



IMPLEMENTACION DE UN SERVICIO WEB ESTILO REST

Tarea 7

Raúl Sanchez Rico
Desarrollo de sistemas distribuidos
4CV2

Introducción

Un servicio web es un concepto abstracto que debe ser implementado mediante un agente concreto. El propósito de servicio web es proveer cierta funcionalidad a nombre de su propietario (una persona o una organización). La entidad proveedora es aquella persona u organización que provee un agente que implementa un determinado servicio.

Para entender el concepto básico de un servicio web, se verá cómo crear un servicio web estilo REST utilizando el API de Java JAX-RS sobre el servidor de aplicaciones Tomcat. Primeramente instalaremos Tomcat y las bibliotecas necesarias para la implementación de servicios web estilo REST los cuales podrán acceder una base de datos MySQL.

Desarrollo

Lo primero que se tiene que hacer es el crear una máquina virtual en Azure, esto como ya se ha visto en tareas pasadas, se tiene que hacer desde el portal de Azure (Ilustración 1), tenido en cuenta que se usa una cuenta institucional. Como ya se ha visto a detalle estos pasos, solamente se muestra el resumen de nuestra máquina virtual (Ilustración 2). Una vez aceptado los detalles, se crea nuestra máquina virtual donde se muestran todos los detalles más desglosados que es nuestra máquina virtual (Ilustración 3).

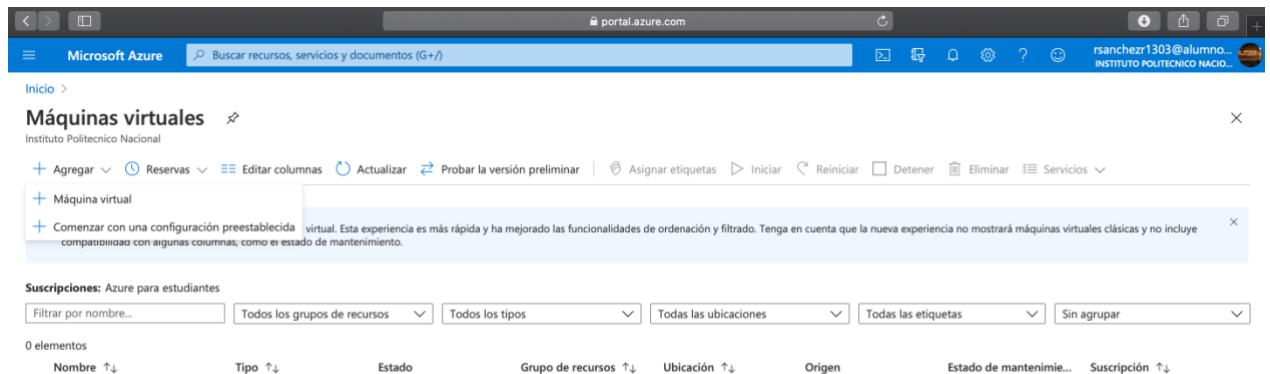


Ilustración 1 Portal de Azure



Ilustración 2 Maquina Virtual Resumen

Ilustración 3 Maquina Virtual Creada

Una vez creado, copiamos la IP de nuestra maquina virtual para poder acceder a ella en los siguientes pasos (Ilustración 4).

Dirección IP pública : **13.65.43.65**

Ilustración 4 IP Maquina Virtual

El siguiente paso es el agregar el puerto de entrada en la pestaña de Red, configuramos el puerto 8080, usando el protocolo TCP (Ilustración 5).

The screenshot shows the Microsoft Azure portal with the URL portal.azure.com. The left sidebar shows the navigation path: Inicio > Tarea7VM > Redes. The main content area displays the 'Redes' configuration for the 'tarea7vm370' interface. It lists several security rules, including:

Prioridad	Nombre	Puerto	Protocolo
300	SSH	22	TCP
65000	AllowVnetInBound	Cualquiera	Cualquier
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquier
65500	DenyAllInBound	Cualquiera	Cualquier

To the right, a modal window titled 'Agregar regla de seguridad de entrada' (Add Network Security Rule) is open. The configuration is as follows:

- Origen:** Any
- Intervalos de puertos de origen:** *
- Destin:** Any
- Intervalos de puertos de destino:** 8080
- Protocolo:** TCP (selected)
- Acción:** Permitir
- Prioridad:** 310
- Nombre:** Port_8080
- Descripción:** (empty)

Agregar button is at the bottom of the modal.

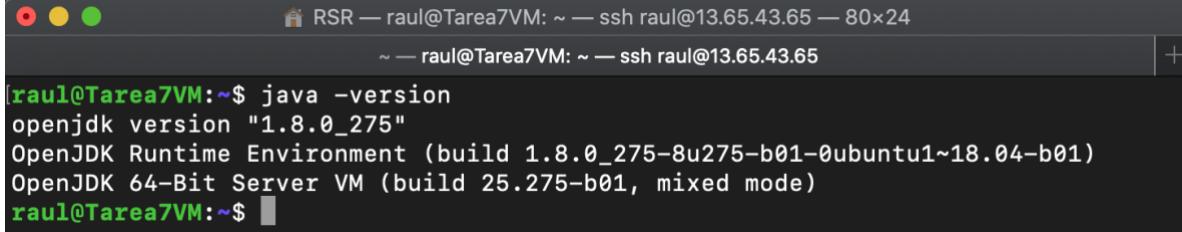
Ilustración 5 Configuración de puerto de red

Una vez creada la maquina virtual, vamos a acceder a ella por ssh desde la terminal, ingresando el comando “`ssh user@IP`”, una vez ingresado a nuestra maquina virtual se mostrará el usuario en consola (Ilustración 6).

```
RSR — raul@Tarea7VM: ~ — ssh raul@13.65.43.65 — 80x24
~ — raul@Tarea7VM: ~ — ssh raul@13.65.43.65
raul@Tarea7VM: ~$
```

Ilustración 6 Maquina Virtual

Ya que ingresamos a nuestra maquina virtual, notamos que no tenemos nada, por ello se tiene que instalar JDK8 con el comando : “sudo apt install openjdk-8-jdk-headless”. Una vez terminando de instalar se mostrar la version para saber que todo esta correcto (Ilustración 7).

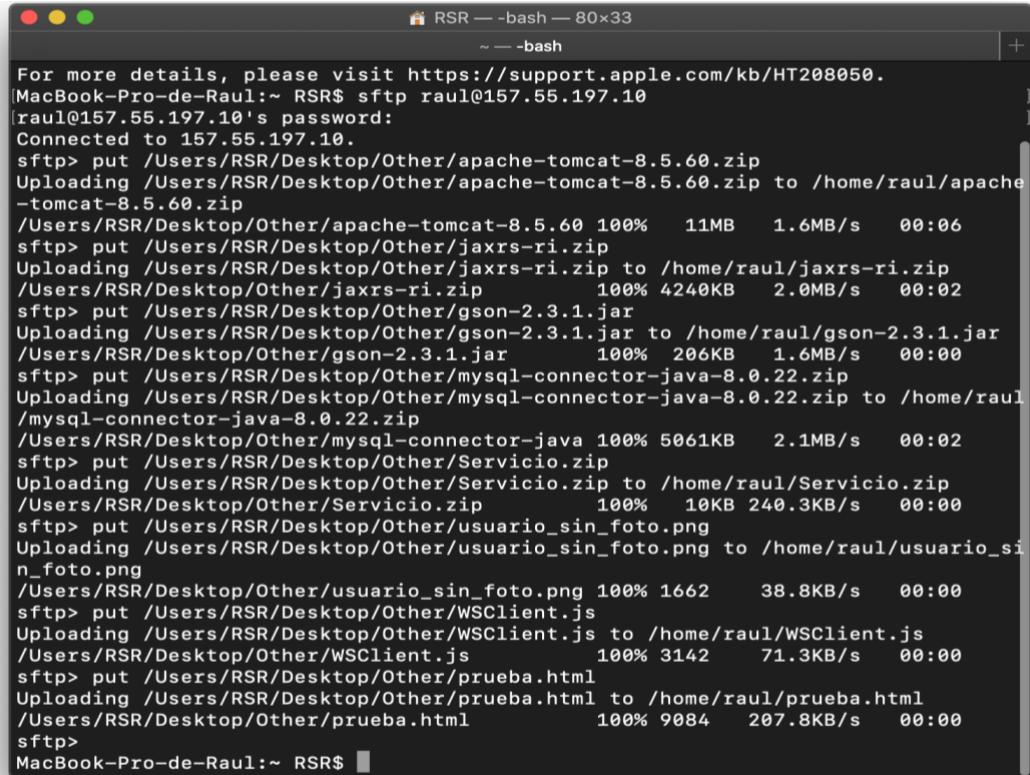


```
RSR — raul@Tarea7VM: ~ — ssh raul@13.65.43.65 — 80x24
~ — raul@Tarea7VM: ~ — ssh raul@13.65.43.65 | +[

raul@Tarea7VM:~$ java -version
openjdk version "1.8.0_275"
OpenJDK Runtime Environment (build 1.8.0_275-8u275-b01-0ubuntu1~18.04-b01)
OpenJDK 64-Bit Server VM (build 25.275-b01, mixed mode)
raul@Tarea7VM:~$
```

Ilustración 7 Instalación JAVA

Una vez instalado el JDK en nuestra maquina virtual se procede a subir los archivos necesarios a ella, con el comando sftp, logramos esto. En la Ilustración 8 se muestra como es que se subieron la mayoría de los archivos necesarios para esta tarea.



```
RSR — -bash — 80x33
~ -- bash | +[

For more details, please visit https://support.apple.com/kb/HT208050.
[MacBook-Pro-de-Raul:~ RSR$ sftp raul@157.55.197.10
raul@157.55.197.10's password:
Connected to 157.55.197.10.
sftp> put /Users/RSR/Desktop/Other/apache-tomcat-8.5.60.zip
Uploading /Users/RSR/Desktop/Other/apache-tomcat-8.5.60.zip to /home/raul/apache-tomcat-8.5.60.zip
/Users/RSR/Desktop/Other/apache-tomcat-8.5.60 100% 11MB 1.6MB/s 00:06
sftp> put /Users/RSR/Desktop/Other/jaxrs-ri.zip
Uploading /Users/RSR/Desktop/Other/jaxrs-ri.zip to /home/raul/jaxrs-ri.zip
/Users/RSR/Desktop/Other/jaxrs-ri.zip 100% 4240KB 2.0MB/s 00:02
sftp> put /Users/RSR/Desktop/Other/gson-2.3.1.jar
Uploading /Users/RSR/Desktop/Other/gson-2.3.1.jar to /home/raul/gson-2.3.1.jar
/Users/RSR/Desktop/Other/gson-2.3.1.jar 100% 206KB 1.6MB/s 00:00
sftp> put /Users/RSR/Desktop/Other/mysql-connector-java-8.0.22.zip
Uploading /Users/RSR/Desktop/Other/mysql-connector-java-8.0.22.zip to /home/raul/mysql-connector-java-8.0.22.zip
/Users/RSR/Desktop/Other/mysql-connector-java 100% 5061KB 2.1MB/s 00:02
sftp> put /Users/RSR/Desktop/Other/Servicio.zip
Uploading /Users/RSR/Desktop/Other/Servicio.zip to /home/raul/Servicio.zip
/Users/RSR/Desktop/Other/Servicio.zip 100% 10KB 240.3KB/s 00:00
sftp> put /Users/RSR/Desktop/Other/usuario_sin_foto.png
Uploading /Users/RSR/Desktop/Other/usuario_sin_foto.png to /home/raul/usuario_sin_foto.png
/Users/RSR/Desktop/Other/usuario_sin_foto.png 100% 1662 38.8KB/s 00:00
sftp> put /Users/RSR/Desktop/Other/WSClient.js
Uploading /Users/RSR/Desktop/Other/WSClient.js to /home/raul/WSClient.js
/Users/RSR/Desktop/Other/WSClient.js 100% 3142 71.3KB/s 00:00
sftp> put /Users/RSR/Desktop/Other/prueba.html
Uploading /Users/RSR/Desktop/Other/prueba.html to /home/raul/prueba.html
/Users/RSR/Desktop/Other/prueba.html 100% 9084 207.8KB/s 00:00
sftp>
MacBook-Pro-de-Raul:~ RSR$
```

Ilustración 8 Comando sftp , subir archivos a Maquina Virtual

Una vez subido nuestro primer archivo, el cual es la distribución de Tomcat 8 se procede a desempacarlo con el comando unzip (Ilustración 9). Una vez hecho esto vemos con el comando ls que ha sido correcto este paso (Ilustración 10).

```
[raul@Tarea7VM:~$ ls  
apache-tomcat-8.5.60.zip  
[raul@Tarea7VM:~$ 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/...]
```

Ilustración 9 Tomcat 8 , comando unzip

```
[raul@Tarea7VM:~$ ls  
__MACOSX apache-tomcat-8.5.60 apache-tomcat-8.5.60.zip  
raul@Tarea7VM:~$ ]
```

Ilustración 10 Tomcat desempacado

Por seguridad se borra el directorio “webapps”(Ilustración 11). Y se crea un nuevo directorio “webapps” con otro directorio dentro llamado “ROOT” (Ilustración 12).

```
[raul@Tarea7VM:~$ ls  
__MACOSX apache-tomcat-8.5.60 apache-tomcat-8.5.60.zip  
[raul@Tarea7VM:~$ cd apache-tomcat-8.5.60  
[raul@Tarea7VM:~/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  
[raul@Tarea7VM:~/apache-tomcat-8.5.60$ rm -r webapps  
[raul@Tarea7VM:~/apache-tomcat-8.5.60$ ls  
BUILDING.txt LICENSE README.md RUNNING.txt conf logs work  
CONTRIBUTING.md NOTICE RELEASE-NOTES bin lib temp  
raul@Tarea7VM:~/apache-tomcat-8.5.60$ ]
```

Ilustración 11 Eliminar directorio webapps

```
[raul@Tarea7VM:~/apache-tomcat-8.5.60$ mkdir webapps  
[raul@Tarea7VM:~/apache-tomcat-8.5.60$ cd webapps/  
[raul@Tarea7VM:~/apache-tomcat-8.5.60/webapps$ mkdir ROOT  
[raul@Tarea7VM:~/apache-tomcat-8.5.60/webapps$ ls  
ROOT
```

Ilustración 12 Creación de directorio webapps/ROOT

Se sube el archivo JAX-RS (Ilustración 13).

```
[raul@Tarea7VM:~$ ls  
__MACOSX apache-tomcat-8.5.60.zip jaxrs-ri.zip  
apache-tomcat-8.5.60 jaxrs-ri  
raul@Tarea7VM:~$ ]
```

Ilustración 13 Archivo 2.24/jaxrs-ri-2.24.zip

Una vez desempaquetado, vamos a copiar todos los archivos “.jar” de todos los directorios al directorio “lib” de Tomcat (Ilustración 14).

```
raul@Tarea7VM:~/apache-tomcat-8.5.60/lib$ pwd
/home/raul/apache-tomcat-8.5.60/lib
raul@Tarea7VM:~/apache-tomcat-8.5.60/lib$ ls
annotations-api.jar           jasper.jar          jersey-media-jaxb.jar      tomcat-i18n-ja.jar
aopalliance-repackaged-2.5.0-b05.jar jaspic-api.jar    jersey-server.jar        tomcat-i18n-ko.jar
catalina-ant.jar               javassist-3.20.0-GA.jar   jsp-api.jar            tomcat-i18n-ru.jar
catalina-ha.jar                javax.annotation-api-1.2.jar org.osgi.core-4.2.0.jar  tomcat-i18n-zh-CN.jar
catalina-storeconfig.jar       javax.inject-2.5.0-b05.jar osgi-resource-locator-1.0.1.jar tomcat-jdbc.jar
catalina-tribes.jar            javax.servlet-api-3.0.1.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                 javax.ws.rs-api-2.0.1.jar  tomcat-api.jar        tomcat-util.jar
el-api.jar                    jersey-client.jar        tomcat-coyote.jar     tomcat-websocket.jar
hk2-api-2.5.0-b05.jar          jersey-common.jar       tomcat-dbcp.jar       validation-api-1.1.0.Final.jar
hk2-locator-2.5.0-b05.jar       jersey-container-servlet-core.jar tomcat-i18n-de.jar   websocket-api.jar
hk2-utils-2.5.0-b05.jar         jersey-container-servlet.jar tomcat-i18n-es.jar
jasper-el.jar                 jersey-guava-2.24.jar     tomcat-i18n-fr.jar
```

Ilustración 14 Archivos lib/*.jar

Borrar el archivo “javax.servelt-api-3.0.1.jar” del directorio “lib” de Tomcat (Ilustración 15). Posteriormente se carga a la maquina virtual el archivo “gson-2.3.1.jar” y moverlo al directorio “lib” de Tomcat (Ilustración 16).

```
raul@Tarea7VM:~/apache-tomcat-8.5.60/lib$ rm javax.servelt-api-3.0.1.jar
```

Ilustración 15 comando para borrar el archioi javax.servelet

```
raul@Tarea7VM:~$ ls
__MACOSX apache-tomcat-8.5.60 apache-tomcat-8.5.60.zip gson-2.3.1.jar jaxrs-ri jaxrs-ri.zip
raul@Tarea7VM:~$ mv gson-2.3.1.jar /home/raul/apache-tomcat-8.5.60/lib
raul@Tarea7VM:~$
```

Ilustración 16 Mover archvio gson al directorio lib de Tomcat

Lo siguiente es cargar el controlador JDBC para MySQL. Una vez en la maquina virtual se desempacan el archivo.jar y se mueve al directorio “lib” de Tomcat (Ilustración 17).

```
raul@Tarea7VM:~$ ls
__MACOSX apache-tomcat-8.5.60 apache-tomcat-8.5.60.zip jaxrs-ri jaxrs-ri.zip mysql-connector-java-8.0.22
raul@Tarea7VM:~$ mv mysql-connector-java-8.0.22 /home/raul/apache-tomcat-8.5.60/lib
raul@Tarea7VM:~$
```

Ilustración 17 Mover archivo MySQL

Posteriormente vamos a iniciar nuestro servidor Tomcast, para lo cual es necesario definir las varibales de entorno siguientes (Ilustración 18).

```
raul@Tarea7VM:~/apache-tomcat-8.5.60$ export CATALINA_HOME=/home/raul/apache-tomcat-8.5.60
raul@Tarea7VM:~/apache-tomcat-8.5.60$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
raul@Tarea7VM:~/apache-tomcat-8.5.60$
```

Ilustración 18 Variables de entorno servidor Tomcast

En la siguiente ilustración se muestra como iniciar el servidor y de igual manera como se debe de detener, sin embargo, para fines prácticos se vuelve a encender para poder cumplir con el objetivo final de la tarea (Ilustración 19).

```

raul@Tarea7VM:~$ sh $CATALINA_HOME/bin/catalina.sh start
Using CATALINA_BASE:  /home/raul/apache-tomcat-8.5.60
Using CATALINA_HOME:  /home/raul/apache-tomcat-8.5.60
Using CATALINA_TMPDIR: /home/raul/apache-tomcat-8.5.60/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH:       /home/raul/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/raul/apache-tomcat-8.5.60/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
raul@Tarea7VM:~$ sh $CATALINA_HOME/bin/catalina.sh stop
Using CATALINA_BASE:  /home/raul/apache-tomcat-8.5.60
Using CATALINA_HOME:  /home/raul/apache-tomcat-8.5.60
Using CATALINA_TMPDIR: /home/raul/apache-tomcat-8.5.60/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH:       /home/raul/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/raul/apache-tomcat-8.5.60/bin/tomcat-juli.jar
Using CATALINA_OPTS:
raul@Tarea7VM:~$ 

```

Ilustración 19 Iniciar/Detener servidor Tomcast

La siguiente parte de la tarea es la instalación de MySQL, por ello primero se tiene que actualizar el paquete default de MySQL (Ilustración 20). Una vez hecho esto se procede a instalar el paquete default de MySQL (Ilustración 20) y de sebe de ejecutar el script de seguridad siguiendo la siguiente secuencia (Ilustración 21):

- 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

```

raul@Tarea7VM:~$ 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 [1776 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1693 kB]
Fetched 3721 kB in 1s (3495 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
raul@Tarea7VM:~$ sudo apt install mysql-server

```

Ilustración 20 Comando para actualizar paquetes

```

raul@Tarea7VM:~$ 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.

```

Ilustración 21 Ejecución de script de seguridad

Se ejecuta el monitos de MySQL (Ilustración 22).

```

raul@Tarea7VM:~$ 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> 

```

Ilustración 22 Ejecución MySQL

Se puede ejecutar la contraseña con el comando ALTER USER y posteriormente se actualizan los privilegios (Ilustración 23). Por ultimo para poder salir del monitor de MySQL, se ejecuta el comando “quit” (Ilustración 23).

```
[mysql]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)  
  
[mysql]> quit  
Bye
```

Ilustración 23 Actualización de privilegios /Salir comando

La siguiente sección de pasos es el crear un nuevo usuario en MySQL, por ello se ejecutan los comandos básicos de bases de datos (Ilustración 24). Y se procede a crear la base de datos con el usuario que se ha creado (Ilustración 25). Se crea la base de datos “servicios_web” y nos conectamos a ella. Una vez dentro, se crean las tablas usuarios y fotos_usuarios así como se crea una regla de integridad y un índice único (Ilustración 26). Y se procede a salir de MySQL.

```
[raul@Tarea7VM:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 13  
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 'tarea7VM2959';  
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> |
```

Ilustración 24 Ejecución de monitor de MySQL

```
[raul@Tarea7VM:~$ mysql -u hugo -p
[Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14
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> ]
```

Ilustración 25 Usuario hugo

```
[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)
  -> );
Query OK, 0 rows affected (0.09 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 (0.12 sec)

mysql> alter table fotos_usuarios add foreign key (id_usuario) references usuarios(id_usuario);
Query OK, 0 rows affected (0.20 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create unique index usuarios_1 on usuarios(email);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Ilustración 26 Creación de tablas en la base de datos.

El siguiente paso es el de cargar y desempacar el archivo Servicios.zip en nuestra maquina virtual (Ilustración 27). Y luego se define la variable de entorno CATALINA_HOME (Ilustración 28). Posteriormente se compila la clase Servicio.java (Ilustración 29).

```
raul@Tarea7VM:~$ ls
Servicio.zip  __MACOSX  apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip  jaxrs-ri  jaxrs-ri.zip  mysql-connector-java-8.0.22.zip
raul@Tarea7VM:~$ unzip Servicio.zip
Archive: Servicio.zip
  creating: Servicio/
  creating: __MACOSX/.Servicio
  creating: Servicio/negocio/
  inflating: __MACOSX/Servicio/.negocio
  creating: Servicio/META-INF/
  inflating: __MACOSX/Servicio/.META-INF
  creating: Servicio/WEB-INF/
  inflating: __MACOSX/Servicio/.WEB-INF
  inflating: Servicio/negocio/Usuario.java
  inflating: __MACOSX/Servicio/negocio/.Usuario.java
  inflating: Servicio/negocio/Servicio.java
  inflating: __MACOSX/Servicio/negocio/.Servicio.java
  inflating: Servicio/negocio/AdaptadorGsonBase64.java
  inflating: __MACOSX/Servicio/negocio/.AdaptadorGsonBase64.java
  inflating: Servicio/negocio/Error.java
  inflating: __MACOSX/Servicio/negocio/.Error.java
  inflating: Servicio/negocio/Foto.java
  inflating: __MACOSX/Servicio/negocio/.Foto.java
  inflating: Servicio/META-INF/context.xml
  inflating: __MACOSX/Servicio/META-INF/.context.xml
  creating: Servicio/WEB-INF/classes/
  inflating: __MACOSX/Servicio/WEB-INF/.classes
  inflating: Servicio/WEB-INF/web.xml
  inflating: __MACOSX/Servicio/WEB-INF/.web.xml
  creating: Servicio/WEB-INF/classes/negocio/
  inflating: __MACOSX/Servicio/WEB-INF/classes/.negocio
raul@Tarea7VM:~$ ls
Servicio  Servicio.zip  __MACOSX  apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip  jaxrs-ri  jaxrs-ri.zip  mysql-connector-java-8.0.22.zip
raul@Tarea7VM:~$
```

Ilustración 27 Desempaquetar archivo Servicios.zip

```
raul@Tarea7VM:~/apache-tomcat-8.5.60$ export CATALINA_HOME=/home/raul/apache-tomcat-8.5.60
Ilustración 28 Variable de entorno CATALINA_HOME
```

```
raul@Tarea7VM:~/Servicio$ ls
META-INF WEB-INF negocio
raul@Tarea7VM:~/Servicio$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
raul@Tarea7VM:~/Servicio$
```

Ilustración 29 compilación DE LA CLASE SERVICIO.JAVA

Se edita el archivo “context.xml” del directorio META-INF, y se define el usuario y contraseña que antes se ha puesto (Ilustración 30).

```
GNU nano 2.9.3                                     ~ — raul@Tarea7VM: ~/Servicio/META-INF — ssh raul@13.65.43.65
context.xml

<Context>
  <Resource name="jdbc/datasource_Servicio" auth="Container" type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="hugo" password="tarea7VM2959"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost/servicio_web?serverTimezone=UTC"/>
</Context>
```

Ilustración 30 Modificación de archivo context.xml

Ejecutar los siguientes comandos para crear el servicio web para Tomcat (Ilustración 31). Y por ultimo para poder desplegar el servicio web se copia el archivo Servicio.war al directorio “webapps” de Tomcast (Ilustración 32).

```
[raul@Tarea7VM:~/Servicio$ cp negocio/*.class WEB-INF/classes/negocio/.
[raul@Tarea7VM:~/Servicio$ jar cvf Servicio.war WEB-INF META-INF
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
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/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%)
adding: WEB-INF/classes/negocio/Usuario.class(in = 899) (out= 518)(deflated 42%)
adding: WEB-INF/web.xml(in = 672) (out= 296)(deflated 55%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 312) (out= 221)(deflated 29%)
raul@Tarea7VM:~/Servicio$ ]
```

Ilustración 31 Comandos

```
[raul@Tarea7VM:~$ cd Servicio/
[raul@Tarea7VM:~/Servicio$ ls
META-INF  Servicio.war  WEB-INF  negocio
[raul@Tarea7VM:~/Servicio$ mv Servicio.war /home/raul/apache-tomcat-8.5.60/webapps]
```

Ilustración 32 Servicio.war

Para emepzar a probar el servicio subimos el archivo usuario_sin_foto.png a la maquina virtual (Ilustración 33) y lo copiamos al subdirectorio webapps/ROOT de Tomcat (Ilustración 34).

```
[raul@Tarea7VM:~$ ls
Servicio      __MACOSX           apache-tomcat-8.5.60.zip    jaxrs-ri.zip          usuario_sin_foto.png
Servicio.zip  apache-tomcat-8.5.60  jaxrs-ri                  mysql-connector-java-8.0.22.zip
raul@Tarea7VM:~$ ]
```

Ilustración 33 Archivo de imagen

```
[raul@Tarea7VM:~$ mv usuario_sin_foto.png /home/raul/apache-tomcat-8.5.60/webapps/ROOT
raul@Tarea7VM:~$ ]
```

Ilustración 34 copiar el archivo imagen al directorio webapps/ROOT

Para probar que se ha hecho correctamente todo hasta el momento, se prueba con el siguiente link “ http://ip-de-la-máquina-virtual:8080/usuario_sin_foto.png ” , una ves dentro visualizamos que esta la imagen que hemos cargado al directorio. (Ilustración 35 y 36).



Ilustración 35 Visualización en navegador



Ilustración 36 Visualización en navegador

Para las siguientes pruebas se suben los archivos WSClient.js y prueba.html a nuestra maquina virtual y se mueven al directorio webapps/ROOT (Ilustración 37).

```
raul@Tarea7VM:~$ ls
Servicio      WSClient.js  apache-tomcat-8.5.60      jaxrs-ri      mysql-connector-java-8.0.22.zip
Servicio.zip  __MACOSX    apache-tomcat-8.5.60.zip  jaxrs-ri.zip  prueba.html
raul@Tarea7VM:~$ mv prueba.html WSClient.js /home(raul/apache-tomcat-8.5.60/webapps/ROOT
raul@Tarea7VM:~$ █
```

Ilustración 37 Archvios WSClient.js & prueba.html

Se ingresa al navegador con el link “<http://ip-de-la-máquina-virtual:8080/prueba.html>” para viauslizar la prueba (Ilustración 38).

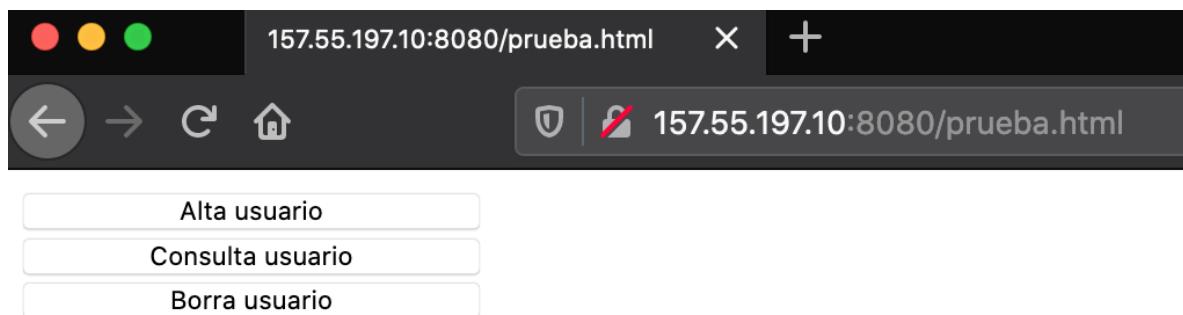


Ilustración 38 Prueba

Se procede a dar de alta a un usuario (Ilustración 39 y 40).

157.55.197.10:8080/prueba.html

Alta de usuario

Email *
gren@gmail.com

Nombre *
Raul

Apellido paterno *
Sanchez

Apellido materno
Rico

Fecha de nacimiento *
16 / 12 / 2020

Teléfono
5566778899

Genero
Masculino


Examinar... FB_IMG_1529902202057.jpg

Alta
Regresa

Ilustración 39 Alta usuario

157.55.197.10:8080/prueba.html

Alta de usuario

Email *
gren@gmail.com

Nombre *
Raul

Apellido paterno *
Sanchez

Apellido materno
Rico

Fecha de nacimiento *
16 / 12 / 2020

Teléfono
5566778899

Genero
Masculino


Examinar... FB_IMG_1529902202057.jpg

Alta
Regresa

OK

Aceptar

Ilustración 40 Confirmación alta

Se intenta dar de alta a otro usuario con el mismo correo marcando error (Ilustración 41 y 42).

Alta de usuario

Email *
gren@gmail.com

Nombre *
Pauly

Apellido paterno *
Vazquez

Apellido materno
Robles

Fecha de nacimiento *
01 / 12 / 2020

Teléfono
5511223344

Género
Femenino


Examinar... 13427780_10157495829865497_7924019854653797221_n.png

Alta

Regresa

Ilustración 41 Alta otro usuario con correo existente

Alta de usuario

Email *
gren@gmail.com

Nombre *
Pauly

Apellido paterno *
Vazquez

Apellido materno
Robles

Fecha de nacimiento *
01 / 12 / 2020

Teléfono
5511223344

Género
Femenino


Examinar... 13427780_10157495829865497_7924019854653797221_n.png

Alta

Regresa

{"message": "El email ya existe"}

Aceptar

Ilustración 42 Mensaje de error alta de usuario

Se consulta el usuario dado de alta anteriormente con el correo (Ilustración 43 y 44).

157.55.197.10:8080/prueba.html

Consulta/Modifica usuario

Email *

gren@gmail.com

Nombre *

Raul

Apellido paterno *

Sanchez

Apellido materno

Rico

Fecha de nacimiento *

16 / 12 / 2020

Teléfono

5566778899

Genero

Masculino

Consulta

Regresa

Ilustración 43 Consulta usuario

157.55.197.10:8080/prueba.html

Consulta/Modifica usuario

Email *

gren@gmail.com

Nombre *

Raul

Apellido paterno *

Sanchez

Apellido materno

Rico

Fecha de nacimiento *

16 / 12 / 2020

Teléfono

5566778899

Genero

Masculino

Examinar... Ningún archivo seleccionado.

Consulta

Modifica

Regresa

Ilustración 44 Consulta usuario

Se modifica algun dato del usuario (Ilustración 45)

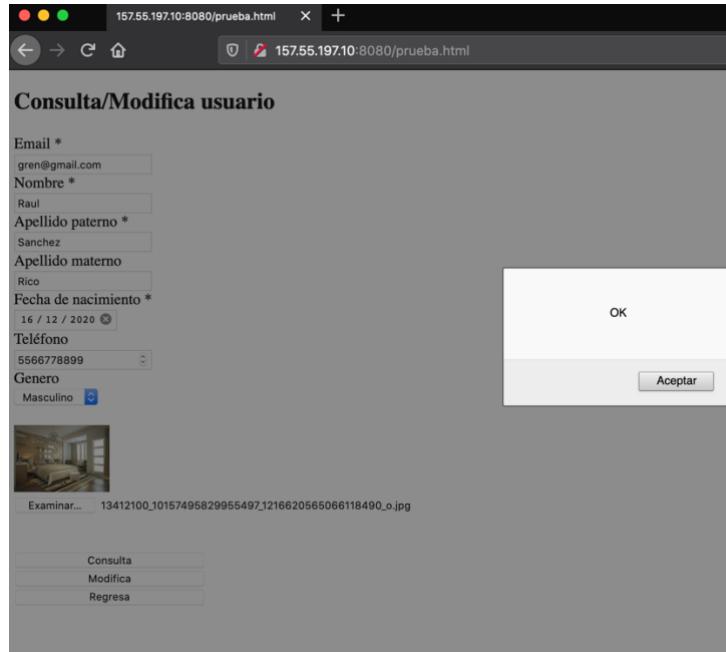


Ilustración 45 Modificación de usuario

Verificar que el usuario se modifico correcatmente al cargar de nuevo el navegador (Ilustración 46).



Ilustración 46 Verificación

Se procede a borrar el usuario (Ilustración 47 y 48).



Borra usuario

Email *

gren@gmail.com

Borra

Regresa

Ilustración 47 Borrar usuario

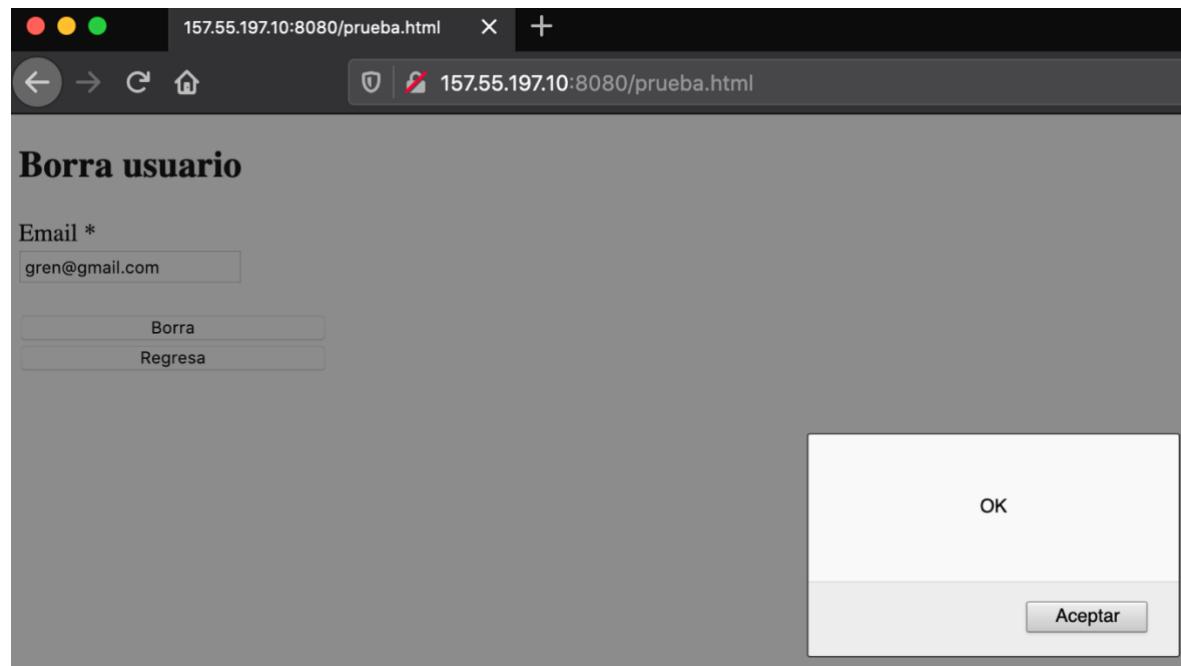


Ilustración 48 Borrar usuario

Por ultimo se visualiza en un dispositivo movil (Ilustración 49,50,51).

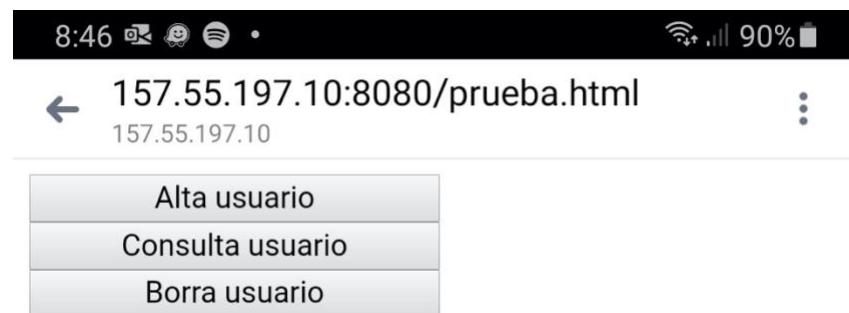


Ilustración 49 Prueba Celular

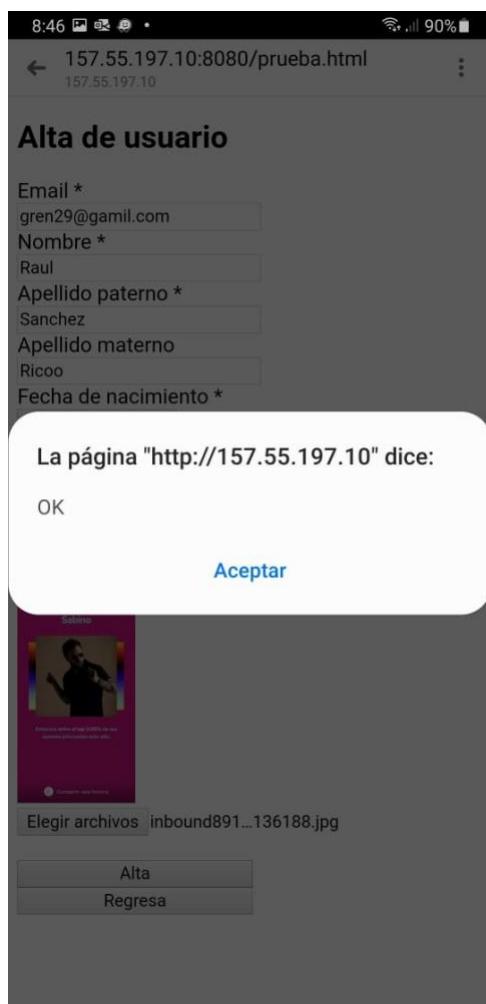


Ilustración 50 Alta usuario

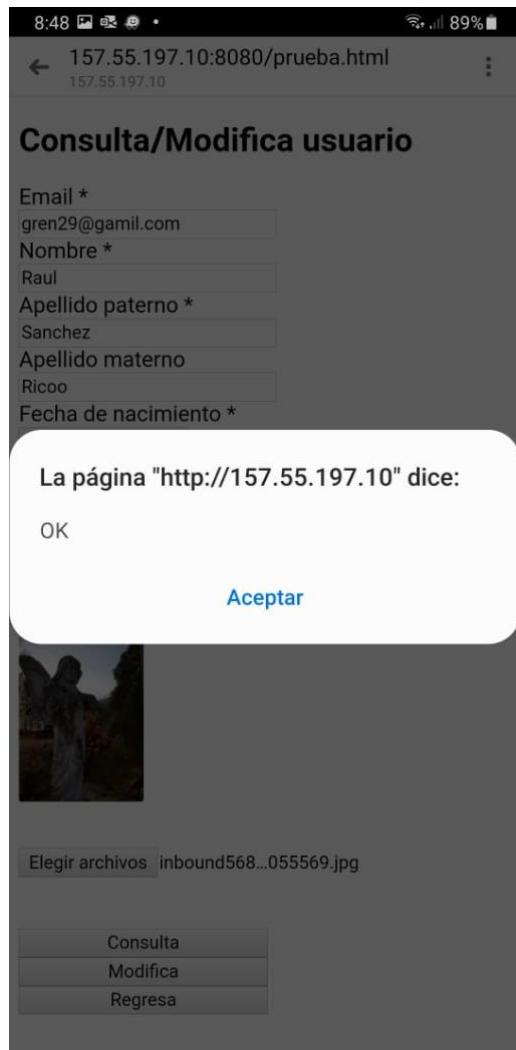


Ilustración 51 Consulta Usuario

Conclusiones

Poco a poco las tareas han sido mas largas sin embargo se ha notado que es mas fácil el poder ir manejando la maquina virtual tanto para crearla como para subir archivos. En lo particular esta tarea ha demostrado ser de gran ayuda para entender que esta tecnología es un arma muy poderosa para hacer posible la comunicación entre un cliente y el servidor (proveedor), haciendo que los desarrollos sean estables. Sin embargo, sigue habiendo algunas limitantes personales como el tiempo de carga de los archivos por las conexiones de red, sin embargo, dejando esto de lado, se ha cumplido con la tarea.