

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.

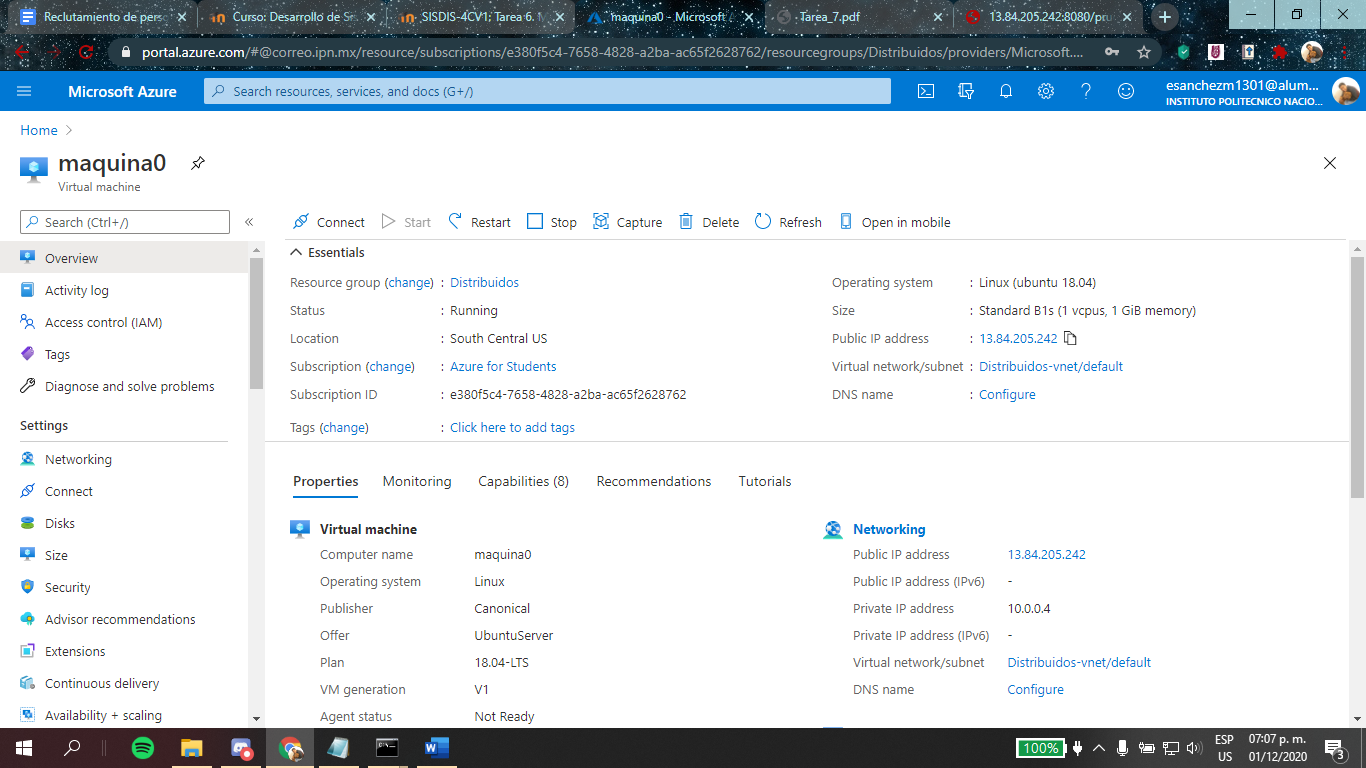
**Desarrolló:**

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

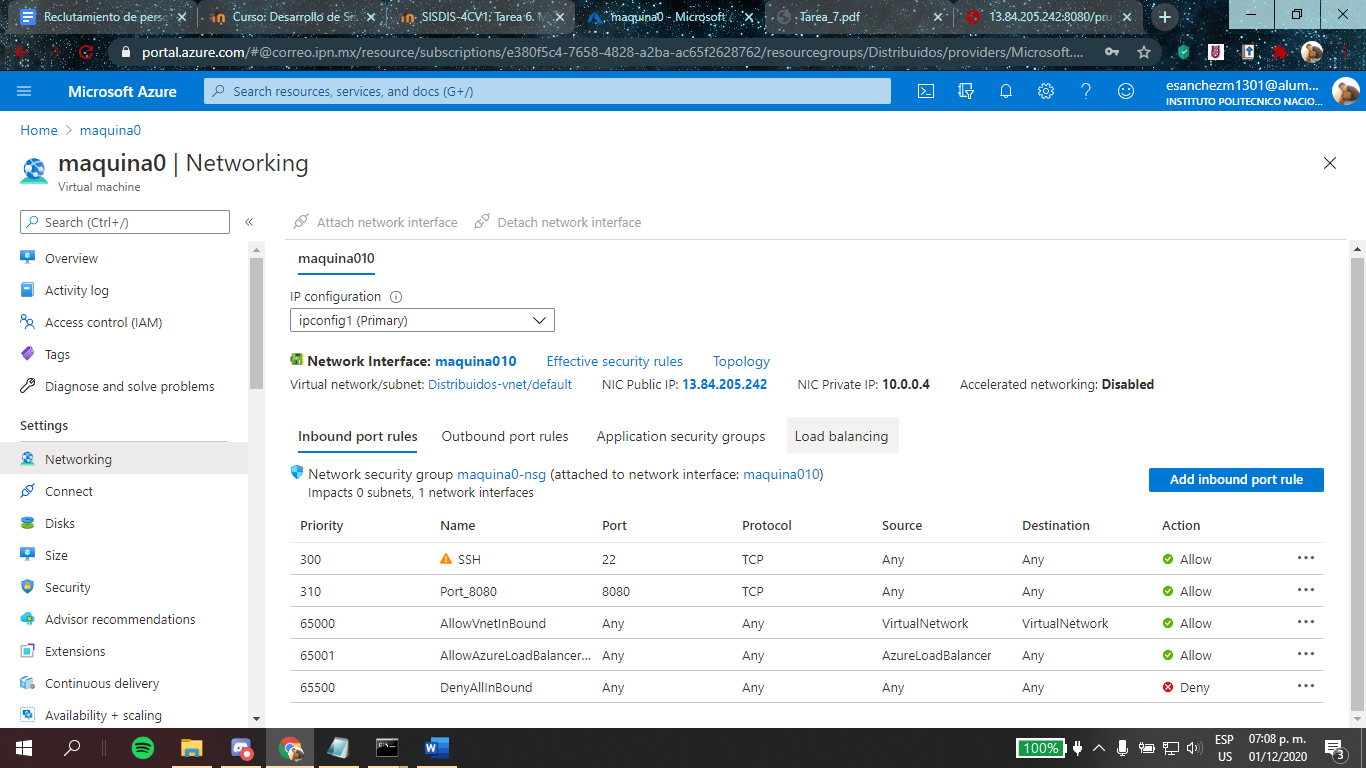
1. #!/bin/bash
2. echo "Actualizando el sistema y instalando JDK y mysql"
3. sudo apt -qq update 2>/dev/null > /dev/null
4. sudo apt -qq -o Dpkg::Use-Pty=0 install openjdk-8-jdk-headless mysql-server unzip -y -qq 2>/dev/null > /dev/null
5. sleep 3
6. stop\_clock $?
8. echo 'Instalando tomcat'
9. sleep 3
10. wget -q https://downloads.apache.org/tomcat/tomcat-8/v8.5.60/bin/apache-tomcat-8.5.60.zip
11. unzip -qq apache\*.zip
12. rm apache\*.zip
13. cd apache\*/
14. rm webapps -r
15. mkdir webapps
16. mkdir webapps/ROOT
17. wget -qq https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.24/jaxrs-ri-2.24.zip
18. unzip -qq jax\*.zip
19. rm jax\*.zip
20. cp jaxrs-ri/api/\*.jar lib
21. cp jaxrs-ri/ext/\*.jar lib
22. cp jaxrs-ri/lib/\*.jar lib
23. rm jaxrs-ri/ -r
24. rm lib/javax.servlet-api-3.0.1.jar
25. cd lib
26. wget -q https://repo1.maven.org/maven2/com/google/code/gson/gson/2.3.1/gson-2.3.1.jar
27. wget -q https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.22.zip
28. unzip -qq mysql\*.zip
29. cp mysql\*/mysql\*.jar .
30. rm mysql\*/ -r
31. rm mysql\*.zip
33. cd ../
34. export CATALINA\_HOME=$(pwd)
35. cd ../
36. export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64
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"
48. echo 'Iniciando la aplicacion'
49. unzip -qq Servicio.zip
50. rm Servicio.zip
51. cd Servicio
52. javac -cp $CATALINA\_HOME/lib/javax.ws.rs-api-2.0.1.jar:$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.



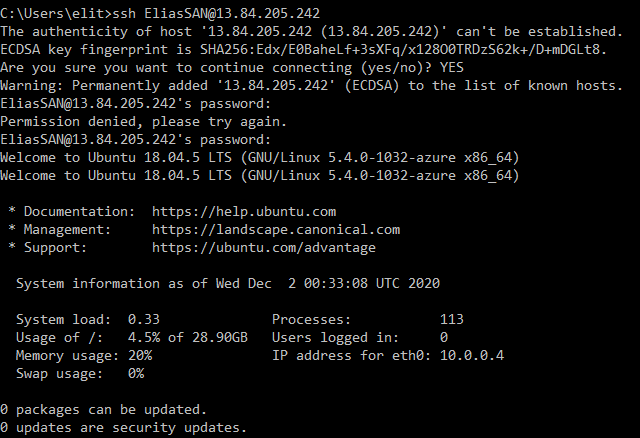
**1**. Creación de VM.

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



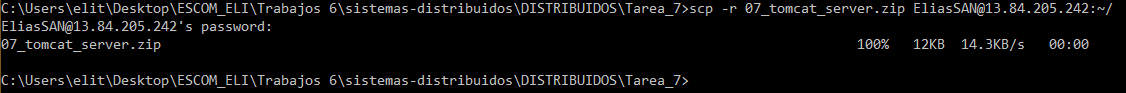
2. Habilitando el puerto 8080.

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

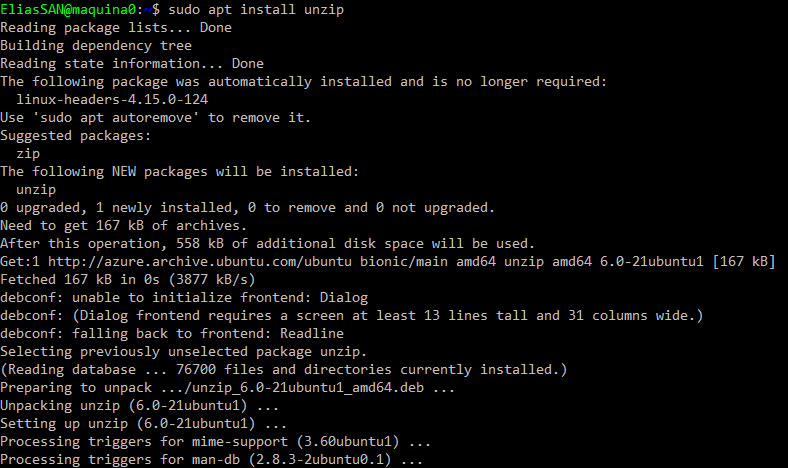


3. Conectándome a la VM.

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.

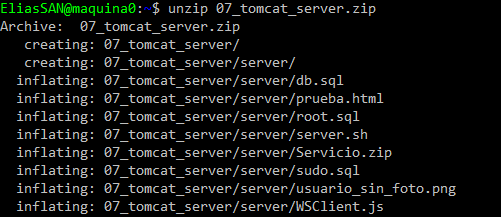


4. Subiendo los archivos a la VM.



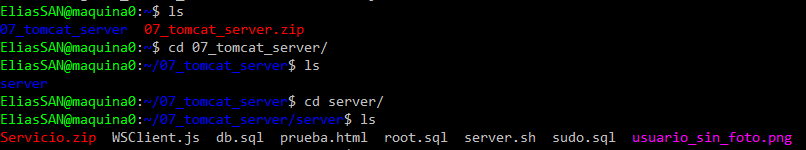
5. Instalando para poder darle unzip

Una vez subido procedemos a descomprimirlo:



**6.** Descomprimiendo los archivos.

Accedemos a la carpeta server para poder ejecutar el script:



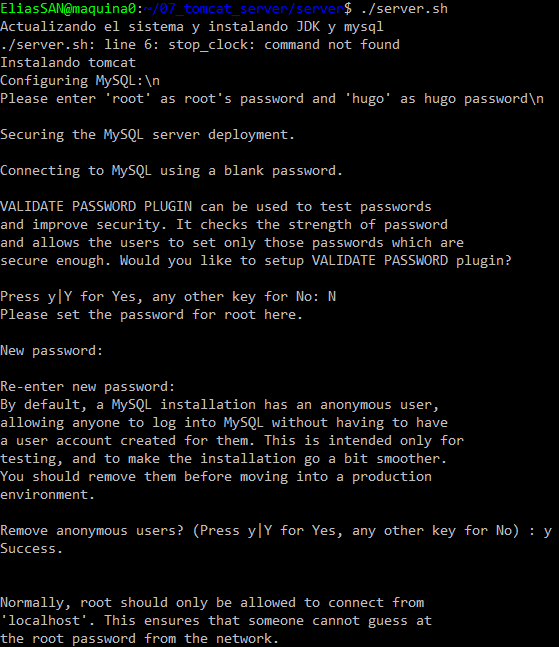
7. Accediendo a las carpetas para poder ejecutar el comando.

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

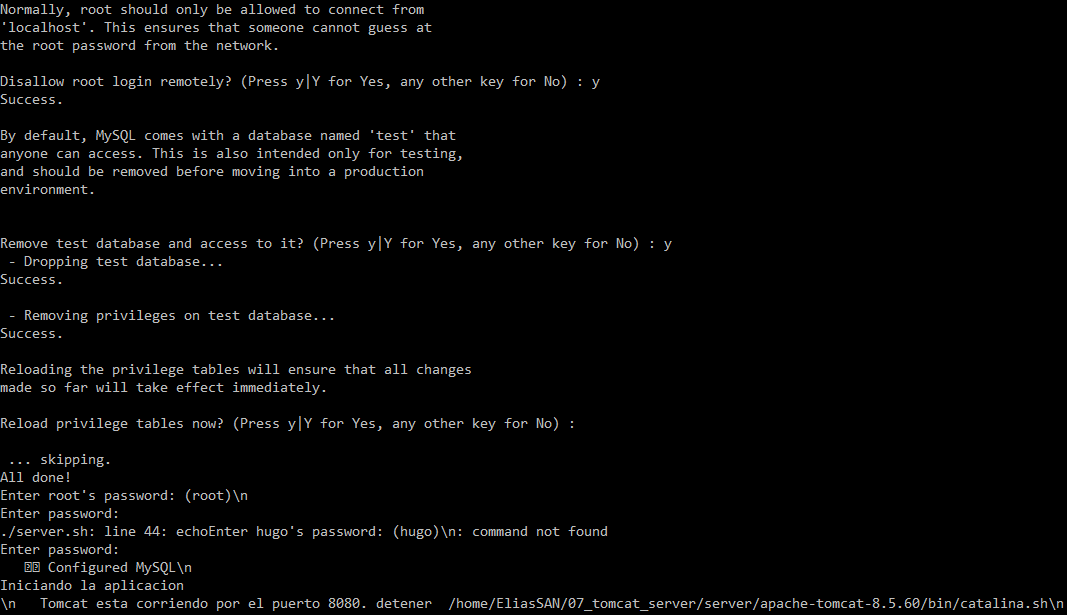


8. Ejecutando el script

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.

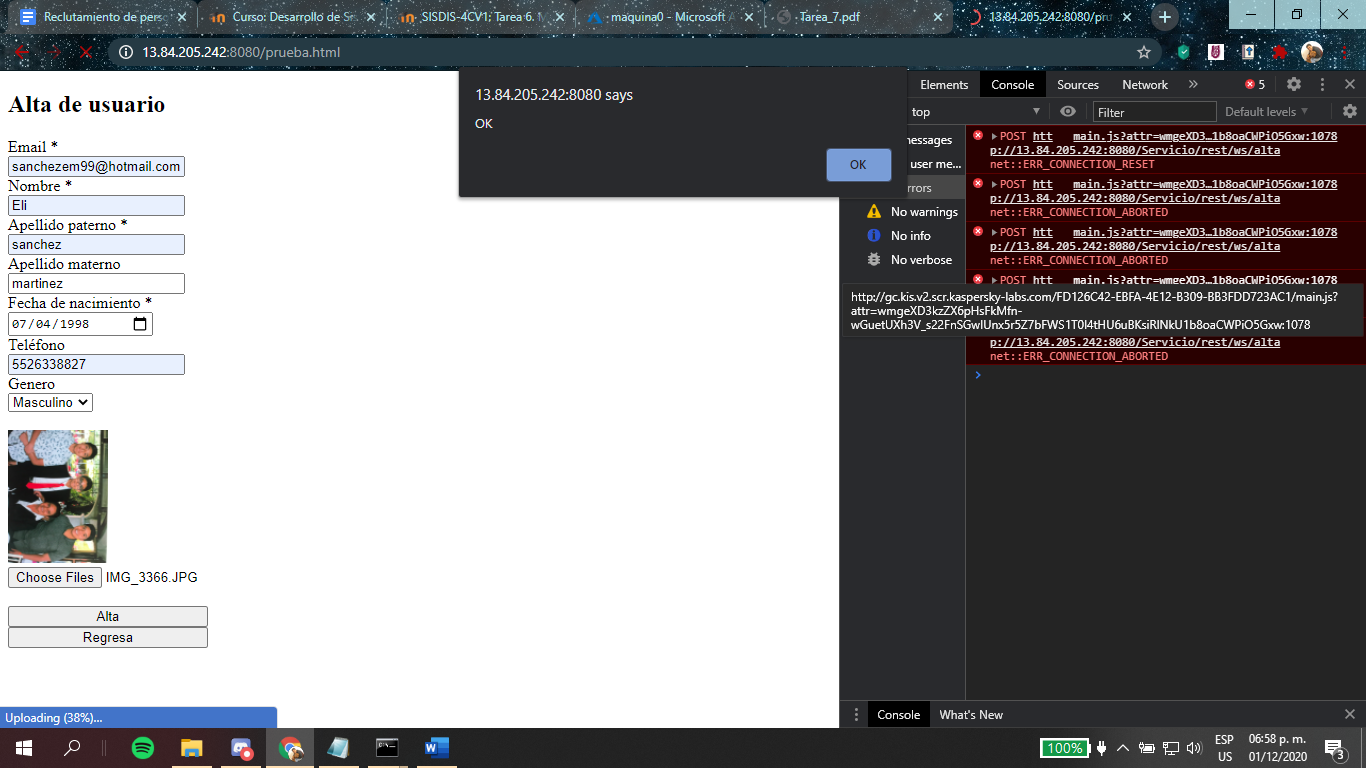


9. Se configura el sql.



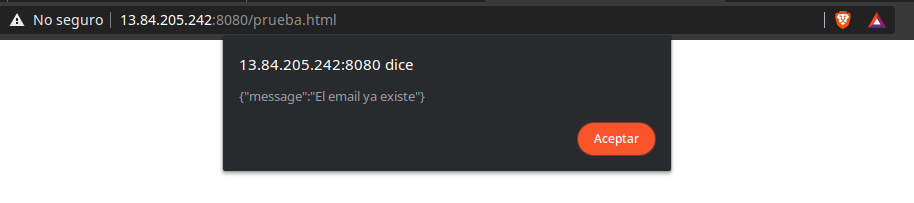
10. Enviamos el server a segundo plano para poder conectarnos.

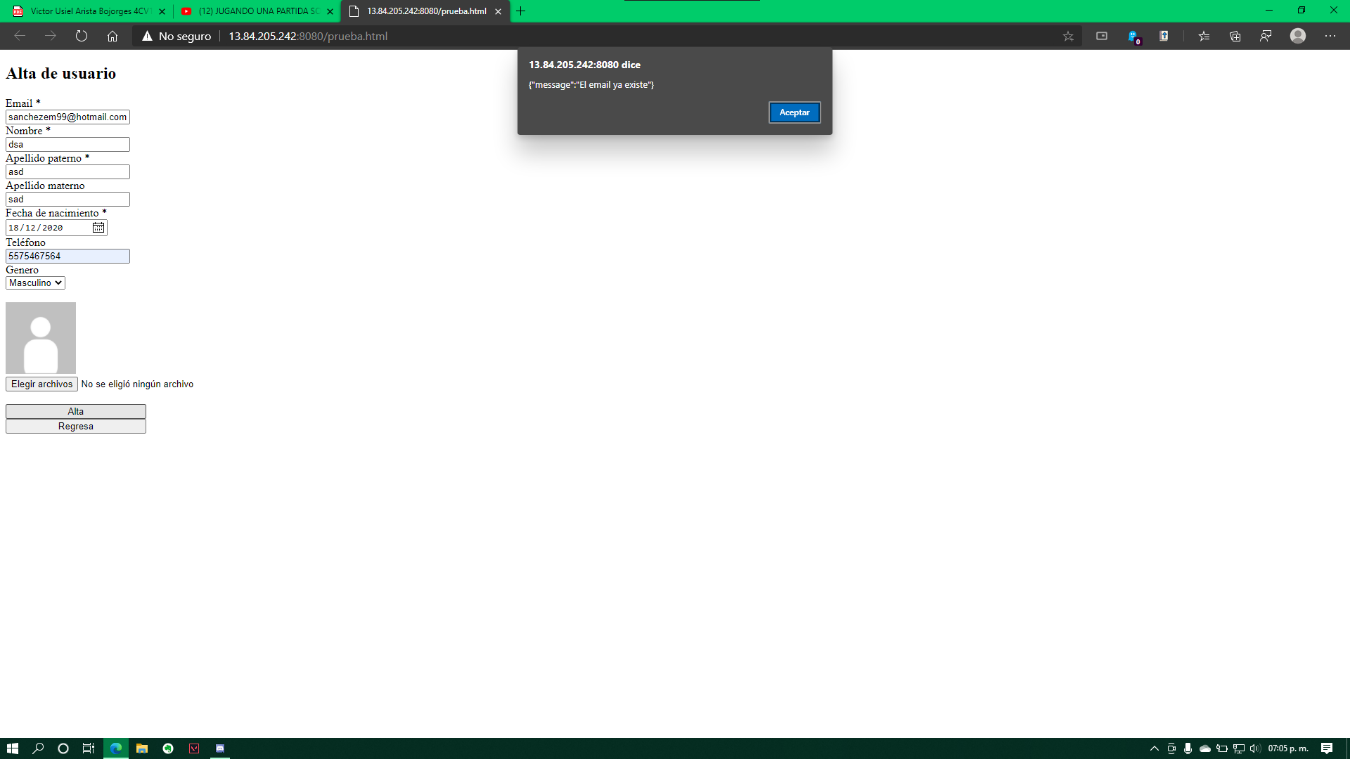
Se accede a la página prueba.html y se da de alta a un usuario.



11. Se agrega un usuario a nuestra prueba.

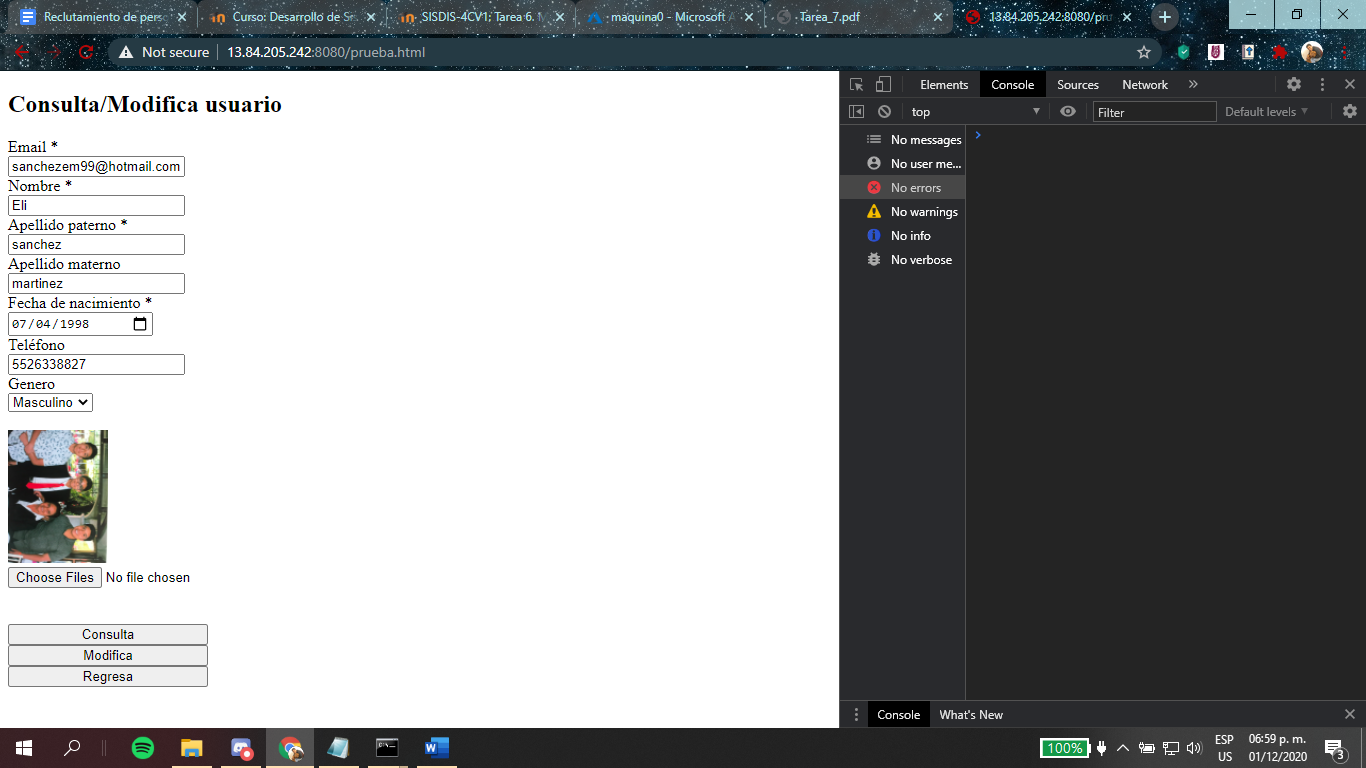
Se agrega a otro usuario con el mismo correo para que nos aparezca la excepción, usando varias computadoras.





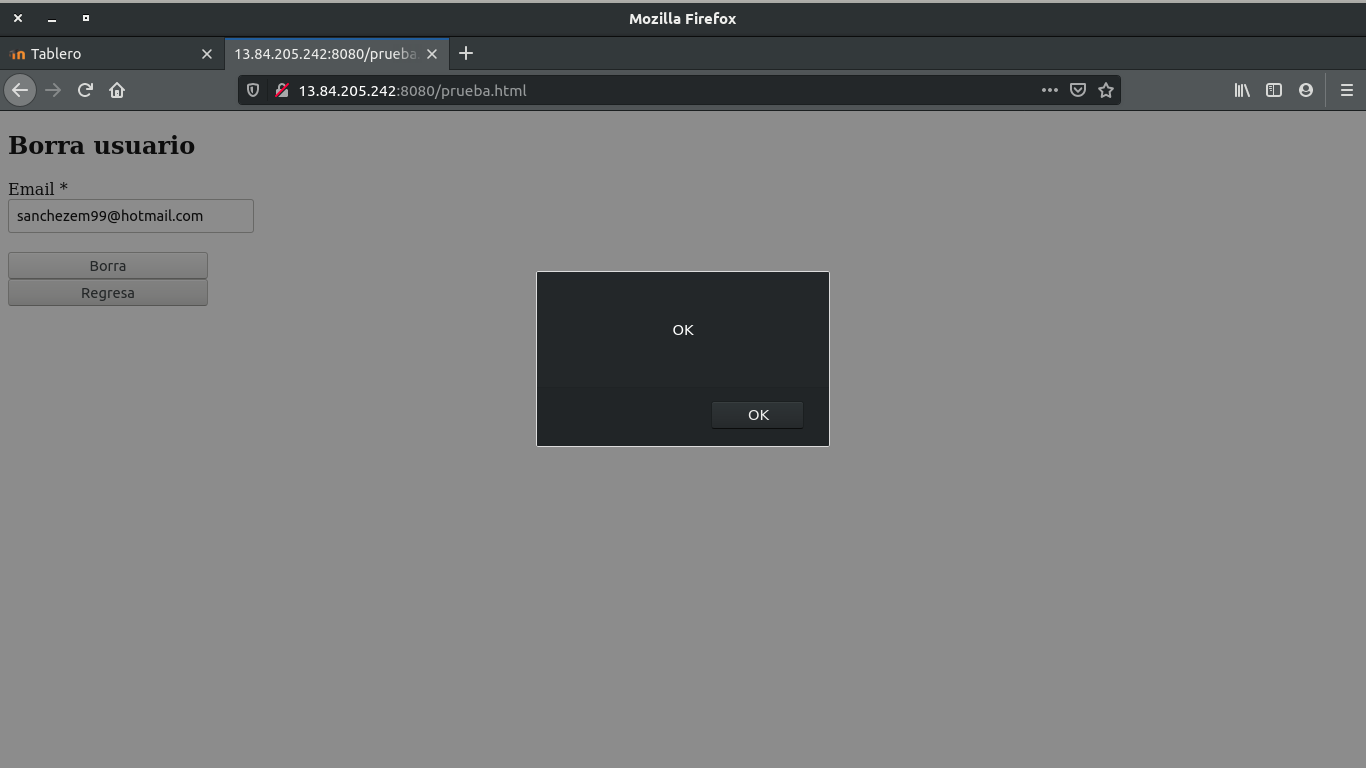
12. Se comprueba que existe la excepción.

Después se hace una consulta al usuario.



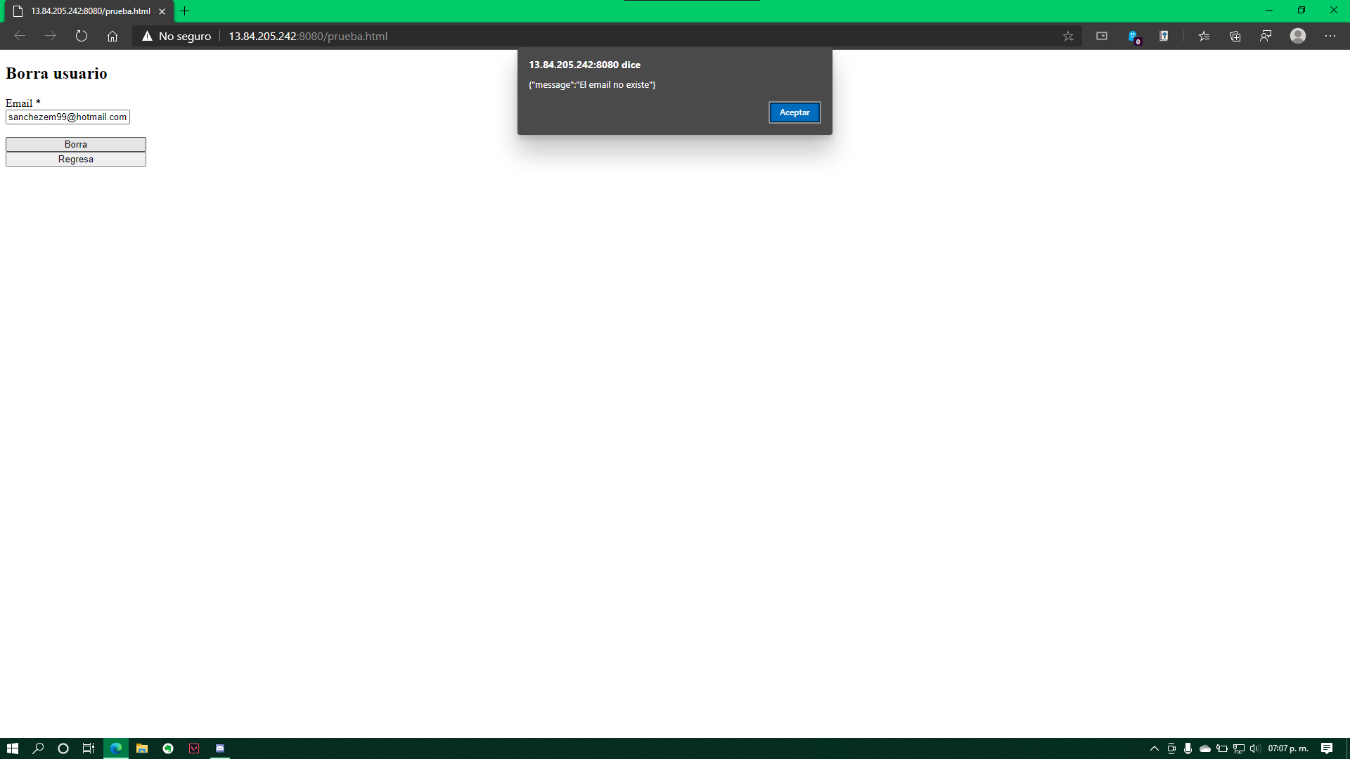
13. Consulta de usuario que se había agregado.

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.



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.