



Desarrollo de Sistemas Distribuidos

Tarea 7

**Eduardo Alberto Pereda
Guzman**

**Prof. Pineda Guerrero
Carlos**



Implementación
de un servicio
web estilo REST

27 de noviembre

2020

Introducción.

REST define un conjunto de principios arquitectónicos para la creación de servicios web. REST fue presentado por Roy Fielding el año 2000 en su disertación doctoral "Architectural Styles and the Design of Network-based Software Architectures".

El diseño de servicios web estilo REST sigue cuatro principios:

- Utilizar métodos HTTP de forma explícita.
 - Un servicio web utiliza los métodos de HTTP para crear un recurso (POST), leer (GET), cambiar el estado o actualizar un recurso (PUT), y borrar un recurso (DELETE).
- Los servicios son sin estado (stateless).
 - Los clientes de servicios web estilo REST deben enviar peticiones completas e independientes, es decir, las peticiones deben incluir todos los datos que permitan completar el servicio, sin la necesidad de guardar un estado entre peticiones.
- Los URIs representan una estructura de directorios.
 - Los URIs (Uniform Resource Identifier) deben ser intuitivos y auto-explicados. Un URI es una jerarquía que corresponde a la estructura de los servicios web definidos en la empresa.
- Se transfiere XML, JSON o ambos.
 - Los recursos que provee un servicio web pueden ser documentos, imágenes, videos y en general objetos. La representación de objetos mediante XML o JSON es fácil e independiente de la plataforma.

Desarrollo.

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.

Se creó una máquina virtual llamada "ServidorREST" con una ip publica: 20.185.80.52, además que se habilitó el puerto 8080.

Search resources, services, and docs (G+)

[Home](#) > [Virtual machines](#) >

Create a virtual machine

Validation passed

PRODUCT DETAILS

Subscription by Microsoft
[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ
0.2007 MXN/hr
[Pricing for other VM sizes](#)

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

You have set SSH port(s) open to the internet. This is only recommended for testing. If you want to change this setting, go back to Basics tab.

Basics

Subscription	Azure for Students
Resource group	(new) ServidorREST_group_11271414
Virtual machine name	ServidorREST
Region	East US
Availability options	No infrastructure redundancy required
Image	Ubuntu Server 18.04 LTS - Gen1
Size	Standard B1s (1 vcpu, 1 GiB memory)
Authentication type	Password
Username	ubuntu
Public inbound ports	SSH
Azure Spot	No

Create

[< Previous](#)
[Next >](#)
[Download a template for automation](#)

Inbound port rules
Outbound port rules
Application security groups
Load balancing

Network security group **SecurityREST-nsg** (attached to network interface: **sevidorrest678**)
Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action	
300	🚩 SSH	22	TCP	Any	Any	🟢 Allow	...
310	Port_8080	8080	Any	Any	Any	🟢 Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	🟢 Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	🟢 Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	🔴 Deny	...

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

Se realiza actualización de paquetes e instalación de Java 8.

```
sudo apt update
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>

```
ubuntu@ServidorREST: ~  
login as: ubuntu  
ubuntu@20.185.80.52's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1031-azure x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
System information as of Thu Nov 26 01:36:12 UTC 2020  
  
System load:  0.38      Processes:      113  
Usage of /:   4.5% of 28.90GB   Users logged in:  0  
Memory usage: 19%      IP address for eth0: 10.0.0.4  
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.  
  
ubuntu@ServidorREST:~$ 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 [857  
0 kB]  
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic/universe Translation-en [494  
1 kB]  
Get:7 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [1  
51 kB]  
Get:8 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [1  
08 kB]  
Get:9 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages  
[1761 kB]  
Get:10 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Pack  
ages [1693 kB]
```

Imagen 2: Ejecución del comando “sudo apt update” y “sudo apt install openjdk-8-jdk-headless”

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

Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.

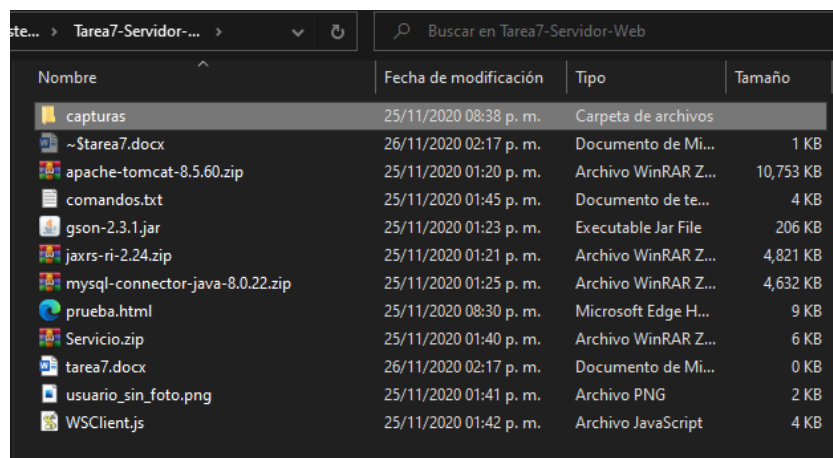


Imagen 3: Directorio con los archivos descargados para la práctica

Se utilizó el programa portable “WinSCP” para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.

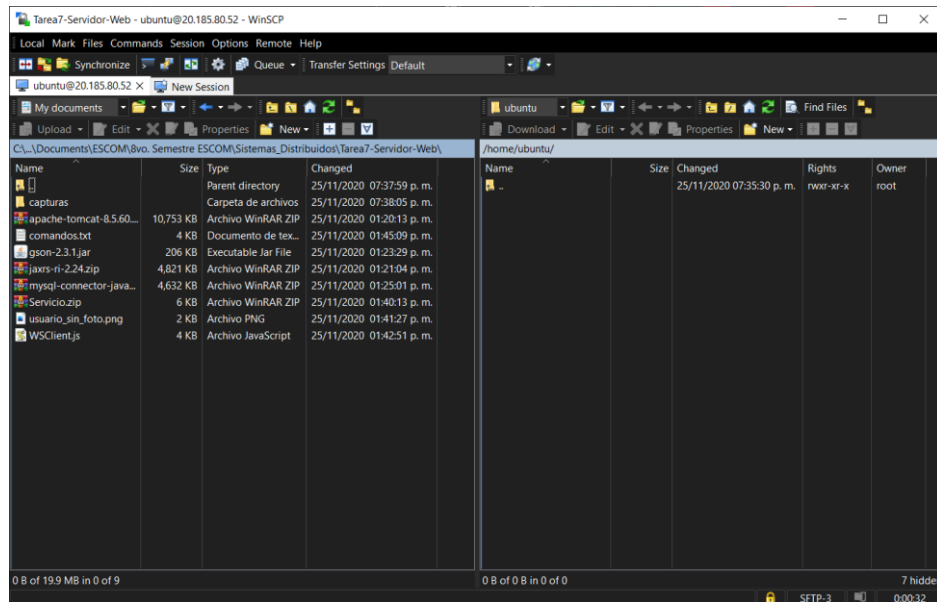


Imagen 4: Interfaz del programa WINSCP con la conexión realizada a la máquina virtual remota.

Se tuvo que instalar el programa “Unzip” para poder realizar el desempaquetado de los archivos comprimidos en zip.

```
ubuntu@ServidorREST: ~  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jarsigner to provide /usr/bin/jarsigner (jarsigner) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/rmic to provide /usr/bin/rmic (rmic) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jhat to provide /usr/bin/jhat (jhat) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jdb to provide /usr/bin/jdb (jdb) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jfr to provide /usr/bin/jfr (jfr) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsgen to provide /usr/bin/wsgen (wsgen) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jcmd to provide /usr/bin/jcmd (jcmd) in auto mode  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jmap to provide /usr/bin/jmap (jmap) in auto mode  
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...  
Processing triggers for systemd (237-3ubuntu0.43) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Processing triggers for ca-certificates (20201027ubuntu0.18.04.1) ...  
Updating certificates in /etc/ssl/certs...  
0 added, 0 removed; done.  
Running hooks in /etc/ca-certificates/update.d...  
  
done.  
done.  
Processing triggers for ureadahead (0.100.0-21) ...  
ubuntu@ServidorREST:~$ pwd  
/home/ubuntu  
ubuntu@ServidorREST:~$ ls  
apache-tomcat-8.5.60.zip  
ubuntu@ServidorREST:~$ unzip apache-tomcat-8.5.60.zip  
  
Command 'unzip' not found, but can be installed with:  
  
sudo apt install unzip  
ubuntu@ServidorREST:~$ sudo apt install unzip
```

Imagen 5: Instalación del programa “Unzip”

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.

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.

```
ubuntu@ServidorREST: ~/apache-tomcat-8.5.60
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/jsp/403.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/jsp/404.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/jsp/connectorCerts.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/jsp/connectorCiphers.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/jsp/connectorTrustedCertificates.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/jsp/sessionDetail.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/jsp/sessionsList.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/WEB-INF/web.xml
inflating: apache-tomcat-8.5.60/webapps/manager/css/manager.css
inflating: apache-tomcat-8.5.60/webapps/manager/images/asf-logo.svg
inflating: apache-tomcat-8.5.60/webapps/manager/images/tomcat.svg
inflating: apache-tomcat-8.5.60/webapps/manager/index.jsp
inflating: apache-tomcat-8.5.60/webapps/manager/status.xsd
inflating: apache-tomcat-8.5.60/webapps/manager/xform.xsl
ubuntu@ServidorREST:~$ ls
apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip
ubuntu@ServidorREST:~$ cd apache-tomcat-8.5.60/
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ ls
BUILDING.txt  LICENSE  README.md  RUNNING.txt  conf  logs  webapps
CONTRIBUTING.md  NOTICE  RELEASE-NOTES  bin  lib  temp  work
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ rmdir webapps/
rmdir: failed to remove 'webapps/': Directory not empty
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ rm -rfv webapps/
removed 'webapps/ROOT/favicon.ico'
removed 'webapps/ROOT/tomcat.svg'
removed 'webapps/ROOT/bg-button.png'
removed 'webapps/ROOT/tomcat.css'
removed 'webapps/ROOT/bg-upper.png'
removed 'webapps/ROOT/WEB-INF/web.xml'
removed directory 'webapps/ROOT/WEB-INF'
removed 'webapps/ROOT/bg-middle.png'
removed 'webapps/ROOT/RELEASE-NOTES.txt'
removed 'webapps/ROOT/bg-nav.png'
removed 'webapps/ROOT/index.jsp'
removed 'webapps/ROOT/asf-logo-wide.svg'
removed directory 'webapps/ROOT'
removed 'webapps/docs/rewrite.html'
removed 'webapps/docs/aio.html'
```

Imagen 6: Eliminación del directorio "webapps".

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>

Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.

ste... > Tarea7-Servidor-... >				Buscar en Tarea7-Servidor-Web	
Nombre	Fecha de modificación	Tipo	Tamaño		
capturas	25/11/2020 08:38 p. m.	Carpeta de archivos			
~Starea7.docx	26/11/2020 02:17 p. m.	Documento de Mi...	1 KB		
apache-tomcat-8.5.60.zip	25/11/2020 01:20 p. m.	Archivo WinRAR Z...	10,753 KB		
comandos.txt	25/11/2020 01:45 p. m.	Documento de te...	4 KB		
gson-2.3.1.jar	25/11/2020 01:23 p. m.	Executable Jar File	206 KB		
jaxrs-ri-2.24.zip	25/11/2020 01:21 p. m.	Archivo WinRAR Z...	4,821 KB		
mysql-connector-java-8.0.22.zip	25/11/2020 01:25 p. m.	Archivo WinRAR Z...	4,632 KB		
prueba.html	25/11/2020 08:30 p. m.	Microsoft Edge H...	9 KB		
Servicio.zip	25/11/2020 01:40 p. m.	Archivo WinRAR Z...	6 KB		
tarea7.docx	26/11/2020 02:17 p. m.	Documento de Mi...	0 KB		
usuario_sin_foto.png	25/11/2020 01:41 p. m.	Archivo PNG	2 KB		
WSClient.js	25/11/2020 01:42 p. m.	Archivo JavaScript	4 KB		

Imagen 7: Directorio con los archivos descargados para la práctica

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

```

ubuntu@ServidorREST: ~/apache-tomcat-8.5.60/lib
ubuntu@ServidorREST:~/jaxrs-ri$ cd lib/
ubuntu@ServidorREST:~/jaxrs-ri/lib$ ls
jersey-client.jar  jersey-container-servlet-core.jar  jersey-media-jaxb.jar
jersey-common.jar  jersey-container-servlet.jar  jersey-server.jar
ubuntu@ServidorREST:~/jaxrs-ri/lib$ cp *.jar /home/ubuntu/apache-tomcat-8.5.60/lib
ubuntu@ServidorREST:~/jaxrs-ri/lib$ cd ..
ubuntu@ServidorREST:~/jaxrs-ri$ cd ..
ubuntu@ServidorREST:~$ ls
apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip  jaxrs-ri  jaxrs-ri-2.24.zip
ubuntu@ServidorREST:~$ cd apache-tomcat-8.5.60/
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ ls
BUILDING.txt  LICENSE  README.md  RUNNING.txt  conf  logs  webapps
CONTRIBUTING.md  NOTICE  RELEASE-NOTES  bin  lib  temp  work
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ cd lib
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$ ls
annotations-api.jar  jersey-guava-2.24.jar
aopalliance-repackaged-2.5.0-b05.jar  jersey-media-jaxb.jar
catalina-ant.jar  jersey-server.jar
catalina-ha.jar  jsp-api.jar
catalina-storeconfig.jar  org.osgi.core-4.2.0.jar
catalina-tribes.jar  osgi-resource-locator-1.0.1.jar
catalina.jar  persistence-api-1.0.jar
ecj-4.6.3.jar  servlet-api.jar
el-api.jar  tomcat-api.jar
hk2-api-2.5.0-b05.jar  tomcat-coyote.jar
hk2-locator-2.5.0-b05.jar  tomcat-dbcp.jar
hk2-utils-2.5.0-b05.jar  tomcat-i18n-de.jar
jasper-el.jar  tomcat-i18n-es.jar
jasper.jar  tomcat-i18n-fr.jar
jaspic-api.jar  tomcat-i18n-ja.jar
javassist-3.20.0-GA.jar  tomcat-i18n-ko.jar
javax.annotation-api-1.2.jar  tomcat-i18n-ru.jar
javax.inject-2.5.0-b05.jar  tomcat-i18n-zh-CN.jar
javax.servlet-api-3.0.1.jar  tomcat-jdbc.jar
javax.ws.rs-api-2.0.1.jar  tomcat-jni.jar
jaxb-api-2.2.7.jar  tomcat-util-scan.jar
jersey-client.jar  tomcat-util.jar
jersey-common.jar  tomcat-websocket.jar
jersey-container-servlet-core.jar  validation-api-1.1.0.Final.jar
jersey-container-servlet.jar  websocket-api.jar
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$ rm javax.servlet-api-3.0.1.jar

```

Imagen 7: copia de todos los archivos .jar a la carpeta ~/apache-tomcat-8.5.60/lib

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).

```
ubuntu@ServidorREST:~/jaxrs-ri/lib$ cd ..
ubuntu@ServidorREST:~/jaxrs-ri$ cd ..
ubuntu@ServidorREST:~$ ls
apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip  jaxrs-ri  jaxrs-ri-2.24.zip
ubuntu@ServidorREST:~$ cd apache-tomcat-8.5.60/
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ ls
BUILDING.txt  LICENSE  README.md  RUNNING.txt  conf  logs  webapps
CONTRIBUTING.md  NOTICE  RELEASE-NOTES  bin  lib  temp  work
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ cd lib
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$ ls
annotations-api.jar          jersey-guava-2.24.jar
aopalliance-repackaged-2.5.0-b05.jar  jersey-media-jaxb.jar
catalina-ant.jar             jersey-server.jar
catalina-ha.jar              jsp-api.jar
catalina-storeconfig.jar      org.osgi.core-4.2.0.jar
catalina-tribes.jar          osgi-resource-locator-1.0.1.jar
catalina.jar                  persistence-api-1.0.jar
ecj-4.6.3.jar                 servlet-api.jar
el-api.jar                    tomcat-api.jar
hk2-api-2.5.0-b05.jar          tomcat-coyote.jar
hk2-locator-2.5.0-b05.jar      tomcat-dbcp.jar
hk2-utils-2.5.0-b05.jar       tomcat-il8n-de.jar
jasper-el.jar                 tomcat-il8n-es.jar
jasper.jar                    tomcat-il8n-fr.jar
jaspic-api.jar                tomcat-il8n-ja.jar
javassist-3.20.0-GA.jar        tomcat-il8n-ko.jar
javax.annotation-api-1.2.jar   tomcat-il8n-ru.jar
javax.inject-2.5.0-b05.jar     tomcat-il8n-zh-CN.jar
javax.servlet-api-3.0.1.jar    tomcat-jdbc.jar
javax.ws.rs-api-2.0.1.jar      tomcat-jni.jar
jaxb-api-2.2.7.jar             tomcat-util-scan.jar
jersey-client.jar              tomcat-util.jar
jersey-common.jar              tomcat-websocket.jar
jersey-container-servlet-core.jar  validation-api-1.1.0.Final.jar
jersey-container-servlet.jar    websocket-api.jar
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$ rm javax.servlet-api-3.0.1.jar
```

Imagen 8: eliminación del archivo javax.servlet-api-3.0.1.jar del directorio ~/apache-tomcat-8.5.60/lib

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>

Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.

Tarea7-Servidor-Web			
Buscar en Tarea7-Servidor-Web			
Nombre	Fecha de modificación	Tipo	Tamaño
capturas	25/11/2020 08:38 p. m.	Carpeta de archivos	
~Starea7.docx	26/11/2020 02:17 p. m.	Documento de Mi...	1 KB
apache-tomcat-8.5.60.zip	25/11/2020 01:20 p. m.	Archivo WinRAR Z...	10,753 KB
comandos.txt	25/11/2020 01:45 p. m.	Documento de te...	4 KB
gson-2.3.1.jar	25/11/2020 01:23 p. m.	Executable Jar File	206 KB
jaxrs-ri-2.24.zip	25/11/2020 01:21 p. m.	Archivo WinRAR Z...	4,821 KB
mysql-connector-java-8.0.22.zip	25/11/2020 01:25 p. m.	Archivo WinRAR Z...	4,632 KB
prueba.html	25/11/2020 08:30 p. m.	Microsoft Edge H...	9 KB
Servicio.zip	25/11/2020 01:40 p. m.	Archivo WinRAR Z...	6 KB
tarea7.docx	26/11/2020 02:17 p. m.	Documento de Mi...	0 KB
usuario_sin_foto.png	25/11/2020 01:41 p. m.	Archivo PNG	2 KB
WSCClient.js	25/11/2020 01:42 p. m.	Archivo JavaScript	4 KB

Imagen 9: Directorio con los archivos descargados para la práctica

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

Se utilizó el programa portable “WinSCP” para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.

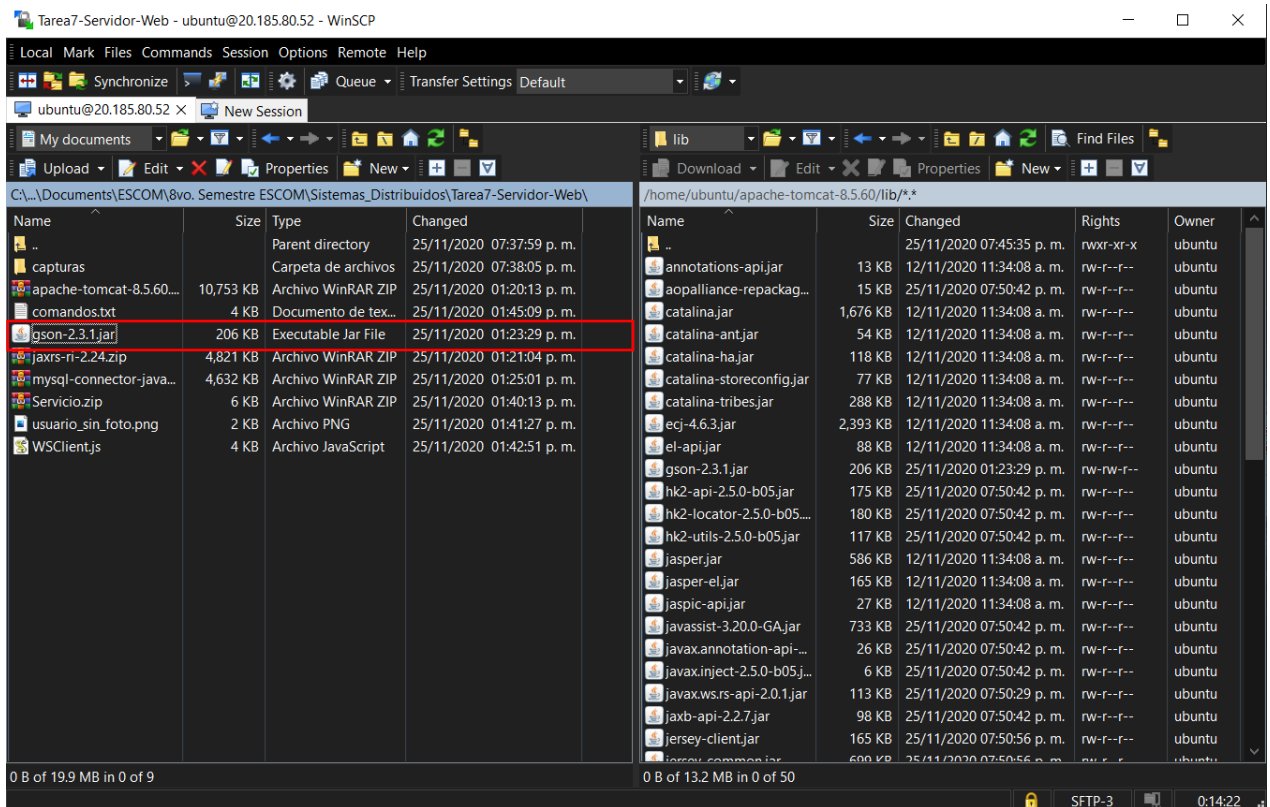


Imagen 9: copia del archivo “gson-2.3.1.jar” al directorio “lib” de Tomcat

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.

Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.

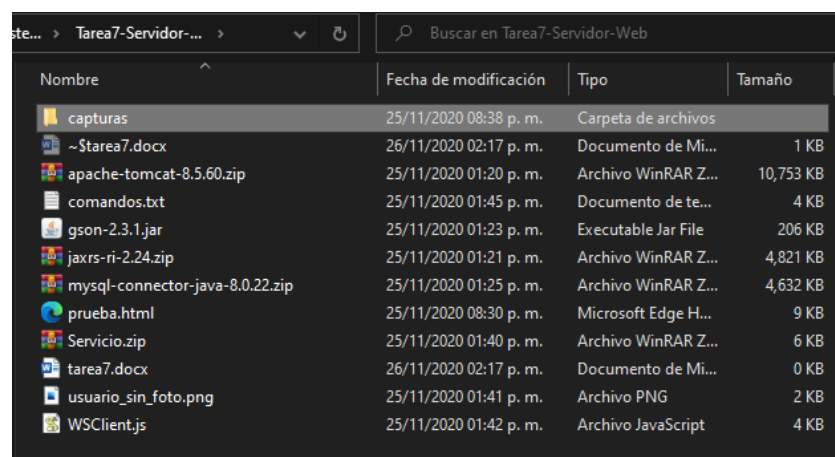


Imagen 10: Directorio con los archivos descargados para la práctica

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

```
ubuntu@ServidorREST: ~  
hk2-utils-2.5.0-b05.jar      tomcat-i18n-es.jar  
jasper-el.jar               tomcat-i18n-fr.jar  
jasper.jar                  tomcat-i18n-ja.jar  
jaspic-api.jar              tomcat-i18n-ko.jar  
javassist-3.20.0-GA.jar     tomcat-i18n-ru.jar  
javax.annotation-api-1.2.jar tomcat-i18n-zh-CN.jar  
javax.inject-2.5.0-b05.jar  tomcat-jdbc.jar  
javax.ws.rs-api-2.0.1.jar   tomcat-jni.jar  
jaxb-api-2.2.7.jar          tomcat-util-scan.jar  
jersey-client.jar           tomcat-util.jar  
jersey-common.jar           tomcat-websocket.jar  
jersey-container-servlet-core.jar validation-api-1.1.0.Final.jar  
jersey-container-servlet.jar websocket-api.jar  
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$ cd ..  
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ cd ..  
ubuntu@ServidorREST:~$ pwd  
/home/ubuntu  
ubuntu@ServidorREST:~$ ls  
apache-tomcat-8.5.60  jaxrs-ri  mysql-connector-java-8.0.22.zip  
apache-tomcat-8.5.60.zip  jaxrs-ri-2.24.zip  
ubuntu@ServidorREST:~$ 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/
```

Imagen 10: Desempaquetado del archivo "mysql-connector-java-8.0.22.zip"

```
ubuntu@ServidorREST: ~/apache-tomcat-8.5.60/lib  
ixSessionTest.java  
  inflating: mysql-connector-java-8.0.22/src/test/java/testsuite/x/internal/XPro  
tocolAsyncTest.java  
  inflating: mysql-connector-java-8.0.22/src/test/java/testsuite/x/internal/XPro  
tocolAuthTest.java  
  inflating: mysql-connector-java-8.0.22/src/test/java/testsuite/x/internal/XPro  
tocolTest.java  
  inflating: mysql-connector-java-8.0.22/src/test/java/testsuite/x/internal/pack  
age-info.java  
ubuntu@ServidorREST:~$ ls  
apache-tomcat-8.5.60  jaxrs-ri  mysql-connector-java-8.0.22  
apache-tomcat-8.5.60.zip  jaxrs-ri-2.24.zip  mysql-connector-java-8.0.22.zip  
ubuntu@ServidorREST:~$ cd mysql-connector-java-8.0.22/  
ubuntu@ServidorREST:~/mysql-connector-java-8.0.22$ ls  
CHANGES  INFO  BIN  INFO  SRC  LICENSE  README  build.xml  mysql-connector-java-8.0.22.jar  src  
ubuntu@ServidorREST:~/mysql-connector-java-8.0.22$ cp *.jar /home/ubuntu/apache-tomcat-8.5.60/lib  
ubuntu@ServidorREST:~/mysql-connector-java-8.0.22$ cd ~/home/ubuntu/apache-tomcat-8.5.60  
-bash: cd: /home/ubuntu/home/ubuntu/apache-tomcat-8.5.60: No such file or directory  
ubuntu@ServidorREST:~/mysql-connector-java-8.0.22$ cd ~/ubuntu/apache-tomcat-8.5.60  
-bash: cd: /home/ubuntu/ubuntu/apache-tomcat-8.5.60: No such file or directory  
ubuntu@ServidorREST:~/mysql-connector-java-8.0.22$ cd ..  
ubuntu@ServidorREST:~$ cd apache-tomcat-8.5.60/  
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ cd lib/  
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$ ls  
annotations-api.jar      javax.inject-2.5.0-b05.jar  tomcat-api.jar  
apollon-repackaged-2.5.0-b05.jar  javax.ws.rs-api-2.0.1.jar  tomcat-coyote.jar  
catalina-ant.jar          jaxb-api-2.2.7.jar         tomcat-dbc.jar  
catalina-ha.jar           jersey-client.jar          tomcat-i18n-de.jar  
catalina-storeconfig.jar  jersey-common.jar          tomcat-i18n-es.jar  
catalina-tribes.jar       jersey-container-servlet-core.jar tomcat-i18n-fr.jar  
catalina.jar              jersey-container-servlet.jar tomcat-i18n-ja.jar  
ecj-4.6.3.jar             jersey-guava-2.24.jar      tomcat-i18n-ko.jar  
el-api.jar                jersey-media-jaxb.jar      tomcat-i18n-ru.jar  
gson-2.3.1.jar            jersey-server.jar          tomcat-i18n-zh-CN.jar  
hk2-api-2.5.0-b05.jar     jsp-api.jar               tomcat-jdbc.jar  
hk2-locator-2.5.0-b05.jar  mysql-connector-java-8.0.22.jar tomcat-jni.jar  
hk2-utils-2.5.0-b05.jar   org.osgi.core-4.2.0.jar    tomcat-util-scan.jar  
jasper-el.jar             osgi-resource-locator-1.0.1.jar tomcat-util.jar  
jasper.jar                persistence-api-1.0.jar    tomcat-websocket.jar  
jaspic-api.jar            servlet-api.jar            validation-api-1.1.0.Final.jar  
javassist-3.20.0-GA.jar   websocket-api.jar  
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$
```

Imagen 11: copia del archivo "mysql-connector...jar" al directorio "lib" de Tomcat.

Iniciar/detener el servidor Tomcat

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

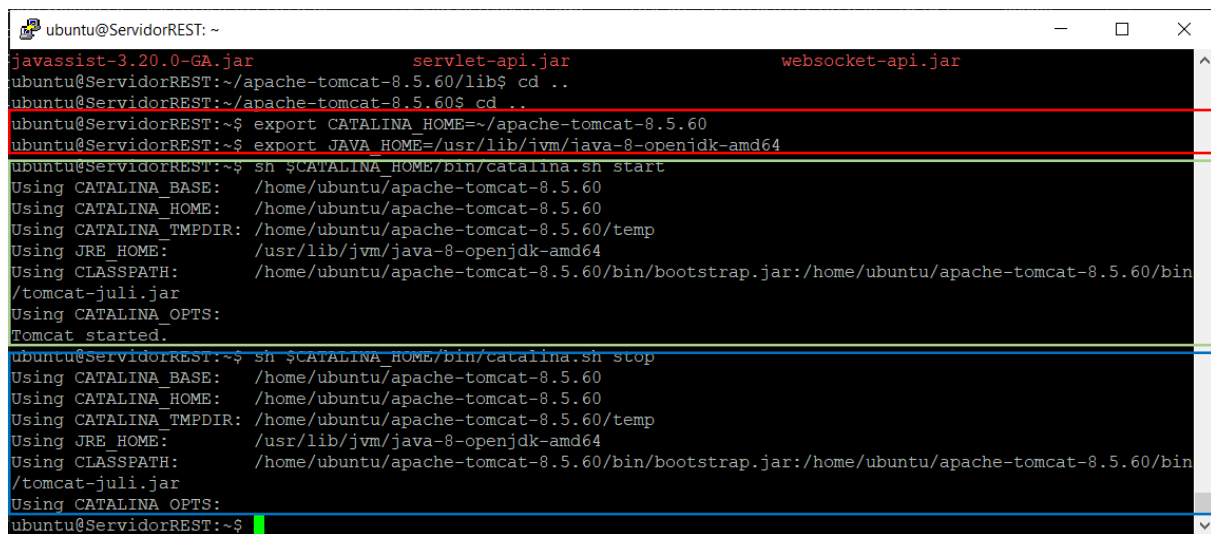
export CATALINA_HOME=aquí va la ruta del directorio de Tomcat 8
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
2. Iniciar la ejecución de Tomcat ejecutando el siguiente comando:

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

3. Para detener la ejecución de Tomcat se deberá ejecutar el siguiente comando:

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

Notar que Tomcat se ejecuta sin permisos de administrador (no se usa "sudo"), lo cual es muy importante para prevenir que algún atacante pueda entrar a nuestro sistema con permisos de super-usuario.



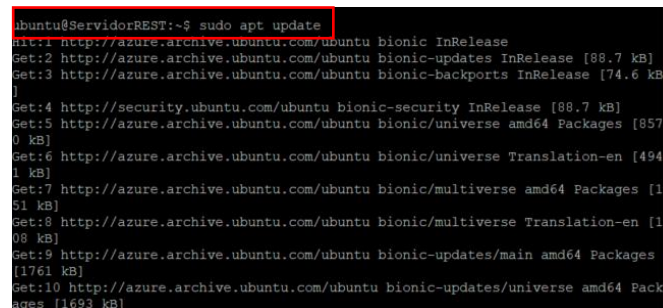
```
ubuntu@ServidorREST: ~  
javassist-3.20.0-GA.jar      servlet-api.jar      websocket-api.jar  
ubuntu@ServidorREST:~/apache-tomcat-8.5.60/lib$ cd ..  
ubuntu@ServidorREST:~/apache-tomcat-8.5.60$ cd ..  
ubuntu@ServidorREST:~$ export CATALINA_HOME=~/apache-tomcat-8.5.60  
ubuntu@ServidorREST:~$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
ubuntu@ServidorREST:~$ sh $CATALINA_HOME/bin/catalina.sh start  
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-8.5.60  
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-8.5.60  
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-8.5.60/temp  
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64  
Using CLASSPATH:       /home/ubuntu/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/ubuntu/apache-tomcat-8.5.60/bin  
/tomcat-juli.jar  
Using CATALINA_OPTS:  
Tomcat started.  
ubuntu@ServidorREST:~$ sh $CATALINA_HOME/bin/catalina.sh stop  
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-8.5.60  
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-8.5.60  
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-8.5.60/temp  
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64  
Using CLASSPATH:       /home/ubuntu/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/ubuntu/apache-tomcat-8.5.60/bin  
/tomcat-juli.jar  
Using CATALINA_OPTS:  
ubuntu@ServidorREST:~$
```

Imagen 12: Inicialización de variables de entorno y ejecución de los comandos “start” y “stop”.

Instalación de MySQL

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

```
sudo apt update
```



```
ubuntu@ServidorREST:~$ 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 [857  
0 kB]  
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic/universe Translation-en [494  
1 kB]  
Get:7 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [1  
51 kB]  
Get:8 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [1  
08 kB]  
Get:9 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages  
[1761 kB]  
Get:10 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Pack  
ages [1693 kB]
```

Imagen 113: Ejecución del comando “sudo apt update” y “sudo apt install openjdk-8-jdk-headless”

2. Instalar el paquete default de MySQL:

```
sudo apt install mysql-server
```

3. Ejecutar el script de seguridad:

```
sudo mysql_secure_installation
```

Press y|Y for Yes, any other key for No: **N**

New password: *contraseña-de-root-en-mysql*

Re-enter new password: *contraseña-de-root-en-mysql*

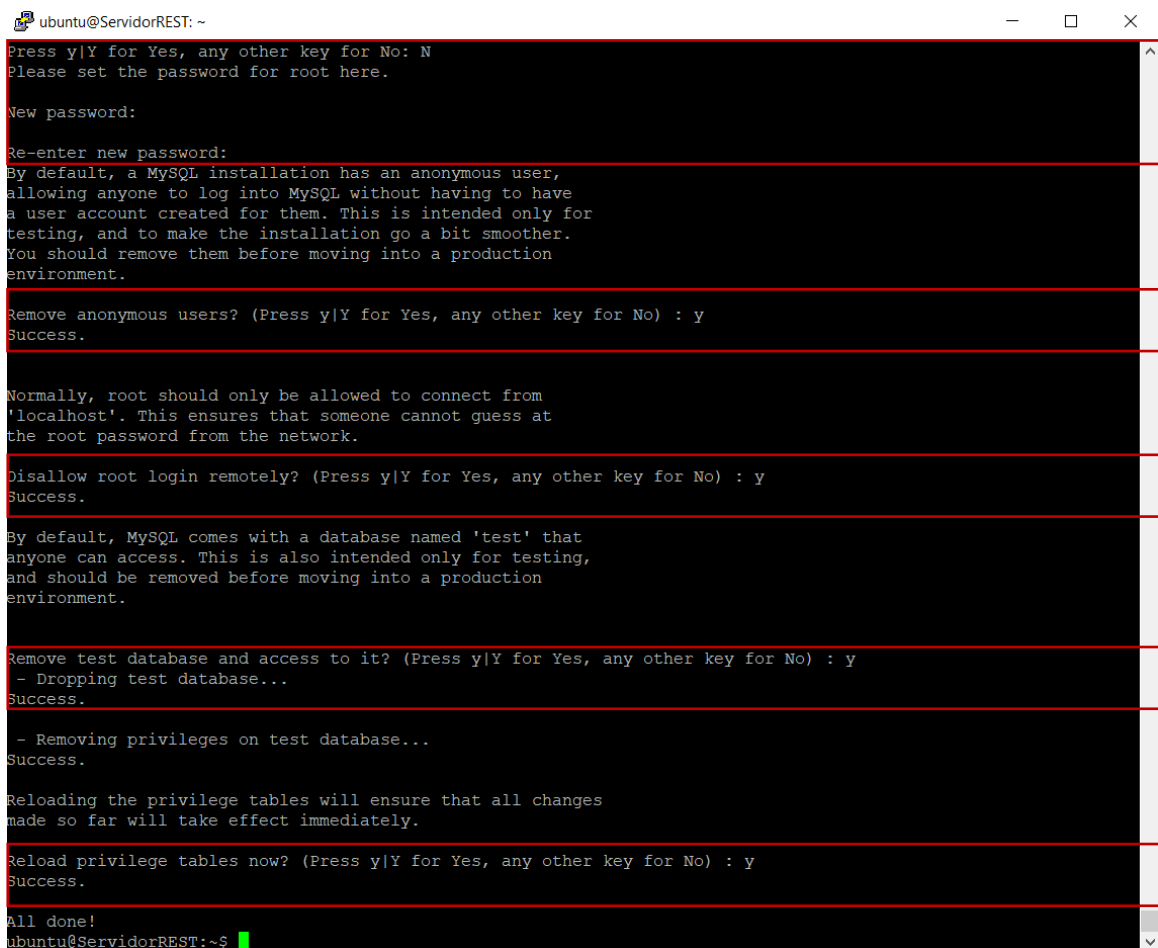
Remove anonymous users? (Press y|Y for Yes, any other key for No) : **Y**

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : **Y**

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : **Y**

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : **Y**

Se utilizó como contraseña "root" tanto para el root como para el usuario "hugo" que posteriormente se crea.



```
ubuntu@ServidorREST: ~  
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!  
ubuntu@ServidorREST:~$
```

Imagen 14: ejecución del comando "sudo mysql secure installation"

4. Ejecutar el monitor de MySQL:

```
sudo mysql
```

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

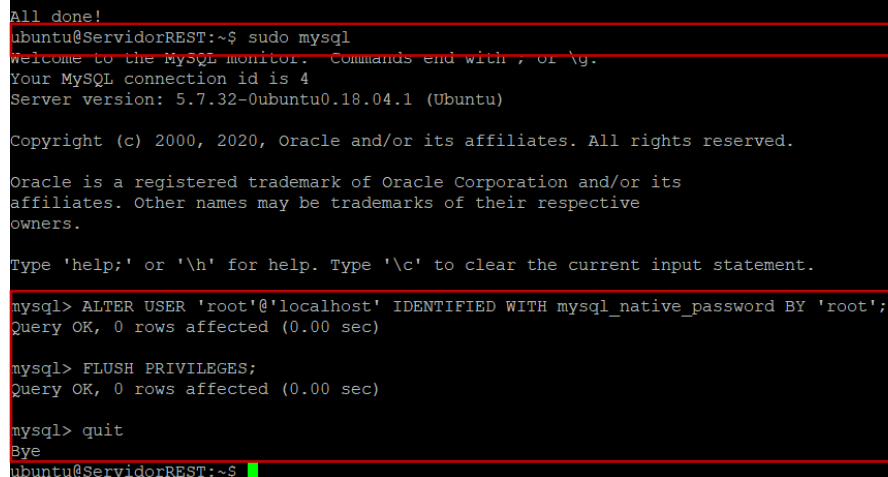
```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'contraseña-de-root-en-mysql';
```

6. Actualizar los privilegios:

```
FLUSH PRIVILEGES;
```

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

```
quit
```



```
All done!  
ubuntu@ServidorREST:~$ 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> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'root';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> quit  
Bye  
ubuntu@ServidorREST:~$
```

Imagen 15: modificación de la contraseña del usuario root

Crear un usuario en MySQL

1. Ejecutar el monitor de MySQL:

```
mysql -u root -p
```

2. Crea el usuario "hugo":

```
create user hugo@localhost identified by 'contraseña-del-usuario-hugo';
```

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

```
grant all on servicio_web.* to hugo@localhost;
```

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

```
quit
```

```
ubuntu@ServidorREST: ~  
mysql> quit  
Bye  
ubuntu@ServidorREST:~$ 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> create user hugo@localhost identified by 'root';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> grant all on servicio_web.* to hugo@localhost;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> quit  
Bye  
ubuntu@ServidorREST:~$
```

Imagen 16: Creación del usuario "hugo".

Crear la base de datos

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

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

2. Crear la base de datos "servicio_web":

```
create database servicio_web;
```

3. Conectar a la base de datos creada anteriormente:

```
use servicio_web;
```

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)
```

```
);
```

```
create table fotos_usuarios
```

```
(  
    id_foto integer auto_increment primary key,  
    foto longblob,  
    id_usuario integer not null
```

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

5. Salir del monitor de MySQL:

```
quit
```

```
ubuntu@ServidorREST: ~  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> create database servicio_web;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> use servicio_web;  
Database changed  
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)  
-> );  
alter table fotos_usuarios add foreign key (id_usuario) references usuarios(id_usuario);  
create unique index usuarios_1 on usuarios(email);  
Query OK, 0 rows affected (0.08 sec)  
  
mysql> create table fotos_usuarios  
-> (  
->   id_foto integer auto_increment primary key,  
->   foto longblob,  
->   id_usuario integer not null  
-> );  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> alter table fotos_usuarios add foreign key (id_usuario) references usuarios(id_usuario);  
Query OK, 0 rows affected (0.18 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> create unique index usuarios_1 on usuarios(email);  
Query OK, 0 rows affected (0.06 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> quit  
Bye  
ubuntu@ServidorREST:~$
```

Imagen 17: Creación de la base de datos y la tabla "usuarios"

Compilar, empaquetar y desplegar el servicio web

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

```
mysql> create unique index usuarios_1 on usuarios(email);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> quit
Bye

ubuntu@ServidorREST:~$ 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
ubuntu@ServidorREST:~$ 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
ubuntu@ServidorREST:~$ ls
```

Imagen 18: Desempaquetado del archivo "Servicio.zip"

2. Definir la variable de ambiente CATALINA_HOME:

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

La variable CATALINA_HOME ya se había definido anteriormente, por lo que se confirmó su ruta realizando el comando "cat \$CATALINA_HOME"

```
ubuntu@ServidorREST:~$ ls
META-INF  apache-tomcat-8.5.60  jaxrs-ri-2.24.zip  negocio
Servicio.zip  apache-tomcat-8.5.60.zip  mysql-connector-java-8.0.22
WEB-INF    jaxrs-ri              mysql-connector-java-8.0.22.zip
ubuntu@ServidorREST:~$ cat $CATALINA_HOME
cat: /home/ubuntu/apache-tomcat-8.5.60: Is a directory
ubuntu@ServidorREST:~$ cd negocio/
```

Imagen 19: Se muestra la ruta guardada en la variable CATALINA_HOME

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

4. Compilar la clase Servicio.java:

javac -cp \$CATALINA_HOME/lib/javawebsocket-api-2.0.1.jar:\$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java

```
ubuntu@ServidorREST:~$ cd negocio/
ubuntu@ServidorREST:~/negocio$ ls
AdaptadorGsonBase64.java  Error.java  Foto.java  Servicio.java  Usuario.java
ubuntu@ServidorREST:~/negocio$ javac -cp $CATALINA_HOME/lib/javawebsocket-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
javac: file not found: negocio/Servicio.java
Usage: javac <options> <source files>
Use -help for a list of possible options
ubuntu@ServidorREST:~/negocio$ cd ..
ubuntu@ServidorREST:~$ javac -cp $CATALINA_HOME/lib/javawebsocket-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
ubuntu@ServidorREST:~$
```

Imagen 20: Compilación de la clase "Servicio.java" dentro del directorio "negocio".

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**.

Se utilizó el editor Nano para poder definir el usuario y el password de la base de datos.

```
ubuntu@ServidorREST: ~  
AdaptadorGsonBase64.java Error.java Foto.java Servicio.java Usuario.java  
ubuntu@ServidorREST:~/negocio$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java  
javac: file not found: negocio/Servicio.java  
Usage: javac <options> <source files>  
use -help for a list of possible options  
ubuntu@ServidorREST:~/negocio$ cd ..  
ubuntu@ServidorREST:~$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java  
ubuntu@ServidorREST:~$ cd META-INF/  
ubuntu@ServidorREST:~/META-INF$ ls  
context.xml  
ubuntu@ServidorREST:~/META-INF$ nano context.xml  
ubuntu@ServidorREST:~/META-INF$ ls
```

Imagen 21: Edición del archivo "context.xml" dentro de la carpeta "META-INF"

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
```

```
ubuntu@ServidorREST: ~  
AdaptadorGsonBase64.java Error.java Foto.java Servicio.java Usuario.java  
ubuntu@ServidorREST:~/negocio$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java  
javac: file not found: negocio/Servicio.java  
Usage: javac <options> <source files>  
use -help for a list of possible options  
ubuntu@ServidorREST:~/negocio$ cd ..  
ubuntu@ServidorREST:~$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java  
ubuntu@ServidorREST:~$ cd META-INF/  
ubuntu@ServidorREST:~/META-INF$ ls  
context.xml  
ubuntu@ServidorREST:~/META-INF$ nano context.xml  
ubuntu@ServidorREST:~/META-INF$ ls  
context.xml  
ubuntu@ServidorREST:~/META-INF$ cd ..  
ubuntu@ServidorREST:~$ ls  
META-INF  apache-tomcat-8.5.60  jaxrs-ri-2.24.zip  negocio  
Servicio.zip  apache-tomcat-8.5.60.zip  mysql-connector-java-8.0.22  
WEB-INF  jaxrs-ri  mysql-connector-java-8.0.22.zip  
ubuntu@ServidorREST:~$ rm WEB-INF/classes/negocio/*  
rm: cannot remove 'WEB-INF/classes/negocio/*': No such file or directory  
ubuntu@ServidorREST:~$ cd WEB-INF/classes/negocio/  
ubuntu@ServidorREST:~/WEB-INF/classes/negocio$ ls  
ubuntu@ServidorREST:~/WEB-INF/classes/negocio$ ls  
ubuntu@ServidorREST:~/WEB-INF/classes/negocio$ cd ~  
ubuntu@ServidorREST:~$ cp negocio/*.class WEB-INF/classes/negocio/.  
ubuntu@ServidorREST:~$ jar cvf Servicio.war WEB-INF META-INF  
added manifest  
adding: WEB-INF/(in = 0) (out= 0) (stored 0%)  
adding: WEB-INF/web.xml(in = 672) (out= 296) (deflated 55%)  
adding: WEB-INF/classes/(in = 0) (out= 0) (stored 0%)  
adding: WEB-INF/classes/negocio/(in = 0) (out= 0) (stored 0%)  
adding: WEB-INF/classes/negocio/Usuario.class(in = 899) (out= 518) (deflated 42%)  
adding: WEB-INF/classes/negocio/AdaptadorGsonBase64.class(in = 1799) (out= 737) (deflated 59%)  
adding: WEB-INF/classes/negocio/Error.class(in = 278) (out= 214) (deflated 23%)  
adding: WEB-INF/classes/negocio/Servicio.class(in = 7578) (out= 3462) (deflated 54%)  
ignoring entry META-INF/  
adding: META-INF/context.xml(in = 304) (out= 212) (deflated 30%)  
ubuntu@ServidorREST:~$ cp Servicio.war apache-tomcat-8.5.60/webapps/  
ubuntu@ServidorREST:~$ cp Servicio.war apache-tomcat-8.5.60/webapps/.  
ubuntu@ServidorREST:~$
```

Imagen 22: Ejecución de los comandos para la creación del servicio web.

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.

```

adding: WEB-INF/classes/negocio/Servicio.class(in = 7578) (out= 3462) (deflated 54%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 304) (out= 212) (deflated 30%)
ubuntu@ServidorREST:~$ cp Servicio.war apache-tomcat-8.5.60/webapps/
ubuntu@ServidorREST:~$ cp Servicio.war apache-tomcat-8.5.60/webapps/.
ubuntu@ServidorREST:~$

```

Imagen 23: Copia del archivo "Servicio.war" al directorio "webapps" de Tomcat

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.

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

Se tuvo que volver a ejecutar el comando "sh \$CATALINA_HOME/bin/catalina.sh start" para iniciar el servicio de Tomcat, ya que se había detenido anteriormente.

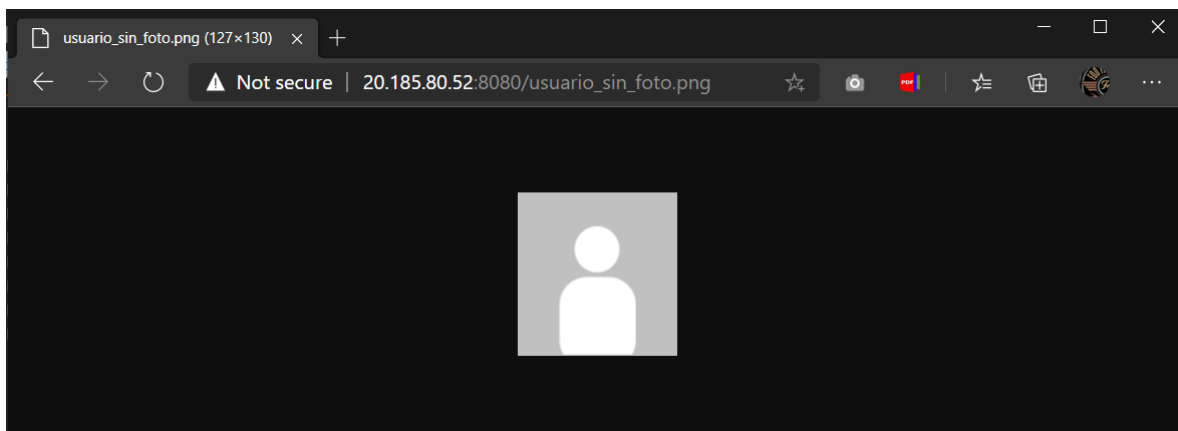


Imagen 24: Captura del servicio Tomcat en línea, mostrando "usuario_sin_foto.png".

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

Se utilizó el programa portable "WinSCP" para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.

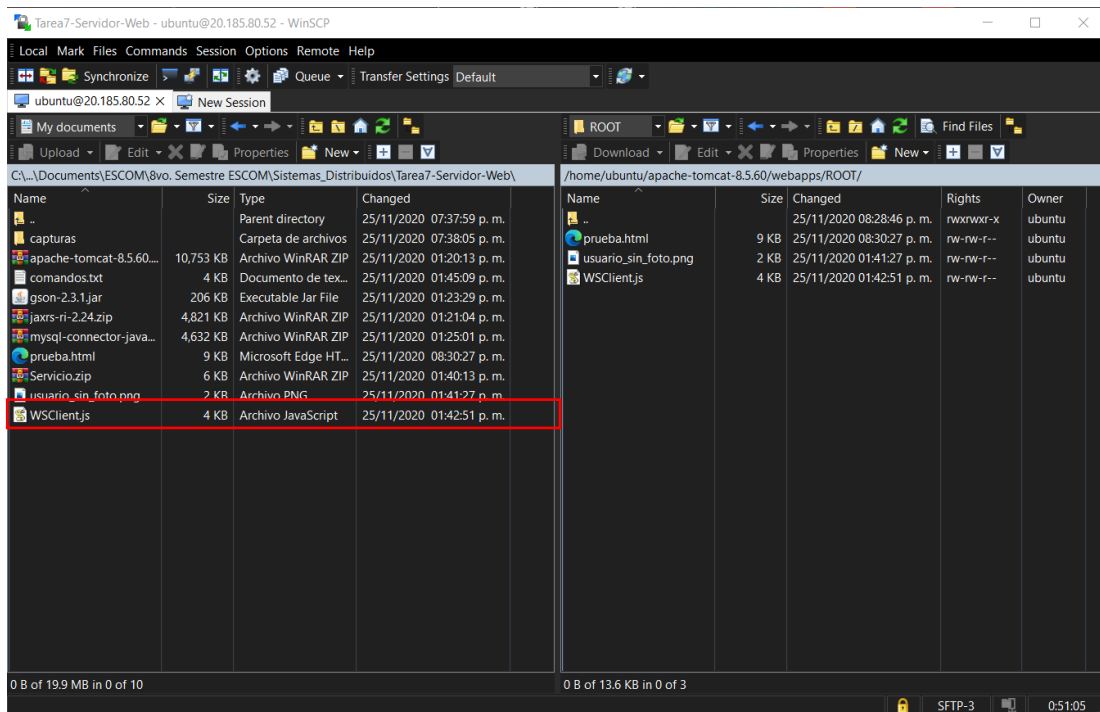


Imagen 24: Transferencia de archivos a la carpeta “webapps/ROOT” de Tomcat de la Maquina Virtual Remota

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

Se utilizó el programa portable “WinSCP” para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.

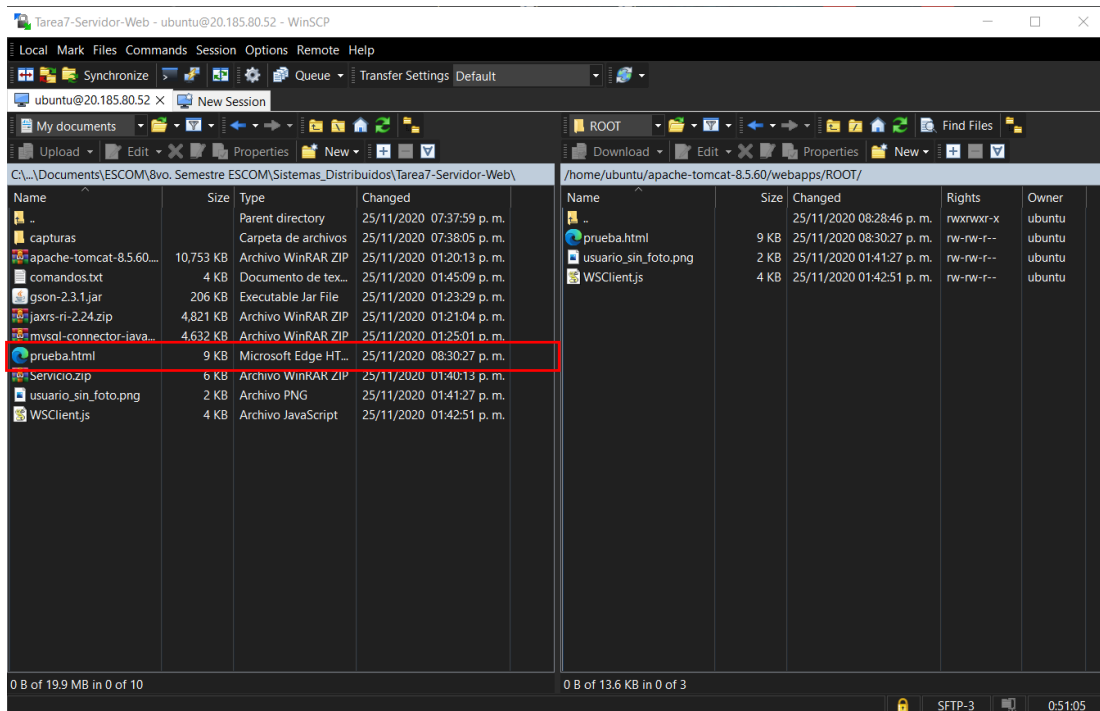


Imagen 25: Transferencia de archivos a la carpeta “webapps/ROOT” de Tomcat de la Maquina Virtual Remota

4. Ingresar la siguiente URL en un navegador:

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



Imagen 26: ingreso a "http://20.185.80.52:8080/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".

A screenshot of a web browser window displaying a form titled "Alta de usuario". The form contains the following fields: "Email *" (with value "eddypereda@gmail.com"), "Nombre *" (with value "Eduardo"), "Apellido paterno *" (with value "Pereda"), "Apellido materno" (with value "Guzmán"), "Fecha de nacimiento *" (with value "02/01/1998"), "Teléfono" (with value "5529552955"), and "Genero" (with value "Masculino"). Below the form is a file upload section with a "Choose Files" button and a file named "fondo_repisa.jpg". At the bottom of the form are two buttons: "Alta" and "Regresa". A modal dialog box is open over the form, displaying the text "20.185.80.52:8080 says" and "OK" with an "OK" button.

Imagen 27: Alta de usuario en el formulario.

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

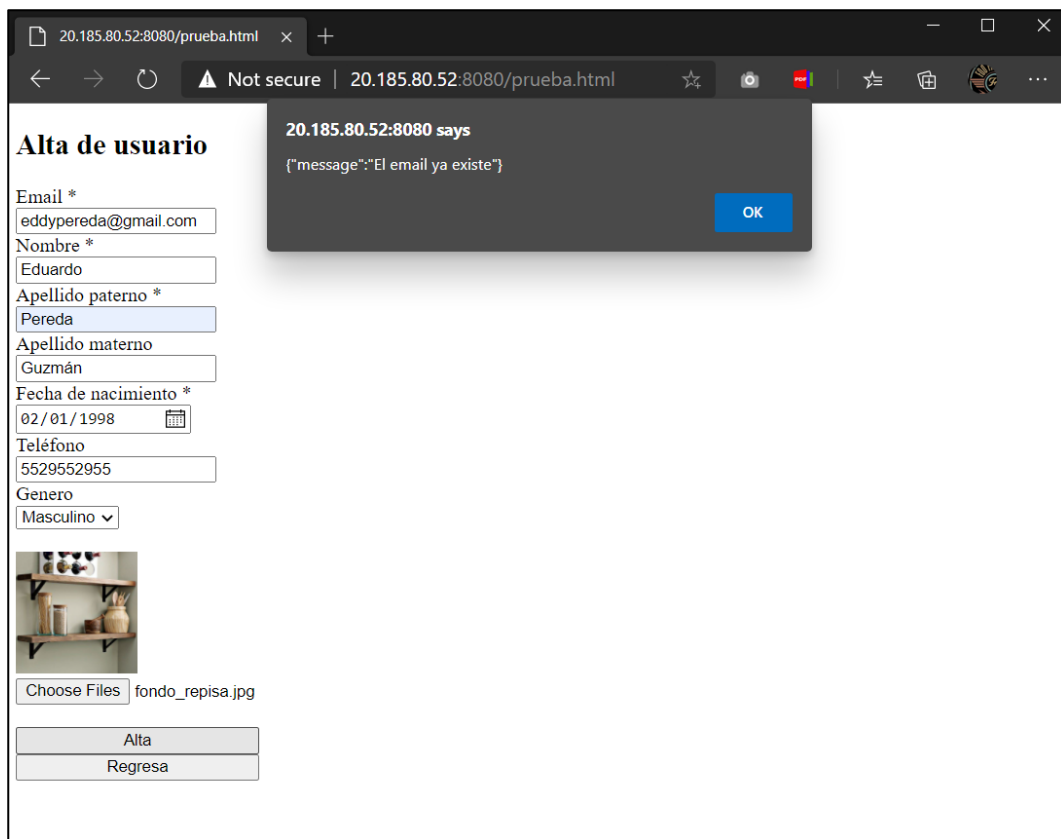


Imagen 28: Alta de usuario en el formulario con error.

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

Una vez que se presionó el botón "Consulta", se rellenaron todos los textbox vacíos, mostrando los datos del usuario ingresado.

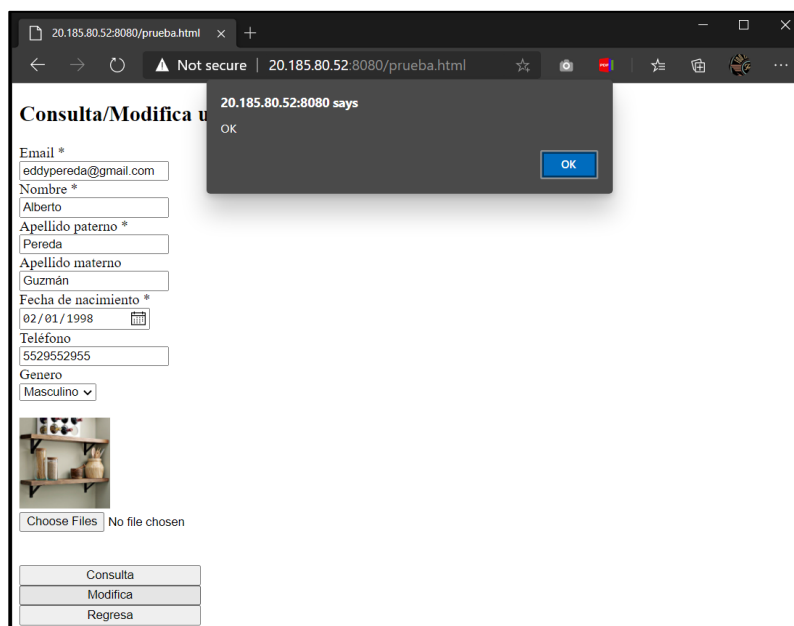


Imagen 28: Consulta y modificación del usuario.

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

Al modificar el usuario, se desplegó un mensaje confirmando la modificación exitosa.

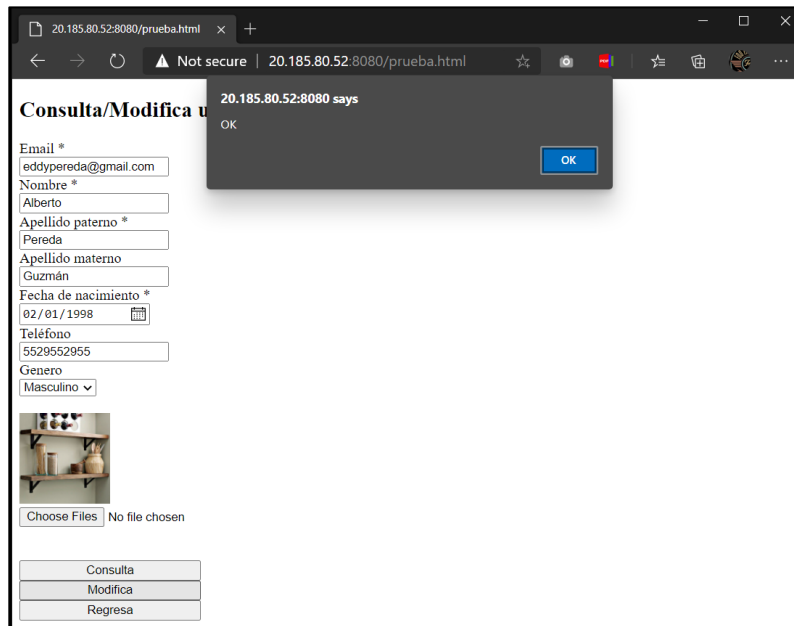


Imagen 29: Consulta y modificación del usuario.

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

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”.

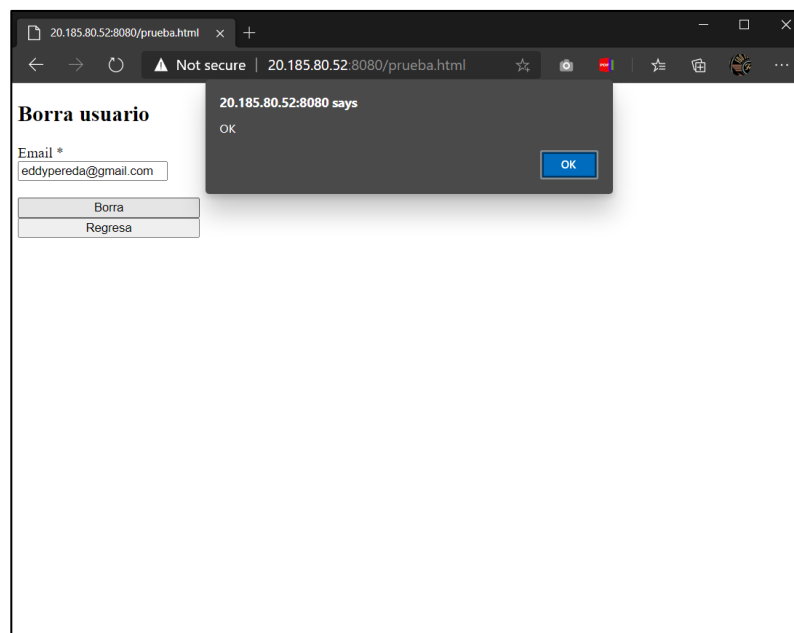


Imagen 30: Eliminación del usuario.

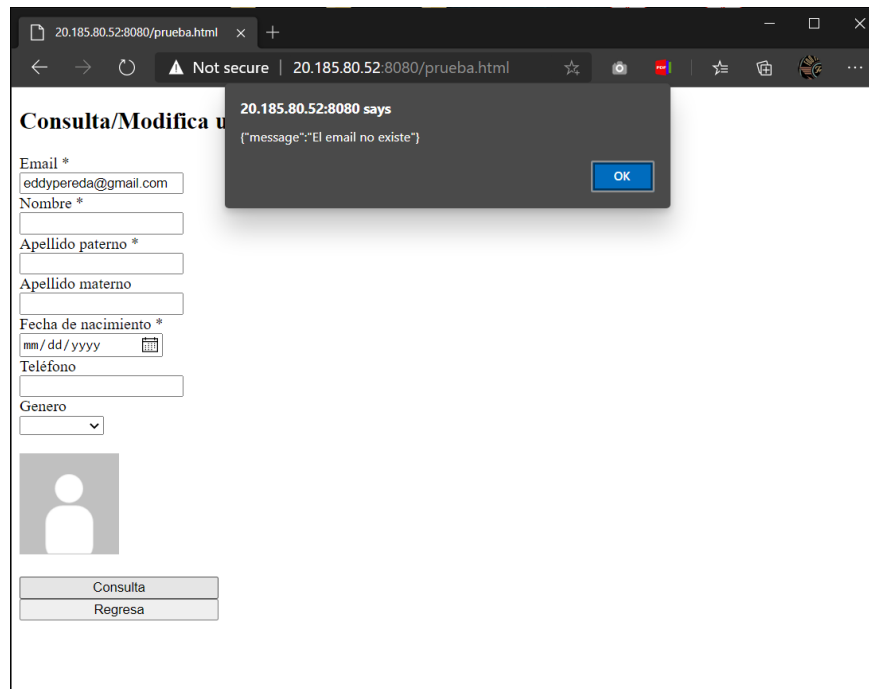


Imagen 31: Intento fallido de consulta de usuario previamente eliminado.

Actividades individuales a realizar

Utilizando un teléfono inteligente y/o una tableta, probar el servicio web accediendo a la siguiente URL en un navegador (Chrome, Firefox, Opera, Safari, etc):

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

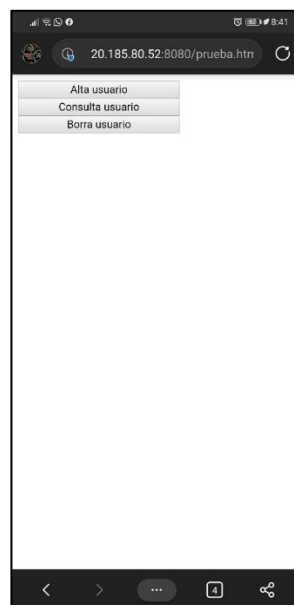


Imagen 32: Acceso al servidor por medio de un smartphone.

Conclusión

El desarrollo de un servicio REST vía HTTP, habilita el uso de los comandos más típicos son GET, POST, PUT/PATCH y DELETE, lo que pone de relieve que su arquitectura solo es apta para la gestión sencilla de datos. Además, el principio de la ausencia de estado en los recursos de REST parece limitar mucho las posibilidades de dicha arquitectura. Sin embargo, hace que la REST interface o interfaz de REST haga posible mucho más que la mera inclusión de conjuntos de datos, tal como en servicios web con procesamiento de transacciones, interpretación de transacciones en el marco de una solicitud, servicios web asincrónicos y servicios web con amplia interoperabilidad.