

Tarea 7

Implementación de un servicio web estilo REST

Pulido Bejarano Raymundo

28 de Noviembre del 2020

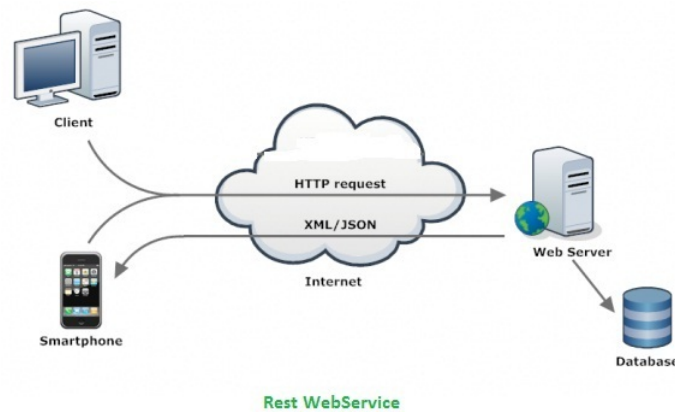


Figura 1: REST

1. Introducción

El propósito de esta tarea es ejecutar las instrucciones otorgadas por el profesor para realizar la implementación de un servicio web de tipo REST.

Pero que es un servicio web, pues un servicio web descrito de forma breve es un sistema que permite la inter operabilidad maquina a máquina, mediante el uso de protocolos de comunicación de red.

En cuanto al servicio web de tipo REST, se podría describir de forma breve, como la implementación del protocolo HTTP, para el trafico de información en formato JSON, XML o ambos, mediante las distintas operaciones HTTP, como son GET, POST, PUT, DELETE entre otras

2. Desarrollo

El desarrollo de la tarea estara dividido por cinco partes, las cuales son: Maquina Virtual, Tomcat, MySQL, Servicio Web, Pruebas

2.1. Maquina Virtual

El primer paso nos indicaba crear una maquina virtual en la nube Azure, con al menos 1Gb de RAM.

Lo que se muestra a continuación:

Grupo de recur... (cambiar)	: Tarea7SerWeb_group	Sistema operativo	: Linux (ubuntu 18.04)
Estado	: En ejecución	Tamaño	: B1s estándar (1 vcpu, 1 GiB de memoria)
Ubicación	: Centro-Sur de EE. UU.	Dirección IP pública	: 104.215.117.28
Suscripción (cambiar)	: Azure para estudiantes	Red virtual/subred	: Tarea7SerWeb_group-vnet/default
Id. de suscripción	: 746ae8d5-1a0f-4832-bec4-8c42cca66b78	Nombre DNS	: Configurar
Etiquetas (cambiar)	: Haga clic aquí para agregar etiquetas.		

Figura 2: Maquina Virtual: Tarea7SerWeb_group

En la imagen anterior se puede observar la implementación de la maquina virtual, con la IP publica:104.215.117.28, a la que en los pasos siguientes accederemos para implementar el servicio web.

A su vez como también se nos indicó en las instrucciones, a esta Maquina Virtual (MV), se le abrió el puerto 8080, para el protocolo TCP, como se puede observar en la siguiente figura:

Red virtual/subred: Tarea7SerWeb_group-vnet/default		IP pública de NIC: 104.215.117.28	IP privada de NIC: 10.0.1.4	Redes aceleradas: Deshabilitado
Reglas de puerto de entrada				
Reglas de puerto de salida				
Grupos de seguridad de aplicación				
Equilibrio de carga				
Grupo de seguridad de red Tarea7SerWeb-nsg (se conectó a la interfaz de red: tarea7serweb700)				
Impactos 0 subredes, 1 interfaces de red				
Prioridad	Nombre	Puerto	Protocolo	Origen
300	SSH	22	TCP	Cualquiera
310	Port_8080	8080	TCP	Cualquiera
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera

Figura 3: Puertos de la Maquina Virtual

Para realizar las siguientes operaciones procedí a conectarme vía ssh con la MV, para crear una carpeta, la cual será la que aloje todos los archivos

enviados a la MV, el path relativo la carpeta dentro de la MV, es -/Ray”.

A continuación, se podrá observar una figura que muestra la forma en el como realice la conexión.

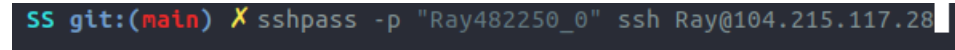


Figura 4: Comando para Realizar la conexión SSH

Ya que primeramente realice la lectura e todos los pasos pude percatarme que realizaríamos el envío de un gran grupo de archivos a la MV, por lo recurrí a realizar la descarga de todos, y posteriormente recurrí a realizar un Script en bash, que me permitiera automatizar este envío. El dicho Script se mostrara a continuación.

```
1 #!/bin/bash/
2
3 IPMV="104.215.117.28"
4 Archivos="*.sh *.html *.js *.zip *.jar *.png"
5 Password="Ray4802250_09"
6 Usuario="Ray"
7 DireccionDest="~/Ray"
8
9 sshpass -p $Password scp $Archivos $Usuario@$IPMV:
   $DireccionDest
```

Script para enviar los Archivos a la MV "mvToMV.sh"

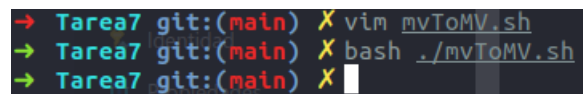


Figura 5: Ejecución del Script mvToMV.sh

Como se puede observar en la figura anterior se puede observar la ejecución del Script anteriormente mostrado, el cual realizo el envío de todos los archivos necesarios para la Tarea.

A continuación les mostrare los archivos ya en la Maquina Virtual

2.2. Tomcat

Una vez la MV fue creada y todos los archivos necesarios se encuentran en ella, procederemos a realizar los siguientes pasos:

```
Ray@Tarea7SerWeb:~/Ray$ ll
total 20476
drwxrwxr-x 2 Ray Ray 4096 Nov 26 23:07 ./
drwxr-xr-x 6 Ray Ray 4096 Nov 26 23:06 ../
-rw-rw-r-- 1 Ray Ray 116 Nov 26 23:07 MySQLInstall.sh
-rw-rw-r-- 1 Ray Ray 5351 Nov 26 23:07 Servicio.zip
-rw-rw-r-- 1 Ray Ray 3142 Nov 26 23:07 WSClient.js
-rw-rw-r-- 1 Ray Ray 11010614 Nov 26 23:07 apache-tomcat-8.5.60.zip
-rw-rw-r-- 1 Ray Ray 400 Nov 26 23:07 despliegue.sh
-rw-rw-r-- 1 Ray Ray 410 Nov 26 23:07 despliegue_2.sh
-rw-rw-r-- 1 Ray Ray 210856 Nov 26 23:07 gson-2.3.1.jar
-rw-rw-r-- 1 Ray Ray 4936287 Nov 26 23:07 jaxrs-ri-2.24.zip
-rw-rw-r-- 1 Ray Ray 212 Nov 26 23:07 mvToMV.sh
-rw-rw-r-- 1 Ray Ray 4742828 Nov 26 23:07 mysql-connector-java-8.0.22.zip
-rw-rw-r-- 1 Ray Ray 9084 Nov 26 23:07 prueba.html
-rw-rw-r-- 1 Ray Ray 1659 Nov 26 23:07 tarea7.sh
-rw-rw-r-- 1 Ray Ray 1662 Nov 26 23:07 usuario_sin_foto.png
Ray@Tarea7SerWeb:~/Ray$
```

Figura 6: Archivos recibidos en la maquina virtual

- Instalacion de JAVA JDK 8
- Desempaquetao de TOMCAT
- Borrado y creacion de directorios
- Desempaquetado de jaxrs*.zip, mysql*.zip
- Copia de todos los archivos *.jar a la carpeta lib, dentro de la carpeta creada al desempacar Tomcat
- Creacion permanente de las variables de entorno tanto de JAVA, como de Tomcat
- Puesta en marcha de Tomcat

Para realizar todos los pasos anteriores de forma que no necesitara mucho tiempo de uso de la mv, y de esta forma recortar el costo por el uso de la MV, decidí realizar un pequeño Script el cual sea mostrado y explicado a continuación.

```
1 #!/bin/bash
2
3 # Instalar Java8
4 echo "Instalando JAVA 8"
5 sudo add-apt-repository ppa:webupd8team/java
6 sudo apt update
7 sudo apt-get install openjdk-8-* -y
8 sudo update-alternatives --display java
```

```

9
10 #read ruta
11 echo "Creando Variable de entorno de JAVA"
12 ruta="/usr/lib/jvm/java-8-openjdk-amd64"
13 sudo echo "JAVA_HOME='$ruta'" >> /etc/environment
14 source /etc/environment
15
16 echo "JAVA_HOME = $JAVA_HOME"
17
18 sudo apt install unzip -y
19
20 if [ -f "./apache-tomcat-8.5.60.zip" ]
21 then
22     echo "Extrallendo Tomcat"
23     unzip apache-tomcat-8.5.60.zip
24     rm -r apache-tomcat-8.5.60/webapps/
25     mkdir -p apache-tomcat-8.5.60/webapps/ROOT
26 else
27     echo "No esta el archivo para instalart Tomcat"
28 fi
29
30 if [ -f "./jaxrs-ri-2.24.zip" ]
31 then
32     echo "Obteniendo todos los archivos *.jar de JAC-
33         RS"
34     unzip jaxrs-ri-2.24.zip
35     mv ./jaxrs-ri/lib/*.jar ./apache-tomcat-8.5.60/
36         lib/
37     mv ./jaxrs-ri/api/*.jar ./apache-tomcat-8.5.60/
38         lib/
39     mv ./jaxrs-ri/ext/*.jar ./apache-tomcat-8.5.60/
40         lib/
41     rm ./apache-tomcat-8.5.60/lib/javax.servlet-api
42         -3.0.1.jar
43 else
44     echo "No existe el archivo jaxrs-ri"
45 fi
46
47 if [ -f "./gson-2.3.1.jar" ]
48 then
49     echo "Copiando gson.jar a Tmcat/lib"
50     mv ./gson-2.3.1.jar ./apache-tomcat-8.5.60/lib/
51 else
52     echo "No existe gson.jar"
53 fi

```

```

49
50 if [ -f "./mysql-connector-java-8.0.22.zip" ]
51 then
52     echo "Descomprimiendo el Driver mysql"
53     unzip mysql-connector-java-8.0.22.zip
54     mv mysql-connector-java-8.0.22/mysql-connector-
        java-8.0.22.jar ./apache-tomcat-8.5.60/lib/
55 else
56     echo "No se encontro el driver mysql"
57 fi
58
59 echo "Creando Variable de entorno de Tomcat"
60 sudo echo "CATALINA_HOME='$(pwd)/apache-tomcat-8.5.60'"
    >> /etc/environment
61 source /etc/environment
62
63 sh $CATALINA_HOME/bin/catalina.sh start

```

Script para automatizar el proceso de instalación y preparación de Tomcat en la MV "tarea7.sh"

Como se puede observar en el script anterior, se realizó la automatización de los pasos anteriormente dicho, pero procederé a realizar una explicación más a fondo de las partes del código.

De la línea 3 a la 16, se realizó la instalación de Java JDK8, como también se agregó la variable de entorno para java "JAVA_HOME"

```

Ray@Tarea7SerWeb:~/Ray$ sudo bash tarea7.sh
Instalando JAVA 8
The Oracle JDK License has changed for releases starting April 16, 2019.

```

Figura 7: Se muestra el principio de la ejecución del Script

```

Creando Variable de entorno de JAVA
JAVA_HOME = /usr/lib/jvm/java-8-openjdk-amd64

```

Figura 8: Creación de la variable de entorno JAVA_HOME

Posteriormente de la línea 20 a la 28 en caso de existir el archivo ". /apache-tomcat-8.5.60.zip".^{el} cual es el empaquetado de Tomcat, se desempaquetará

para posteriormente se elimine el directorio /webapps” por cuestiones de seguridad, y después proseguir a crear tanto /webapps como /webapps/ROOT”, y en caso de no existir, nos mostraría el mensaje de error.

De las linea 30 a la 57, se realizó el desempaquetado de los archivos “.zip”, como el envió de los archivos “.jar” al directorio -/Ray/apache-tomcat-8.5.60/lib/”.

Posteriormente de la linea 59 a la 63, realizaré la escritura de la variable de entorno de manera fija, para que posteriormente inicie el servicio de apache Tomcat.

```
Creando Variable de entorno de Tomcat
Using CATALINA_BASE:   /home/Ray/Ray/apache-tomcat-8.5.60
Using CATALINA_HOME:   /home/Ray/Ray/apache-tomcat-8.5.60
Using CATALINA_TMPDIR: /home/Ray/Ray/apache-tomcat-8.5.60/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/Ray/Ray/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/Ray/Ray/apache-tomcat-8.5.60/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

Figura 9: Creacion de variable de entorno CATALINA_HOME y inicio de servicio

2.3. MySQL

A continuación, se realizó la instalación y configuración de MySQL, para poder trabajar con Bases de Datos e nuestro Servicio web.

Los pasos descritos en clase son:

- Instalacion de MySQL
- Mysql-secure script
- Permisos para root
- Creacion de usuarios
- Creacion de la base de datos
- Creacion de las tablas

Primeramente realizamos la instalación de “mysql-server”, para proseguir con la ejecución del script “mysql_secure_installation”,

A continuación se mostrará una parte de la ejecución:

Cambiamos la contraseña del usuario ROOT:

Para después crea un usuario nuevo llamado “hugo”, como se muestra a continuación:

```

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: 

```

Figura 10: Script mysql_secure_installation

```

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!
Ray@Tarea7SerWeb:~/Ray$ 

```

Figura 11: Fin Script mysql_secure_installation

```

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '4802250';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> quit

```

Figura 12: Cambiando contraseña de root

```

mysql> create user hugo@localhost identified by '4802250';
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

```

Figura 13: Creamos usuario nuevo

Creamos la base de datos y las tablas dentro de ella:

```
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 (0.08 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 (0.21 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create unique index usuarios_1 on usuarios(email);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> quit
```

Figura 14: Creamos la base de datos y la tabla

2.4. Servicio Web

En esta sección procederemos a realizar los siguientes pasos del proceso:

- Desempaquetar y compilarar el servicio
- modificar el archivo context.xml
- Crear el archivo .war
- Copiar elementos a los directorios de tomcat.

Para realizar el desempaquetado y compilado del Servicio cree un pequeño script, el cual le mostraré a continuación:

```
1  #!/bin/bash/
2
3  if [ -f "./Servicio.zip" ]
4  then
5      echo "Desempacando el servicio"
6      mkdir Servicio
7      cd Servicio
8      unzip Servicio.zip
9  else
10     echo "No se encuentra el archivo Servicio.zip"
11 fi
12
13 echo "Compilando Servicio.java"
14 javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:
15     $CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.
16     java
17
18 echo "Modificar el archivo context.html con tu auraio y
19     contrase a"
```

Script para automatizar el proceso desempaquetado y compilado del servicio

Una vez realizado lo anteriormente comentado, procedimos a modificar el archivo Context.xml.^agregándole en los campos usuarios y password, la información del usuario que creamos en mysql, los cuales en nuestro caso son "hugoz "4802250 como se puede observar a continuación:

Ahora para finalizar de implementar el servicio solo queda crear el archivo ".war" moverlo junto a los otros archivos necesarios para su funcionamiento

Para realizar esto de una forma más ágil y sin errores, desarrolle el siguiente Script:

```

<Context>
  <Resource name="jdbc/datasource_Servicio" auth="Container" type="javax.s
ql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="hugo" password="4802250"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost/servicio_web?serverTimezone=UTC"/>
</Context>

```

Figura 15: Agregamos usuario y contraseña al archivo context.html”

```

1  #!/bin/bash/
2  cd ~/Ray/Servicio/
3  rm WEB-INF/classes/negocio/*
4  cp negocio/*.class WEB-INF/classes/negocio/.
5  jar cvf Servicio.war WEB-INF META-INF
6  sudo cp Servicio.war ~/Ray/apache-tomcat-8.5.60/webapps/
7
8
9  sudo cp ~/Ray/usuario_sin_foto.png ~/Ray/apache-tomcat
   -8.5.60/webapps/ROOT
10 sudo cp ~/Ray/WSCClient.js ~/Ray/apache-tomcat-8.5.60/
   webapps/ROOT
11 sudo cp ~/Ray/prueba.html ~/Ray/apache-tomcat-8.5.60/
   webapps/ROOT

```

Script para automatizar el proceso desempaquetado y compilado del servicio

Una vez finalizado esto únicamente queda realizar las pruebas.

2.5. Pruebas

Con el fin de realizar las diversas pruebas se accedió a la siguiente URL: <http://104.215.117.28:8080/prueba.html>

Como se puede observar a continuación se realizó el acceso desde un móvil y desde la PC:

En las figuras anteriores se puede mostrar el correcto funcionamiento del sistema en ambos dispositivos.

A continuación procederé a poner las Capturas de pantalla de las pruebas sugeridas, estas pruebas fueron realizadas en el dispositivo celular.



Figura 16: Prueba de acceso desde una Celular

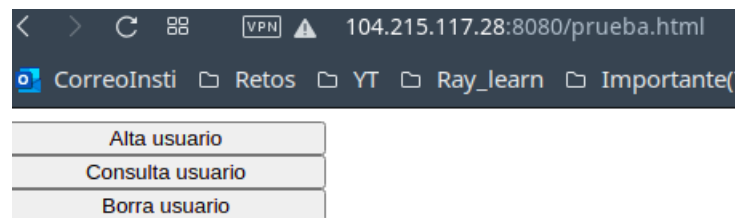


Figura 17: Prueba de acceso desde una PC

En la figura anterior se puede observar el mensaje al realizar correctamente el alta de un usuario.

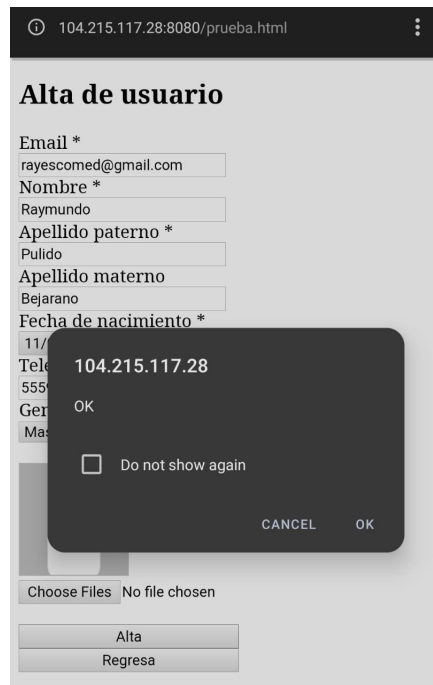








Figura 18: Dando de alta Usuario

En la figura anterior se puede observar los datos del usuario que acabamos de dar de alta.

5:35 PM    ...    56

Consulta/Modifica usuario

Email *
rayescomed@gmail.com

Nombre *
Raymundo


Apellido paterno *
Pulido

Apellido materno
Bejarano

Fecha de nacimiento *
11/09/2020 ▼

Teléfono
5559044234

Genero
Masculino ▼

 Choose Files No file chosen

Consulta
Modifica
Regresa

Figura 19: Consultando Usuario

En la figura anterior se puede observar que modifique los datos del usuario, más específicamente le agregue un nombre.



Figura 20: Modificando Usuario

En la figura anterior se puede observar como se borró el usuario.



Figura 21: Borrando Usuario

En la figura anterior se puede observar como a intentar borrar un usuario que no se encuentra muestra un error.

3. Conclusion

Para finalizar este reporte no me queda más que decir que sumamente conveniente el hecho que al crear una MV en Microsoft Azure nos genere una IP publica, ya que nos permite interactuar con los servidores existentes en la MV, referente al servidor REST, eh de realizar la notación que en mi perspectiva personal, encontré que era sumamente sencillo de implementar y de entender los conceptos que rodean a este.

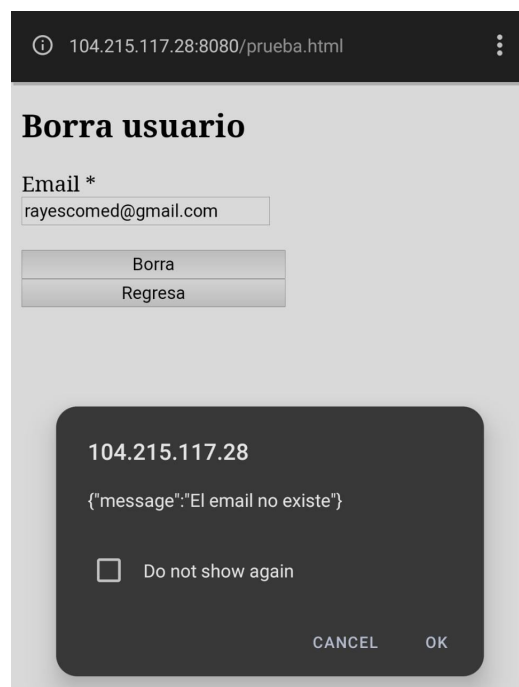


Figura 22: Error al borrar un Usuario