



# Instituto Politécnico Nacional Escuela Superior de Cómputo

Desarrollo de Sistemas Distribuidos

Actividad: Implementación de un servicio web estilo REST

Curso impartido por el profesor: Pineda Guerrero Carlos

Grupo: 4CV1

21/01

Alumno: Adrian González Pardo



Ultima fecha modificado: 25 de noviembre de 2020

# 1. Desarrollo

Para esta practica solo fue necesario seguir los pasos que dejo escrito el profesor así como el escribir todo de forma natural para la ejecución de la aplicación, o en su defecto desarrollar un script que realice todas las tareas necesarias que permitan la automatización de la tarea a resolver, que es la instalación de todos los modulos (JDK 8, MySQL, Apache Tomcat, Webservice para REST)

## 2. Códigos y scripts

### 2.1. Script para copia de datos a VM

```
1 #!/usr/bin/env bash
2 # @author Adrian Gonzalez Pardo
3 ip="157.55.200.127"
4 echo "Conexion de emparejamiento de llaves rsa"
5 ssh adrian@$ip
6 echo "Envio de datos"
7 sshpass -p "adrianPardo_99" scp *.jar *.zip *.png *.js *.html *.sql script_rest.
  sh adrian@$ip:~/
8 echo -e "Conexion por primera vez a la VM\nEjecutar script_rest.sh para preparar
  toda la VM"
9 sshpass -p "adrianPardo_99" ssh adrian@$ip
```

### 2.2. Script para toda la automatización de crear la app web

```
1 #!/usr/bin/env bash
2 # @author Adrian Gonzalez Pardo
3 echo "Actualizando sistema"
4 v=$(sudo apt update)
5 echo "Instalando JDK8 y MySQL Server"
6 v=$(sudo apt install openjdk-8-jdk-headless mysql-server unzip -y)
7 echo "Extrae todos los archivos zip"
8 v=$(unzip apache-tomcat-8.5.60.zip)
9 v=$(unzip Servicio.zip)
10 v=$(unzip jaxrs-ri-2.24.zip)
11 v=$(unzip mysql-connector-java-8.0.22.zip)
12 echo "Elimina archivos innecesarios de tomcat"
13 rm -rf apache-tomcat-8.5.60/webapps/ROOT
14 echo "Crea directorio"
15 mkdir apache-tomcat-8.5.60/webapps/ROOT
16 echo "Copiando todos los archivos jar necesarios para el rest"
17 val=$(find ./jaxrs-ri | grep ".jar")
18 val1=$(find ./mysql-connector-java-8.0.22 | grep ".jar")
19 cp $val1 apache-tomcat-8.5.60/lib
20 cp $val apache-tomcat-8.5.60/lib
21 cp gson-2.3.1.jar apache-tomcat-8.5.60/lib
22 echo "Copia todos los archivos para la demostracion"
23 cp *.png *.js *.html apache-tomcat-8.5.60/webapps/ROOT
24 echo "Crea variables de entorno permanentes"
25 where=$(pwd)
26 export CATALINA_HOME=$(pwd)/apache-tomcat-8.5.60
27 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
28 echo -e "export CATALINA_HOME=${where}/apache-tomcat-8.5.60\nexport JAVA_HOME=/
  usr/lib/jvm/java-8-openjdk-amd64" >> ~/.bashrc
29 source ~/.bashrc
30 echo -e "Llenar los datos con:\n\n\ncontraseña-de-root-en-mysql\ncontraseña-de-
  root-en-mysql\nY\nY\nY\nY\n"
31 sudo mysql_secure_installation
32 echo -e "Configurando root"
33 sudo mysql -e "source conf_root.sql"
34 echo -e "Add new user (hugo)"
35 mysql -u root -pcontraseña-de-root-en-mysql -e "source add_hugo.sql"
36 echo -e "Create new table and conf"
```

```

37 mysql -u hugo -pcontraseña-del-usuario-hugo -e "source web_rest.sql"
38 echo "Compilacion de datos"
39 javac -cp $CATALINA_HOME/lib/javawebsocket-api-2.0.1.jar:$CATALINA_HOME/lib/gson
    -2.3.1.jar:. negocio/Servicio.java
40 echo "Modificacion de archivo context.xml"
41 echo "<Context>"
42     <Resource name="\jdbc/datasource_Servicio\" auth="\Container\" type="\javax.
        sql.DataSource\"
43         maxActive="\100\" maxIdle="\30\" maxWait="\10000\"
44         username="\hugo\" password="\contraseña-del-usuario-hugo\"
45         driverClassName="\com.mysql.jdbc.Driver\"
46         url="\jdbc:mysql://localhost/servicio_web?serverTimezone=UTC\"/>
47 </Context>" > META-INF/context.xml
48 echo "Creacion y modificacion de archivos web-inf"
49 rm WEB-INF/classes/negocio/*
50 cp negocio/*.class WEB-INF/classes/negocio/
51 jar cvf Servicio.war WEB-INF META-INF
52 echo "Copia del web rest al tomcat"
53 cp Servicio.war apache-tomcat-8.5.60/webapps/
54 echo "Ejecucion de la aplicacion"
55 sh $CATALINA_HOME/bin/catalina.sh start

```

## 2.3. Códigos de SQL que forman parte de script\_rest.sh

```

1 # @author Adrian Gonzalez Pardo
2 ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'contrase
    ña-de-root-en-mysql';
3 FLUSH PRIVILEGES;

```

```

1 # @author Adrian Gonzalez Pardo
2 create user hugo@localhost identified by 'contraseña-del-usuario-hugo';
3 grant all on servicio_web.* to hugo@localhost;

```

```

1 # @author Adrian Gonzalez Pardo
2 create database if not exists servicio_web;
3 use servicio_web;
4 create table if not exists 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));
5 create table if not exists fotos_usuarios( id_foto integer auto_increment
    primary key, foto longblob, id_usuario integer not null);
6 alter table fotos_usuarios add foreign key (id_usuario) references usuarios(
    id_usuario);
7 create unique index usuarios_1 on usuarios(email);

```

## 2.4. Código de llenado de datos con Python

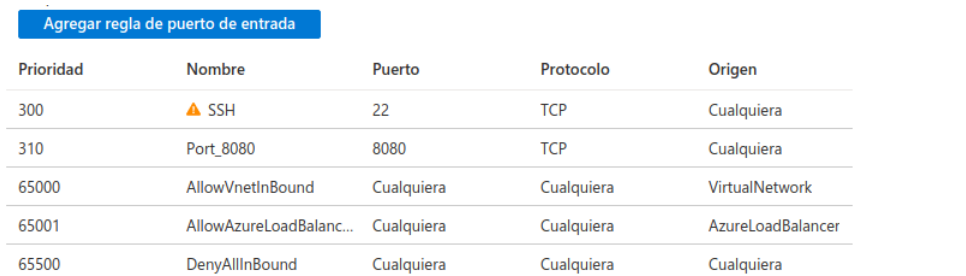
```

1 #!/usr/bin/env python3
2 # @author Adrian Gonzalez Pardo
3 import pyautogui, time
4 time.sleep(5)
5 pyautogui.typewrite("gozapaadr")
6 pyautogui.hotkey('@')
7 pyautogui.typewrite("gmail.com")
8 pyautogui.press("tab")
9 text=["Adrian","Gonzalez","Pardo","10051999","5518179786","M"]
10 for word in text:
11     pyautogui.typewrite(word)
12     pyautogui.press("tab")

```

Con esto finalmente solo es necesario abrir la aplicación web desde nuestro navegador, destacando que hay que dar un forwarding del puerto 8080

### 3. Capturas



The screenshot shows a web interface for configuring VM ports. At the top, there is a blue button labeled "Agregar regla de puerto de entrada". Below it is a table with five columns: "Prioridad", "Nombre", "Puerto", "Protocolo", and "Origen". The table contains five rows of configuration rules.

Prioridad	Nombre	Puerto	Protocolo	Origen
300	🚩 SSH	22	TCP	Cualquiera
310	Port_8080	8080	TCP	Cualquiera
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork
65001	AllowAzureLoadBalanc...	Cualquiera	Cualquiera	AzureLoadBalancer
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera

*Figura 1: Pantalla de configuración de puertos de la VM*

```
adrian@azureRestJava: ~  
adrian@azureRestJava: ~ 169x41  
d3vcr4ck at illBewWithYou in ~/D/d/restApp  
bash script_cp.sh  
Conexion de emparejamiento de llaves rsa. servicios y documentos (G+)  
The authenticity of host '157.55.200.127 (157.55.200.127)' can't be established.  
ECDSA key fingerprint is SHA256:Cbq0K05y/iSYTEkUIGdcudGpwsdVKQWp+a5/7II9MWA.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '157.55.200.127' (ECDSA) to the list of known hosts.  
adrian@157.55.200.127's password:  
Permission denied, please try again.  
adrian@157.55.200.127's password:  
Permission denied, please try again.  
adrian@157.55.200.127's password:  
adrian@157.55.200.127: Permission denied (publickey,password).  
Envío de datos  
Conexion por primera vez a la VM  
Ejecutar script rest.sh para preparar toda la VM  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1031-azure x86_64)  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
System information as of Thu Nov 26 01:55:32 UTC 2020 : Haga clic aquí para agregar etiquetas.  
System load: 0.77 Processes: 116  
Usage of /: 4.6% of 28.90GB Users logged in: supervisión  
Memory usage: 20% IP address for eth0: 10.0.0.4  
Swap usage: 0%  
Máquina virtual  
Nombre del equipo: azureRestJava  
Sistema operativo: Linux  
Publicador: Canonical  
Oferta: UbuntuServer  
Plan: 18.04-LTS  
Estado del agente: Not Ready  
Versión del agente: Unknown  
Redes  
Dirección IP pública: 157.55.200.127  
Dirección IP pública (IPv6): -  
Dirección IP privada: 10.0.0.4  
Dirección IP privada (IPv6): -  
Red virtual/subred: Distribuidos-vnet/default  
Nombre DNS: Configurar  
Tamaño  
0 packages can be updated.  
0 updates, are security updates.  
New release 20.04.1 LTS is available.  
Run 'do-release-upgrade' to upgrade to it.  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo root" for details.  
adrian@azureRestJava:~$
```

Figura 2: Ejecución del primer script para conectarse con la VM

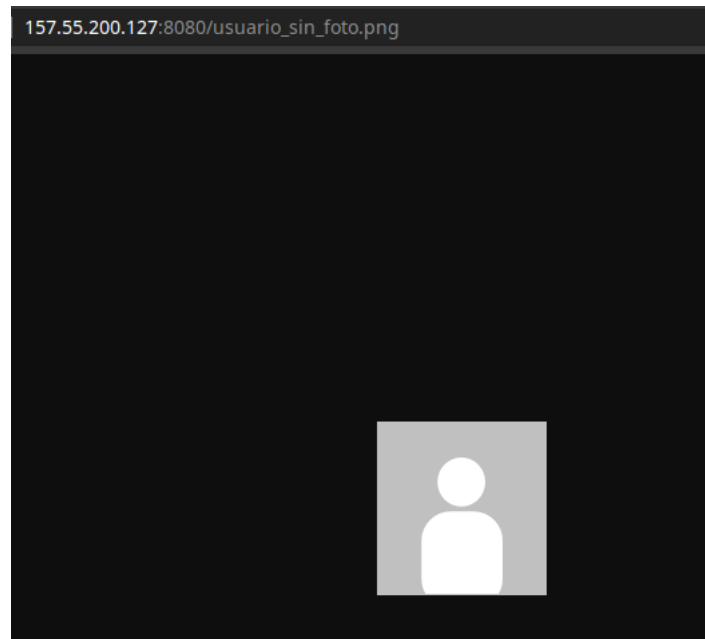
```
adrian@azureRestJava: ~  
adrian@azureRestJava: ~ 169x41  
- Removing privileges on test database...  
Success. Microsoft Azure  
Reloading the privilege tables will ensure that all changes  
made so far will take effect immediately.  
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y  
Success.  
All done!  
Configurando root  
Add new user: (hugo)al  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Create new table and conf  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Compilacion de datos  
Modificacion de archivo context.xml  
Creacion y modificacion de archivos web-inf  
rm: cannot remove 'WEB-INF/classes/negocio/*': No such file or directory  
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 = 899) (out= 518)(deflated 42%)  
adding: WEB-INF/classes/negocio/AdaptadorGsonBase64.class(in = 1799) (out= 737)(deflated 59%)  
adding: WEB-INF/classes/negocio/Error.class(in = 278) (out= 214)(deflated 23%)  
adding: WEB-INF/classes/negocio/Servicio.class(in = 7578) (out= 3462)(deflated 54%)  
ignoring entry META-INF/  
adding: META-INF/context.xml(in = 301) (out= 222)(deflated 26%)  
Copia del web rest al tomcat  
Ejecucion de la aplicacion  
Using CATALINA_BASE: /home/adrian/apache-tomcat-8.5.60  
Using CATALINA_HOME: /home/adrian/apache-tomcat-8.5.60  
Using CATALINA_TMPDIR: /home/adrian/apache-tomcat-8.5.60/temp  
Using JRE_HOME: /usr/lib/jvm/java-8-openjdk-amd64  
Using CLASSPATH: /home/adrian/apache-tomcat-8.5.60/bin/bootstrap.jar:/home/adrian/apache-tomcat-8.5.60/bin/tomcat-juli.jar  
Using CATALINA_OPTS:  
Tomcat started.  
adrian@azureRestJava: ~$
```

Regla de seguridad creada 20:02  
La regla de seguridad 'Tomcat' se creó correctamente.

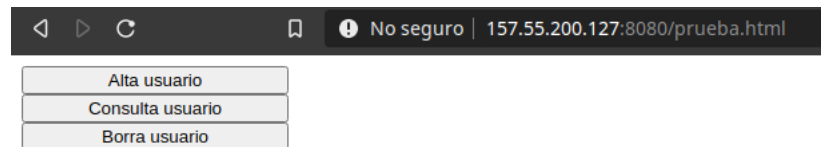
Reglas de seguridad vigentes

Regla de seguridad	Protocolo	Origen	Destino	Acción
Regla de seguridad de entrada	TCP	Cualquiera	Cualquiera	Permitir
Reglas de puerto de salida	Cualquiera	VirtualNetwork	VirtualNetwork	Permitir
Grupos de seguridad de aplicación	Cualquiera	AzureLoadBalancer	Cualquiera	Permitir
Equilibrio de carga	Cualquiera	Cualquiera	Cualquiera	Denegar

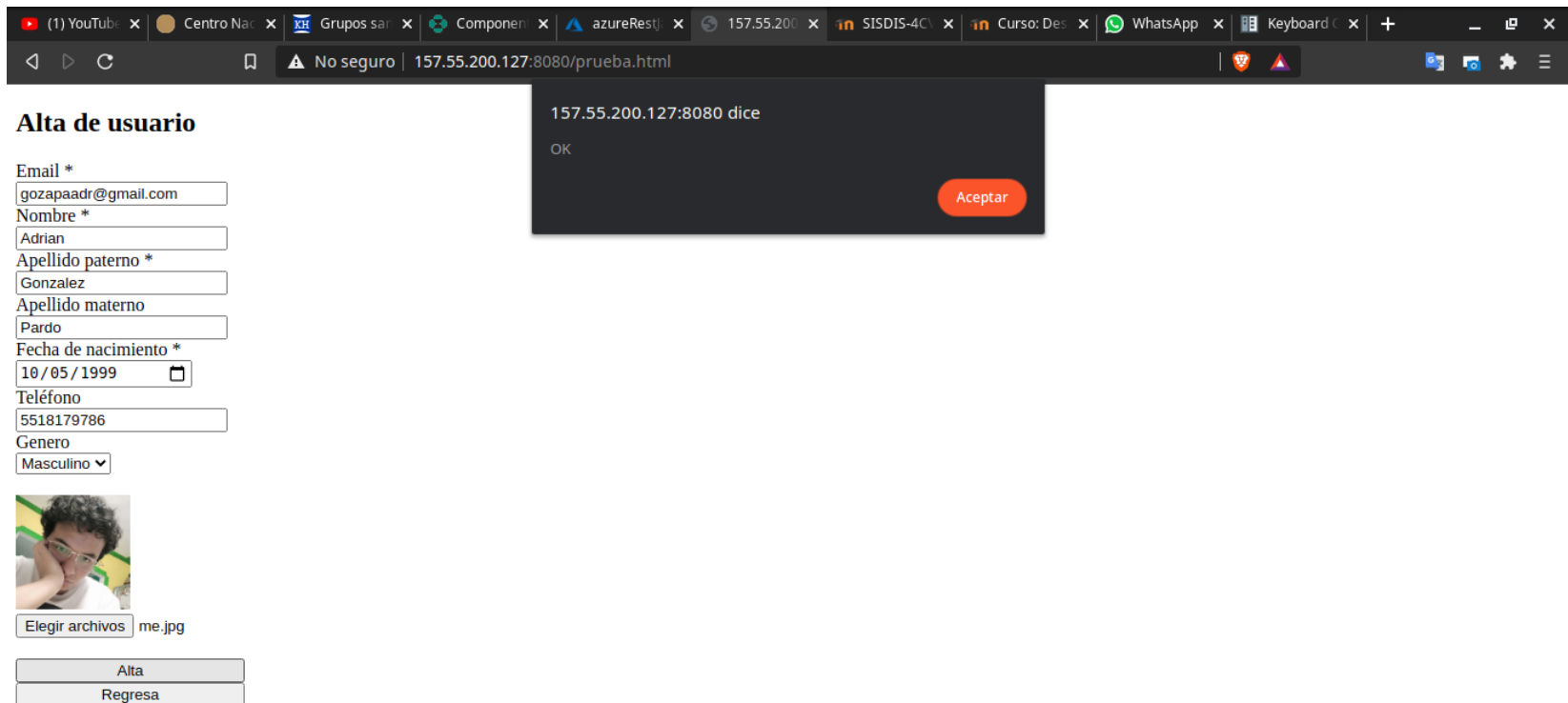
Figura 3: Ejecución del segundo script para instalar todo la aplicación REST



*Figura 4: Primer parte de la aplicación REST*



*Figura 5: Segunda parte de la aplicación REST vista de los botones*



*Figura 6: Segunda parte de la aplicación REST captura de datos de forma automatica*



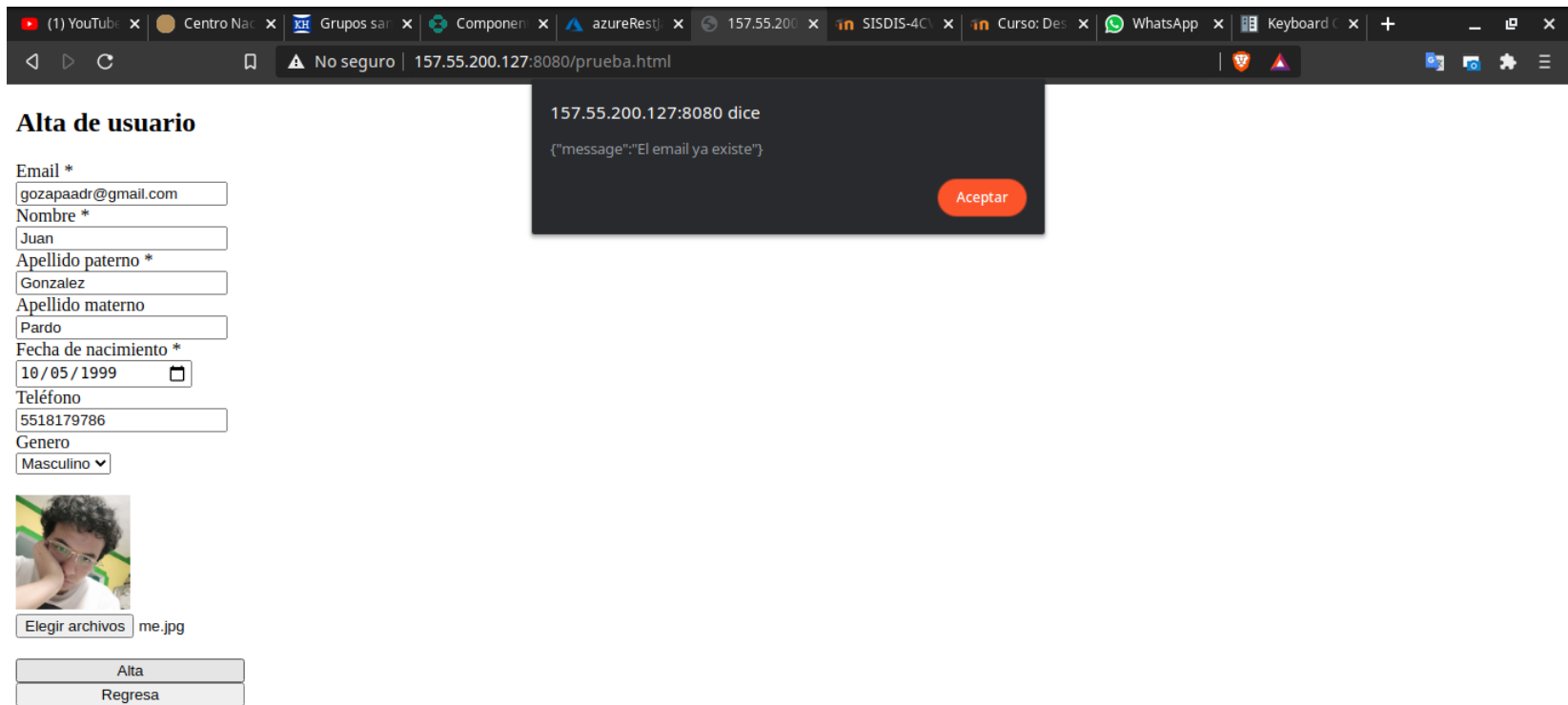
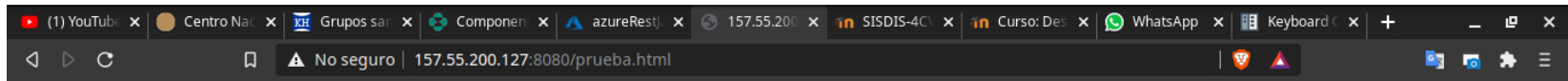


Figura 7: Segunda parte de la aplicación REST captura de datos con redundancia en el correo (Cambia nombre)



## Consulta/Modifica usuario

Email \*

Nombre \*

Apellido paterno \*

Apellido materno

Fecha de nacimiento \*

Teléfono

Genero

Masculino ▾



Elegir archivos

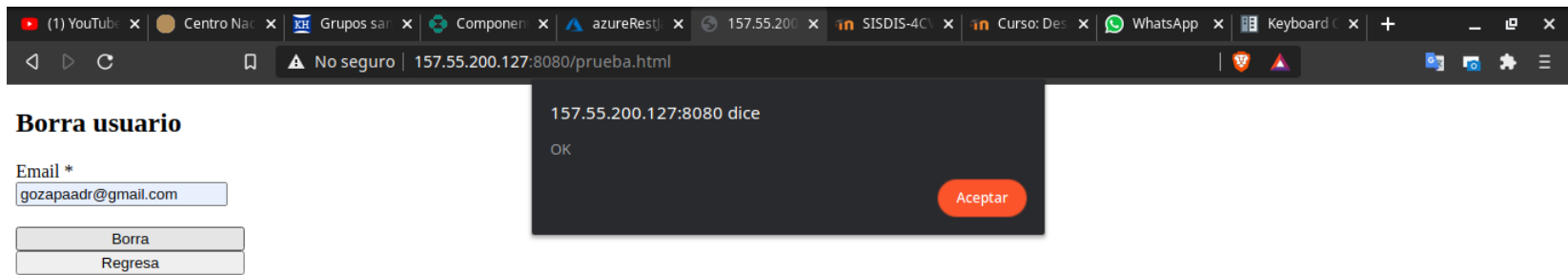
No se eligió archivo

Consulta

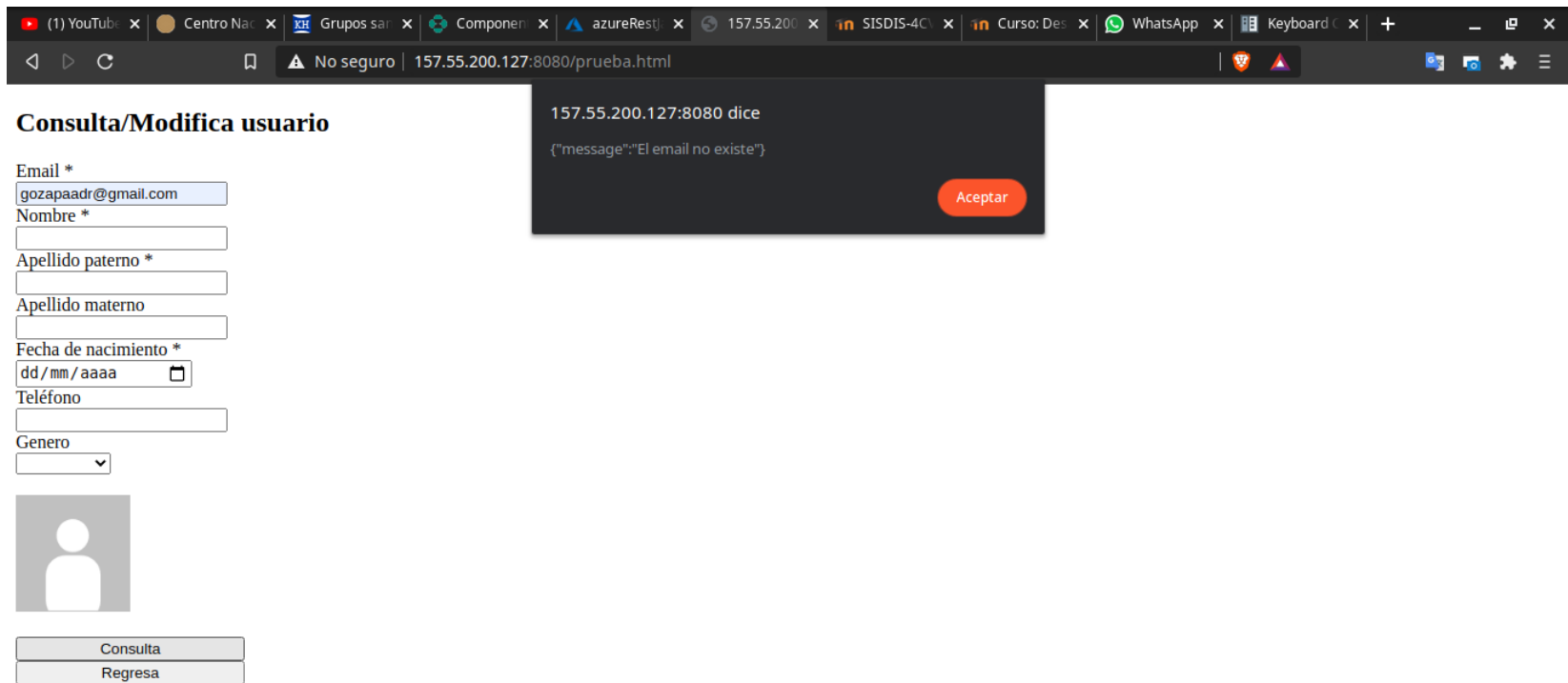
Modifica

Regresa

*Figura 8: Segunda parte de la aplicación REST consulta de datos*



*Figura 9: Segunda parte de la aplicación REST eliminacion de datos*



*Figura 10: Segunda parte de la aplicación REST consulta de datos despues de eliminar*

## 4. Conclusiones

El realizar aplicaciones basadas en webservices como lo es REST nos permite saber el como trabajan algunas aplicaciones a la hora de almacenar y manejar ciertos, datos, si bien Java puede ser visto como una alternativa para esto, tambien existen otros lenguajes en conjunto de sus frameworks como lo son Ruby on Rails, Sinatra (Ruby), Django (Python), Flask (Python), Go, entre otros lenguajes más que nos permiten hacer aplicaciones REST es importante como se realizaban con lenguajes un poco más antaños como lo es Java.