



**Instituto Politecnico Nacional**

**ESCOM “ESCUELA SUPERIOR DE  
CÓMPUTO”**



*DESARROLLO DE SISTEMAS DISTRIBUIDOS*

*TAREA 7. IMPLEMENTACIÓN DE UN SERVICIO WEB ESTILO REST*

PROFE: CARLOS PINEDA GUERRERO

ALUMNO: Rojas Alvarado Luis Enrique

GRUPO: 4CM5

Una vez creada nuestra máquina virtual de ubuntu 18.04 con 1GB de RAM, activaremos el puerto 8080 de TCP, esto se hace llenando al apartado “Agregar regla de puerto de entrada” en el menú de “Redes” dentro de la información de nuestro recurso.

Microsoft Azure

Inicio > CreateVm-Canonical.UbuntuServer-18.04-LTS-20201130135046 > WebService

### WebService | Redes

Máquina virtual

Buscar (Ctrl+/)

Información general  
Registro de actividad  
Control de acceso (IAM)  
Etiquetas  
Diagnosticar y solucionar prob...

Configuración

Redes

Discos  
Tamaño  
Seguridad  
Recomendaciones de Advisor  
Extensiones  
Entrega continua

Adjuntar interfaz de red Desasociar interfaz de red

Configuración de IP (Principal)

Interfaz de red: **webservice235** Reglas de seguridad vigentes Topología

Red virtual/subred: UbuntuVM-vnet/default IP pública de NIC: **13.65.231.157** IP privada de NIC: **10.0.0.4** Redes aceleradas: **Deshabilitado**

Reglas de puerto de entrada Reglas de puerto de salida Grupos de seguridad de aplicación Equilibrio

Grupo de seguridad de red **WebService-nsg** (se conectó a la interfaz de red: **webservice235**)  
Impactos 0 subredes, 1 interfaces de red

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
300	SSH	22	TCP	Cualquiera	Cualquiera	Permitir
320	HTTP	8080	TCP	Cualquiera	Cualquiera	Permitir
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Permitir
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Permitir
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Denegar

Agregar regla de puerto de entrada

Al dar click en el botón, podremos activar el puerto 8080 de TCP, simplemente seleccionando la casilla etiquetada con el nombre “TCP” y escribimos el puerto 8080 en el campo “Intervalo de puertos de destino y permitimos.

Microsoft Azure

Inicio > CreateVm-Canonical.UbuntuServer-18.04-LTS-20201130135046 > WebService

### WebService | Redes

Máquina virtual

Buscar (Ctrl+/)

Información general  
Registro de actividad  
Control de acceso (IAM)  
Etiquetas  
Diagnosticar y solucionar prob...

Configuración

Redes

Conectar  
Discos  
Tamaño  
Seguridad  
Recomendaciones de Advisor  
Extensiones  
Entrega continua

Adjuntar interfaz de red Desasociar interfaz de red

Configuración de IP (Principal)

Interfaz de red: **webservice235** Reglas de seguridad vigentes Topología

Red virtual/subred: UbuntuVM-vnet/default IP pública de NIC: **13.65.231.157** IP privada de NIC: **10.0.0.4** Redes aceleradas: **Deshabilitado**

Reglas de puerto de entrada Reglas de puerto de salida Grupo

Grupo de seguridad de red **WebService-nsg** (se conectó a la interfaz de red: **webservice235**)  
Impactos 0 subredes, 1 interfaces de red

Prioridad	Nombre	Puerto
300	SSH	22
320	HTTP	80
65000	AllowVnetInBound	Cualquiera
65001	AllowAzureLoadBalancerInBound	Cualquiera
65500	DenyAllInBound	Cualquiera

Agregar regla de puerto de entrada

HTTP

WebService-nsg

Guardar Descartar Básica Eliminar

Origen \* Any

Intervalos de puertos de origen \* \*

Destino \* Any

Intervalos de puertos de destino \* 8080

Protocolo \* Any TCP UDP ICMP

Acción \* Permitir Denegar

Prioridad \* 320

Nombre HTTP

Descripción

Hecho esto, procedemos a conectarnos a nuestra máquina virtual. En esta ocasión usaremos el puerto 22 de SSH, simplemente escribiremos nuestro usuario con el que creamos nuestro recurso en Azure y la dirección IP pública, que se encuentra en la información del mismo. Al escribir `ssh usuario@ip_Pública` nos preguntará que si queremos conectarnos intercambiando llaves SSH a lo que responderemos “yes” e ingresaremos la contraseña y nos dejará conectarnos con nuestra máquina virtual.

```
wicho@wicho:~$ ssh luis@13.65.231.157
The authenticity of host '13.65.231.157 (13.65.231.157)' can't be established.
ECDSA key fingerprint is SHA256:QGpKdUg2yQtGxD4tV/agHJem70jAdwByLBJA7rPPg14.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.65.231.157' (ECDSA) to the list of known hosts.
luis@13.65.231.157'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 Mon Nov 30 20:06:47 UTC 2020

System load: 0.0          Processes:              110
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.

luis@WebService:~$
```

Ahora debemos instalar JDK8 ejecutando los siguientes comandos en la máquina virtual: `sudo apt update` y `sudo apt install openjdk-8-jdk-headless`.

```
luis@WebService:~$ 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)

luis@WebService:~$
```

Comprobamos la instalación con `java -version`

Con wget <https://downloads.apache.org/tomcat/tomcat-8/v8.5.60/bin/apache-tomcat-8.5.60.zip> hacia la url donde se encuentra el recurso .zip de tomcat y desempacarlo utilizando el comando unzip.

```
luis@WebService:~$ wget https://downloads.apache.org/tomcat/tomcat-8/v8.5.60/bin/apache-tomcat-8.5.60.zip
--2020-11-30 20:20:37-- https://downloads.apache.org/tomcat/tomcat-8/v8.5.60/bin/apache-tomcat-8.5.60.zip
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 2a01:4f8:10a:201a::2
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11010614 (10M) [application/zip]
Saving to: 'apache-tomcat-8.5.60.zip'
la siguiente URL (descargar la opción Core "zip"):
apache-tomcat-8.5.6 100%[=====] 10.50M 6.15MB/s in 1.7s

2020-11-30 20:20:39 (6.15 MB/s) - 'apache-tomcat-8.5.60.zip' saved [11010614/11010614]
luis@WebService:~$ ls
apache-tomcat-8.5.60.zip
luis@WebService:~$
```

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.

```
luis@WebService:~$ ls
apache-tomcat-8.5.60  apache-tomcat-8.5.60.zip
luis@WebService:~$ cd apache-tomcat-8.5.60/
luis@WebService:~/apache-tomcat-8.5.60$ ls -l
total 152
-rw-r--r-- 1 luis luis 19882 Nov 12 17:34 BUILDING.txt
-rw-r--r-- 1 luis luis 5544 Nov 12 17:34 CONTRIBUTING.md
-rw-r--r-- 1 luis luis 58068 Nov 12 17:34 LICENSE
-rw-r--r-- 1 luis luis 1777 Nov 12 17:34 NOTICE
-rw-r--r-- 1 luis luis 3336 Nov 12 17:34 README.md
-rw-r--r-- 1 luis luis 7314 Nov 12 17:34 RELEASE-NOTES
-rw-r--r-- 1 luis luis 16984 Nov 12 17:34 RUNNING.txt
drwxr-xr-x 2 luis luis 4096 Nov 12 17:34 bin
drwxr-xr-x 2 luis luis 4096 Nov 12 17:34 conf
drwxr-xr-x 2 luis luis 4096 Nov 12 17:34 lib
drwxr-xr-x 2 luis luis 4096 Nov 12 17:32 logs
drwxr-xr-x 2 luis luis 4096 Nov 12 17:34 temp
drwxr-xr-x 7 luis luis 4096 Nov 12 17:34 webapps
drwxr-xr-x 2 luis luis 4096 Nov 12 17:32 work
luis@WebService:~/apache-tomcat-8.5.60$
```



```

luis@WebService:~/apache-tomcat-8.5.60$ rm -r webapps/
luis@WebService:~/apache-tomcat-8.5.60$ mkdir webapps
luis@WebService:~/apache-tomcat-8.5.60$ cd webapps/
luis@WebService:~/apache-tomcat-8.5.60/webapps$ mkdir ROOT
luis@WebService:~/apache-tomcat-8.5.60/webapps$ ls -l
total 4
drwxrwxr-x 2 luis luis 4096 Nov 30 20:24 ROOT
luis@WebService:~/apache-tomcat-8.5.60/webapps$

```

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

```

luis@WebService:~/apache-tomcat-8.5.60/lib$ wget https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.24/jaxrs-ri-2.24.zip
--2020-11-30 20:29:40-- https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.24/jaxrs-ri-2.24.zip
Resolving repo1.maven.org (repo1.maven.org)... 151.101.48.209
Connecting to repo1.maven.org (repo1.maven.org)|151.101.48.209|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4936287 (4.7M) [application/zip]
Saving to: 'jaxrs-ri-2.24.zip'
jaxrs-ri-2.24.zip 100%[=====] 4.71M --.-KB/s in 0.09s

2020-11-30 20:29:40 (54.5 MB/s) - 'jaxrs-ri-2.24.zip' saved [4936287/4936287]

luis@WebService:~/apache-tomcat-8.5.60/lib$ ls -l
total 13484
-rw-r--r-- 1 luis luis 112373 Nov 12 17:34 annotations-api.jar
-rw-r--r-- 1 luis luis 54666 Nov 12 17:34 catalina-ant.jar
-rw-r--r-- 1 luis luis 120077 Nov 12 17:34 catalina-ha.jar
-rw-r--r-- 1 luis luis 78086 Nov 12 17:34 catalina-storeconfig.jar
-rw-r--r-- 1 luis luis 294855 Nov 12 17:34 catalina-tribes.jar
-rw-r--r-- 1 luis luis 1715715 Nov 12 17:34 catalina.jar
-rw-r--r-- 1 luis luis 2450404 Nov 12 17:34 ecj-4.6.3.jar
-rw-r--r-- 1 luis luis 89695 Nov 12 17:34 el-api.jar
-rw-r--r-- 1 luis luis 168381 Nov 12 17:34 jasper-el.jar
-rw-r--r-- 1 luis luis 599430 Nov 12 17:34 jasper.jar
-rw-r--r-- 1 luis luis 26872 Nov 12 17:34 jaspic-api.jar
-rw-rw-r--r 1 luis luis 4936287 Oct 27 2016 jaxrs-ri-2.24.zip
-rw-r--r-- 1 luis luis 62004 Nov 12 17:34 jsp-api.jar
-rw-r--r-- 1 luis luis 246849 Nov 12 17:34 servlet-api.jar
-rw-r--r-- 1 luis luis 10672 Nov 12 17:34 tomcat-api.jar
-rw-r--r-- 1 luis luis 875259 Nov 12 17:34 tomcat-coyote.jar
-rw-r--r-- 1 luis luis 284035 Nov 12 17:34 tomcat-dbcp.jar
-rw-r--r-- 1 luis luis 72678 Nov 12 17:34 tomcat-i18n-de.jar
-rw-r--r-- 1 luis luis 106197 Nov 12 17:34 tomcat-i18n-es.jar
-rw-r--r-- 1 luis luis 155308 Nov 12 17:34 tomcat-i18n-fr.jar
-rw-r--r-- 1 luis luis 174486 Nov 12 17:34 tomcat-i18n-ja.jar
-rw-r--r-- 1 luis luis 173985 Nov 12 17:34 tomcat-i18n-ko.jar

```

```

luis@WebService:~/apache-tomcat-8.5.60/lib$ unzip jaxrs-ri-2.24.zip
Archive:  jaxrs-ri-2.24.zip
  creating: jaxrs-ri/
  inflating: jaxrs-ri/Jersey-LICENSE.txt
  inflating: jaxrs-ri/third-party-license-readme.txt
  creating: jaxrs-ri/api/
  extracting: jaxrs-ri/api/javax.ws.rs-api-2.0.1.jar
  creating: jaxrs-ri/lib/
  extracting: jaxrs-ri/lib/jersey-common.jar
  extracting: jaxrs-ri/lib/jersey-media-jaxb.jar
  extracting: jaxrs-ri/lib/jersey-client.jar
  extracting: jaxrs-ri/lib/jersey-server.jar
  extracting: jaxrs-ri/lib/jersey-container-servlet-core.jar
  extracting: jaxrs-ri/lib/jersey-container-servlet.jar
  creating: jaxrs-ri/ext/
  extracting: jaxrs-ri/ext/javax.inject-2.5.0-b05.jar
  extracting: jaxrs-ri/ext/osgi-resource-locator-1.0.1.jar
  extracting: jaxrs-ri/ext/javax.annotation-api-1.2.jar
  extracting: jaxrs-ri/ext/jersey-guava-2.24.jar
  extracting: jaxrs-ri/ext/hk2-api-2.5.0-b05.jar
  extracting: jaxrs-ri/ext/hk2-utils-2.5.0-b05.jar
  extracting: jaxrs-ri/ext/aopalliance-repackaged-2.5.0-b05.jar
  extracting: jaxrs-ri/ext/hk2-locator-2.5.0-b05.jar
  extracting: jaxrs-ri/ext/javassist-3.20.0-GA.jar
  extracting: jaxrs-ri/ext/validation-api-1.1.0.Final.jar
  extracting: jaxrs-ri/ext/org.osgi.core-4.2.0.jar
  extracting: jaxrs-ri/ext/jaxb-api-2.2.7.jar
  extracting: jaxrs-ri/ext/javax.servlet-api-3.0.1.jar
  extracting: jaxrs-ri/ext/persistence-api-1.0.jar

```

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.

Para lo anterior, nos ubicamos en la carpeta que se descomprimió y nos ubicamos en cada carpeta para escribir el comando `mv *.jar /home/luis/apache-tomcat-8.5.60/lib/` que moverá todos los archivos jar a la carpeta lib de tomcat.

```

luis@WebService:~/apache-tomcat-8.5.60/lib/jaxrs-ri/ext$ mv *.jar /home/luis/apache-tomcat-8.5.60/lib/
luis@WebService:~/apache-tomcat-8.5.60/lib/jaxrs-ri/ext$ ls
luis@WebService:~/apache-tomcat-8.5.60/lib/jaxrs-ri/ext$ cd ..
luis@WebService:~/apache-tomcat-8.5.60/lib/jaxrs-ri$ ls
Jersey-LICENSE.txt  api  ext  lib  third-party-license-readme.txt
luis@WebService:~/apache-tomcat-8.5.60/lib/jaxrs-ri$ cd lib
luis@WebService:~/apache-tomcat-8.5.60/lib/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
luis@WebService:~/apache-tomcat-8.5.60/lib/jaxrs-ri/lib$ mv *.jav /home/luis/apache-tomcat-8.5.60/lib/
mv: cannot stat '*.jav': No such file or directory
luis@WebService:~/apache-tomcat-8.5.60/lib/jaxrs-ri/lib$ mv *.jar /home/luis/apache-tomcat-8.5.60/lib/

```



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). para lo que escribimos el comando `rm -r javax.servlet-api-3.0.1.jar` y con esto quedará eliminado.

```
luis@WebService:~/apache-tomcat-8.5.60/lib$ ls -l javax.servlet-api-3.0.1.jar
ls: cannot access 'javax.servlet-api-3.0.1.jar': No such file or directory
luis@WebService:~/apache-tomcat-8.5.60/lib$
```

Descargar el archivo gson-2.3.1.jar con

wget <https://repo1.maven.org/maven2/com/google/code/gson/gson/2.3.1/gson-2.3.1.jar> y copiarlo en la carpeta lib de tomcat

```
luis@WebService:~/apache-tomcat-8.5.60/lib$ ls -l gson-2.3.1.jar
-rw-rw-r-- 1 luis luis 210856 Nov 20 2014 gson-2.3.1.jar
luis@WebService:~/apache-tomcat-8.5.60/lib$
```

Para descargar el conector JDBC DE MySQL escribimos el comando wget <https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.22.zip> y lo desempaquetamos y copiamos el archivo jar a la carpeta lib de tomcat. Para esto entramos a la carpeta que descomprimos de mysql y tecleamos el comando `mv mysql-connector-java-8.0.22.jar /home/luis/apache-tomcat-8.5.60/lib/`

```
luis@WebService:~/apache-tomcat-8.5.60/lib$ ls -l mysql-connector-java-8.0.22.jar
-rw-r--r-- 1 luis luis 2389216 Sep 10 21:42 mysql-connector-java-8.0.22.jar
luis@WebService:~/apache-tomcat-8.5.60/lib$
```

## Iniciar/detener el servidor Tomcat

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`

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

Iniciar la ejecución de Tomcat ejecutando el siguiente comando:

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

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

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

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

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

## Instalación de MySQL

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

```
sudo apt update
```

Instalar el paquete default de MySQL:

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

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**

Ejecutar el monitor de MySQL:

```
sudo mysql
```



```

luis@WebService:~/apache-tomcat-8.5.60/lib$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)
mysql>
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>

```

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

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

```

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'dis
sidia123';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Dis
sidia$123';
Query OK, 0 rows affected (0.00 sec)
mysql>

```

Actualizar los privilegios:

`FLUSH PRIVILEGES;`

```

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

```

Ejecutar el siguiente comando para salir del monitor de MySQL:

`quit`

```

mysql> quit
Bye
luis@WebService:~/apache-tomcat-8.5.60/lib$

```

## Crear un usuario en MySQL

Ejecutar el monitor de MySQL:

`mysql -u root -p`

Crea el usuario "hugo":

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

Otorgar todos los permisos al usuario "hugo" sobre la base de datos "servicio\_web":

*grant all on servicio\_web.\* to hugo@localhost;*

Ejecutar el siguiente comando para salir del monitor de MySQL:

*quit*

```
luis@WebService:~/apache-tomcat-8.5.60/lib$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
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 'Dissidia$123';
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
luis@WebService:~/apache-tomcat-8.5.60/lib$
```

## Crear la base de datos

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

`mysql -u hugo -p`

Crear la base de datos "servicio\_web":

`create database servicio_web;`

Conectar a la base de datos creada anteriormente:

`use servicio_web;`

```
luis@WebService:~/apache-tomcat-8.5.60/lib$ mysql -u hugo -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
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 database servicio_web;
Query OK, 1 row affected (0.00 sec)

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

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
```

```
);
```



```
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 (0.13 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);  
create unique index usuarios\_1 on usuarios(email);  
Salir del monitor de MySQL:  
quit

```
mysql> alter table fotos_usuarios add foreign key (id_usuario) references usuarios
(id_usuario);
Query OK, 0 rows affected (0.26 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

mysql> quit
Bye
luis@WebService: ~/apache-tomcat-8.5.60/lib$ █
```

## Compilar, empacar y desplegar el servicio web

Descargar desempacar los archivos. Para más comodidad se subieron los archivos a mi repositorio personal de gitHub, de donde descargaremos el repositorio para poder usar los archivos 'Servicio.zip', 'Prueba.html', 'usuario\_sin\_foto.png' y 'WSClient.js'. Todo esto con wget <https://github.com/Wicho1313/Desarrollo-de-Sistemas-Distribuidos/archive/main.zip>

```

luis@WebService:~$ ls
Desarrollo-de-Sistemas-Distribuidos-main  apache-tomcat-8.5.60  prueba.html
Servicio.zip                               apache-tomcat-8.5.60.zip
WSClient.js                               main.zip
luis@WebService:~$

```

Eliminamos la carpeta Desarrollo-de-Sistemas-Distribuidos-main del repositorio de GitHub que descargamos anteriormente, puesto que no lo vamos a utilizar más.

Definir la variable de ambiente CATALINA\_HOME:

*export CATALINA\_HOME=aquí va la ruta completa del directorio de Tomcat 8*

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

```

luis@WebService:~$ 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
luis@WebService:~$ ls
META-INF      WEB-INF      apache-tomcat-8.5.60  negocio
Servicio.zip  WSClient.js  main.zip              prueba.html
luis@WebService:~$

```

Compilar la clase Servicio.java:

```

javac -cp
$CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:.
negocio/Servicio.java

```

```

luis@WebService:~$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
luis@WebService:~$

```

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**.

Para eso tenemos que escribir el comando nano y el nombre del archivo, situarnos en los campos que vamos a editar (username, password) y escribir 'hugo' y 'Dissidia\$123' que previamente configuramos.

```
GNU nano 2.9.3          context.xml          Modified

<Context>
    <Resource name="jdbc/datasource_Servicio" auth="Container" type="javax.sql$
        maxActive="100" maxIdle="30" maxWait="10000"
        username="hugo" password="Dissidia$123"
        driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost/servicio_web?serverTimezone=UTC"/>
</Context>
```

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
```

```
luis@WebService:~$ cp negocio/*.class WEB-INF/classes/negocio/.
luis@WebService:~$ 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)(de
flated 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 = 312) (out= 219)(deflated 29%)
luis@WebService:~$
```

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.

```
luis@WebService:~$ ls -l
total 48
drwxrwxr-x 2 luis luis 4096 Nov 30 22:21 META-INF
-rw-rw-r-- 1 luis luis 7046 Nov 30 22:23 Servicio.war
-rw-rw-r-- 1 luis luis 5351 Nov 30 21:42 Servicio.zip
drwxrwxr-x 3 luis luis 4096 Nov 30 22:19 WEB-INF
-rw-rw-r-- 1 luis luis 3141 Nov 30 21:42 WSClient.js
drwxr-xr-x 9 luis luis 4096 Nov 30 20:24 apache-tomcat-8.5.60
drwxrwxr-x 2 luis luis 4096 Nov 30 22:19 negocio
-rw-rw-r-- 1 luis luis 9084 Nov 30 21:42 prueba.html
luis@WebService:~$ cp Servicio.war apache-tomcat-8.5.60/webapps/
luis@WebService:~$ cd apache-tomcat-8.5.60/webapps/
luis@WebService:~/apache-tomcat-8.5.60/webapps$ ls
ROOT  Servicio  Servicio.war
luis@WebService:~/apache-tomcat-8.5.60/webapps$
```



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.

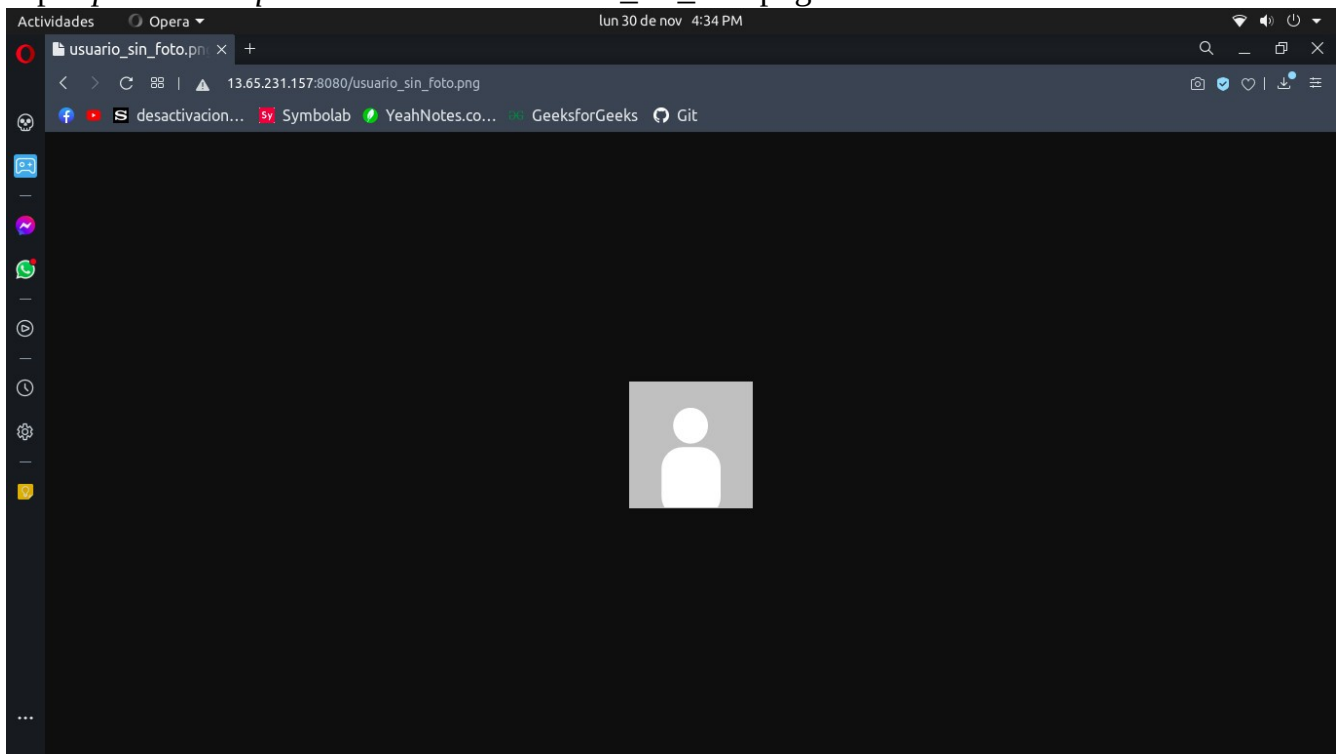
## Probar el servicio web utilizando HTML-Javascript

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](http://ip-de-la-máquina-virtual:8080/usuario_sin_foto.png)



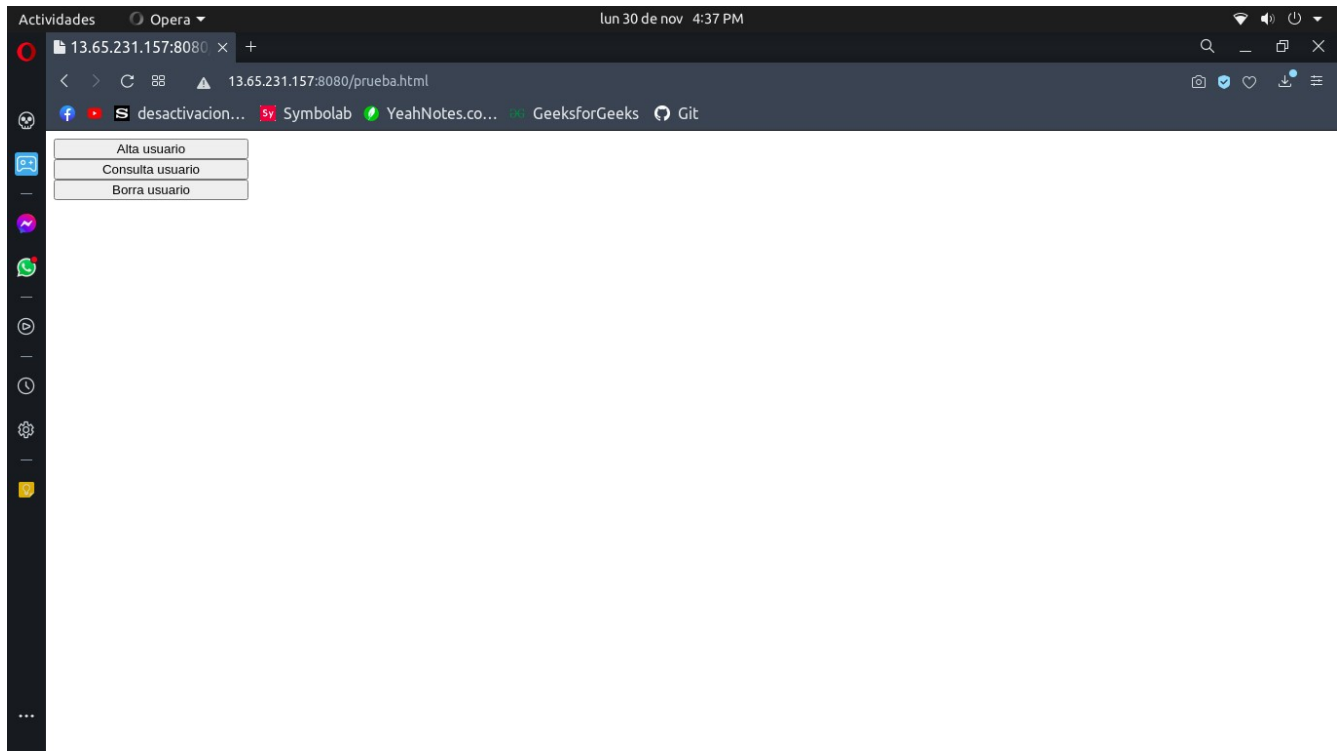
Copiar el archivo [WSClient.js](#) al directorio webapps/ROOT de Tomcat.

Copiar el archivo [prueba.html](#) al directorio webapps/ROOT de Tomcat.

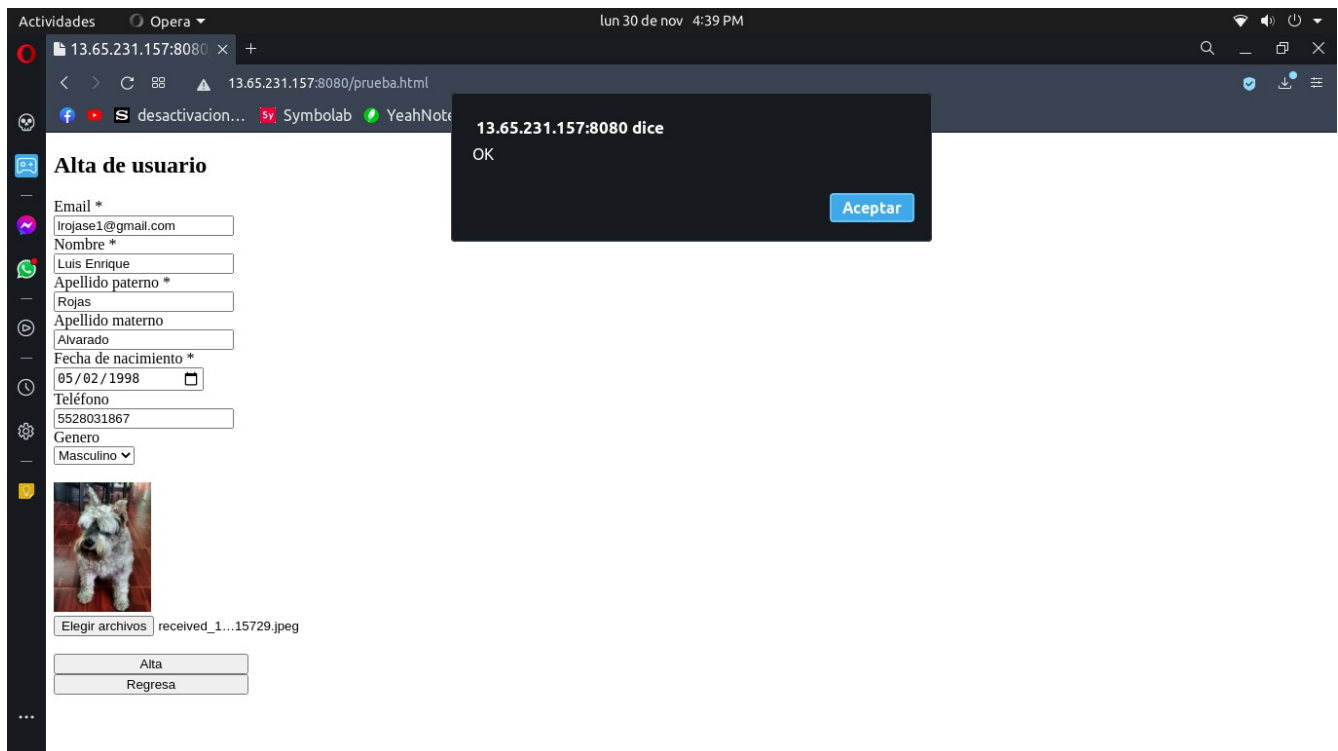
```
luis@WebService:~$ cp usuario_sin_foto.png /home/luis/apache-tomcat-8.5.60/webapps/ROOT/
luis@WebService:~$ cp WSClient.js /home/luis/apache-tomcat-8.5.60/webapps/ROOT/
luis@WebService:~$ cp prueba.html /home/luis/apache-tomcat-8.5.60/webapps/ROOT/
luis@WebService:~$
```

Ingresa la siguiente URL en un navegador:

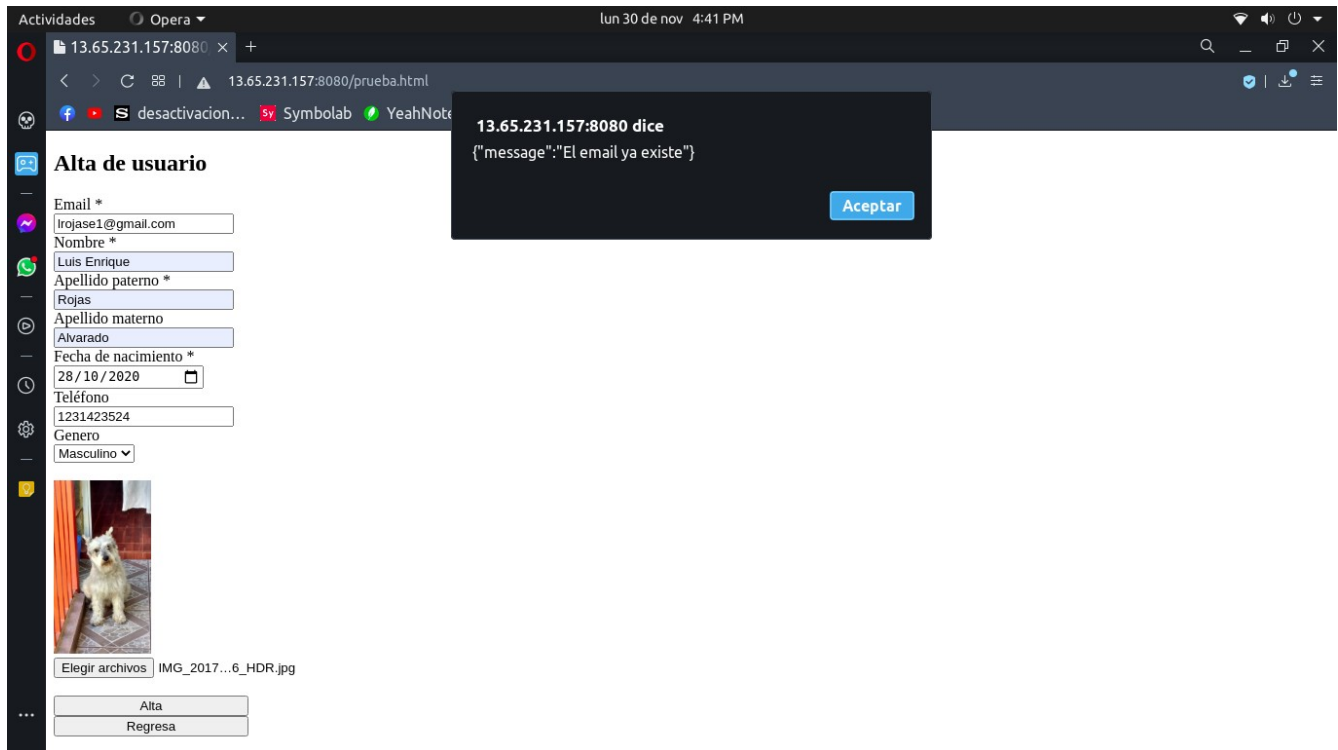
`http://ip-de-la-máquina-virtual:8080/prueba.html`



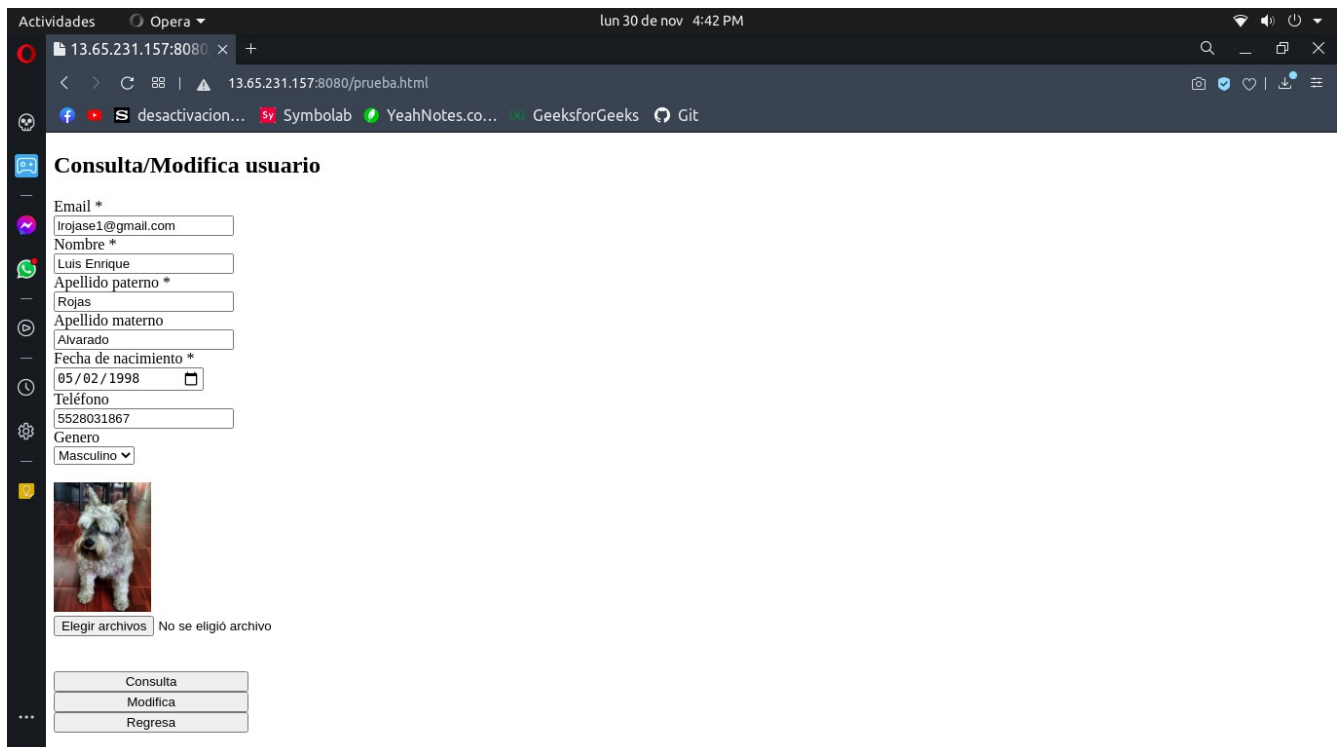
Dar clic en el botón “Alta usuario” para dar de alta un nuevo usuario. Capturar los campos y dar clic en el botón “Alta”.



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

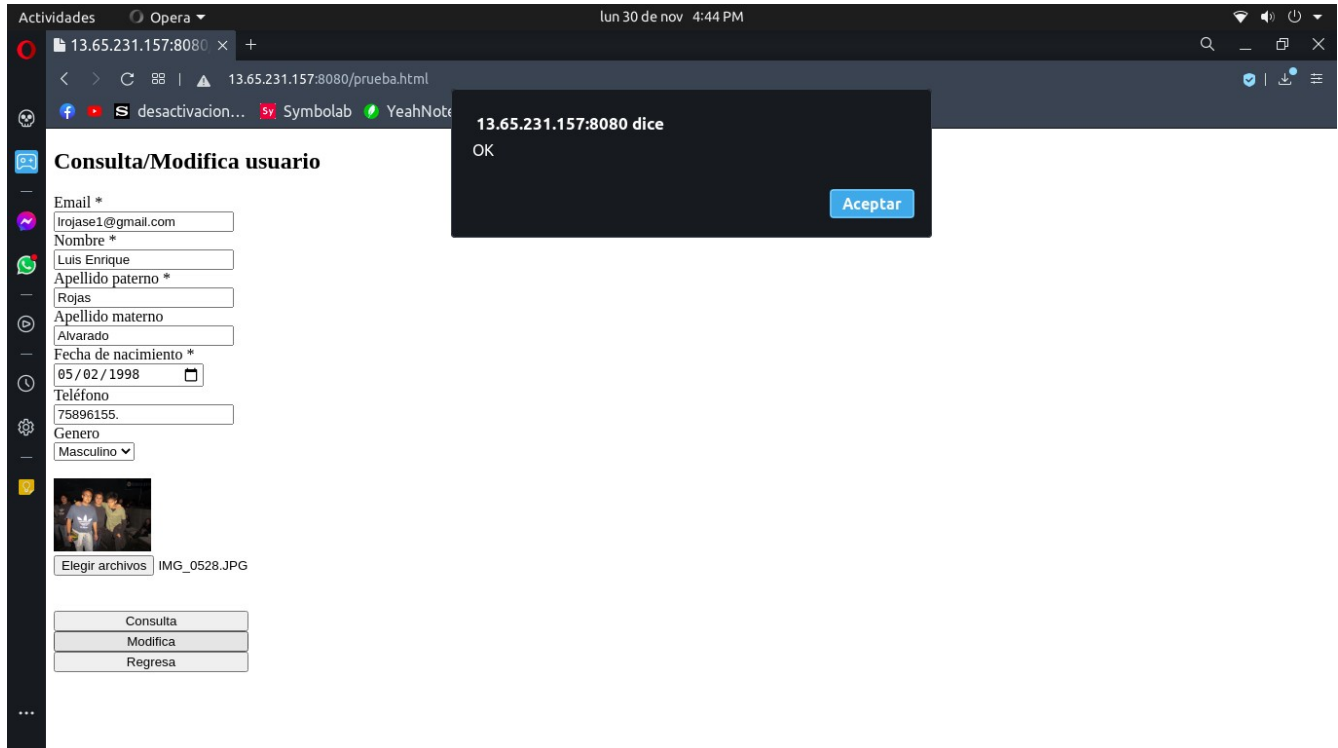


Dar clic en el botón “Consulta usuario” para consultar el usuario dado de alta en el paso anterior Capturar el email y dar clic en el botón “Consulta”,

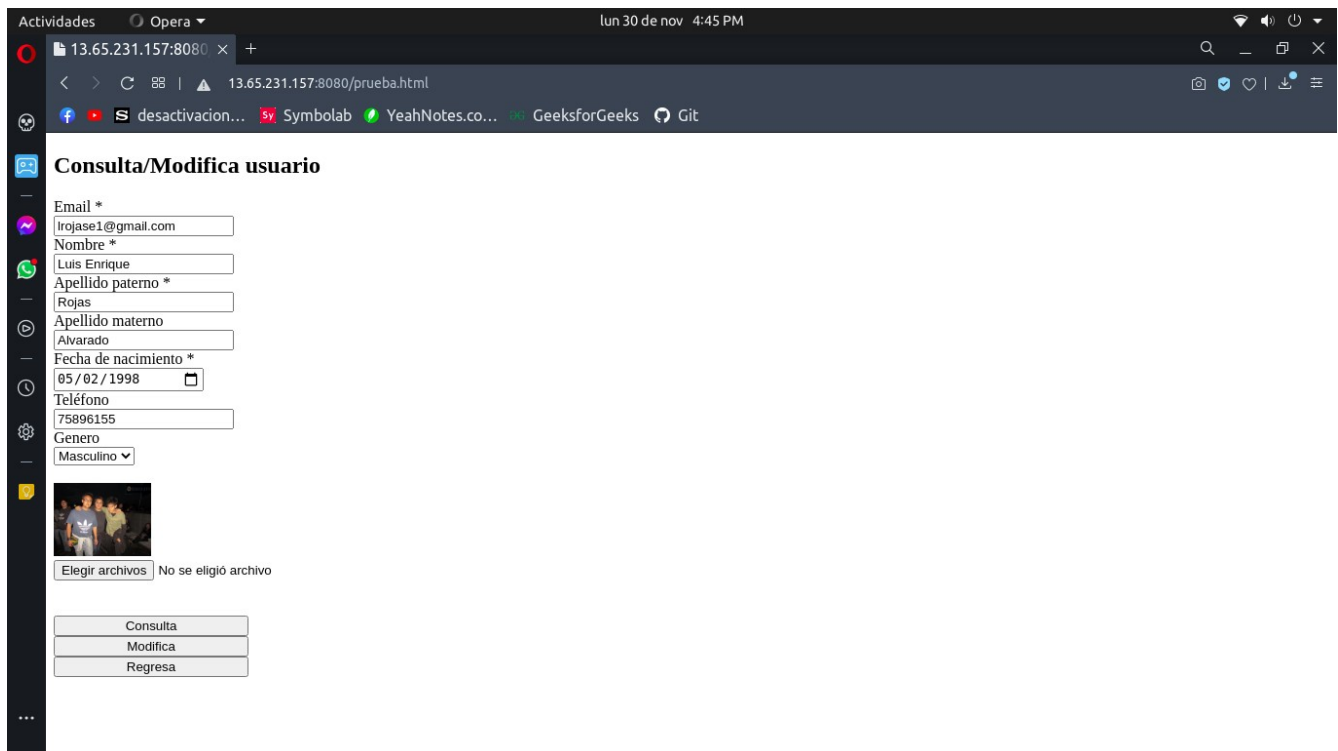




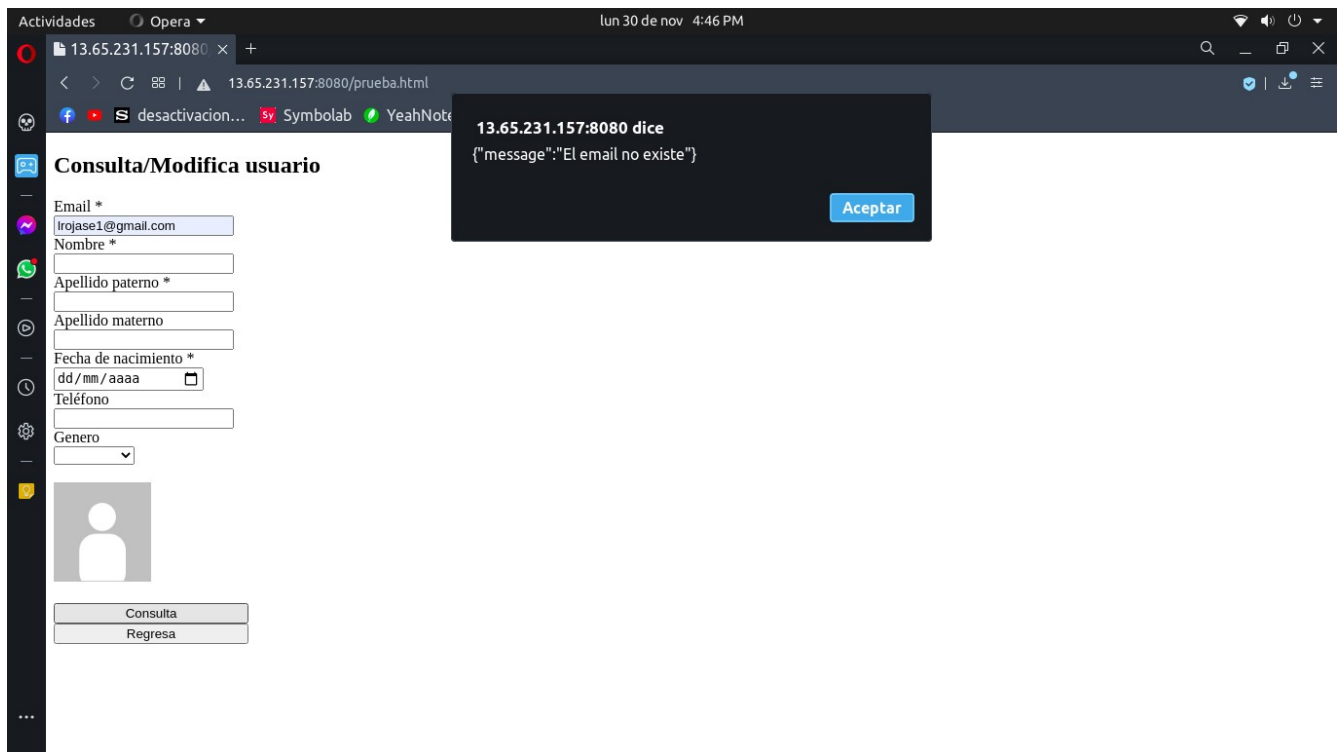
Modificar algún dato del usuario y dar clic en el botón “Modifica”:



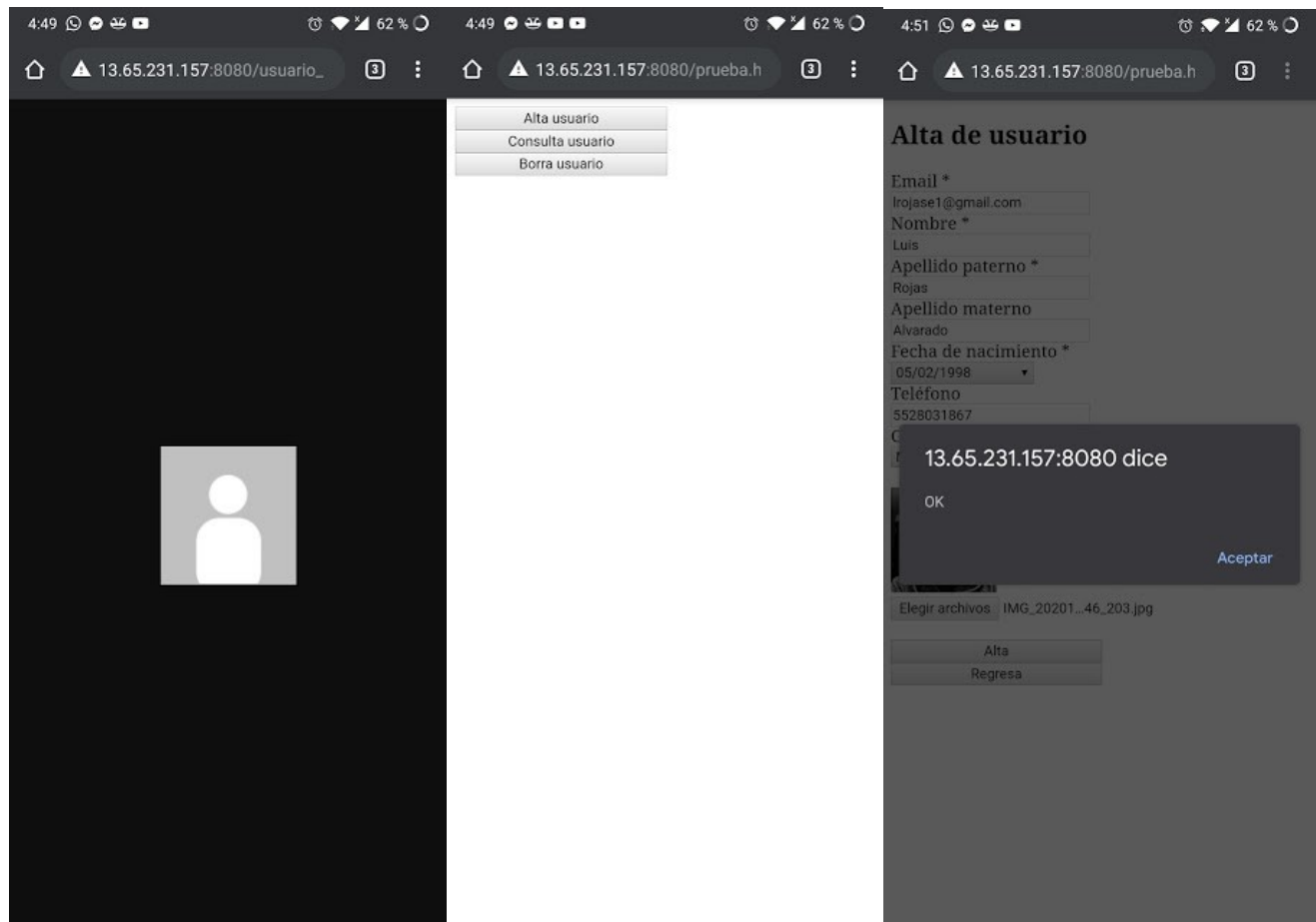
Recargar la página actual y consultar el usuario modificado, para verificar que la modificación se realizó.

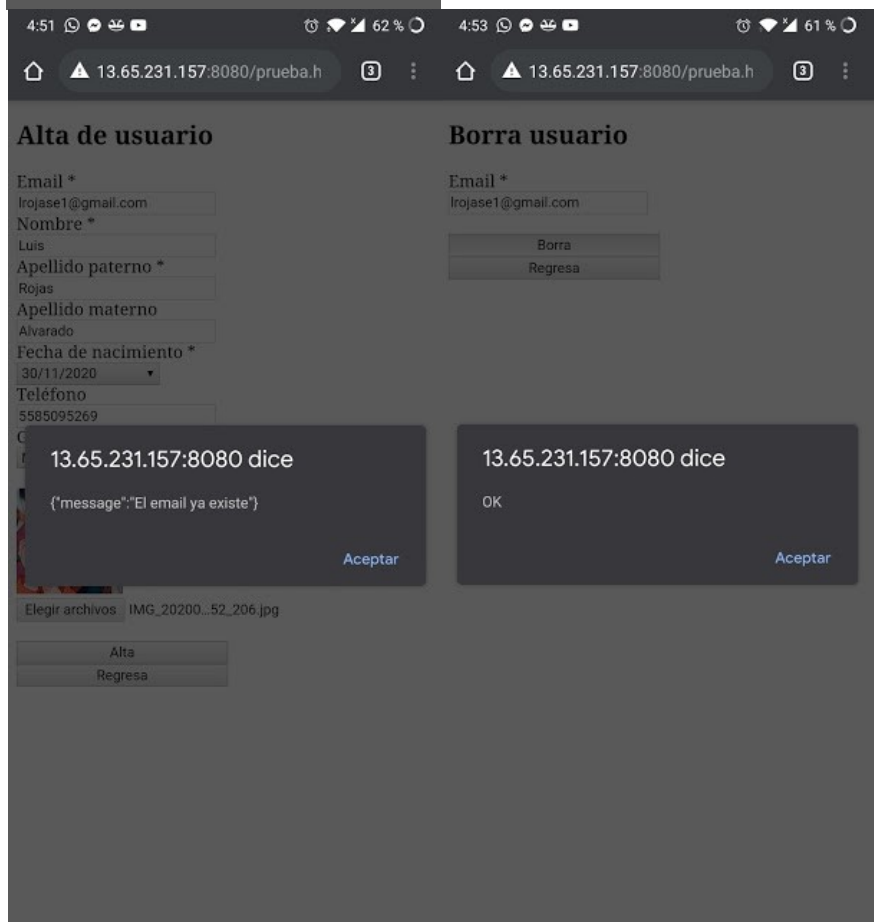
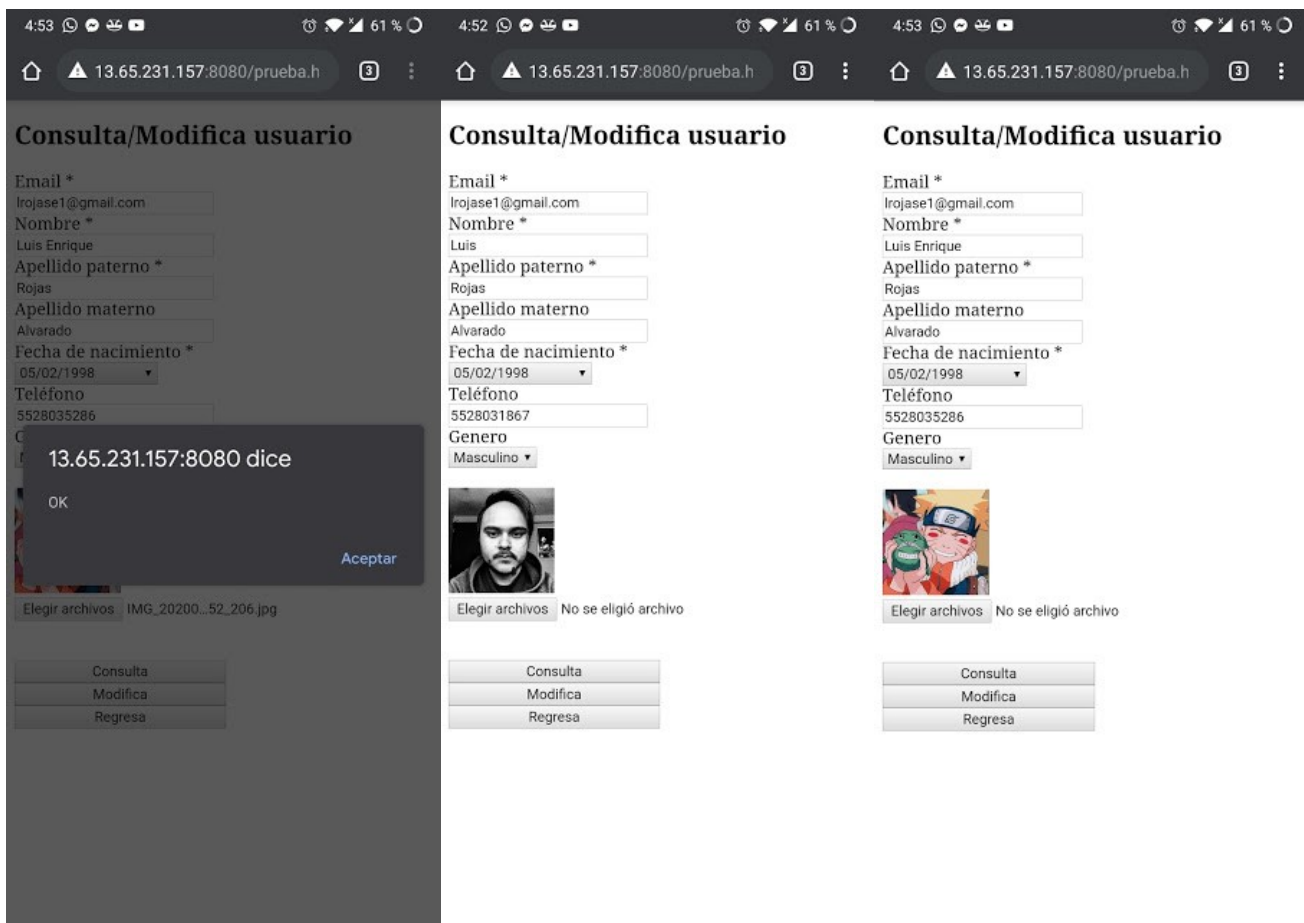


Dar clic en el botón “Borra usuario” para borrar el usuario. Capturar el email del usuario a borrar y dar clic en el botón “Consulta”.



Haciendo las mismas pruebas desde un teléfono móvil.







## **CONCLUSIONES.**

En esta práctica pudimos observar claramente lo que se trata una aplicación REST, y su comportamiento básico. El desplegar la aplicación en una máquina virtual y a su vez montarlo en un servidor web Tomcat, nos hace apreciar el sistema distribuido como si fuese una sola aplicación corriendo en una consola de Linux. Se aprecia correctamente el uso de un back-end por parte de la aplicación, haciendo uso de MySQL como manejador de base de datos y compilando el programa con JDBC para que pudiera tener la relación que esperábamos al conectar la base de datos. Y la interacción que tiene con el front-end por parte de lo que muestra la aplicación al usuario final y su interacción a su vez con este.