

MANUAL DE DESPLIEGUE

Práctica Final – Arquitectura y diseño de sistemas web y c/s

Grupo M2

Tabla de contenido

REQUISITOS PREVIOS.....	2
1.1 Instalación y configuración de Tomcat	2
1.2 Instalación de MySQLWorkbench.....	7
 CREACIÓN DE LA BASE DE DATOS.....	 9
 DESPLIEGUE DE LA APLICACIÓN EN EL SERVIDOR.....	 10

La aplicación funciona con 2 componentes fundamentales que son Tomcat y MYSQL. Durante el desarrollo de la aplicación se hizo uso de la herramienta git para el manejo de cambios de la aplicación.

REQUISITOS PREVIOS

Para el despliegue de la aplicación es necesario tener instalado Tomcat y MYSQLWorkbench. A continuación se procederá a la explicación de los procesos de instalación de ambos.

1.1 Instalación y configuración de Tomcat

El servidor Apache-Tomcat es uno de los servidores de aplicaciones más utilizados.

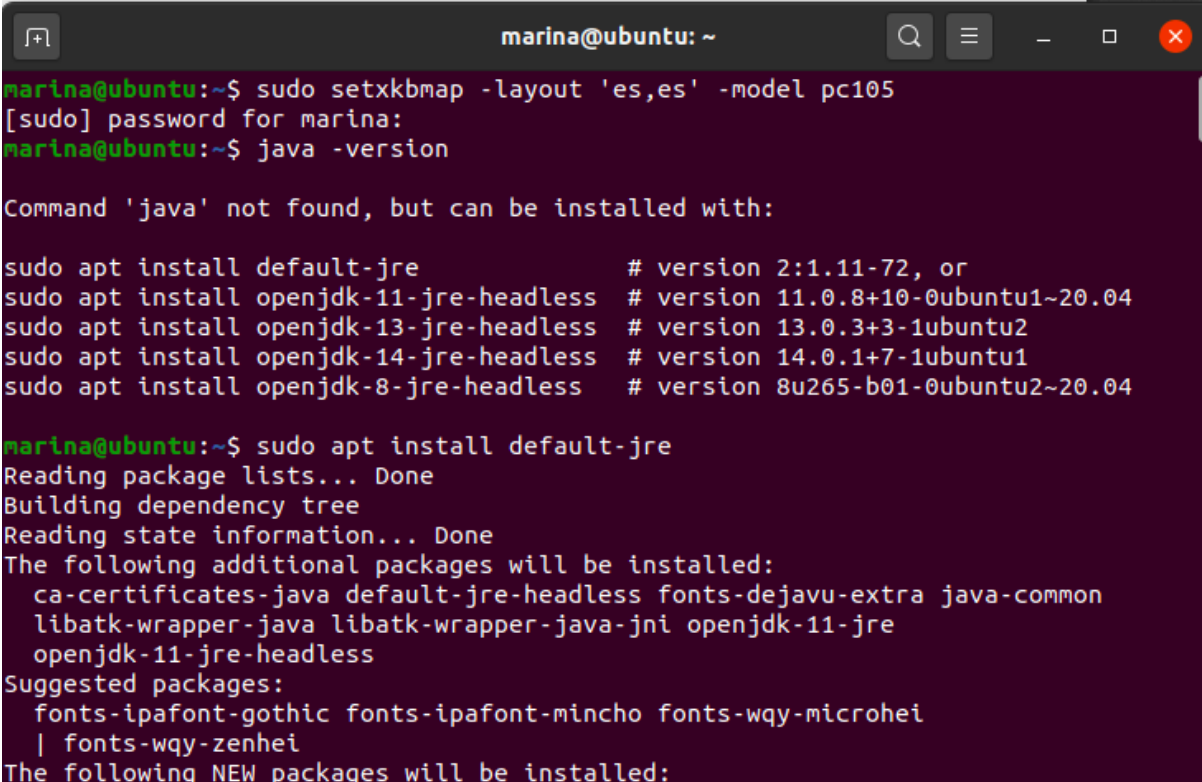
Para la instalación y configuración de Tomcat es necesario que la máquina en la que se va a llevar a cabo disponga de la máquina virtual Java.

El primer paso es comprobar si esta está instalada mediante el comando '`java -version`' y en caso negativo la instalación tanto del JRE como del JDK.

Para la instalación de estos mediante la línea de comandos utilizamos:

JRE: `sudo apt install default-jre`

JDK: `sudo apt install default-jdk`

A terminal window titled 'marina@ubuntu: ~' with standard Ubuntu window controls. The user runs 'sudo setxkbmap -layout 'es,es' -model pc105', enters their password, and then runs 'java -version'. The output shows 'Command 'java' not found, but can be installed with:' followed by a list of available Java packages and their versions. The user then runs 'sudo apt install default-jre'. The output shows the package lists being read, the dependency tree being built, and the state information being read. It then lists the additional packages to be installed: 'ca-certificates-java', 'default-jre-headless', 'fonts-dejavu-extra', 'java-common', 'libatk-wrapper-java', 'libatk-wrapper-java-jni', and 'openjdk-11-jre-headless'. It also suggests some font packages. Finally, it states that the following NEW packages will be installed.

```
marina@ubuntu:~$ sudo setxkbmap -layout 'es,es' -model pc105
[sudo] password for marina:
marina@ubuntu:~$ java -version

Command 'java' not found, but can be installed with:

sudo apt install default-jre                # version 2:1.11-72, or
sudo apt install openjdk-11-jre-headless    # version 11.0.8+10-0ubuntu1~20.04
sudo apt install openjdk-13-jre-headless    # version 13.0.3+3-1ubuntu2
sudo apt install openjdk-14-jre-headless    # version 14.0.1+7-1ubuntu1
sudo apt install openjdk-8-jre-headless     # version 8u265-b01-0ubuntu2~20.04

marina@ubuntu:~$ sudo apt install default-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre-headless
  openjdk-11-jre-headless
Suggested packages:
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
```

```

marina@ubuntu:~$ javac -version
javac 11.0.8
marina@ubuntu:~$ java -version
openjdk version "11.0.8" 2020-07-14
OpenJDK Runtime Environment (build 11.0.8+10-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.8+10-post-Ubuntu-0ubuntu120.04, mixed mode,
sharing)

```

Una vez comprobado que ambos han sido correctamente instalados procedemos a la instalación de Apache TomCat:

El primer paso es crear un nuevo grupo y un usuario tomcat miembro de ese grupo.

```

marina@ubuntu:~$ sudo groupadd tomcat
[sudo] password for marina:
marina@ubuntu:~$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
marina@ubuntu:~$ cd /tmp

```

A continuación descargamos Tomcat usando curl:

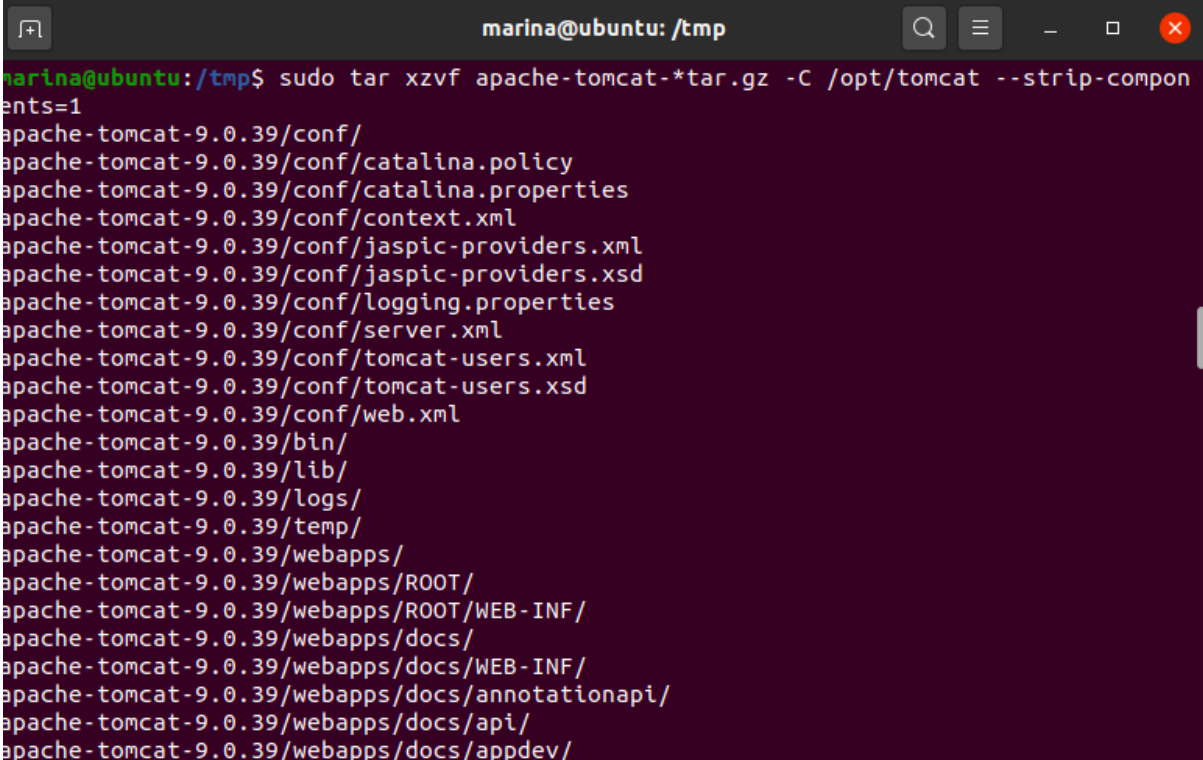
curl -O <https://downloads.apache.org/tomcat/tomcat-9/v9.0.39/bin/apache-tomcat-9.0.39.tar.gz>

```

marina@ubuntu:/tmp$ curl -O https://downloads.apache.org/tomcat/tomcat-9/v9.0.39/bin/apache-tomcat-9.0.39.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 10.7M  100 10.7M    0     0  3866k      0  0:00:02  0:00:02 --:--:-- 3864k

```

Creamos un directorio y extraemos el archivo:



```

marina@ubuntu:/tmp$ sudo tar xzvf apache-tomcat-*.tar.gz -C /opt/tomcat --strip-components=1
apache-tomcat-9.0.39/conf/
apache-tomcat-9.0.39/conf/catalina.policy
apache-tomcat-9.0.39/conf/catalina.properties
apache-tomcat-9.0.39/conf/context.xml
apache-tomcat-9.0.39/conf/jaspic-providers.xml
apache-tomcat-9.0.39/conf/jaspic-providers.xsd
apache-tomcat-9.0.39/conf/logging.properties
apache-tomcat-9.0.39/conf/server.xml
apache-tomcat-9.0.39/conf/tomcat-users.xml
apache-tomcat-9.0.39/conf/tomcat-users.xsd
apache-tomcat-9.0.39/conf/web.xml
apache-tomcat-9.0.39/bin/
apache-tomcat-9.0.39/lib/
apache-tomcat-9.0.39/logs/
apache-tomcat-9.0.39/temp/
apache-tomcat-9.0.39/webapps/
apache-tomcat-9.0.39/webapps/ROOT/
apache-tomcat-9.0.39/webapps/ROOT/WEB-INF/
apache-tomcat-9.0.39/webapps/docs/
apache-tomcat-9.0.39/webapps/docs/WEB-INF/
apache-tomcat-9.0.39/webapps/docs/annotationapi/
apache-tomcat-9.0.39/webapps/docs/api/
apache-tomcat-9.0.39/webapps/docs/appdev/

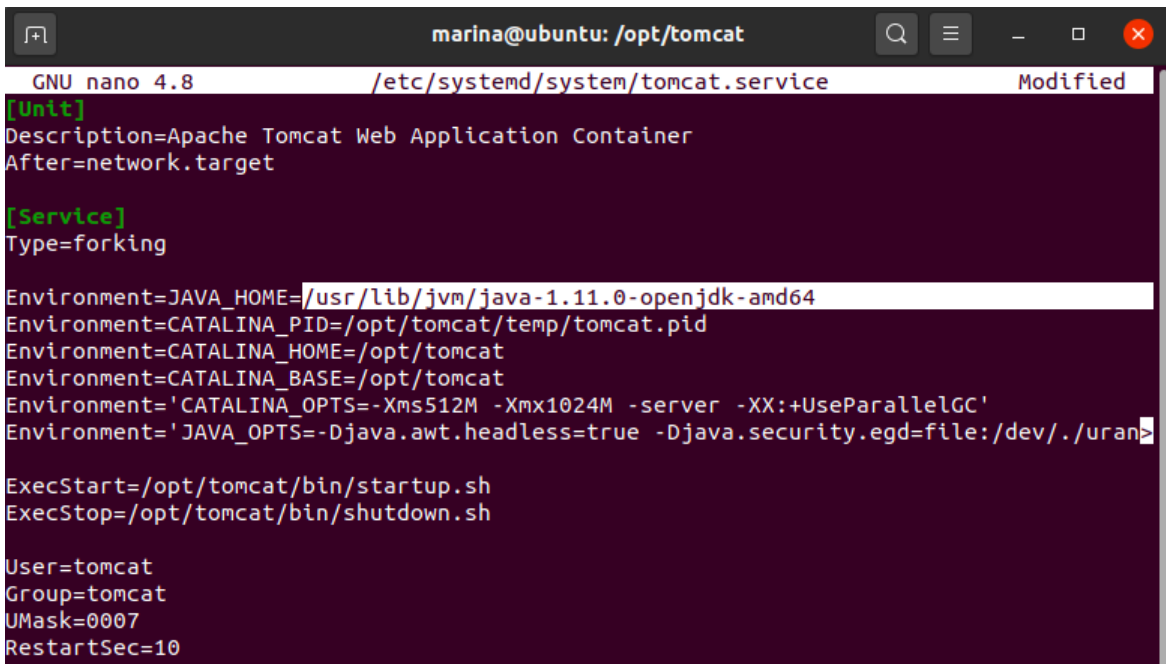
```

Configuramos el directorio que hemos creado para la instalación de tal forma que el usuario tomcat tenga permisos de lectura y ejecución sobre este y que sea el propietario de los directorios work, webapps, temp y logs utilizando los comandos *chmod* y *chown* respectivamente.

```
marina@ubuntu:/tmp$ cd /opt/tomcat
marina@ubuntu:/opt/tomcat$ sudo chgrp -R tomcat /opt/tomcat
[sudo] password for marina:
marina@ubuntu:/opt/tomcat$ sudo chmod -R g+r conf
marina@ubuntu:/opt/tomcat$ sudo chmod g+x conf
marina@ubuntu:/opt/tomcat$ sudo chown -R tomcat webapps/ work/ temp/ logs/
```

Para ejecutarse como servicio, Tomcat necesita saber la ruta en la que hemos instalado Java anteriormente. Obtenemos la ruta de Java gracias al comando *update-java-alternatives -l* y creamos un archivo tomcat.service introduciendo esta ruta en el parámetro JAVA_HOME:

```
marina@ubuntu:/opt/tomcat$ sudo update-java-alternatives -l
[sudo] password for marina:
java-1.11.0-openjdk-amd64      1111      /usr/lib/jvm/java-1.11.0-openjdk-amd64
marina@ubuntu:/opt/tomcat$ sudo nano /etc/systemd/system/tomcat.service
```



```
marina@ubuntu: /opt/tomcat
GNU nano 4.8 /etc/systemd/system/tomcat.service Modified
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
```

Una vez guardado los cambios y cerrado el archivo tomcat.service reiniciamos systemctl e iniciamos el servicio Tomcat usando *sudo systemctl daemon-reload* y *sudo systemctl start tomcat* como se muestra a continuación, comprobando que todo ha funcionado según lo esperado con el comando *sudo systemctl status tomcat*.

Para el correcto funcionamiento de Apache Tomcat es necesario configurar el Firewall para que permita el acceso a los puertos utilizados por defecto. Para ello ejecutamos `sudo ufw enable` para activar el Firewall y `sudo ufw allow` especificando la configuración deseada:

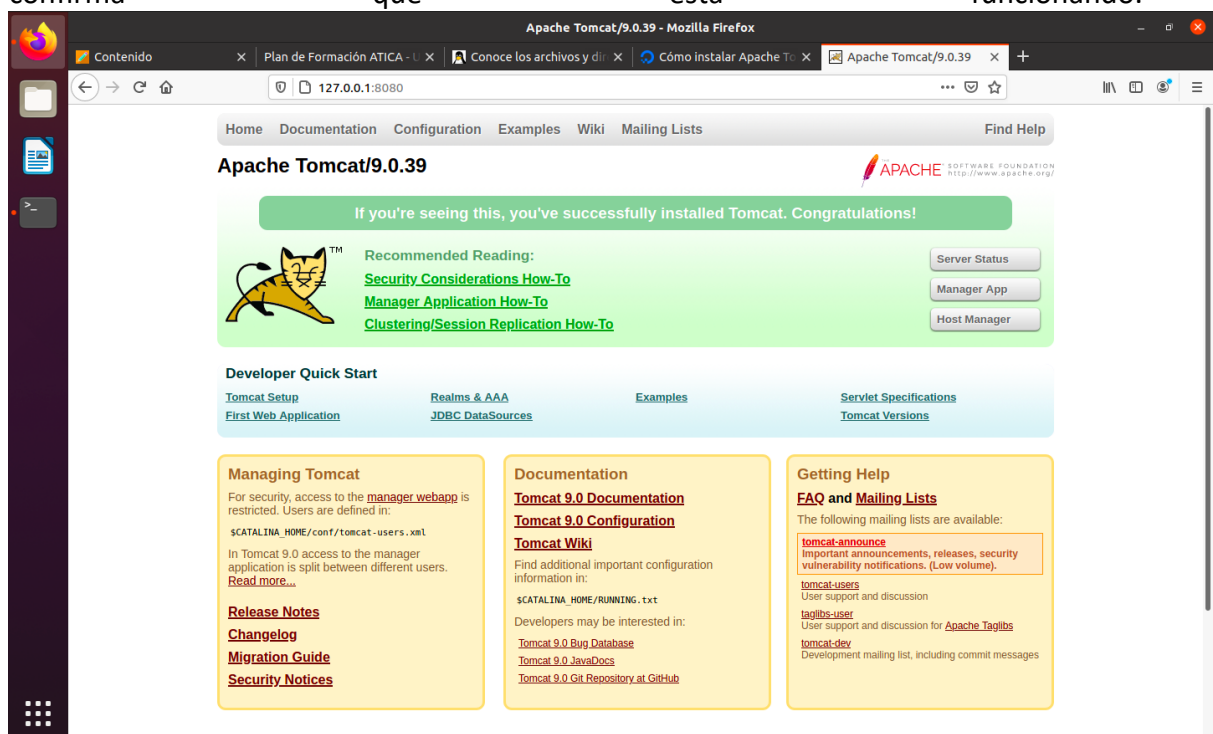
```
marina@ubuntu:~$ sudo ufw allow 8080
[sudo] password for marina:
Rule added
Rule added (v6)
```

Una vez activado el Firewall podemos comprobar el tráfico HTTP permitido:

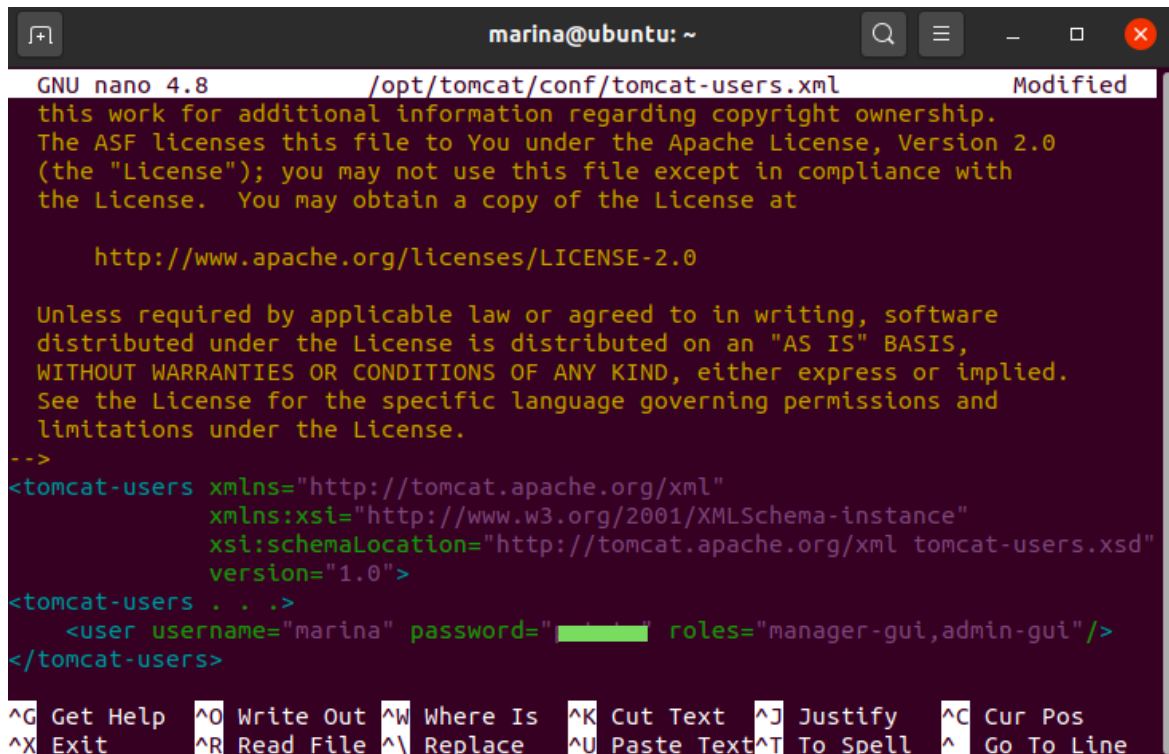
```
marina@ubuntu:~$ sudo ufw status
Status: active

To Action From
--
Apache 8080 ALLOW Anywhere
Apache (v6) 8080 (v6) ALLOW Anywhere (v6)
```

Para comprobar que se ha instalado y configurado correctamente se accede a `http://127.0.0.1:8080` desde el navegador. La página de Apache Tomcat por defecto nos confirma que está funcionando.



Una vez se ha instalado correctamente añadimos una cuenta para el inicio de sesión en el servidor Tomcat editando el archivo tomcat-users.xml. Añadimos un usuario con acceso a manager-gui y admin-gui:



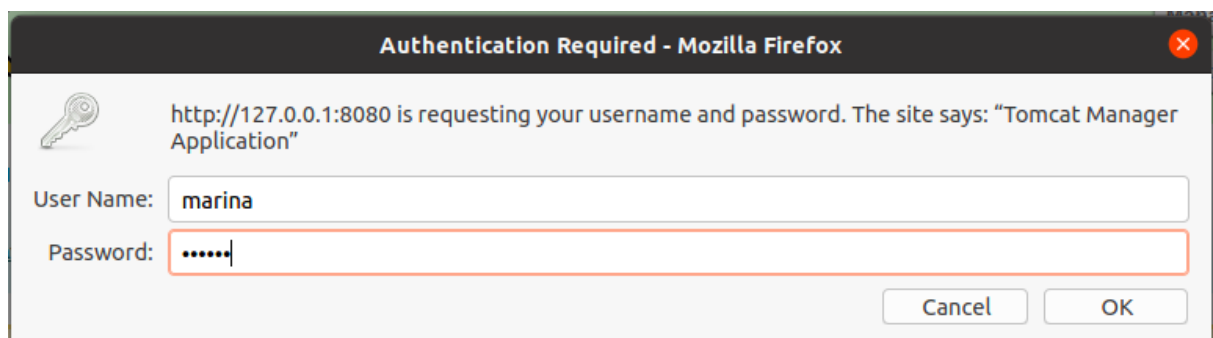
```
GNU nano 4.8 /opt/tomcat/conf/tomcat-users.xml Modified
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
<tomcat-users . . .>
  <user username="marina" password="marina" roles="manager-gui,admin-gui"/>
</tomcat-users>
```

Después de realizar estos cambios y para que sean efectivos es necesario reiniciar Tomcat *sudo systemctl restart tomcat*

Tras el reinicio comprobamos que podemos acceder a la ventana de application manager con los credenciales establecidos anteriormente desde *http://127.0.0.1:8080/manager/html*



Tomcat Web Application Manager

Message: OK

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle >= 30 minutes

Deploy

1.2 Instalación de MySQLWorkbench

MySQL Workbench es una aplicación que nos permite gestionar bases de datos MySQL y ejecutar consultas sobre ellas. Puedes descargar MySQL Workbench desde este enlace: <https://dev.mysql.com/downloads/workbench/>

Selecciona tu sistema operativo y haz clic en **download**.

General Availability (GA) Releases **Archives**

Select Operating System...

- Microsoft Windows
- Ubuntu Linux
- Red Hat Enterprise Linux / Oracle Linux
- Fedora
- ✓ macOS
- Source Code

! Packages for Catalina (10.15) are compatible with Big Sur (11)

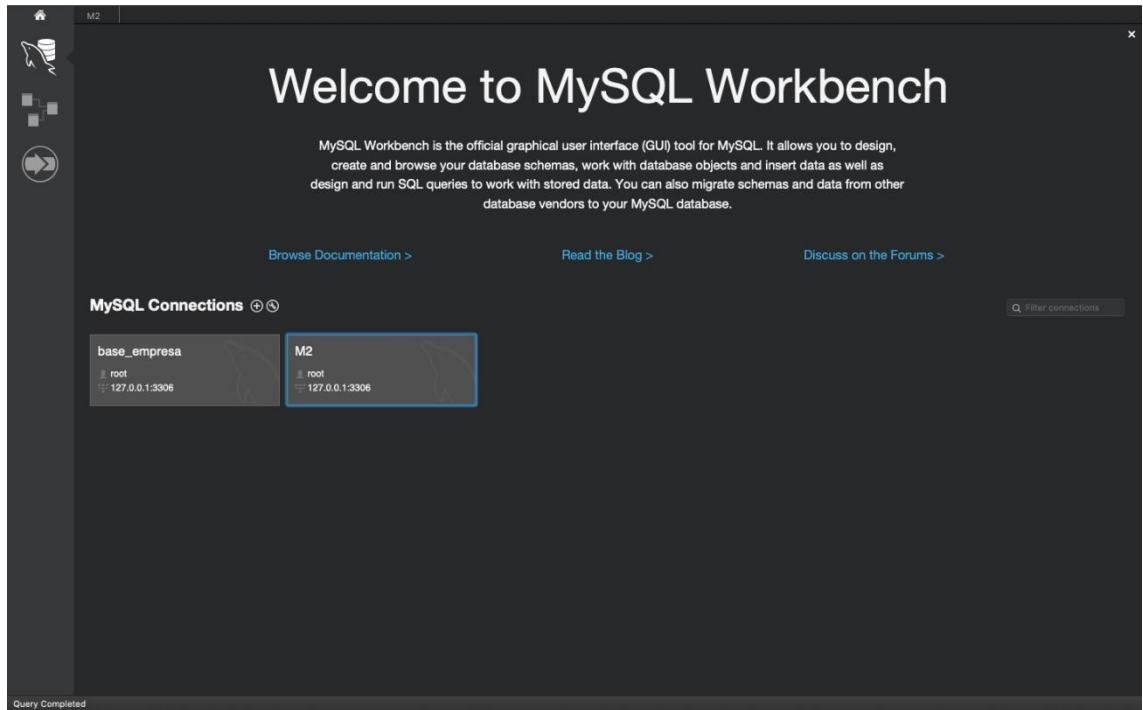
Platform	Version	Size	Action
macOS (x86, 64-bit), DMG Archive	8.0.23	110.8M	Download

(mysql-workbench-community-8.0.23-macos-x86_64.dmg) MD5: d9b8dd4ee928813a716f38aaeb9655ca | [Signature](#)

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

Una vez descargado ejecuta el instalador y sigue los pasos indicados. Tras finalizar la instalación hace falta configurar la conexión a una base de datos:

1. Inicia MySQL Workbench
2. Desde la página inicial o espacio de trabajo haz clic en el icono con un más (+), justo al lado de donde pones MySQL Connections. Se mostrará el **gestor de conexiones**.

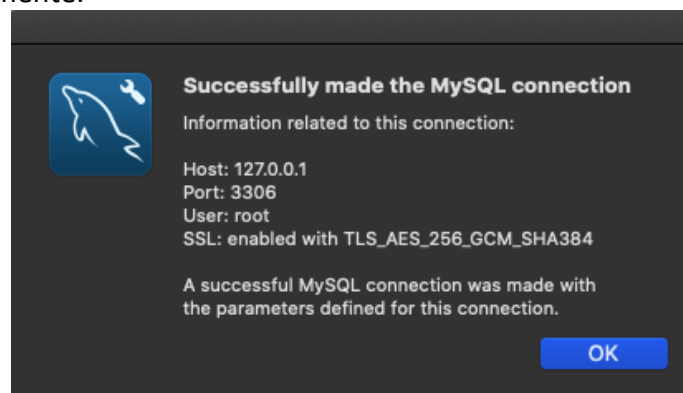


3. En el campo reservado para el **nombre de la conexión** (*Connection Name*), escribe el nombre de la misma. En este caso se debe llamar “M2”.
4. En la lista de **métodos de conexión** (*Connection Method*), selecciona Standard (TCP/IP).
5. Seguidamente, en la pestaña de parámetros, podrás ver un campo para el nombre del host (*Hostname*). Debe escribir *localhost*.
6. Deberás seleccionar también el **puerto de acceso MySQL** del servidor, que por defecto es **3306**.
7. En el campo para el **nombre de usuario** (*Username*), escribe el nombre del usuario con acceso a la base de datos, cuyos permisos debes haber asignado previamente en el instalador. Recomendamos usuario: root contraseña: admin.

8. Si ya previamente se ha instalado MySQL y el usuario es diferente a usuario: root contraseña: admin y/o el puerto no es el 3306, se procederá a cambiar estos parámetros en la clase ConexionBd.java como se muestra a continuación:

```
public class ConexionBd {  
  
    private static Connection conexion = null;  
  
    public static Connection getConexion() {  
        String driver = "com.mysql.cj.jdbc.Driver";  
        String nombreBd = "M2";  
        String puerto = "3306";  
        String user = "root";  
        String password = "admin";  
        String url = "jdbc:mysql://localhost:"+ puerto + "/" + nombreBd + "?useTimezone=true&serverTimezone=UTC";  
    }  
}
```

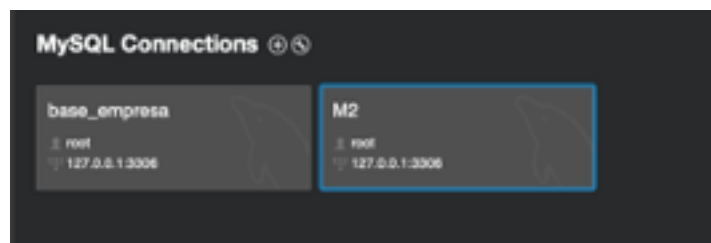
9. Haciendo click en 'Test connection' comprobamos si todo se ha configurado correctamente.



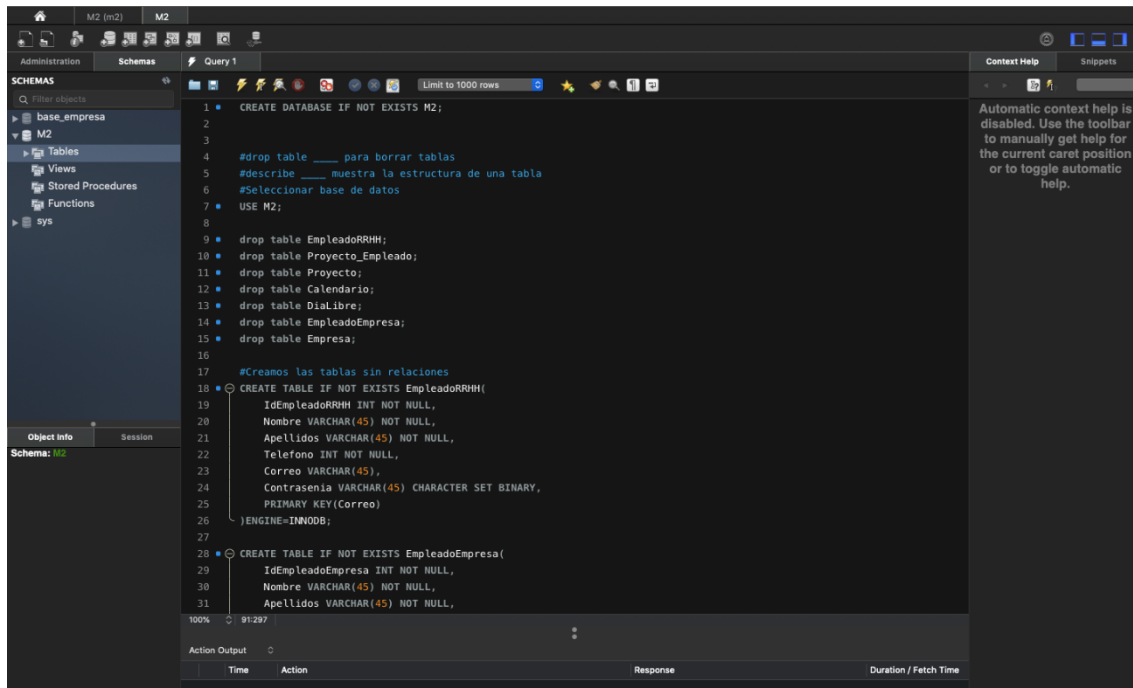
CREACIÓN DE LA BASE DE DATOS

Desde la conexión añadida en el apartado anterior debemos cargar las tablas necesarias para el correcto funcionamiento de la aplicación.

Abrimos la conexión haciendo doble click sobre aquella a la que queremos acceder, en este caso la marcada en azul en la imagen.



Una vez aquí una opción es copiar el archivo BaseEmpresa.sql en la herramienta para queries de MySQLWorkbench y ejecutarlo tal y como se muestra en la captura inferior:



La base de datos ya estaría lista para usarse.

DESPLIEGUE DE LA APLICACIÓN EN EL SERVIDOR

Accedemos al manager de Tomcat configurado con anterioridad desde <http://127.0.0.1:8080/manager/html>

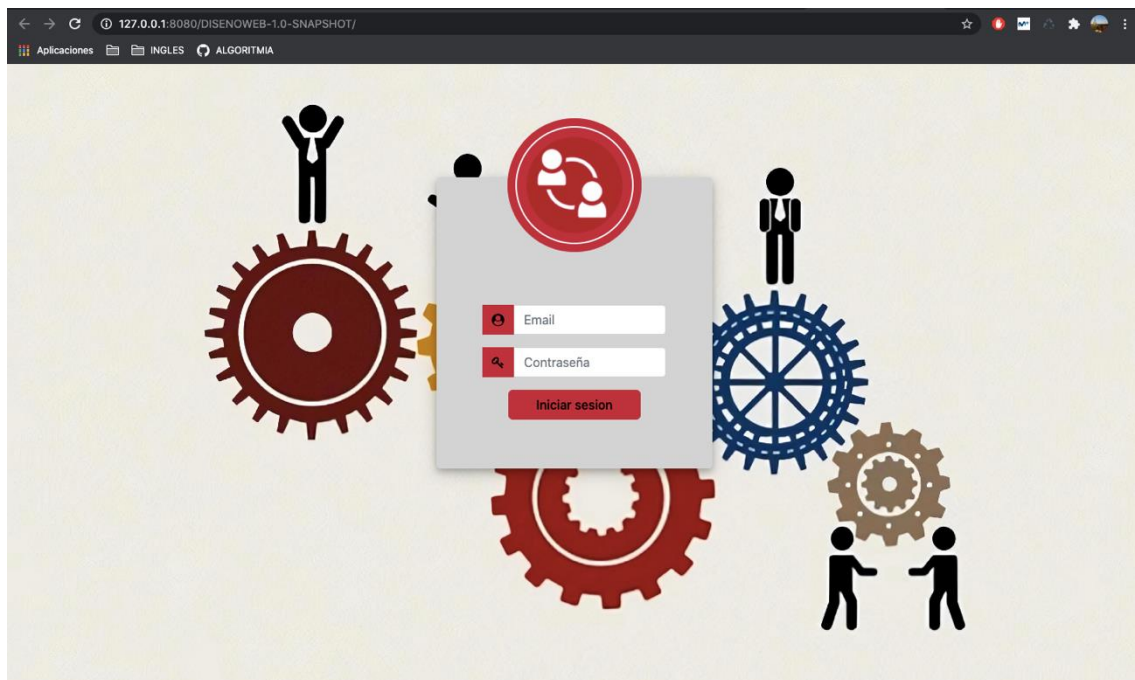
En el apartado 'Desplegar' debemos seleccionar el archivo .war proporcionado DISENOWEB-1.0-SNAPSHOT.war y pinchar en desplegar.

Desplegar	
Desplegar directorio o archivo WAR localizado en servidor	
Trayectoria de Contexto (opcional): <input type="text"/> Version (for parallel deployment): <input type="text"/> URL de archivo de Configuración XML: <input type="text"/> URL de WAR o Directorio: <input type="text"/>	
Desplegar	
Archivo WAR a desplegar	
Seleccione archivo WAR a cargar: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado <div style="text-align: center;">Desplegar</div>	

Una vez listo nos aparecerá lo siguiente:

Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Ingenieros a peso	true	0	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/DISENOWEB-1.0-SNAPSHOT	Ninguno especificado	Ingenieros a peso	true	0	<div>Arrancar Parar Recargar Replegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>

Haciendo click sobre la ruta accedemos a la página principal de la aplicación.



Algunos de los usuarios disponibles para poder iniciar sesión son:

Empleados RRHH

- Usuario: carmenlozano@correo.com – Contraseña: 5678
- Usuario: emilianallanos@correo.com – Contraseña: 5678

Empleados empresa

- Usuario: guillermoplata@correo.com – Contraseña: 1234
- Usuario: angelagoncavales@correo.com – Contraseña: 1234

Para probar más usuarios, puede consultar la base de datos.