



INSTITUTO POLITÉCNICO NACIONAL.
ESCUELA SUPERIOR DE CÓMPUTO.



Sistemas Distribuidos

Tarea 07: Reporte de “Desarrollo de un cliente para un servicio web estilo REST”

Alumno: Oaxaca Pérez David Arturo

Grupo:

4CV12

A cargo del profesor:

PINEDA GUERRERO CARLOS

Contenido

| | |
|---|----|
| Introducción | 3 |
| Desarrollo | 3 |
| Creación y configuración de la máquina virtual..... | 3 |
| Compilación y ejecución del cliente..... | 13 |
| Conclusiones | 18 |

Introducción

En esta práctica nuevamente utilizaremos el servicio web de estilo REST (REpresentational State Transfer) que fue realizado durante la tarea pasado y se conservó al crear la imagen de la máquina virtual que estábamos usando, usando la definición previamente dada tenemos que este estilo de arquitectura para un servicio web define un conjunto de principios arquitectónicos para la creación de servicios web, fue diseñado por Roy Fielding en el año 2000 en su tesis doctoral "Architectural Styles and the Design of Network-base Software Architectures".

El diseño de servicios web estilos REST sigue cuatro principios:

- Utilizar métodos HTTP de forma explícita
- Los servicios son "sin estado" (stateless)
- Los URIs representan una estructura de directorios
- Se transfiere XML, JSON o ambos

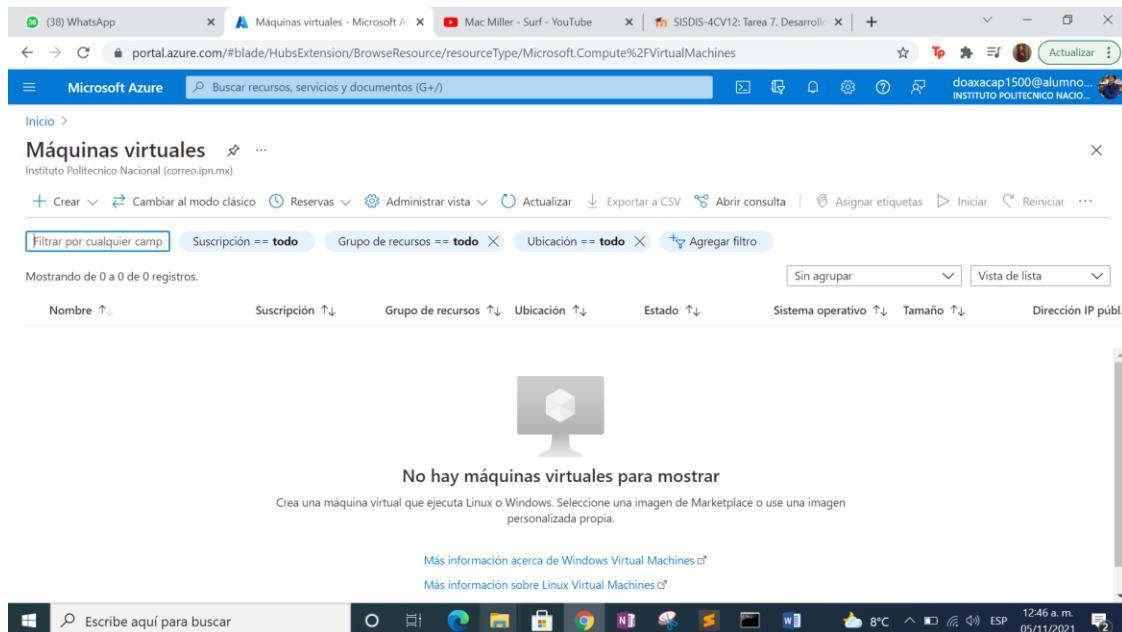
En esta tarea crearemos un cliente desde Java que realicé peticiones mediante el método POST y envié los parámetros necesarios para realizar el query, que en este caso puede ser un objeto de tipo Usuario o una variable de tipo int que representara el ID del usuario.

Desarrollo

En el desarrollo se podrá ver como se fue trabajando la práctica, como se implementó la máquina virtual y las pruebas que se realizaron.

Creación y configuración de la máquina virtual

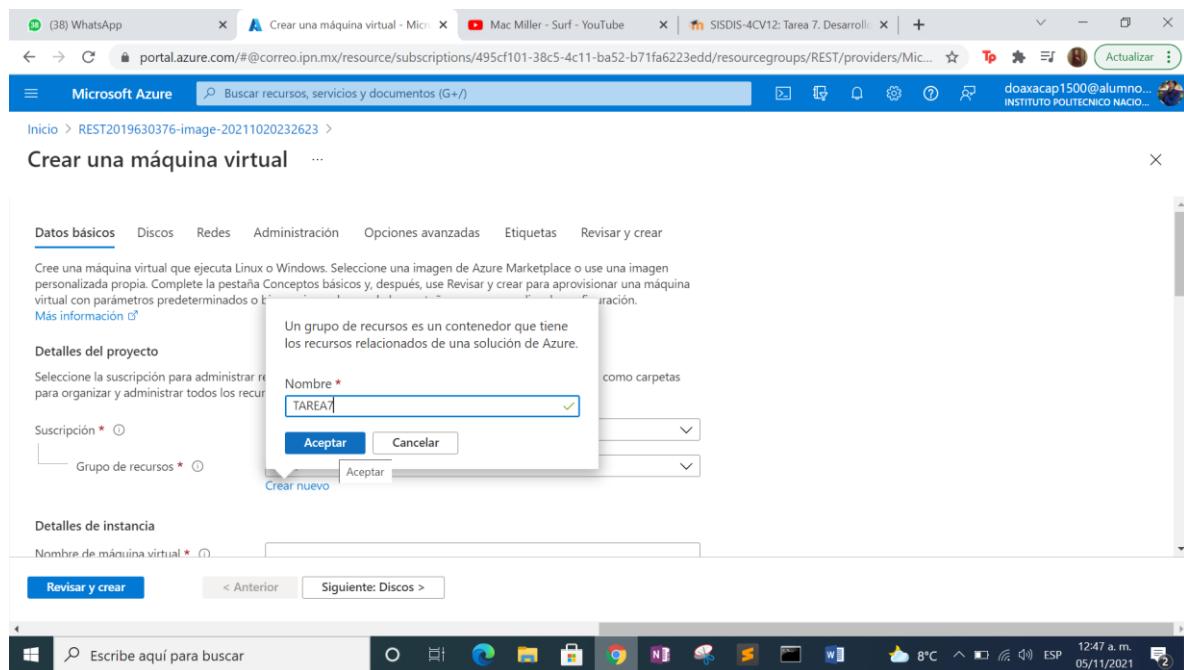
En un inicio podemos ver que no contamos con ninguna máquina virtual creada por el momento en la cuenta de Azure.



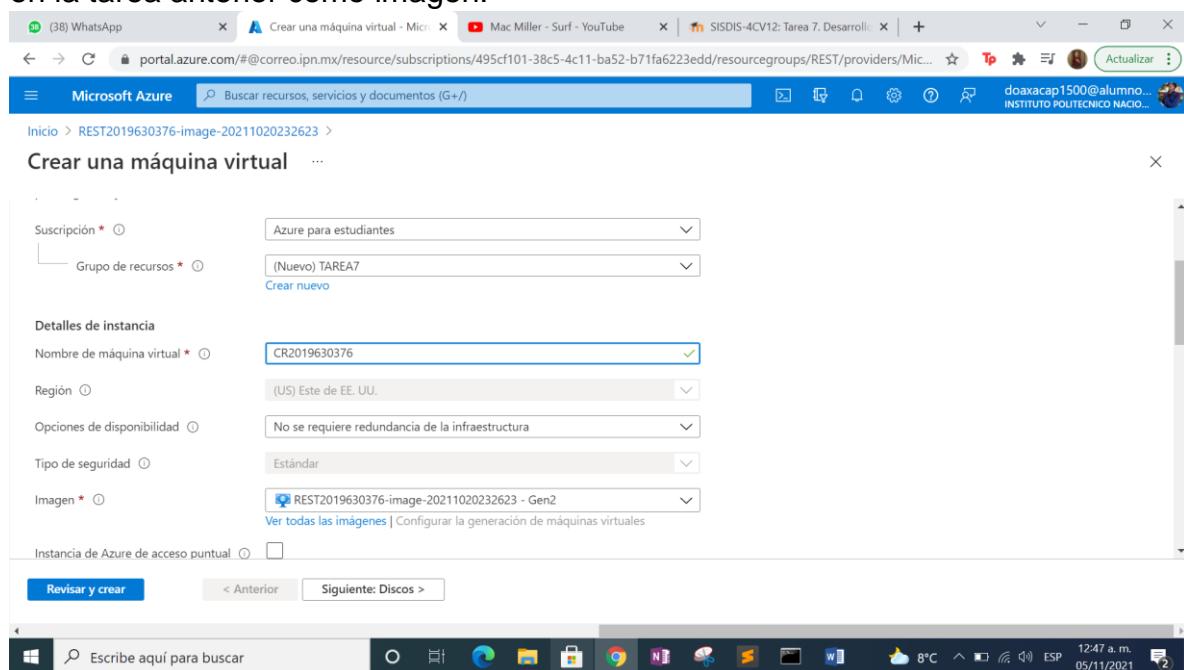
Para crear una máquina virtual en esta ocasión, necesitaremos la imagen de la máquina virtual que creamos en tarea anterior, vamos a la información general de la imagen y presionamos en crear una máquina virtual para usar esta imagen como base.

Tras eso, podemos ver en la siguiente captura de pantalla la suscripción con la que se creara la nueva maquina virtual.

Creamos un nuevo grupo de recursos, que nos facilitara borrar todos los recursos exceptuando la imagen al terminar la práctica.



En la siguiente captura podemos observar que el nombre usado será "CR2019630376" de acuerdo con los lineamientos de la práctica y también podemos observar que se está usando la imagen de la máquina virtual construida en la tarea anterior como imagen.



Posteriormente podemos ver que se escogerán 2GB de tamaño para que cualquier proceso que tengamos que hacer no sea demasiado tardado.

Nombre de máquina virtual * CR2019630376

Región (US) Este de EE. UU.

Opciones de disponibilidad No se requiere redundancia de la infraestructura

Tipo de seguridad Estándar

Imagen * REST2019630376-image-20211020232623 - Gen2

Instancia de Azure de acceso puntual

Tamaño * Standard_B1ms - 1 vcpu, 2 GiB de memoria (318,84 MXN/mes)

Cuenta de administrador Cloudmilkinea CCW

Revisar y crear < Anterior Siguiente: Discos >

Después de eso escogemos el uso de contraseña como método de autenticación, usaremos el nombre de usuario “Ubuntu” y una contraseña que cumple con los requisitos solicitados por Azure. Además de eso, dejaremos abierto el puerto 22 para conectarnos por medio de SSH.

Tipo de autenticación Contraseña

Nombre de usuario * Ubuntu

Contraseña * *****

Confirmar contraseña * *****

Reglas de puerto de entrada

Selección de puertos de entrada públicos * Ninguno

Seleccionar puertos de entrada * SSH (22)

Revisar y crear < Anterior Siguiente: Discos >

En la sección de discos duros escogemos HDD estándar como disco duro para el sistema operativo ya que esto nos ayudara a economizar en el uso de la máquina virtual.

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The current step is 'Disks'. Under 'Options of disk', 'HDD standard (almacenamiento con redundancia local)' is selected. A note states: 'El tamaño de la máquina virtual seleccionada es compatible con los discos premium. Se recomienda SSD Premium para elevadas cargas de trabajo de E/S por segundo. Las máquinas virtuales con discos SSD Premium optan al acuerdo de nivel de servicio de conectividad del 99,9%.' Below it, 'Tipo de cifrado' is set to '(Predeterminado) Cifrado en reposo con una clave administrada por la pl...' and 'Habilitar compatibilidad con Ultra Disks' is unchecked. At the bottom are buttons for 'Revisar y crear' (Review + Create) and 'Siguiente: Redes >' (Next: Networks).

Deshabilitamos los diagnósticos de arranque y nos dirigimos al botón de “Revisar y crear”.

The screenshot shows the 'Administration' step of the 'Create a virtual machine' wizard. Under 'Azure Security Center', there is a checked checkbox: 'La suscripción está protegida por el plan básico de Azure Security Center.' In the 'Supervisión' section, 'Deshabilitar' is selected for 'Diagnosticos de arranque'. There is also an unchecked checkbox for 'Habilitar diagnósticos del SO invitado'. At the bottom are buttons for 'Revisar y crear' (Review + Create), '< Anterior' (Previous), and 'Siguiente: Opciones avanzadas >' (Next: Options advanced).

En la siguiente captura podemos ver que la validación para la creación de la máquina virtual ha sido superada.

The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The validation status is marked as 'Validación superada' (Validation passed). The configuration summary indicates the creation of a VM named 'REST2019630376-image-20211020232623' with 1 vCPU and 2 GiB of memory. A warning message states: 'Ha establecido los siguientes puertos abiertos para Internet: SSH. Esto solo se recomienda para las pruebas. Si quiere cambiar esta configuración, vuelva a la pestaña de aspectos básicos.' (You have set the following ports open for the Internet: SSH. This is only recommended for testing. If you want to change this configuration, go back to the basic options tab.)

Y en esta captura podemos ver que el recurso esta en implementación.

The screenshot shows the 'Deployment Details' blade for the VM 'CreateVm-REST2019630376-image-20211020232623'. It displays the message 'La implementación está en curso...' (Deployment is in progress...) and 'Se está realizando la implementación en el grupo de recursos 'TAREA7''. The blade also lists the resources being deployed, including 'TAREA7-vnet', 'CR2019630376-ip', and 'CR2019630376-nsg', all in a 'Created' state. The deployment started at 5/11/2021 0:49:21.

Aquí podemos ver la información de la máquina virtual al acceder al recurso una vez que fue creada, por cuestiones de tiempo, esta fue apagada antes de la ejecución del programa para economizar así que posteriormente aparece con una dirección IP diferente que fue la que finalmente se usó al correr el cliente.

The screenshot shows the Microsoft Azure portal interface. The main title bar includes tabs for WhatsApp, Microsoft Azure, Mac Miller - Surf - YouTube, and SISDIS-4CV12: Tarea 7. Desarrollo. The Microsoft Azure tab is active, showing the search bar 'Buscar recursos, servicios y documentos (G+ /)'. Below the search bar, the navigation menu shows 'Máquina virtual' selected. The main content area displays the details for 'CR2019630376'. Under the 'Información general' section, it shows the subscription 'TAREA7', status 'En ejecución', location 'Este de EE. UU.', and IP address '137.135.90.140'. The 'Propiedades' tab is selected, showing the machine name 'CR2019630376' and its public IP '137.135.90.140'. The bottom status bar shows the date and time as 05/11/2021 12:56 a.m.

En esta captura podemos ver como se crea una nueva regla de seguridad en redes para abrir el puerto 8080 hacia peticiones por medio de protocolo TCP que serán las que enviara el cliente.

The screenshot shows the Microsoft Azure portal interface, specifically the 'Redes' (Network) section for the VM 'CR2019630376'. The left sidebar shows 'Redes' selected. A modal window titled 'Agregar regla de seguridad de entrada' (Add inbound security rule) is open. In the 'Intervalos de puertos de destino' field, '8080' is entered. The 'Protocolo' section has 'TCP' selected. In the 'Reglas de puerto de salida' table, there is one entry: 'Port_8080' with 'Nombre' 'Port_8080'. At the bottom of the modal, there are 'Agregar' (Add) and 'Cancelar' (Cancel) buttons. The bottom status bar shows the date and time as 05/11/2021 12:56 a.m.

Esta captura muestra la nueva dirección de la máquina virtual una vez que la volvimos a encender al día siguiente.

The screenshot shows the Microsoft Azure portal interface. The main window displays the details of a virtual machine named "CR2019630376". The "Información esencial" (Essential information) section provides the following details:

- Grupo de recursos (Resource group): TAREA7
- Sistema operativo (Operating system): Linux
- Tamaño (Size): B1ms estándar (1 vcpu, 2 GiB de memoria)
- Dirección IP pública (Public IP address): 20.120.31.23
- Red virtual/subred (Virtual network/subnet): TAREA7-vnet/default
- Nombre DNS (DNS name): Sin configurar (Not configured)

The left sidebar shows the navigation menu with options like "Información general" (General information), "Configuración" (Configuration), and "Propiedades" (Properties). The "Propiedades" tab is selected, showing sections for "Máquina virtual" (Virtual machine) and "Redes" (Networks). The status bar at the bottom indicates the date and time as 05/11/2021 10:40 a.m.

Aquí podemos ver como accedemos a la máquina virtual por medio de SSH.

The screenshot shows a terminal session within the Microsoft Azure portal, connected to the virtual machine "CR2019630376". The terminal output shows the user logging in to the Ubuntu 18.04.6 LTS system:

```
Ubuntu@CR2019630376:~$
```

The terminal also displays system information and upgrade notifications:

```
System information as of Fri Nov  5 16:39:22 UTC 2021
System load: 0.11      Processes:    103
Usage of /:  7.6% of 28.90GB  Users logged in:   0
Memory usage: 18%          IP address for eth0: 10.0.0.4
Swap usage:  0%
* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

5 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Nov  5 07:09:27 2021 from 189.203.231.167
Ubuntu@CR2019630376:~$
```

The terminal window has a black background with white text. The Azure interface elements like the sidebar and status bar are visible at the top and bottom of the window.

Posteriormente revisamos que en efecto tenga todo lo que debería contener al ser creada a partir de una imagen de la máquina virtual que se creó en la tarea anterior.

```

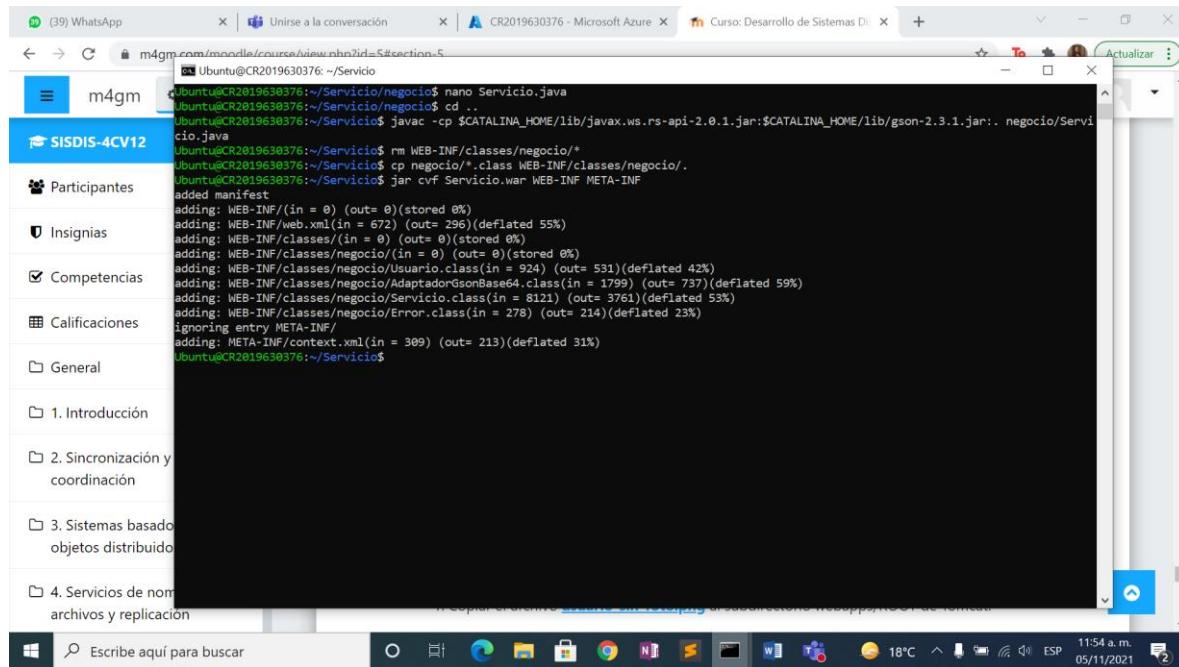
Ubuntu@CR2019630376:~/Servicio
ubuntu@CR2019630376:~$ ls
servicio apache-tomcat-8.5.72 jaxrs-ri mysql-connector-java-8.0.27
Servicio.zip apache-tomcat-8.5.72.zip jaxrs-ri-2.24.zip mysql-connector-java-8.0.27.zip
Ubuntu@CR2019630376:~$ cd apache-tomcat-8.5.72
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ ls
BUILDING.txt LICENSE README.md RUNNING.txt conf logs webapps
CONTRIBUTING.md NOTICE RELEASE-NOTES bin lib temp work
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ cd webapps
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ ls
ROOT Servicio Servicio.war
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ rm -r Servicio
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ ls
ROOT Servicio.war
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ rm Servicio.war
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ ls
ROOT
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ cd ..
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ cd ..
Ubuntu@CR2019630376:~$ ls
Servicio apache-tomcat-8.5.72 jaxrs-ri mysql-connector-java-8.0.27
Servicio.zip apache-tomcat-8.5.72.zip jaxrs-ri-2.24.zip mysql-connector-java-8.0.27.zip
Ubuntu@CR2019630376:~$ cd Servicio
Ubuntu@CR2019630376:~/Servicio$ ls
META-INF WEB-INF negocio
Ubuntu@CR2019630376:~/Servicio$ pwd
/home/Ubuntu/Servicio
Ubuntu@CR2019630376:~/Servicio$ export CATALINA_HOME=/home/Ubuntu/Servicio/apache-tomcat-8.5.72
Ubuntu@CR2019630376:~/Servicio$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Ubuntu@CR2019630376:~/Servicio$
```

Posteriormente borramos el archivo war actual y el directorio de Servicio que se encuentra en Apache Tomcat y creamos las variables CATALINA_HOME y JAVA_HOME (La primera se creó varias veces debido a algunos errores al escribir la ruta).

```

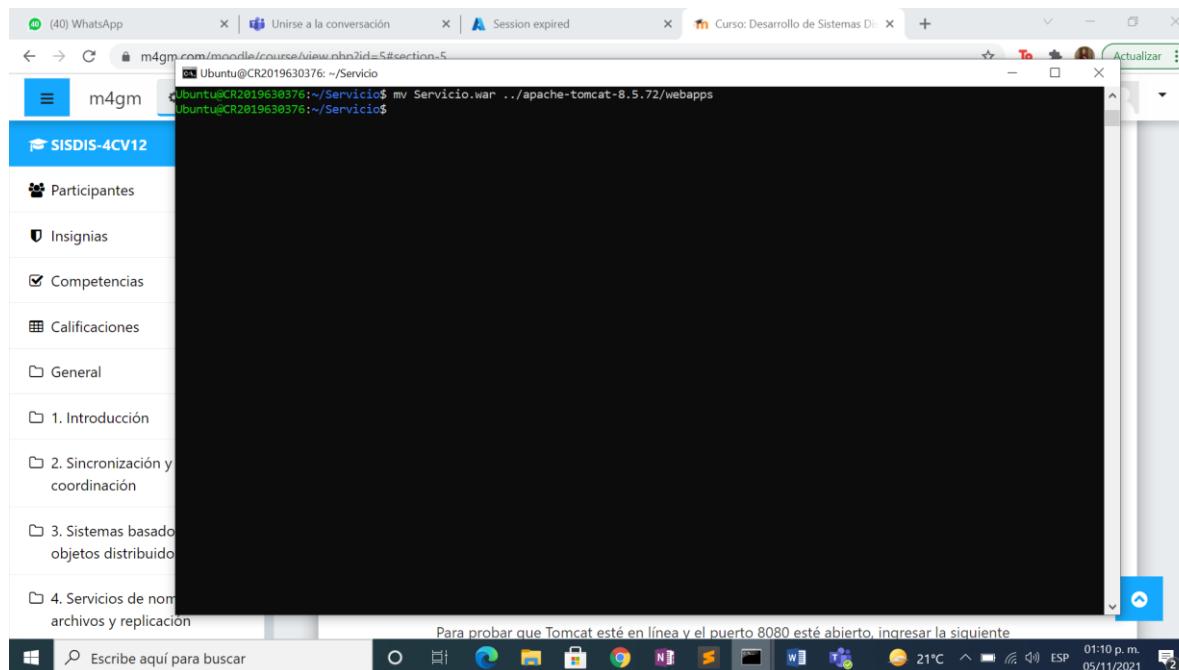
Ubuntu@CR2019630376:~/Servicio
ubuntu@CR2019630376:~$ cd webapps
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ ls
ROOT Servicio Servicio.war
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ rm -r Servicio
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ ls
ROOT Servicio.war
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ rm Servicio.war
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ ls
ROOT
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ cd ..
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ cd ..
Ubuntu@CR2019630376:~$ ls
Servicio apache-tomcat-8.5.72 jaxrs-ri mysql-connector-java-8.0.27
Servicio.zip apache-tomcat-8.5.72.zip jaxrs-ri-2.24.zip mysql-connector-java-8.0.27.zip
Ubuntu@CR2019630376:~$ cd Servicio
Ubuntu@CR2019630376:~/Servicio$ ls
META-INF WEB-INF negocio
Ubuntu@CR2019630376:~/Servicio$ pwd
/home/Ubuntu/Servicio
Ubuntu@CR2019630376:~/Servicio$ export CATALINA_HOME=/home/Ubuntu/Servicio/apache-tomcat-8.5.72
Ubuntu@CR2019630376:~/Servicio$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Ubuntu@CR2019630376:~/Servicio$ cd ..
Ubuntu@CR2019630376:~$ export CATALINA_HOME=/home/Ubuntu/Servicio/apache-tomcat-8.5.72
Ubuntu@CR2019630376:~$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Ubuntu@CR2019630376:~$ cd Servicio
Ubuntu@CR2019630376:~/Servicio$ ls
META-INF WEB-INF negocio
Ubuntu@CR2019630376:~/Servicio$ cd ..
Ubuntu@CR2019630376:~$ export CATALINA_HOME=/home/Ubuntu/apache-tomcat-8.5.72
Ubuntu@CR2019630376:~$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Ubuntu@CR2019630376:~$ cd Servicio
Ubuntu@CR2019630376:~/Servicio$ export CATALINA_HOME=/home/Ubuntu/apache-tomcat-8.5.72
Ubuntu@CR2019630376:~/Servicio$
```

Posteriormente compilamos el nuevo archivo de Servicio.java con todas las modificaciones que hicimos para que efectivamente trabaje con el ID de usuario y retorne este al ser dado de alta un nuevo usuario.



```
Ubuntu@CR2019630376:~/Servicio$ nano Servicio.java
Ubuntu@CR2019630376:~/Servicio$ cd ..
Ubuntu@CR2019630376:~/Servicio$ javac -cp $CATALINA_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java
Ubuntu@CR2019630376:~/Servicio$ rm WEB-INF/classes/negocio/*
Ubuntu@CR2019630376:~/Servicio$ cp negocio/*.class WEB-INF/classes/negocio/
Ubuntu@CR2019630376:~/Servicio$ jar cvf Servicio.war WEB-INF META-INF
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/web.xml(in = 672) (out= 296)(deflated 55%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/negocio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/negocio/Usuario.class(in = 924) (out= 531)(deflated 42%)
adding: WEB-INF/classes/negocio/AdaptadorGsonBase64.class(in = 1799) (out= 737)(deflated 59%)
adding: WEB-INF/classes/negocio/Servicio.class(in = 8121) (out= 3761)(deflated 53%)
adding: WEB-INF/classes/negocio/Error.class(in = 278) (out= 214)(deflated 23%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 309) (out= 213)(deflated 31%)
Ubuntu@CR2019630376:~/Servicio$
```

Posteriormente movemos el nuevo archivo war al directorio webapps de Apache.



```
Ubuntu@CR2019630376:~/Servicio$ mv Servicio.war ../apache-tomcat-8.5.72/webapps
Ubuntu@CR2019630376:~/Servicio$
```

En esta captura se puede ver como iniciamos el web service.

A screenshot of a Windows desktop environment. In the center is a terminal window titled 'Ubuntu@CR2019630376:~'. The terminal shows a series of commands being run to start an Apache Tomcat 8.5.72 web service on an Ubuntu system. The commands include navigating to the 'webapps' directory, changing to root privileges, starting the service, and setting environment variables like CATALINA_HOME and JAVA_HOME. The output ends with 'Tomcat started.' Below the terminal, a code editor window titled 'java.net.Socket' displays Java code related to network sockets. At the bottom of the screen is the Windows taskbar with various pinned icons.

```
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ cd webapps
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ ls
ROOT Servicio.war
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72/webapps$ cd ..
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ sh $CATALINA_HOME/bin/catalina.sh start
sh: 0: Can't open /bin/catalina.sh
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ sh $CATALINA_HOME/bin/catalina.sh stop
sh: 0: Can't open /bin/catalina.sh
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ pwd
/home/ubuntu
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ export CATALINA_HOME=/home/ubuntu/apache-tomcat-8.5.72
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Ubuntu@CR2019630376:~/apache-tomcat-8.5.72$ sh $CATALINA_HOME/bin/catalina.sh start
Using CATALINA_BASE: /home/ubuntu/apache-tomcat-8.5.72
Using CATALINA_HOME: /home/ubuntu/apache-tomcat-8.5.72
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-8.5.72/temp
Using JRE_HOME: /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH: /home/ubuntu/apache-tomcat-8.5.72/bin/bootstrap.jar:/home/ubuntu/apache-tomcat-8.5.72/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
Ubuntu@CR2019630376:~$
```

Compilación y ejecución del cliente

En la siguiente captura podemos ver como compilamos y ejecutamos el cliente, en este caso hay que tener en cuenta la consideración de que utilizaremos el JAR para GSON y que ocuparemos una instancia de la clase AdaptadorGsonBase64.java

A screenshot of a Windows desktop environment. In the center is a terminal window titled 'Símbolo del sistema - java -cp gson-2.3.1.jar. ClienteREST.java'. The terminal shows the compilation of 'ClienteREST.java' using 'javac -cp gson-2.3.1.jar. ClienteREST.java' and then the execution of the compiled class using 'java -cp gson-2.3.1.jar;. ClienteREST.java'. The code editor window on the left shows the Java code for the 'ClienteREST' class, which interacts with a 'AdaptadorGsonBase64' instance to manage user operations like Alta, Consulta, Borrar, and Salir. The code editor has syntax highlighting for Java and shows line numbers from 244 to 265. The Windows taskbar is visible at the bottom.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    usuario usuario;
    int id_usuario;
    char opc = ' ';
    while(opc != 'a') {
        System.out.println("Alta usuario");
        System.out.println("Consulta usuario");
        System.out.println("Borra usuario");
        System.out.println("Salir");
        System.out.println("Opción: ");
        System.out.print("Opción: ");
        opc = sc.next().charAt(0);
        sc.nextLine();
        System.out.println("Opción: " + opc);
        switch(opc) {
            case 'a':
                alta_usuario(usuario);
                break;
            case 'c':
                consultar_usuario(usuario);
                break;
            case 'b':
                borrar_usuario(usuario);
                break;
            case 's':
                salir();
                break;
            default:
                System.out.println("Opción no válida");
        }
    }
}
```

Prueba no. 1: Dar de alta un usuario.

Posteriormente daremos de alta a un nuevo usuario como se puede ver en la siguiente captura de pantalla y se nos regresa el ID del nuevo usuario creado.

The screenshot shows a Sublime Text window with two tabs open. The active tab contains Java code for a REST client, specifically for managing users. The code includes a menu system with options like Alta usuario, Consulta usuario, Borra usuario, and Salir. It also handles user input for email, name, and phone number, and outputs the created user ID. The second tab shows the command line interface where the Java code is run, displaying the output of the program's interaction with the user.

```
C:\Users\tdwda\OneDrive\Escritorio\Tarea7\ClienteREST.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Go >
Símbolo del sistema - java -cp gson-2.3.1.jar; ClienteREST.java

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Usuario usuario;
    int id_usuario;
    char opc = 'a';
    String nombre, apellido_paterno, apellido_materno;
    String email, telefono;
    String genero;
    String fecha_nacimiento;
    String respuesta;

    System.out.println("Opción: a");
    while(opc != 'b') {
        System.out.println("Introduzca el e-mail: David_Oaxaca11@gmail.com");
        email = sc.nextLine();
        System.out.println("Introduzca el nombre: David Arturo");
        nombre = sc.nextLine();
        System.out.println("Introduzca el apellido paterno: Oaxaca");
        apellido_paterno = sc.nextLine();
        System.out.println("Introduzca el apellido materno: Perez");
        apellido_materno = sc.nextLine();
        switch(opc) {
            case 'a':
                System.out.println("Introduzca la fecha de nacimiento: 2000-11-11");
                fecha_nacimiento = sc.nextLine();
                System.out.println("Introduzca el numero de telefono: 123456789");
                telefono = sc.nextLine();
                System.out.println("Introduzca el genero: M");
                genero = sc.nextLine();
                System.out.println("Dando de alta al usuario (Al terminar se desplegara su ID)...");
                respuesta = sc.nextLine();
                System.out.println("ID del usuario creado: 9");
                id_usuario = 9;
                sc.nextLine();
                System.out.println("a. Alta usuario");
                System.out.println("b. Consulta usuario");
                System.out.println("c. Borra usuario");
                System.out.println("d. Salir");
                System.out.println("Opción: ");
                opc = sc.nextLine();
                break;
            case 'b':
        }
    }
}
```

Prueba no. 2: Consultar el usuario previamente dado de alta.

Al seleccionar la opción b y escribir el ID del usuario que se nos retornó podemos observar todos sus datos en pantalla como se puede observar en la siguiente captura.

The screenshot shows a Sublime Text window with the following details:

- Title Bar:** CAUsers\tdwda\OneDrive\Escritorio\Tarea7\ClienteREST.java - Sublime Text (UNREGISTERED)
- File Menu:** File Edit Selection Find View Got
- Code Area:** The code is a Java program for managing users. It uses a Scanner to read input from the user. The user can choose to add a new user ('a'), delete an existing user ('b'), or exit ('c'). If they choose to add a user, the program prompts for the user's ID, name, last name, mother's last name, date of birth, phone number, and gender. It then asks if the user wants to modify the data. The code includes several print statements to show the user information.
- Status Bar:** Line 301, Column 23
- Bottom Icons:** Standard Sublime Text icons for file operations like Open, Save, Find, and Preferences.

Prueba no. 3: Modificar algún dato del usuario.

Presionamos 's' después de consultar los datos del usuario para empezar a modificarlos, los campos que dejemos en blanco se quedaran como están, en este caso solo modificaremos el e-mail, el nombre y el teléfono.

```
C:\Users\tdwda\OneDrive\Escritorio\Tarea7\ClienteREST.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Go! Símbolo del sistema - java -cp gson-2.3.1.jar; ClienteREST.java
< > untitled Client Apellido paterno: Perez
225 public static void main(String[] args) {
226     Scanner sc = new Scanner(System.in);
227     Usuario usuario;
228     int id_usuario;
229     char opc;
230     while(opc != 'b') {
231         System.out.println("Apellido paterno: Perez");
232         System.out.println("Apellido materno: Oaxaca");
233         System.out.println("Fecha de nacimiento: 2000-11-11");
234         System.out.println("Telefono: 123456789");
235         System.out.println("Genero: M");
236         System.out.println("*****");
237         System.out.println("Desea modificar los datos del usuario? (s/n):");
238         System.out.println("Introduzca los siguientes datos:");
239         System.out.println("Introduzca el e-mail: David_Oax11@gmail.com");
240         System.out.println("Introduzca el nombre: David");
241         switch(opc) {
242             case 's':
243                 System.out.println("Introduzca el apellido paterno:");
244                 sc.nextLine();
245                 System.out.println("Introduzca el apellido materno:");
246                 sc.nextLine();
247                 System.out.println("Introduzca la fecha de nacimiento:");
248                 sc.nextLine();
249                 System.out.println("Introduzca el numero de telefono: 987654321");
250                 sc.nextLine();
251                 System.out.println("Introduzca el genero:");
252                 sc.nextLine();
253                 System.out.println("Guardando modificaciones...");
254                 System.out.println("El usuario ha sido modificado");
255                 sc.nextLine();
256                 System.out.println("a. Alta usuario");
257                 System.out.println("b. Consulta usuario");
258                 System.out.println("c. Borra usuario");
259                 System.out.println("d. Salir");
260                 System.out.println("Opción: ");
261                 opc = sc.nextLine();
262                 break;
263             case 'b':
264                 break;
265         }
266     }
267 }
268 
```

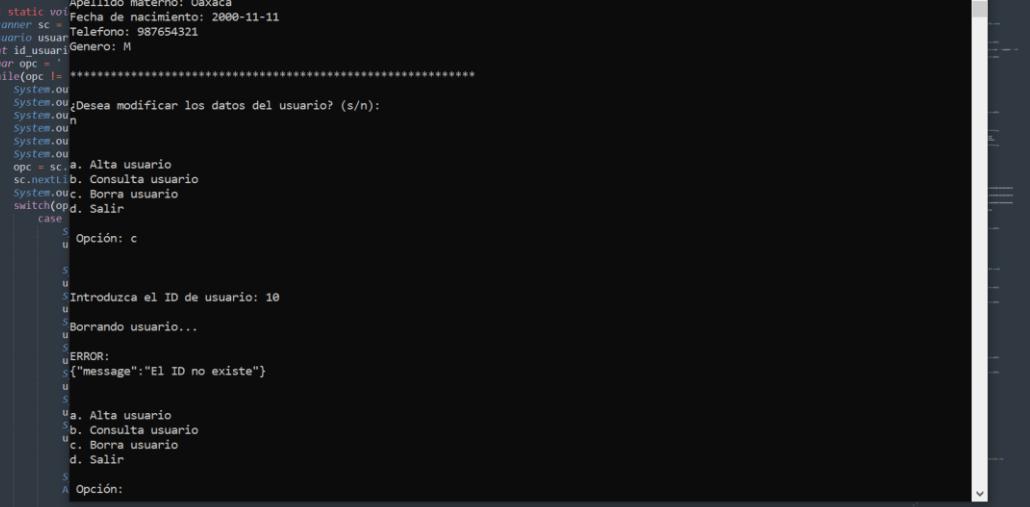
Prueba no. 4: Consultar el usuario modificado, para verificar que la modificación se realizó.

Podemos observar en la siguiente pantalla que todos los cambios que realizamos se guardaron correctamente y la consulta trae consigo esos cambios.

```
C:\Users\tdwda\OneDrive\Escritorio\Tarea7\ClienteREST.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Go! Símbolo del sistema - java -cp gson-2.3.1.jar; ClienteREST.java
< > untitled Client Introduzca el numero de telefono: 987654321
225 public static void main(String[] args) {
226     Scanner sc = new Scanner(System.in);
227     Usuario usuario;
228     int id_usuario;
229     char opc;
230     while(opc != 'b') {
231         System.out.println("Introduzca el genero:");
232         sc.nextLine();
233         System.out.println("Guardando modificaciones...");
234         System.out.println("El usuario ha sido modificado");
235         sc.nextLine();
236         System.out.println("a. Alta usuario");
237         System.out.println("b. Consulta usuario");
238         System.out.println("c. Borra usuario");
239         System.out.println("d. Salir");
240         System.out.println("Opción: ");
241         opc = sc.nextLine();
242         switch(opc) {
243             case 's':
244                 System.out.println("Introduzca el ID de usuario: 9");
245                 sc.nextLine();
246                 System.out.println("Realizando consulta...");
247                 sc.nextLine();
248                 System.out.println("*****");
249                 System.out.println("E-mail: David_Oax11@gmail.com");
250                 System.out.println("Nombre: David");
251                 System.out.println("Apellido paterno: Perez");
252                 System.out.println("Apellido materno: Oaxaca");
253                 System.out.println("Fecha de nacimiento: 2000-11-11");
254                 System.out.println("Telefono: 987654321");
255                 System.out.println("Genero: M");
256                 System.out.println("*****");
257                 sc.nextLine();
258             case 'b':
259                 break;
260             case 's':
261                 System.out.println("Desea modificar los datos del usuario? (s/n):");
262                 sc.nextLine();
263                 break;
264             case 'b':
265                 break;
266         }
267     }
268 }
269 
```

Prueba no. 5: Intentar borrar un usuario que no existe.

En este caso intentaremos borrar el usuario con ID = 10, este es un usuario que aún no ha sido creado y cómo podemos observar, nos despliega un mensaje de error.



```
C:\Users\tdwda\OneDrive\Escritorio\Tarea7\ClienteREST.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Go
Símbolo del sistema - java -cp gson-2.3.1.jar; ClienteREST.java
ClientApellido paterno: Perez
Apellido materno: Oaxaca
Fecha de nacimiento: 2000-11-11
Scanner sc = new Scanner(System.in);
Usuario usuario;
int id_usuario;
Genero: M
char op;
while(op != 'q') {
    System.out.println("Desea modificar los datos del usuario? (s/n):");
    System.out.println("a. Alta usuario");
    System.out.println("b. Consulta usuario");
    System.out.println("c. Borra usuario");
    System.out.println("d. Salir");
    switch(op) {
        case 'a':
            sc.nextLine();
            System.out.print("Introduzca el ID de usuario: ");
            id_usuario = sc.nextInt();
            sc.nextLine();
            System.out.print("Borrando usuario...");
            sc.nextLine();
            System.out.println("ERROR:");
            System.out.println("message": "El ID no existe");
            break;
        case 'b':
            sc.nextLine();
            System.out.println("a. Alta usuario");
            System.out.println("b. Consulta usuario");
            System.out.println("c. Borra usuario");
            System.out.println("d. Salir");
            System.out.print("Opción: ");
            op = sc.nextLine().charAt(0);
            break;
        case 'c':
            sc.nextLine();
            System.out.println("Borrando usuario...");
            sc.nextLine();
            System.out.println("ERROR:");
            System.out.println("message": "El ID no existe");
            break;
        case 'd':
            System.out.println("Adios!");
            break;
    }
}
Line 301, Column 23
```

Prueba no. 6: Borrar el usuario dado de alta.

Nuevamente presionamos la opción de borrar pero esta vez escribimos el 9 como ID, este si existe y será borrado como se puede observar en la captura.

The screenshot shows a Sublime Text window with two tabs: 'unfiled' and 'Cliente'. The 'Cliente' tab contains Java code for a client application. The code uses a Scanner to read user input and a switch statement to handle options. It interacts with a REST API using Gson. The terminal window shows the execution of the Java code and its interaction with the user.

```
File Edit Selection Find View Go < > unfiled Cliente

225 public static void main(String[] args) {
226     Scanner sc = new Scanner(System.in);
227     Usuario usur;
228     int id_usuar;
229     char op = ' ';
230     String message;
231     while(op != 'q') {
232         System.out.println("Introduzca el ID de usuario: ");
233         id_usuar = sc.nextInt();
234         if(sc.hasNextLine()) {
235             System.out.println("ERROR: " + message);
236             System.out.println("El ID no existe");
237             System.out.println("Alta usuario");
238             sc.nextLine();
239             System.out.println("Consulta usuario");
240             switch(op) {
241                 case 'c': Borra usuario
242                 case 'd': Salir
243                 case 's': Opcion: s
244                 case 'u': Opcion: u
245                 default: System.out.println("Opcion invalida");
246             }
247         }
248         System.out.println("Introduzca el ID de usuario: ");
249         id_usuar = sc.nextInt();
250         System.out.println("Borrando usuario...");
251         sc.nextLine();
252         System.out.println("El usuario ha sido borrado");
253         System.out.println("Alta usuario");
254         System.out.println("Consulta usuario");
255         System.out.println("Borra usuario");
256         System.out.println("Salir");
257         System.out.println("Opcion: ");
258         op = sc.next().charAt(0);
259         switch(op) {
260             case 'a': Opcion: a
261             case 'b': Opcion: b
262             case 'c': Opcion: c
263             case 'd': Opcion: d
264             case 's': Opcion: s
265             case 'u': Opcion: u
266             case 'q': break;
267         }
268     }
269 }

Line 301, Column 23
```

En esta captura podemos ver como si intentamos consultar al usuario que borramos previamente este ya no aparece y nos dice que su ID no existe.

The screenshot shows a Windows desktop environment. In the center is a terminal window titled "Símbolo del sistema" with the command "java -cp gson-2.3.1.jar, ClienteREST.java". The terminal output is as follows:

```
Line 301, Column 23
C:\Users\tdwda\OneDrive\Escritorio\Tarea7>java -cp gson-2.3.1.jar, ClienteREST.java
File Edit Selection Find View Gof Client
        Opción: c
225  public static void main(String[] args) {
226      Scanner sc = new Scanner(System.in);
227      Usuario usuario;
228      int id_usuario;
229      char opc;
230      while(true) {
231          System.out.println("Introduzca el ID de usuario: ");
232          id_usuario = sc.nextInt();
233          System.out.println("Borrando usuario...");
234          System.out.println("El usuario ha sido borrado");
235          System.out.println("Opción: a. Alta usuario");
236          System.out.println("b. Consulta usuario");
237          System.out.println("c. Borra usuario");
238          System.out.println("d. Salir");
239          switch(opc) {
240              case 'a':
241                  System.out.println("Opción: b");
242              case 'b':
243                  System.out.println("Introduzca el ID de usuario: ");
244                  id_usuario = sc.nextInt();
245                  System.out.println("Realizando consulta...");
246                  System.out.println("ERROR: ");
247                  System.out.println("{"message": "El ID no existe"}");
248              case 'c':
249                  System.out.println("a. Alta usuario");
250                  System.out.println("b. Consulta usuario");
251                  System.out.println("c. Borra usuario");
252                  System.out.println("d. Salir");
253                  System.out.println("Opción: d");
254                  break;
255              case 'd':
256                  System.out.println("Adios!");
257                  break;
258              default:
259                  System.out.println("Opción incorrecta");
260                  break;
261          }
262      }
263  }
```

The terminal window has a status bar at the bottom showing "Line 301, Column 23", "Tab Size: 4", and "Java". Below the terminal is a taskbar with various icons and the system tray showing the date and time as "05/11/2021 03:17 p.m." and the temperature as "21°C".

Finalmente salimos del programa.

The screenshot shows a Windows desktop environment. In the center is a terminal window titled "Símbolo del sistema" with the command "java -cp gson-2.3.1.jar, ClienteREST.java". The terminal output is as follows:

```
Line 301, Column 23
C:\Users\tdwda\OneDrive\Escritorio\Tarea7>java -cp gson-2.3.1.jar, ClienteREST.java
File Edit Selection Find View Gof Client
        Introduzca el ID de usuario: 9
225  public static void main(String[] args) {
226      Scanner sc = new Scanner(System.in);
227      Usuario usuario;
228      int id_usuario;
229      char opc;
230      while(true) {
231          System.out.println("Introduzca el ID de usuario: ");
232          id_usuario = sc.nextInt();
233          System.out.println("Borrando usuario...");
234          System.out.println("El usuario ha sido borrado");
235          System.out.println("Opción: a. Alta usuario");
236          System.out.println("b. Consulta usuario");
237          System.out.println("c. Borra usuario");
238          System.out.println("d. Salir");
239          switch(opc) {
240              case 'a':
241                  System.out.println("Introduzca el ID de usuario: ");
242                  id_usuario = sc.nextInt();
243                  System.out.println("Realizando consulta...");
244                  System.out.println("ERROR: ");
245                  System.out.println("{"message": "El ID no existe"}");
246              case 'c':
247                  System.out.println("a. Alta usuario");
248                  System.out.println("b. Consulta usuario");
249                  System.out.println("c. Borra usuario");
250                  System.out.println("d. Salir");
251                  System.out.println("Opción: d");
252                  break;
253              case 'd':
254                  System.out.println("Adios!");
255                  break;
256              default:
257                  System.out.println("Opción incorrecta");
258                  break;
259          }
260      }
261  }
```

The terminal window has a status bar at the bottom showing "Line 301, Column 23", "Tab Size: 4", and "Java". Below the terminal is a taskbar with various icons and the system tray showing the date and time as "05/11/2021 03:17 p.m." and the temperature as "21°C".

Conclusiones

Esta práctica reforzó bastante lo que sabía sobre la implementación de un servicio estilo REST y me ayudo a entender como es la construcción de un cliente y como trabaja en conjunto con un servidor, por ejemplo, al hacer una petición POST, es interesante como se pasan los parámetros de un objeto usando JSON y como el servidor los recibe, mucho de lo visto en clases anteriores se puso en práctica para implementarse en conjunto y profundizo con lo que previamente habíamos hecho en la práctica anterior para complementar lo visto respecto a los servicios web.