****

**Instituto Politecnico Nacional**

**ESCOM “ESCUELA SUPERIOR DE CÓMPUTO”**

*DESARROLLO DE SISTEMAS DISTRIBUIDOS*

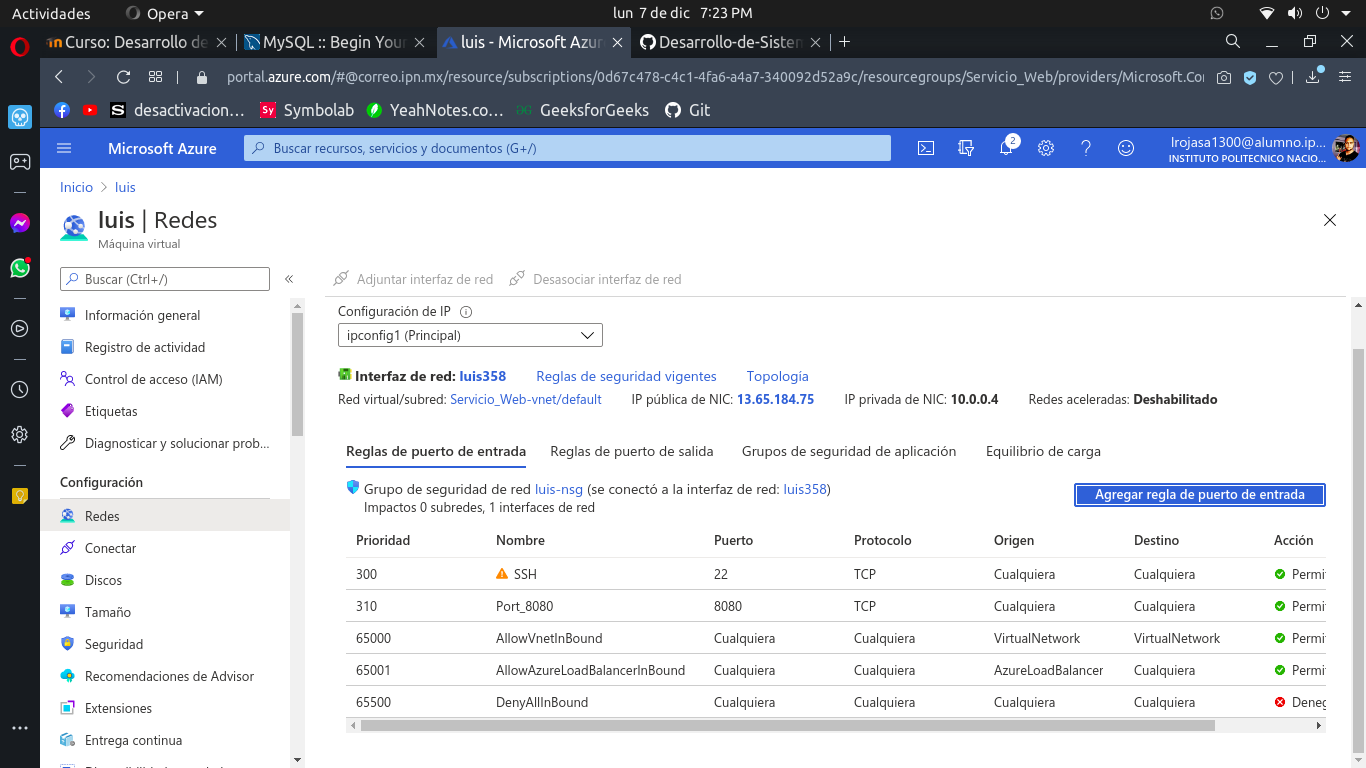
*TAREA 8. DESARROLLO DE UN CLIENTE PARA UN SERVICIO WEB REST*

PROFE: CARLOS PINEDA GUERRERO

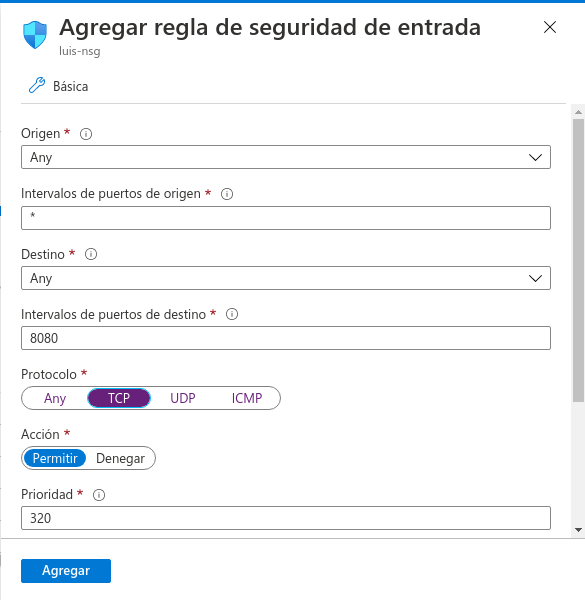
ALUMNO: Rojas Alvarado Luis Enrique

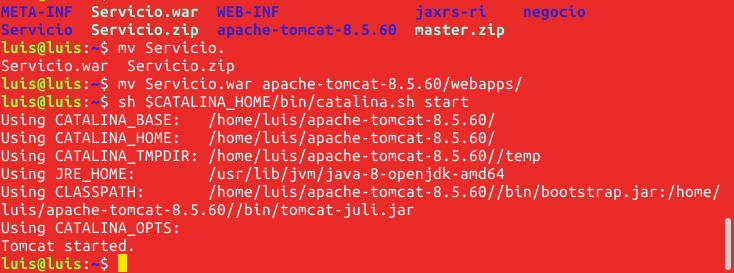
GRUPO: 4CM5

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”.

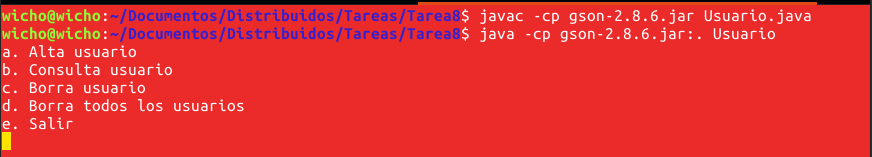


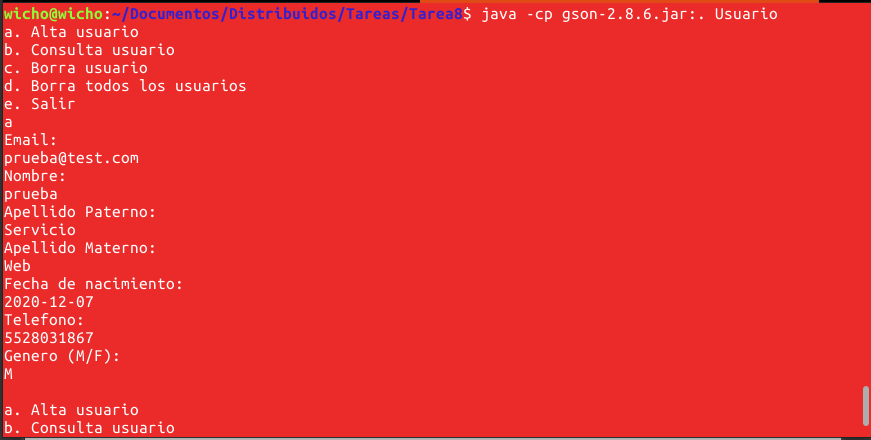
Una vez hecho click se debe seleccionar el protocolo TCP y en el campo “Intervalos de puerto destino” debemos escribir el “8080”, sin olvidar seleccionar “permitir” en el campo de Acción. Para finalizar daremos clic en el botón de “agregar”.



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 correctamenteactivado

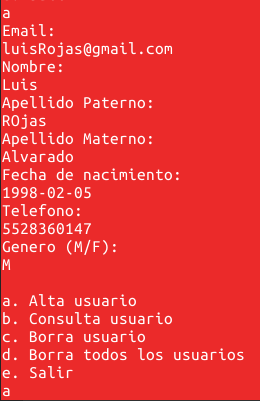
Compilación y ejecución del programa Usuario.java

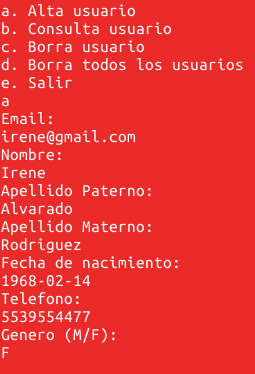
Para probar programa que consume el servicio web, se dan de alta 3 usuarios con sus respectivos datos.



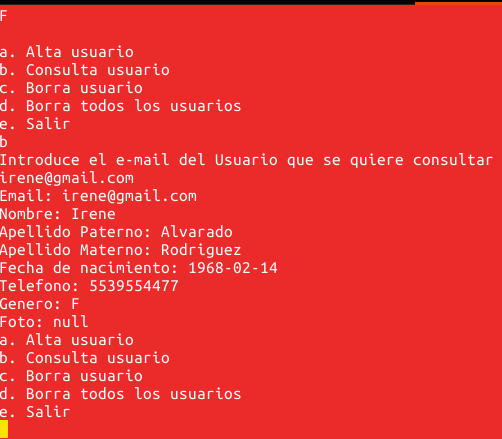
El primer usuario en dar de alta está descrito con el correo de prueba@test.com.

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.

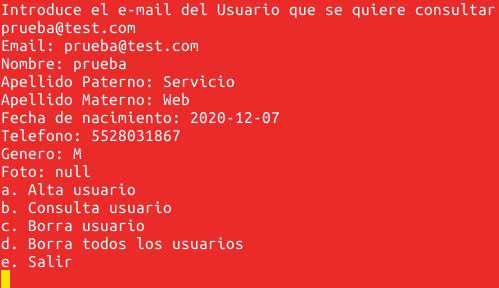




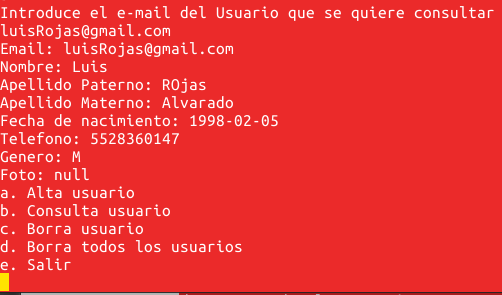
Consultando los usuarios que acabamos de dar de alta. El primero de ellos será Irene. Para ello ingresamos el correo del usuario: “[irene@gmail.com](mailto:irene@gmail.com)” y podemos observar la información ingresada es la misma y nos devuelve de nuevo al servicio web.



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.

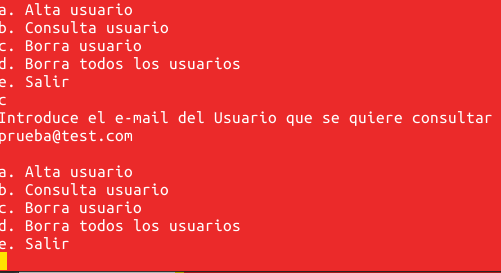


El próximo a consultar es el usuario Luis, con el correo “[luisRojas@gmail.com](mailto:luisRojas@gmail.com)”

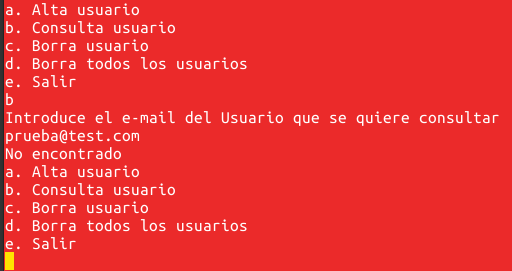


Ahora procedemos a eliminar un usuario. Para eliminar un usuario solo debemos de introducir el e-mail del usuario que queramos eliminar. Entonces colocamos el usuario de prueba “[prueba@test.com](mailto:prueba@test.com)”. Para consultar los cambios que hemos hecho procederemos a consultar el usuario 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.

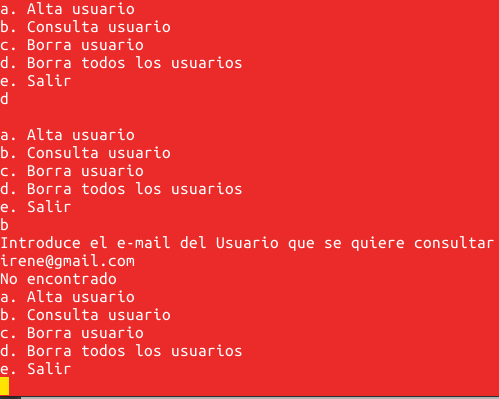


Entonces para comprobar colocamos el correo del usuario que eliminamos “[prueba@test.com](mailto:prueba@test.com)“ y después manda un error que nos comunica que el usuario no ha sido encontrado.



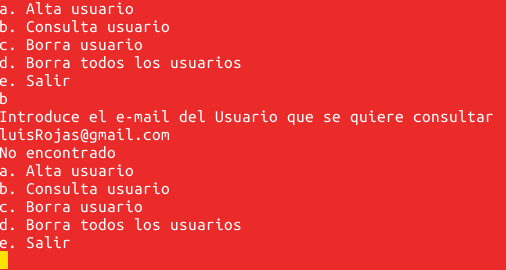
El siguiente paso es borrar todos los usuarios, recordemos que nos quedan 2 usuarios en la base de datos (Luis Rojas Alvarado, con correo [luisRojas@gmail.com](mailto:luisRojas@gmail.com) e Irene Alvarado Rodriguez con correo [irene@gmail.com](mailto:irene@gmail.com))*,* por lo que seleccionamos la opción “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.



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.



**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.