, servidor, etc.). Existen varios tipos de contenedores y la idea es que un conjunto de contenedores genere un SAW.

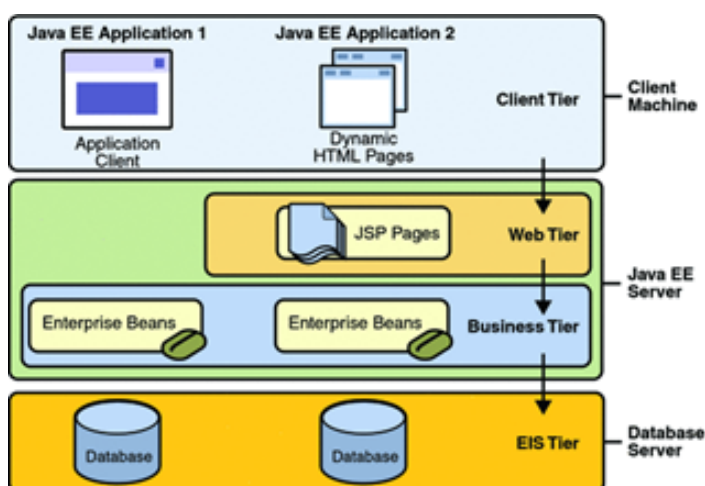
3. **Componentes:** Son objetos de Java que tienen dos características. La primera es que llevan la lógica de negocio de la aplicación, y la segunda es que usen los servicios del contenedor. Como los componentes tienen diferentes funcionalidades y características necesitan de varios contenedores para poder ejecutarse.

Imagen 1: Representación básica de la programación por capas.



Fuente: Elaboración propia.

Imagen 2: Aplicaciones de varios niveles (físico) de acuerdo a Java EE



Fuente: [Distributed Multitiered Applications - The Java EE 5 Tutorial](#)

Más tarde, enumeramos algunos de los servidores de aplicaciones más conocidos, pero es importante no confundir a Apache Tomcat como servidor de aplicaciones, cuando

en realidad es un contenedor de Servlets , aunque en la práctica pueda funcionar como un SAW.

¿Pero qué es un Servlet? Es un programa exclusivamente desarrollado en Java que se ejecuta en un servidor web. La diferencia que hay entre JSP (JavaServer Pages) y el Servlet, es que el primero es un programa escrito en Java y HTML, en cambio, un servlet es un programa más complejo escrito únicamente en Java. JSP se suele utilizar para ejecutar servidores en Aplicaciones sencillas, a diferencia de los servlets. No obstante, esta división no es excluyente, ya que ambas tecnologías tienen compatibilidad para coexistir y cooperar, sobre todo cuando se trata de ejecutar aplicaciones potentes y complejas . ¿Entonces para qué sirve el contenedor de servlets (Apache Tomcat)? La misión del contenedor de servlets «es recibir peticiones de páginas web y redireccionar estas peticiones a un objeto Servlet (edu4java.com: ¿fecha?)».

Una vez explicado esto, vamos a analizar las características de algunos de los SSAW disponibles:

- **Jetty:** Este es un servidor HTTP, de código abierto y completamente construido en Java. Además, Jetty también hace a la vez de contenedor de Servlets para el IDE (Entorno de Desarrollo Integrado) Eclipse. Según Facilcloud (Vergara 2016), Jetty tiene 5 características que le hacen una herramienta poderosa, y estas son: tiempos de carga reducidos, es liviano (gasta poca memoria), es escalable en cuanto a recepción de peticiones, tiene conexiones seguras y goza de un gran respaldo en la comunidad. Todo este conjunto de ventajas hacen que sea una tecnología fiable con la que poder construir un proyecto de SSAW escalable y de calidad.
- **Glassfish:** Es un servidor de aplicaciones, open source, que fue desarrollado por Sun Microsystems (ahora propiedad de Oracle), implementa los estándares

de Java EE y soporta un amplio abanico de tecnologías (JSP, Servlets, etc.). Al igual que Jetty también un gran apoyo en la comunidad y es una tecnología altamente fiable y utilizada (Delgado et al. 2017: 24-25) . La única diferencia respecto a Jetty es que el primero maneja únicamente peticiones en HTTP a diferencia de Glassfish que puede soportar más protocolos (aunque el primero es el más común).

- **Wildfly:** Es un servidor de aplicaciones, open source, implementado completamente en Java y soporta un amplio abanico de tecnologías al igual que los anteriores. Al igual que Jetty y Glassfish está apoyado por una gran comunidad y al igual que Glassfish tiene el certificado de Java EE, por tanto, goza de todas las garantías para compatibilizarlo con esta plataforma .
- **Apache Tomcat:** Tal y como se explicó anteriormente, Tomcat no es un Servidor de Aplicaciones a diferencia de Wildfly y Jetty, sino un contenedor de Servlets. Pero lo interesante de esta tecnología es este conjunto de características:

«El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache [que sí lo es] [...]. Hoy en día [...] Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad (Wikipedia 2021)».

Por ello, aunque Tomcat no es estrictamente un servidor de aplicaciones, se ha convertido en una especie de herramienta all-in-one, tal y como ocurre paralelamente con otros softwares de programación: por poner un ejemplo, el “editor” Visual Studio Code no es un IDE, pero es una herramienta muy poderosa que aglutina tantos plugins, que realmente es cada vez más difícil de distinguirlo de otros IDE como Eclipse o Visual Studio Community.

3. Instalar y configurar el servidor de aplicaciones web elegido

El consultor, en uno de los vídeos facilitados como recursos, recomienda usar Tomcat.

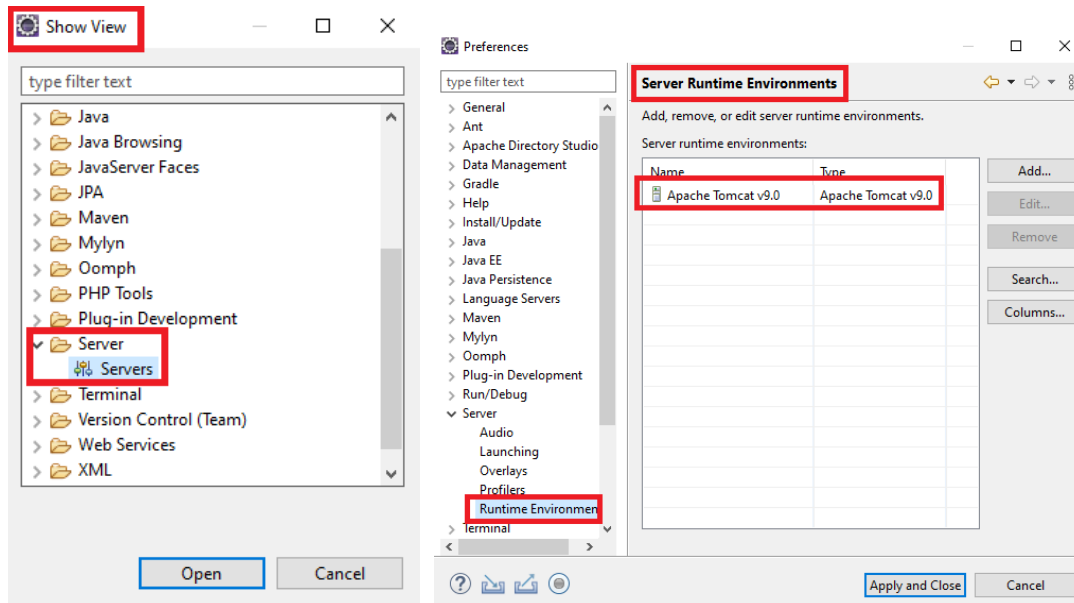
Así que es el seleccionado para desarrollar esta parte del producto.



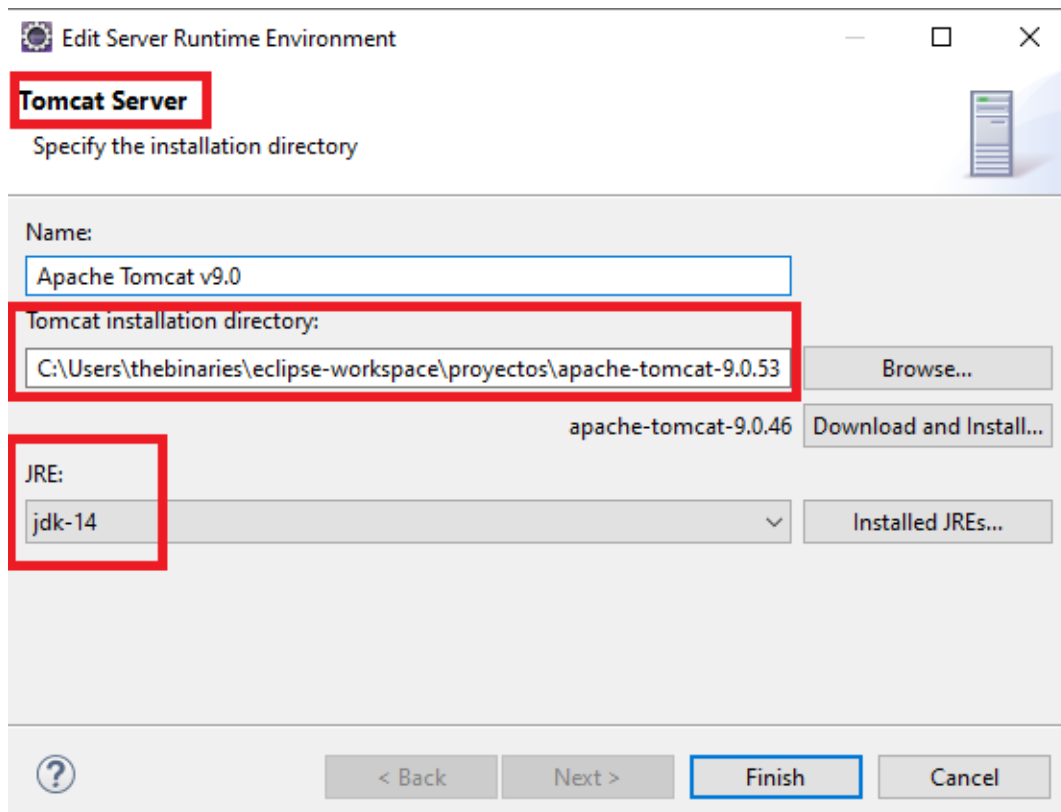
[tar.gz \(pgp, sha512\)](#)

| proyecto apache-tomcat-9.0.53 | | | |
|--------------------------------------|-----------------------|---------------------|--------|
| Buscar en apache-tomcat-9.0.53 | | | |
| Nombre | Fecha de modificación | Tipo | Tamaño |
| bin | 26/09/2021 21:30 | Carpeta de archivos | |
| conf | 26/09/2021 21:30 | Carpeta de archivos | |
| lib | 26/09/2021 21:30 | Carpeta de archivos | |
| logs | 26/09/2021 21:30 | Carpeta de archivos | |
| temp | 26/09/2021 21:30 | Carpeta de archivos | |
| webapps | 26/09/2021 21:30 | Carpeta de archivos | |
| work | 26/09/2021 21:30 | Carpeta de archivos | |
| BUILDING | 06/09/2021 21:09 | Documento de te... | 19 KB |
| CONTRIBUTING.md | 06/09/2021 21:09 | Archivo MD | 7 KB |
| LICENSE | 06/09/2021 21:09 | Archivo | 56 KB |
| NOTICE | 06/09/2021 21:09 | Archivo | 3 KB |
| README.md | 06/09/2021 21:09 | Archivo MD | 4 KB |
| RELEASE-NOTES | 06/09/2021 21:09 | Archivo | 7 KB |
| RUNNING | 06/09/2021 21:09 | Documento de te... | 17 KB |

El siguiente paso es instalar Tomcat como el servidor de nuestro IDE Eclipse



Se define el JDK-14

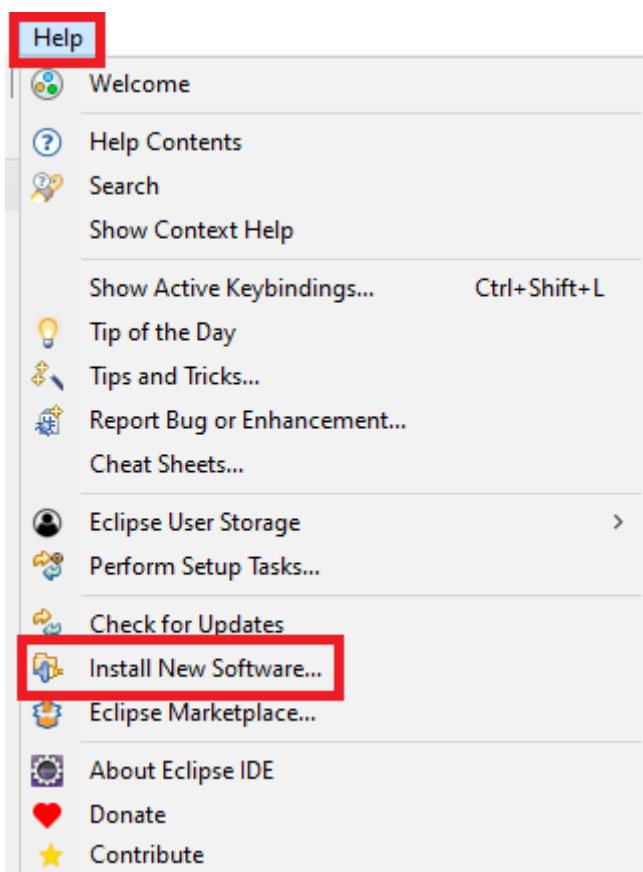


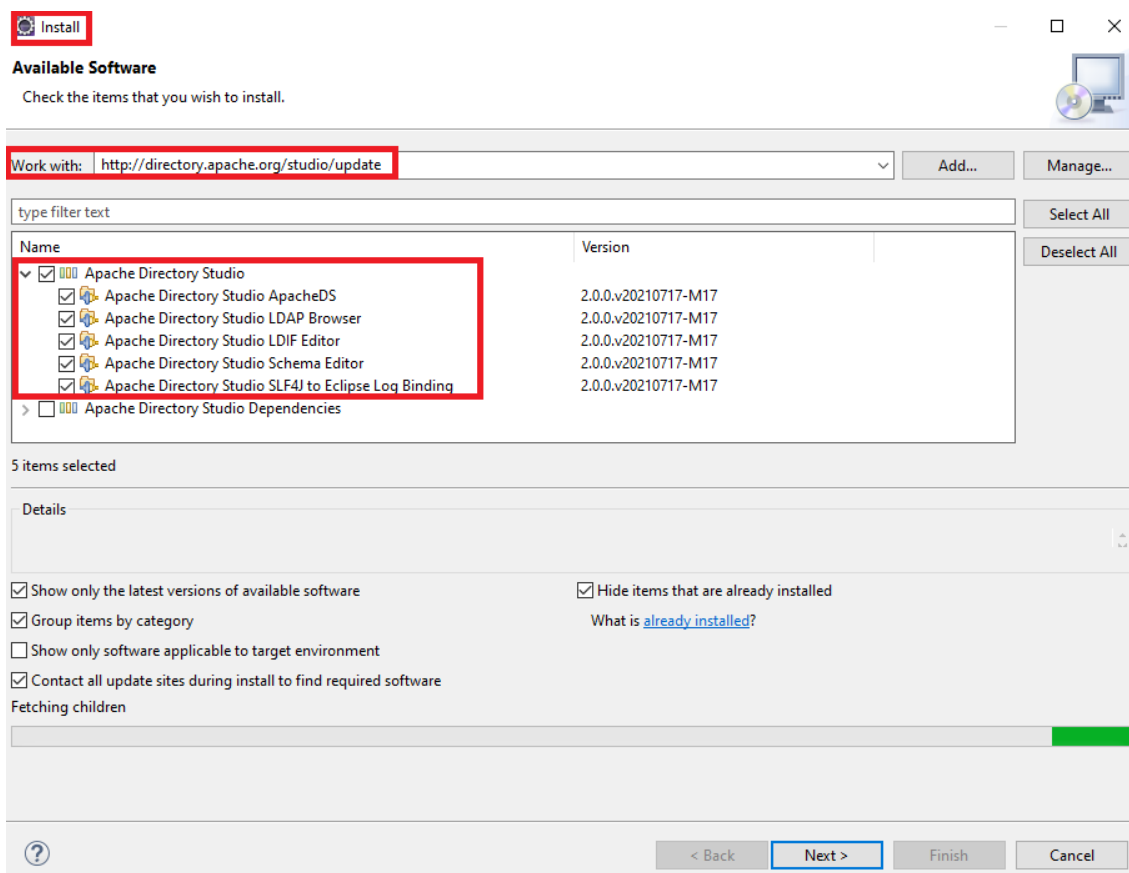
4. Desarrollar un programa que realice la autenticación de usuarios utilizando LDAP y Servlets

4.1. Apache Directory Studio

La herramienta de directorio recomendada por el consultor es “Apache Directory Studio” pensada para ser usada con cualquier servidor LDAP. Aunque su uso está diseñado principalmente para ApacheDS.

Comenzamos instalando los plugins en Eclipse siguiendo los pasos de la página oficial.

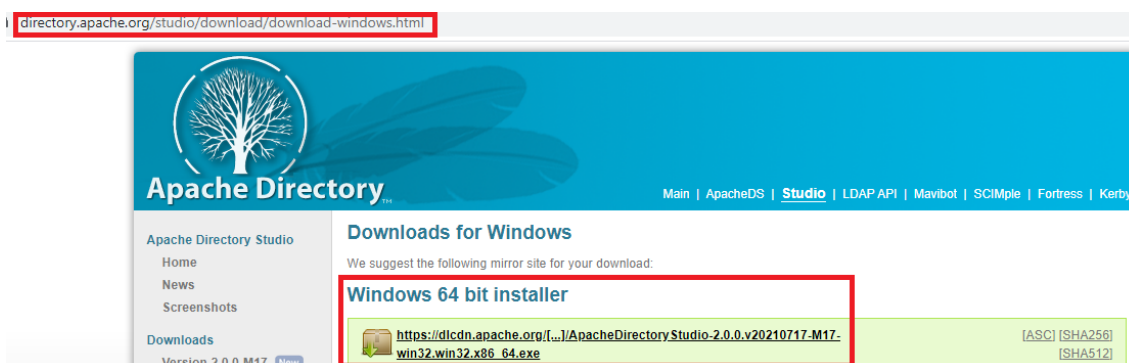


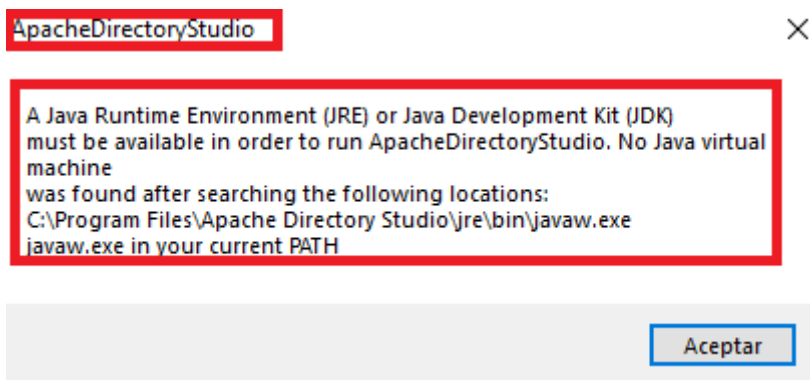
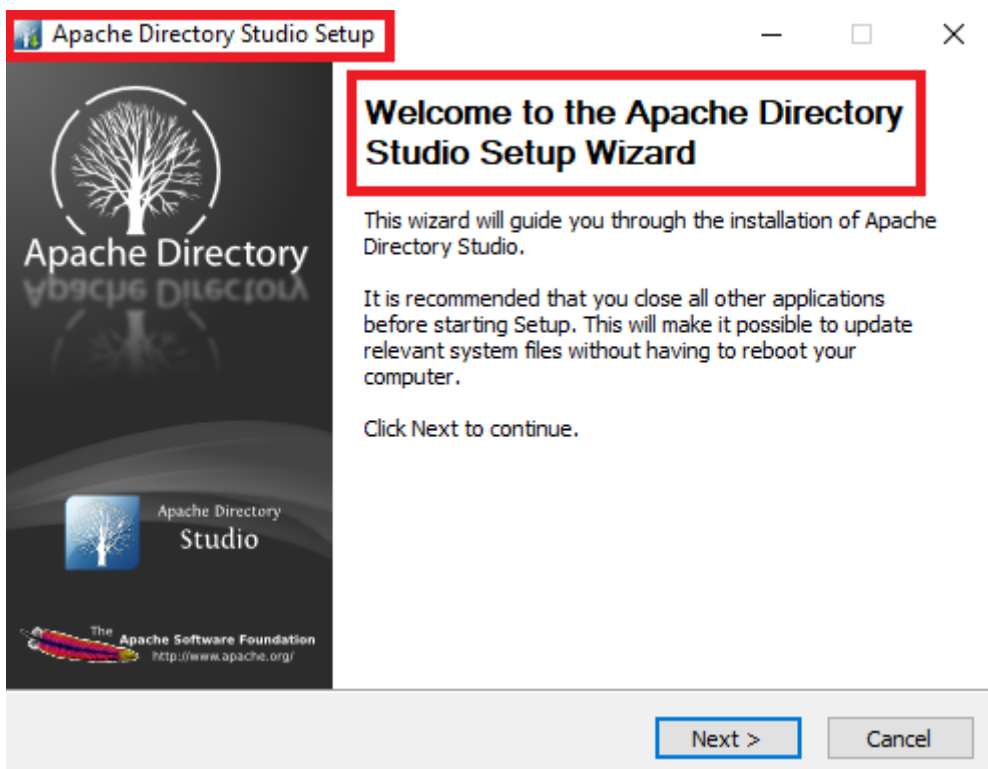


Se aceptan los términos del acuerdo de licencia. Después se instalan los programas informáticos no firmados cuya autenticidad o validez no pueda establecerse. Por último, se reinicia Eclipse para finalizar la instalación.

4.2. Instalar en Windows

<https://directory.apache.org/studio/downloads.html>





Para solucionar este problema, abrir fichero **ApacheDirectoryStudio.ini** en modo administrador, se descomenta la línea -vm y añadimos la ruta de nuestra máquina de java.


```
*ApacheDirectoryStudio: Bloc de notes
Archivo Edición Formato Ver Ayuda
ApacheDirectoryStudio.ini

-startup
plugins/org.eclipse.equinox.launcher_1.6.0.v20200915-1508.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.2.0.v20200915-1442
/studio-rcp/resources/icons/linux/studio.xpm
###
#Uncomment_to_configure_the_language
#https://directory.apache.org/studio/faqs.html#how-to-set-the-language-of-studio
#-nl
#en
###
#Uncomment_to_configure_Java_version_to_use
#https://directory.apache.org/studio/faqs.html#how-to-set-the-java-vm-to-use

-vm
C:\Users\thebinaries\eclipse-workspace\proyectos\jdk-14\bin

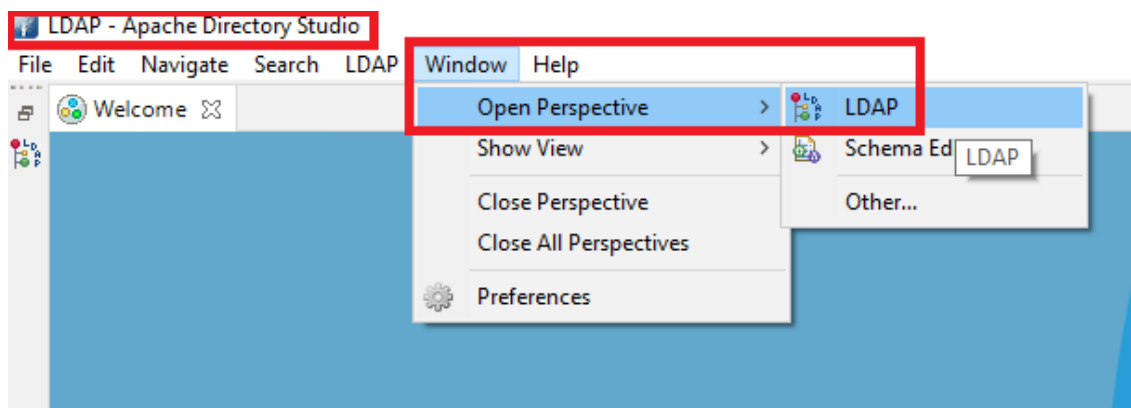
#/usr/lib/jvm/java-11-openjdk/bin/java
-vmargs
-Dosgi.requiredJavaVersion=11
```

En algunos casos puede mantenerse el error, de modo que tendremos que instalar el open-jdk, el cual podremos descargar de su página oficial, [aquí](#).

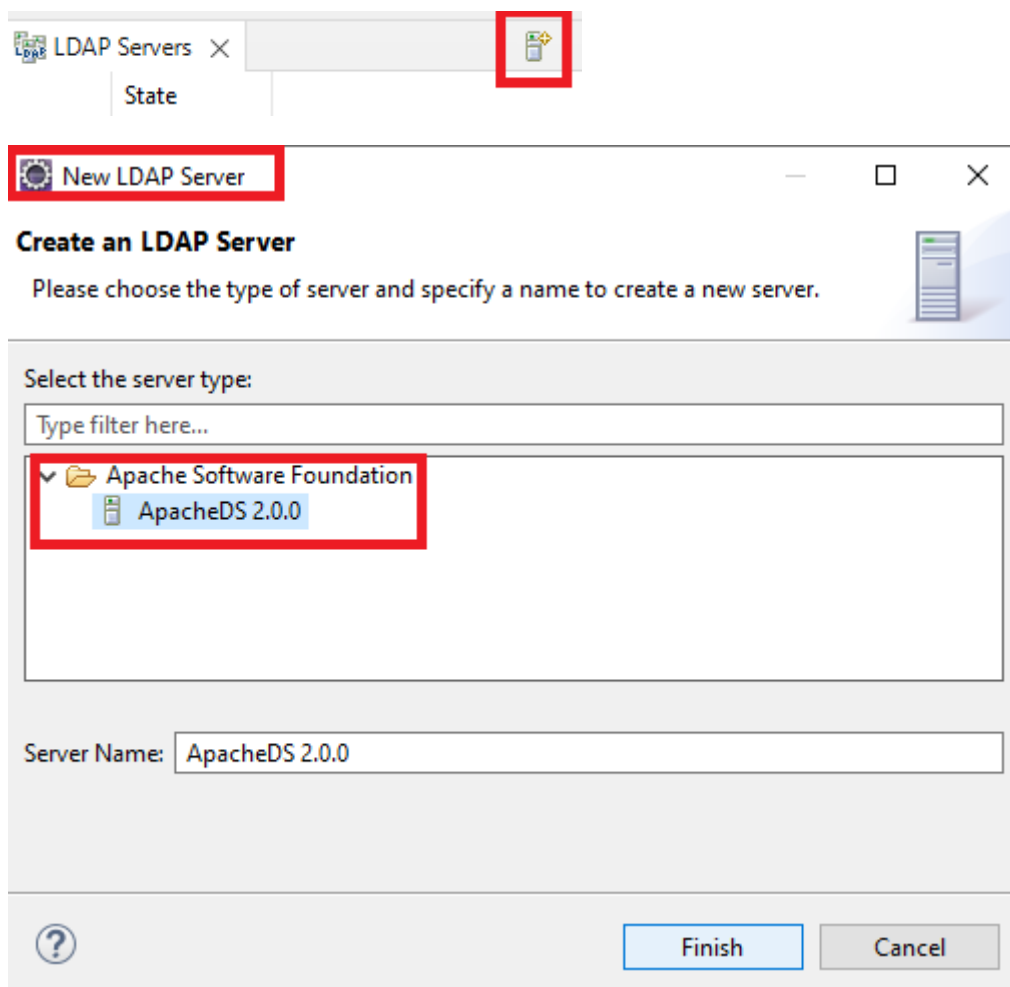
Una vez instalado ponemos el path donde tenemos instalado open-jdk, en el archivo de configuración.ini , de modo que se muestra en la siguiente captura de pantalla.

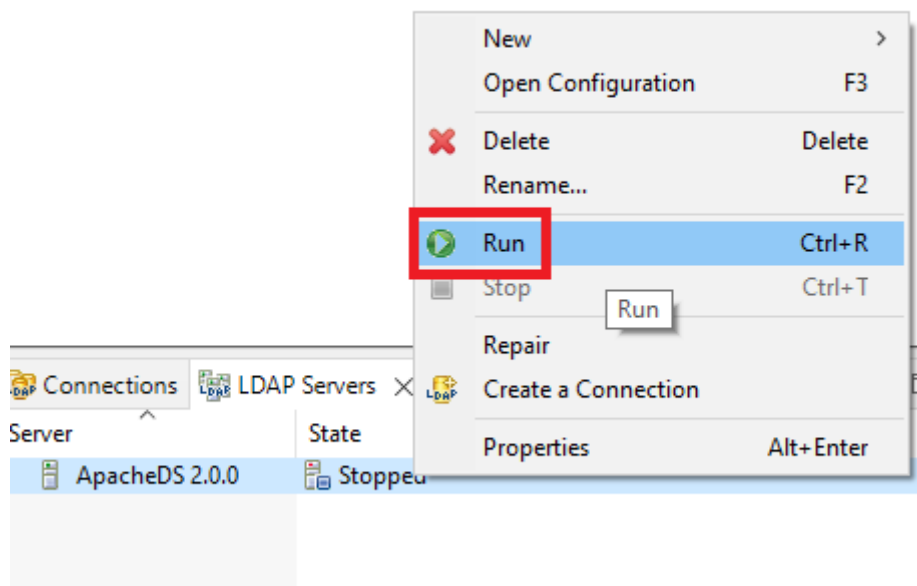
```
ApacheDirectoryStudio.ini [3]
1 -startup
2 plugins/org.eclipse.equinox.launcher_1.6.0.v20200915-1508.jar
3 --launcher.library
4 plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.2.0.v20200915-1442
5 /studio-rcp/resources/icons/linux/studio.xpm
6 ###
7 #Uncomment_to_configure_the_language
8 #https://directory.apache.org/studio/faqs.html#how-to-set-the-language-of-studio
9 #-nl
10 #en
11 ###
12 #Uncomment_to_configure_Java_version_to_use
13 #https://directory.apache.org/studio/faqs.html#how-to-set-the-java-vm-to-use
14 -vm
15 C:\Program Files\Java\jdk-11\bin
16 -vmargs
17 -Dosgi.requiredJavaVersion=11
18 ###
19 #Uncomment_to_configure_heap_memory
20 #https://directory.apache.org/studio/faqs.html#how-to-increase-the-heap-memory
21 #-Xms1g
22 #-Xmx2g
23
```

4.3. Configuración LDAP

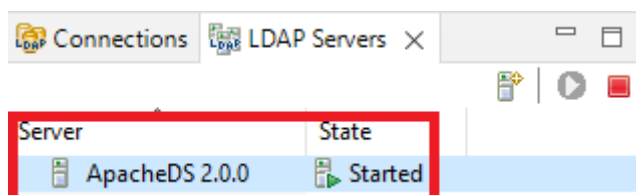



Se añade un servidor pulsando en el icono de la derecha.






En caso de tener problemas, se recomienda desactivar el firewall. Una vez arrancado, se procede a establecer una nueva conexión.



 **New LDAP Connection**

Network Parameter

Please enter connection name and network parameters.



Connection name:

Network Parameter

Hostname:


Port:

Connection timeout (s):

Encryption method:

Server certificates for LDAP connections can be managed in the '[Certificate Validation](#)' preference page.

☐ Read-Only (prevents any add, delete, modify or rename operation)



The image shows two overlapping Windows-style dialog boxes. The top dialog, titled 'New LDAP Connection', has a tab labeled 'Authentication'. It contains a message: 'Please select an authentication method and input authentication data.' Below this, there is a section for 'Authentication Method' with a dropdown menu set to 'Simple Authentication'. Underneath is the 'Authentication Parameter' section, which includes a 'Bind DN or user:' field with the value 'uid=admin,ou=system' and a 'Bind password:' field with masked characters (dots). There is a checkbox for 'Save password' which is checked, and a 'Check Authentication' button. The bottom dialog, titled 'Check Authentication', displays an information icon and the message 'The authentication was successful.' It has an 'OK' button. At the bottom of the second dialog are navigation buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

New LDAP Connection

Authentication

Please select an authentication method and input authentication data.

Authentication Method
Simple Authentication

Authentication Parameter
Bind DN or user: uid=admin,ou=system
Bind password: ●●●●●●
☒ Save password
Check Authentication

Check Authentication

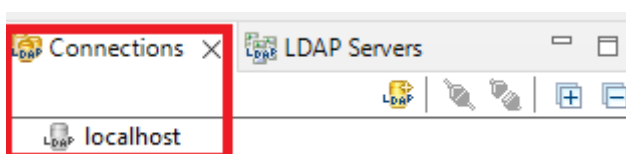
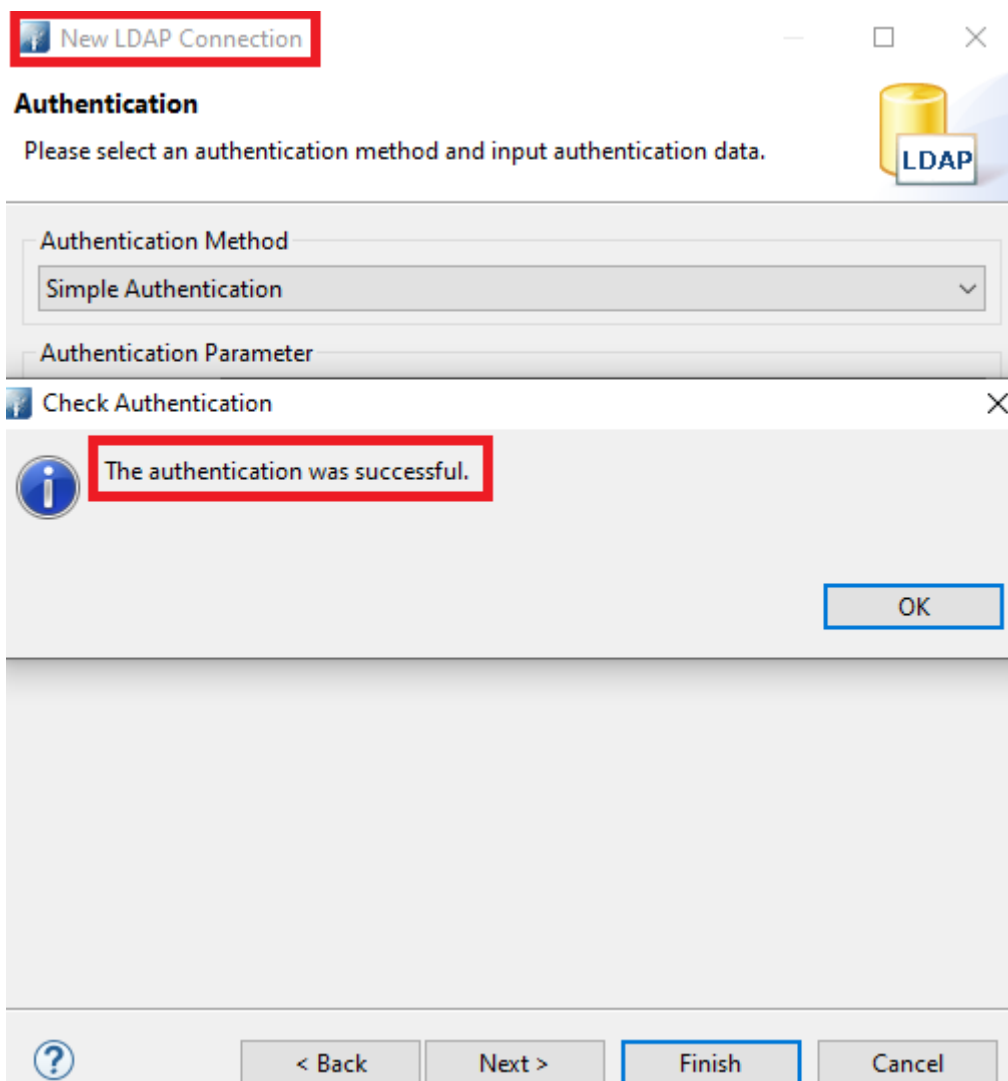
The authentication was successful.

OK

? < Back Next > Finish Cancel

Bind DN or user: uid=admin,ou=system

Bind password: secret



Para comprobar que la conexión LDAP funciona correctamente, se realiza un test unitario (Test JUnit) a modo de prueba.

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left highlights the file `testConnectionLDAP.java` within the `sopaDeLetras` project. The main editor displays the code for `testConnectionLDAP.java`, which includes package declarations, imports, and a `@Test` method. The console at the bottom shows the output of the JUnit test, including the message "LDAP Test Connection" and "Connected".

```

1 package sopaDeLetras;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class testConnectionLDAP {
6
7     @Test
8     void test() {
9         System.out.println("LDAP Test Connection");
10
11         Hashtable<String, String> env = new Hashtable<String, String>();
12
13         env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
14         env.put(Context.PROVIDER_URL, "ldap://localhost:10389");
15         env.put(Context.SECURITY_AUTHENTICATION, "simple");
16         env.put(Context.SECURITY_PRINCIPAL, "uid=admin,ou=system");
17         env.put(Context.SECURITY_CREDENTIALS, "secret");
18
19         try {
20             DirContext context = new InitialDirContext(env);
21             System.out.println("Connected");
22             System.out.println(context.getEnvironment());
23             context.close();
24         } catch (AuthenticationNotSupportedException e) {
25             System.out.println("The authentication is not supported by the");
26         }
27     }
28 }


```

```


terminated> testConnectionLDAP [JUnit] C:\Users\thebinaries\workspace\proyectos\jdk-14\bin\javaw.exe (27 sept 2021 21:42:02 - 2
LDAP Test Connection
Connected
(java.naming.provider.url=ldap://localhost:10389, java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFa

```

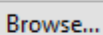
El siguiente paso que se realiza es **crear un Servlet**, o lo que es lo mismo, una clase en Java para ampliar las capacidades de un servidor. Entre otras funciones, la más común, es generar páginas web de forma dinámica a partir de una serie de parámetros que envía al navegador.

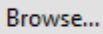
 **Create Servlet**

Create Servlet
Specify class file destination.

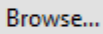


Project: maven-webapp

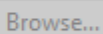
Source folder: /maven-webapp/src/main/java 


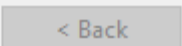
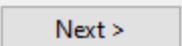
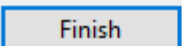
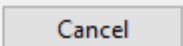
Java package: com.thebinaries.web.Ldaplogin.servlet 

Class name: LdapLogi

Superclass: javax.servlet.http.HttpServlet 

☐ Use an existing Servlet class or JSP

Class name: LdapLogi 

The screenshot shows the Eclipse IDE interface. The top toolbar includes menus like File, Edit, Navigate, Search, Project, Run, Window, and Help. The Project Explorer on the left shows a project named 'maven-webapp' with a sub-project 'com.thebinaries.web.LdapLogin'. The 'LdapLogin.java' file is selected. The main editor window displays the URL 'http://localhost:8080/maven-webapp/LdapLogin'. The console window at the bottom shows the following output:

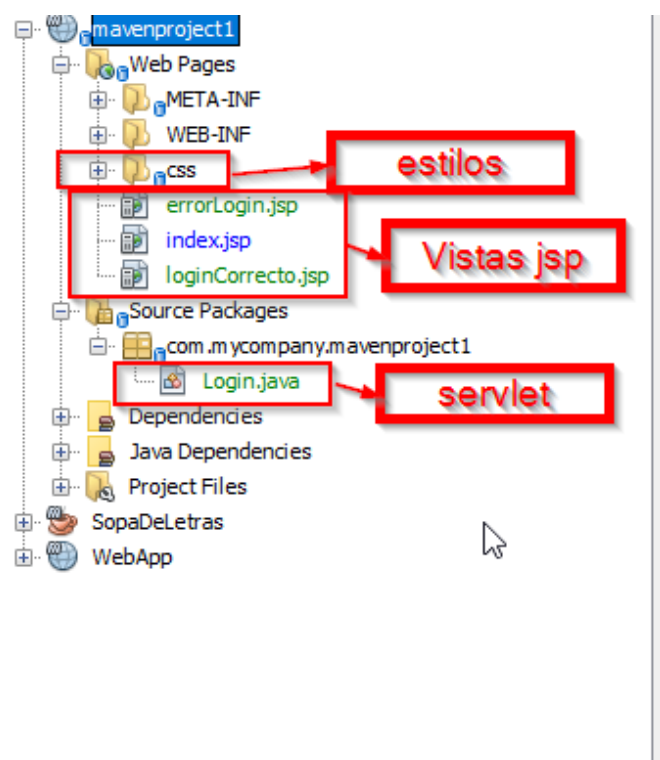
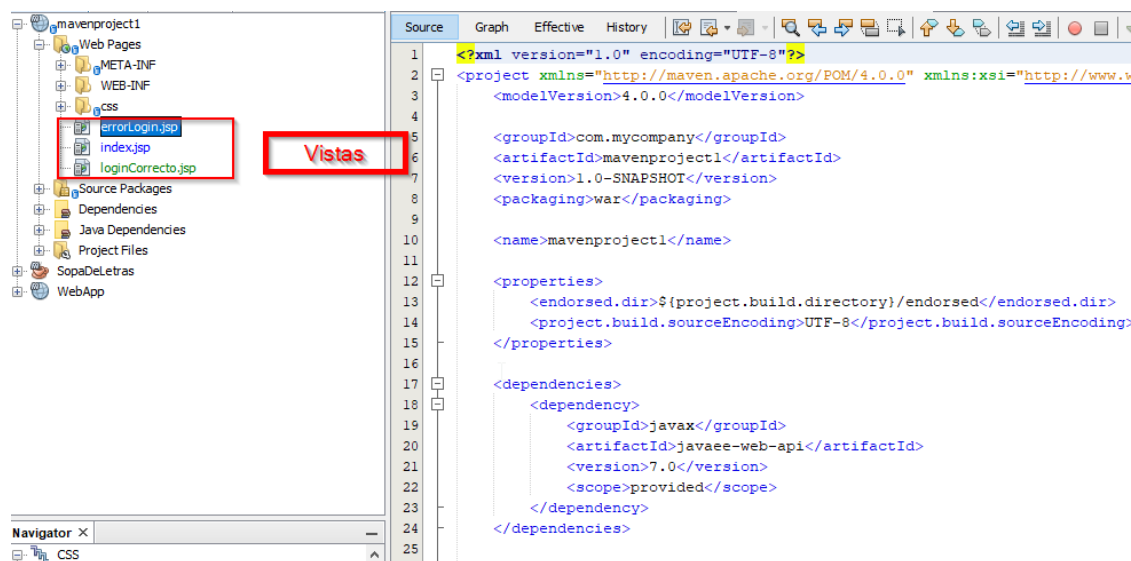
```

Served at: /maven-webapp
LDAP Test Connection
Connected
{java.naming.provider.url=ldap://localhost:10389,
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory,
java.naming.factory.url.pkgs=org.apache.naming,
java.naming.security.principal=uid=admin,ou=system,
java.naming.security.authentication=simple, java.naming.security.credentials=secret,
com.sun.jndi.ldap.domainname=}
  
```

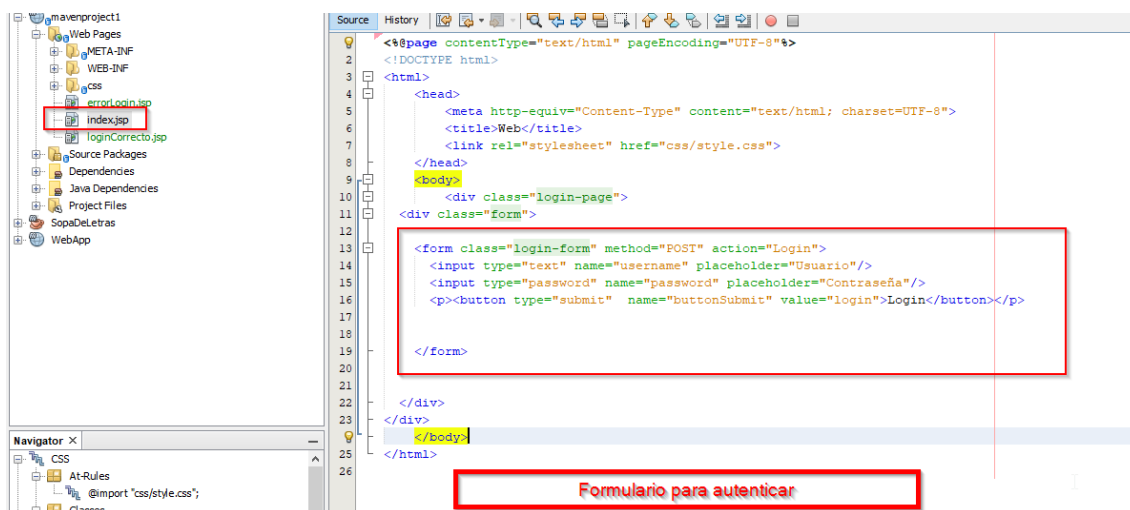
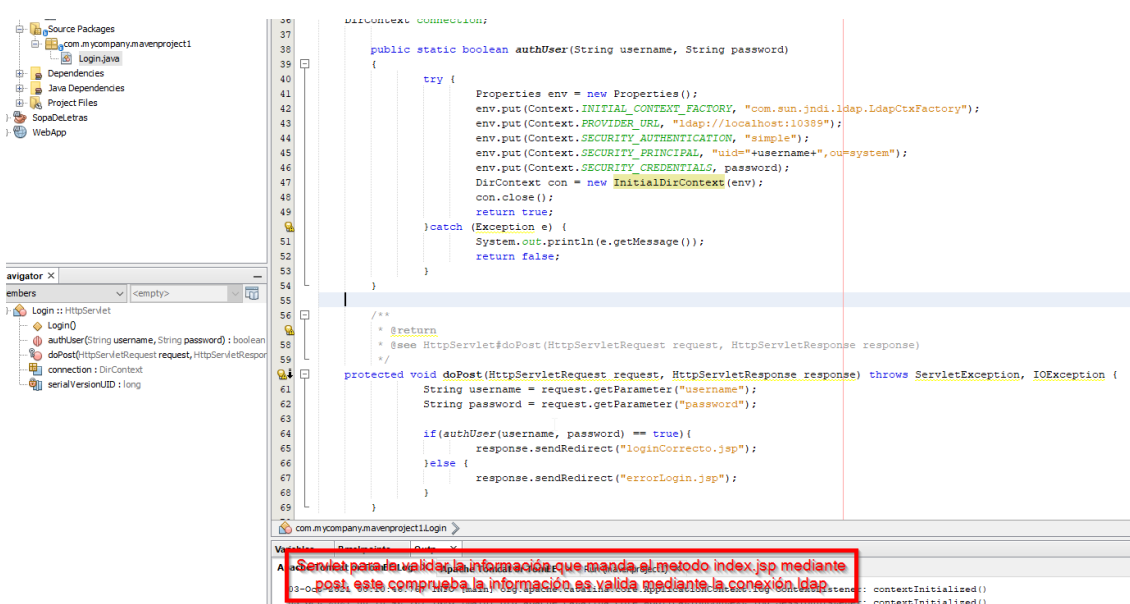
The console also shows Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\thebinaries\eclipse-workspace\proyectos\jdk-14\bin\javaw.exe (27 sept 2021 13:13:37) and a message: 'INFO: Ha comenzado la recarga de Contexto [/maven-webapp] sept. 27, 2021 1:19:37 P. M. org.apache.catalina.core.StandardContext reload'.

4.4 Desarrollar un programa que realice la autenticación de usuarios utilizando LDAP y Servlets.

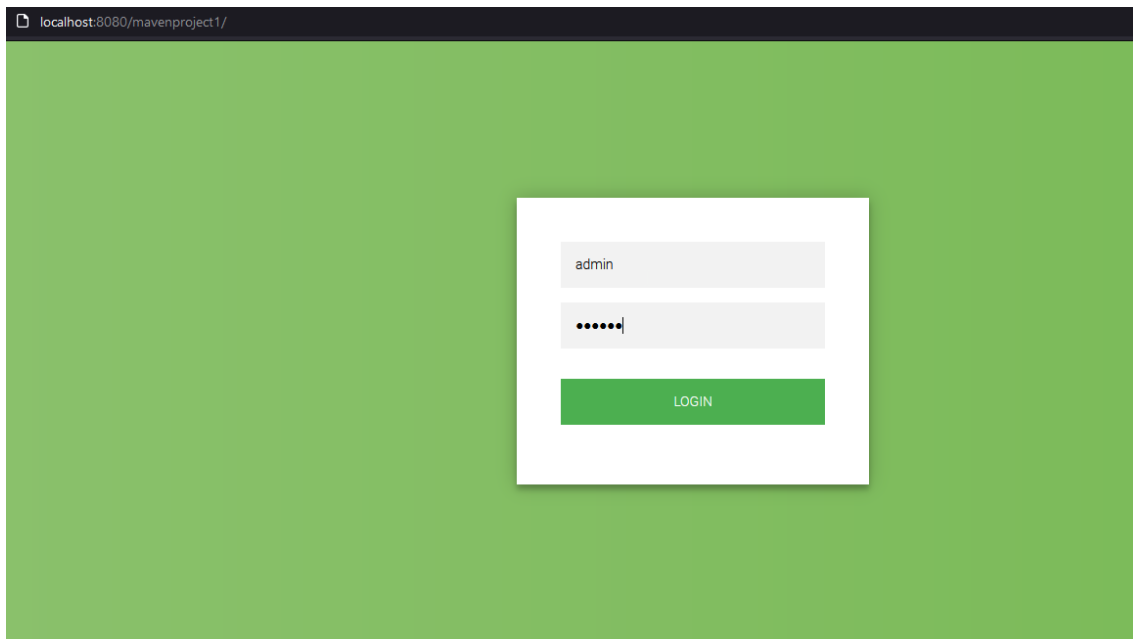
Creamos 3 archivos con contenido HTML para visualizar con Java (.jsp)



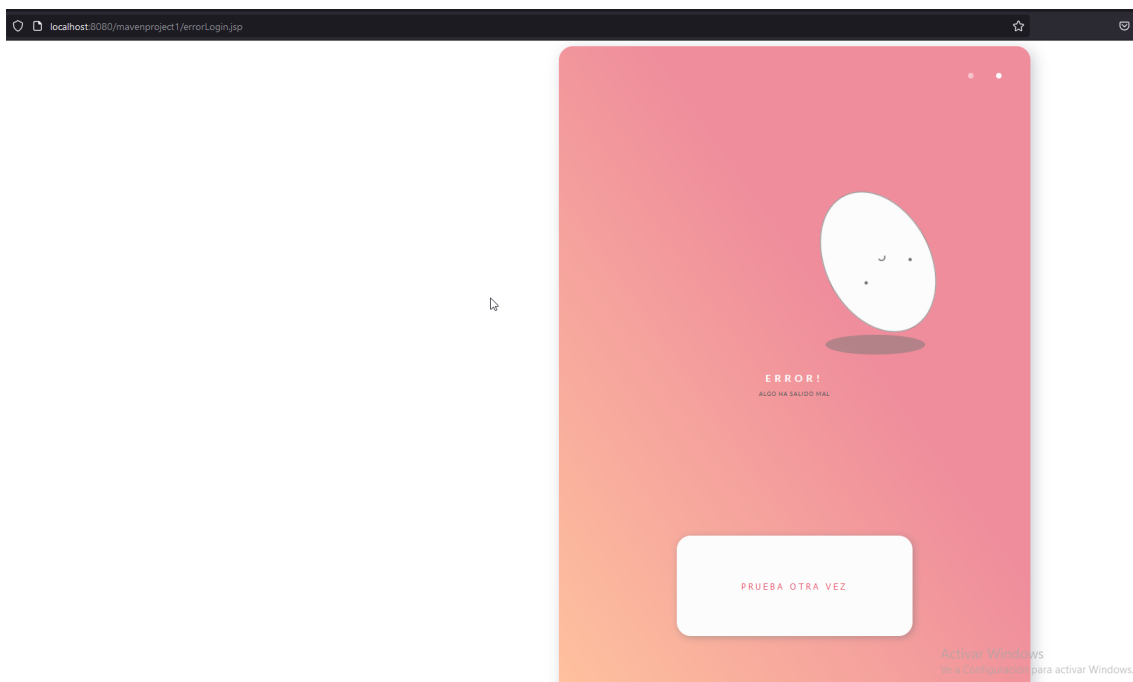
Luego para realizar la autenticación usamos nuestro Servlet



El resultado es el siguiente, si las credenciales son correctas o no te redirige, teniendo en cuenta los archivos creados y su contenido. En esta imagen podemos ver el login de inicio.



En el caso de que sea incorrecto aparecerá lo siguiente:



Y en el caso de que las credenciales sean correctas aparecerá el siguiente mensaje.



Link Github: <https://github.com/Oen844/Backend-P7-TheBinaries>

5. Referencias.

1. ALEX ROEL CODE: [Cómo Descargar e Instalar Java JDK 16 🖱️ en Windows 10 - 2021](#) [Consultada el 1 de Octubre del 2021].
2. ARSYS: [¿Por qué usar Tomcat sobre Servidores Cloud? - Blog de arsys.es](#) [Consultada el 29 de Septiembre del 2021].
3. DELGADO ALTAMIRANO Jorge E. et al. *Análisis comparativos de servidores de aplicaciones open SOURCE para la plataforma JAVA EE. caso práctico: módulo de gestión de juntas administradoras de agua potable y riego para la dirección provincial de Chimborazo de la Senagua*. Universidad Nacional del Chimborazo, 2017. URI: <http://dspace.unach.edu.ec/handle/51000/3397>
4. EDU4JAVA.COM: [Que es un Contenedor de Servlets. Instalación Apache Tomcat](#) [Consultada el 29 de Septiembre del 2021].
5. FACILCLOUD.COM: [Top 5 ventajas de Jetty web serverTech blog for developers](#) [Consultada el 1 de Octubre del 2021].
6. FUNDESEM.ES: [Java EE y el desarrollo web: Un enfoque de aprendizaje](#) [Consultada el 29 de Septiembre del 2021].
7. IBM: [IBM Servidores de Aplicaciones](#) [Consultada el 29 de Septiembre del 2021].
8. IONOS: [Runtime environment: ¿qué es y cómo funciona? Explicación y ejemplos](#) [Consultada el 29 de Septiembre del 2021].
9. MATIXA: [Instalar y configurar Apache Tomcat 9 en NetBeans](#) [Consultada el 1 de Octubre del 2021].
10. PÍLDORAS INFORMÁTICAS: [Curso Java. Servlets I. Vídeo 246](#) [Consultada el 29 de Septiembre del 2021].
11. QSTACK.MX: [¿Cuál es la diferencia entre el servidor de aplicaciones y el servidor web?](#) [Consultada el 1 de Octubre del 2021].
12. SIN FLOO: [Sistema de Ventas en Java Web - Patron MVC y MySQL - Parte Login](#) [Consultada el 1 de Octubre del 2021].

13. WIKIPEDIA:

- [GlassFish](#) [Consultada el 1 de Octubre del 2021].
- [Java EE](#) [Consultada el 29 de Septiembre del 2021].
- [Programación por capas](#) [Consultada el 30 de Septiembre del 2021].
- [Tomcat](#) [Consultada el 1 de Octubre del 2021].