

Diseño de sistemas distribuidos



Fecha:02/12/2020

Tarea 7. Implementación de un servicio web estilo REST

Alumno:

- Victor Hugo Magaña Bautista

Maestro: Carlos Pineda Guerrero

Group: 4CV2

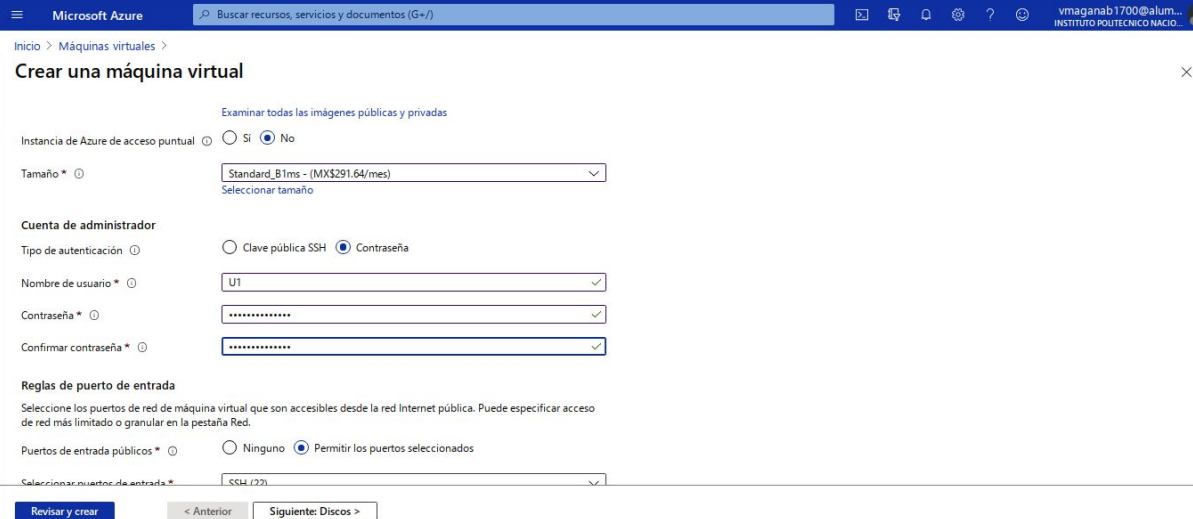
Escuela Superior de Cómputo



Instalación de Tomcat con soporte REST

1. Crear una máquina virtual con Ubuntu 18 con al menos 1GB de memoria RAM. Abrir el puerto 8080 para el protocolo TCP.

Creación de la máquina virtual con Ubuntu



Microsoft Azure

Inicio > Máquinas virtuales >

Crear una máquina virtual

Examinar todas las imágenes públicas y privadas

Instancia de Azure de acceso puntual ☐ Sí ☒ No

Tamaño *
Seleccionar tamaño

Cuenta de administrador

Tipo de autenticación ☐ Clave pública SSH ☒ Contraseña

Nombre de usuario *

Contraseña *

Confirmar contraseña *

Reglas de puerto de entrada

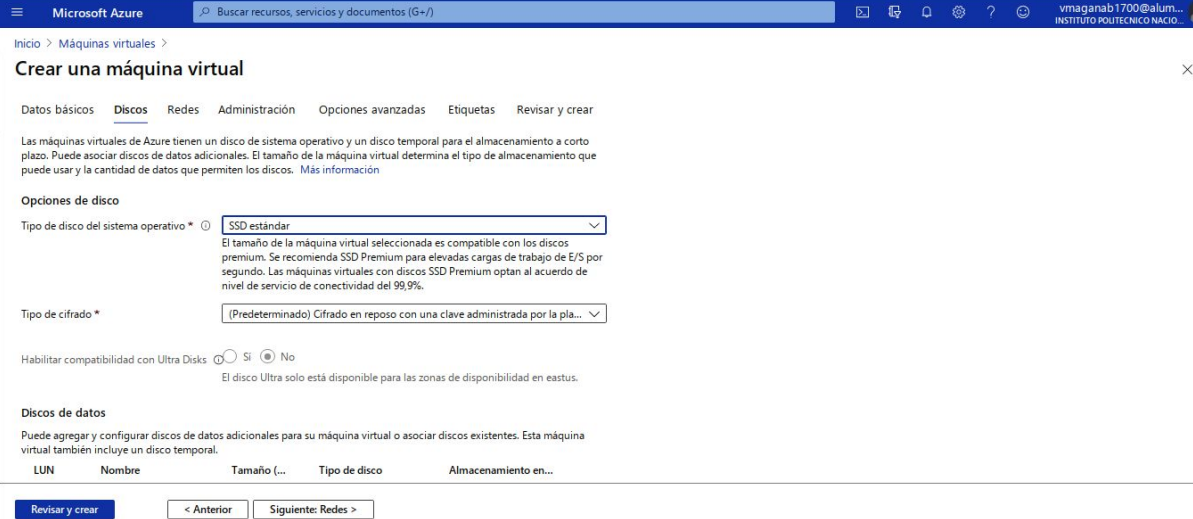
Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos * ☐ Ninguno ☒ Permitir los puertos seleccionados

Selección de puertos de entrada *

[Revisar y crear](#) [< Anterior](#) [Siguiente: Discos >](#)

Figure 1: Creación de la máquina virtual Ubuntu12, sección: Datos básicos.



Microsoft Azure

Inicio > Máquinas virtuales >

Crear una máquina virtual

Datos básicos **Discos** Redes Administración Opciones avanzadas Etiquetas Revisar y crear

Las máquinas virtuales de Azure tienen un disco de sistema operativo y un disco temporal para el almacenamiento a corto plazo. Puede asociar discos de datos adicionales. El tamaño de la máquina virtual determina el tipo de almacenamiento que puede usar y la cantidad de datos que permiten los discos. [Más información](#)

Opciones de disco

Tipo de disco del sistema operativo *
El tamaño de la máquina virtual seleccionada es compatible con los discos premium. Se recomienda SSD Premium para elevadas cargas de trabajo de E/S por segundo. Las máquinas virtuales con discos SSD Premium optan al acuerdo de nivel de servicio de conectividad del 99,9%.

Tipo de cifrado *

Habilitar compatibilidad con Ultra Disks ☐ Sí ☒ No
El disco Ultra solo está disponible para las zonas de disponibilidad en eastus.

Discos de datos

Puede agregar y configurar discos de datos adicionales para su máquina virtual o asociar discos existentes. Esta máquina virtual también incluye un disco temporal.

LUN	Nombre	Tamaño (...)	Tipo de disco	Almacenamiento en...
-----	--------	--------------	---------------	----------------------

[Revisar y crear](#) [< Anterior](#) [Siguiente: Redes >](#)

Figure 2: Creación de la máquina virtual Ubuntu12, sección: Discos.

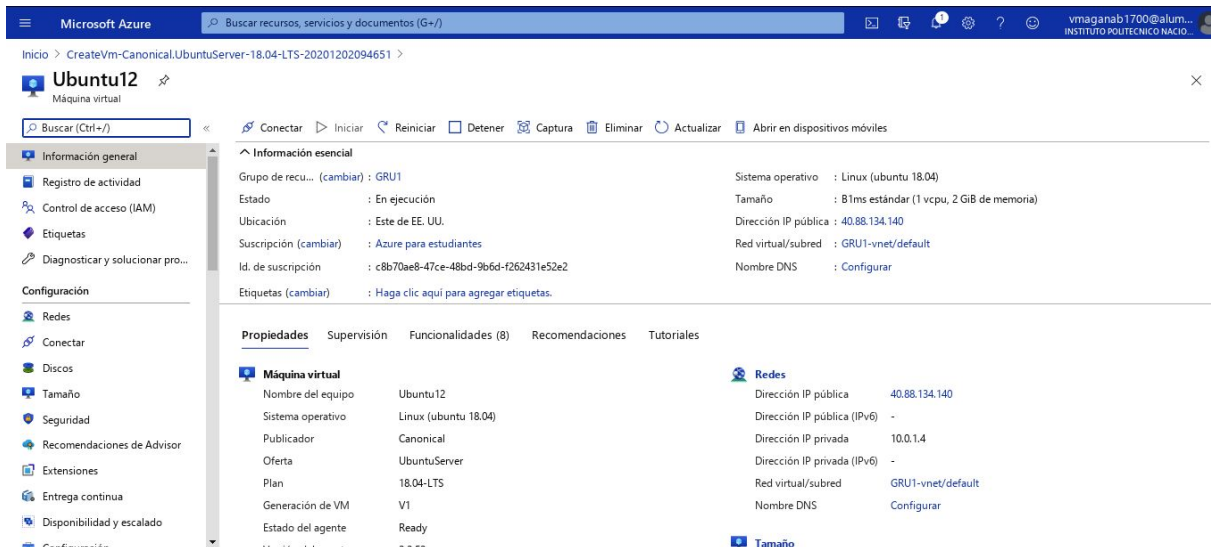


Figure 3: Máquina virtual Ubuntu12 en ejecución.

Abrir el puerto 8080 para el protocolo TCP.

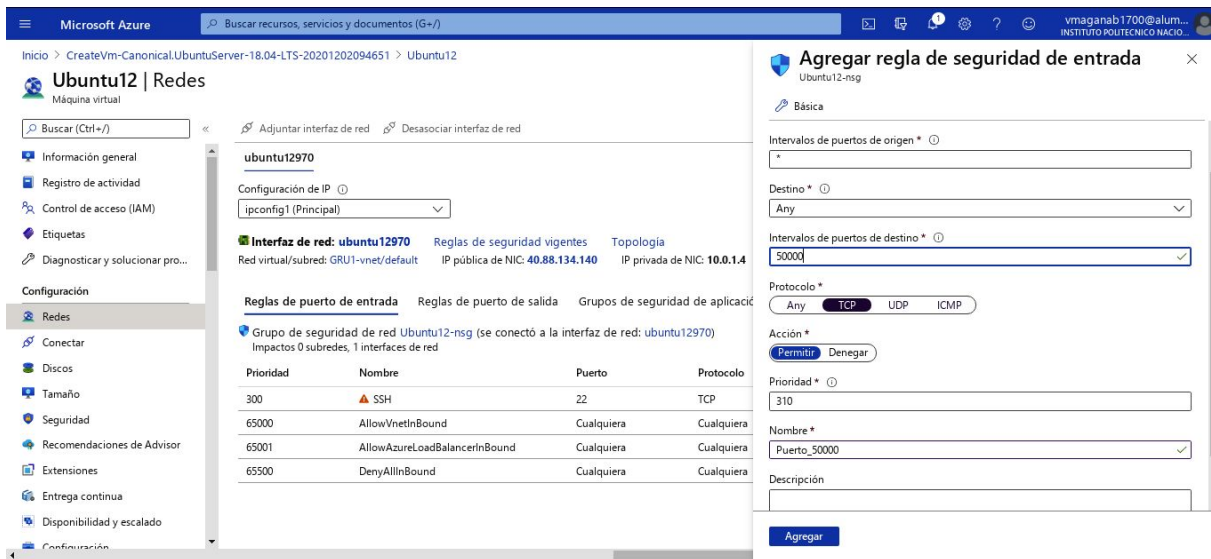


Figure 4: Apertura del puerto 50000 en la máquina virtual Ubuntu12

Conexión a la máquina virtual Ubuntu12

```
U1@Ubuntu12: ~  
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
  
hugo@debian:~$ ssh U1@40.88.134.140  
U1@40.88.134.140's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1032-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Wed Dec  2 16:26:09 UTC 2020  
  
System load:  0.0          Processes:      112  
Usage of /:   4.5% of 28.9GB Users logged in:  0  
Memory usage: 9%          IP address for eth0: 10.0.1.4  
Swap usage:   0%  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
U1@Ubuntu12:~$
```

Figure 5: Conexi3n mediante el comando: `ssh name_userQ@IP_publica`

2. Instalar JDK8 ejecutando los siguientes comandos en la m1quina virtual:

```
sudo apt update  
sudo apt install openjdk-8-jdk-headless
```

```
U1@Ubuntu12: ~  
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
  
U1@Ubuntu12:~$ sudo apt update  
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease  
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]  
Get:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]  
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]  
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]  
Get:7 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]  
Get:8 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]  
Get:9 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1789 kB]  
Get:10 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [376 kB]  
Get:11 http://azure.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [206 kB]  
Get:12 http://azure.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [28.0 kB]  
Get:13 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1695 kB]  
Get:14 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [357 kB]  
Get:15 http://azure.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [36.1 kB]  
Get:16 http://azure.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [7308 B]  
Get:17 http://azure.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [10.0 kB]  
Get:18 http://azure.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 B]  
Get:19 http://azure.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [10.3 kB]  
Get:20 http://azure.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4588 B]  
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [1453 kB]  
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [284 kB]  
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [184 kB]  
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [24.4 kB]  
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1091 kB]  
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [244 kB]  
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [13.3 kB]  
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2996 B]  
Fetched 21.8 MB in 5s (4143 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
4 packages can be upgraded. Run 'apt list --upgradable' to see them.  
U1@Ubuntu12:~$
```

Figure 6: Ejecuci3n del comando: `sudo apt update`

```
U1@Ubuntu12: ~
Archivo Editar Ver Buscar Terminal Ayuda
U1@Ubuntu12:~$ sudo apt install openjdk-8-jdk-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-headers-4.15.0-124
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2 libnspr4 libnss3
  libpcsc-lite libx11 libxrender1 libxtst6 openjdk-8-jdk-headless x11-common
Suggested packages:
  default-jre liblcms2-utils pcscd openjdk-8-demo openjdk-8-source libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2 libnspr4 libnss3
  libpcsc-lite libx11 libxrender1 libxtst6 openjdk-8-jdk-headless x11-common
0 upgraded, 17 newly installed, 0 to remove and 4 not upgraded.
Need to get 39.4 MB of archives.
After this operation, 153 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libjpeg-turbo8 amd64 1.5.2-0ubuntu5.18.04.4 [110 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 java-common all 0.68ubuntu1-18.04.1 [14.5 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 liblcms2-2 amd64 2.9-1ubuntu0.1 [139 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libjpeg8 amd64 8c-2ubuntu8 [2194 B]
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 fonts-dejavu-core all 2.37-1 [1041 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 fontconfig-config all 2.12.6-0ubuntu2 [55.8 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libfontconfig1 amd64 2.12.6-0ubuntu2 [137 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libnspr4 amd64 2:4.18-1ubuntu1 [112 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnss3 amd64 2:3.35-2ubuntu2.12 [1220 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libpcsc-lite amd64 1.8.23-1 [21.3 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libx11 amd64 2:1.7.9-1 [29.2 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libxrender1 amd64 1:0.9.10-1 [18.7 kB]
Get:13 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 x11-common all 1:7.7+19ubuntu7.1 [22.5 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libxtst6 amd64 2:1.2.3-1 [12.8 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 openjdk-8-jre-headless amd64 8u275-b01-0ubuntu1-18.04 [28.2 MB]
Get:16 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 ca-certificates-java all 20180516ubuntu1-18.04 [12.2 kB]
```

Figure 7: Ejecución del comando: `sudo apt install openjdk-8-jdk-headless`

3. Descargar la distribución binaria de Tomcat 8 de la siguiente URL (descargar la opción Core "zip"): <https://tomcat.apache.org/download-80.cgi>

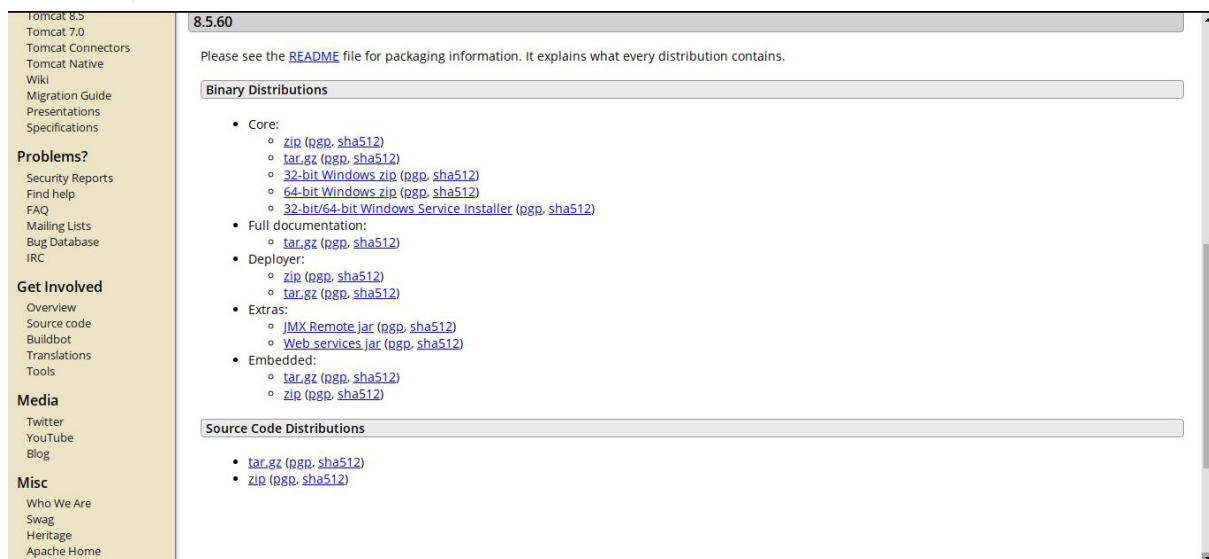


Figure 8: Selección de la distribución binaria de Tomcat 8

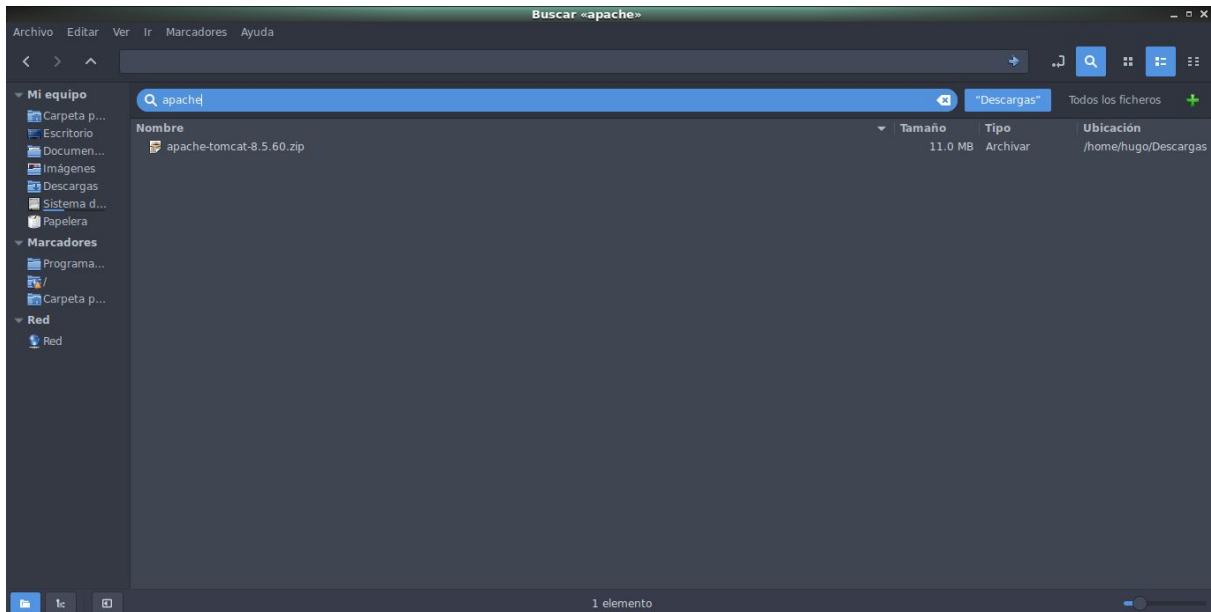


Figure 9: Descarga de la distribución binaria de Tomcat 8

4. Copiar a la máquina virtual el archivo ZIP descargado anteriormente y desempacar utilizando el comando unzip.

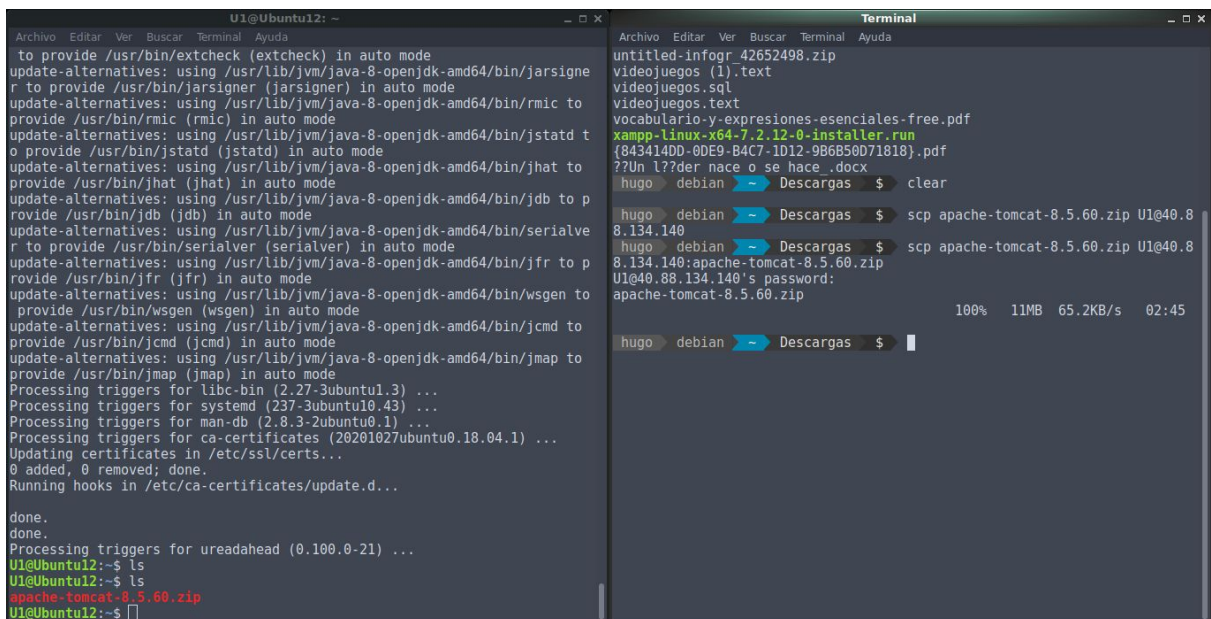


Figure 10: Copiar el archivo apache-tomcat-8.5.60.zip de la máquina física a la máquina virtual

```
U1@Ubuntu12: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
U1@Ubuntu12:~$ sudo apt install unzip  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  linux-headers-4.15.0-124  
Use 'sudo apt autoremove' to remove it.  
Suggested packages:  
  zip  
The following NEW packages will be installed:  
  unzip  
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.  
Need to get 167 kB of archives.  
After this operation, 558 kB of additional disk space will be used.  
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 unzip amd64 6.0-21ubuntu1 [167 kB]  
Fetched 167 kB in 0s (10.5 MB/s)  
Selecting previously unselected package unzip.  
(Reading database ... 77274 files and directories currently installed.)  
Preparing to unpack .../unzip_6.0-21ubuntu1_amd64.deb ...  
Unpacking unzip (6.0-21ubuntu1) ...  
Setting up unzip (6.0-21ubuntu1) ...  
Processing triggers for mime-support (3.60ubuntu1) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
U1@Ubuntu12:~$ unzip apache-tomcat-8.5.60.zip  
Archive:  apache-tomcat-8.5.60.zip  
  creating: apache-tomcat-8.5.60/  
  creating: apache-tomcat-8.5.60/bin/  
  creating: apache-tomcat-8.5.60/conf/  
  creating: apache-tomcat-8.5.60/lib/  
  creating: apache-tomcat-8.5.60/logs/  
  creating: apache-tomcat-8.5.60/temp/  
  creating: apache-tomcat-8.5.60/webapps/  
  creating: apache-tomcat-8.5.60/webapps/ROOT/  
  creating: apache-tomcat-8.5.60/webapps/ROOT/WEB-INF/  
  creating: apache-tomcat-8.5.60/webapps/docs/  
  creating: apache-tomcat-8.5.60/webapps/docs/WEB-INF/
```

Figure 11: Instalación del comando unzip y desempaquetado del archivo apache-tomcat-8.5.60.zip

5. Eliminar el directorio webapps el cual se encuentra dentro del directorio de Tomcat. Crear un nuevo directorio webapps y dentro de éste se deberá crear el directorio ROOT.

```
U1@Ubuntu12:~/apache-tomcat-8.5.60$ ls  
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work  
U1@Ubuntu12:~/apache-tomcat-8.5.60$ rm -r webapps  
U1@Ubuntu12:~/apache-tomcat-8.5.60$ ls  
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  work  
U1@Ubuntu12:~/apache-tomcat-8.5.60$
```

Figure 12: Eliminación del directorio webapps

```
U1@Ubuntu12:~/apache-tomcat-8.5.60$ mkdir webapps  
U1@Ubuntu12:~/apache-tomcat-8.5.60$ cd webapps/  
U1@Ubuntu12:~/apache-tomcat-8.5.60/webapps$ mkdir ROOT/  
U1@Ubuntu12:~/apache-tomcat-8.5.60/webapps$ cd ROOT/  
U1@Ubuntu12:~/apache-tomcat-8.5.60/webapps/ROOT$
```

Figure 13: Creación del directorio webapps y el subdirectorio ROOT

NOTA DE SEGURIDAD: Lo anterior se recomienda debido a que se han detectado vulnerabilidades en algunas aplicaciones que vienen con Tomcat, estas aplicaciones se encuentran originalmente instaladas en los directorios webapps y webapps/ROOT.

6. Descargar la biblioteca "Jersey" de la siguiente URL. Jersey es una implementación de JAX-RS lo cual permite ejecutar servicios web estilo REST sobre Tomcat:

<https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.24/jaxrs-ri-2.24.zip>

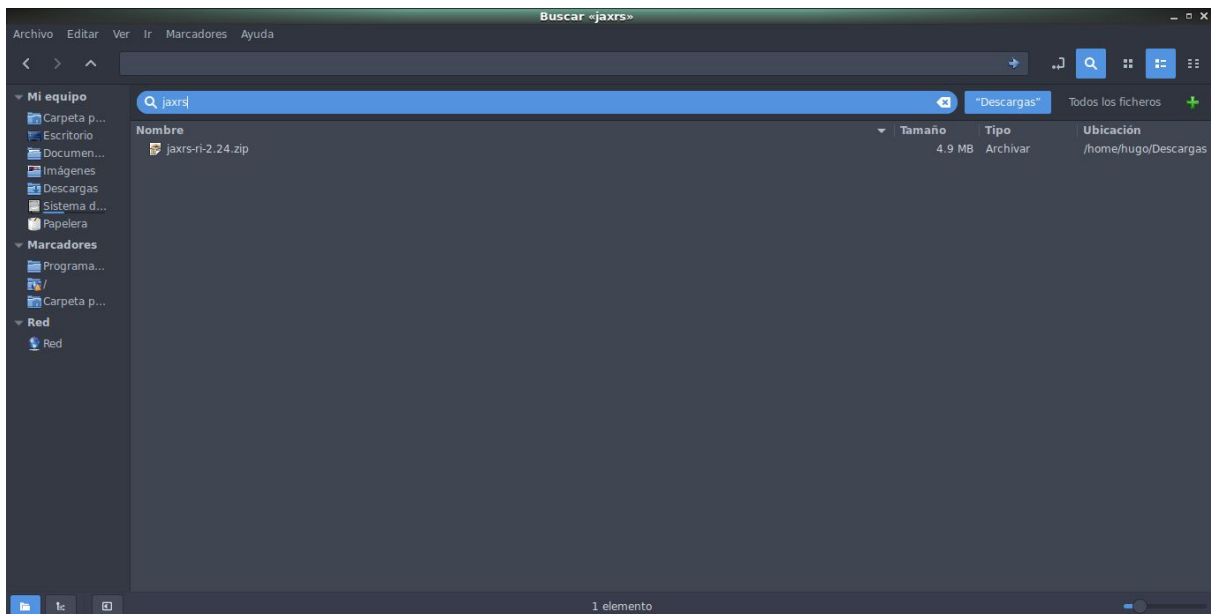


Figure 14: Descarga de la biblioteca "Jersey"

7. Copiar a la máquina virtual el archivo descargado anteriormente, desempacar y copiar todos los archivos con extensión “.jar” de todos los directorios desempacados, al directorio "lib" de Tomcat.

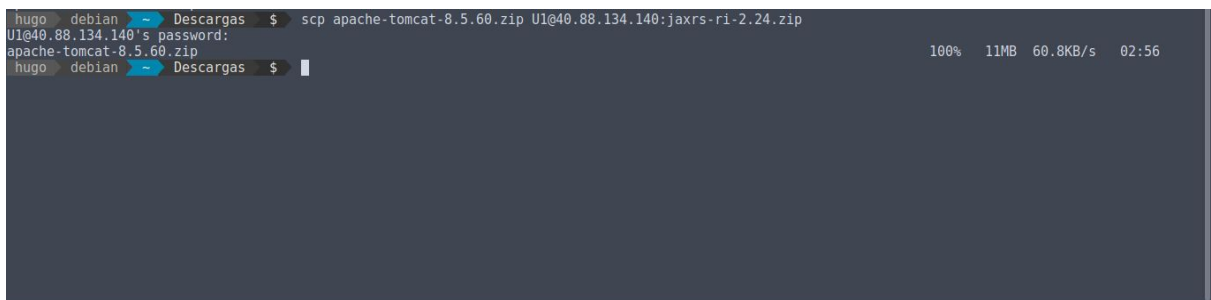


Figure 15: Copiar el archivo jaxrs-ri-2.24.zip de la máquina física a la máquina virtual

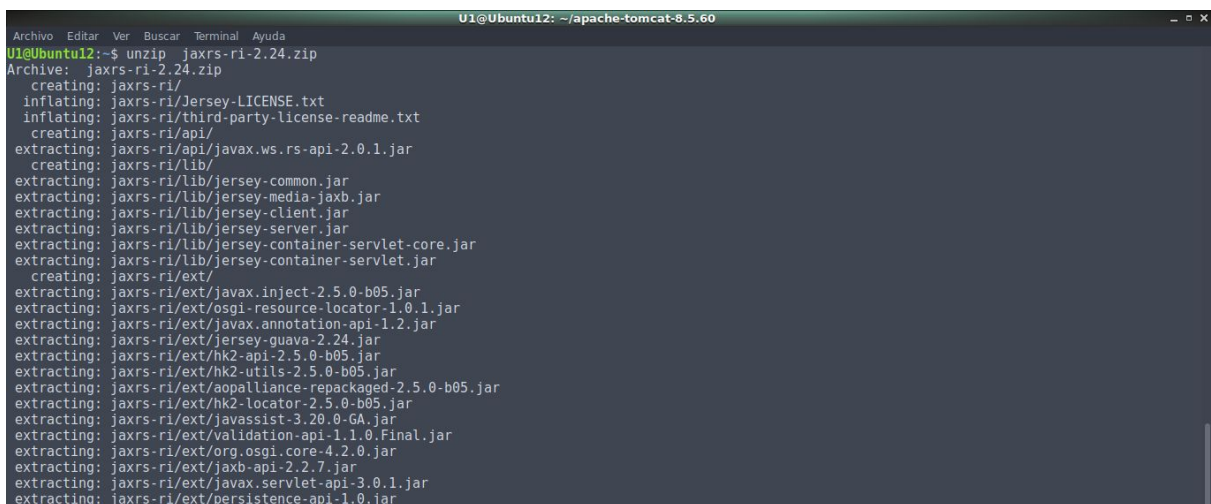


Figure 16: Desempaquetar el archivo jaxrs-ri-2.24.zip de la máquina física a la máquina virtual


```

U1@Ubuntu12:~/jaxrs-ri$ cd ext
U1@Ubuntu12:~/jaxrs-ri/ext$ ls
aopalliance-repackaged-2.5.0-b05.jar  javassist-3.20.0-GA.jar  jaxb-api-2.2.7.jar  persistence-api-1.0.jar
hk2-api-2.5.0-b05.jar                javax.annotation-api-1.2.jar  jersey-guava-2.24.jar  validation-api-1.1.0.Final.jar
hk2-locator-2.5.0-b05.jar             javax.inject-2.5.0-b05.jar  org.osgi.core-4.2.0.jar
hk2-utils-2.5.0-b05.jar               javax.servlet-api-3.0.1.jar  osgi-resource-locator-1.0.1.jar
U1@Ubuntu12:~/jaxrs-ri/ext$ cp *.jar /home/U1/apache-tomcat-8.5.60/lib
U1@Ubuntu12:~/jaxrs-ri/ext$ ls
aopalliance-repackaged-2.5.0-b05.jar  javassist-3.20.0-GA.jar  jaxb-api-2.2.7.jar  persistence-api-1.0.jar
hk2-api-2.5.0-b05.jar                javax.annotation-api-1.2.jar  jersey-guava-2.24.jar  validation-api-1.1.0.Final.jar
hk2-locator-2.5.0-b05.jar             javax.inject-2.5.0-b05.jar  org.osgi.core-4.2.0.jar
hk2-utils-2.5.0-b05.jar               javax.servlet-api-3.0.1.jar  osgi-resource-locator-1.0.1.jar
U1@Ubuntu12:~/jaxrs-ri/ext$ cd ..
U1@Ubuntu12:~/jaxrs-ri$ ls
Jersey-LICENSE.txt  api  ext  lib  third-party-license-readme.txt
U1@Ubuntu12:~/jaxrs-ri$ cd api
U1@Ubuntu12:~/jaxrs-ri/api$ ls
javax.ws.rs-api-2.0.1.jar
U1@Ubuntu12:~/jaxrs-ri/api$ cp *.jar /home/U1/apache-tomcat-8.5.60/lib
U1@Ubuntu12:~/jaxrs-ri/api$ cd ..
U1@Ubuntu12:~/jaxrs-ri$ ls
Jersey-LICENSE.txt  api  ext  lib  third-party-license-readme.txt
U1@Ubuntu12:~/jaxrs-ri$

```

Figure 17: Copiar todos los archivos con extensión “.jar” de todos los directorios desempaquetados, al directorio “lib” de Tomcat.

8. Borrar el archivo javax.servlet-api-3.0.1.jar del directorio “lib” de Tomcat (esto debe hacerse ya que existe una incompatibilidad entre Tomcat y Jersey 2).

```

U1@Ubuntu12:~/apache-tomcat-8.5.60$ cd lib/
U1@Ubuntu12:~/apache-tomcat-8.5.60/lib$ ls
annotations-api.jar          jersey-media-jaxb.jar      tomcat-i18n-ja.jar
aopalliance-repackaged-2.5.0-b05.jar  jasper.jar               tomcat-i18n-ko.jar
catalina-ant.jar             jaspic-api.jar           tomcat-i18n-ru.jar
catalina-ha.jar              javassist-3.20.0-GA.jar   tomcat-i18n-zh-CN.jar
catalina-storeconfig.jar      javax.annotation-api-1.2.jar  tomcat-jdbc.jar
catalina-tribes.jar           javax.inject-2.5.0-b05.jar  tomcat-jni.jar
catalina.jar                 javax.servlet-api-3.0.1.jar  tomcat-util-scan.jar
ecj-4.6.3.jar                javax.ws.rs-api-2.0.1.jar   tomcat-util.jar
el-api.jar                   jaxb-api-2.2.7.jar        tomcat-websocket.jar
hk2-api-2.5.0-b05.jar          jersey-client.jar         validation-api-1.1.0.Final.jar
hk2-locator-2.5.0-b05.jar      jersey-common.jar         websocket-api.jar
hk2-utils-2.5.0-b05.jar        jersey-container-servlet-core.jar
jasper-el.jar                jersey-container-servlet.jar
                             jersey-guava-2.24.jar      tomcat-i18n-de.jar
                                                         tomcat-i18n-es.jar
                                                         tomcat-i18n-fr.jar
U1@Ubuntu12:~/apache-tomcat-8.5.60/lib$ rm javax.servlet-api-3.0.1.jar
U1@Ubuntu12:~/apache-tomcat-8.5.60/lib$ ls
annotations-api.jar          jersey-server.jar          tomcat-i18n-ko.jar
aopalliance-repackaged-2.5.0-b05.jar  jsp-api.jar              tomcat-i18n-ru.jar
catalina-ant.jar             javax.annotation-api-1.2.jar  tomcat-i18n-zh-CN.jar
catalina-ha.jar              osgi-resource-locator-1.0.1.jar  tomcat-jdbc.jar
catalina-storeconfig.jar      persistence-api-1.0.jar      tomcat-jni.jar
catalina-tribes.jar          servlet-api.jar             tomcat-util-scan.jar
catalina.jar                 tomcat-api.jar             tomcat-util.jar
ecj-4.6.3.jar                tomcat-coyote.jar          tomcat-websocket.jar
el-api.jar                   tomcat-dbcp.jar            validation-api-1.1.0.Final.jar
hk2-api-2.5.0-b05.jar          tomcat-i18n-de.jar         websocket-api.jar
hk2-locator-2.5.0-b05.jar      tomcat-i18n-es.jar
hk2-utils-2.5.0-b05.jar        tomcat-i18n-fr.jar
jasper-el.jar                 tomcat-i18n-ja.jar
U1@Ubuntu12:~/apache-tomcat-8.5.60/lib$

```

Figure 18: Eliminación del archivo javax.servlet-api-3.0.1.jar

9. Descargar el archivo gson-2.3.1.jar de la URL:

<https://repo1.maven.org/maven2/com/google/code/gson/gson/2.3.1/gson-2.3.1.jar>

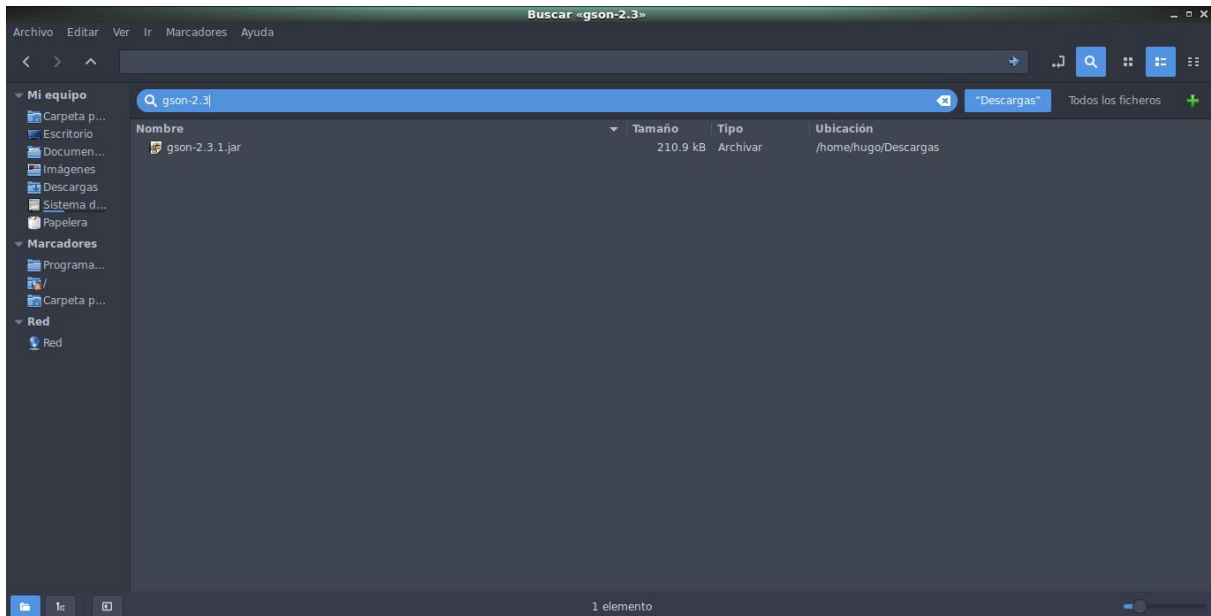


Figure 12: Descarga del archivo gson-2.3.1.jar

10. Copiar el archivo gson-2.3.1.jar al directorio "lib" de Tomcat.

```

U1@Ubuntu12:~/apache-tomcat-8.5.60/lib$ mv /home/U1/gson-2.3.1.jar /home/U1/apache-tomcat-8.5.60/lib
U1@Ubuntu12:~/apache-tomcat-8.5.60/lib$ ls
annotations-api.jar      jasper-el.jar           jersey-media-jaxb.jar   tomcat-i18n-ja.jar
aopalliance-repackaged-2.5.0-b05.jar  jasper.jar             jersey-server.jar       tomcat-i18n-ko.jar
catalina-ant.jar         jaspic-api.jar         jsp-api.jar            tomcat-i18n-ru.jar
catalina-ha.jar          javassist-3.20.0-GA.jar org.osgi.core-4.2.0.jar tomcat-i18n-zh-CN.jar
catalina-storeconfig.jar javax.annotation-api-1.2.jar osgi-resource-locator-1.0.1.jar tomcat-jdbc.jar
catalina-tribes.jar      javax.inject-2.5.0-b05.jar persistence-api-1.0.jar tomcat-jni.jar
catalina.jar             javax.ws.rs-api-2.0.1.jar servlet-api.jar         tomcat-util-scan.jar
ecj-4.6.3.jar            jaxb-api-2.2.7.jar     tomcat-api.jar         tomcat-util.jar
el-api.jar              jersey-client.jar       tomcat-coyote.jar      tomcat-websocket.jar
gson-2.3.1.jar           jersey-common.jar       tomcat-dbcp.jar        validation-api-1.1.0.Final.jar
hk2-api-2.5.0-b05.jar    jersey-container-servlet-core.jar tomcat-i18n-de.jar     websocket-api.jar
hk2-locator-2.5.0-b05.jar jersey-container-servlet.jar tomcat-i18n-es.jar
hk2-utils-2.5.0-b05.jar jersey-guava-2.24.jar   tomcat-i18n-fr.jar
U1@Ubuntu12:~/apache-tomcat-8.5.60/lib$

```

Figure 12: Copiar el archivo gson-2.3.1.jar

11. Ahora vamos a instalar el driver de JDBC para MySQL. Ingresar a la siguiente URL:

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

Seleccionar "Platform independent" y descargar el archivo ZIP.

MySQL Community Downloads

Connector/J

General Availability (GA) Releases Archives

Connector/J 8.0.22

Select Operating System:
Platform Independent

Looking for previous GA versions?

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.22.tar.gz)	8.0.22	3.8M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.22.zip)	8.0.22	4.5M	Download

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

Figure 13: Selección del driver de JDBC para MySQL

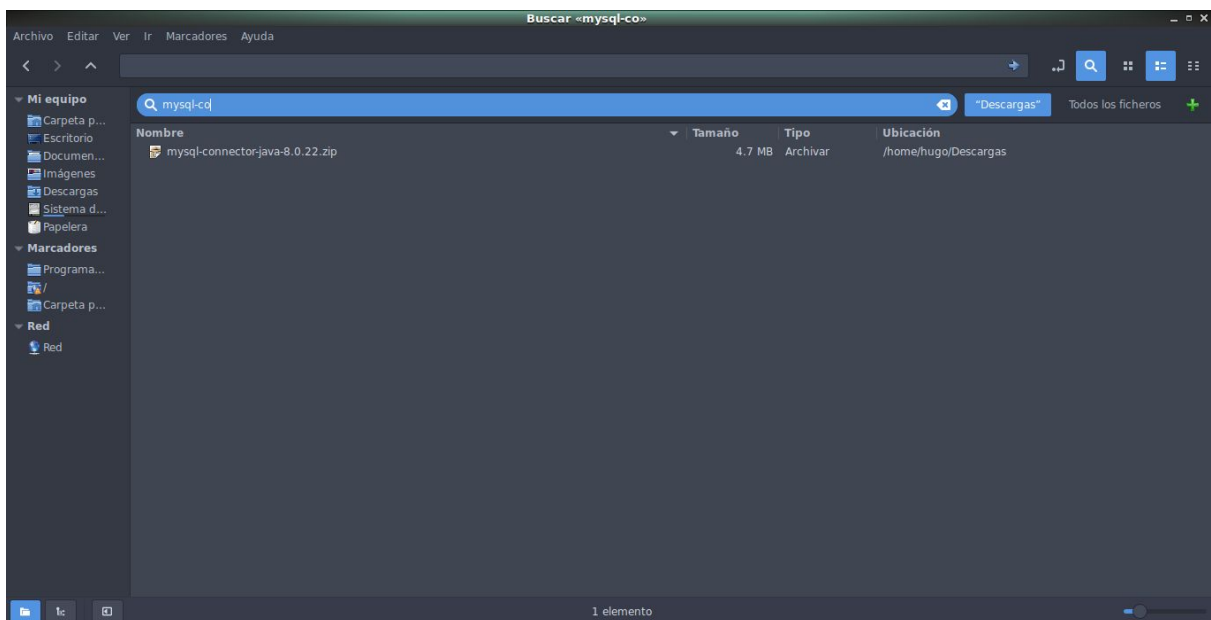


Figure 14: Descarga del driver de JDBC para MySQL

12. Copiar el archivo descargado a la máquina virtual, desempacar y copiar el archivo mysql-connector...jar al directorio "lib" de Tomcat.

```
hugo@debian: Descargas$ scp mysql-connector-java-8.0.22.zip U1@40.88.134.140:mysql-connector-java-8.0.22.zip
U1@40.88.134.140's password:
Permission denied, please try again.
U1@40.88.134.140's password:
mysql-connector-java-8.0.22.zip 100% 4632KB 63.7KB/s 01:12
hugo@debian: Descargas$
```

Figure 15: Envío del archivo descargado a la máquina virtual

```
U1@Ubuntu12: ~
Archivo Editar Ver Buscar Terminal Ayuda
U1@Ubuntu12:~$ ls
apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip  jaxrs-ri  jaxrs-ri-2.24.zip  mysql-connector-java-8.0.22.zip
U1@Ubuntu12:~$ unzip mysql-connector-java-8.0.22.zip
Archive:  mysql-connector-java-8.0.22.zip
  creating: mysql-connector-java-8.0.22/
  creating: mysql-connector-java-8.0.22/src/
```

Figure 16: Archivo recibido en la máquina virtual

```
U1@Ubuntu12:~$ ls
apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip  jaxrs-ri  jaxrs-ri-2.24.zip  mysql-connector-java-8.0.22  mysql-connector-java-8.0.22.zip
U1@Ubuntu12:~$ cd mysql-connector-java-8.0.22/
-bash: cd: too many arguments
U1@Ubuntu12:~$ cd mysql-connector-java-8.0.22/
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ ls
CHANGES  INFO  BIN  INFO  SRC  LICENSE  README  build.xml  mysql-connector-java-8.0.22.jar  src
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ cp mysql-connector-java-8.0.22.jar /home/U1/apache-tomcat-8.5.60/lib
U1@Ubuntu12:~/mysql-connector-java-8.0.22$
```

cp mysql-connector-java-8.0.22.zip

cp mysql-connector-java-8.0.22.jar /home/U1/apache-tomcat-8.5.60/lib

scp mysql-connector-java-8.0.22.zip U1@40.88.134.140:mysql-connector-java-8.0.22.zip

Iniciar/detener el servidor Tomcat

1. Para iniciar el servidor Tomcat es necesario definir las siguientes variables de entorno:

```
export CATALINA_HOME=aquí va la ruta del directorio de Tomcat 8
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

```
U1@Ubuntu12:~/apache-tomcat-8.5.60$ pwd
/home/U1/apache-tomcat-8.5.60
U1@Ubuntu12:~/apache-tomcat-8.5.60$ export CATALINA_HOME=/home/U1/apache-tomcat-8.5.60
U1@Ubuntu12:~/apache-tomcat-8.5.60$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
U1@Ubuntu12:~/apache-tomcat-8.5.60$ sh $CATALINA_HOME/bin/catalina.sh start
Using CATALINA_BASE:   /home/U1/apache-tomcat-8.5.60
Using CATALINA_HOME:   /home/U1/apache-tomcat-8.5.60
Using CATALINA_TMPDIR: /home/U1/apache-tomcat-8.5.60/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH:       /home/U1/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/U1/apache-tomcat-8.5.60/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
U1@Ubuntu12:~/apache-tomcat-8.5.60$
```

2. Iniciar la ejecución de Tomcat ejecutando el siguiente comando:

```
sh $CATALINA_HOME/bin/catalina.sh start
```

```
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ sh $CATALINA_HOME/bin/catalina.sh start
Using CATALINA_BASE:   /home/U1/apache-tomcat-8.5.60
Using CATALINA_HOME:   /home/U1/apache-tomcat-8.5.60
Using CATALINA_TMPDIR: /home/U1/apache-tomcat-8.5.60/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH:       /home/U1/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/U1/apache-tomcat-8.5.60/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
U1@Ubuntu12:~/mysql-connector-java-8.0.22$
```

Instalación de MySQL

1. Actualizar los paquetes en la máquina virtual ejecutando el siguiente comando:

```
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1789 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1695 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [244 kB]
Fetched 3981 kB in 1s (3394 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
U1@Ubuntu12:~/mysql-connector-java-8.0.22$
```

2. Instalar el paquete default de MySQL:


```
U1@Ubuntu12: ~/mysql-connector-java-8.0.22
Archivo Editar Ver Buscar Terminal Ayuda
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-headers-4.15.0-124
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycsa
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 4 not upgraded.
Need to get 19.7 MB of archives.
After this operation, 157 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 mysql-common all 5.8+1.0.4 [7388 B]
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libaio1 amd64 0.3.110-Subuntu0.1 [6476 B]
Get:3 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.32-0ubuntu0.18.04.1 [6660 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-5.7 amd64 5.7.32-0ubuntu0.18.04.1 [1943 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-server-core-5.7 amd64 5.7.32-0ubuntu0.18.04.1 [7455 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libevent-core-2.1-6 amd64 2.1.8-stable-4build1 [85.9 kB]
```

3. Ejecutar el script de seguridad:

```
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ sudo mysql_secure_installation
Securing the MySQL server deployment.
Connecting to MySQL using a blank password.
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?
Press y|Y for Yes, any other key for No: N
Please set the password for root here.
New password:
Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.
Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y
Dropping test database...
Success.
Removing privileges on test database...
Success.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.

All done!
U1@Ubuntu12:~/mysql-connector-java-8.0.22$
```

4. Ejecutar el monitor de MySQL:

sudo mysql

```
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

5. Ejecutar el siguiente comando SQL para modificar la contraseña de root:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
'contraseña-de-root-en-mysql';
```



```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'hugo9';
Query OK, 0 rows affected (0.00 sec)

mysql> █
```

6. Actualizar los privilegios:

FLUSH PRIVILEGES;

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> █
```

7. Ejecutar el siguiente comando para salir del monitor de MySQL:

quit

```
mysql> quit
Bye
```

Crear un usuario en MySQL

1. Ejecutar el monitor de MySQL:

mysql -u root -p

```
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

2. Crea el usuario "hugo":

create user hugo@localhost identified by 'hugo09';

```
mysql> create user hugo@localhost identified by 'hugo09';
Query OK, 0 rows affected (0.00 sec)

mysql> █
```

3. Otorgar todos los permisos al usuario "hugo" sobre la base de datos "servicio_web":

grant all on servicio_web.* to hugo@localhost;

```
mysql> grant all on servicio web.* to hugo@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> █
```

4. Ejecutar el siguiente comando para salir del monitor de MySQL:

quit

```
mysql> quit
Bye
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ █
```

Crear la base de datos

1. Ejecutar el monitor de MySQL (notar que ahora se utiliza el usuario "hugo"):

```
mysql -u hugo -p
```

```
U1@Ubuntu12:~/mysql-connector-java-8.0.22$ mysql -u hugo -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2. Crear la base de datos "servicio_web":

```
create database servicio_web;
```

```
mysql> create database servicio_web;
Query OK, 1 row affected (0.00 sec)

mysql>
```

3. Conectar a la base de datos creada anteriormente:

```
use servicio_web;
```

```
mysql> use servicio_web;
Database changed
mysql>
```

4. Crear las tablas "usuarios" y "fotos_usuarios", así mismo, se crea una regla de integridad referencial y un índice único:

```
create table usuarios
```

```
(
    id_usuario integer auto_increment primary key,
    email varchar(256) not null,
    nombre varchar(100) not null,
    apellido_paterno varchar(100) not null,
    apellido_materno varchar(100),
    fecha_nacimiento date not null,
    telefono varchar(20),
    genero char(1)
);
```

```
mysql> create table usuarios
-> (
-> id_usuario integer auto increment primary key,
-> email varchar(256) not null,
-> nombre varchar(100) not null,
-> apellido_paterno varchar(100) not null,
-> apellido_materno varchar(100),
-> fecha_nacimiento date not null,
-> telefono varchar(20),
-> genero char(1)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql>
```

```
create table fotos_usuarios
(
    id_foto integer auto_increment primary key,
    foto longblob,
    id_usuario integer not null
);
```

```
mysql> create table fotos_usuarios
-> (
-> id_foto integer auto_increment primary key,
-> foto longblob,
-> id_usuario integer not null
-> );
Query OK, 0 rows affected (0.04 sec)
mysql>
```

```
alter table fotos_usuarios add foreign key (id_usuario) references
usuarios(id_usuario);
create unique index usuarios_1 on usuarios(email);
```

```
mysql> alter table fotos_usuarios add foreign key (id_usuario) references usuarios(id_usuario);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create unique index usuarios_1 on usuarios(email);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

5. Salir del monitor de MySQL:

```
quit
```

```
mysql> quit
Bye
U1@Ubuntu12:~/mysql-connector-java-8.0.22$
```

Compilar, empaquetar y desplegar el servicio web

1. Descargar de la plataforma y desempacar el archivo [Servicio.zip](#).

```
U1@Ubuntu12:~$ ls
Servicio.zip  apache-tomcat-8.5.60.zip  jaxrs-ri-2.24.zip  mysql-connector-java-8.0.22.zip
apache-tomcat-8.5.60  jaxrs-ri  mysql-connector-java-8.0.22
U1@Ubuntu12:~$ unzip Servicio.zip
Archive: Servicio.zip
  creating: META-INF/
  inflating: META-INF/context.xml
  creating: WEB-INF/
  inflating: WEB-INF/web.xml
  creating: WEB-INF/classes/
  creating: WEB-INF/classes/negocio/
  creating: negocio/
  inflating: negocio/Usuario.java
  inflating: negocio/Foto.java
  inflating: negocio/AdaptadorGsonBase64.java
  inflating: negocio/Error.java
  inflating: negocio/Servicio.java
```

2. Definir la variable de ambiente CATALINA_HOME:

```
export CATALINA_HOME=aquí va la ruta completa del directorio de Tomcat
8
```

3. Cambiar al directorio dónde se desempacó el archivo [Servicio.zip](#) (en ese directorio se encuentra el directorio "negocio").

```

U1@Ubuntu12:~$ ls
META-INF  WEB-INF  apache-tomcat-8.5.60.zip  jaxrs-ri-2.24.zip  mysql-connector-java-8.0.22.zip
Servicio.zip  apache-tomcat-8.5.60  jaxrs-ri  mysql-connector-java-8.0.22  negocio
U1@Ubuntu12:~$ cd negocio/
U1@Ubuntu12:~/negocio$ ls
AdaptadorGsonBase64.java  Error.java  Foto.java  Servicio.java  Usuario.java
U1@Ubuntu12:~/negocio$

```

4. Compilar la clase Servicio.java:

```

javac -cp
$CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.
3.1.jar:. negocio/Servicio.java

```

```

U1@Ubuntu12:~$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
U1@Ubuntu12:~$ ls
META-INF  WEB-INF  apache-tomcat-8.5.60.zip  jaxrs-ri-2.24.zip  mysql-connector-java-8.0.22.zip
Servicio.zip  apache-tomcat-8.5.60  jaxrs-ri  mysql-connector-java-8.0.22  negocio
U1@Ubuntu12:~$

```

5. Editar el archivo "context.xml" que está en el directorio "META-INF" y definir el username de la base de datos y el password correspondiente. El usuario "hugo" fue creado en el paso 2 de la sección Crear un usuario en MySQL.

```

U1@Ubuntu12: ~/META-INF
GNU nano 2.9.3 context.xml Modified
<Context>
  <Resource name="jdbc/datasource_Servicio" auth="Container" type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="hugo" password="hugo89"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost/servicio_web?serverTimezone=UTC"/>
</Context>

```

6. Ejecutar los siguientes comandos para crear el servicio web para Tomcat (notar que los servicios web para Tomcat son archivos JAR con la extensión .war):

```

rm WEB-INF/classes/negocio/*
cp negocio/*.class WEB-INF/classes/negocio/.
jar cvf Servicio.war WEB-INF META-INF

```

7. Para desplegar (*deploy*) el servicio web, copiar el archivo Servicio.war al directorio "webapps" de Tomcat. Notar que Tomcat desempaca automáticamente los archivos con extensión .war que se encuentran en el directorio webapps de Tomcat.

```
U1@Ubuntu12:~$ cp Servicio.war /home/U1/apache-tomcat-8.5.60/webapps
U1@Ubuntu12:~$ cd apache-tomcat-8.5.60/
U1@Ubuntu12:~/apache-tomcat-8.5.60$ cd webapps/
U1@Ubuntu12:~/apache-tomcat-8.5.60/webapps$ ls
ROOT  Servicio  Servicio.war
```

Para eliminar el servicio web se deberá eliminar el archivo "Servicio.war" y el directorio "Servicio", en éste orden.

Cada vez que se modifique el archivo Servicio.java se deberá compilar, generar el archivo Servicio.war, borrar el archivo Servicio.war y el directorio Servicio del directorio webapps de Tomcat, y copiar el archivo Servicio.war al directorio webapps de Tomcat.

Probar el servicio web utilizando HTML-Javascript

1. Copiar el archivo [usuario_sin_foto.png](#) al subdirectorio webapps/ROOT de Tomcat.

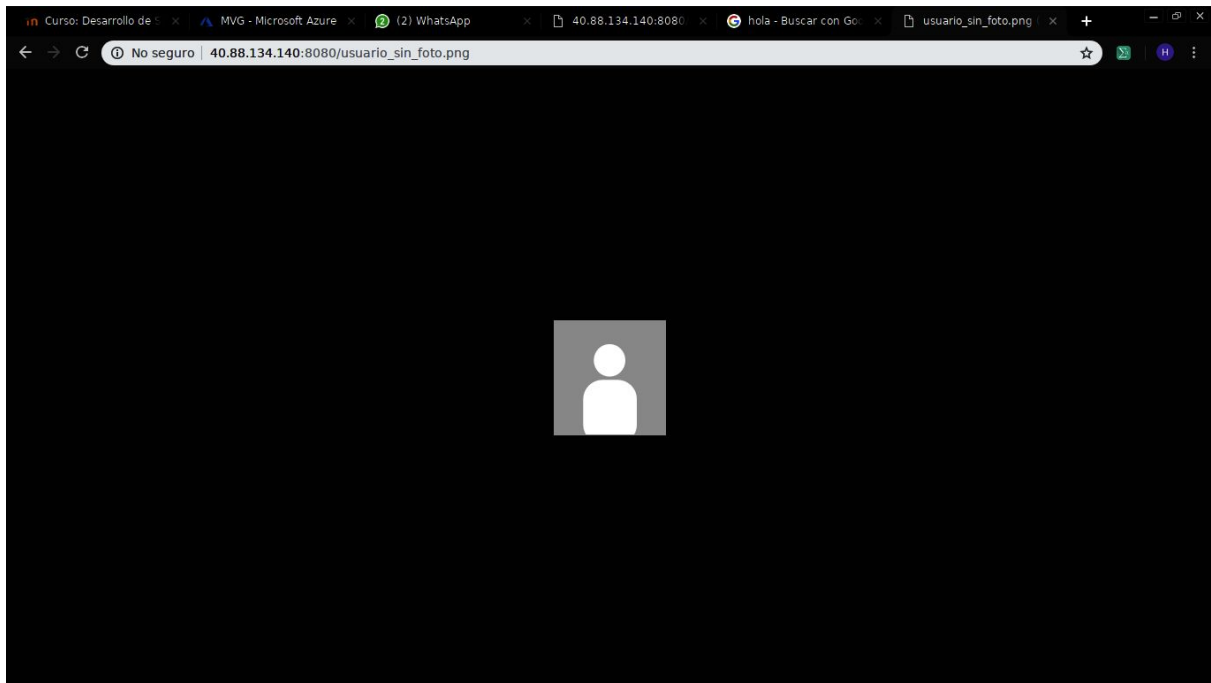
Notar que todos los archivos que se encuentran en el directorio webapps/ROOT de Tomcat son accesibles públicamente.

```
hugo@debian: Descargas 2 $ scp usuario_sin_foto.png U1@40.88.134.140:usuario_sin_foto.png
U1@40.88.134.140's password:
usuario_sin_foto.png
hugo@debian: Descargas 2 $
```

```
U1@Ubuntu12:~/apache-tomcat-8.5.60/webapps/ROOT$ ls
usuario_sin_foto.png
```

Para probar que Tomcat esté en línea y el puerto 8080 esté abierto, ingresar la siguiente URL en un navegador:

`http://ip-de-la-máquina-virtual:8080/usuario_sin_foto.png`



2. Copiar el archivo [WSClient.js](#) al directorio webapps/ROOT de Tomcat.

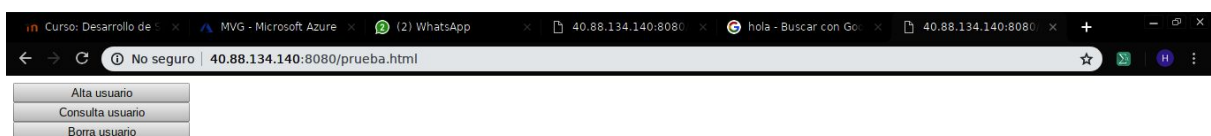
```
U1@Ubuntu12:~/apache-tomcat-8.5.60/webapps/ROOT$ nano WSClient.js
```

3. Copiar el archivo prueba.html al directorio webapps/ROOT de Tomcat.

```
U1@Ubuntu12:~/apache-tomcat-8.5.60/webapps/ROOT$ nano prueba.html
```

4. Ingresar la siguiente URL en un navegador:

`http://ip-de-la-máquina-virtual:8080/prueba.html`



5. Dar clic en el botón “Alta usuario” para dar de alta un nuevo usuario. Capturar los campos y dar clic en el botón “Alta”.

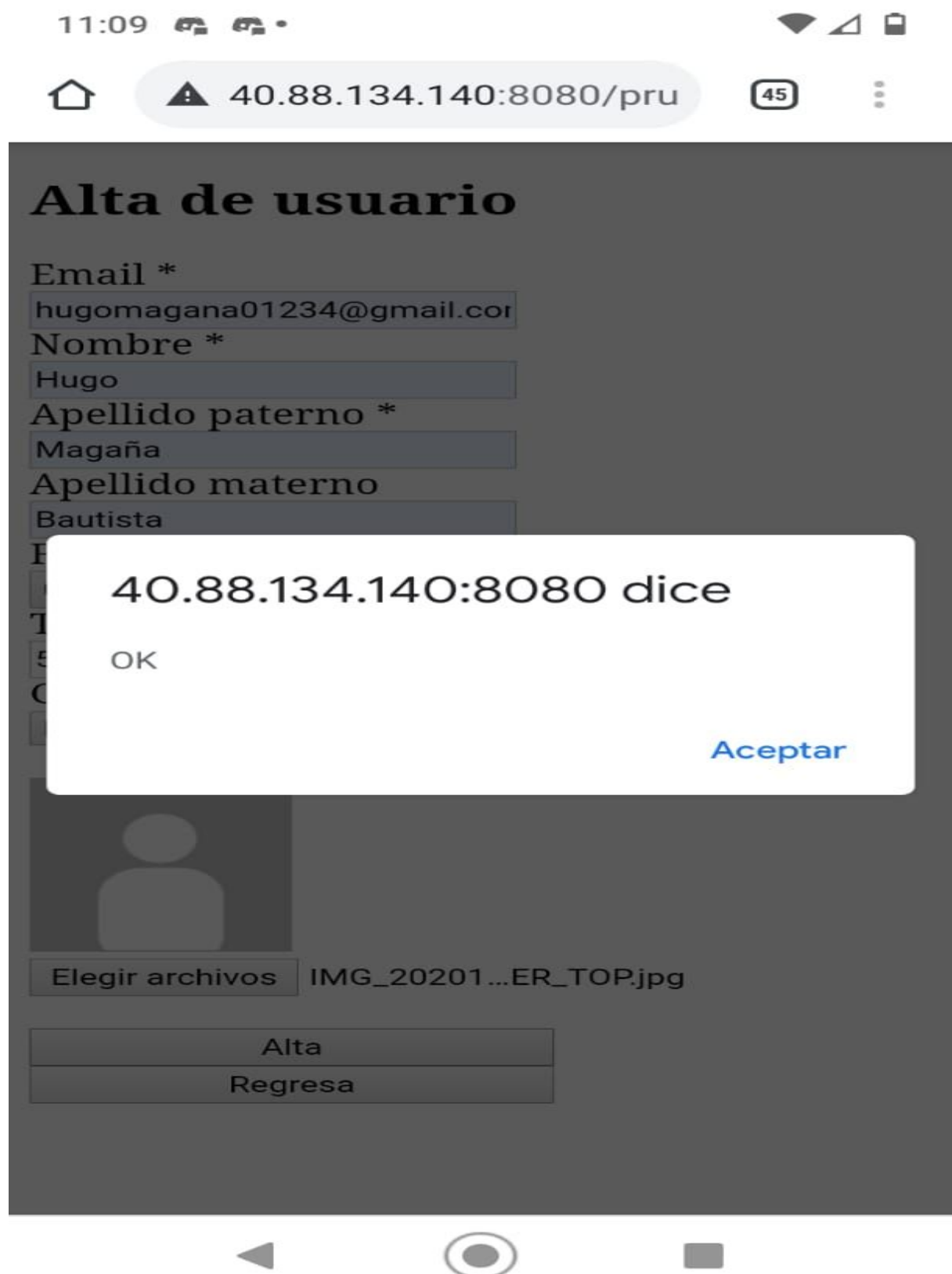


Figure 50: Registrar usuario con correo hugomagana01234@gmai.com

6. Intentar dar de alta otro usuario con el mismo email (se deberá mostrar una ventana de error indicando que el email ya existe)



Figure 50: Volver a registrar usuario con correo hugomagana01234@gmail.com

7. Dar clic en el botón "Consulta usuario" para consultar el usuario dado de alta en el paso 5. Capturar el email y dar clic en el botón "Consulta",

11:06



40.88.134.140:8080/pru

45



Consulta/Modifica usuario

Email *

hugomagana01234@gmail.com

Nombre *

Hugo

Apellido paterno *

Magaña

Apellido materno

Bautista

Fecha de nacimiento *

04/11/2020

Teléfono

5623124575

Genero

Masculino



Elegir archivos

No se eligió archivo

Consulta

Modifica

Regresa

Figure 50: Consultar usuario con correo hugomagana01234@gmail.com

8. Modificar algún dato del usuario y dar clic en el botón “Modifica”:

Consulta/Modifica usuario

Email *
hugomagana01234@gmail.com

Nombre *
Sebastian

Apellido paterno *
Magaña

Apellido materno
Bautista

Fecha de nacimiento *
04/11/2020

Teléfono
5623236396

Genero
Masculino ▼


Elegir archivos No se eligió archivo

Consulta
Modifica
Regresa

Figure 50: Cambiar nombre de Hugo a Sebastian al usuario con correo hugomagana01234@gmail.com

9. Recargar la página actual y consultar el usuario modificado, para verificar que la modificación se realizó.

Consulta/Modifica usuario

Email *
hugomagana01234@gmail.com

Nombre *
Sebastian

Apellido paterno *
Magaña

Apellido materno
Bautista

Fecha de nacimiento *
04/11/2020

Teléfono
5623236396

Genero
Masculino ▼


Elegir archivos No se eligió archivo

Consulta
Modifica
Regresa

Figure 50: Verificar nombre del usuario con correo hugomagana01234@gmail.com

10. Dar clic en el botón “Borra usuario” para borrar el usuario. Capturar el email del usuario a borrar y dar clic en el botón “Consulta”.

11:06



40.88.134.140:8080/pru

45



Borra usuario

Email *

hugomagana01234@gmail.cor

Borra

Regresa

40.88.134.140:8080 dice

OK

Aceptar

Figure 50: Eliminar usuario con correo hugomagana01234@gmail.com

Conclusiones:

En esta práctica aprendí a realizar la Implementación de un servicio web estilo REST, el cual ampliamente utilizado en la industria y lo considero se suma relevancia para cuando me incorpore al mercado.

Además, aprendí más del lenguaje de programación java, debido a que es un lenguaje en el que tenía un conocimiento intermedio, pero al realizar prácticas de este estilo me doy cuenta del gran potencial que tiene esta herramienta que aunque existen infinitas de lenguajes con lo que compete directamente aún es deseable profundizar en el conocimiento.

Por último, conforme vamos avanzando en los temas vamos desarrollando sistemas más robustos, y este es un ejemplo puesto que este programa nos permite aplicar varios de los conocimientos que hemos venido aprendiendo a lo largo del curso.