



Instituto Politécnico Nacional. Escuela Superior de Cómputo (ESCOM).

Desarrollo de Sistemas Distribuidos.

Tarea 7. “Implementación de un servicio web estilo REST”.

**Profesor Carlos Pineda
Alumno: Sánchez Martínez Eli
Grupo: 4CM5**

01/12/2020

Objetivo: Se deberá probar el servicio web utilizando la aplicación web prueba.html tal como se explicó en clase.

Desarrolló:

Para el desarrollo de esta práctica primero se creó un script recompilando todos los comandos que van desde la instalación del jdk y la ejecución en segundo plano de nuestro sitio web. También dentro del script se le agrego una función llamada spinner la cual solo da el efecto grafico del script.

Script server.sh

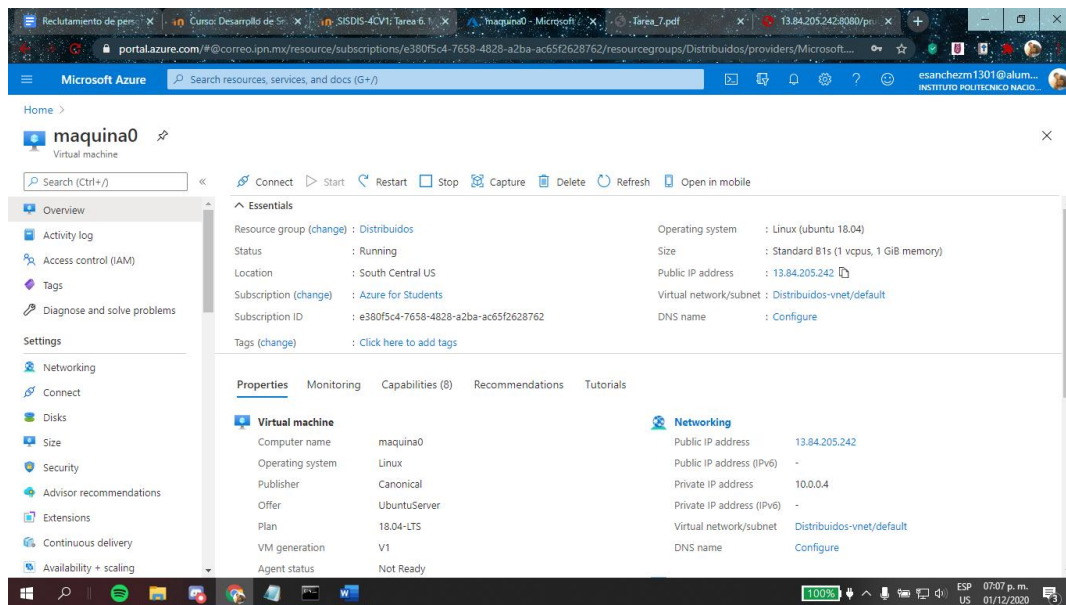
```
1. #!/bin/bash
2. echo "Actualizando el sistema y instalando JDK y mysql"
3. sudo apt -qq update 2>/dev/null > /dev/null
4. sudo apt -qq -o Dpkg::Use-Pty=0 install openjdk-8-jdk-headless mysql-
  server unzip -y -qq 2>/dev/null > /dev/null
5. sleep 3
6. stop_clock $?
7.
8. echo 'Instalando tomcat'
9. sleep 3
10. wget -q https://downloads.apache.org/tomcat/tomcat-8/v8.5.60/bin/apache-tomcat-
    8.5.60.zip
11. unzip -qq apache*.zip
12. rm apache*.zip
13. cd apache*/
14. rm webapps -r
15. mkdir webapps
16. mkdir webapps/ROOT
17. wget -qq https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-
    ri/2.24/jaxrs-ri-2.24.zip
18. unzip -qq jax*.zip
19. rm jax*.zip
20. cp jaxrs-ri/api/*.jar lib
21. cp jaxrs-ri/ext/*.jar lib
22. cp jaxrs-ri/lib/*.jar lib
23. rm jaxrs-ri/ -r
24. rm lib/javax.servlet-api-3.0.1.jar
25. cd lib
26. wget -q https://repo1.maven.org/maven2/com/google/code/gson/gson/2.3.1/gson-
    2.3.1.jar
27. wget -q https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-
    8.0.22.zip
28. unzip -qq mysql*.zip
29. cp mysql*/mysql*.jar .
30. rm mysql*/ -r
31. rm mysql*.zip
32.
33. cd ../
34. export CATALINA_HOME=$(pwd)
35. cd ../
36. export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
37.
38. echo "Configuring MySQL:\n"
39. echo "Please enter 'root' as root's password and 'hugo' as hugo password\n"
40. sudo mysql_secure_installation
```

```

41. sudo mysql < sudo.sql
42. echo "Enter root's password: (root)\n"
43. mysql -u root -p < root.sql
44. echo "Enter hugo's password: (hugo)\n"
45. mysql -u hugo -p < db.sql
46. echo "      Configured MySQL\n"
47.
48. echo 'Iniciando la aplicacion'
49. unzip -qq Servicio.zip
50. rm Servicio.zip
51. cd Servicio
52. javac -cp $CATALINA_HOME/lib/javawebsocket-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
53. rm -f WEB-INF/classes/negocio/*
54. cp negocio/*.class WEB-INF/classes/negocio/
55. jar cvf Servicio.war WEB-INF META-INF > /dev/null
56. rm -f $CATALINA_HOME/webapps/Servicio.war
57. rm -f -r $CATALINA_HOME/webapps/Servicio
58. cp Servicio.war $CATALINA_HOME/webapps
59. cd ../
60. mv usuario_sin_foto.png $CATALINA_HOME/webapps/ROOT/
61. mv WSClient.js $CATALINA_HOME/webapps/ROOT/
62. mv prueba.html $CATALINA_HOME/webapps/ROOT/
63. sh $CATALINA_HOME/bin/catalina.sh start > /dev/null
64. echo "\n Tomcat esta corriendo por el puerto 8080. detener ${CATALINA_HOME}/bin/catalina.sh\n"

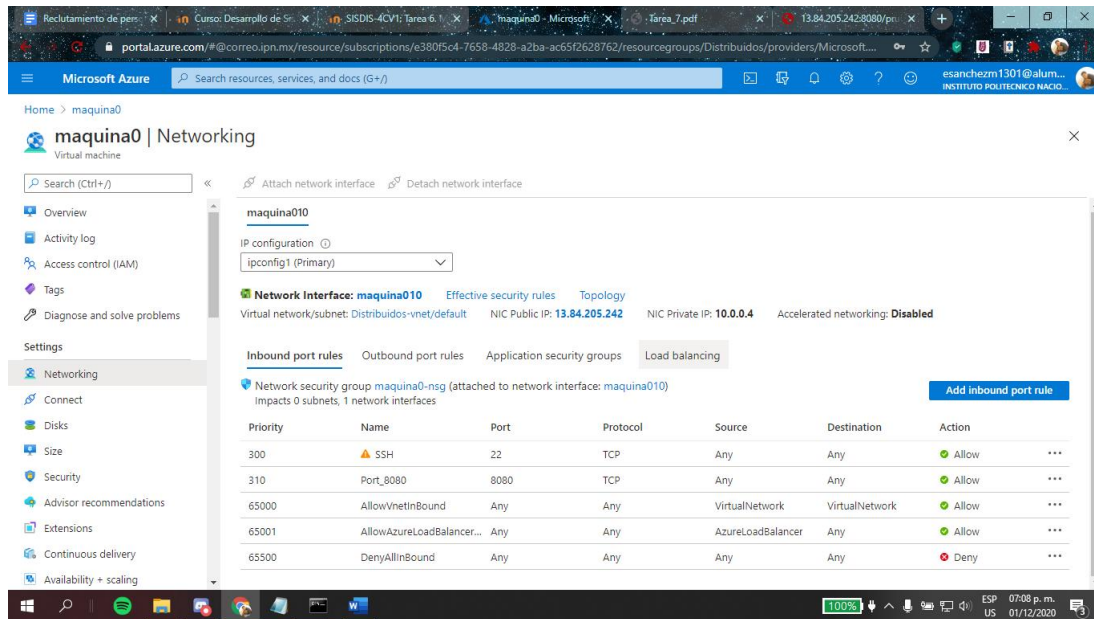
```

El primer paso es crear la máquina virtual en el portal de Azure de Microsoft.



1. Creación de VM.

Después se habilita el puerto 8080 para la interacción con nuestro servidor.



2. *Habilitando el puerto 8080.*

Nos conectamos remotamente a nuestra máquina virtual con el siguiente comando:

```
C:\Users\elit>ssh EliasSAN@13.84.205.242
The authenticity of host '13.84.205.242 (13.84.205.242)' can't be established.
ECDSA key fingerprint is SHA256:Edx/E0BahelF+3sXFq/x12800TRDzS62k+/D+mDGLt8.
Are you sure you want to continue connecting (yes/no)? YES
Warning: Permanently added '13.84.205.242' (ECDSA) to the list of known hosts.
EliasSAN@13.84.205.242's password:
Permission denied, please try again.
EliasSAN@13.84.205.242's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1032-azure x86_64)
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1032-azure x86_64)
```

3. *Conectándome a la VM.*

Una vez adentro procedemos en subir nuestros archivos, en este caso como todos los archivos los guardamos en un zip, el cual contiene adentro nuestro script procedemos a ejecutarlo.

```
C:\Users\elit\Desktop\ESCOM_ELI\Trabajos 6\sistemas-distribuidos\DISTRIBUIDOS\Tarea_7>scp -r 07_tomcat_server.zip EliasSAN@13.84.205.242:~/
EliasSAN@13.84.205.242's password:
07_tomcat_server.zip 100% 12KB 14.3KB/s 00:00
C:\Users\elit\Desktop\ESCOM_ELI\Trabajos 6\sistemas-distribuidos\DISTRIBUIDOS\Tarea_7>
```

4. *Subiendo los archivos a la VM.*

```

EliasSAN@maquina0:~$ sudo apt install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-headers-4.15.0-124
Use 'sudo apt autoremove' to remove it.
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 167 kB of archives.
After this operation, 558 kB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 unzip amd64 6.0-21ubuntu1 [167 kB]
Fetched 167 kB in 0s (3877 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend requires a screen at least 13 lines tall and 31 columns wide.)
debconf: falling back to frontend: Readline
Selecting previously unselected package unzip.
(Reading database ... 76700 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-21ubuntu1_amd64.deb ...
Unpacking unzip (6.0-21ubuntu1) ...
Setting up unzip (6.0-21ubuntu1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

```

5. *Instalando para poder darle unzip*

Una vez subido procedemos a descomprimirlo:

```

EliasSAN@maquina0:~$ unzip 07_tomcat_server.zip
Archive:  07_tomcat_server.zip
  creating: 07_tomcat_server/
  creating: 07_tomcat_server/server/
  inflating: 07_tomcat_server/server/db.sql
  inflating: 07_tomcat_server/server/prueba.html
  inflating: 07_tomcat_server/server/root.sql
  inflating: 07_tomcat_server/server/server.sh
  inflating: 07_tomcat_server/server/Servicio.zip
  inflating: 07_tomcat_server/server/sudo.sql
  inflating: 07_tomcat_server/server/usuario_sin_foto.png
  inflating: 07_tomcat_server/server/WSCClient.js

```

6. *Descomprimiendo los archivos.*

Accedemos a la carpeta `server` para poder ejecutar el script:

```
EliasSAN@maquina0:~$ ls
07_tomcat_server  07_tomcat_server.zip
EliasSAN@maquina0:~$ cd 07_tomcat_server/
EliasSAN@maquina0:~/07_tomcat_server$ ls
server
EliasSAN@maquina0:~/07_tomcat_server$ cd server/
EliasSAN@maquina0:~/07_tomcat_server/server$ ls
Servicio.zip  WSClient.js  db.sql  prueba.html  root.sql  server.sh  sudo.sql  usuario_sin_foto.png
```

7. Accediendo a las carpetas para poder ejecutar el comando.

Procedemos a darle permiso para que se pueda ejecutar el script y lo ejecutamos:

```
EliasSAN@maquina0:~/07_tomcat_server/server$ chmod +x server.sh
EliasSAN@maquina0:~/07_tomcat_server/server$ ./server.sh
```

8. Ejecutando el script

Se instalan todas las dependencias, se borran los archivos que no son necesarios, se ejecuta para la configuración de administrador de sql y se ejecuta el servidor en el background.

```
EliasSAN@maquina0:~/07_tomcat_server/server$ ./server.sh
Actualizando el sistema y instalando JDK y mysql
./server.sh: line 6: stop_clock: command not found
Instalando tomcat
Configuring MySQL:\n
Please enter 'root' as root's password and 'hugo' as hugo password\n

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.
```

9. Se configura el sql.

```

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.

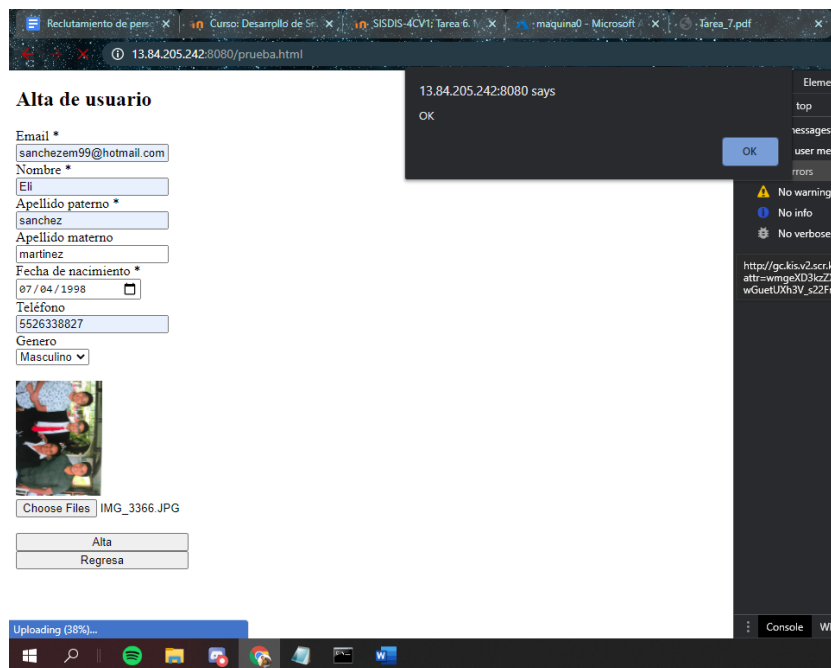
Reload privilege tables now? (Press y|Y for Yes, any other key for No) :

... skipping.
All done!
Enter root's password: (root)\n
Enter password:
./server.sh: line 44: echoEnter hugo's password: (hugo)\n: command not found
Enter password:
  Configured MySQL\n
Iniciando la aplicacion
\n Tomcat esta corriendo por el puerto 8080. detener /home/EliasSAN/07_tomcat_server/server/apache-tomcat-8.5.60/bin/catalina.sh\n

```

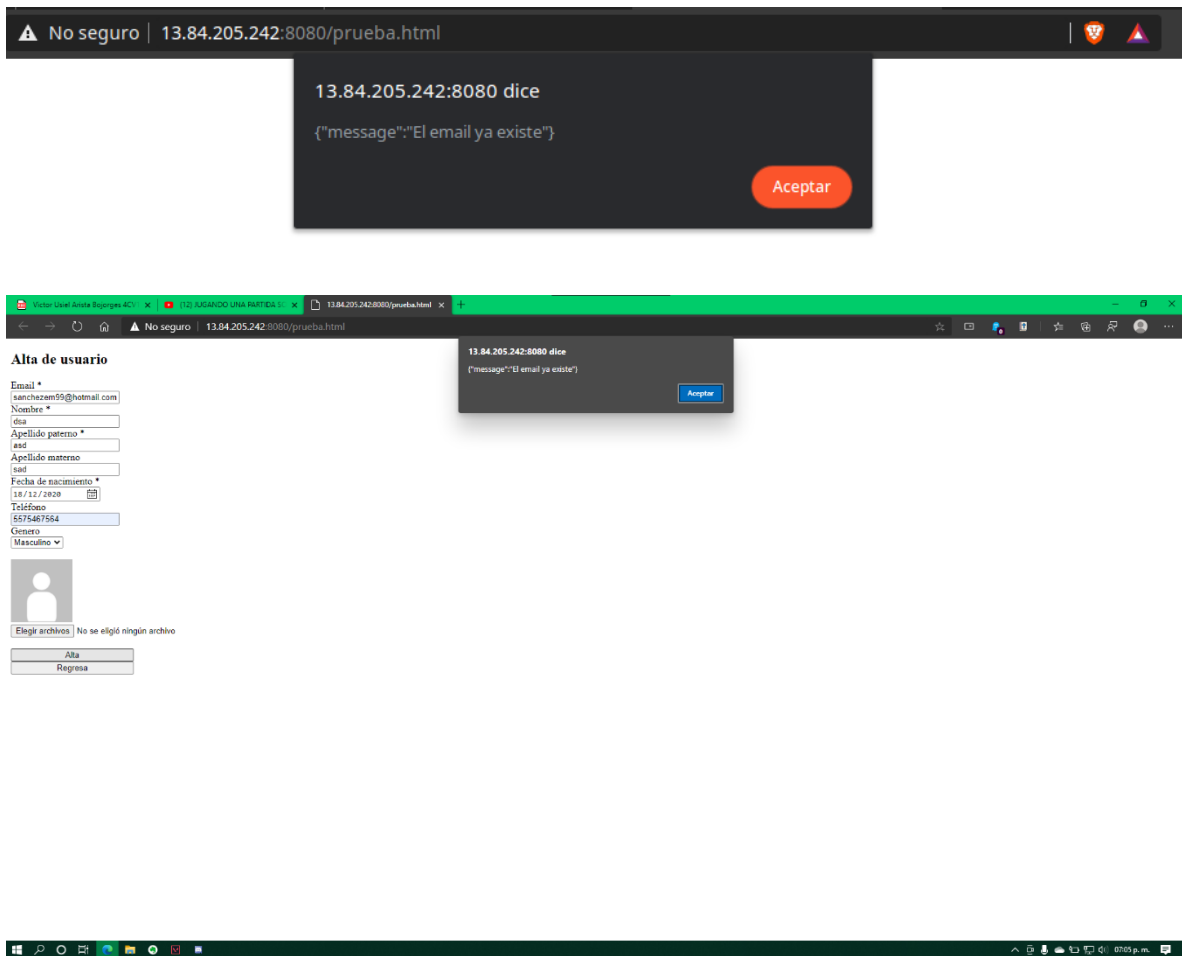
10. Enviamos el server a segundo plano para poder conectarnos.

Se accede a la página prueba.html y se da de alta a un usuario.



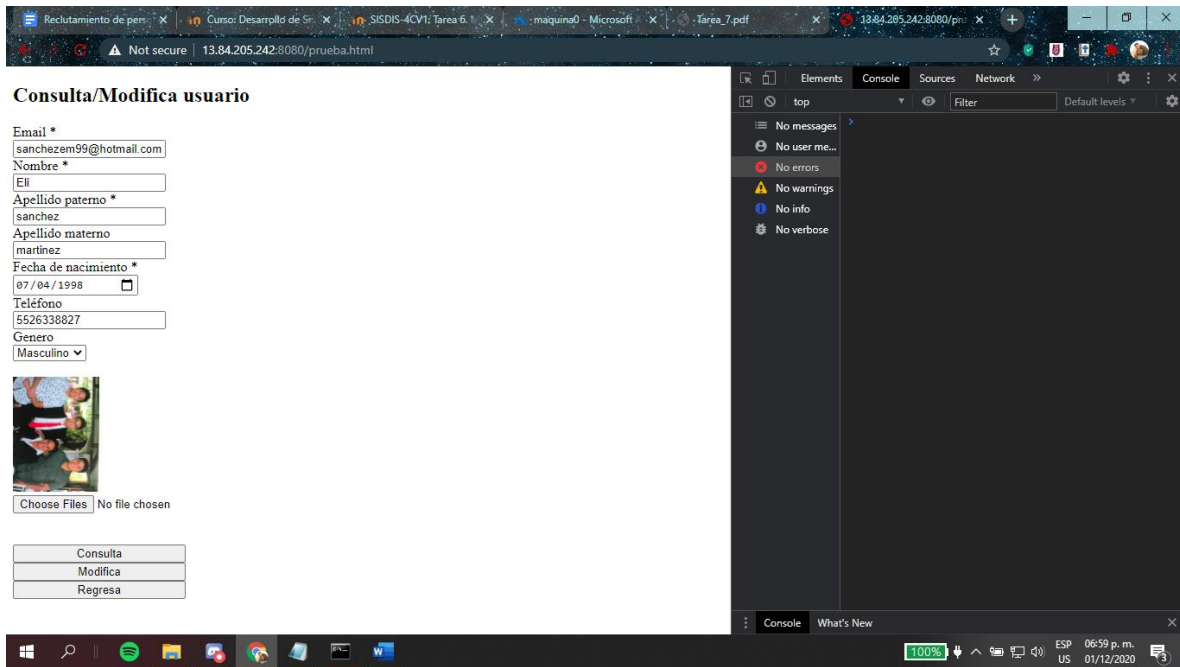
11. Se agrega un usuario a nuestra prueba.

Se agrega a otro usuario con el mismo correo para que nos aparezca la excepción, usando varias computadoras.



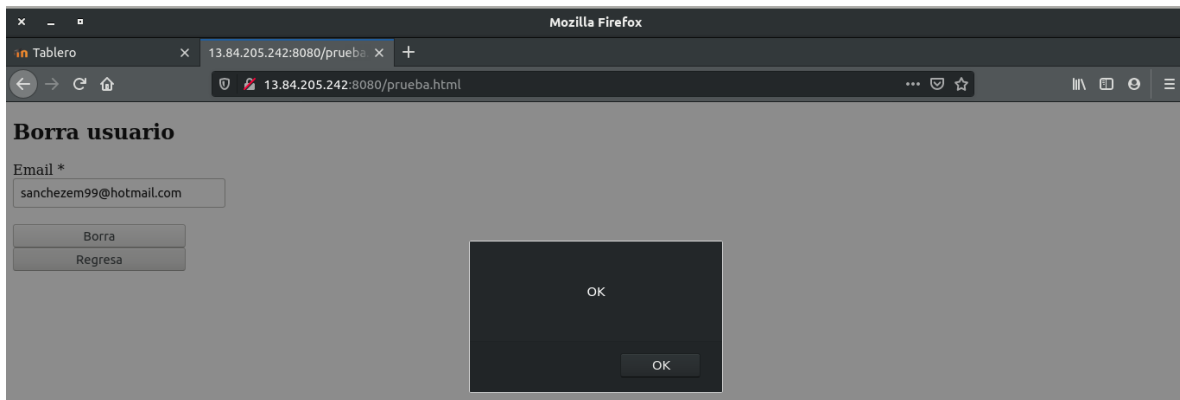
12. Se comprueba que existe la excepción.

Después se hace una consulta al usuario.



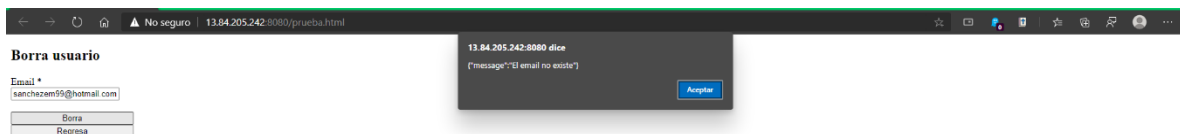
13. Consulta de usuario que se había agregado.

Se elimina el usuario que se dio de alta desde otra máquina.



14. Se comprueba que se eliminó el usuario.

Se confirma que se haya eliminado el usuario, desde otra computadora.



15. Confirmando que se haya eliminado el usuario.

Ingresamos desde el celular a nuestra prueba.html.



The screenshot shows a mobile browser interface with a status bar at the top displaying the time 7:08, signal strength, and battery level. The address bar shows 'Not Secure' and the URL '13.84.205.242'. The page title is 'Alta de usuario'. The form includes the following fields: Email (with a dropdown showing 'sanchez99@hotmail.com'), Nombre (with a dropdown showing 'Ela'), Apellido paterno (with a dropdown showing 'Aja'), Apellido materno (with a dropdown showing 'Haja'), Fecha de nacimiento (with a dropdown showing 'Nov 15, 2020'), Teléfono (with a text input showing '618919283281'), and Género (with a dropdown showing 'Masculino'). Below the form is a photo upload section with a preview of a dog's face and a 'Choose Files' button. At the bottom are 'Alta' and 'Regresa' buttons. The mobile OS navigation bar is visible at the very bottom.

16.Ingresando desde el celular.

Conclusión.

Realizar esta práctica basada en webservices usando REST nos permite saber el como trabajan algunas aplicaciones a la hora de almacenar y manejar ciertos valores de usuarios. Sin embargo, el uso de java en esta parte podría facilitarnos y seria otra alternativa muy viable. Esta tarea nos ayuda a comprende mas el uso entre front-end y back-end.