Instituto

Politécnico

Nacional

Escuela Superior de Cómputo

Desarrollo de un cliente para un servicio web estilo REST

TAREA 7

Materia:

Desarrollo De Sistemas Distribuidos

Grupo:

4CV14

Profesor:

Pineda Guerrero Carlos

Alumno:

Castro Cruces Jorge Eduardo

Boleta:

2015080213

Fecha:

Viernes, 5 de noviembre de 2021

# Desarrollo del programa

Desarrollar un programa Java consola (modo carácter) cliente del servicio web REST que implementamos anteriormente. Se deberá realizar las siguientes modificaciones al servicio web:

1. Agregar el campo id\_usuario a la clase Usuario (Usuario.java).

2. Modificar el método web "alta\_usuario" (Servicio.java), de manera que al dar de alta un usuario el método web deberá regresar al cliente el id del usuario agregado. Se deberá desplegar el id del usuario dado de alta. El campo id\_usuario es auto\_increment en la base de datos, por tanto se deberá recuperar el ID inmediatamente después de ejecutar la instrucción INSERT.

3. Modificar el método web "consulta\_usuario" (Servicio.java), ahora la consulta se deberá realizar mediante el id del usuario no el email.

4. Modificar el método web "modifica\_usuario" (Servicio.java), utilizar el id del usuario en el WHERE de las instrucciones SQL en lugar del email. No deberá modificar el id de un usuario ya que se trata de la llave primaria.

5. Modificar el método web "borra\_usuario" (Servicio.java), utilizando como clave el id del usuario no el email.

El programa cliente deberá desplegar el siguiente menú:

MENU

a. Alta usuario  
b. Consulta usuario

c. Borra usuario

d. Salir

Opción: \_

Las opciones deberán implementar la siguiente funcionalidad:

* + La opción "Alta usuario" leerá del teclado el email, el nombre del usuario, el apellido paterno, el apellido materno, la fecha de nacimiento, el teléfono y el género ("M" o "F"). Entonces se invocará el método web "alta\_usuario". Se deberá desplegar el id del usuario dado de alta, o bien, el mensaje de error que regresa el servicio web. Notar que el método web "alta\_usuario" recibe como parámetro una instancia de la clase Usuario, recordemos que esta clase se deberá definir de la siguiente manera:

    class Usuario

    {

      int id\_usuario;

      String email;

      String nombre;

      String apellido\_paterno;

      String apellido\_materno;

      String fecha\_nacimiento;

      String telefono;

      String genero;

      byte[] foto;

    }

Para invocar el método web "alta\_usuario" desde el cliente Java es necesario crear una instancia de la clase Usuario y asignar los valores a los campos (en este caso el campo "foto" será null). Una vez que se tenga el objeto de tipo Usuario se deberá utilizar **GSON**para convertir el objeto a una string JSON, entonces se deberá codificar como URL y el resultado se utilizará como valor del parámetro.

* + La opción "Consulta usuario" leerá del teclado el id de un usuario previamente dado de alta. Entonces se invocará el método web "consulta\_usuario". Si el usuario existe, se desplegará en pantalla el nombre del usuario, el apellido paterno, el apellido materno, la fecha de nacimiento, el teléfono y el género. Debido a que el programa no es gráfico, la foto del usuario se ignorará. Notar que el método web "consulta\_usuario" regresa una string JSON la cual representa un objeto de tipo Usuario, por tanto será necesario utilizar **GSON**para convertir la string JSON a un objeto Java de tipo Usuario y posteriormente desplegar los campos del objeto (excepto el campo "foto"). Si hubo error, se desplegará el mensaje que regresa el servicio web. Una vez desplegados los datos del usuario se preguntará "¿Desea modificar los datos del usuario (s/n)?", si se responde con un caracter "s" entonces se leerá del teclado el email, el nombre del usuario, el apellido paterno, el apellido materno, la fecha de nacimiento, el teléfono y el género ("M" o "F"), si un campo se presiona solo Enter, entonces el campo queda sin modificar.  Entonces se invocará el método web "modifica\_usuario". Este método recibe como parámetro un objeto de tipo Usuario, por tanto se deberá utilizar **GSON**para convertir el objeto a una string JSON, entonces se deberá codificar como URL y el resultado se utilizará como valor del parámetro. Se deberá desplegar "El usuario ha sido modificado" si se pudo modificar el usuario, o bien, el mensaje de error que regresa el servicio web.
  + La opción "Borra usuario" leerá del teclado el id de un usuario previamente dado de alta. Entonces se invocará el método "borra\_usuario" del servicio web. Se deberá desplegar "El usuario ha sido borrado" si se pudo borrar el usuario, o bien, el mensaje de error que regresa el servicio web.
  + La opción "Salir" terminará el programa.

Se deberá realizar las siguientes pruebas:

1. Dar de alta un nuevo usuario.
2. Consultar el usuario dado de alta anteriormente.
3. Modificar algún dato del usuario.
4. Consultar el usuario modificado, para verificar que la modificación se realizó.
5. Intentar borrar un usuario que no exista, se deberá desplegar el mensaje de error indicando que el id no existe.
6. Borrar el usuario dado de alta en el paso 1.

Se deberá entregar un reporte en formato PDF que incluya las capturas de pantalla donde se muestre cada paso, desde la creación de la máquina virtual hasta la compilación y ejecución de las pruebas. El reporte deberá tener portada y conclusiones. Se deberá entregar también todos los archivos utilizados incluyendo el código fuente del servicio web (todos los archivos .java) y el cliente.

El nombre de la máquina virtual deberá ser el número de boleta del alumno, si el número de boleta del alumno es 12345678, entonces la máquina virtual deberá llamarse: CR12345678. **No se admitirá la tarea** si la máquina virtual no se nombra como se indicó anteriormente.

Valor de la tarea: 30% (2.1 puntos de la segunda evaluación parcial)

# Pruebas de escritorio

Ingresar al portal de Azure en la siguiente URL:

[**https://azure.microsoft.com/es-mx/features/azure-portal/**](https://azure.microsoft.com/es-mx/features/azure-portal/)  
  
1. Dar click al botón "Iniciar sesión".  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
2. En el portal de Azure seleccionar "Máquinas virtuales".  
Una captura de pantalla de una computadora

Descripción generada automáticamente

3. Seleccionar la opción "+Crear".  
  
4. Seleccionar la opción "+Virtual machine"  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
5. Seleccionar el grupo de recursos o crear uno nuevo. Un grupo de recursos es similar a una carpeta dónde se pueden colocar los diferentes recursos de nube que se crean en Azure.  
  
6. Ingresar el nombre de la máquina virtual.  
  
7. Seleccionar la región dónde se creará la máquina virtual. Notar que el costo de la máquina virtual depende de la región.  
  
8. Seleccionar la imagen, en este caso vamos a seleccionar Ubuntu Server 18.04 LTS.  
  
9. Dar click en "Seleccionar tamaño" de la máquina virtual, en este caso vamos a seleccionar una máquina virtual con 1 GB de memoria RAM. Dar click en el botón "Seleccionar".  
  
10. En tipo de autenticación seleccionamos "Contraseña".  
  
11. Ingresamos el nombre del usuario, por ejemplo: ubuntu  
  
12. Ingresamos la contraseña y confirmamos la contraseña. La contraseña debe tener al menos 12 caracteres, debe al menos una letra minúscula, una letra mayúscula, un dígito y un carácter especial.  
  
13. En las "Reglas de puerto de entrada" se deberá dejar abierto el puerto 22 para utilizar SSH (la terminal de secure shell).

Una captura de pantalla de una computadora

Descripción generada automáticamente  
14. Dar click en el botón "Siguiente: Discos>"  
  
15. Seleccionar el tipo de disco de sistema operativo, en este caso vamos a seleccionar HDD estándar.  
Captura de pantalla de computadora

Descripción generada automáticamente

16. Dar click en el botón "Siguiente: Redes>"  
  
17. Dar click en el botón "Siguiente: Administración>"  
  
18. En el campo "Diagnóstico de arranque" seleccionar "Desactivado".  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
19. Dar click en el botón "Revisar y crear".  
  
20. Dar click en el botón "Crear".  
Una captura de pantalla de una computadora

Descripción generada automáticamente

21. Dar click a la campana de notificaciones (barra superior de la pantalla) para verificar que la maquina virtual se haya creado.  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
22. Dar click en el botón "Ir al recurso". En la página de puede ver la direción IP pública de la máquina virtual. Esta dirección puede cambiar cada vez que se apague y se encienda la máquina virtual.  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

23. Para conectarnos a la máquina virtual vamos a utilizar el programa ssh disponible en Windows, Linux y MacOS.

24. En una ventana de comandos de Windows o una terminal de Linux o MacOS ejecutar el programa ssh así:

ssh usuario@ip

Captura de pantalla de computadora

Descripción generada automáticamente

Donde **usuario**es el usuario que ingresamos en el paso 11, **ip**es la ip pública de la máquina virtual.

25. Para enviar o recibir archivos de la máquina virtual, se puede utilizar el programa sftp disponible en Windows, Linux y MacOS. Se ejecuta así:

sftp usuario@ip

Para enviar archivos se utiliza el comando put y para recibir archivos se utiliza el comando get.

Para mayor información sobre sftp ver:

[**https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server-es**](https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server-es)  
  
  
**Abrir un puerto de entrada**

Para que los programas que ejecutan en la máquina virtual pueda recibir conexiones a través de un determinado puerto, es necesario crear una regla de entrada para el puerto.

Por ejemplo, vamos a abrir el puerto 50000 en la máquina virtual que acabamos de crear:

1. Entrar al portal de Azure
2. Seleccionar "Maquinas virtuales".
3. Seleccionar la máquina virtual.
4. Dar clic en "Redes".
5. Dar clic en el botón "Agregar regla de puerto de entrada".

Una captura de pantalla de una computadora

Descripción generada automáticamente

1. En el campo "Intervalos de puertos de destino" ingresar: 8080
2. Seleccionar el protocolo: TCP
3. En el campo "Nombre" ingresar un nombre para la regla: Port\_8080

Una captura de pantalla de una computadora

Descripción generada automáticamente

1. Una vez que realizamos la conexión mediante ssh, procedemos a arealizar cada uno de los pasos vistos en clase para la implementación del servidor de aplicaciones Tomcat.

Primeramente instalaremos Tomcat y las bibliotecas necesarias para la implementación de servicios web estilo REST los cuales podrán acceder una base de datos MySQL.

**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.

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

sudo apt update

Pantalla de computadora

Descripción generada automáticamente

sudo apt install openjdk-8-jdk-headless

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

3. Para los siguientes, pasos que realizamos en clase, tomé la decisión de hacerlos previamente en Windows, antes de utilizar SFTP para copiarlos a la máquina virtual, por lo que solamente tuve que enviar dos archivos: Servicio.zip y apache-tomcat-8.5.72.zip, ya con las modificaciones necesarias.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Iniciar/detener el servidor Tomcat**

Previamente, se tuvo que descomprimir ambos archivos ya mencionados.

Pantalla de computadora

Descripción generada automáticamente

Pantalla de computadora

Descripción generada automáticamente

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

Captura de pantalla de computadora

Descripción generada automáticamente

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

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

Captura de pantalla de computadora

Descripción generada automáticamente

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.

**Instalación de MySQL**

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

sudo apt update

2. Instalar el paquete default de MySQL:

sudo apt install mysql-server

Texto

Descripción generada automáticamente

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**

Texto

Descripción generada automáticamente

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

Captura de pantalla de computadora

Descripción generada automáticamente

**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

Captura de pantalla de computadora

Descripción generada automáticamente

**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

Captura de pantalla de computadora

Descripción generada automáticamente

**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=509).

2. Definir la variable de ambiente CATALINA\_HOME:

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

3. Cambiar al directorio dónde se desempacó el archivo [**Servicio.zip**](https://m4gm.com/moodle/mod/resource/view.php?id=509) (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

Captura de pantalla de computadora

Descripción generada automáticamente

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**.

Captura de pantalla de computadora

Descripción generada automáticamente

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

Pantalla de computadora

Descripción generada automáticamente

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.

Pantalla de computadora con fondo negro

Descripción generada automáticamente con confianza media

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.

**Etapa de pruebas del Servicio:**

Se deberá realizar las siguientes pruebas:

1. Dar de alta un nuevo usuario.

Captura de pantalla de computadora

Descripción generada automáticamente

1. Consultar el usuario dado de alta anteriormente.

Captura de pantalla de computadora

Descripción generada automáticamente

1. Modificar algún dato del usuario.(**Telefono**)

Captura de pantalla de computadora

Descripción generada automáticamente

1. Consultar el usuario modificado, para verificar que la modificación se realizó.

Captura de pantalla de computadora

Descripción generada automáticamente

1. Intentar borrar un usuario que no exista, se deberá desplegar el mensaje de error indicando que el id no existe.

Texto

Descripción generada automáticamente

1. Borrar el usuario dado de alta en el paso 1.

Texto

Descripción generada automáticamente

Por último, vamos a **eliminar la máquina virtual** cuando no la use, con la finalidad de ahorrar el saldo de sus cuentas de Azure.

Una captura de pantalla de una computadora

Descripción generada automáticamente

# Conclusiones

En esta práctica se llevó a cabo la implementación de un servicio web para altas, consultas, modificación y borrar datos de usuario.

Una de las partes más complicadas de la práctica fue la configuración del servidor, además, la modificación del cliente para preguntarle al usuario si desea modificar el registro también tuvo un gran nivel de dificultad.

Las partes más sencillas fue la modificación del archivo servicio.java ya que eran pocas líneas de código y solamente era cambiar al id\_usuario.

Esta práctica ha sido una de las más complicadas de elaborar, pero con una gran curva de aprendizaje a su término-