



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO

Desarrollo de Sistemas Distribuidos



# Tarea 7

## Desarrollo de un cliente para un servicio web estilo REST

**PROFESOR:** Pineda Guerrero Carlos

**Alumna:** Ramirez Galindo Karina

**GRUPO:** 4CV11

**Fecha de realización:** 12/04/2022

**Fecha de entrega:** 13/04/2022

## Contenido

Introducción.....	1
Conceptos básicos de Servicios web.....	1
Servicio web estilo REST .....	2
.....	3
Desarrollo .....	4
Creación de la máquina virtual.....	4
Configuración de la máquina virtual.....	16
Compilar, empacar y desplegar el servicio web .....	19
Código .....	22
Pruebas .....	27
Concusiones.....	36
Referencias .....	36

## Introducción

### Conceptos básicos de Servicios web

En el documento Web Services Architecture (2004) del World Wide Web Consortium (W3C) define un servicio web como:

*“Un sistema de software diseñado para soportar la interacción interoperable de máquina-a-máquina sobre una red. Este cuenta con una interface descrita en un formato el cual puede ser procesado por una computadora (específicamente WSDL). Otros sistemas interactúan con el servicio web en una manera prescrita por su descripción usando mensajes SOAP, típicamente transportados usando HTTP con una serialización XML en conjunción con otros estándares relativos a la Web”.*

Un servicio web es un concepto abstracto que se implementa mediante un agente concreto.

- Un **agente** es el software o hardware que envía y recibe mensajes.
- El **servicio** es el recurso caracterizado por la funcionalidad que provee. Un servicio web no cambia aun cuando cambie el agente, es decir, la funcionalidad es independiente de la implementación de ésta.
- El propósito de un servicio web es proveer cierta funcionalidad a nombre de su propietario (una persona o una organización).
- La **entidad proveedora** es aquella persona u organización que provee un agente que implementa un determinado servicio.
- Una **entidad solicitante** es una persona u organización que desea hacer uso del servicio mediante un **agente solicitante** (también llamado *solicitante del servicio*) que intercambia mensajes con el **agente proveedor** (también llamado *proveedor del servicio*).
- En la mayoría de los casos el agente solicitante es el que inicia la comunicación con el agente proveedor, aunque no siempre es así, no obstante, se sigue llamando agente solicitante, aunque no sea el que inicia la comunicación.
- La **semántica** de un servicio web es la expectativa compartida sobre el comportamiento del servicio, en particular el comportamiento en respuesta a los mensajes que recibe.
- Se le llama **contrato** al acuerdo entre la entidad solicitante y la entidad proveedora. Un contrato puede ser explícito o implícito, escrito u oral, establecido entre las personas y/o las computadoras, legal o informal.

Hay dos tipos de contratos:

- 1) La **descripción del servicio** es el contrato que gobierna la mecánica de interacción con un servicio en particular

- 2) La semántica del servicio es el contrato que gobierna el significado y propósito de la interacción. Sin embargo, puede haber contratos "híbridos" que incluyan elementos de descripción y elementos de semántica.

## Participación en un servicio web

Una entidad solicitante puede participar de un servicio web de diferentes maneras. La siguiente figura muestra el proceso general de participación en un servicio web.

1. Las entidades solicitante y proveedora se conocen una a la otra, o por lo menos una conoce a la otra.
2. Las entidades acuerdan la descripción (WSD: *Web Service Description*) y semántica del servicio.
3. La descripción y la semántica son implementadas por el agente solicitante y el agente proveedor.
4. Los agentes solicitante y proveedor intercambian mensajes.

## Servicio web estilo REST

REST (*REpresentational State Transfer*) define un conjunto de principios arquitectónicos para la creación de servicios web. REST fue presentado por Roy Fielding el año 2000 en su tesis doctoral "Architectural Styles and the Design of Network-based Software Architectures".

El diseño de servicios web estilo REST sigue cuatro principios:

- Utilizar métodos HTTP de forma explícita.
  - Un servicio web utiliza los métodos de HTTP para crear un recurso (POST), leer (GET), cambiar el estado o actualizar un recurso (PUT), y borrar un recurso (DELETE).
- Los servicios son "sin estado" (*stateless*).
  - Los clientes de servicios web estilo REST deben enviar peticiones completas e independientes, es decir, las peticiones deben incluir todos los datos que permitan completar el servicio, sin la necesidad de guardar un estado entre peticiones.
- Los URIs representan una estructura de directorios.
  - Los URIs (*Uniform Resource Identifier*) deben ser intuitivos y auto-explicados. Un URI es una jerarquía que corresponde a la estructura de los servicios web definidos en la empresa.
- Se transfiere XML, JSON o ambos.

- Los recursos que provee un servicio web pueden ser documentos, imágenes, videos y en general objetos. La representación de objetos mediante XML o JSON es fácil e independiente de la plataforma. [1]

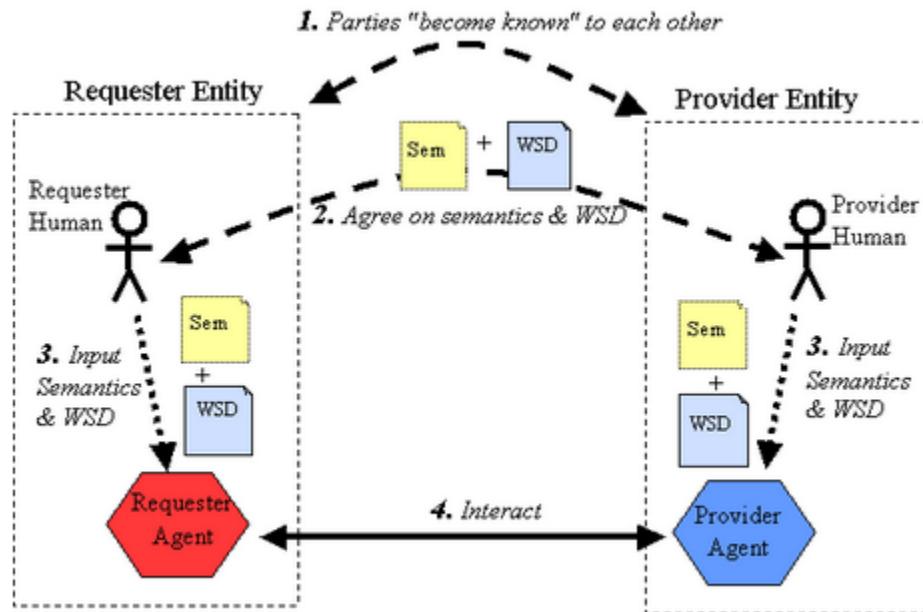


Figura 1. Servicio web

## Desarrollo

Se deberá crear una máquina virtual Ubuntu con 1 GB de RAM y disco HD. El nombre de la máquina virtual deberá ser la letra "X" y el número de boleta del alumno, para este caso, el número de boleta del alumno es 2015170921, entonces la máquina virtual deberá llamarse: X2015170921.

### Creación de la máquina virtual

Se utilizará una máquina virtual creada a partir de la imagen obtenida la tarea anterior

#### Creación paso a paso:

Ingresar al portal de Azure en la siguiente URL:

<https://azure.microsoft.com/es-mx/features/azure-portal/>

1. Ir a la sección "Todos los recursos" en el portal de Azure como se muestra en la Figura 2.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with links like 'Inicio - Microsoft Azure', 'SISDIS-4CV11: Tarea 7. Desarrollo', and a search bar. Below the navigation bar, the main content area is titled 'Servicios de Azure'. In the center, a box labeled 'Todos los recursos' is highlighted with a red border. This box contains a grid icon, a 'Crear' button, and a '...' button. To the left of this box, there's a 'Recursos recientes' section listing 'R2015170921-image' and 'R2015170921'. To the right, there are links for 'App Services', 'Cuentas de almacenamiento', 'SQL Database', and 'Todos los servicios'. At the bottom of the main content area, there are sections for 'Herramientas' (Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, Administración de costos) and 'Vínculos útiles' (Documentación técnica, Herramientas de migración de Azure, Servicios de Azure, Actualizaciones recientes de Azure). On the far right, there's an 'Aplicación móvil de Azure' section with links for the App Store and Google Play. The bottom of the screen shows the Windows taskbar with various pinned icons and system status indicators.

Figura 2. Sección "Todos los recursos" en el portal de Azure

2. Seleccionar la imagen de la máquina virtual como se muestra en la Figura 3.

Inicio >

## Todos los recursos

Microsoft Azure Buscar recursos, servicios y documentos (G+ /)

+ Crear Administrar vista Actualizar Exportar a CSV Abrir consulta Asignar etiquetas Eliminar Comentarios

Filtrar por cualquier ca... Suscripción == todo Grupo de recursos == todo Tipo == todo Ubicación == todo Agregar filtro

0 Recursos no seguros Sin agrupar Vista de lista

Nombre	Tipo	Grupo de recursos	Ubicación	Suscripción
R2015170921-image	Imagen	R2015170921	East US	Azure for Students

< Anterior Página 1 de 1 Siguiente > Mostrando de 1 a 1 de 1 registros.

Windows Start button, Taskbar icons (File Explorer, Edge, Mail, etc.), Weather (20°C, Muy despejado), Date (13/04/2022), Time (12:01 a.m.)

Figura 3. Selección de la imagen de la máquina virtual.

3. Seleccionar la opción "+Crear máquina virtual" como se observa en la Figura 4.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar titled 'Todos los recursos' (All resources) under 'Instituto Politecnico Nacional'. The main area is titled 'R2015170921-image' and shows 'Imagen' (Image). A red box highlights the '+ Crear máquina virtual' (Create virtual machine) button in the top right corner of the main content area. Below this, there are sections for 'Información general' (General information), 'Configuración' (Configuration), 'Automation', and 'Disco del SO' (Operating system disk). At the bottom, there's a navigation bar with icons for various services like Mail, File Explorer, and Edge, along with a taskbar showing the date and time.

Figura 4. Opción “Crear” máquina virtual.

4. Seleccionar el grupo de recursos dónde se creará la máquina virtual.
5. Ingresar el nombre de la máquina virtual.  
(véase la Figura 5).

Cree una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración.

[Más información](#)

**Detalles del proyecto**

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* ⓘ Azure for Students

Grupo de recursos \* ⓘ R2015170921 [Crear nuevo](#)

**Detalles de instancia**

Nombre de máquina virtual \* ⓘ X2015170921

Región ⓘ (US) East US

Opciones de disponibilidad ⓘ Zona de disponibilidad

Zona de disponibilidad \* ⓘ Zonas 1

Ahora puede seleccionar varias zonas. Si selecciona varias zonas, se creará una VM por zona. [Más información](#)

Tipo de seguridad ⓘ Estándar

Imagen \* ⓘ R2015170921-image - Gen2

[Revisar y crear](#) [< Anterior](#) [Siguiente: Discos >](#)

Figura 5. Datos básicos para la creación de la máquina virtual en Azure.

6. Seleccionar el tamaño de la máquina virtual.
7. Seleccionar el tipo de autenticación (Clave pública SSH o Contraseña). En su caso, ingresar el usuario y contraseña.  
(véase la Figura 6).

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The current step is 'Set the administrator password'. The form includes fields for 'Nombre de usuario' (KarinaRG), 'Contraseña' (a masked password), and 'Confirmar contraseña' (also a masked password). The 'Tipo de autenticación' is set to 'Contraseña'. Under 'Reglas de puerto de entrada', the 'Puertos de entrada públicos' are set to 'Permitir los puertos seleccionados' (SSH port 22 is selected). A warning message states: 'Este permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.' In the 'Licencias' section, the 'Tipo de licencia' dropdown is empty. At the bottom, there are buttons for 'Revisar y crear' (Review + Create) and 'Siguiente: Discos >' (Next: Disks >).

Figura 6. Configuración de "Contraseña" de la máquina virtual en Azure.

8. Dar clic en el botón "Siguiente: Discos >"
9. Seleccionar el tipo de disco del sistema operativo (p.e. HDD estándar).  
(véase la Figura 7).

**Crear una máquina virtual**

**Discos**

Las máquinas virtuales de Azure tienen un disco de sistema operativo y un disco temporal para el almacenamiento a corto plazo. Puede asociar discos de datos adicionales. El tamaño de la máquina virtual determina el tipo de almacenamiento que puede usar y la cantidad de datos que permiten los discos. [Más información](#)

**Opciones de disco**

Tipo de disco del sistema operativo \* ⓘ **HDD estándar (almacenamiento con redundancia local)**

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%.

Eliminar con VM ⓘ

Cifrado en el host ⓘ

ⓘ El cifrado en el host no está registrado para la suscripción seleccionada. [Más información sobre cómo habilitar esta característica](#)

Tipo de cifrado \* ⓘ **(Predeterminado) Cifrado en reposo con una clave administrada por la plata...**

Habilitar compatibilidad con Ultra Disks ⓘ

**Discos de datos para X2015170921**

Puede agregar y configurar discos de datos adicionales para su máquina virtual o asociar discos existentes. Esta máquina virtual también incluye un disco temporal.

LUN	Nombre	Tamaño...	Tipo de disco	Almacena...	Eliminar con VM ⓘ

**Revisar y crear** < Anterior Siguiente: Redes >

Figura 7. Configuración de "Discos" de la máquina virtual en Azure.

10. Dar click en el botón "Siguiente: Redes>" (véase la Figura 8).

En esta pestaña no es necesario cambiar nada, se dejan los valores por defecto.

Datos básicos Discos Redes Administración Opciones avanzadas Etiquetas Revisar y crear

Configure la tarjeta de interfaz de red (NIC) a fin de definir la conectividad de red para la máquina virtual. Puede controlar los puertos y la conectividad entrante y saliente con reglas de grupos de seguridad o bien aplicar una solución de equilibrio de carga ya existente. [Más información](#)

**Interfaz de red**

Al crear una máquina virtual, se crea una interfaz de red automáticamente.

Red virtual \* (nuevo) R2015170921-vnet [Crear nuevo](#)

Subred \* (nuevo) default (10.0.0.0/24)

IP pública (nuevo) X2015170921-ip [Crear nuevo](#)

Grupo de seguridad de red de NIC Ninguno  Básico  Opciones avanzadas

Puertos de entrada públicos \* Ninguno  Permitir los puertos seleccionados

Seleccionar puertos de entrada \* SSH (22)

**Advertencia:** Esto permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

**Revisar y crear** < Anterior Siguiente: Administración >

Figura 8. Configuración de "Redes" de la máquina virtual en Azure.

11. Dar click en el botón "Siguiente: Administración>"

En el campo "Diagnóstico de arranque" seleccionar "Desactivado". (véase la Figura 9).

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The top navigation bar includes links for 'Crear una máquina virtual - Micr...', 'SISDIS-4CV11: Tarea 7. Desarroll...', and a search bar. The main content area is titled 'Crear una máquina virtual' and has a sub-section 'Administración'. The 'Administración' tab is selected, showing configuration options for monitoring, identity, and Azure AD. A note at the bottom states: '⚠ Esta imagen no admite el inicio de sesión con Azure AD.' (This image does not support Azure AD sign-in). At the bottom of the page are buttons for 'Revisar y crear' (Review + Create) and navigation links '< Anterior' and 'Siguiente: Opciones avanzadas >'.

Figura 9. Pestaña “Administración” de la máquina virtual en Azure.

12. En la pestaña de “opciones avanzadas” todos los valores se quedan por defecto.

Dar click en el botón "Revisar y crear". (véase la Figura 10).

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The top navigation bar includes links for 'Crear una máquina virtual - Micr...', 'SISDIS-4CV11: Tarea 7. Desarroll...', and 'https://portal.azure.com/#@correo.ipn.mx/resource/subscriptions/f5424dc8-20c3-44ed-9fdb-22eb0237d401/resourceGrou...'. The main title is 'Crear una máquina virtual'. The 'Opciones avanzadas' tab is selected.

**Extensiones**  
Agregue configuración, agentes, scripts o aplicaciones adicionales mediante las extensiones de máquina virtual o cloud-init.

**Aplicaciones de máquina virtual (versión preliminar)**  
Las aplicaciones de máquina virtual contienen archivos de aplicación que se descargan de forma segura y confiable en la máquina virtual después de la implementación. Además de los archivos de aplicación, se incluyen un script de instalación y desinstalación en la aplicación. Después de la creación, puede añadir o quitar aplicaciones fácilmente en la máquina virtual.  
[Más información](#)

**Datos personalizados**  
Pase un script, un archivo de configuración u otros datos a la máquina virtual **mientras se aprovisiona**. Los datos se guardarán en la VM en una ubicación conocida. [Más información sobre los datos personalizados para las VM](#)

**Datos personalizados**: A large empty rectangular input field.

**Notificación**: Una nota indica: "La imagen debe tener un código para admitir el consumo de datos personalizados. Si la imagen admite cloud-init, este es..."

**Botones de navegación**: 'Revisar y crear' (en azul), '< Anterior' y 'Siguiente: Etiquetas >'.

Figura 10. Configuración de “opciones avanzadas” de la máquina virtual en Azure.

13. Dar clic en el botón "Crear"(véase la Figura 11).

Crear una máquina virtual - Microsoft Azure | SISDIS-4CV11: Tarea 7. Desarroll... | +

https://portal.azure.com/#@correo.ipn.mx/resource/subscriptions/f5424dc8-20c3-44ed-9fdb-22eb0237d401/resourceGroups/R2015170921/images

Google Redes Sociales ESCOM Diseño Redes 2 SISDIS Instru Arqui Web Azure Glow Up Mi unidad - Google... GitHub Otros favoritos

Microsoft Azure Buscar recursos, servicios y documentos (G+)

Iniciar sesión kramirezg1400@alumn... INSTITUTO POLITÉCNICO NACIONAL

Crear una máquina virtual

Validación superada

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.

R2015170921-image	Standard B1s
Imagen	1 vcpu, 1 GiB de memoria

Datos básicos

Suscripción	Azure for Students
Grupo de recursos	R2015170921
Nombre de máquina virtual	X2015170921
Región	East US
Opciones de disponibilidad	Zona de disponibilidad
Zona de disponibilidad	1
Tipo de seguridad	Estándar
Imagen	R2015170921-image - Gen2
Tamaño	Standard B1s (1 vcpu, 1 GiB de memoria)
Tipo de autenticación	Contraseña
Nombre de usuario	KarinaRG
Puertos de entrada públicos	SSH
Azul de acceso puntual	No

Crear < Anterior Siguiente > Descargar una plantilla para la automatización

20°C Muy despejado 12:05 a.m. 13/04/2022

Figura 11. Opción "Crear" máquina virtual en Azure.

14. Dar click a la campana de notificaciones (barra superior de la pantalla) para verificar que la máquina virtual se haya creado. (véase la Figura 12)

Inicio >

## CreateVm-R2015170921-image-20220413000209 | Información general

Buscar (Ctrl+ /) Eliminar Cancelar Volver a implementar Actualizar

Se completó la implementación

Nombre de implementación: CreateVm-R2015170921-image-20220413000209 Hora de inicio: 13/4/2022, 0:05:52  
Suscripción: Azure for Students Id. de correlación: 21780a01-97b4-405e-a2dc-98dd4d...  
Grupo de recursos: R2015170921

Detalles de implementación (Descargar)

Pasos siguientes

Ir al recurso

Cost Management  
Obtenga una notificación para dentro del presupuesto y evíe inesperados en su factura.  
Configurar alertas de costo

Microsoft Defender for Cloud  
Proteja sus aplicaciones e infraestructuras.  
Ir a Microsoft Defender for Cloud

Tutoriales gratuitos de Microsoft  
Comience a aprender hoy >

Trabajar con un experto  
Los expertos de Azure son proveedores de servicios que administran sus recursos en primera línea de soporte técnico.  
Buscar un experto de Azure

Figura 12. Creación exitosa de la máquina virtual en Azure.

15. Dar click en el botón "Ir al recurso". En la página de puede ver la dirección IP pública de la máquina virtual. Esta dirección puede cambiar cada vez que se apague y se encienda la máquina virtual. (véase la Figura 13).

Figura 13. Información general de la máquina virtual.

Para que los programas que ejecutan en la máquina virtual puedan recibir conexiones a través de un determinado puerto, es necesario crear una regla de entrada para el puerto. En este caso se utilizará el puerto 8080 para el protocolo TCP, para habilitarlo hay que seguir los siguientes pasos: (véase la figura 14).

1. Dar clic en "Redes".
2. Dar clic en el botón "Aregar regla de puerto de entrada".
3. En el campo "Intervalos de puertos de destino" ingresar: 8080
4. Seleccionar el protocolo: TCP
5. En el campo "Nombre" ingresar un nombre para la regla: Puerto\_8080

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various navigation options like 'Información general', 'Registro de actividad', 'Control de acceso (IAM)', 'Etiquetas', 'Diagnosticar y solucionar problemas', 'Configuración' (selected), 'Redes' (selected), 'Conectar', 'Discos', 'Tamaño', 'Seguridad', 'Recomendaciones de Advisor', 'Extensiones + aplicaciones', 'Entrega continua', 'Disponibilidad y escalado', 'Configuración', 'Identidad', 'Propiedades', 'Bloqueos', 'Operaciones', and 'Rastreo'. The main content area is titled 'Aregar regla de seguridad de entrada' (Add security rule) for 'x2015170921-nsg'. It shows a table of existing security rules:

Prioridad	Nombre	Puerto
300	SSH	22
65000	AllowVnetInBound	Cualquier puerto
65001	AllowAzureLoadBalancerInBound	Cualquier puerto
65500	DenyAllInBound	Cualquier puerto

To the right, there's a form to add a new rule for port 8080:

- Origen:** Any
- Intervalos de puertos de origen:** \*
- Destino:** Any
- Servicio:** Custom
- Intervalos de puertos de destino:** 8080
- Protocolo:** TCP (selected)
- Acción:** Permitir (selected)
- Prioridad:** 310
- Nombre:** Port\_8080
- Descripción:**

At the bottom right of the dialog are 'Agregar' (Add) and 'Cancelar' (Cancel) buttons.

Figura 14. Habilitación del puerto 8080.

## Configuración de la máquina virtual

Para conectarse a la máquina virtual se utiliza el programa ssh disponible en Windows, Linux y MacOS.

- Mediante ssh y por medio de Windows con la siguiente línea se establece la conexión utilizando la IP de la máquina virtual:

**KarinaRG@20.127.153.126**

- Y se introduce la contraseña de autenticación de Azure (véase la Figura 15).

```
KarinaRG@X2015170921:~ x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnologia PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Karina> ssh KarinaRG@20.127.153.126
KarinaRG@20.127.153.126's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1074-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Wed Apr 13 22:07:33 UTC 2022

System load: 0.36      Processes: 104
Usage of /: 6.8% of 28.90GB  Users logged in: 0
Memory usage: 38%          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

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

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Apr 13 21:41:48 2022 from 189.151.53.57
KarinaRG@X2015170921:~$
```

Figura 15. Acceso a la máquina virtual por ssh.

Revisamos que la maquina virtual tenga todos los elementos necesarios para el servicio web. (véase la Figura16).

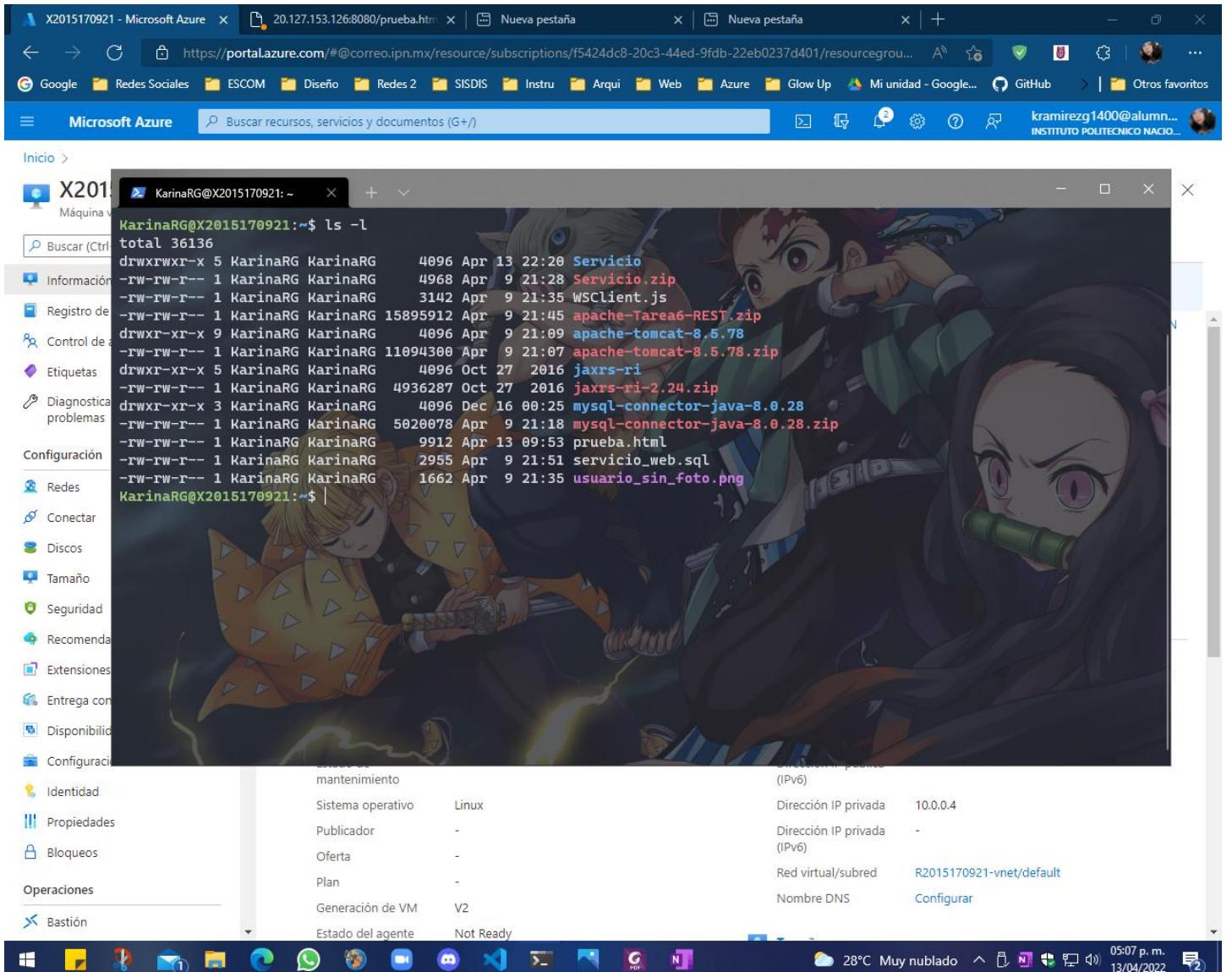


Figura 16. Recursos necesarios para trabajar en la máquina virtual.

Para eliminar el servicio web se deberá eliminar el archivo "Servicio.war" y el directorio "Servicio", en este 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. (véase la Figura 17).

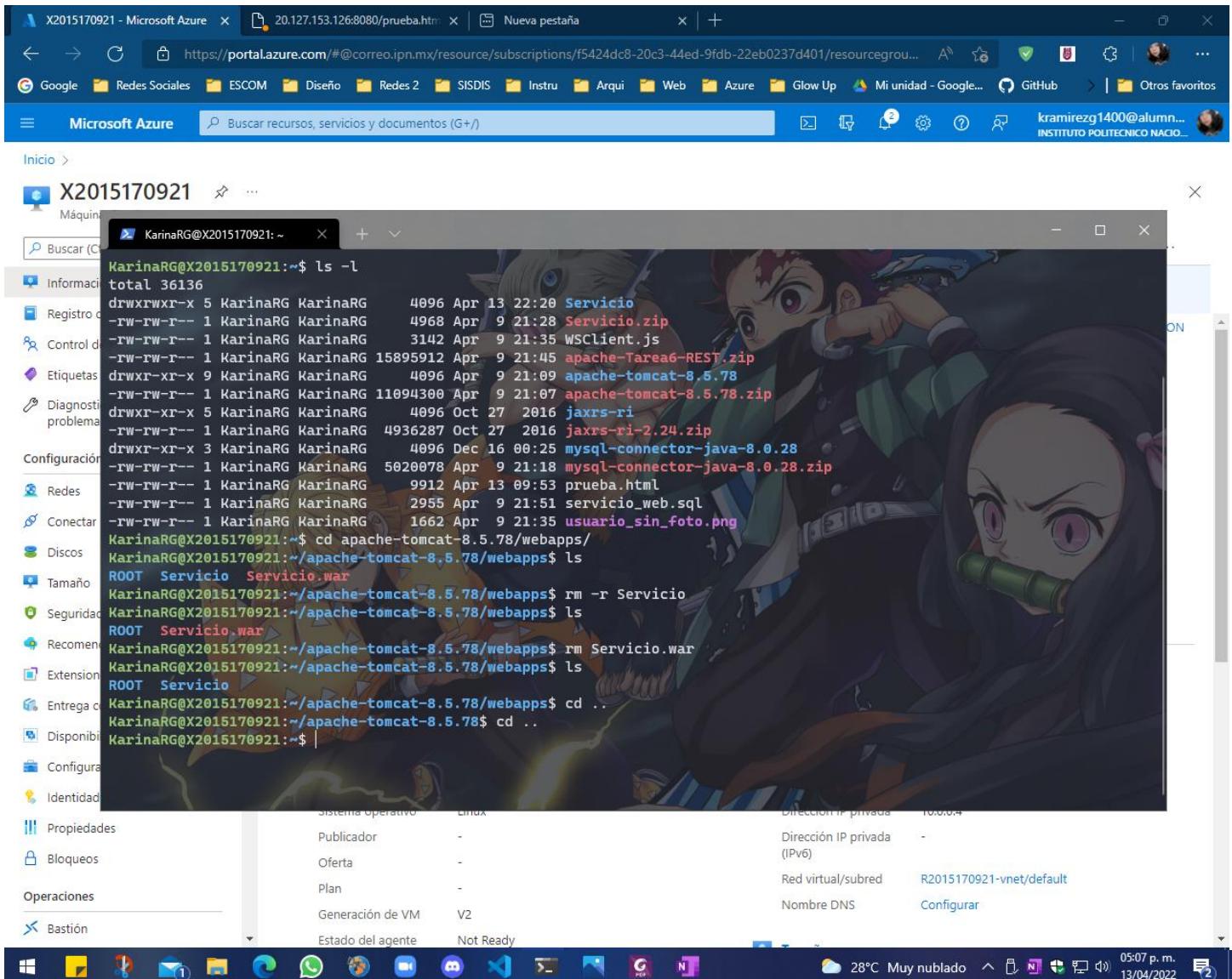


Figura 17. Eliminación del servicio web

### Compilar, empacar y desplegar el servicio web

1. Definir la variable de ambiente CATALINA\_HOME: export CATALINA\_HOME=*aquí va la ruta completa del directorio de Tomcat 8*
2. Cambiar al directorio dónde se desempacó el archivo **Servicio.zip** (en ese directorio se encuentra el directorio "negocio").
3. 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
```
4. 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

5. 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.  
(véase la Figura 18)

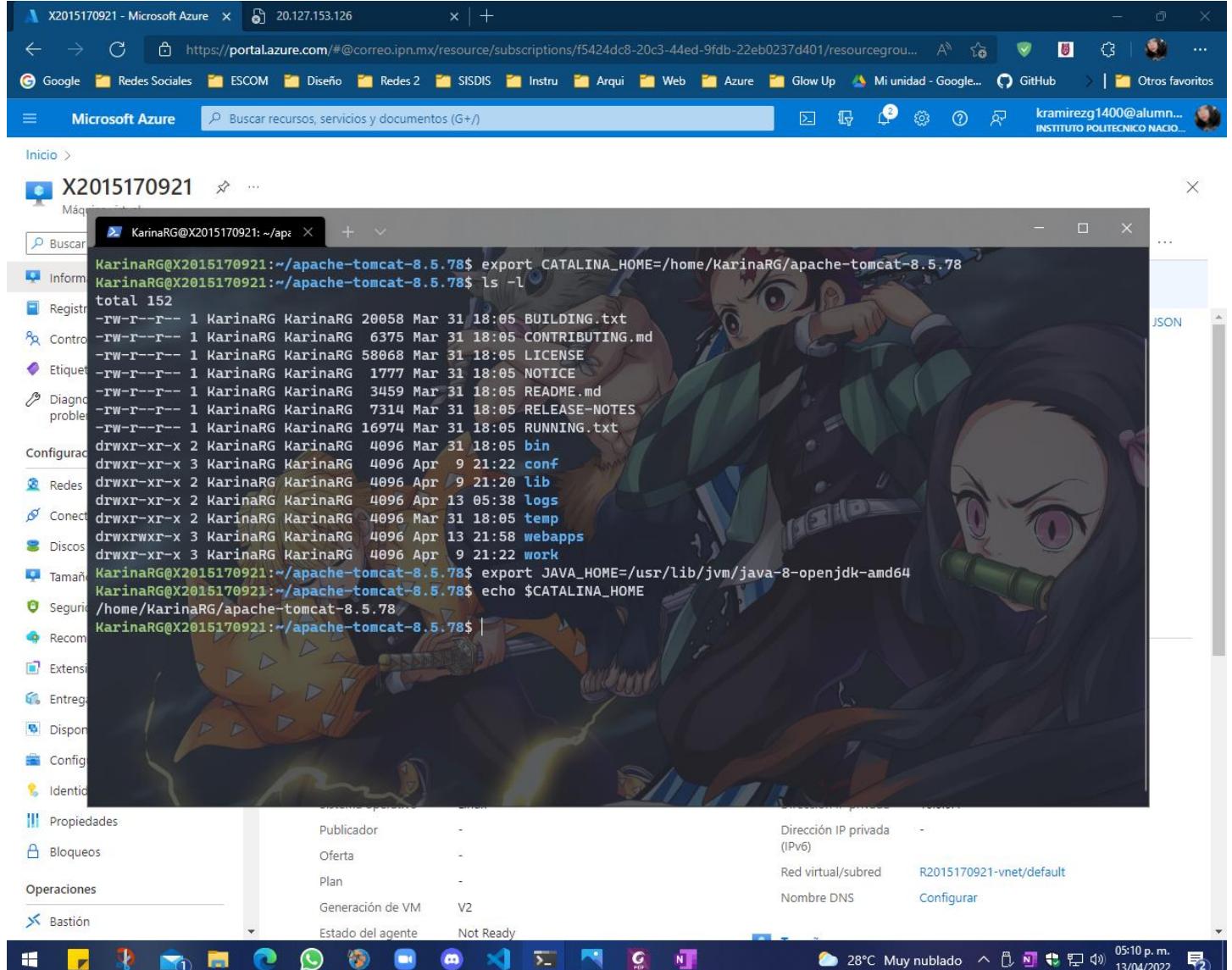


Figura 18. Despliegue del servicio web

El inicio del servicio web se muestra en la Figura 19.

The screenshot shows a Microsoft Azure desktop environment. In the center is a terminal window titled 'KarinaRG@X2015170921: ~ / apache-tomcat-8.5.78\$'. The terminal displays the following log output:

```
total 152
-rw-r--r-- 1 KarinaRG KarinaRG 20058 Mar 31 18:05 BUILDING.txt
-rw-r--r-- 1 KarinaRG KarinaRG 6375 Mar 31 18:05 CONTRIBUTING.md
-rw-r--r-- 1 KarinaRG KarinaRG 58068 Mar 31 18:05 LICENSE
-rw-r--r-- 1 KarinaRG KarinaRG 1777 Mar 31 18:05 NOTICE
-rw-r--r-- 1 KarinaRG KarinaRG 3459 Mar 31 18:05 README.md
-rw-r--r-- 1 KarinaRG KarinaRG 7314 Mar 31 18:05 RELEASE-NOTES
-rw-r--r-- 1 KarinaRG KarinaRG 16974 Mar 31 18:05 RUNNING.txt
drwxr-xr-x 2 KarinaRG KarinaRG 4096 Mar 31 18:05 bin
drwxr-xr-x 3 KarinaRG KarinaRG 4096 Apr  9 21:22 conf
drwxr-xr-x 2 KarinaRG KarinaRG 4096 Apr  9 21:20 lib
drwxr-xr-x 2 KarinaRG KarinaRG 4096 Apr 13 05:38 logs
drwxr-xr-x 2 KarinaRG KarinaRG 4096 Mar 31 18:05 temp
drwxrwxr-x 3 KarinaRG KarinaRG 4096 Apr 13 21:58 webapps
drwxr-xr-x 3 KarinaRG KarinaRG 4096 Apr  9 21:22 work
KarinaRG@X2015170921:~/apache-tomcat-8.5.78$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
KarinaRG@X2015170921:~/apache-tomcat-8.5.78$ echo $CATALINA_HOME
/home/KarinaRG/apache-tomcat-8.5.78
KarinaRG@X2015170921:~/apache-tomcat-8.5.78$ SH $CATALINA_HOME/bin/catalina.sh start
SH: command not found
KarinaRG@X2015170921:~/apache-tomcat-8.5.78$ sh $CATALINA_HOME/bin/catalina.sh start
Using CATALINA_BASE: /home/KarinaRG/apache-tomcat-8.5.78
Using CATALINA_HOME: /home/KarinaRG/apache-tomcat-8.5.78
Using CATALINA_TMPDIR: /home/KarinaRG/apache-tomcat-8.5.78/temp
Using JRE_HOME: /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH: /home/KarinaRG/apache-tomcat-8.5.78/bin/bootstrap.jar:/home/KarinaRG/apache-tomcat-8.5.78/bin/tom
cat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
KarinaRG@X2015170921:~/apache-tomcat-8.5.78$ |
```

The desktop interface includes a sidebar with various Azure service icons like Google, Redes Sociales, ESCOM, Diseño, Redes 2, SISDIS, Instru, Arqui, Web, Azure, Glow Up, Mi unidad - Google..., GitHub, and Otros favoritos. The taskbar at the bottom shows standard Windows icons for File Explorer, Task View, Start, and others, along with system status indicators like battery level, network, and date/time (05:11 p.m., 13/04/2022).

Figura 19. Inicio del servicio web

## Código

Cada alumno o alumna deberá desarrollar un programa Java consola (modo carácter) cliente del servicio web REST que implementamos en la tarea anterior.

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 "OK" si se pudo dar de alta el usuario, 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 definida en el archivo Usuario.java. 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 debido a que el cliente no es gráfico). Una vez que se tenga el objeto de tipo Usuario se deberá utilizar **JSON** para convertir el objeto a una string JSON, entonces se deberá codificar como URL y el resultado se utilizará como valor del parámetro. (véase la Figura 20).

```

File Edit Selection View Go Run Terminal Help Clientejava - SISDIS - Visual Studio Code
Clientejava 9+, M X
Tarea_07 > Cliente.java > Cliente
87     default:
88         System.out.println("Opcion no valida");
89         break;
90     }
91 } // fin main
92
93 /**
94 * Método para dar de alta un Usuario
95 */
96 public static void altaUsuario(Usuario usuario) throws Exception {
97     URL url = new URL("http://20.127.153.126:8080/Servicio/rest/ws/alta_usuario");
98     HttpURLConnection conexion = (HttpURLConnection) url.openConnection();
99     conexion.setDoOutput(true); // true si se va a enviar un "body", en este caso el "body" son los parámetros
100    conexion.setRequestMethod("POST"); // en este caso utilizamos el método POST de HTTP
101    conexion.setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); // indica que la petición estará codificada en formato URL
102    GsonBuilder builder = new GsonBuilder();
103    builder.serializeNulls();
104    Gson gson = builder.create();
105    String body = gson.toJson(usuario);
106    String parametros = "usuario=" + URLEncoder.encode(body, "UTF-8");
107    OutputStream os = conexion.getOutputStream();
108    os.write(parametros.getBytes());
109    os.flush();
110    /* se debe verificar si hubo error */
111    if (conexion.getResponseCode() == 200) { // no hubo error
112        BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getInputStream())));
113        String respuesta;
114        // el método web regresa una string en formato JSON
115        while ((respuesta = br.readLine()) != null) System.out.println("BIENVENIDO"+respuesta);
116    } else { // hubo error
117        BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getErrorStream())));
118        String respuesta; // el método web regresa una instancia de la clase Error en formato JSON
119        System.out.println("ERROR:");
120        while ((respuesta = br.readLine()) != null) System.out.println(respuesta);
121        //throw new RuntimeException("Código de error HTTP: " + conexion.getResponseCode()); // dispara una excepción para terminar
122    }
123    conexion.disconnect();
124
125 /**
126 *
127 * @param email
128 * @throws IOException
129 */
130 public static void consultarUsuario(String email) throws IOException {
131
132

```

Figura 20. Método “alta usuario”.

- La opción **“Consulta usuario”** leerá del teclado el email de un usuario previamente dado de alta. Entonces se invocará el método web **“consulta\_usuario”**. Si el usuario existe, se desplegará en pantalla el email del usuario, 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 (si existe) 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 **JSON** para convertir la string JSON a un objeto Java de tipo Usuario y posteriormente desplegar los campos del objeto (excepto el campo “foto”). **No se deberá desplegar la string JSON**. Si hubo error, se desplegará el mensaje que regresa el servicio web. (véase la Figura 21)

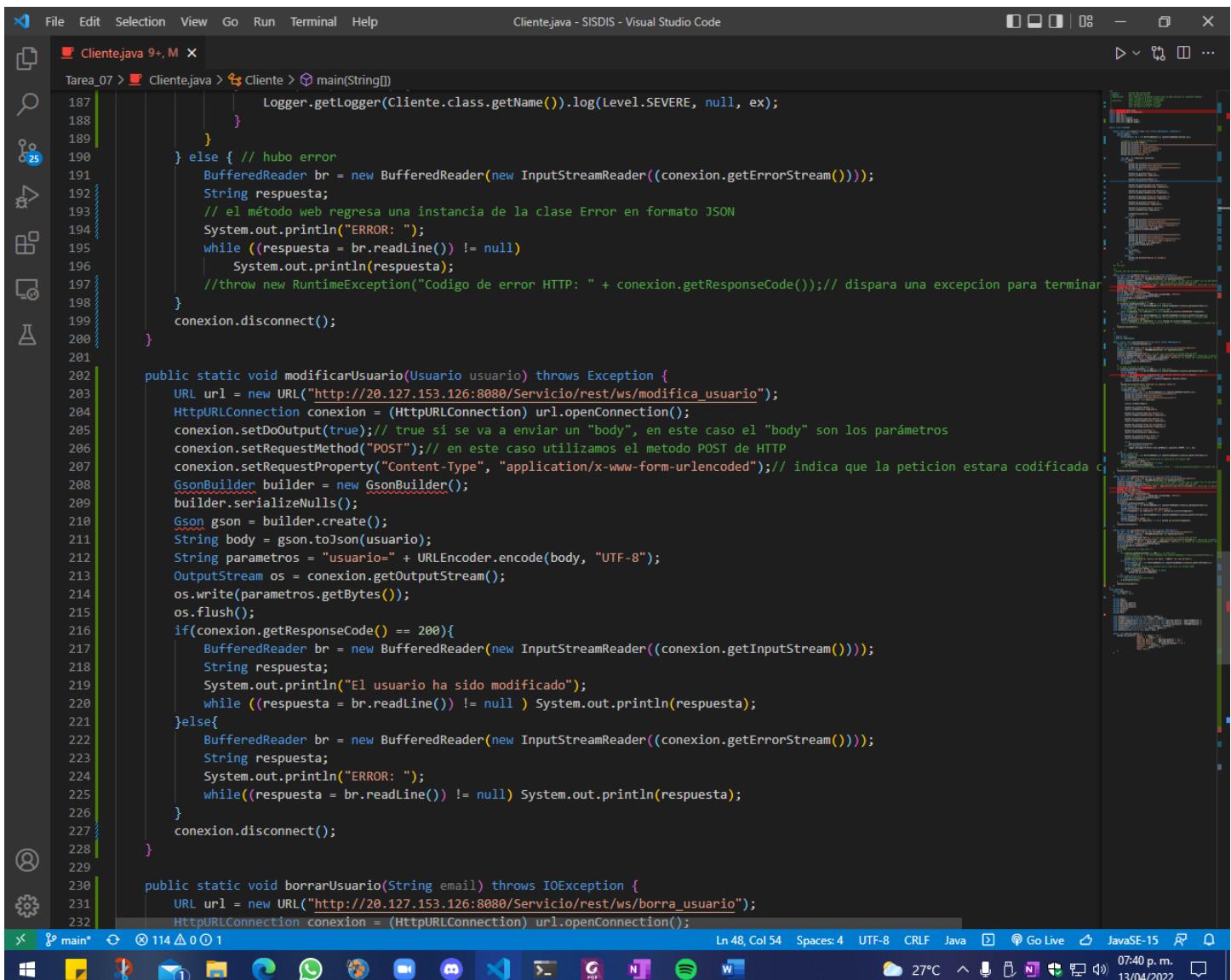
```

132 public static void consultarUsuario(String email) throws IOException {
133     Scanner sc = new Scanner(System.in);
134     //char opc = ' ';
135     URL url = new URL("http://20.127.153.126:8080/Servicio/rest/ws/consulta_usuario");
136     HttpURLConnection conexion = (HttpURLConnection) url.openConnection();
137     conexion.setDoOutput(true);
138     conexion.setRequestMethod("POST");// en este caso utilizamos el metodo POST de HTTP
139     conexion.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");// indica que la peticion estara codificada
140     String parametros = "email=" + URLEncoder.encode(email, "UTF-8");// el metodo web "consultarUsuario" recibe como parametro el
141     try (OutputStream os = conexion.getOutputStream()) {
142         os.write(parametros.getBytes());
143         os.flush();
144     }
145     /* se debe verificar si hubo error */
146     if (conexion.getResponseCode() == 200) { // no hubo error
147         BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getInputStream())));
148         String respuesta;
149         Gson j = new GsonBuilder().setDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS").create();
150         while ((respuesta = br.readLine()) != null){
151             Usuario usuario = (Usuario) j.fromJson(respuesta, Usuario.class);
152             usuario.Imprimir_datos();
153         }
154         System.out.println("Desea modificar el usuario? (S/N):");
155         //opc = sc.next().charAt(0);
156         String modifica = sc.nextLine();
157         if(modifica.equals("S")){
158             BufferedReader br2 = new BufferedReader(new InputStreamReader(System.in));
159             System.out.println("*****");
160             System.out.println("Modifica Usuario");
161             System.out.println("*****");
162             Usuario usuario = new Usuario();
163
164             usuario.setEmail(email);
165
166             System.out.println("Nombre:");
167             usuario.setNombre(br2.readLine());
168
169             System.out.println("Apellido Paterno:");
170             usuario.setApellidoPaterno(br2.readLine());
171
172             System.out.println("Apellido Materno:");
173             usuario.setApellidoMaterno(br2.readLine());
174
175             System.out.println("Fecha de nacimiento:");
176             usuario.setFechaNacimiento(br2.readLine());
177
}

```

Figura 21. Método "consulta usuario".

Una vez desplegados los datos del usuario **se** deberá preguntar "**Desea modificar el usuario (s/n)?**", si se responde con el carácter "s" entonces se leerá del teclado el nombre del usuario, el apellido paterno, el apellido materno, la fecha de nacimiento, el teléfono y el género ("M" o "F"). No se deberá modificar el email. Si en 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 "OK" si se pudo modificar el usuario, o bien, el mensaje de error que regresa el servicio web. (véase la Figura 22).



```

187     }
188 }
189 } else { // hubo error
190     BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getErrorStream())));
191     String respuesta;
192     // el método web regresa una instancia de la clase Error en formato JSON
193     System.out.println("ERROR: ");
194     while ((respuesta = br.readLine()) != null)
195         System.out.println(respuesta);
196     //throw new RuntimeException("Codigo de error HTTP: " + conexion.getResponseCode());// dispara una excepcion para terminar
197 }
198 conexion.disconnect();
199 }
200 }

201 public static void modificarUsuario(Usuario usuario) throws Exception {
202     URL url = new URL("http://20.127.153.126:8080/Servicio/rest/ws/modifica_usuario");
203     HttpURLConnection conexion = (HttpURLConnection) url.openConnection();
204     conexion.setDoOutput(true); // true si se va a enviar un "body", en este caso el "body" son los parámetros
205     conexion.setRequestMethod("POST"); // en este caso utilizamos el metodo POST de HTTP
206     conexion.setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); // indica que la petición estará codificada
207     GsonBuilder builder = new GsonBuilder();
208     builder.serializeNulls();
209     Gson gson = builder.create();
210     String body = gson.toJson(usuario);
211     String parametros = "usuario=" + URLEncoder.encode(body, "UTF-8");
212     OutputStream os = conexion.getOutputStream();
213     os.write(parametros.getBytes());
214     os.flush();
215     if(conexion.getResponseCode() == 200){
216         BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getInputStream())));
217         String respuesta;
218         System.out.println("El usuario ha sido modificado");
219         while ((respuesta = br.readLine()) != null ) System.out.println(respuesta);
220     }else{
221         BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getErrorStream())));
222         String respuesta;
223         System.out.println("ERROR: ");
224         while((respuesta = br.readLine()) != null) System.out.println(respuesta);
225     }
226     conexion.disconnect();
227 }
228 }

229 public static void borrarUsuario(String email) throws IOException {
230     URL url = new URL("http://20.127.153.126:8080/Servicio/rest/ws/borra_usuario");
231     HttpURLConnection conexion = (HttpURLConnection) url.openConnection();
232 }

```

The screenshot shows a Java code editor in Visual Studio Code. The file is named 'Cliente.java' and contains methods for modifying and deleting users via a service endpoint. The code uses HttpURLConnection to interact with a RESTful web service at 'http://20.127.153.126:8080/Servicio/rest/ws'. It handles both successful responses (status 200) and errors (status > 200). The code uses Gson to serialize and deserialize objects. The interface includes standard VS Code icons for file operations, search, and navigation.

Figura 22. Método "modifica usuario".

- La opción "**Borra usuario**" leerá del teclado el email de un usuario previamente dado de alta. Entonces se invocará el método "**borra\_usuario**" del servicio web. Se deberá desplegar "OK" si se pudo borrar el usuario, o bien, el mensaje de error que regresa el servicio web. (véase la Figura 23)

The screenshot shows a Java code editor in Visual Studio Code. The file being edited is `Cliente.java`. The code implements a client for a REST service. It includes methods for sending POST requests to delete users and for reading user data from a URL. The code uses `HttpURLConnection` to handle the HTTP connections. The interface includes standard VS Code icons for file operations, search, and navigation.

```
224     System.out.println("ERROR: ");
225     while((respuesta = br.readLine()) != null) System.out.println(respuesta);
226   }
227   conexion.disconnect();
228 }
229
230 public static void borrarUsuario(String email) throws IOException {
231   URL url = new URL("http://20.127.153.126:8080/Servicio/rest/ws/borra_usuario");
232   HttpURLConnection conexion = (HttpURLConnection) url.openConnection();
233   conexion.setDoOutput(true);
234   conexion.setRequestMethod("POST");// en este caso utilizamos el metodo POST de HTTP
235   conexion.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");// indica que la peticion estara codificada en formato URL
236   String parametros = "email=" + URLEncoder.encode(email, "UTF-8");// el metodo web "consultarUsuario" recibe como parametro el email
237   OutputStream os = conexion.getOutputStream();
238   os.write(parametros.getBytes());
239   os.flush();
240   /* se debe verificar si hubo error */
241   try {
242     if (conexion.getResponseCode() == 200) { // no hubo error
243       /*BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getInputStream())));
244       String respuesta;*/
245       System.out.println("El usuario con email: "+email+" ha sido borrado");
246     } else { // hubo error
247       BufferedReader br = new BufferedReader(new InputStreamReader((conexion.getErrorStream())));
248       String respuesta;
249       // el metodo web regresa una instancia de la clase Error en formato JSON
250       System.out.println("ERROR: ");
251       while ((respuesta = br.readLine()) != null)
252         System.out.println(respuesta);
253     }
254   } catch (IOException e) {
255     // TODO Auto-generated catch block
256     e.printStackTrace();
257   }
258   conexion.disconnect();
259 }
260 }
261 class Usuario{
262   public Usuario() {
263     this.foto = null;
264   }
265   String email;
266   String nombre;
267   String apellido_paterno;
268   String apellido_materno;
```

Figura 23. Método “borra usuario”.

- La opción "Salir" terminará el programa.

## Pruebas

En la Figura 24 se observa la vista del menú creado en consola:

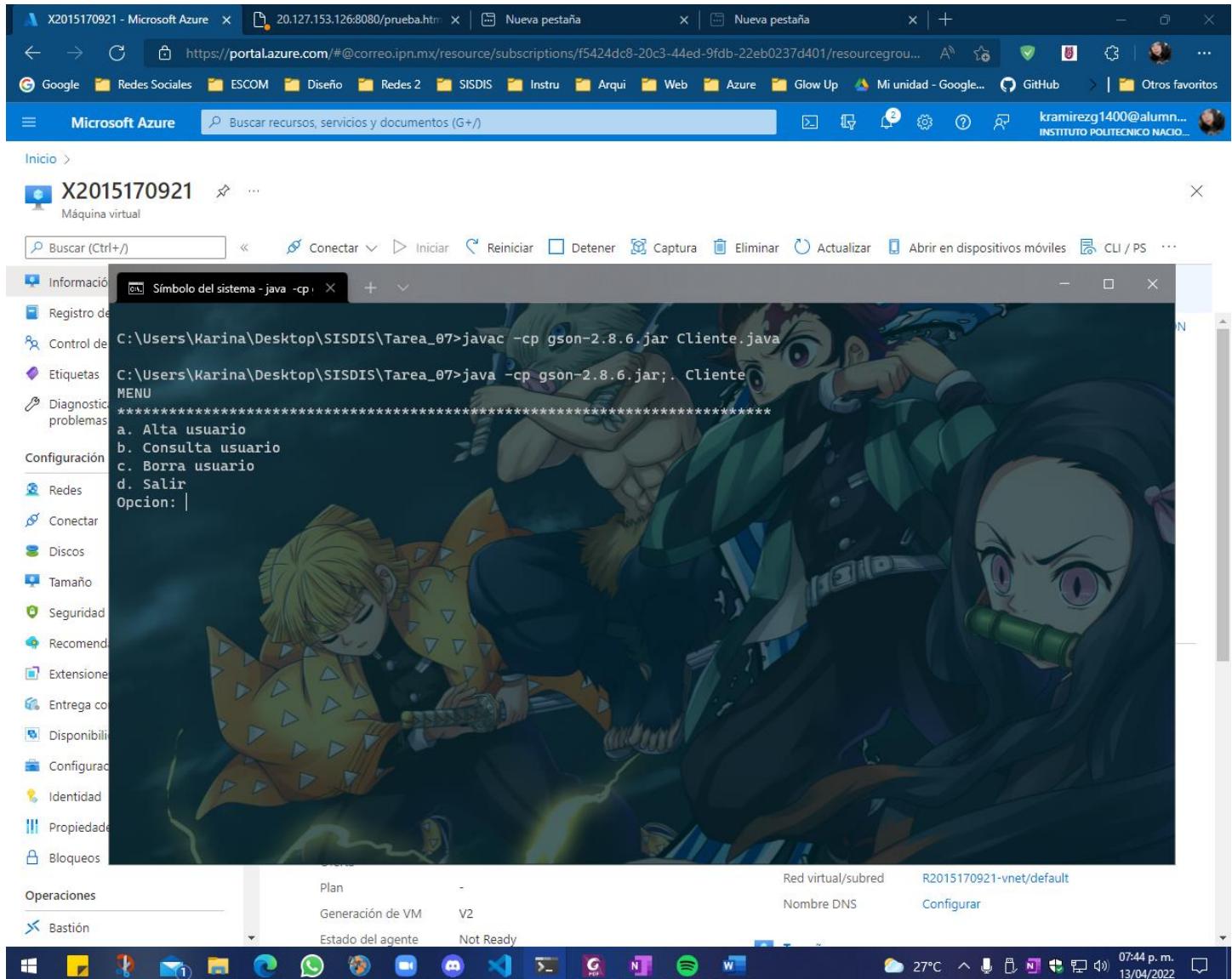


Figura 24. Menú de Usuario.

Se deberá realizar las siguientes pruebas:

1. Dar de alta un nuevo usuario. (véase la Figura 25).

En este caso registraremos los siguientes datos:

- ▶ Email: [karyRmz@gmail.com](mailto:karyRmz@gmail.com)
- ▶ Nombre: Karina
- ▶ Apellido paterno: Ramirez
- ▶ Apellido Materno: Galindo

- Fecha de nacimiento: 1999/09/03
- Teléfono: 1234
- Género: F

The screenshot shows a Windows desktop environment. In the center is a terminal window titled "Símbolo del sistema - java -cp". The command entered is "javac -cp gson-2.8.6.jar Cliente.java". Below this, the output of the compilation is shown. A red box highlights the menu options and the user input for creating a new user:

```
C:\Users\Karina\Desktop\SISDIS\Tarea_07>javac -cp gson-2.8.6.jar Cliente.java
C:\Users\Karina\Desktop\SISDIS\Tarea_07>java -cp gson-2.8.6.jar:. Cliente
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: a
*****
Alta Usuario
*****
Email:
karyRmz@gmail.com
Nombre:
Karina
Apellido Paterno:
Ramirez
Apellido Materno:
Galindo
Fecha de nacimiento:
1999/09/03
Telefono:
1234
Genero (M/F):
F
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion:
```

The terminal window is overlaid on a background image of an anime-style illustration featuring several characters, including a wolf-like character and a girl with purple hair.

Figura 25. "Alta" de un Usuario.

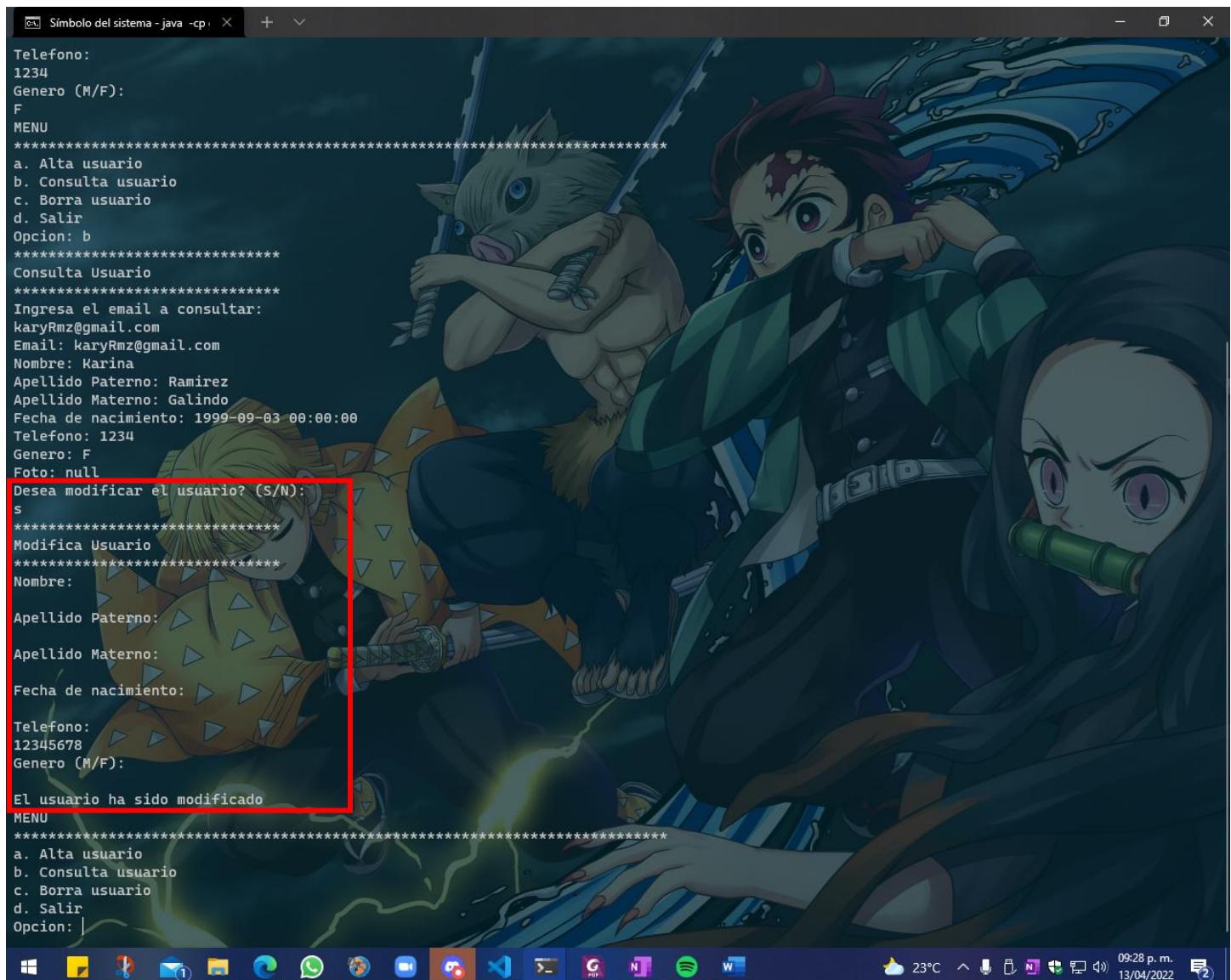
2. Consultar el usuario dado de alta anteriormente. (véase la Figura 26).

```
C:\Users\Karina\Desktop\SISDIS\Tarea_07>java -cp gson-2.8.6.jar;. Cliente
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: a
*****
Alta Usuario
*****
Email:
karyRmz@gmail.com
Nombre:
Karina
Apellido Paterno:
Ramirez
Apellido Materno:
Galindo
Fecha de nacimiento:
1999/09/03
Telefono:
1234
Genero (M/F):
F
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: b
*****
Consulta Usuario
*****
Ingrésa el email a consultar:
karyRmz@gmail.com
Email: karyRmz@gmail.com
Nombre: Karina
Apellido Paterno: Ramirez
Apellido Materno: Galindo
Fecha de nacimiento: 1999-09-03 00:00:00
Telefono: 1234
Genero: F
Foto: null
Desea modificar el usuario? (S/N):
```

Figura 26. "Consulta" de un Usuario.

### 3. Modificar el teléfono del usuario.

En la pregunta “Desea modificar el usuario (S/N)?” escribimos S para poder modificar el teléfono. (véase la Figura 27).



```
Símbolo del sistema - java -cp . *.java
Telefono: 1234
Genero (M/F): F
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: b
*****
Consulta Usuario
*****
Ingresa el email a consultar:
KaryRmz@gmail.com
Email: KaryRmz@gmail.com
Nombre: Karina
Apellido Paterno: Ramirez
Apellido Materno: Galindo
Fecha de nacimiento: 1999-09-03 00:00:00
Telefono: 1234
Genero: F
Foto: null
Desea modificar el usuario? (S/N):
S
*****
Modifica Usuario
*****
Nombre:
Apellido Paterno:
Apellido Materno:
Fecha de nacimiento:
Telefono:
12345678
Genero (M/F):
El usuario ha sido modificado
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: |
```

Figura 27. "Modificación" de un Usuario.

4. Consultar el usuario modificado, para verificar que la modificación se realizó. (véase la Figura 28).

Nótese que el numero anterior era “1234”, ahora con la modificación el nuevo número es “12345678”

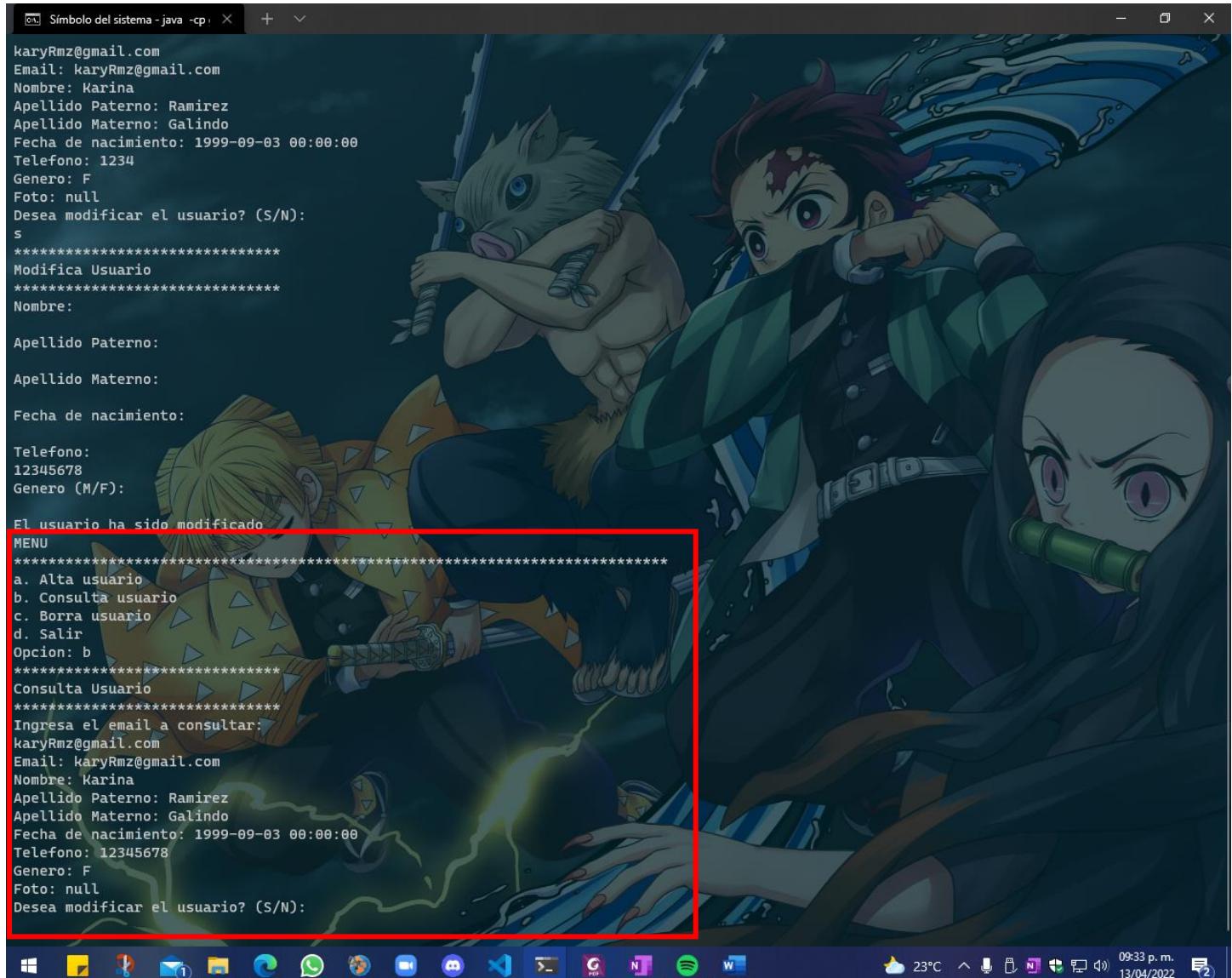
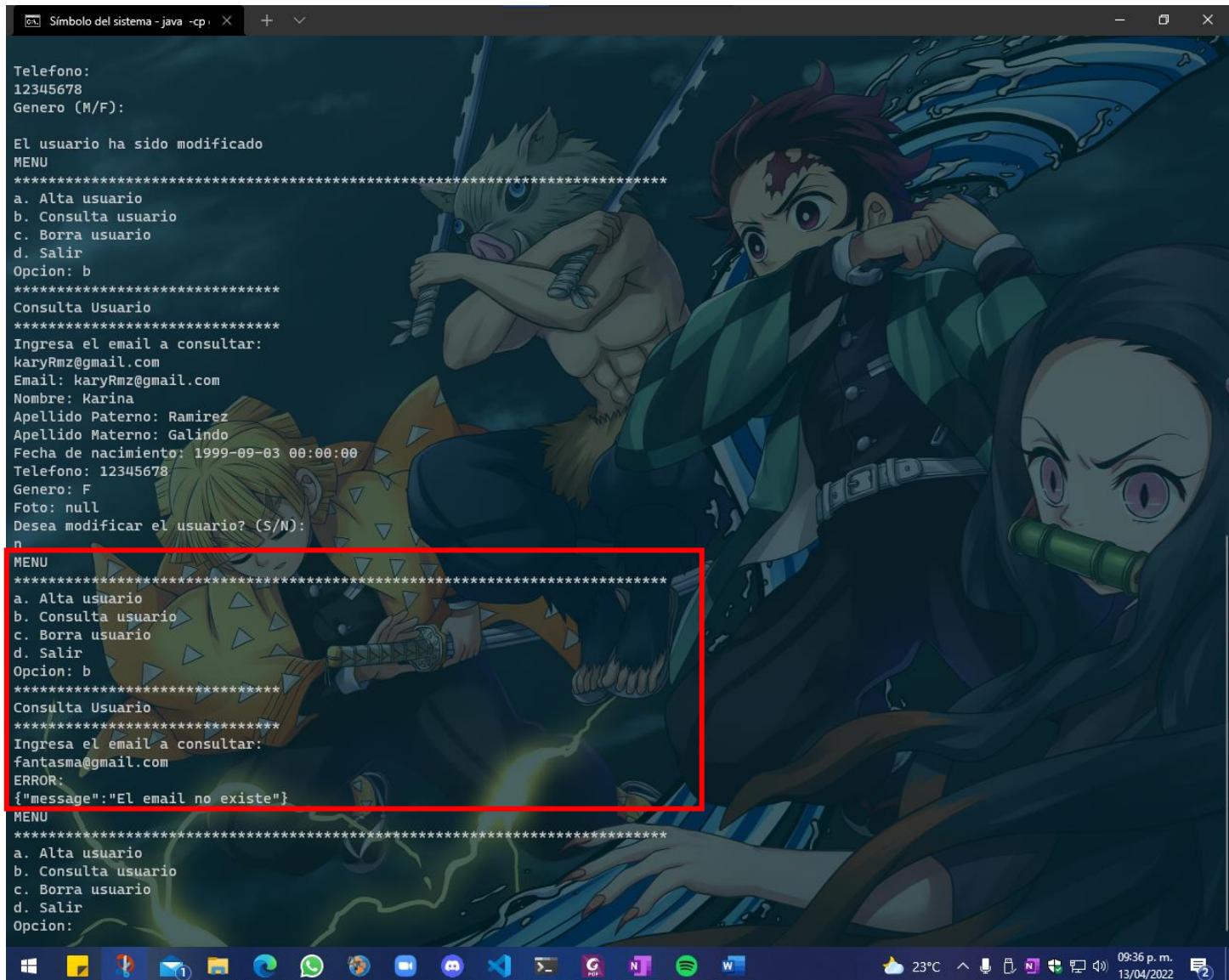


Figura 28. Teléfono de usuario modificado.

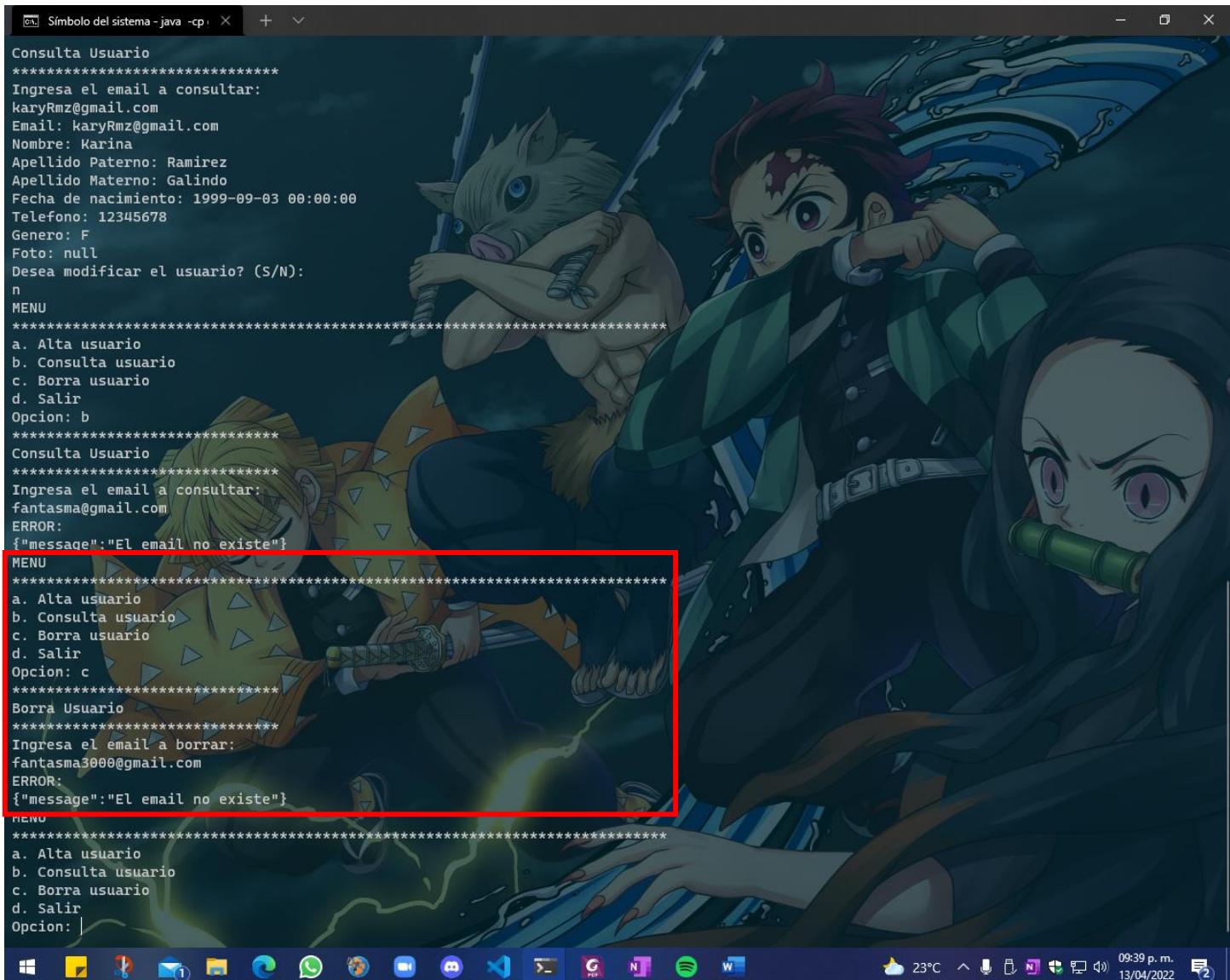
5. Consultar un usuario que no existe, se deberá desplegar el mensaje de error. (véase la Figura 29).



```
Símbolo del sistema - java -cp . . . . .  
Telefono:  
12345678  
Genero (M/F):  
El usuario ha sido modificado  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion: b  
*****  
Consulta Usuario  
*****  
Ingresa el email a consultar:  
karyRmz@gmail.com  
Email: karyRmz@gmail.com  
Nombre: Karina  
Apellido Paterno: Ramirez  
Apellido Materno: Galindo  
Fecha de nacimiento: 1999-09-03 00:00:00  
Telefono: 12345678  
Genero: F  
Foto: null  
Desea modificar el usuario? (S/N):  
n  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion: b  
*****  
Consulta Usuario  
*****  
Ingresa el email a consultar:  
fantasma@gmail.com  
ERROR:  
{"message": "El email no existe"}  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion:
```

Figura 29. “Consulta” de un usuario inexistente.

6. Intentar borrar un usuario que no existe, se deberá desplegar el mensaje de error. (véase la Figura 30).



```
Símbolo del sistema - java -cp . Consulta Usuario
*****
Ingresa el email a consultar:
karyRmz@gmail.com
Email: karyRmz@gmail.com
Nombre: Karina
Apellido Paterno: Ramirez
Apellido Materno: Galindo
Fecha de nacimiento: 1999-09-03 00:00:00
Telefono: 12345678
Genero: F
Foto: null
Desea modificar el usuario? (S/N):
n
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: b
*****
Consulta Usuario
*****
Ingresa el email a consultar:
fantasma@gmail.com
ERROR:
{"message": "El email no existe"}
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: c
*****
Borra Usuario
*****
Ingresa el email a borrar:
fantasma3000@gmail.com
ERROR:
{"message": "El email no existe"}
MENU
*****
a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Salir
Opcion: |
```

Figura 30. “Borrar” un usuario inexistente.

7. Intentar dar de alta un usuario con un email que ya existe, se deberá desplegar el mensaje de error. (véase la Figura 31).

Se intento dar de alta el usuario registrado en la prueba 1, por lo que nos manda un mensaje de que el email ya existe.

```
ERROR:  
{"message": "El email no existe"}  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion: c  
*****  
Borra Usuario  
*****  
Ingresa el email a borrar:  
fantasma3000@gmail.com  
ERROR:  
{"message": "El email no existe"}  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion: a  
*****  
Alta Usuario  
*****  
Email:  
karyRmz@gmail.com  
Nombre:  
lola  
Apellido Paterno:  
perez  
Apellido Materno:  
lopez  
Fecha de nacimiento:  
1999/10/10  
Telefono:  
1234  
Genero (M/F):  
F  
ERROR:  
{"message": "El email ya existe"}  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion:
```

Figura 31. “Alta” un usuario existente.

8. Borrar el usuario dado de alta anteriormente. (véase la Figura 32).

```
Símbolo del sistema - java -cp . . . . .  
karyRmz@gmail.com  
Nombre:  
Lola  
Apellido Paterno:  
perez  
Apellido Materno:  
lopez  
Fecha de nacimiento:  
1999/10/10  
Telefono:  
1234  
Genero (M/F):  
F  
ERROR:  
{"message": "El email ya existe"}  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion: c  
*****  
Borra Usuario  
*****  
Ingresa el email a borrar:  
karyRmz@gmail.com  
El usuario con email: karyRmz@gmail.com ha sido borrado  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion: b  
*****  
Consulta Usuario  
*****  
Ingresa el email a consultar:  
karyRmz@gmail.com  
ERROR:  
{"message": "El email no existe"}  
MENU  
*****  
a. Alta usuario  
b. Consulta usuario  
c. Borra usuario  
d. Salir  
Opcion:
```

Figura 32. “Borrar” un usuario existente.

## Concusiones

La web es muy bien ejemplo de un sistema distribuido, en esta practica haciendo uso de protocolos HTTP al momento de implementar un cliente web estilo REST. Se reforzó el conocimiento de la tarea anterior de montar el servicio web con apache tomcat.

Y se tuvo una interacción directa al momento de estar haciendo albas, bajas y cambios en el menú de usuario y comprobando que cada una de las acciones se hicieran correctamente pasando los parámetros de objetos usando JSON.

## Referencias

- [1] P. G. Carlos, «Desarrollo de sistemas Distribuidos - 4CV11 Plataforma Educativa Moodle,» [En línea]. Available: <https://m4gm.com/moodle>.