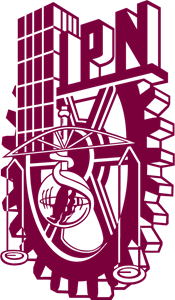
**Instituto Politécnico Nacional.**

**Escuela Superior De Cómputo.**





**Materia:**

**Desarrollo de Sistemas Distribuidos.**

**Tema:**

**Implementación de un servicio web estilo REST.**

**Profesor:**

**Carlos Pineda Guerrero.**

**Alumno:**

**Mario Alberto Miranda Sandoval.**

**Grupo:**

**4CM5.**

**Objetivo.**

Probar el procedimiento de instalación y configuración de Tomcat y MySQL para un servicio REST.

**Desarrollo.**

Primeramente, creamos la máquina virtual que usaremos, posteriormente en agregar regla de puerto de entrada, creamos una regla para habilitar el puerto 8080 en la máquina virtual, las reglas de la máquina virtual se ven en la siguiente imagen.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ahora lo siguiente es conectarnos mediante ssh a la máquina virtual.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Una vez ya en la máquina virtual, procedemos a instalar java, para ello hacemos uso de los siguientes comandos.

**sudo apt update**

**sudo apt-get install openjdk-8-headless**

Una vez terminado el proceso, usamos el comando javac para corroborar la instalación exitosa del jdk.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ahora debemos descargar Tomcat, para ello obtenemos el enlace de descarga del archivo (es el enlace que sale al momento de que se permite descargar Tomcat), y para descargarlo hacemos uso del wget con la siguiente sintaxis.

**wget <enlace archivo>**

Una captura de pantalla de una computadora

Descripción generada automáticamente

Usamos el comando unzip para extraer los archivos del Tomcat con la siguiente sintaxis.

**unzip <nombre archivo .zip>**

*Si existe el error de que el comando unzip no fue encontrado, se puede instalar fácilmente con* ***sudo apt-get install unzip****.*

Como se puede ver en la imagen, tenemos ya los archivos de Tomcat, posteriormente pasamos a eliminar el directorio webapps, con el siguiente comando:

**rm -r <nombre directorio>**

Una captura de pantalla de una computadora

Descripción generada automáticamente

Inmediatamente, se crea el directorio webapps nuevamente y dentro de este el directorio ROOT, este directorio es el que contendrá todos los archivos a los que el archivo.war podrá acceder.

Los directorios los creamos con: **mkdir <nombre directorio>**

Una captura de pantalla de una computadora

Descripción generada automáticamente

Al igual que como se hizo con Tomcat hacemos lo mismo para descargar jaxrs como se ve en la imagen.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Volvemos a extraer el archivo zip con unzip.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ahora, tenemos que copiar todos los archivos .jar que estén en el directorio jaxrs-ri, el directorio tiene 3 directorios internos en los cuales se encuentran los .jar que deseamos, estos directorios son lib, ext y bin.

Para copiar los archivos usamos el comando:

**cp <ruta archivo a copiar> <ruta destino>**

*Cabe destacar que se debe incluir el nombre del archivo a copiar en la ruta.*

*Para facilitar la tarea se usó el comodín \*, de la siguiente manera \*.jar, a modo de copiar todos los archivos .jar que se encuentran en el directorio especificado.*

Captura de pantalla de computadora

Descripción generada automáticamente

Como se puede ver en la siguiente imagen, todos los archivos han sido copiados a Tomcat.

Texto

Descripción generada automáticamente

 Aunque no se menciona, se debe borrar el archivo javax.servlet-api-3.0.1.jar del directorio lib de Tomcat, esto con el comando:

**rm <nombre archivo>**

Posteriormente descargamos el Gson usando wget.

Una captura de pantalla de una computadora

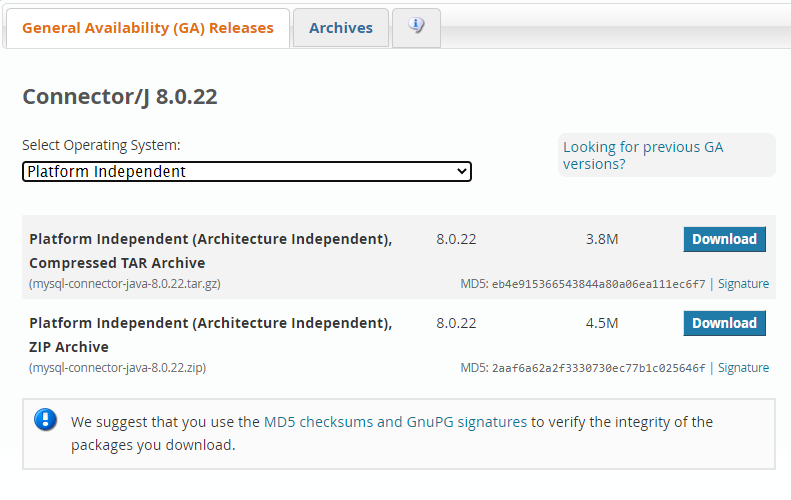
Descripción generada automáticamente

De nueva cuenta procedemos a copiar el gson a la carpeta lib de Tomcat.

Interfaz de usuario gráfica, Texto

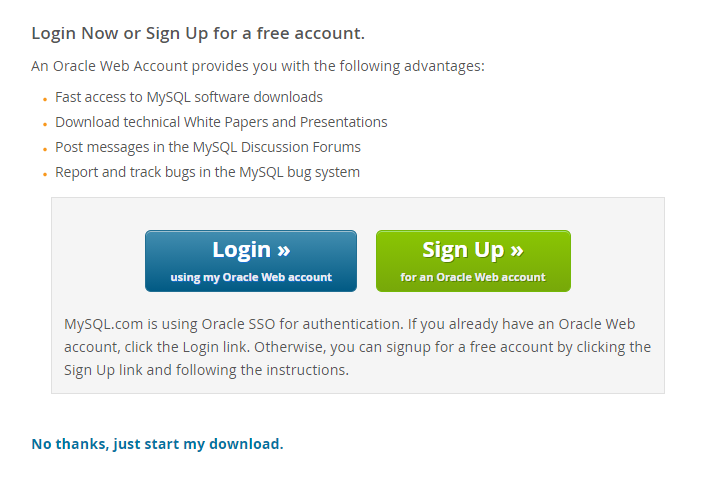
Descripción generada automáticamente

Continuamos con la instalación de MySQL, entrando a su página y buscamos el Connector/J, en la opción de Select Operative System, seleccionamos Platform Independent.



Posteriormente damos click al botón de download para el archivo.zip

Nos llevará a una página donde nos pedirá que nos registremos, le daremos click derecho a donde dice “No thanks, just start my download” y seleccionaremos copiar enlace.



Con el enlace copiado procedemos a descargar los archivos de MySQL usando el wget.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Procedemos a configurar el MySQL, aquí tendremos que configurar la contraseña root.

Texto

Descripción generada automáticamente

Ahora usamos sudo mysql, para poder modificar la contraseña del root.

Texto

Descripción generada automáticamente

Creamos un nuevo usuario **hugo,** con la contraseña **passwordHugo** y le damos permisos a para tener acceso a la base de datos del servicio web.

Texto

Descripción generada automáticamente

Posteriormente usando el usuario Hugo, creamos la base de datos servicio\_web.

Texto

Descripción generada automáticamente

Posteriormente creamos las variables de entorno para CATALINA\_HOME y JAVA\_HOME, usando la siguiente sintaxis.

**export CATALINA\_HOME=ruta de tomcat.**

**export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64**

Ahora usamos el comando **printenv | less,** donde nos muestra la información de las variables de entorno de la máquina.

Texto

Descripción generada automáticamente

Para probar corremos el servidor Tomcat con el comando:

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

Texto

Descripción generada automáticamente

Ahora procedemos a descargar los archivos del servidor, para mayor comodidad se subieron los archivos primeramente a un repositorio de GitHub donde después fueron descargados usando el wget.

Situamos los archivos a la misma altura donde tenemos Tomcat, compilamos los archivos y los dejamos dentro de la carpeta negocio.

Texto

Descripción generada automáticamente

Ahora en el archivo META-INF colocamos la información de nuestro usuario creado para los campos username y password, como se ve en la siguiente imagen.

Captura de pantalla de computadora

Descripción generada automáticamente

Antes de crear el archivo.war eliminamos todos los archivos dentro del directorio WEB-INF/classes/negocio usando el comando:

**rm WEB-INF/classes/negocio/\***

Ahora los archivos que generamos con la compilación usando javac las movemos al directorio donde borramos todos los archivos hace unos momentos.

Creamos el archivo.war con el comando:

**jar cvf Servicio.war WEB-INF META-INF**

Por último, movemos al directorio webapps de Tomcat el Servicio.war y para probarlo movemos los archivos prueba.html, usuario\_sin\_foto.png y WSClient.js al directorio ROOT que creamos al inicio de la configuración.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ahora para entrar al servicio, abrimos un navegador y en la URL ponemos:

**http://<IP pública de la máquina virtual>:<Puerto>/prueba.html**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al entrar observamos que hay 3 botones.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al entrar a la página de alta usuario, nos encontramos con un formulario, el cual al llenarlo y presionar en el botón de alta nos saldrá un mensaje de alerta con el texto “Ok” esto nos indica que nuestro usuario fue creado con éxito.

Si se intenta registrar un usuario con el mismo email se lanza un mensaje de error.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En la página de consulta, tenemos que escribir el email del usuario y darle a consulta, de este modo se nos mostrará la información relacionada a este usuario.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al hacer una modificación se despliega un mensaje de que el usuario ha sido modificado y al recargar la página se muestran los cambios.

Ahora procedemos a borrar un usuario, en la página de borrar colocamos el email del usuario a borrar, una vez ejecutado se nos despliega el mensaje de confirmación.

Interfaz de usuario gráfica, Aplicación, Teams

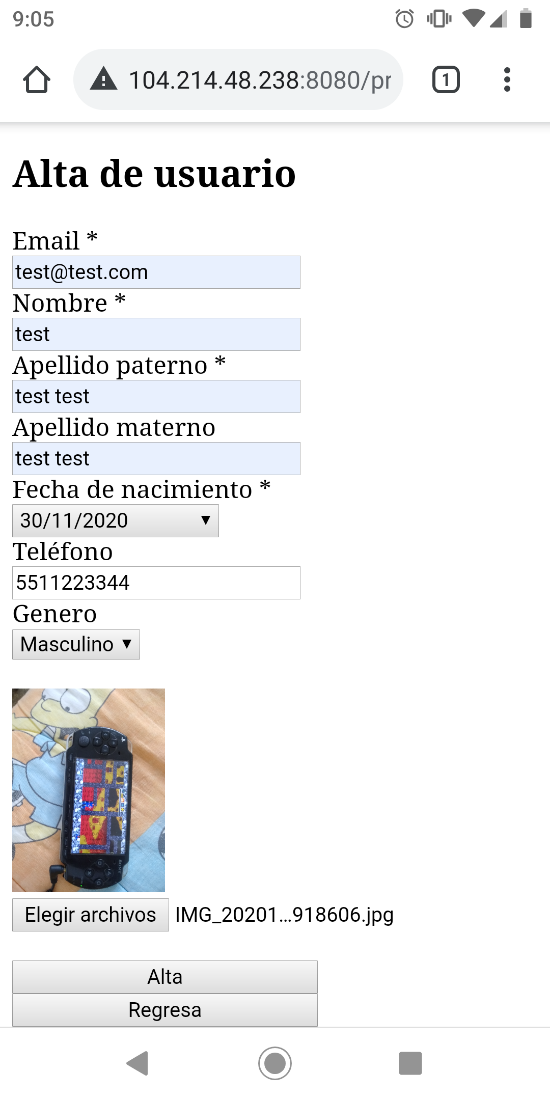
Descripción generada automáticamente

Ahora si queremos consultar el usuario borrado, nos muestra el mensaje de que el usuario no existe.

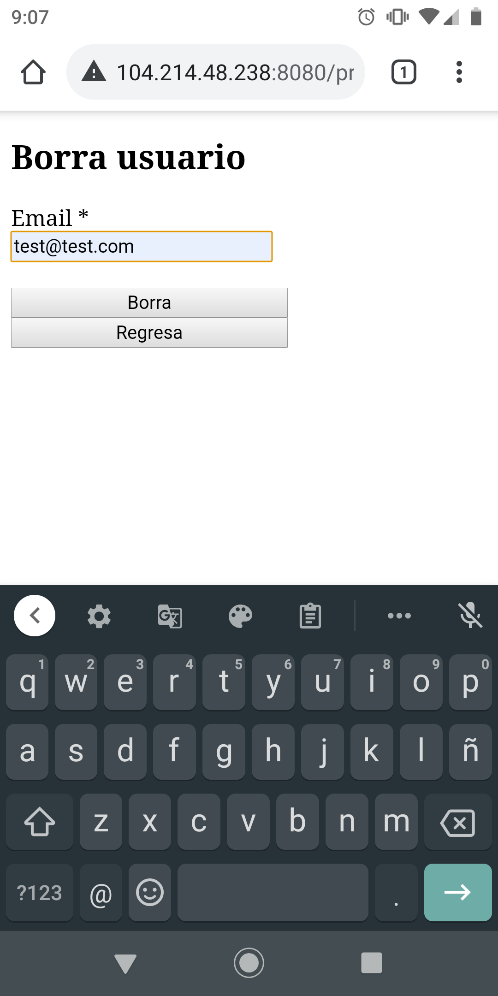
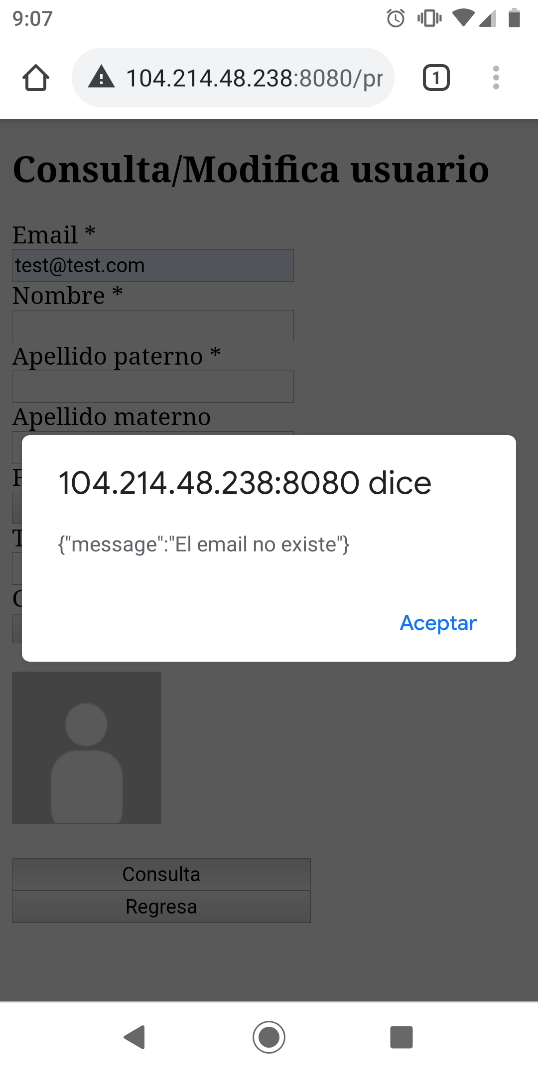
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Repetimos el mismo proceso, pero ahora usando un celular, debido a que el proceso es el mismo, evitare los comentarios.



**Conclusiones.**

En esta tarea se puede observar el funcionamiento básico de las APIs estilo REST, las cuales cada día van siendo más comunes de ver, además de su despliegue en una máquina virtual lo cual hace que la información este completamente distribuida, además que en esta tarea se puede ver claramente el uso y relación del front-end y back-end.