

Tarea 7

Desarrollo de Sistemas Distribuidos

Implementación de un servicio web estilo REST

**2020**

**Eduardo Alberto Pereda Guzman**

**Prof. Pineda Guerrero Carlos**

**27 de noviembre**

# Introducción.

REST define un conjunto de principios arquitectónicos para la creación de servicios web. REST fue presentado por Roy Fielding el año 2000 en su disertación doctoral "Architectural Styles and the Design of Network-based Software Architectures".

El diseño de servicios web estilo REST sigue cuatro principios:

* Utilizar métodos HTTP de forma explícita.
  + Un servicio web utiliza los métodos de HTTP para crear un recurso (POST), leer (GET), cambiar el estado o actualizar un recurso (PUT), y borrar un recurso (DELETE).
* Los servicios son sin estado (stateless).
  + Los clientes de servicios web estilo REST deben enviar peticiones completas e independientes, es decir, las peticiones deben incluir todos los datos que permitan completar el servicio, sin la necesidad de guardar un estado entre peticiones.
* Los URIs representan una estructura de directorios.
  + Los URIs (Uniform Resource Identifier) deben ser intuitivos y auto-explicados. Un URI es una jerarquía que corresponde a la estructura de los servicios web definidos en la empresa.
* Se transfiere XML, JSON o ambos.
  + Los recursos que provee un servicio web pueden ser documentos, imágenes, videos y en general objetos. La representación de objetos mediante XML o JSON es fácil e independiente de la plataforma.

# Desarrollo.

**Instalación de Tomcat con soporte REST**

1. Crear una máquina virtual con Ubuntu 18 con al menos 1GB de memoria RAM. Abrir el puerto 8080 para el protocolo TCP.

**Se creó una máquina virtual llamada “ServidorREST” con una ip publica: 20.185.80.52, además que se habilitó el puerto 8080.**

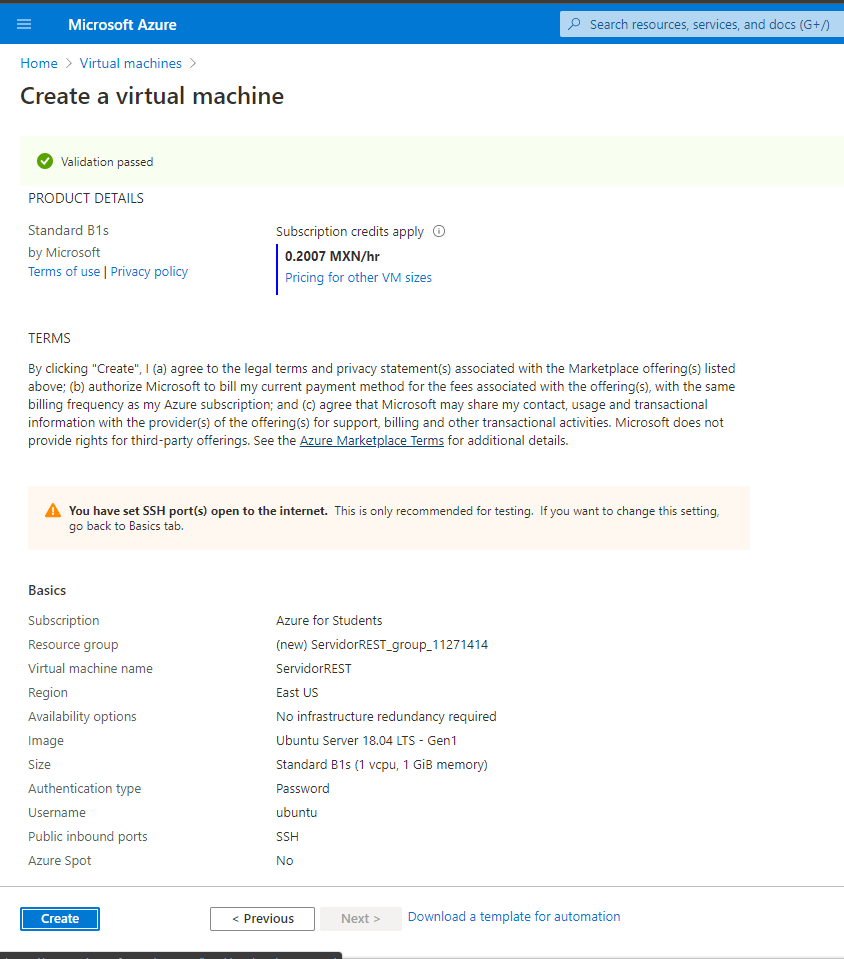
****

Imagen : Creación de la máquina virtual

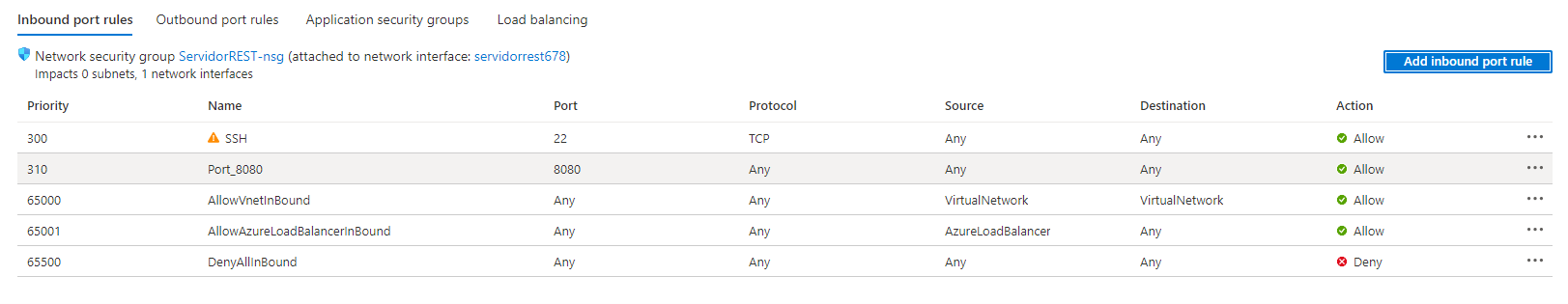


Imagen 1: Puerto 8080 habilitado en la maquina virtual.

2. Instalar JDK8 ejecutando los siguientes comandos en la máquina virtual:

**Se realiza actualización de paquetes e instalación de Java 8.**

sudo apt update

sudo apt install openjdk-8-jdk-headless

3. Descargar la distribución binaria de Tomcat 8 de la siguiente URL (descargar la opción Core "zip"):<https://tomcat.apache.org/download-80.cgi>

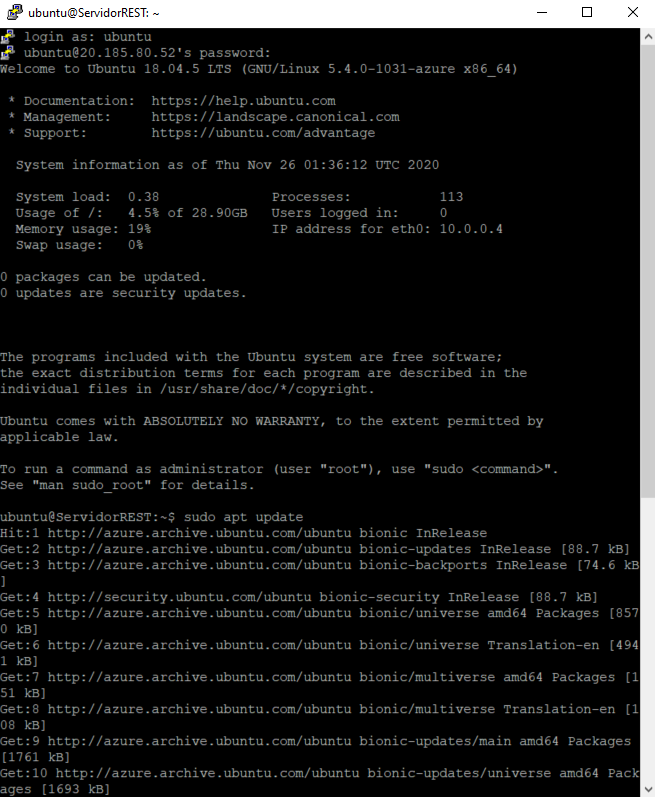


Imagen : Ejecución del comando “sudo apt update” y “sudo apt install openjdk-8-jdk-headless”

4. Copiar a la máquina virtual el archivo ZIP descargado anteriormente y desempacarlo utilizando el comando unzip.

**Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.**

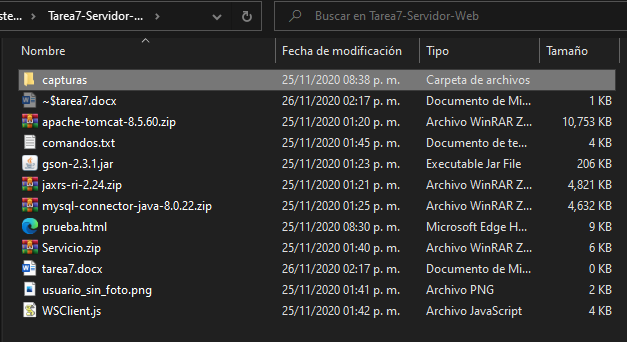
****

Imagen 3: Directorio con los archivos descargados para la práctica

**Se utilizó el programa portable “WinSCP” para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.**

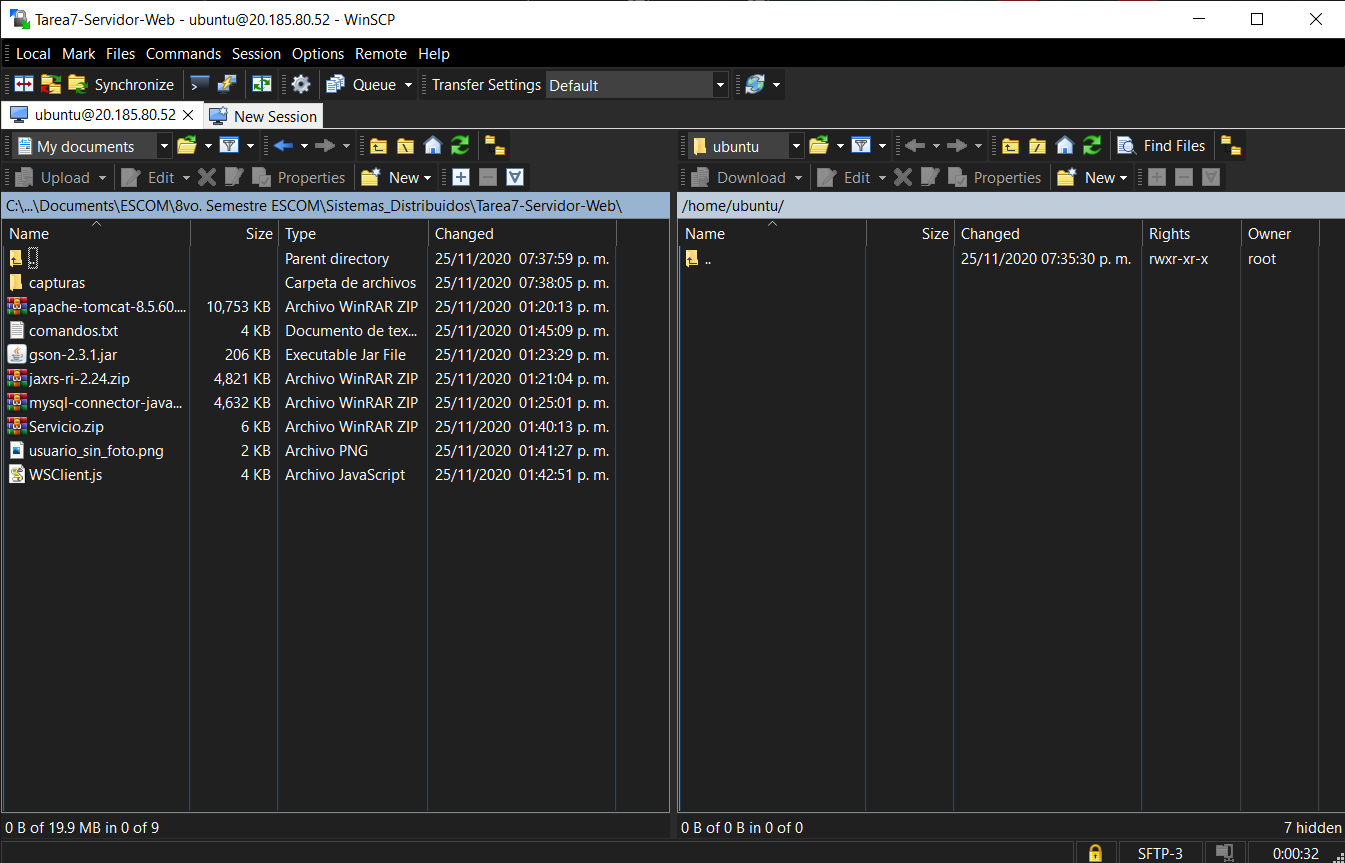


Imagen : Interfaz del programa WINSCP con la conexión realizada a la máquina virtual remota.

**Se tuvo que instalar el programa “Unzip” para poder realizar el desempaquetado de los archivos comprimidos en zip.**

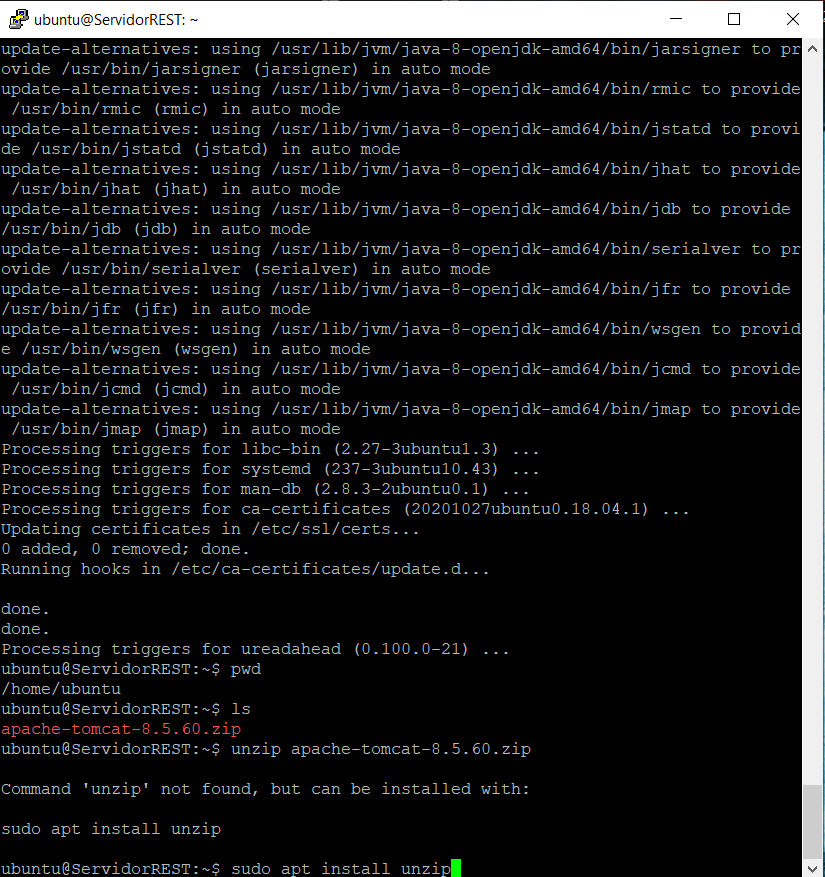
****

Imagen : Instalación del programa “Unzip”

5.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.

NOTA DE SEGURIDAD: Lo anterior se recomienda debido a que se han detectado vulnerabilidades en algunas aplicaciones que vienen con Tomcat, estas aplicaciones se encuentran originalmente instaladas en los directorios webapps y webapps/ROOT.

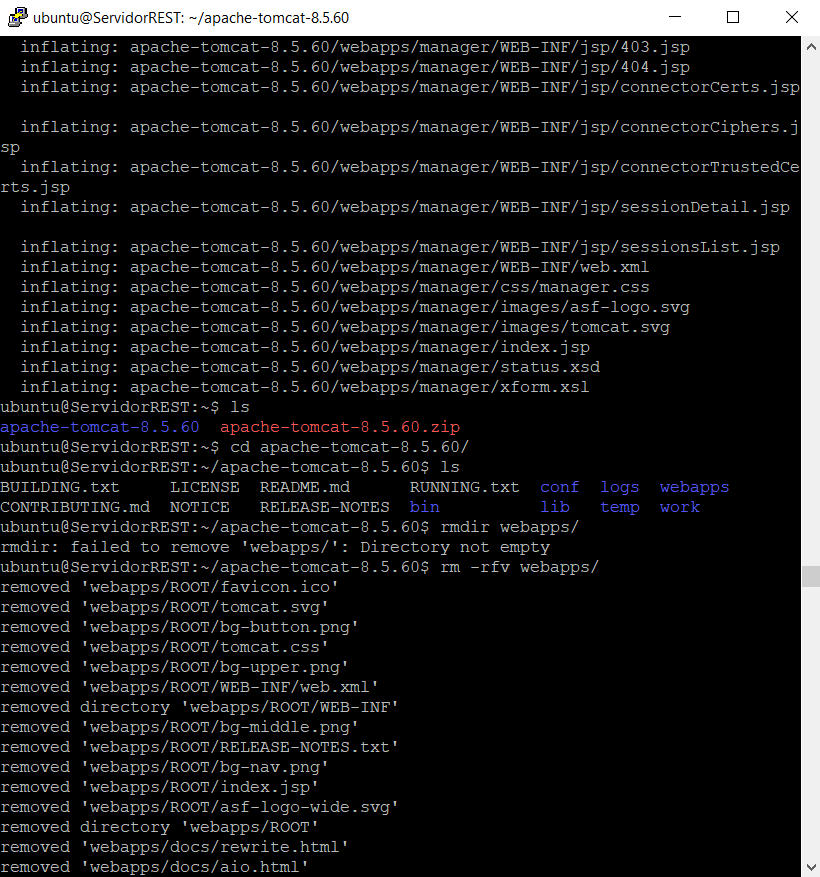


Imagen : Eliminación del directorio “webapps”.

6. 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:

[**https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.24/jaxrs-ri-2.24.zip**](https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.24/jaxrs-ri-2.24.zip)

**Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.**

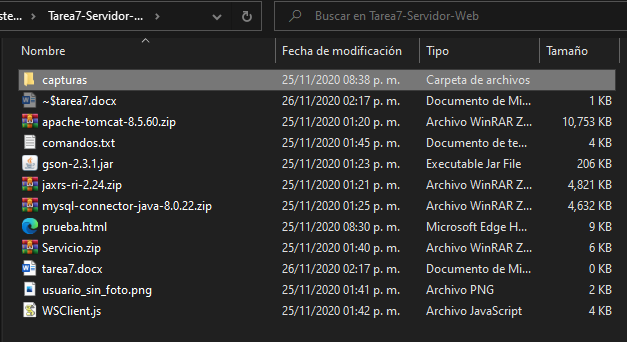
****

Imagen 7: Directorio con los archivos descargados para la práctica

7. 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.

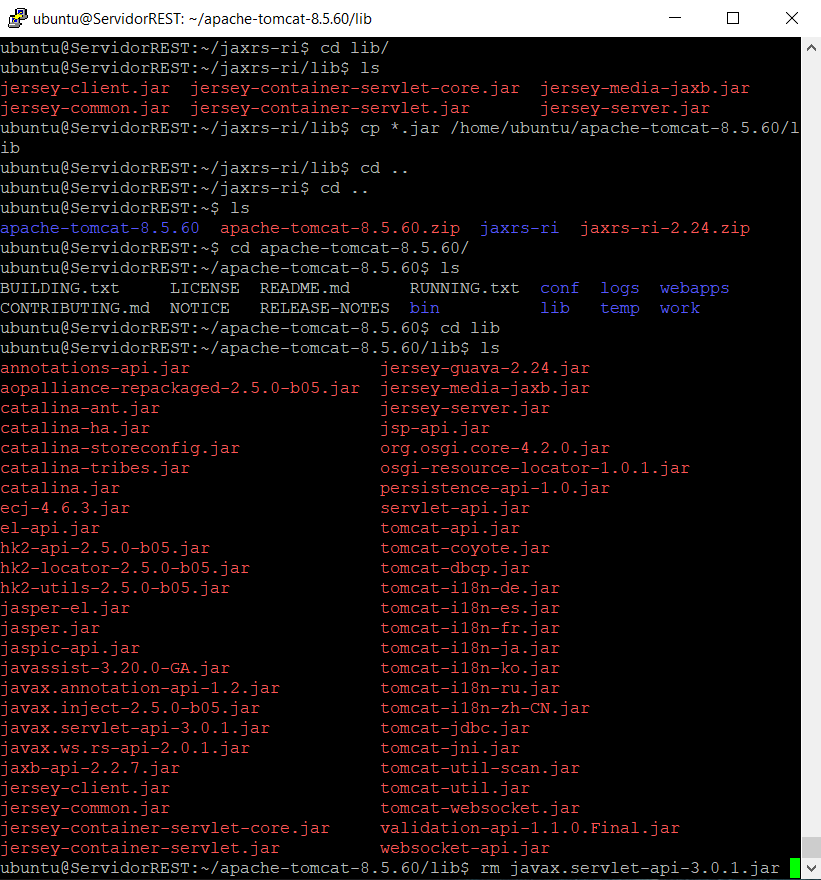


Imagen : copia de todos los archivos .jar a la carpeta ~/apache-tomcat-8.5.60/lib

8. 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).

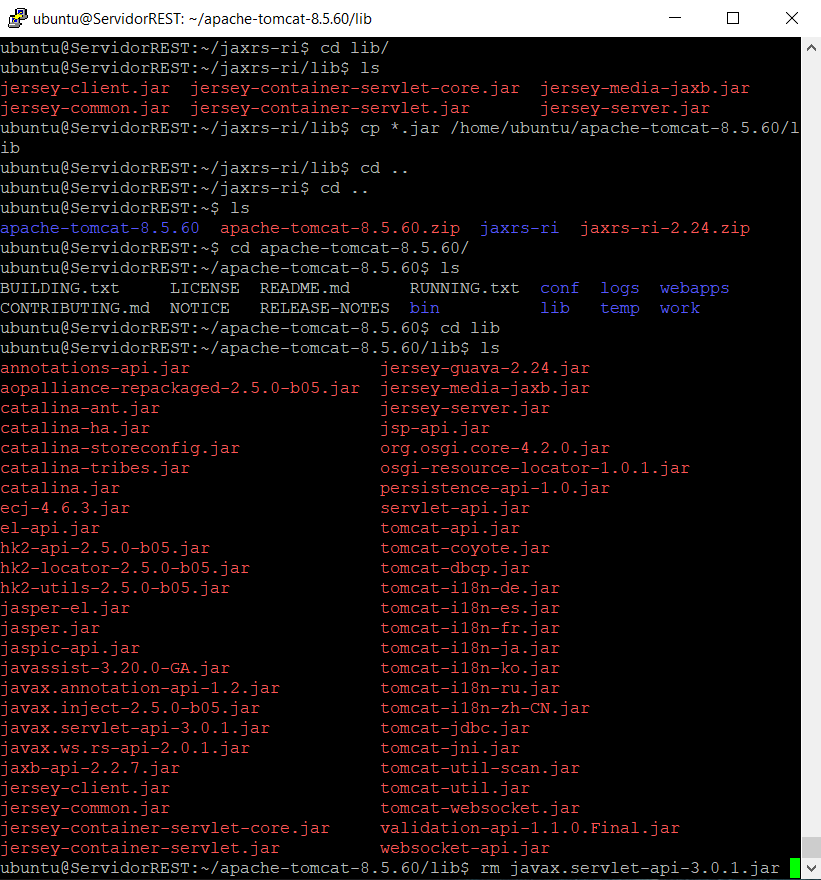


Imagen : eliminación del archivo javax.servlet-api-3.0.1.jar del directorio ~/apache-tomcat-8.5.60/lib

9. Descargar el archivo gson-2.3.1.jar de la URL:

[**https://repo1.maven.org/maven2/com/google/code/gson/gson/2.3.1/gson-2.3.1.jar**](https://repo1.maven.org/maven2/com/google/code/gson/gson/2.3.1/gson-2.3.1.jar)

**Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.**

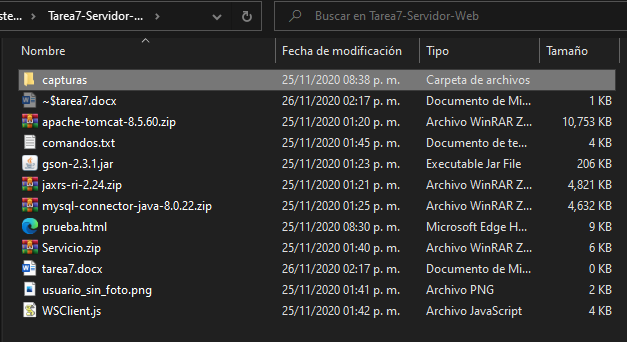
****

Imagen 9: Directorio con los archivos descargados para la práctica

10. Copiar el archivo gson-2.3.1.jar al directorio "lib" de Tomcat.

**Se utilizó el programa portable “WinSCP” para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.**

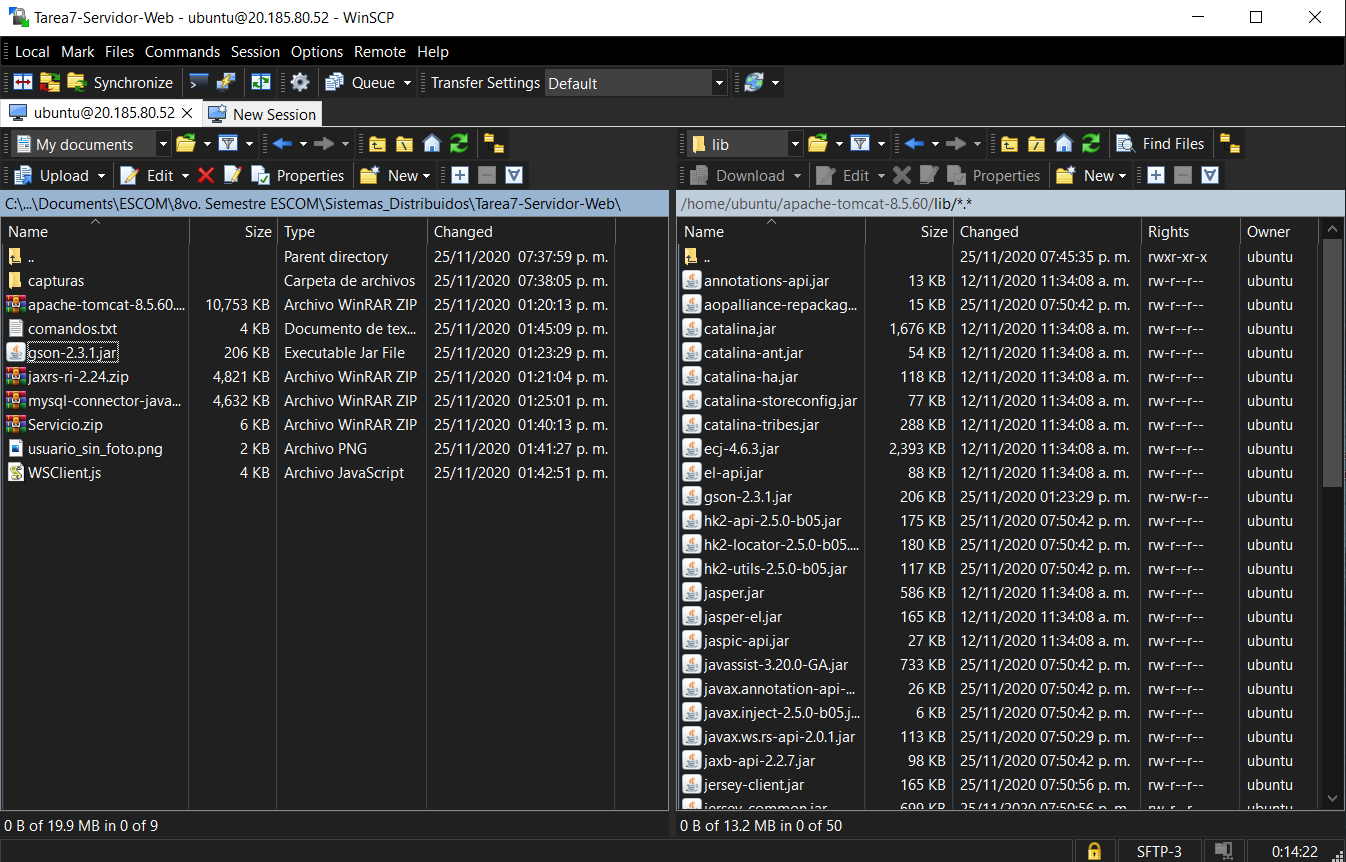


Imagen : copia del archivo “gson-2.3.1.jar” al directorio “lib” de Tomcat

11. Ahora vamos a instalar el driver de JDBC para MySQL. Ingresar a la siguiente URL:

[**https://dev.mysql.com/downloads/connector/j/**](https://dev.mysql.com/downloads/connector/j/)

Seleccionar “Platform independent" y descargar el archivo ZIP.

**Todos los archivos necesarios para la práctica se descargaron y se alojaron en una sola carpeta.**

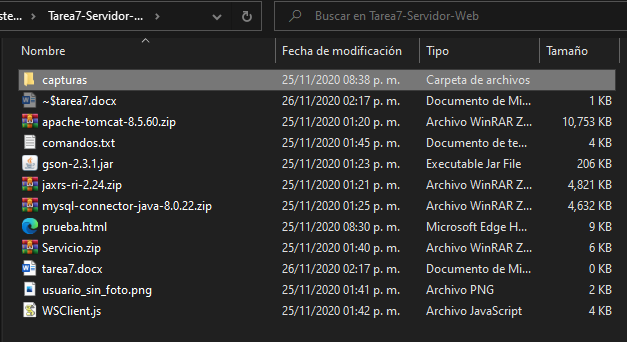
****

Imagen 10: Directorio con los archivos descargados para la práctica

12. Copiar el archivo descargado a la máquina virtual, desempacarlo y copiar el archivo mysql-connector...jar al directorio "lib" de Tomcat.

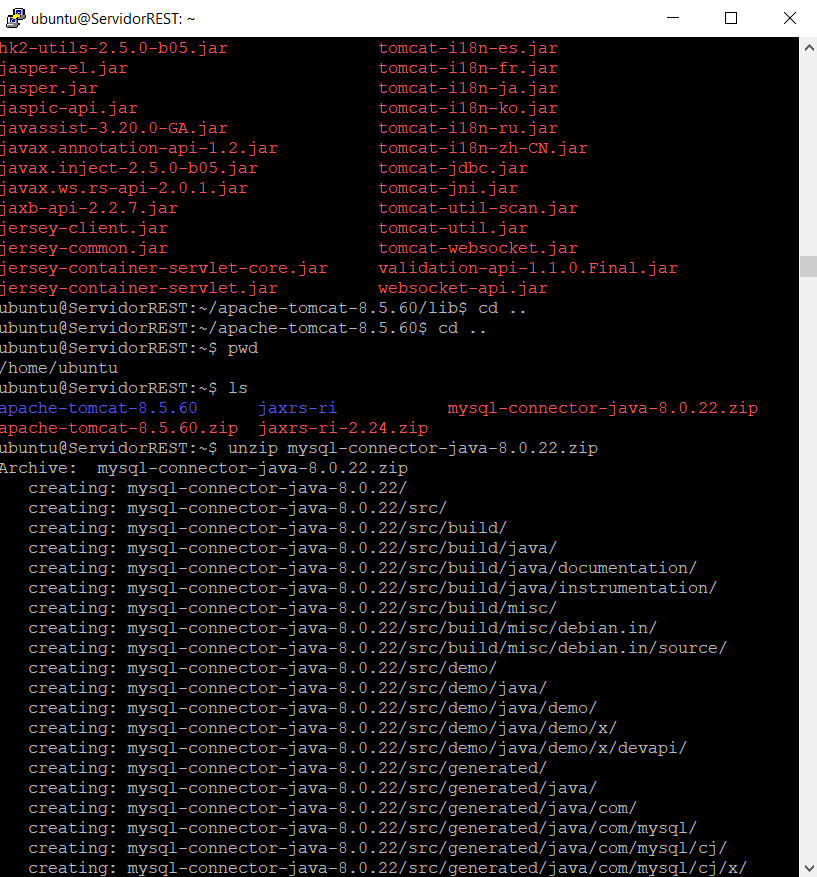


Imagen : Desempaquetado del archivo “mysql-connector-java-8.0.22.zip”

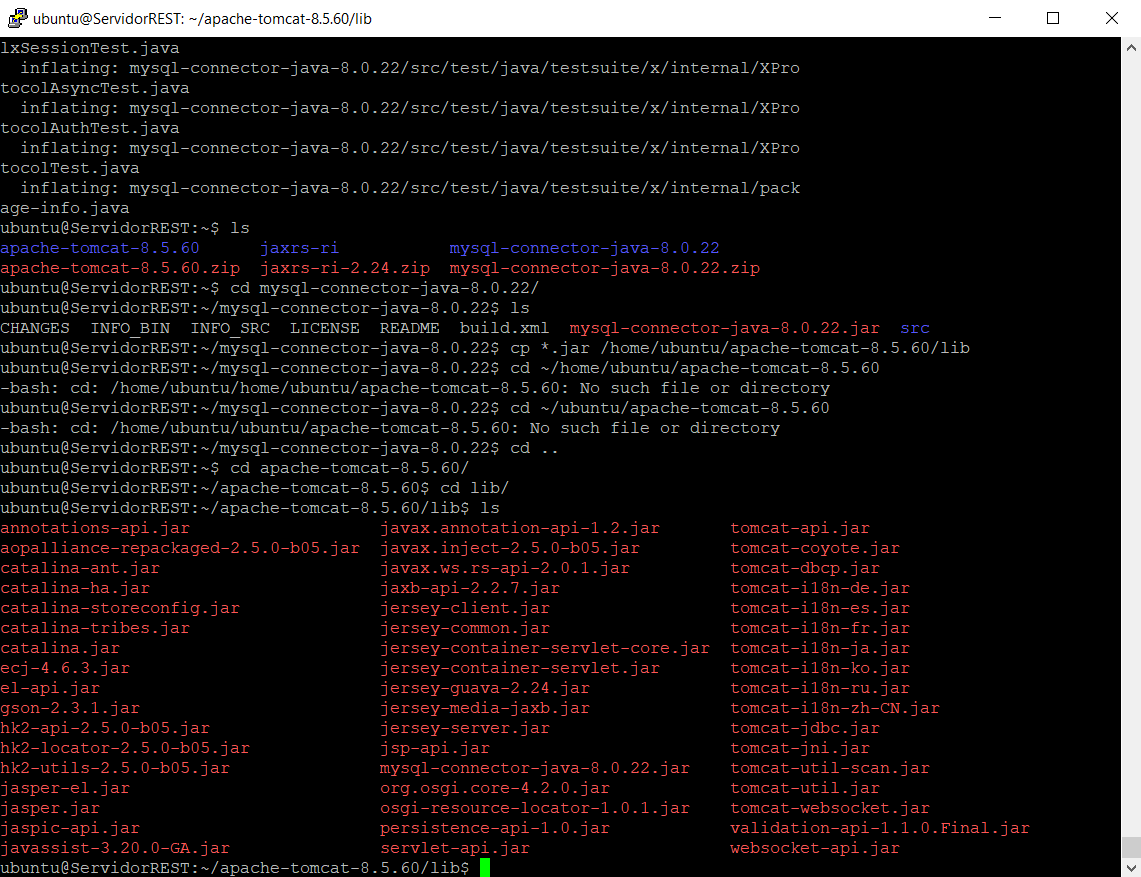


Imagen : copia del archivo “mysql-connector…jar” al directorio “lib” de Tomcat.

**Iniciar/detener el servidor Tomcat**

1. 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

2. Iniciar la ejecución de Tomcat ejecutando el siguiente comando:

sh $CATALINA\_HOME/bin/catalina.sh start

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

sh $CATALINA\_HOME/bin/catalina.sh stop

Notar que Tomcat se ejecuta sin permisos de administrador (no se usa "sudo"), lo cual es muy importante para prevenir que algún atacante pueda entrar a nuestro sistema con permisos de super-usuario.

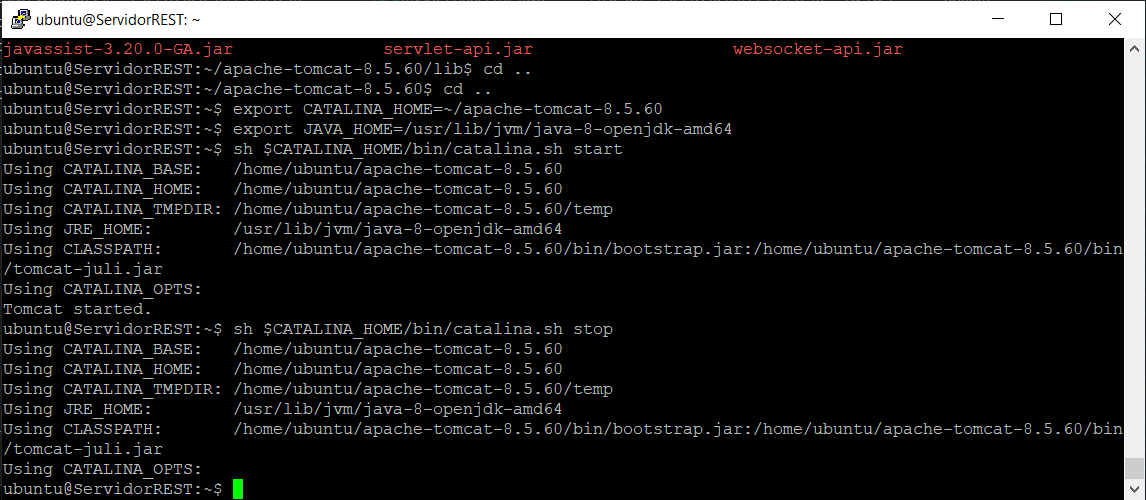


Imagen : Inicialización de variables de entorno y ejecución de los comandos “start” y “stop”.

**Instalación de MySQL**

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

sudo apt update

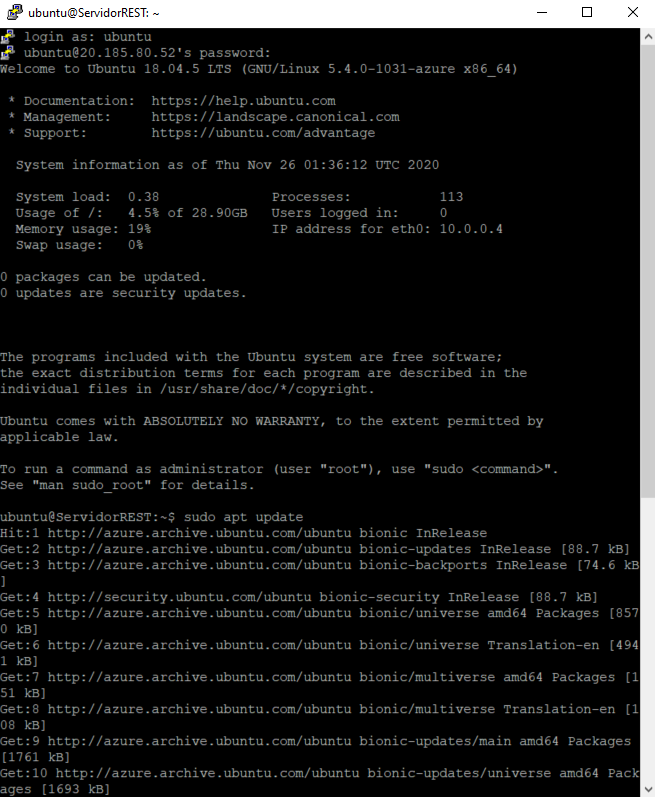


Imagen 113: Ejecución del comando “sudo apt update” y “sudo apt install openjdk-8-jdk-headless”

2. Instalar el paquete default de MySQL:

sudo apt install mysql-server

3. 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**

**Se utilizó como contraseña “root” tanto para el root como para el usuario “hugo” que posteriormente se crea.**

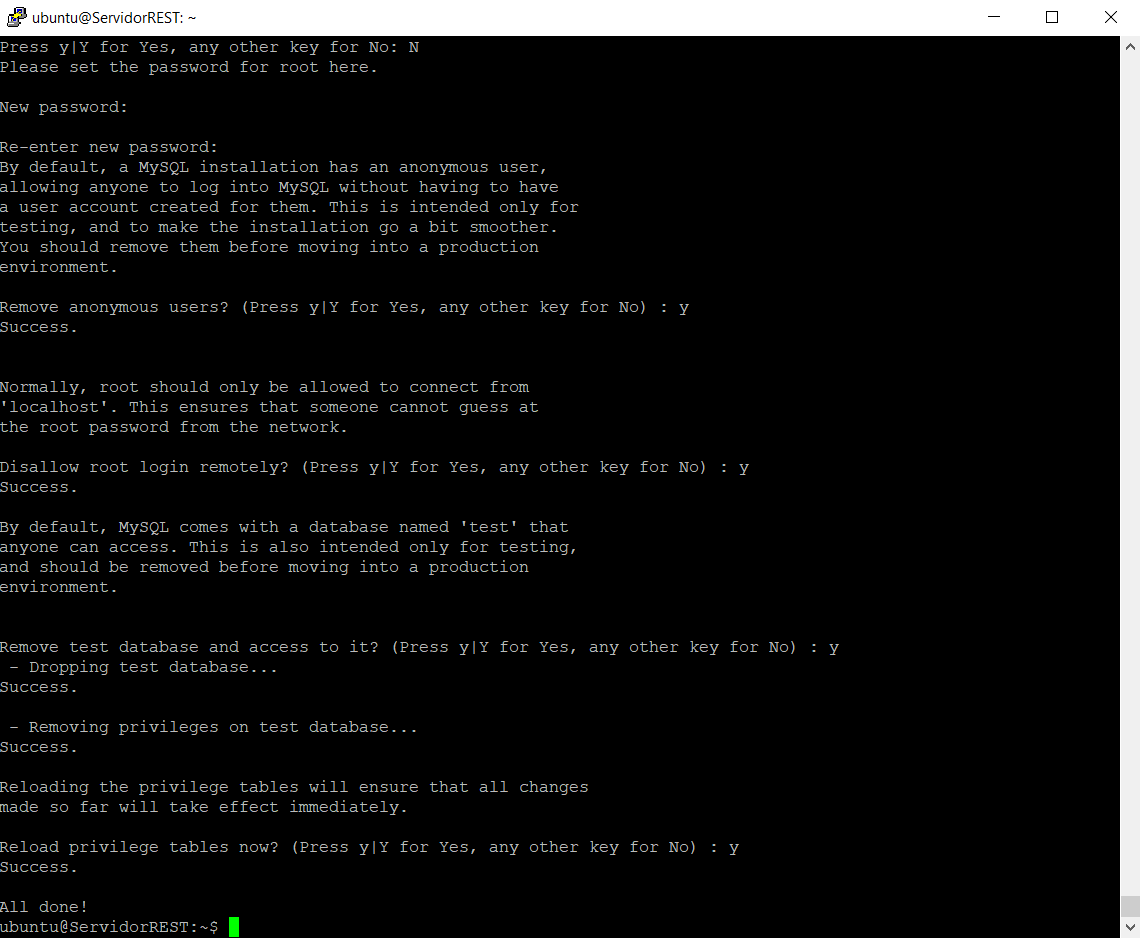


Imagen : ejecución del comando “sudo mysql secure installation”

4. Ejecutar el monitor de MySQL:

sudo mysql

5. 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*';

6. Actualizar los privilegios:

FLUSH PRIVILEGES;

7. Ejecutar el siguiente comando para salir del monitor de MySQL:

quit

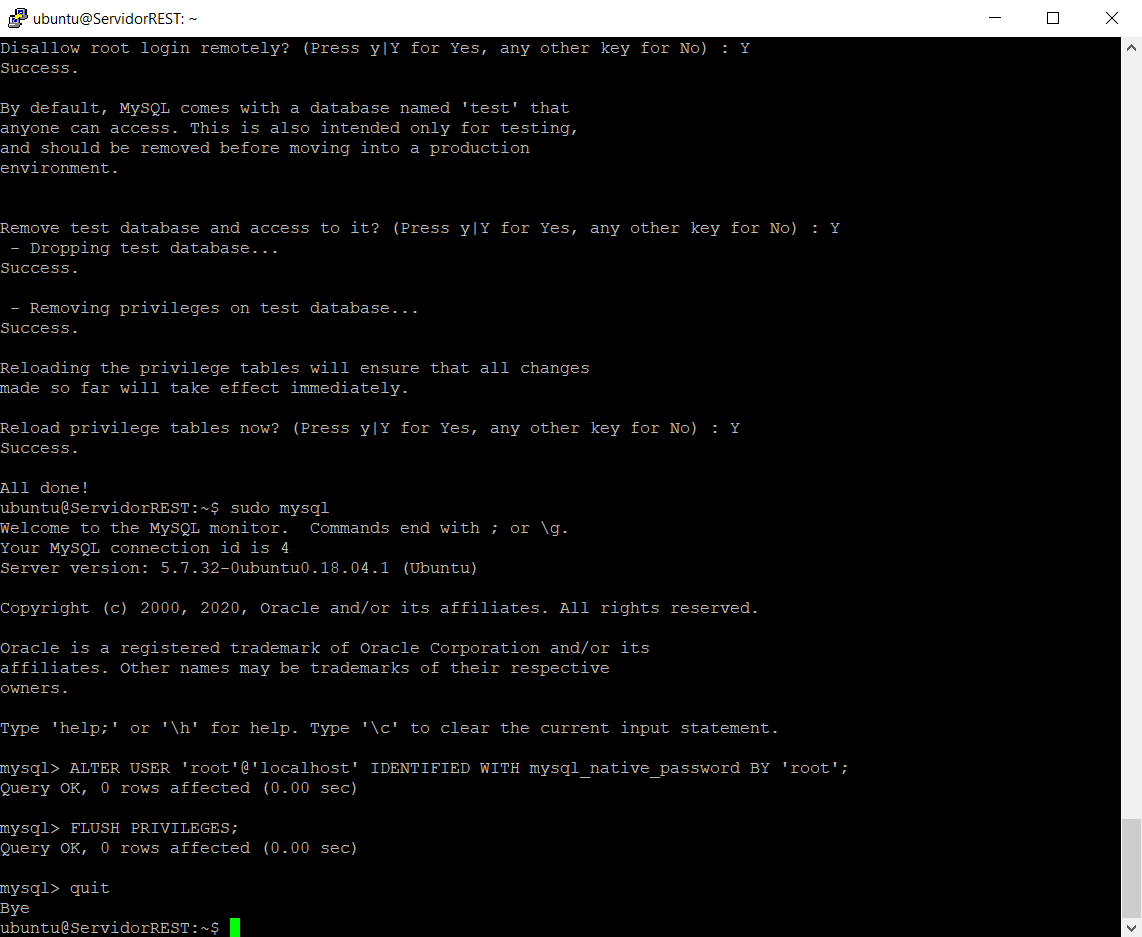


Imagen : modificación de la contraseña del usuario root

**Crear un usuario en MySQL**

1. Ejecutar el monitor de MySQL:

mysql -u root -p

2. Crea el usuario "hugo":

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

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

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

4. Ejecutar el siguiente comando para salir del monitor de MySQL:

quit

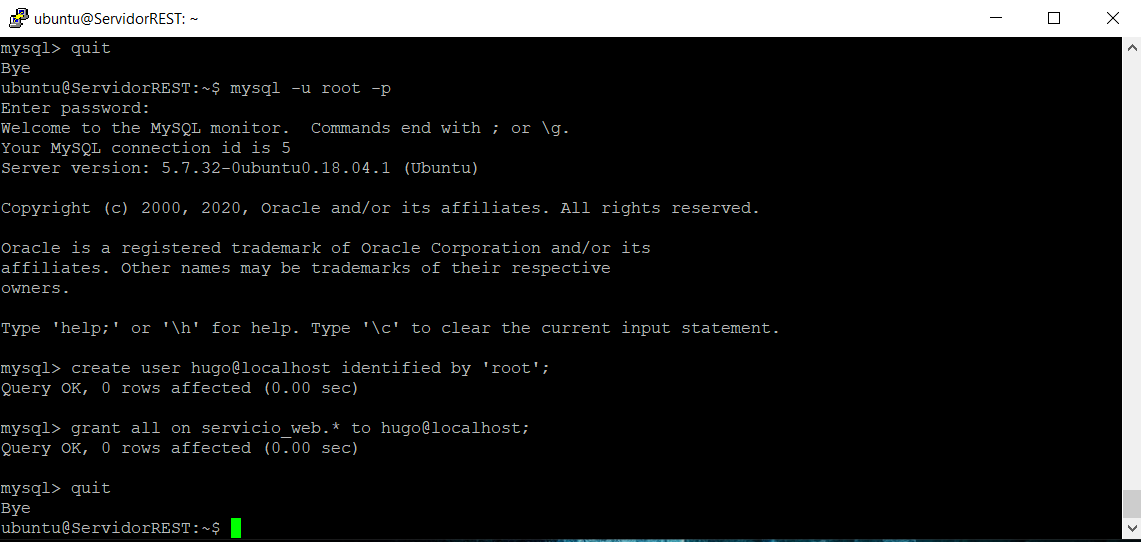


Imagen : Creación del usuario “hugo”.

**Crear la base de datos**

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

mysql -u hugo -p

2. Crear la base de datos "servicio\_web":

create database servicio\_web;

3. Conectar a la base de datos creada anteriormente:

use servicio\_web;

4. 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  
);  
alter table fotos\_usuarios add foreign key (id\_usuario) references usuarios(id\_usuario);  
create unique index usuarios\_1 on usuarios(email);

5. Salir del monitor de MySQL:

quit

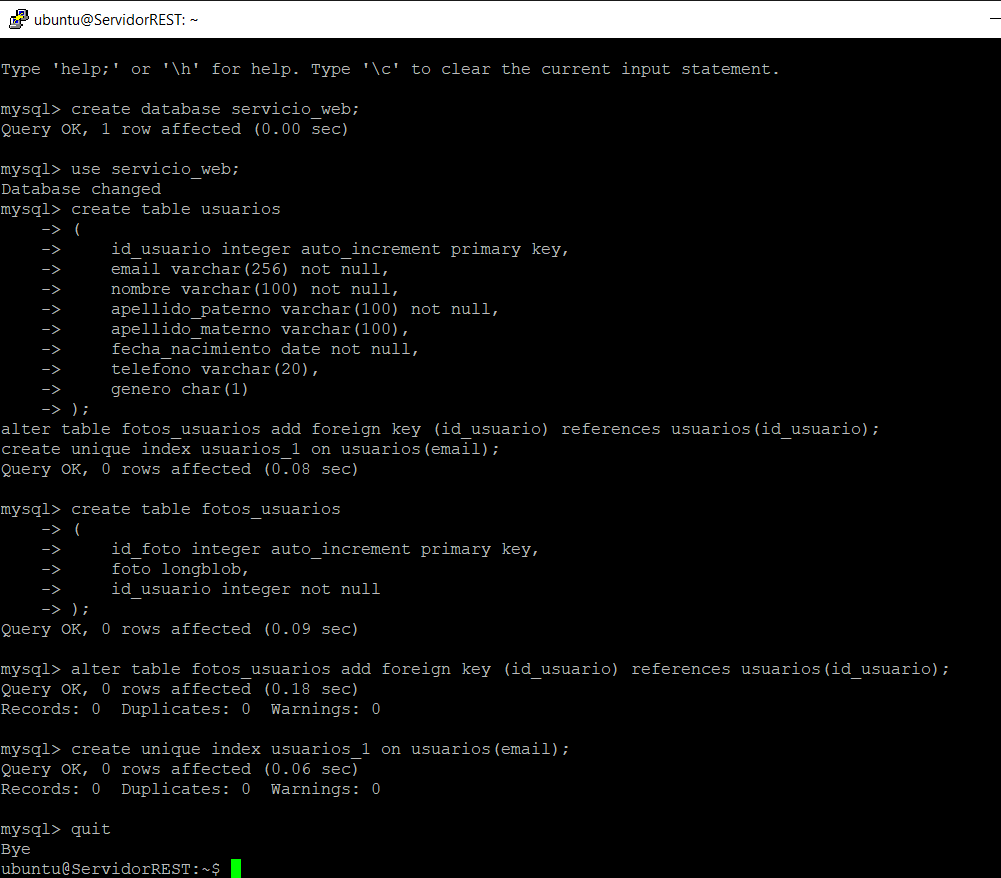


Imagen : Creación de la base de datos y la tabla “usuarios”

**Compilar, empacar y desplegar el servicio web**

1. Descargar de la plataforma y desempacar el archivo [**Servicio.zip**](https://m4gm.com/moodle/mod/resource/view.php?id=1523).

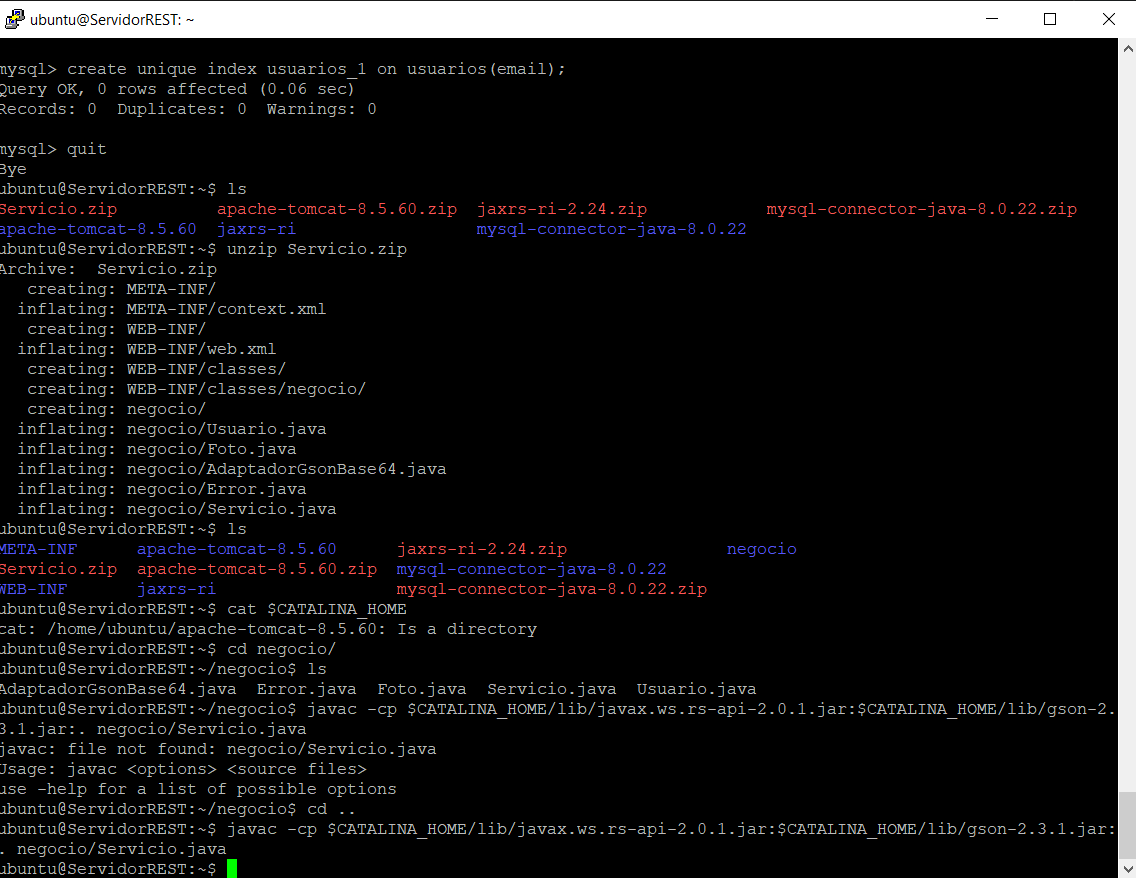


Imagen : Desempaquetado del archivo “Servicio.zip”

2. Definir la variable de ambiente CATALINA\_HOME:

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

**La variable CATALINA\_HOME ya se había definido anteriormente, por lo que se confirmó su ruta realizando el comando “cat $CATALINA\_HOME”**

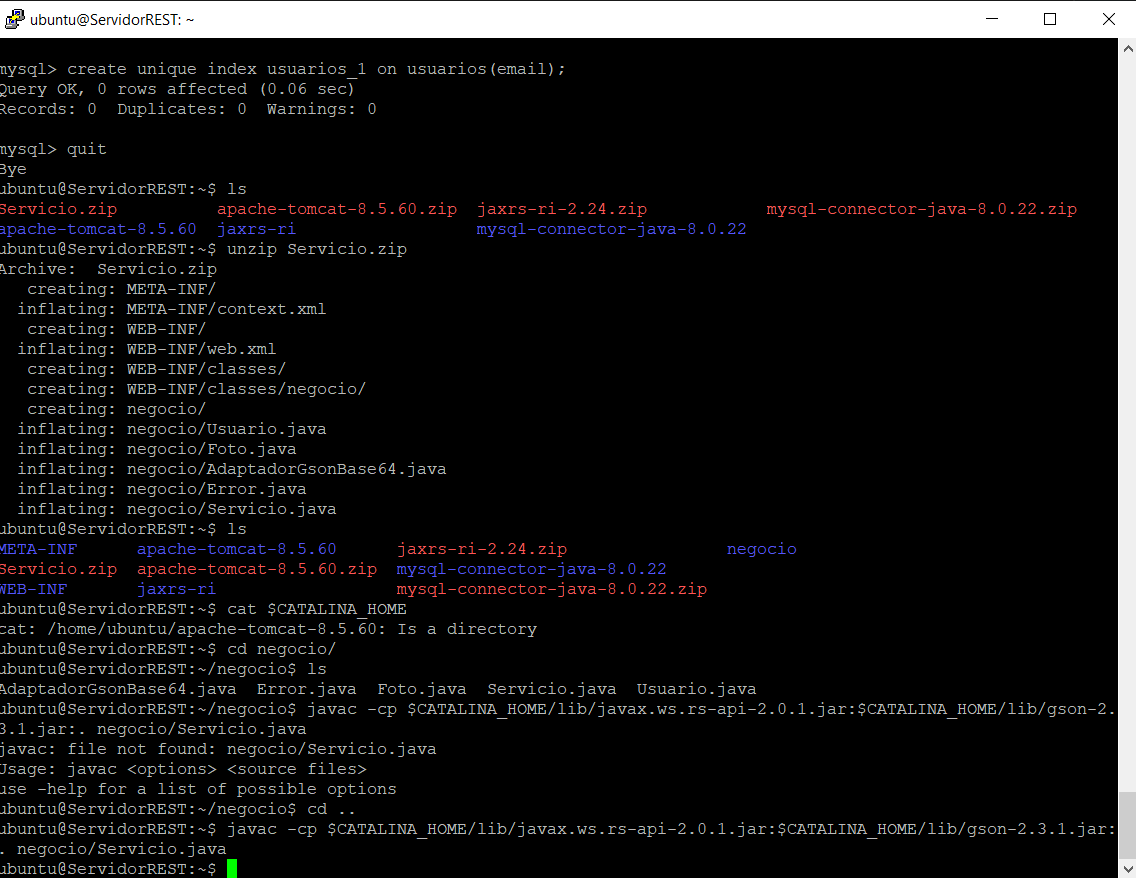
****

Imagen : Se muestra la ruta guardada en la variable CATALINA\_HOME

3. Cambiar al directorio dónde se desempacó el archivo [**Servicio.zip**](https://m4gm.com/moodle/mod/resource/view.php?id=1523) (en ese directorio se encuentra el directorio "negocio").

4. 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

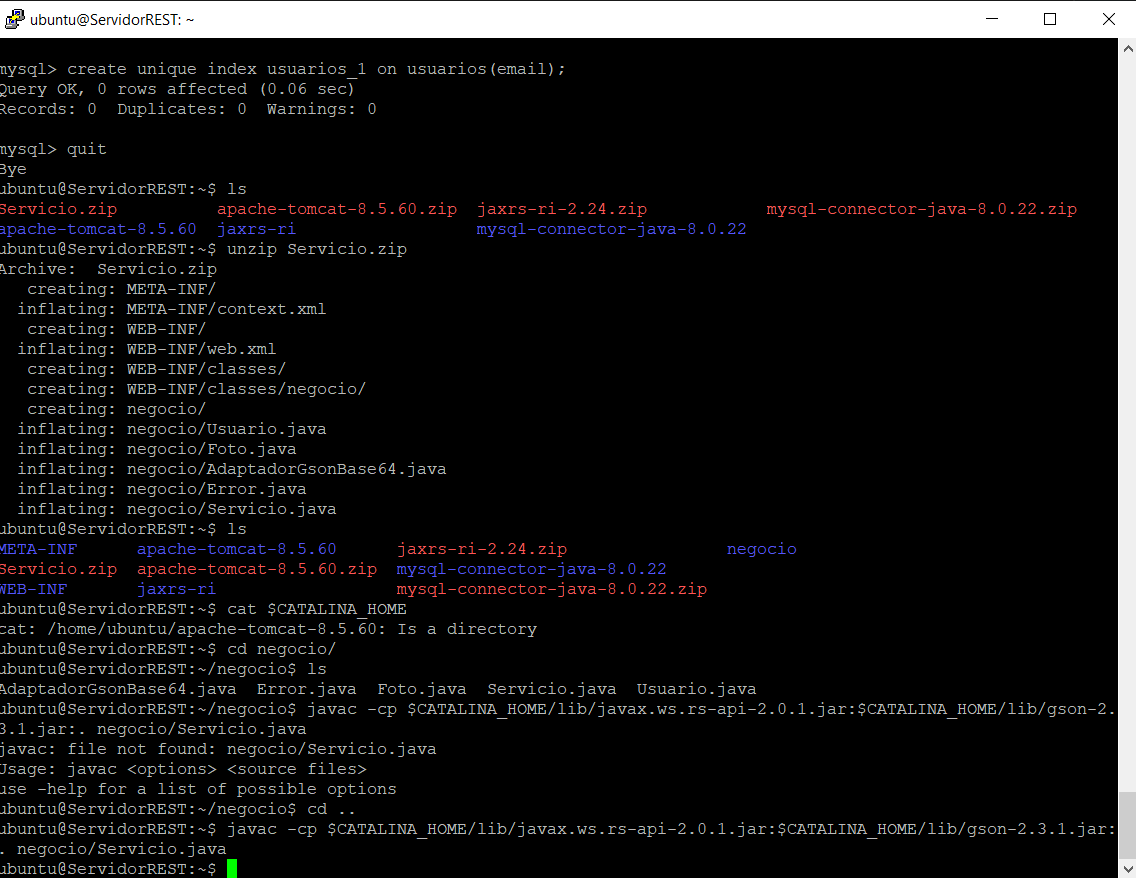


Imagen : Compilación de la clase “Servicio.java” dentro del directorio “negocio”.

5. 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**.

**Se utilizó el editor Nano para poder definir el usuario y el password de la base de datos.**

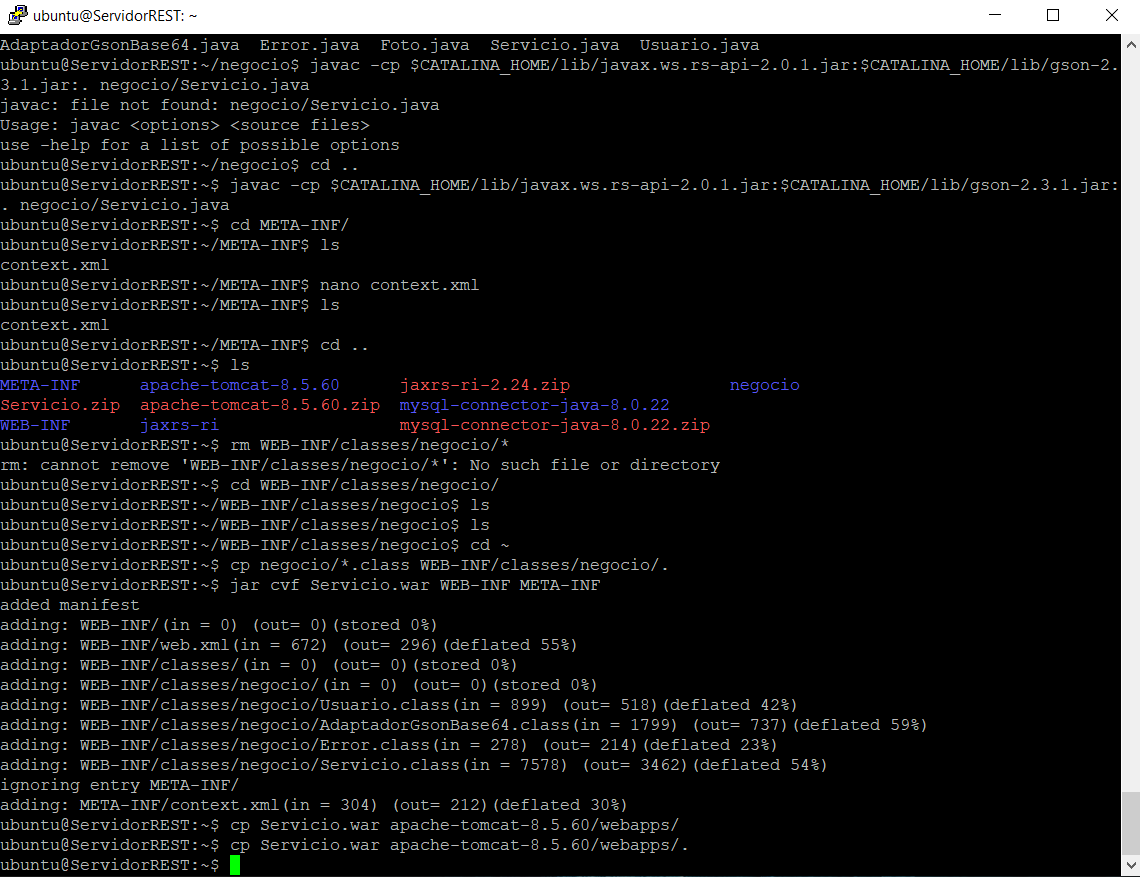


Imagen : Edición del archivo “context.xml” dentro de la carpeta “META-INF”

6. 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

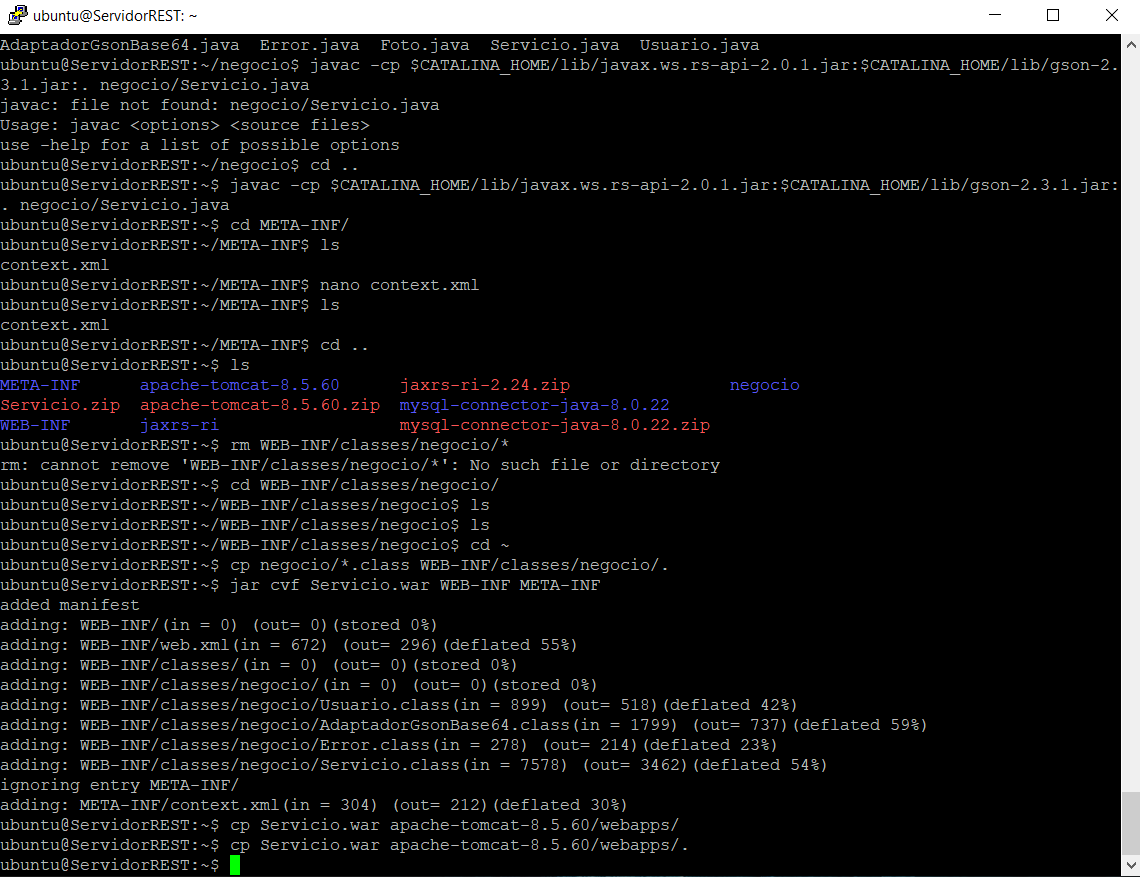


Imagen : Ejecución de los comandos para la creación del servicio web.

7. 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.

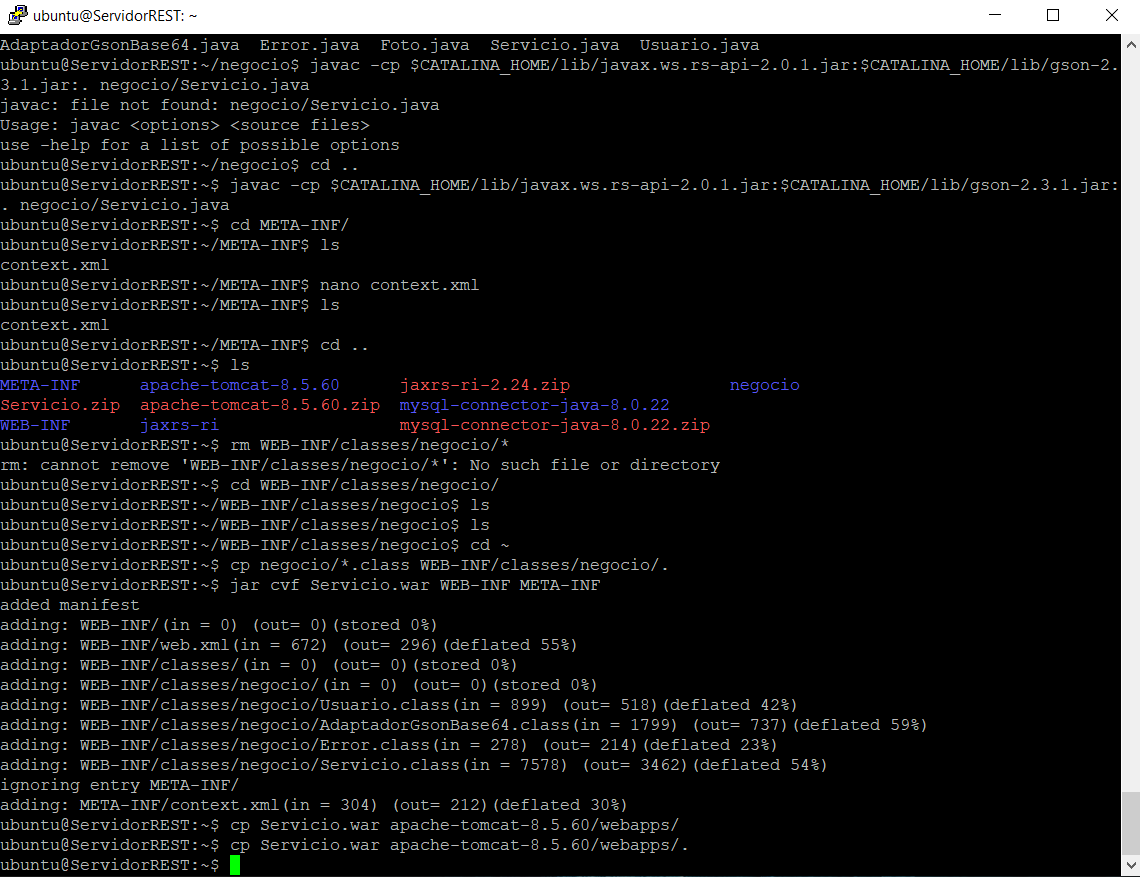


Imagen : Copia del archivo “Servicio.war” al directorio “webapps” de Tomcat

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**

1. Copiar el archivo [**usuario\_sin\_foto.png**](https://m4gm.com/moodle/mod/resource/view.php?id=1525) 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](https://m4gm.com/moodle/mod/resource/view.php?id=1525)

**Se tuvo que volver a ejecutar el comando “**sh $CATALINA\_HOME/bin/catalina.sh start” **para iniciar el servicio de Tomcat, ya que se había detenido anteriormente.**

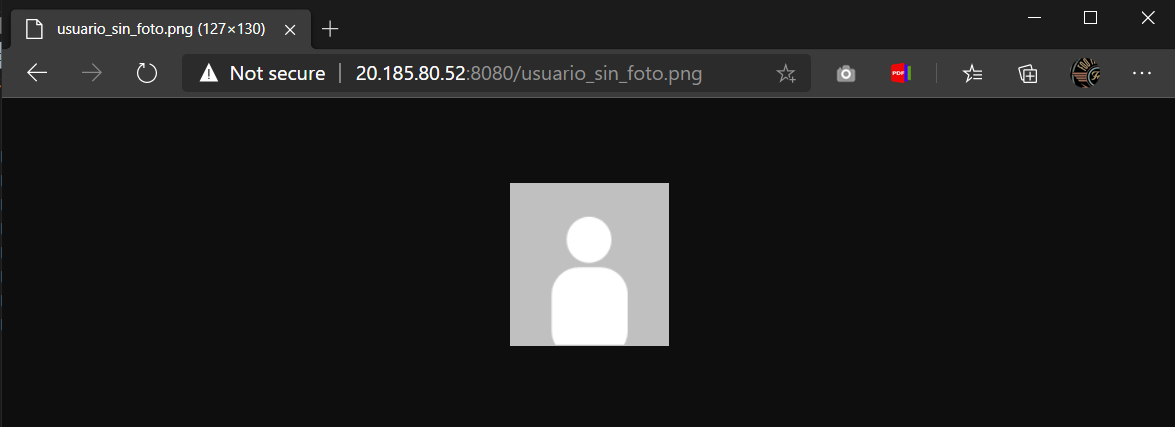


Imagen : Captura del servicio Tomcat en línea, mostrando “usuario\_sin\_foto.png”.

2. Copiar el archivo [**WSClient.js**](https://m4gm.com/moodle/mod/resource/view.php?id=1524) al directorio webapps/ROOT de Tomcat.

**Se utilizó el programa portable “WinSCP” para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.**

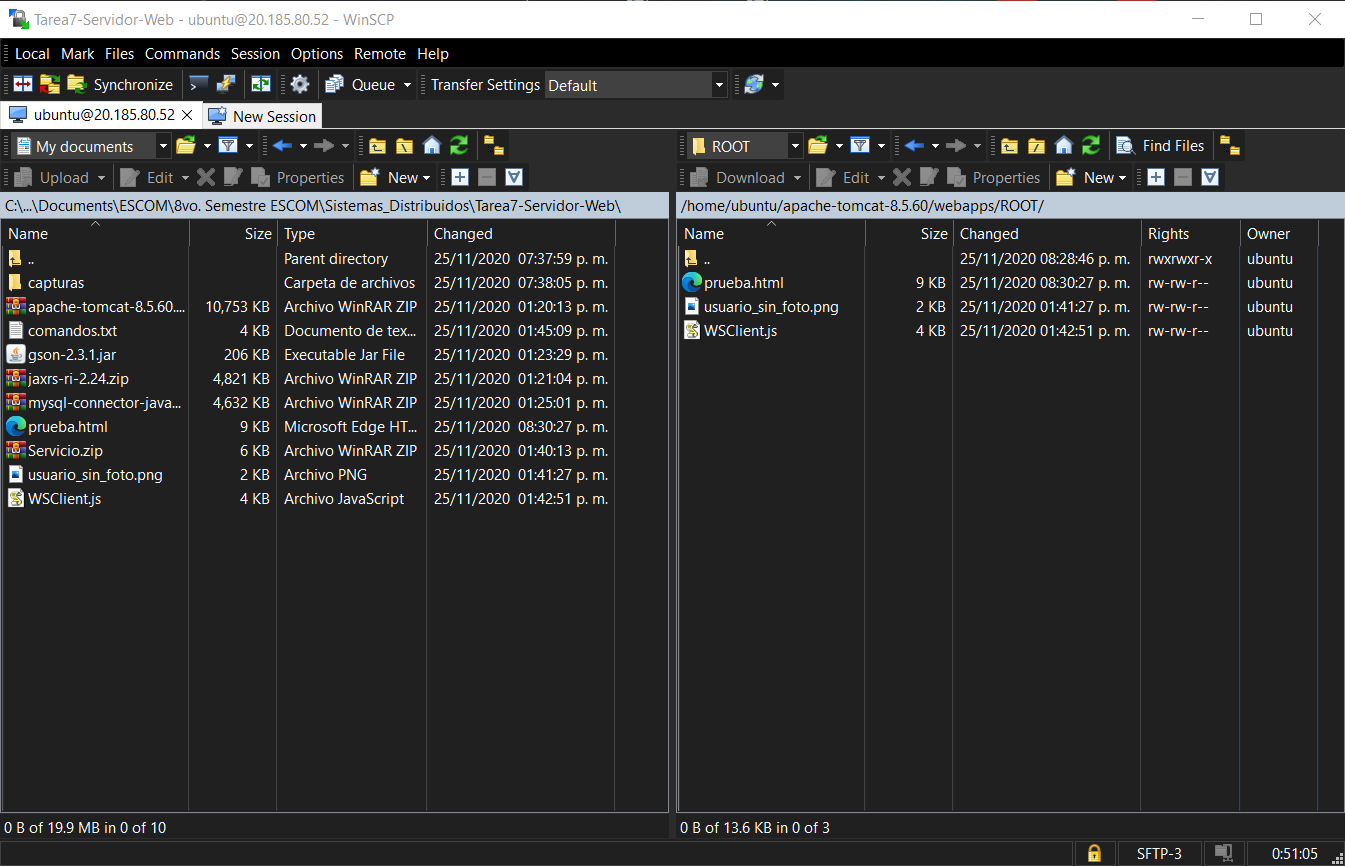


Imagen 24: Transferencia de archivos a la carpeta “webapps/ROOT” de Tomcat de la Maquina Virtual Remota

3. Copiar el archivo [prueba.html](https://m4gm.com/moodle/mod/resource/view.php?id=1526) al directorio webapps/ROOT de Tomcat.

**Se utilizó el programa portable “WinSCP” para la transferencia de archivos desde mi computadora a la máquina virtual remota. Se utilizó el mismo puerto 22 habilitado para las conexiones ssh para la transferencia de archivos.**

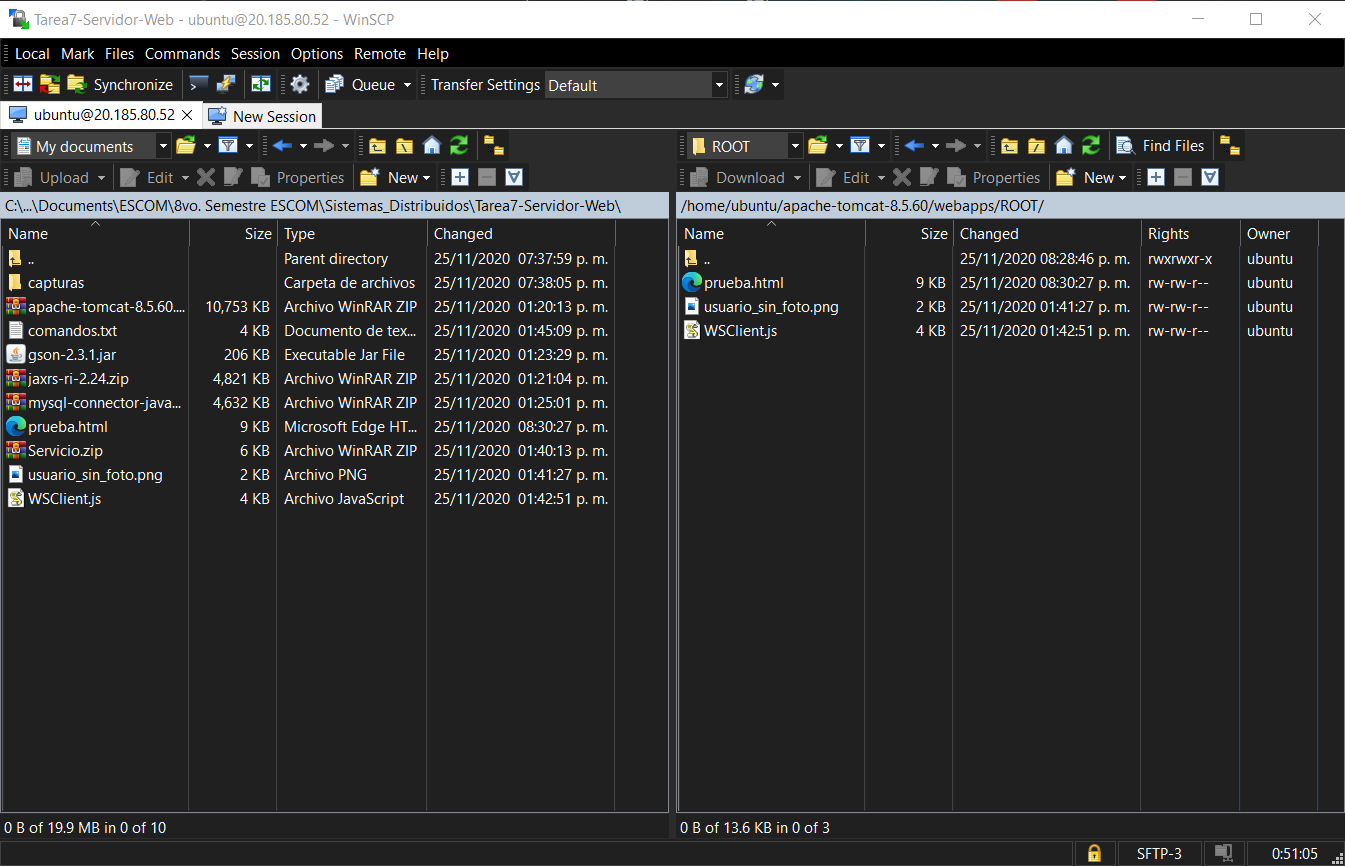


Imagen : Transferencia de archivos a la carpeta “webapps/ROOT” de Tomcat de la Maquina Virtual Remota

4. Ingresar la siguiente URL en un navegador:

http://*ip-de-la-máquina-virtual*:8080/[prueba.html](https://m4gm.com/moodle/mod/resource/view.php?id=1526)



Imagen : ingreso a “http://20.185.80.52:8080/prueba.html”

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

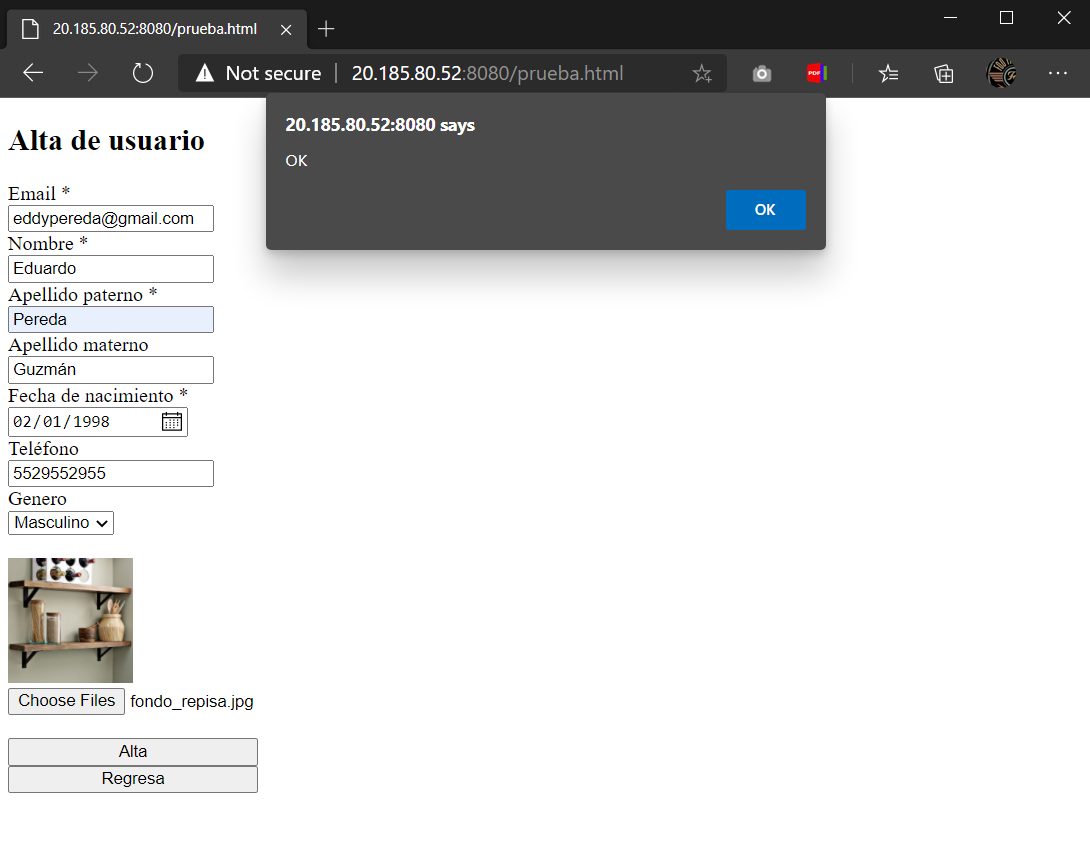


Imagen : Alta de usuario en el formulario.

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

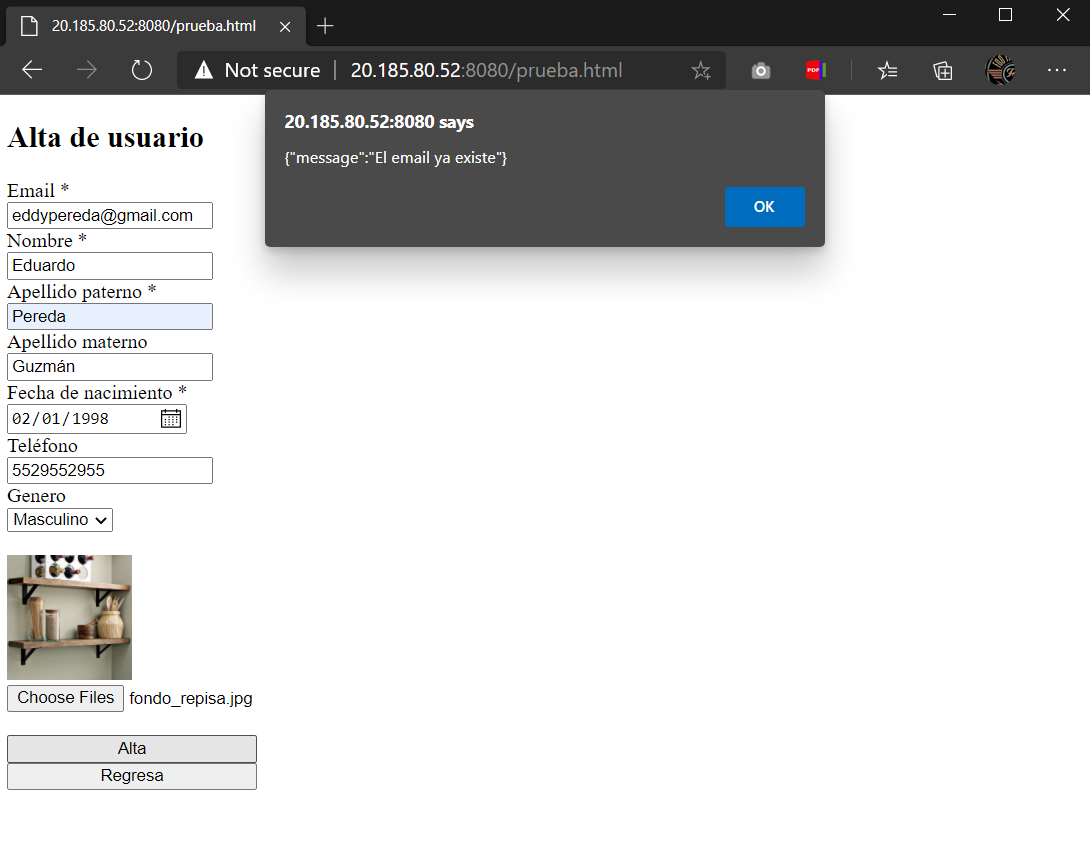


Imagen : Alta de usuario en el formulario con error.

7. Dar clic en el botón “Consulta usuario” para consultar el usuario dado de alta en el paso 5.  Capturar el email y dar clic en el botón “Consulta”,

**Una vez que se presionó el botón “Consulta”, se rellenaron todos los textbox vacíos, mostrando los datos del usuario ingresado.**

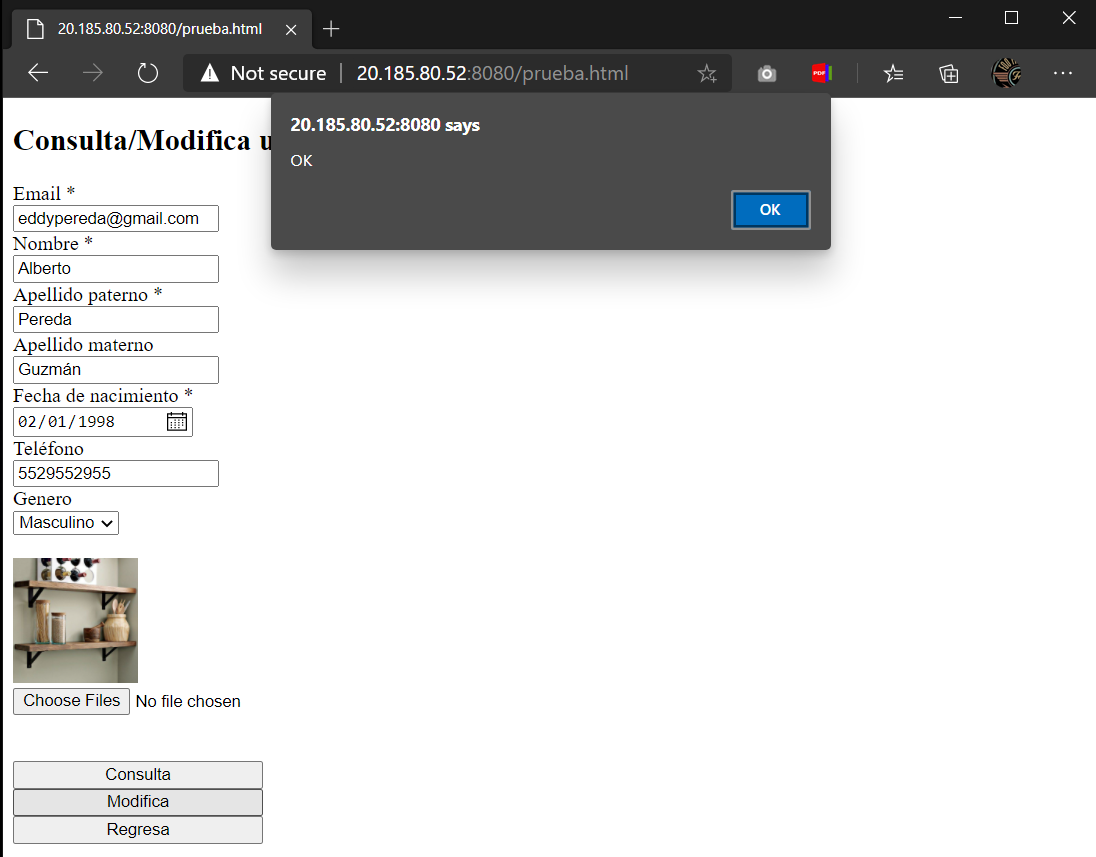


Imagen 28: Consulta y modificación del usuario.

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

**Al modificar el usuario, se desplegó un mensaje confirmando la modificación exitosa.**

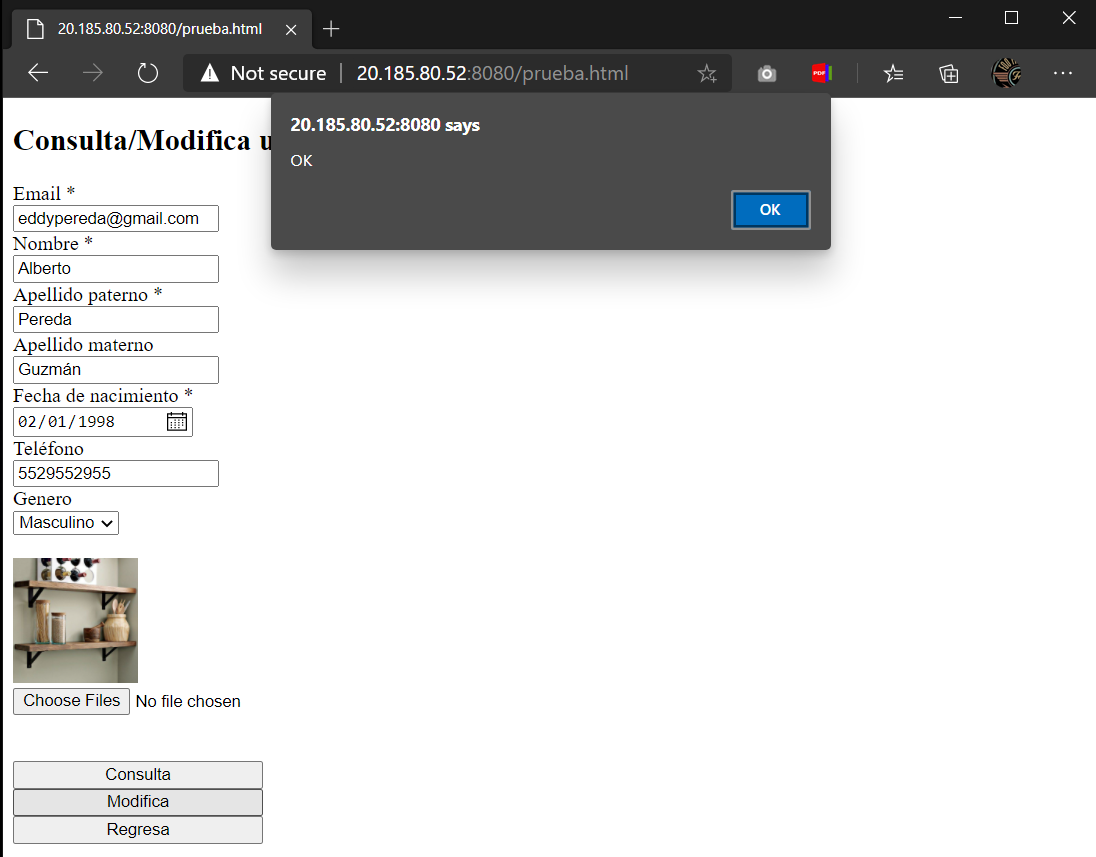


Imagen : Consulta y modificación del usuario.

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

10. 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”.

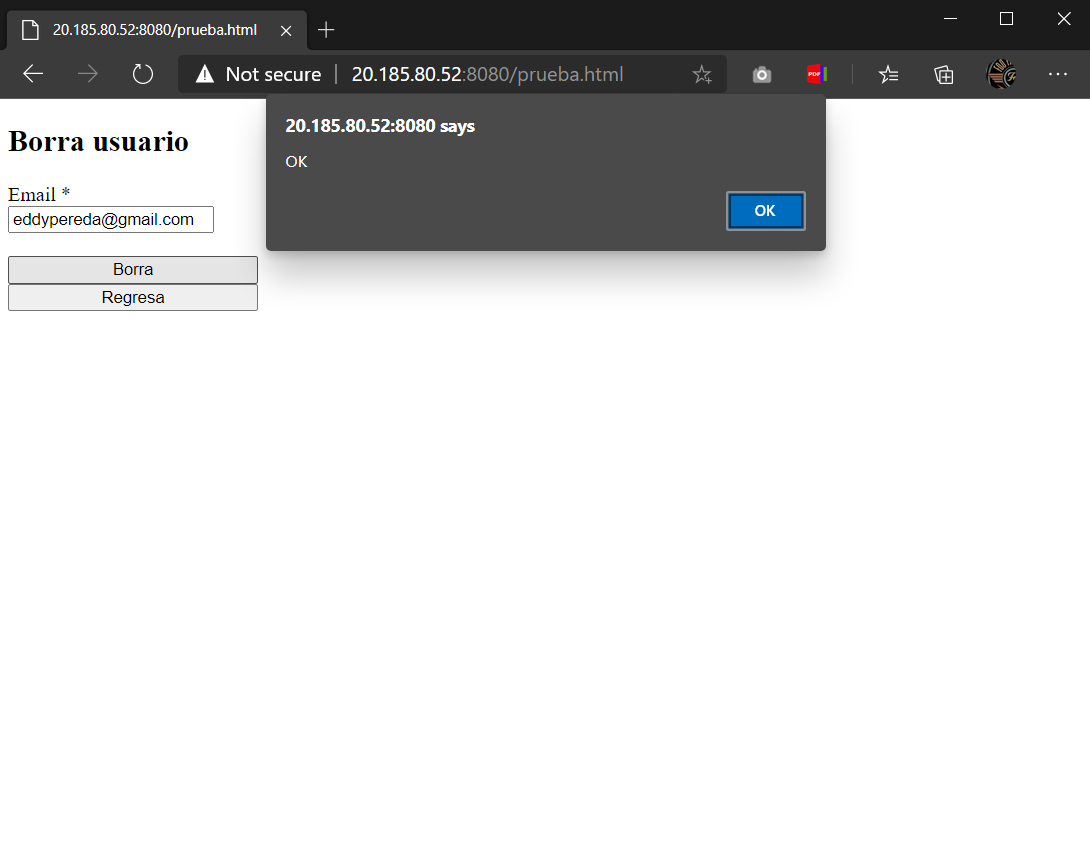


Imagen : Eliminación del usuario.

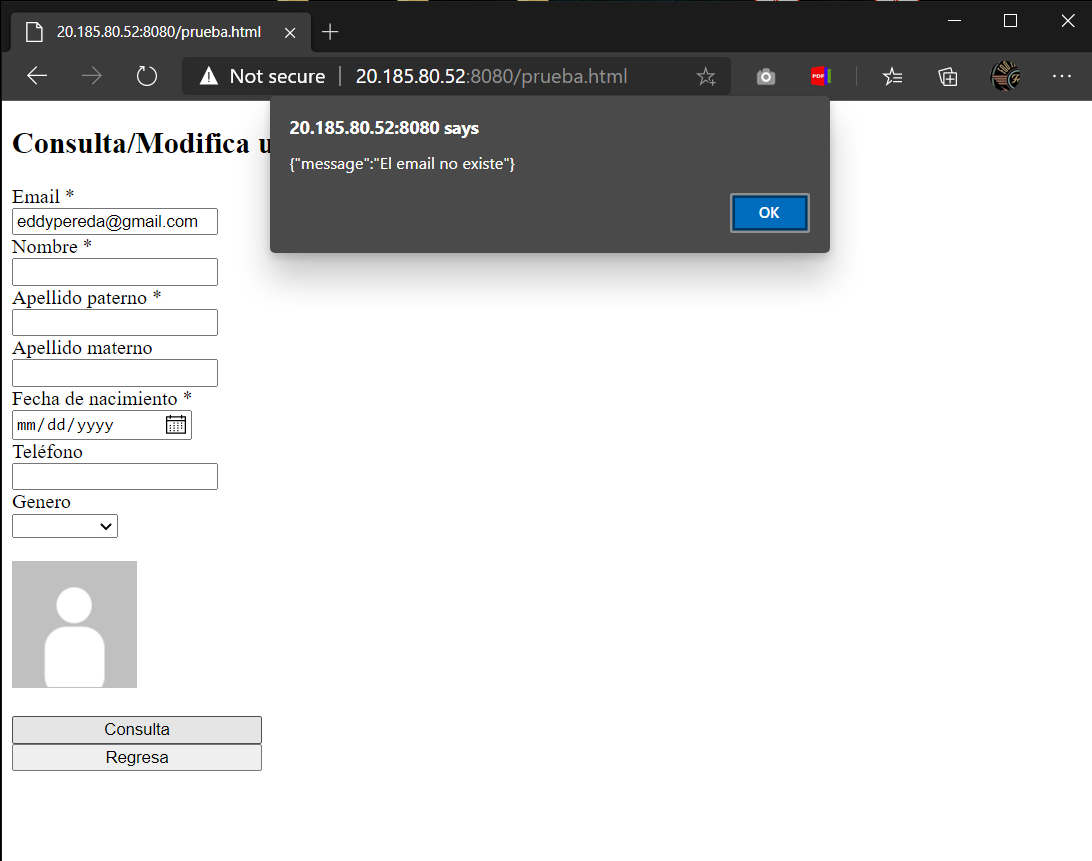


Imagen : Intento fallido de consulta de usuario previamente eliminado.

**Actividades individuales a realizar**

Utilizando un teléfono inteligente y/o una tableta, probar el servicio web accediendo a la siguiente URL en un navegador (Chrome, Firefox, Opera, Safari, etc):

http://*ip-de-la-máquina-virtual*:8080/[prueba.html](https://m4gm.com/moodle/mod/resource/view.php?id=1526)



Imagen : Acceso al servidor por medio de un smartphone.

# Conclusión

El desarrollo de un servicio REST vía HTTP, habilita el uso de los comandos más típicos son GET, POST, PUT/PATCH y DELETE, lo que pone de relieve que su arquitectura solo es apta para la gestión sencilla de datos. Además, el **principio de la ausencia de estado en los recursos de REST** parece limitar mucho las posibilidades de dicha arquitectura. Sin embargo, hace que la REST interface o interfaz de REST haga posible mucho más que la mera inclusión de conjuntos de datos, tal como en servicios web con procesamiento de transacciones, interpretación de transacciones en el marco de una solicitud, servicios web asincrónicos y servicios web con amplia interoperabilidad.