



**Instituto Politecnico Nacional**

**ESCOM “ESCUELA SUPERIOR DE  
CÓMPUTO”**



*DESARROLLO DE SISTEMAS DISTRIBUIDOS*

*TAREA 8. DESARROLLO DE UN CLIENTE PARA UN SERVICIO WEB REST*

PROFE: CARLOS PINEDA GUERRERO

ALUMNO: Rojas Alvarado Luis Enrique

GRUPO: 4CM5

Para esta práctica se desplegó el servicio web que ya habíamos configurado en la tarea anterior, pero antes se tuvo que habilitar el puerto 8080 para el consumo del servicio, entrando al apartado de redes y dando click en el campo “agregar regla de puerto de entrada”.

luis | Redes  
Máquina virtual

Buscar (Ctrl+/) << Adjuntar interfaz de red Desasociar interfaz de red

Configuración de IP ⓘ  
ipconfig1 (Principal)

Interfaz de red: **luis358** Reglas de seguridad vigentes Topología  
Red virtual/subred: Servicio\_Web-vnet/default IP pública de NIC: 13.65.184.75 IP privada de NIC: 10.0.0.4 Redes aceleradas: Deshabilitado

Reglas de puerto de entrada Reglas de puerto de salida Grupos de seguridad de aplicación Equilibrio de carga

Grupo de seguridad de red **luis-nsg** (se conectó a la interfaz de red: **luis358**)  
Impactos 0 subredes, 1 interfaces de red

**Agregar regla de puerto de entrada**

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
300	SSH	22	TCP	Cualquiera	Cualquiera	Permitir
310	Port_8080	8080	TCP	Cualquiera	Cualquiera	Permitir
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Permitir
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Permitir
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Denegar

Una vez hecho click se debe seleccionar el protocolo TCP y en el campo “Intervalos de puerto destino” debemos escribir el “8080”, sin olvidar seleccionar “permitir” en el campo de Acción. Para finalizar daremos clic en el botón de “agregar”.

**Agregar regla de seguridad de entrada** ×  
luis-nsg

Básica

Origen \* ⓘ  
Any

Intervalos de puertos de origen \* ⓘ  
\*

Destino \* ⓘ  
Any

Intervalos de puertos de destino \* ⓘ  
8080

Protocolo \*  
Any TCP UDP ICMP

Acción \*  
Permitir Denegar

Prioridad \* ⓘ  
320

**Agregar**

Recordemos que debemos desempacar el archivo “Servicio.zip” y mover el archivo Servicio.war al directorio webapps de tomcat 8. y por último iniciar y probar el puerto 8080 esté correctamente activado

```
META-INF Servicio.war WEB-INF jaxrs-ri negocio
Servicio Servicio.zip apache-tomcat-8.5.60 master.zip
luis@luis:~$ mv Servicio.
Servicio.war Servicio.zip
luis@luis:~$ mv Servicio.war apache-tomcat-8.5.60/webapps/
luis@luis:~$ sh $CATALINA_HOME/bin/catalina.sh start
Using CATALINA_BASE:   /home/luis/apache-tomcat-8.5.60/
Using CATALINA_HOME:   /home/luis/apache-tomcat-8.5.60/
Using CATALINA_TMPDIR: /home/luis/apache-tomcat-8.5.60/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH:       /home/luis/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/
luis/apache-tomcat-8.5.60/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
luis@luis:~$
```

Compilación y ejecución del programa Usuario.java

```
wicho@wicho:~/Documentos/Distribuidos/Tareas/Tarea8$ javac -cp gson-2.8.6.jar Usuario.java
wicho@wicho:~/Documentos/Distribuidos/Tareas/Tarea8$ java -cp gson-2.8.6.jar:. Usuario
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

Para probar programa que consume el servicio web, se dan de alta 3 usuarios con sus respectivos datos.

```
wicho@wicho:~/Documentos/Distribuidos/Tareas/Tarea8$ java -cp gson-2.8.6.jar:. Usuario
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
a
Email:
prueba@test.com
Nombre:
prueba
Apellido Paterno:
Servicio
Apellido Materno:
Web
Fecha de nacimiento:
2020-12-07
Telefono:
5528031867
Genero (M/F):
M
a. Alta usuario
b. Consulta usuario
```

El primer usuario en dar de alta está descrito con el correo de prueba@test.com.

Dando de alta 2 usuarios más, 2 personas (masculina y femenina) con los correos [luisRojas@gmail.com](mailto:luisRojas@gmail.com) e [irene@gmail.com](mailto:irene@gmail.com) respectivamente.

```
a
Email:
luisRojas@gmail.com
Nombre:
Luis
Apellido Paterno:
ROjas
Apellido Materno:
Alvarado
Fecha de nacimiento:
1998-02-05
Telefono:
5528360147
Genero (M/F):
M
```

```
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

```
a
```

```
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

```
a
Email:
irene@gmail.com
Nombre:
Irene
Apellido Paterno:
Alvarado
Apellido Materno:
Rodriguez
Fecha de nacimiento:
1968-02-14
Telefono:
5539554477
Genero (M/F):
F
```

Consultando los usuarios que acabamos de dar de alta. El primero de ellos será Irene. Para ello ingresamos el correo del usuario: “[irene@gmail.com](mailto:irene@gmail.com)” y podemos observar la información ingresada es la misma y nos devuelve de nuevo al servicio web.

```
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
b
Introduce el e-mail del Usuario que se quiere consultar
irene@gmail.com
Email: irene@gmail.com
Nombre: Irene
Apellido Paterno: Alvarado
Apellido Materno: Rodriguez
Fecha de nacimiento: 1968-02-14
Telefono: 5539554477
Genero: F
Foto: null
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

El siguiente para consultar es el usuario prueba, para consultar su usuario ingresamos el correo con el que lo dimos de alta, el cual es "[prueba@test.com](mailto:prueba@test.com)" y nos despliega la información que ingresamos.

```
Introduce el e-mail del Usuario que se quiere consultar
prueba@test.com
Email: prueba@test.com
Nombre: prueba
Apellido Paterno: Servicio
Apellido Materno: Web
Fecha de nacimiento: 2020-12-07
Telefono: 5528031867
Genero: M
Foto: null
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

El próximo a consultar es el usuario Luis, con el correo "[luisRojas@gmail.com](mailto:luisRojas@gmail.com)"

```
Introduce el e-mail del Usuario que se quiere consultar
luisRojas@gmail.com
Email: luisRojas@gmail.com
Nombre: Luis
Apellido Paterno: ROjas
Apellido Materno: Alvarado
Fecha de nacimiento: 1998-02-05
Telefono: 5528360147
Genero: M
Foto: null
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

Ahora procedemos a eliminar un usuario. Para eliminar un usuario solo debemos de introducir el e-mail del usuario que queramos eliminar. Entonces colocamos el usuario de prueba "[prueba@test.com](mailto:prueba@test.com)". Para consultar los cambios que hemos hecho procederemos a consultar el usuario que acabamos de eliminar para comprobar que realmente se ha eliminado el usuario.

Podemos observar que al ingresar el e-mail del usuario que eliminaremos nos devuelve otra vez al servicio.

```
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
c
Introduce el e-mail del Usuario que se quiere consultar
prueba@test.com

a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

Entonces para comprobar colocamos el correo del usuario que eliminamos “[prueba@test.com](mailto:prueba@test.com)” y después manda un error que nos comunica que el usuario no ha sido encontrado.

```
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
b
Introduce el e-mail del Usuario que se quiere consultar
prueba@test.com
No encontrado
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

El siguiente paso es borrar todos los usuarios, recordemos que nos quedan 2 usuarios en la base de datos (Luis Rojas Alvarado, con correo [luisRojas@gmail.com](mailto:luisRojas@gmail.com) e Irene Alvarado Rodriguez con correo [irene@gmail.com](mailto:irene@gmail.com)), por lo que seleccionamos la opción “d” del servicio y borrar todos los usuarios, una vez borrados, nos devolverá al servicio y después podremos consultar si lo que elegimos es correcto (hacemos consulta de los usuarios borrados para comprobar que realmente se han eliminado).

Entonces seleccionamos la opción “d” y eliminamos todos los usuarios, solo para ser redirigidos al mismo servicio después de haber seleccionado la opción.

```
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
d

a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
b
Introduce el e-mail del Usuario que se quiere consultar
irene@gmail.com
No encontrado
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

Comprobamos primeramente el usuario de Irene, ingresando la opción “b” y tecleando por consola su correo electrónico: [irene@gmail.com](mailto:irene@gmail.com). Podemos observar que el servicio nos devuelve un mensaje de error al no encontrar al usuario.

De la misma manera consultamos al usuario Luis con el correo [luisRojas@gmail.com](mailto:luisRojas@gmail.com). Y como podemos observar, el servicio nos devuelve el mismo mensaje de error al no encontrar el usuario.

```
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
b
Introduce el e-mail del Usuario que se quiere consultar
luisRojas@gmail.com
No encontrado
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir
```

## Conclusiones

En esta práctica se observa el comportamiento de un servicio web y un cliente que lo consume en una máquina local en cualquier parte del mundo. Es una fuerte herramienta porque nos muestra como es el mundo real de las aplicaciones que consumen los servicios web en la nube y en un servidor. Por lo que también se pueden hacer aplicaciones para android que requieren consumir servicios web REST y que por consiguiente tienen un alcance bastante amplio en el mundo de la web. Hablando del código, solo tenemos que acoplarnos y saber qué es lo que se está mandando y qué está recibiendo el servidor, ya que si no son compatibles los tipos de datos es posible que haya una ruptura o no funcione correctamente la aplicación que consume el servicio.