*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Desarrollo de Sistemas Distribuidos*

## *Tarea 7. Implementación de un servicio web estilo REST*

***Nombre:*** *Sampayo Hernández Mauro*

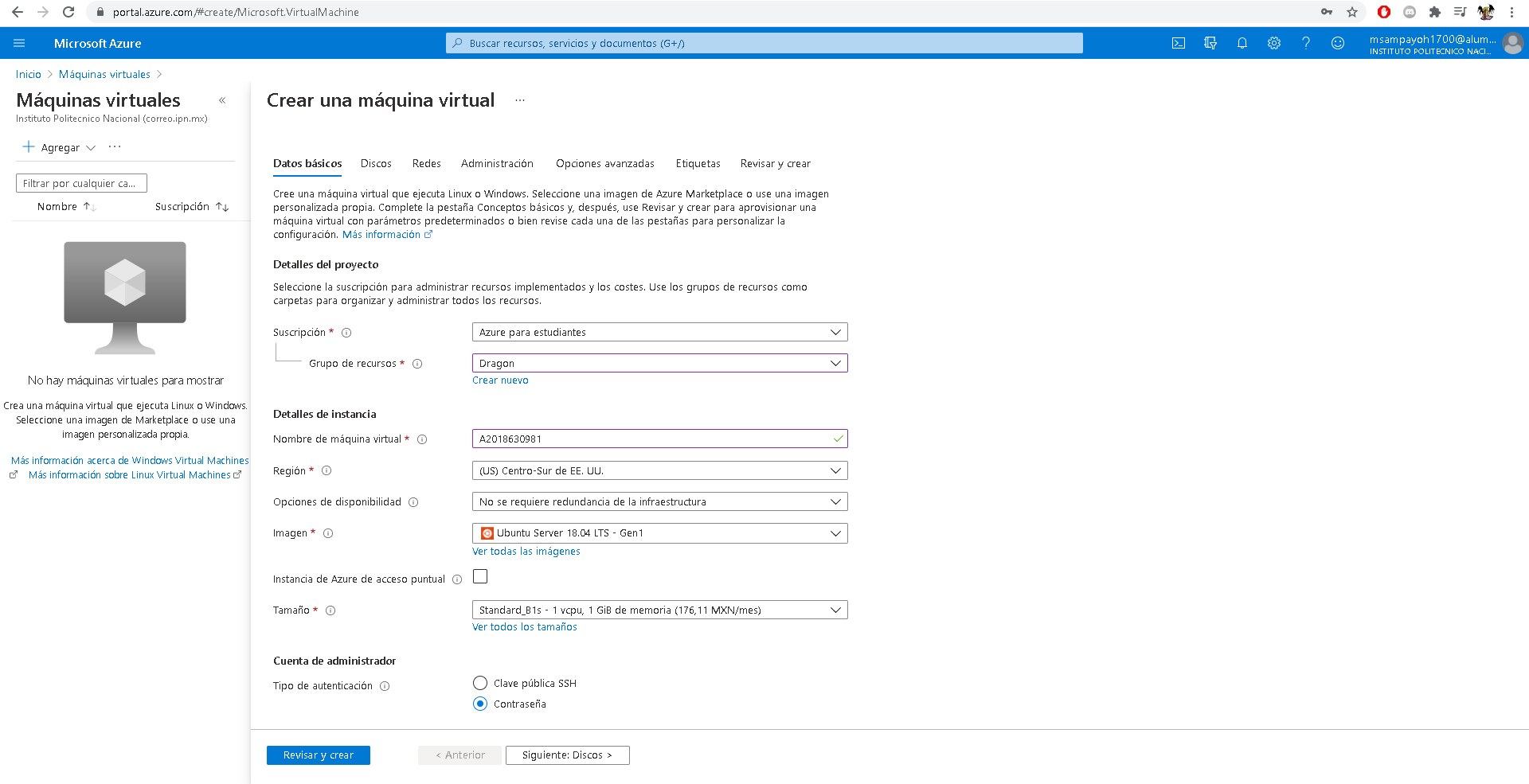
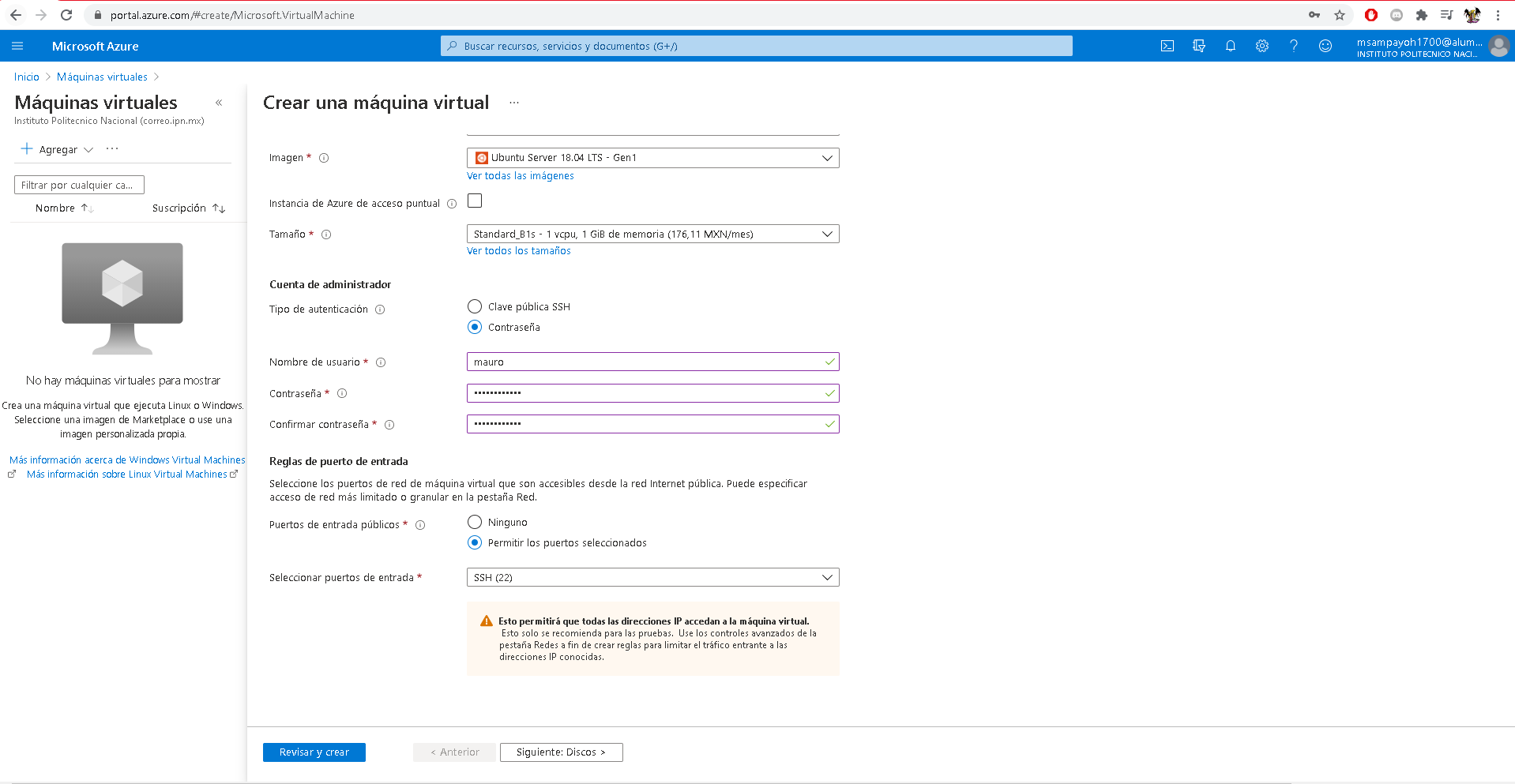
***Grupo:*** *4CV1*

***Profesor:*** *Pineda Guerrero Carlos*

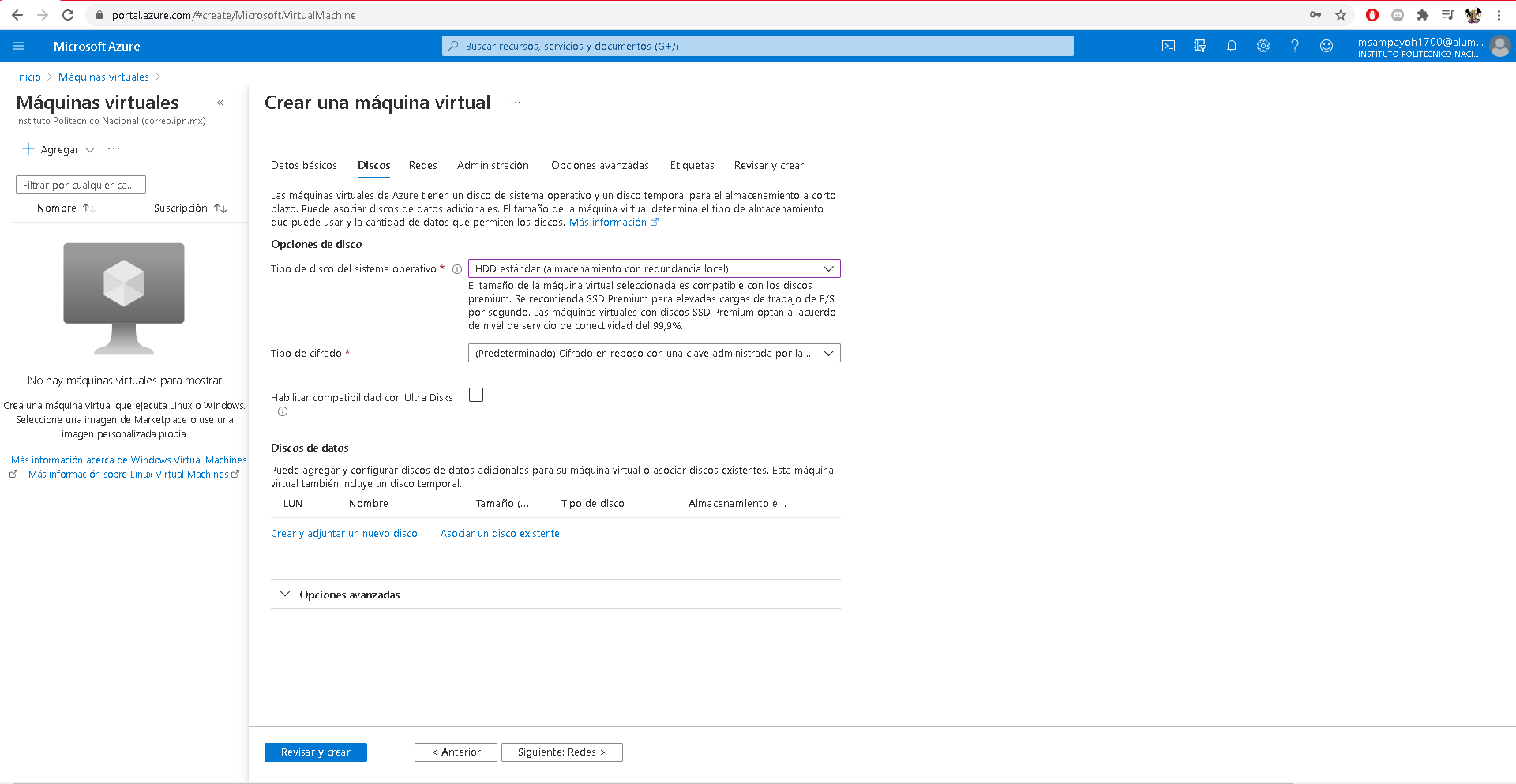
**Creación de las máquinas virtuales:**

Se realiza la creación de una máquina virtual la cual estará configurada con una imagen del S.O. Ubuntu Server 18.04 LTS.

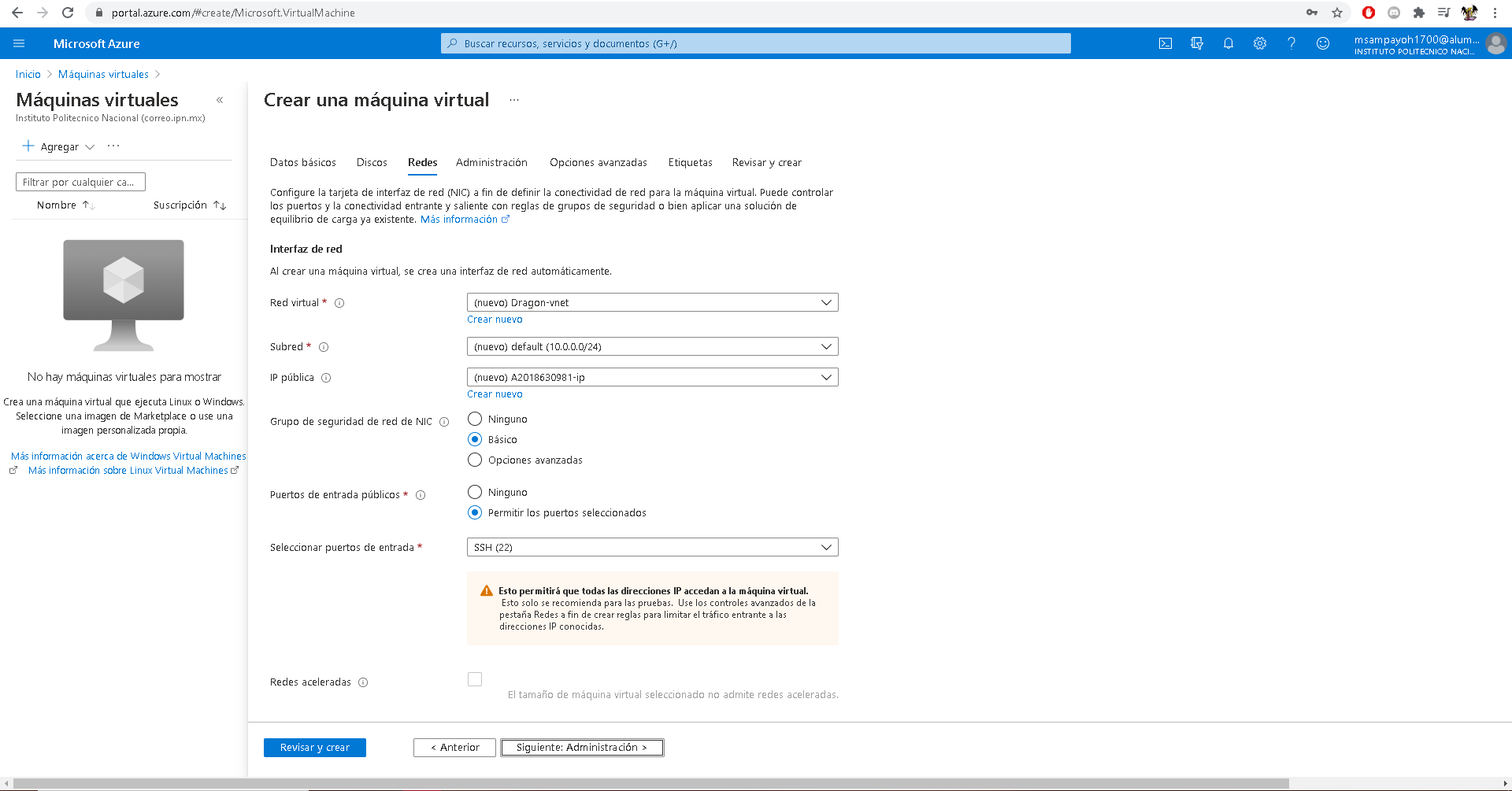
Se inicia configurando la sección de “Datos Básicos” de la máquina virtual, donde ingresaremos el nombre tanto de usuario como de la máquina virtual, y la contraseña; seleccionaremos la región y el tamaño de la máquina virtual; y dejaremos abierto en las "Reglas de puerto de entrada" el puerto 22 para utilizar SSH (la terminal de secure shell).

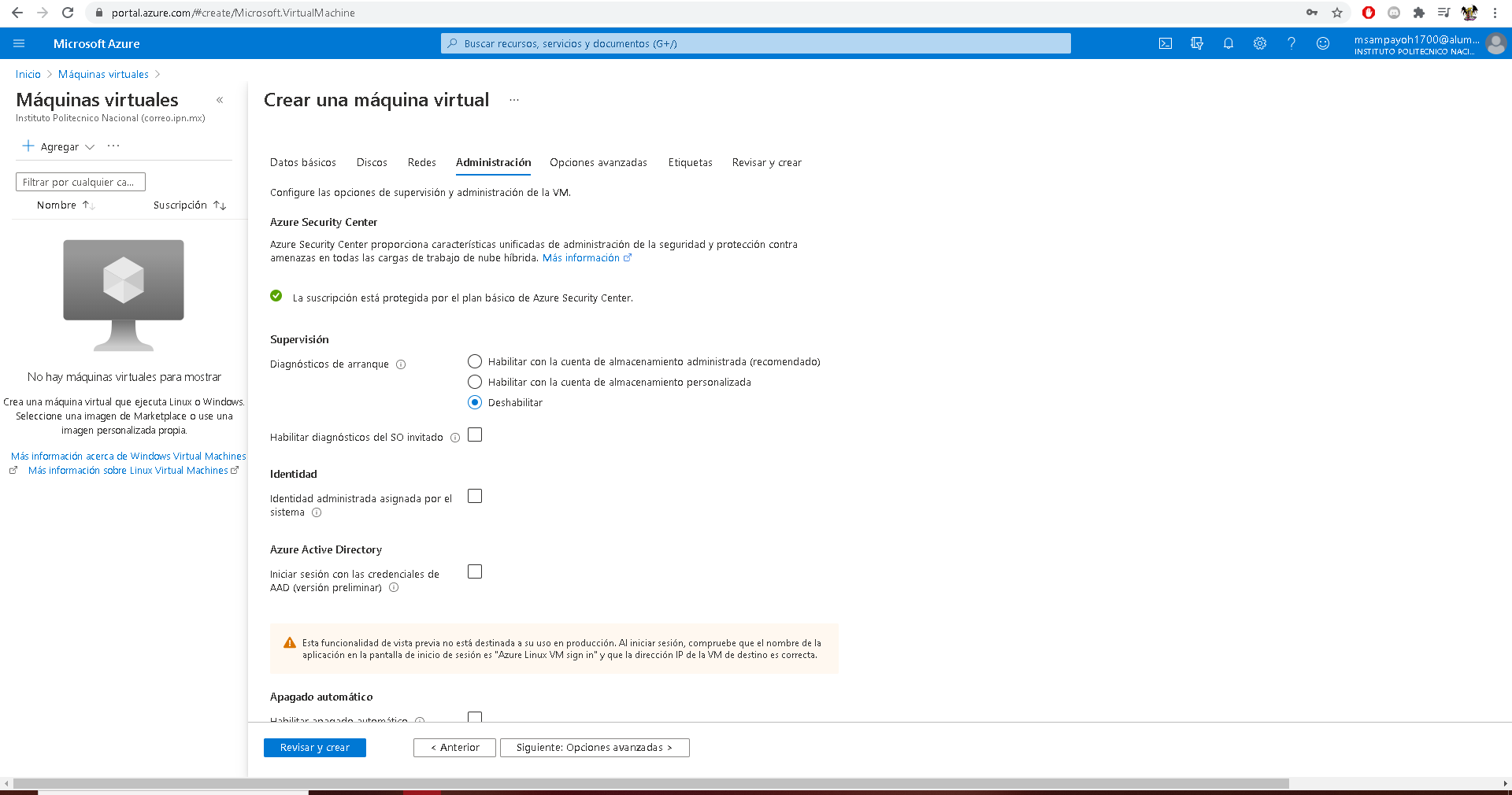
En la sección de “Discos” seleccionaremos la opción de HDD estándar como tipo de disco de sistema operativo.



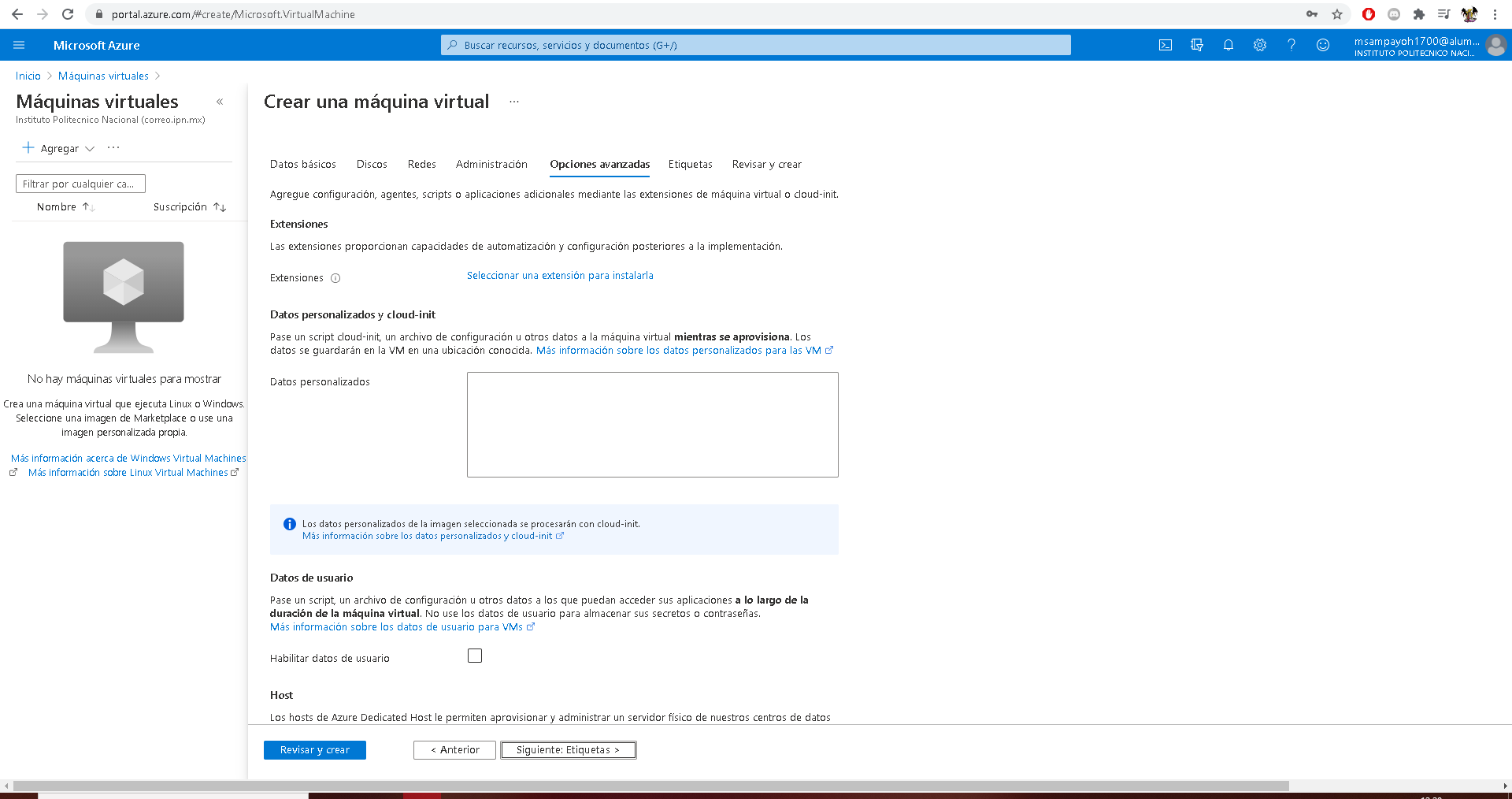
La sección de “Redes” se deja sin alterar, y con su configuración por defecto.

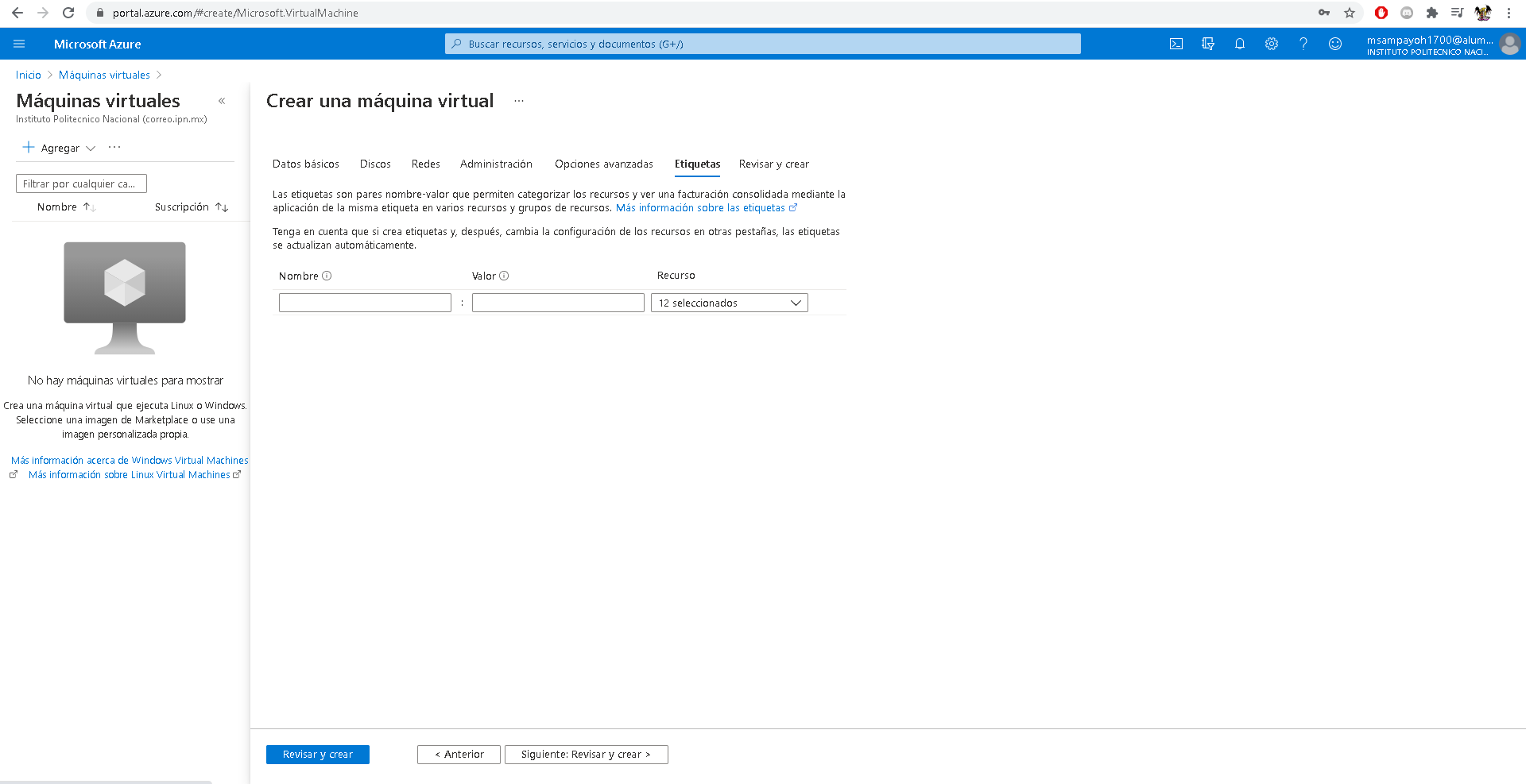


En la sección de “Administración” dejaremos el campo de “Diagnóstico de Arranque” como “Deshabilitado”.

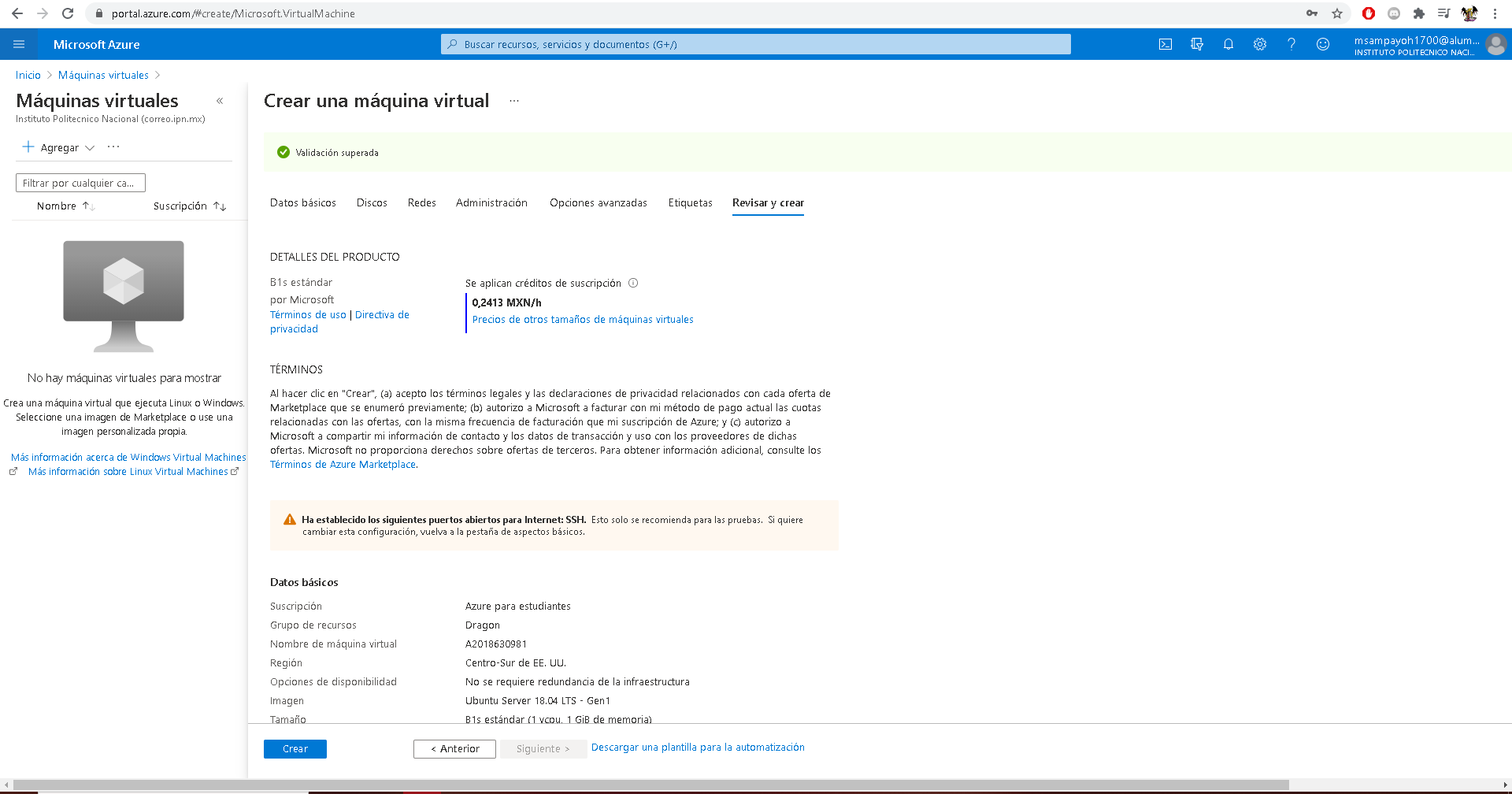


Finalmente, las secciones “Opciones Avanzadas” y “Etiquetas” se dejarán sin modificación alguna, con su configuración por defecto.

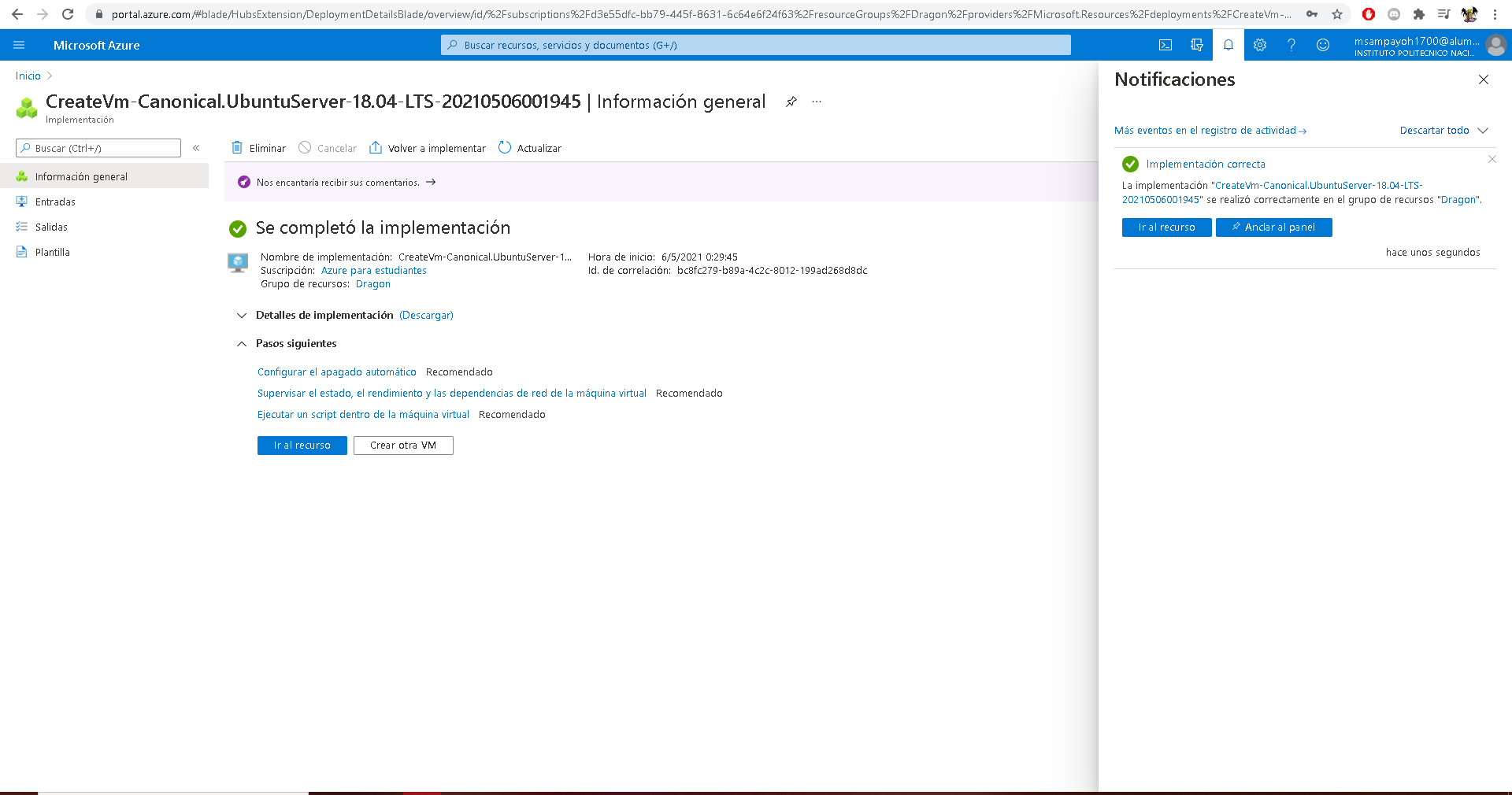




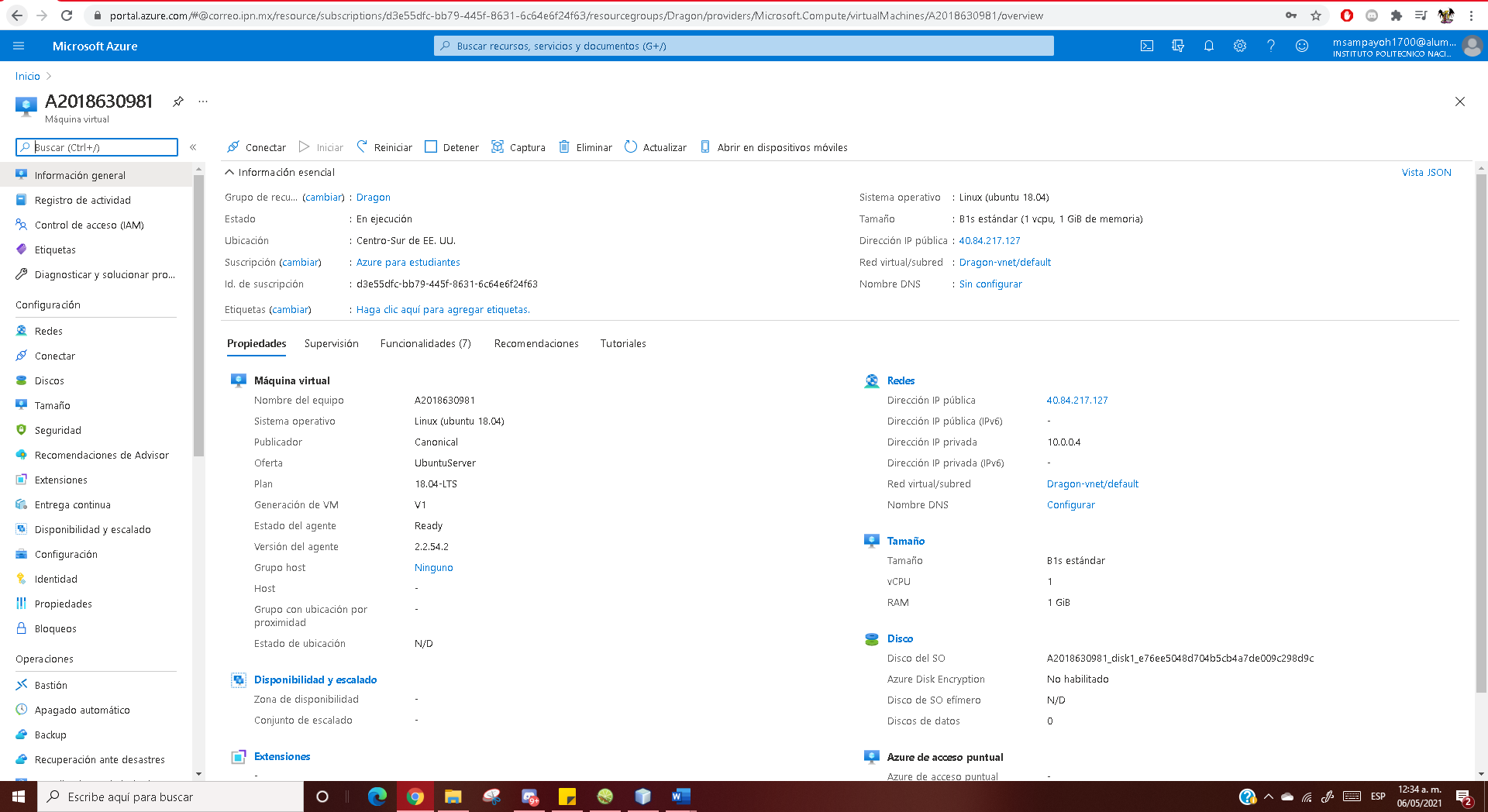
Damos click en el botón “Revisar y Crear”.



Una vez la máquina virtual es validada, se da click en el botón de “Crear” para así completar su implementación.



Una vez completado lo anteriormente mencionado se da click a la campana de notificaciones para verificar que la máquina virtual se haya creado.  
 y posteriormente se da click al botón “Ir al Recurso”.

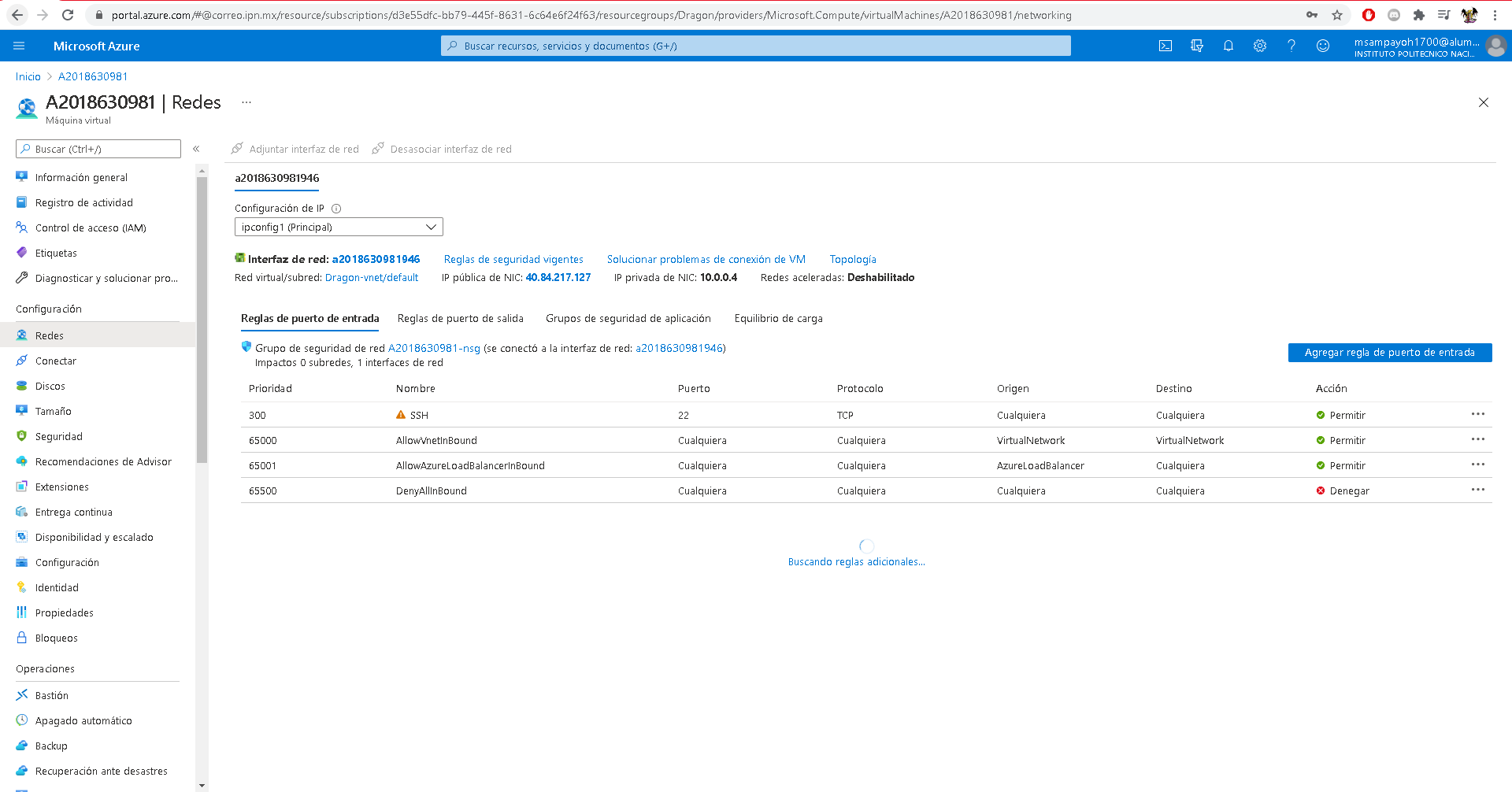


A partir de este punto se continuará con la implementación de la practica tomando como base el procedimiento propuesto por el profesor:

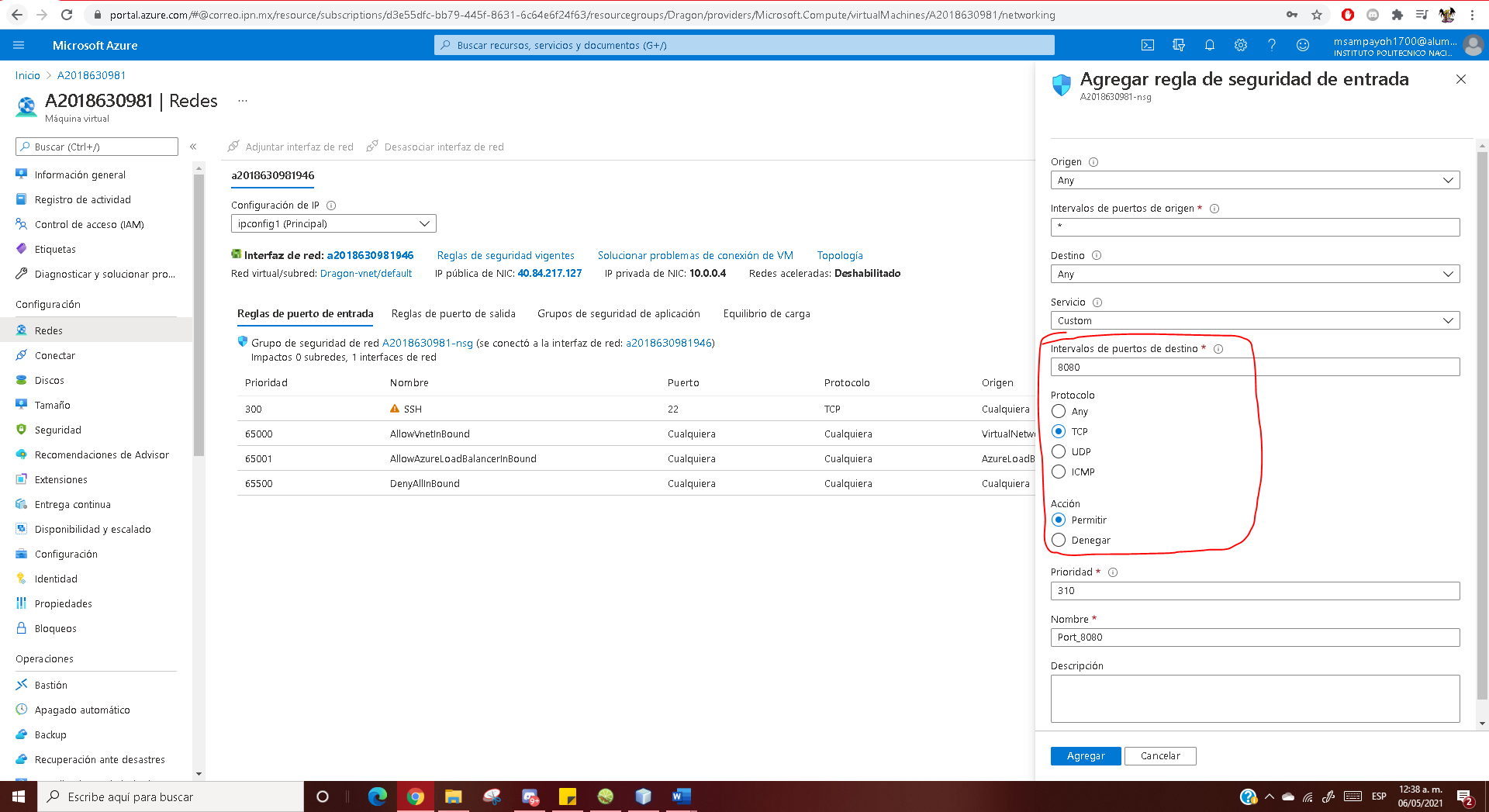
**Instalación de Tomcat con soporte REST**

1. Crear una máquina virtual con Ubuntu 18 con al menos 1GB de memoria RAM. Abrir el puerto 8080 para el protocolo TCP.

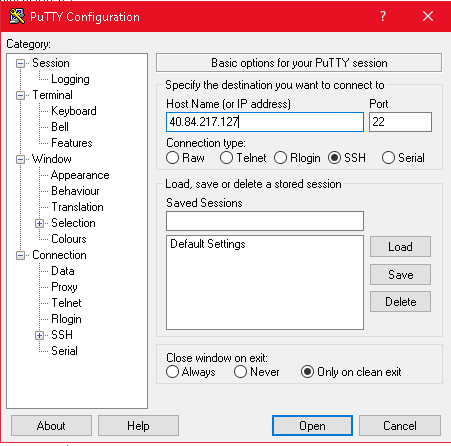
Para este paso solo hace falta abrir el puerto 8080 lo cual se realiza yendo a la sección de “Redes”



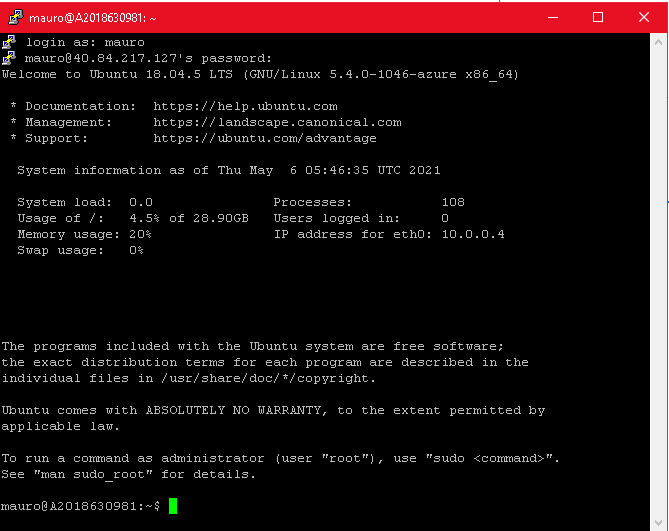
Una vez ahí, se selecciona la opción que dice “Agregar regla de puerto de entrada” que nos desplegará un menú, en el cuál seleccionaremos la casilla con la opción “TCP”, y en el apartado “Intervalos de puertos de destino” se seleccionará el puerto el cuál para nuestro caso será el 8080. Finalmente solo queda dar click en “Agregar”.



Hecho esto se procede a conectarse a la máquina virtual por medio del uso el programa putty.exe ingresando la IP pública de esta.



Hecho esto, accedemos a la máquina virtual con el usuario y la contraseña que configuramos previamente al momento de su creación.



A partir de aquí se puede proceder con el resto de los pasos.

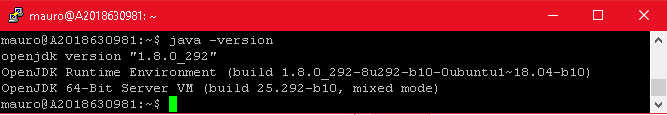
1. Instalar JDK8 ejecutando los siguientes comandos en la máquina virtual:

sudo apt update



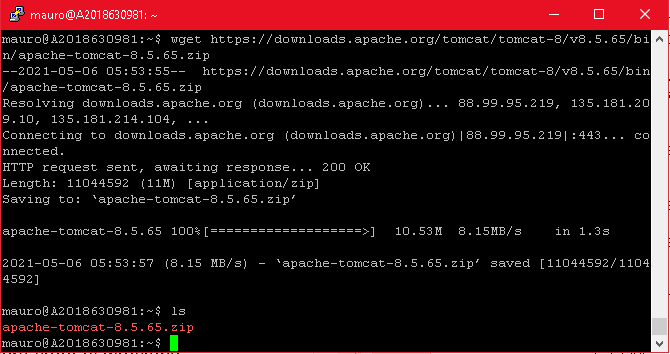
sudo apt install openjdk-8-jdk-headless

Comprobamos su correcta instalación por medio del comando “java -version”.



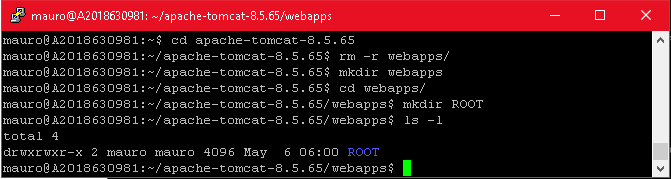
1. Descargar la distribución binaria de Tomcat 8 de la siguiente URL (descargar la opción Core "zip"):<https://tomcat.apache.org/download-80.cgi>
2. Copiar a la máquina virtual el archivo ZIP descargado anteriormente y desempacarlo utilizando el comando unzip.

Para estos dos pasos utilizaremos como alternativa el comando “wget” con la url <https://downloads.apache.org/tomcat/tomcat-8/v8.5.65/bin/apache-tomcat-8.5.65.zip> en la cual se encuentra el archivo ZIP de tomcat para descargarlo directamente en la máquina virtual.



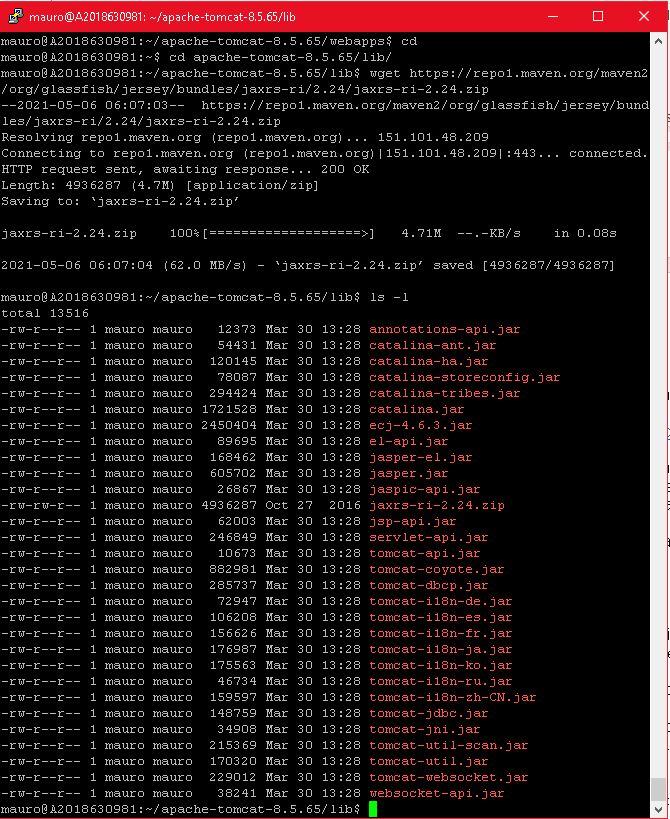
Desempacamos el archivo con el comando “unzip”

1. Eliminar el directorio webapps el cual se encuentra dentro del directorio de Tomcat. Crear un nuevo directorio webapps y dentro de éste se deberá crear el directorio ROOT.

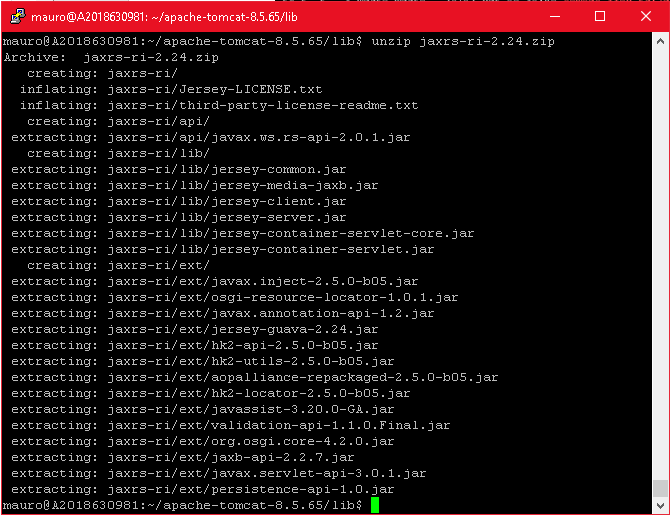


1. Descargar la biblioteca "Jersey" de la siguiente URL. Jersey es una implementación de JAX-RS lo cual permite ejecutar servicios web estilo REST sobre Tomcat: <https://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.24/jaxrs-ri-2.24.zip>
2. Copiar a la máquina virtual el archivo descargado anteriormente, desempacarlo y copiar todos los archivos con extensión “.jar” de todos los directorios desempacados, al directorio "lib" de Tomcat.

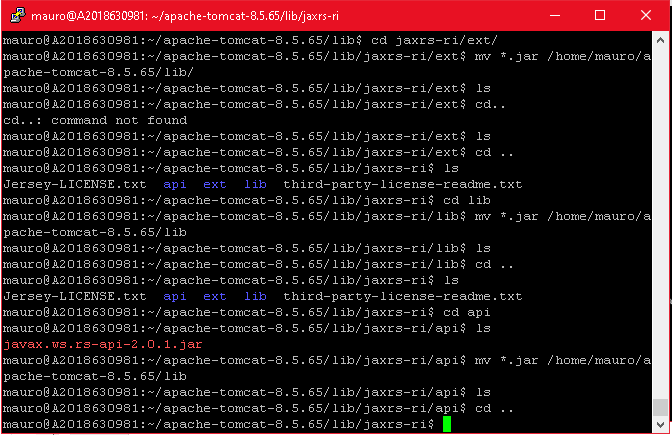
Nuevamente utilizaremos el comando “wget” para descargarlo directamente en la máquina virtual.



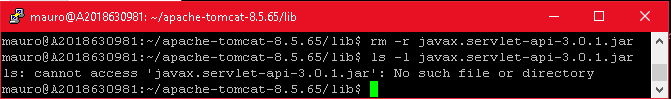
Desempacamos el archivo con el comando “unzip”



Copiamos todos los 1. “.jar” de todos los directorios desempacados, al directorio "lib" de Tomcat.



1. Borrar el archivo javax.servlet-api-3.0.1.jar del directorio "lib" de Tomcat (esto debe hacerse ya que existe una incompatibilidad entre Tomcat y Jersey 2).

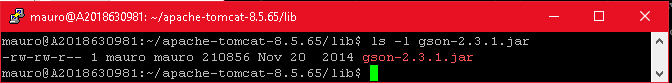


1. Descargar el archivo gson-2.3.1.jar de la URL: <https://repo1.maven.org/maven2/com/google/code/gson/gson/2.3.1/gson-2.3.1.jar>

Usando “wget”

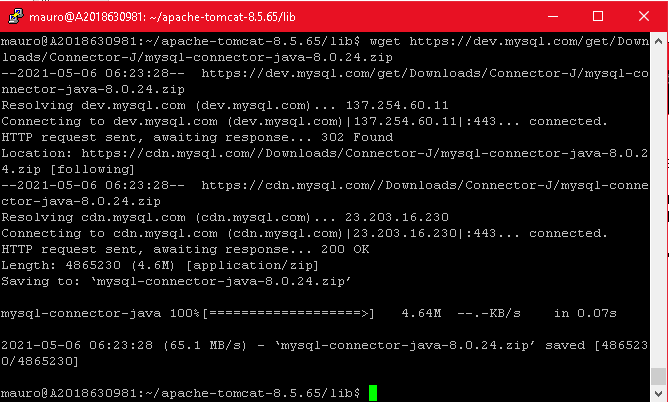


1. Copiar el archivo gson-2.3.1.jar al directorio "lib" de Tomcat.



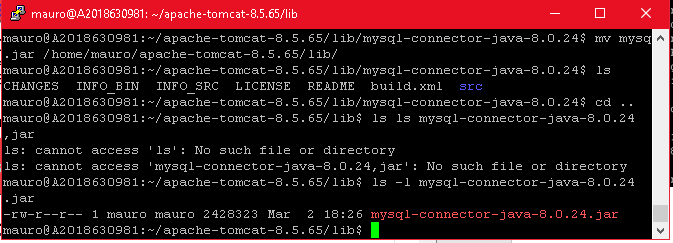
1. Ahora vamos a instalar el driver de JDBC para MySQL. Ingresar a la siguiente URL: <https://dev.mysql.com/downloads/connector/j/>. Seleccionar “Platform independent" y descargar el archivo ZIP.

Usamos “wget” con la url https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.24.zip en la cual se encuentra el archivo ZIP para descargarlo directamente en la máquina virtual.



1. Copiar el archivo descargado a la máquina virtual, desempacarlo y copiar el archivo mysql-connector.jar al directorio "lib" de Tomcat.

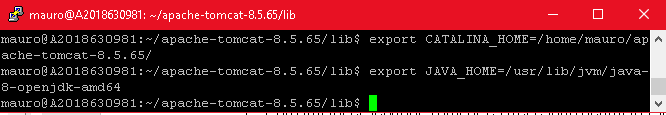


 **Iniciar/detener el servidor Tomcat**

1. Para iniciar el servidor Tomcat es necesario definir las siguientes variables de entorno:

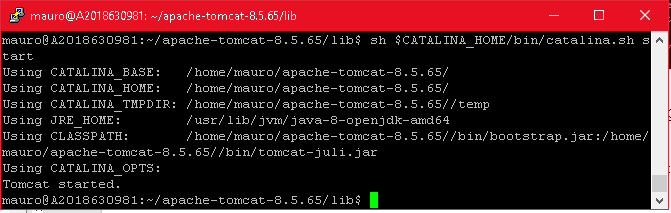
export CATALINA\_HOME=aquí va la ruta del directorio de Tomcat 8

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64



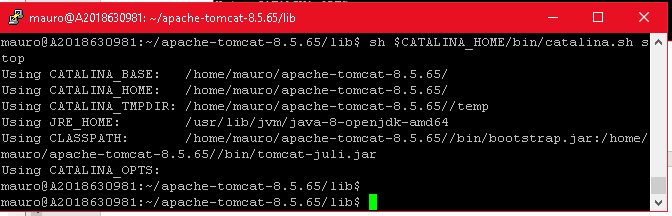
1. Iniciar la ejecución de Tomcat ejecutando el siguiente comando:

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



1. Para detener la ejecución de Tomcat se deberá ejecutar el siguiente comando:

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



**Instalación de MySQL**

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

sudo apt update

1. Instalar el paquete default de MySQL:

sudo apt install mysql-server

1. Ejecutar el script de seguridad:

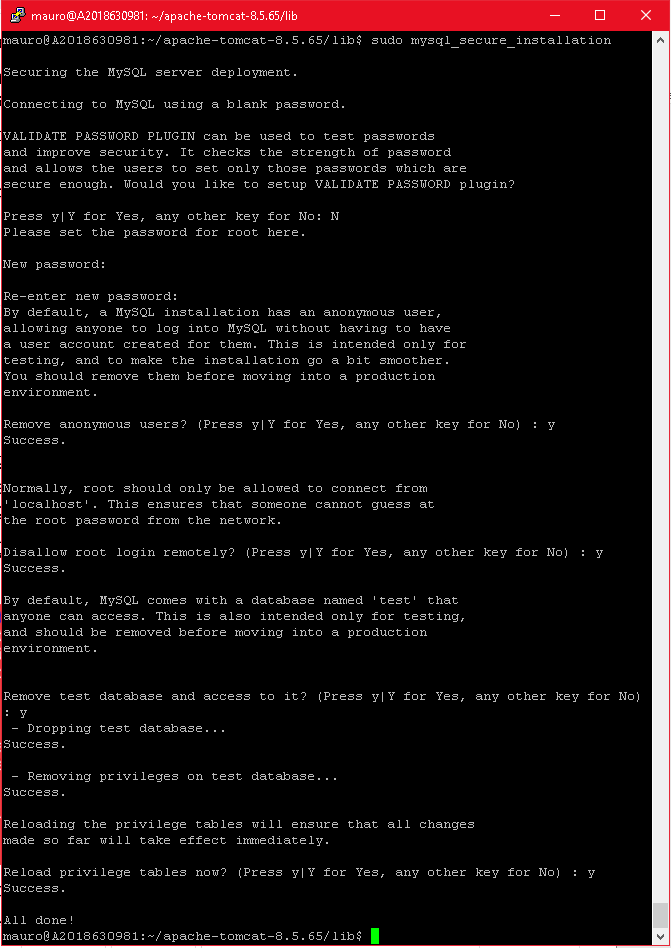
sudo mysql\_secure\_installation

Press y|Y for Yes, any other key for No: N

New password: contraseña-de-root-en-mysql

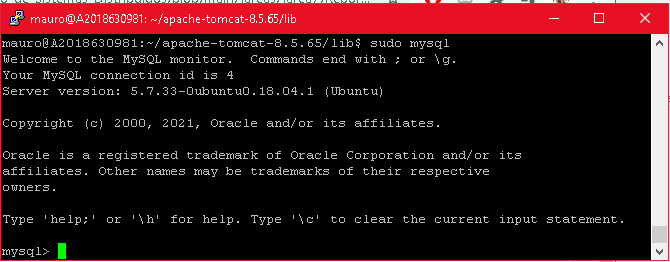
Re-enter new password: contraseña-de-root-en-mysql

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y  
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y  
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y  
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y



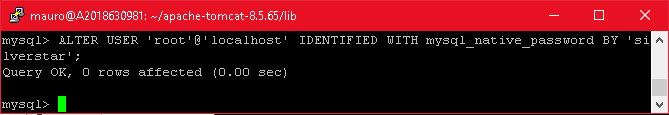
1. Ejecutar el monitor de MySQL:

sudo mysql



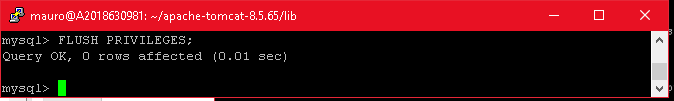
1. Ejecutar el siguiente comando SQL para modificar la contraseña de root:

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'contraseña-de-root-en-mysql';



1. Actualizar los privilegios:

FLUSH PRIVILEGES;



1. Ejecutar el siguiente comando para salir del monitor de MySQL:

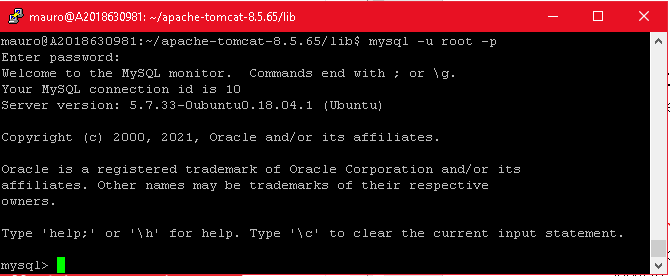
Quit



**Crear un usuario en MySQL**

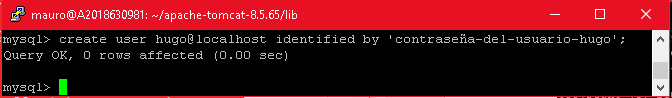
1. Ejecutar el monitor de MySQL:

mysql -u root -p



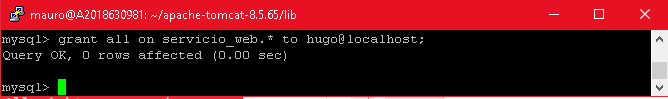
1. Crea el usuario "hugo":

create user hugo@localhost identified by 'contraseña-del-usuario-hugo';



1. Otorgar todos los permisos al usuario "hugo" sobre la base de datos "servicio\_web":

grant all on servicio\_web.\* to hugo@localhost;



1. Ejecutar el siguiente comando para salir del monitor de MySQL:

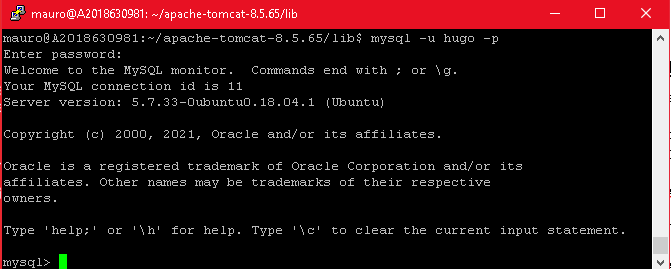
Quit



**Crear la base de datos**

1. Ejecutar el monitor de MySQL (notar que ahora se utiliza el usuario "hugo"):

mysql -u hugo -p



1. Crear la base de datos "servicio\_web":

create database servicio\_web;



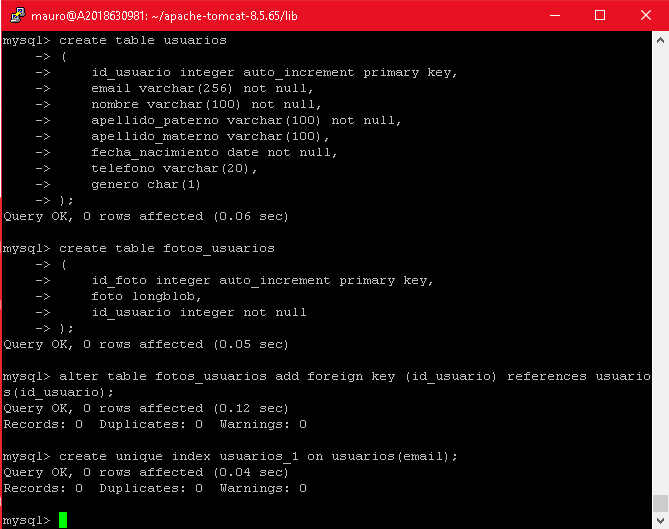
1. Conectar a la base de datos creada anteriormente:

use servicio\_web;



1. Crear las tablas "usuarios" y "fotos\_usuarios", así mismo, se crea una regla de integridad referencial y un índice único:

create table usuarios  
(  
    id\_usuario integer auto\_increment primary key,  
    email varchar(256) not null,  
    nombre varchar(100) not null,  
    apellido\_paterno varchar(100) not null,  
    apellido\_materno varchar(100),  
    fecha\_nacimiento date not null,  
    telefono varchar(20),  
    genero char(1)  
);  
create table fotos\_usuarios  
(  
    id\_foto integer auto\_increment primary key,  
    foto longblob,  
    id\_usuario integer not null  
);  
alter table fotos\_usuarios add foreign key (id\_usuario) references usuarios(id\_usuario);  
create unique index usuarios\_1 on usuarios(email);



1. Salir del monitor de MySQL:

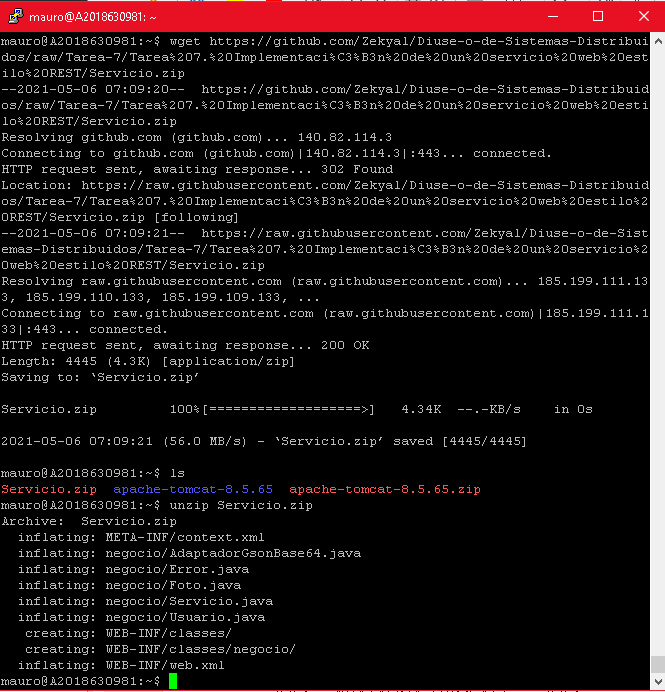
Quit



**Compilar, empacar y desplegar el servicio web**

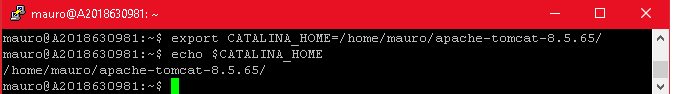
1. Descargar de la plataforma y desempacar el archivo [Servicio.zip](https://m4gm.com/moodle/mod/resource/view.php?id=1842).

Para este paso se usará el comando “wget”, con el cuál se exportará el archivo “[Servicio.zip](https://m4gm.com/moodle/mod/resource/view.php?id=1842)” el cuál está alojado en un repositorio. Posteriormente descomprimimos con el comando “unzip”.

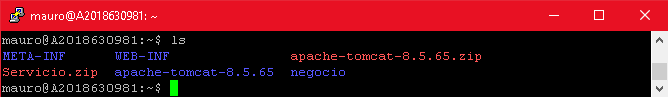


1. Definir la variable de ambiente CATALINA\_HOME:

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



1. Cambiar al directorio dónde se desempacó el archivo [Servicio.zip](https://m4gm.com/moodle/mod/resource/view.php?id=1842) (en ese directorio se encuentra el directorio "negocio").



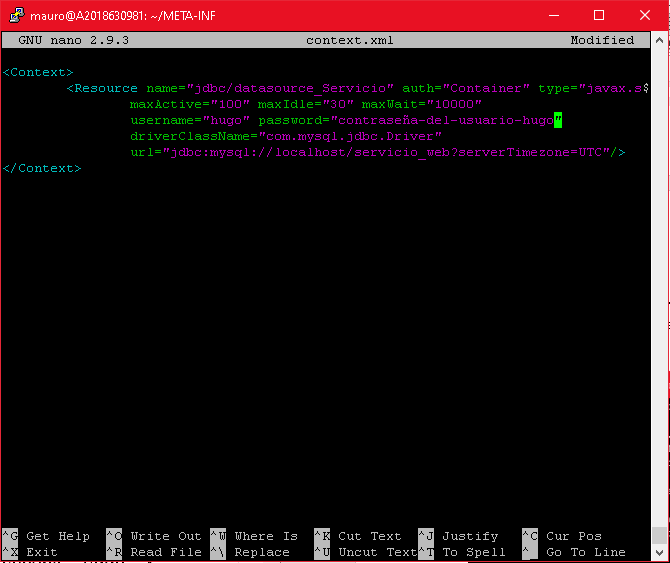
1. Compilar la clase Servicio.java:

javac -cp $CATALINA\_HOME/lib/javax.ws.rs-api-2.0.1.jar:$CATALINA\_HOME/lib/gson-2.3.1.jar:. negocio/Servicio.java



1. Editar el archivo "context.xml" que está en el directorio "META-INF" y definir el username de la base de datos y el password correspondiente. El usuario "hugo" fue creado en el paso 2 de la sección Crear un usuario en MySQL.

Para realizar esta modificación se utiliza el comando “nano context.xml” una vez estemos en el directorio “META.INF”. Una vez realizada dicha modificación guardamos los cambios y salimos.

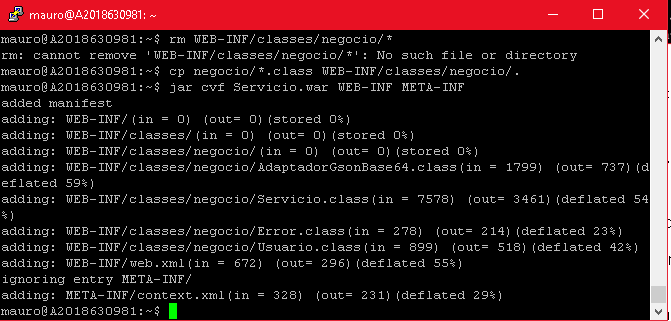


1. Ejecutar los siguientes comandos para crear el servicio web para Tomcat (notar que los servicios web para Tomcat son archivos JAR con la extensión .war):

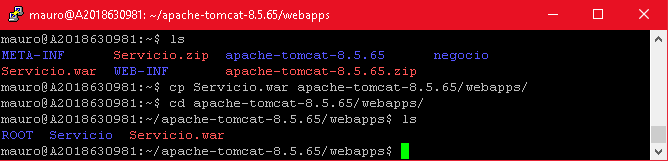
rm WEB-INF/classes/negocio/\*

cp negocio/\*.class WEB-INF/classes/negocio/.

jar cvf Servicio.war WEB-INF META-INF



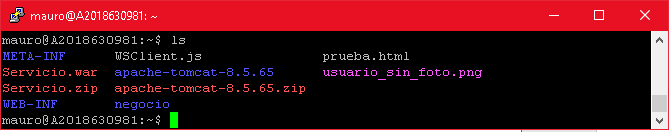
1. Para desplegar (deploy) el servicio web, copiar el archivo Servicio.war al directorio "webapps" de Tomcat. Notar que Tomcat desempaca automáticamente los archivos con extensión .war que se encuentran en el directorio webapps de Tomcat.



Para eliminar el servicio web se deberá eliminar el archivo "Servicio.war" y el directorio "Servicio", en éste orden.

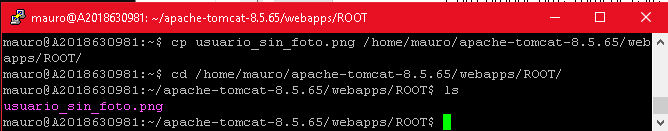
Cada vez que se modifique el archivo Servicio.java se deberá compilar, generar el archivo Servicio.war, borrar el archivo Servicio.war y el directorio Servicio del directorio webapps de Tomcat, y copiar el archivo Servicio.war al directorio webapps de Tomcat.

Antes de dar continuidad con el procedimiento, se realizará la exportación de los archivos “WSClient.js”, “prueba.html” y “usuario\_sin\_foto.png” a la máquina virtual por medio del comando “wget” desde un repositorio en el cuál dichos archivos fueron posteriormente guardados.



**Probar el servicio web utilizando HTML-Javascript**

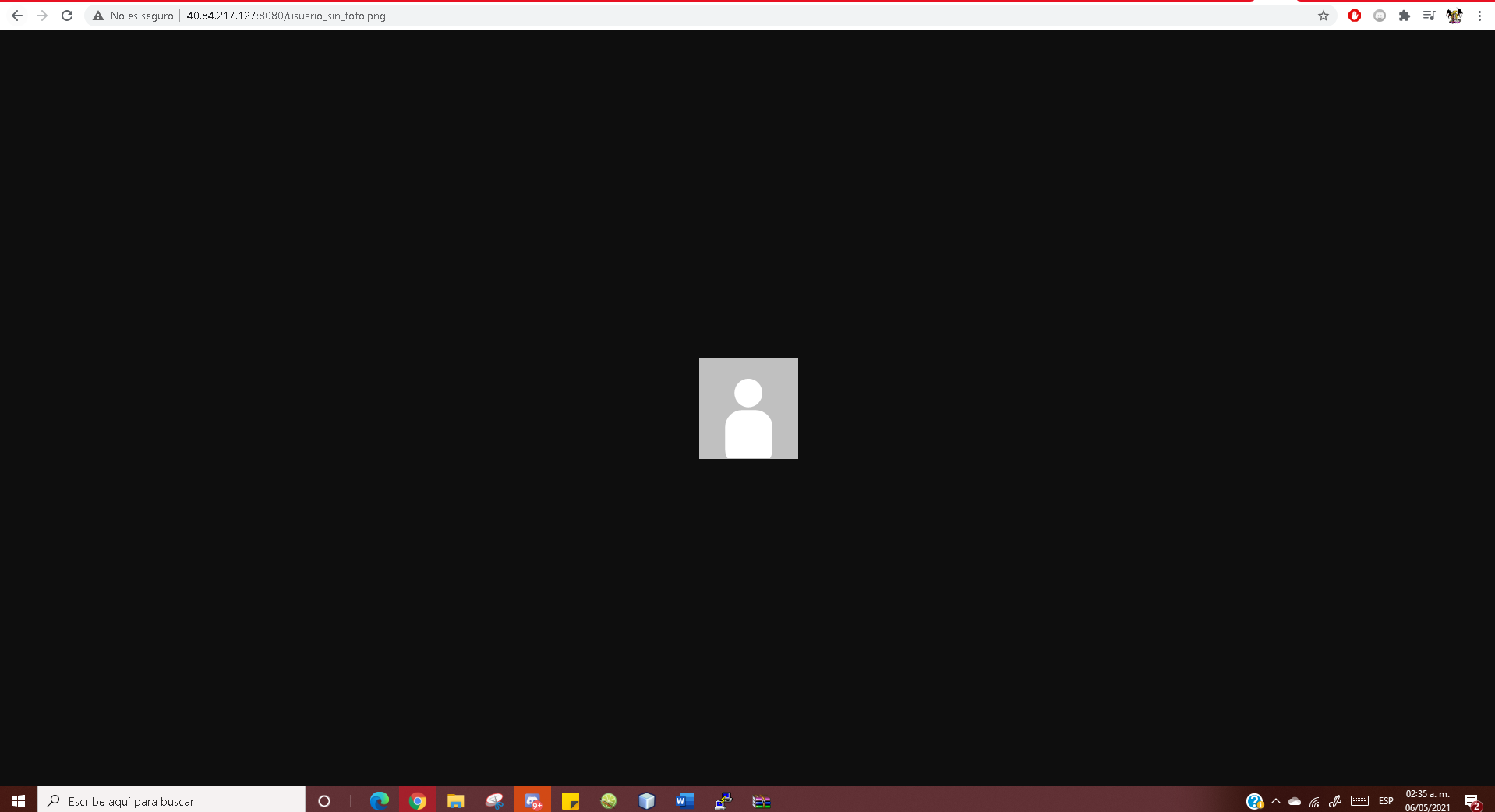
1. Copiar el archivo [usuario\_sin\_foto.png](https://m4gm.com/moodle/mod/resource/view.php?id=1845) al subdirectorio webapps/ROOT de Tomcat.



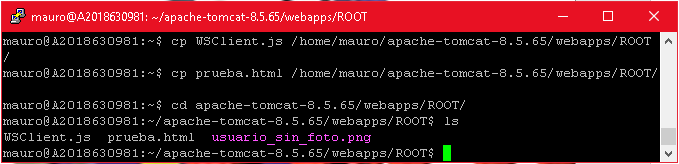
Notar que todos los archivos que se encuentran en el directorio webapps/ROOT de Tomcat son accesibles públicamente.

Para probar que Tomcat esté en línea y el puerto 8080 esté abierto, ingresar la siguiente URL en un navegador:

http://ip-de-la-máquina-virtual:8080/[usuario\_sin\_foto.png](https://m4gm.com/moodle/mod/resource/view.php?id=1845)

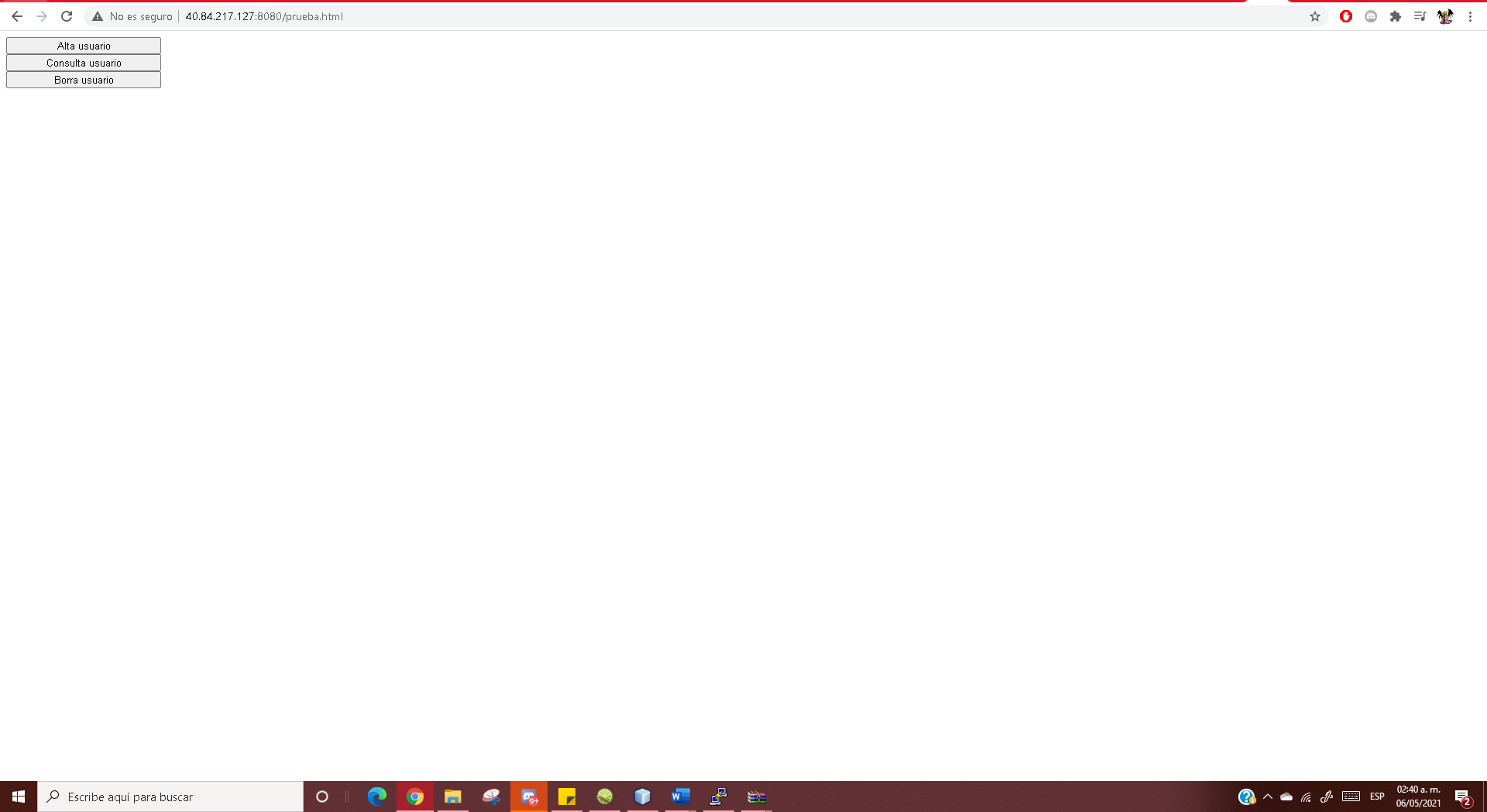


1. Copiar el archivo [WSClient.js](https://m4gm.com/moodle/mod/resource/view.php?id=1844) al directorio webapps/ROOT de Tomcat.
2. Copiar el archivo [prueba.html](https://m4gm.com/moodle/mod/resource/view.php?id=1843) al directorio webapps/ROOT de Tomcat.

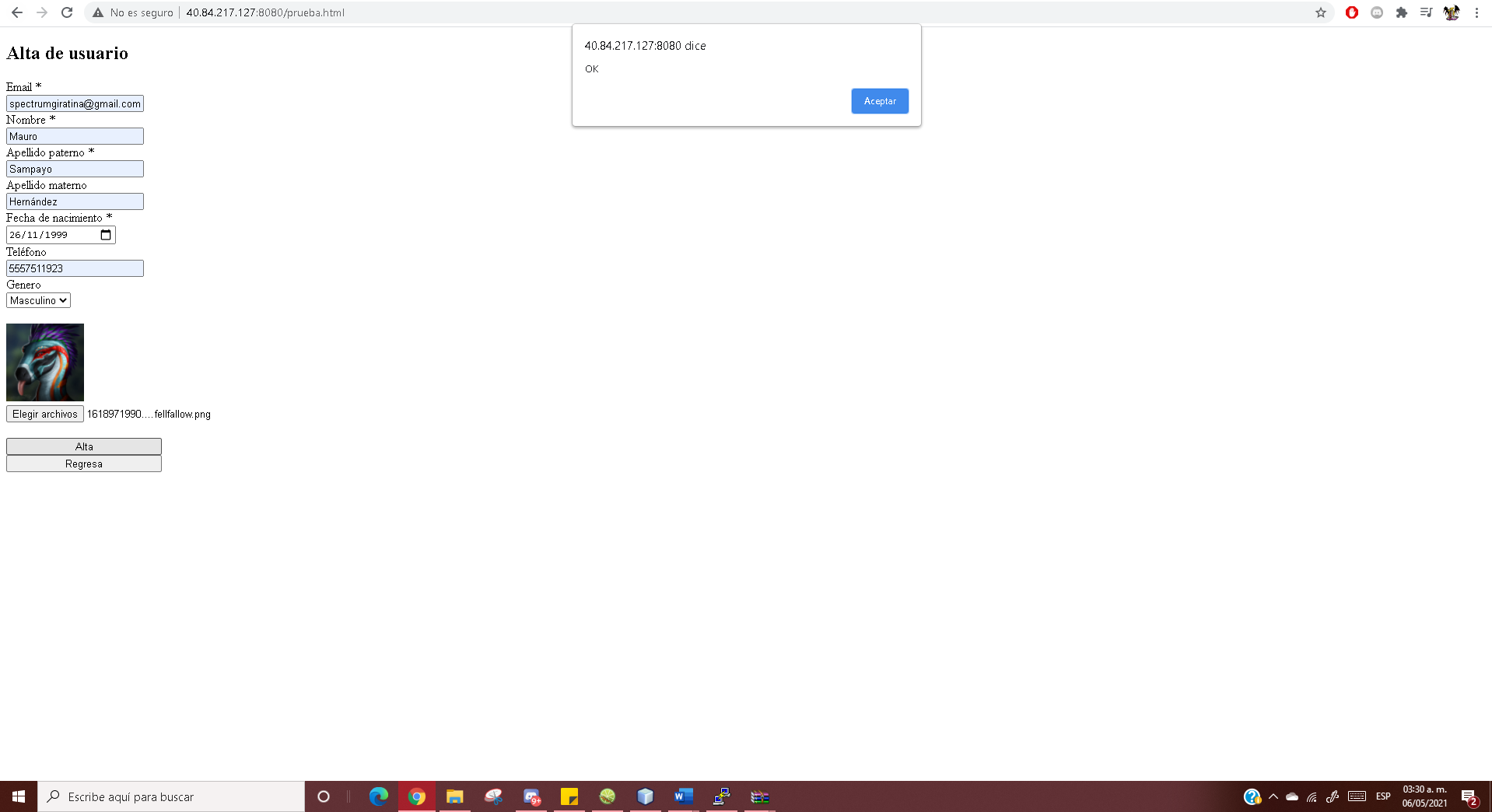


1. Ingresar la siguiente URL en un navegador:

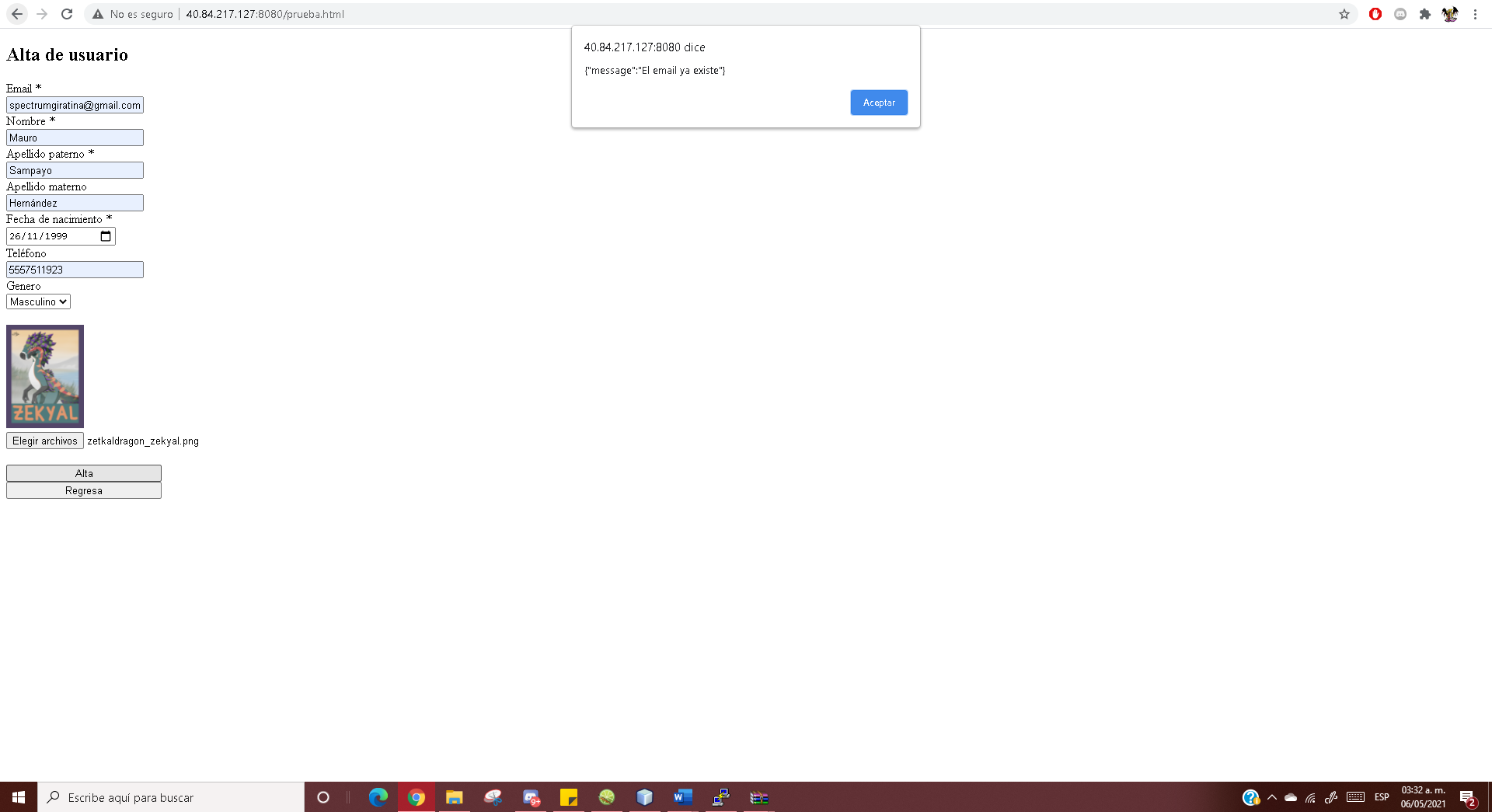
http://ip-de-la-máquina-virtual:8080/[prueba.html](https://m4gm.com/moodle/mod/resource/view.php?id=1843)



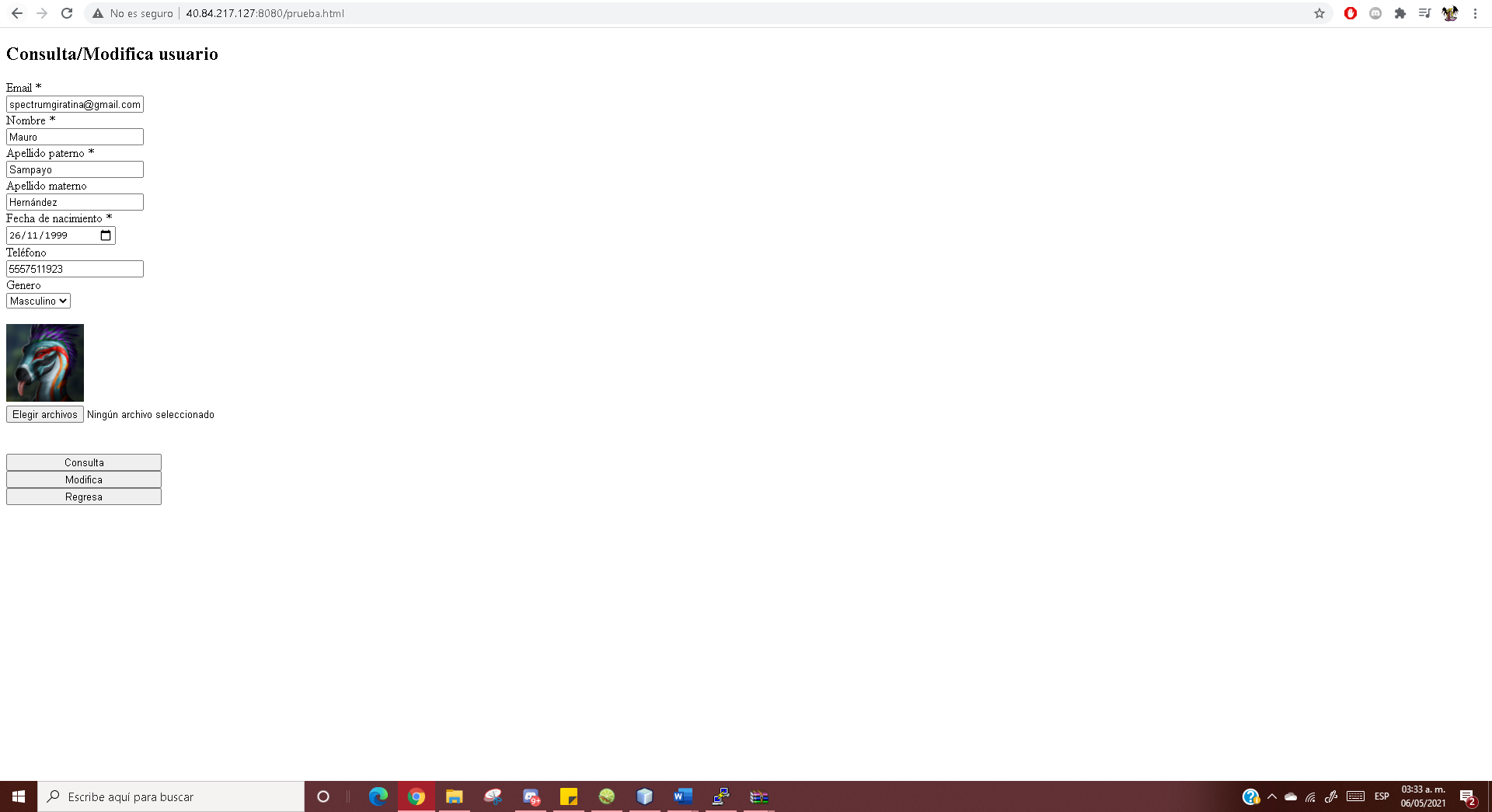
1. Dar clic en el botón “Alta usuario” para dar de alta un nuevo usuario. Capturar los campos y dar clic en el botón “Alta”.



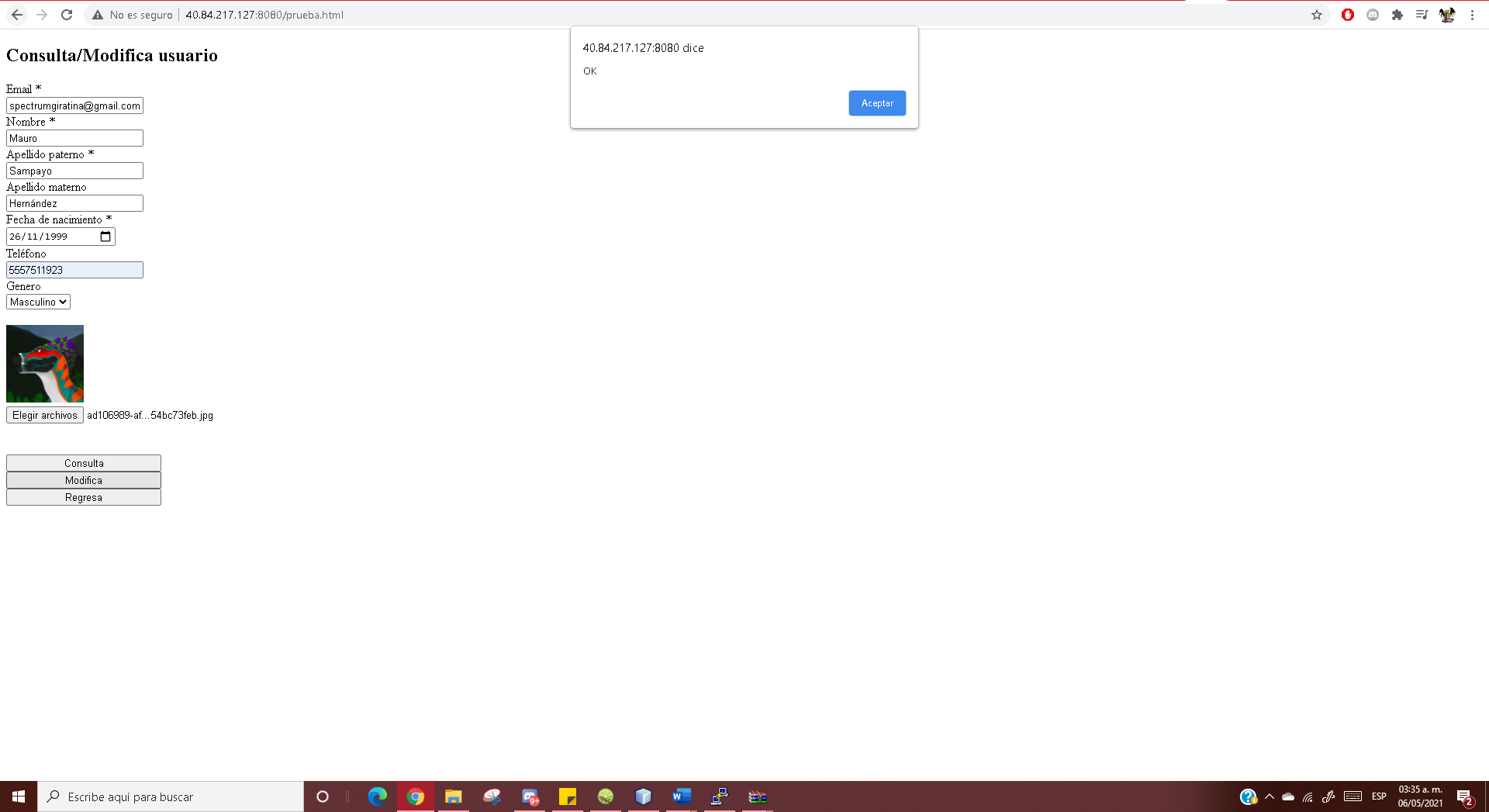
1. Intentar dar de alta otro usuario con el mismo email (se deberá mostrar una ventana de error indicando que el email ya existe)



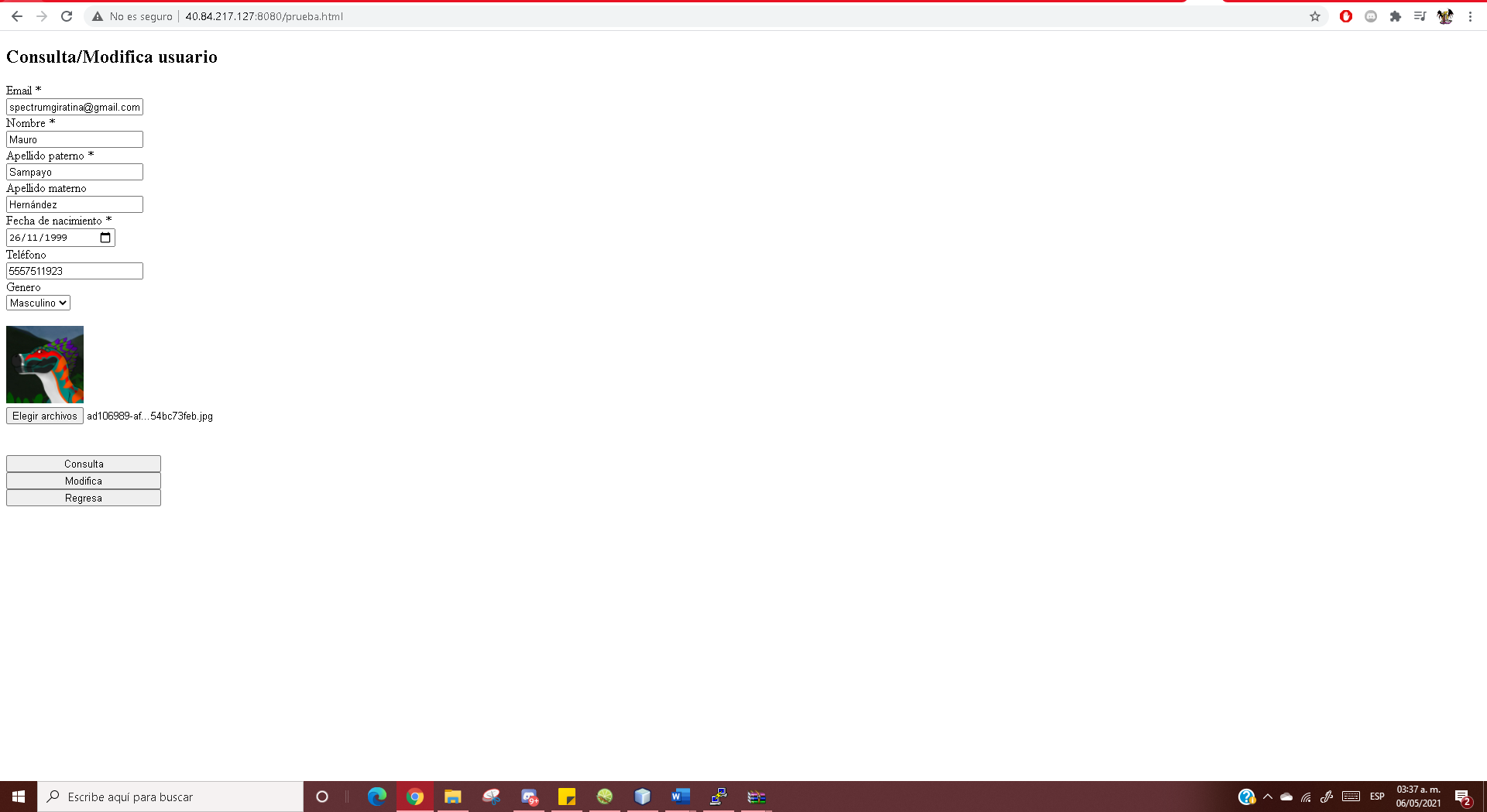
1. Dar clic en el botón “Consulta usuario” para consultar el usuario dado de alta en el paso 5.  Capturar el email y dar clic en el botón “Consulta”.



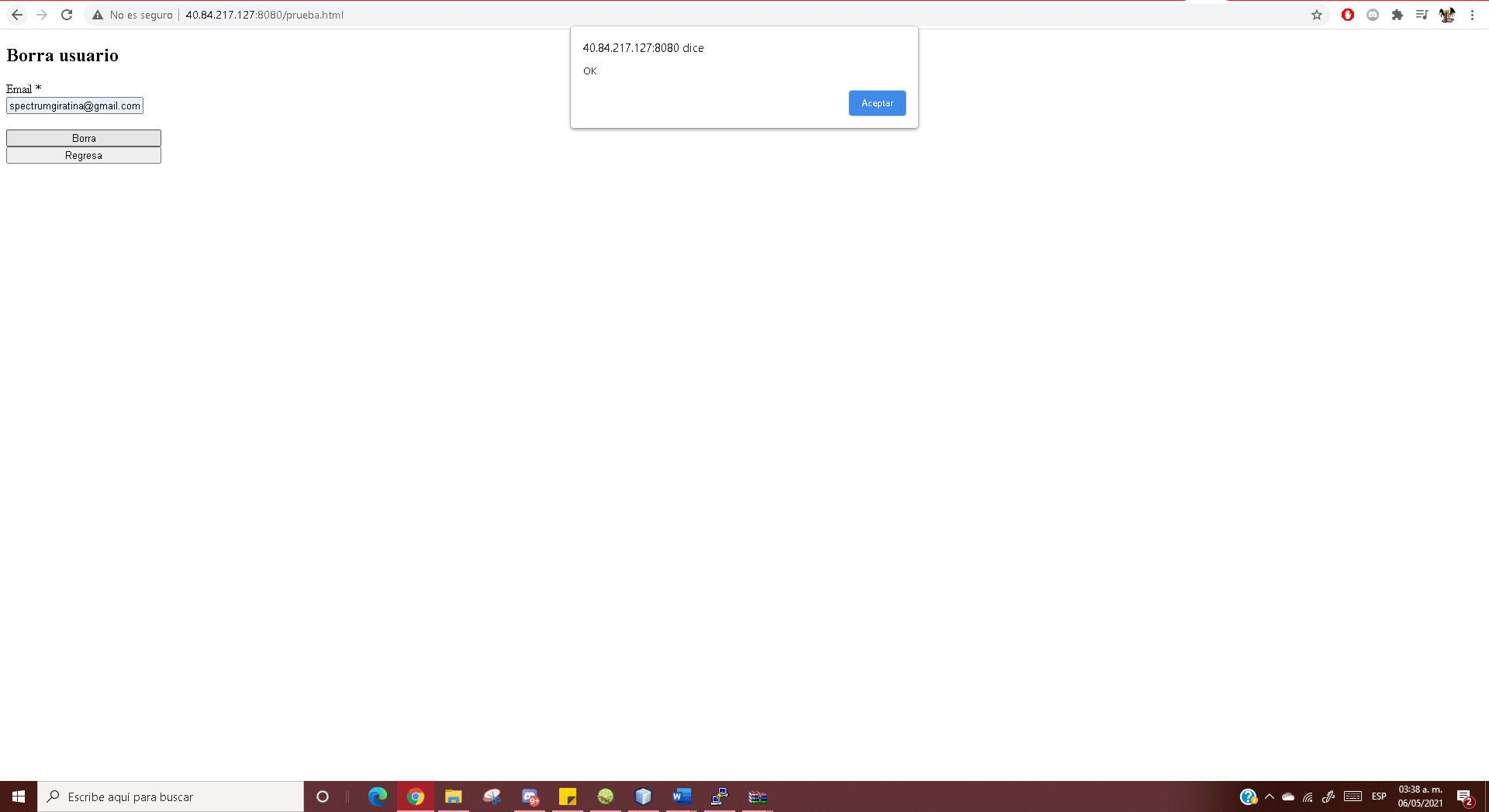
1. Modificar algún dato del usuario y dar clic en el botón “Modifica”:

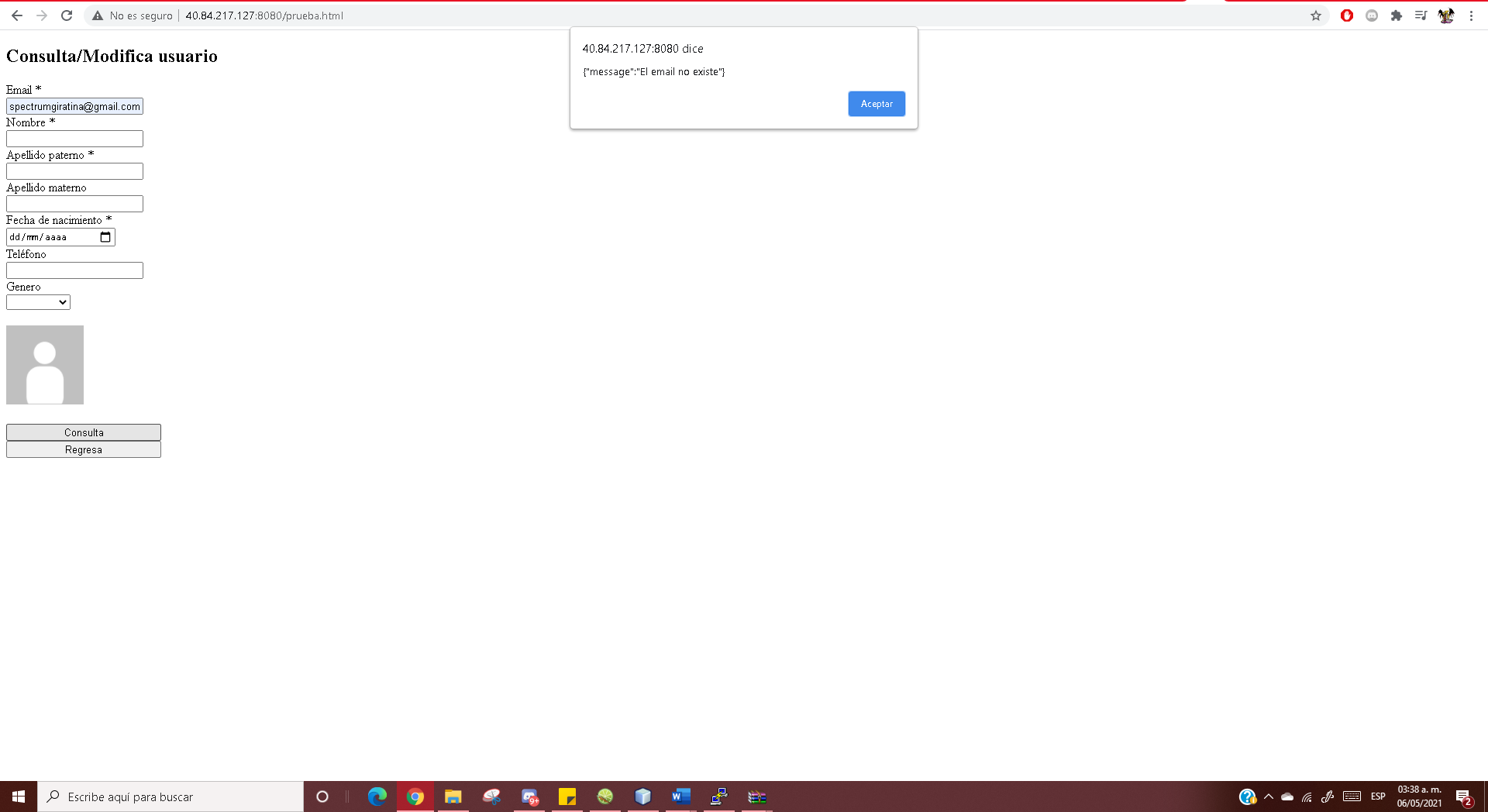


1. Recargar la página actual y consultar el usuario modificado, para verificar que la modificación se realizó.



1. Dar clic en el botón “Borra usuario” para borrar el usuario. Capturar el email del usuario a borrar y dar clic en el botón “Consulta”.





**Conclusión:**

Las aplicaciones de tipo REST resultan ser una herramienta bastante útil, pues con estas se puede lograr ejecutar un sistema distribuido como si esta fuese una sola aplicación. Esto se pudo ver claramente cuando se montó el servidor TomCat, la base de datos de MySQL y los programas de back-end y front-end de la página web en una misma máquina virtual, haciendo que todas estos elementos trabajarán a la vez en esta misma y se complementarán entre ellas.