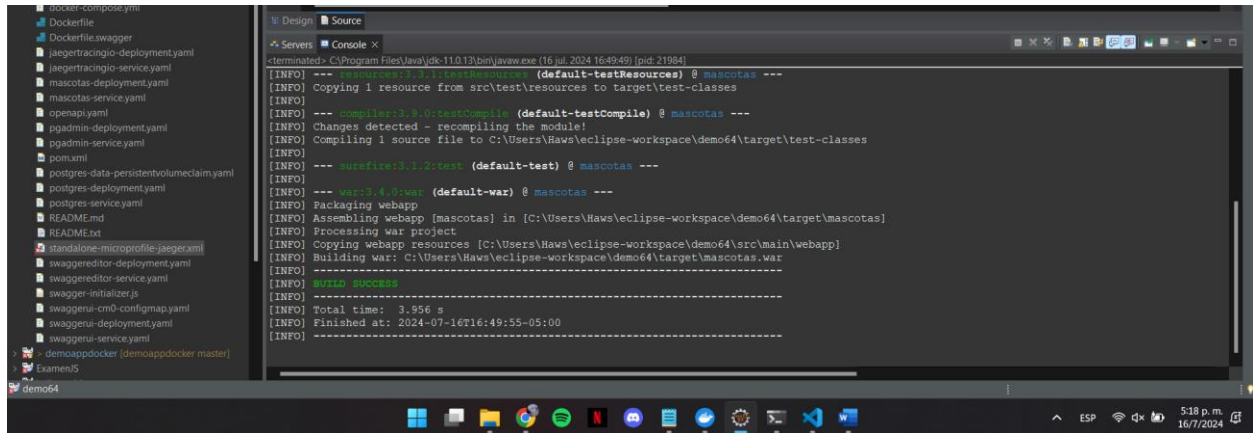


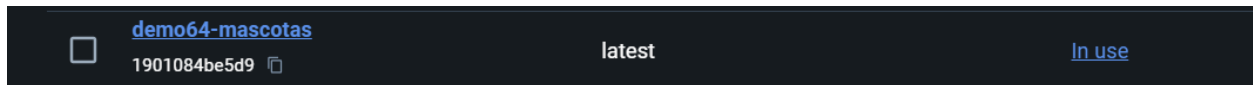
## Tarea 1: Desarrollo del Microservicio en Java (30 minutos)

<https://github.com/ElGat02/PruebaMicroservicios.git>

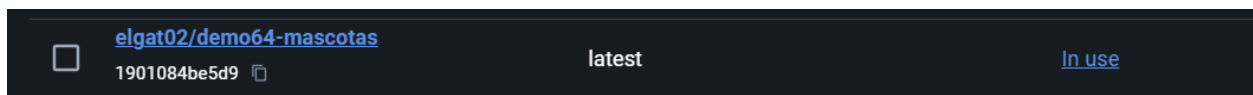


```
<terminated> C:\Program Files\Java\jdk-11.0.13\bin\java.exe (16 jul 2024 16:49:49) [pid: 21984]
[INFO] --- resources:3.1.0:resources (default-testResources) @ mascotas ---
[INFO] Copying 1 resource from src\test\resources to target\test-classes
[INFO] --- compiler:3.9.0:testCompile (default-testCompile) @ mascotas ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\Haws\workspace\demo64\target\test-classes
[INFO] --- surefire:3.1.2:test (default-test) @ mascotas ---
[INFO] --- war:3.4.0:war (default-war) @ mascotas ---
[INFO] Packaging webapp
[INFO] Assembling webapp [mascotas] in [C:\Users\Haws\workspace\demo64\target\mascotas]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\Haws\workspace\demo64\src\main\webapp]
[INFO] Building war: C:\Users\Haws\workspace\demo64\target\mascotas.war
[INFO] BUILD SUCCESS
[INFO] Total time: 3.956 s
[INFO] Finished at: 2024-07-16T16:49:55-05:00
[INFO]
```

## Tarea 2: Dockerización del Microservicio y la Base de Datos (20 minutos)



Luego le hice un pull de mi repo de dockerhub



Creación del Docker File



```

53
54 swaggerui:
55   depends_on:
56     - mascotas
57   image: swaggerapi/swagger-ui
58   environment:
59     SWAGGER_JSON: /api/openapi.yaml
60   ports:
61     - 8888:8080
62   volumes:
63     - ./openapi.yaml:/api/openapi.yaml
64   networks:
65     - datalogger_net
66
67 swaggereditor:
68   depends_on:
69     - mascotas
70   image: swaggerapi/swagger-editor
71   ports:
72     - 9999:8080
73   networks:
74     - datalogger_net
75

```

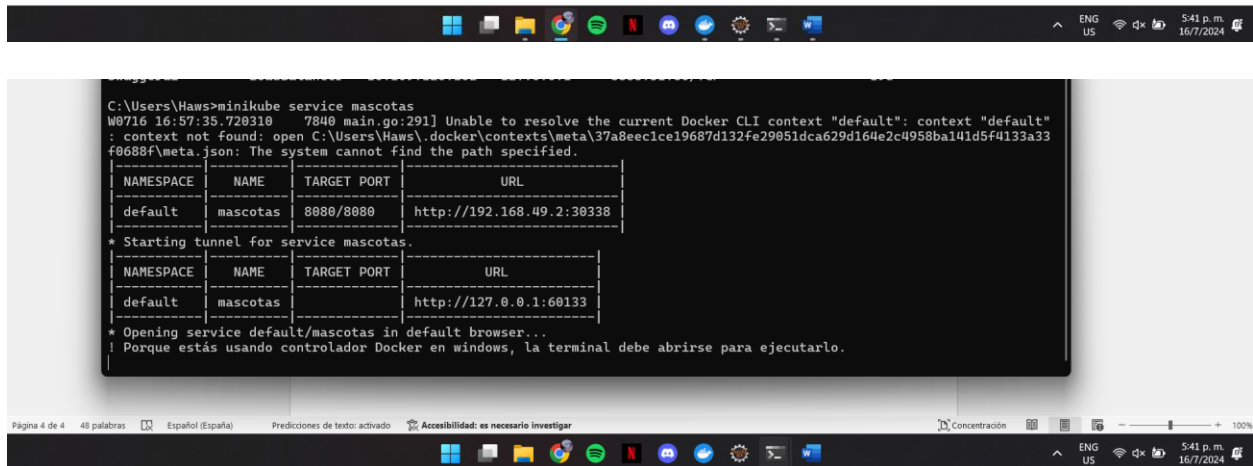
The screenshot shows an IDE window with a file named `postgres-deployment.yaml`. The file contains a Kubernetes Deployment manifest for PostgreSQL. The manifest includes metadata, annotations for kompose, labels, and a spec with a single replica. The container is named `postgres` and uses the `postgres` image. It includes environment variables for `POSTGRES_DB`, `POSTGRES_HOST`, `POSTGRES_PASSWORD`, and `POSTGRES_USER`.

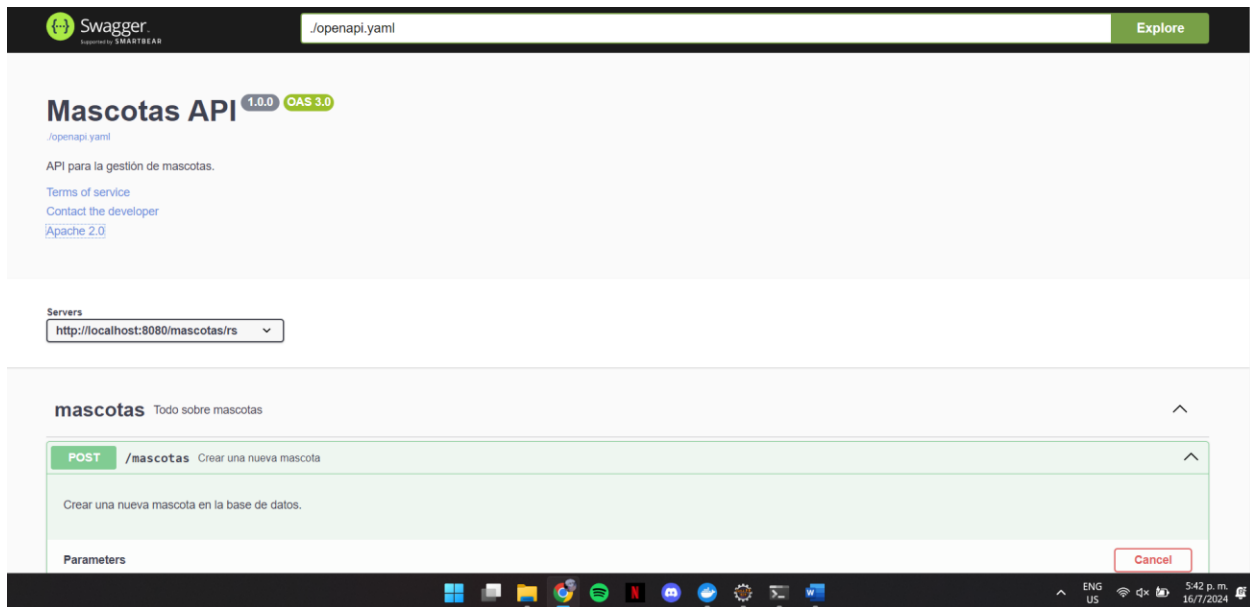
```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    annotations:
5      kompose.cmd: kompose convert
6      kompose.version: 1.34.0 (cbf2835db)
7    labels:
8      io.kompose.service: postgres
9    name: postgres
10 spec:
11   replicas: 1
12   selector:
13     matchLabels:
14       io.kompose.service: postgres
15   strategy:
16     type: Recreate
17   template:
18     metadata:
19       annotations:
20         kompose.cmd: kompose convert
21         kompose.version: 1.34.0 (cbf2835db)
22       labels:
23         io.kompose.service: postgres
24     spec:
25       containers:
26       - env:
27         - name: POSTGRES_DB
28           value: mascotasdb
29         - name: POSTGRES_HOST
30           value: basededatosmascotas
31         - name: POSTGRES_PASSWORD
32           value: postgres
33         - name: POSTGRES_USER
34           value: postgres
35       image: postgres
36       name: basededatosmascotas

```

Despliegue





## Agregar

Responses	
Code	Description
200	<div>Operación exitosa</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div><pre>{  "nombre": "Rex",  "especie": "Perro",  "raza": "Labrador"}  </pre></div>
400	<div>Error al guardar la mascota</div> <div>Media type</div> <div>application/json</div> <div>Example Value   Schema</div> <div><pre>{  "code": 0,  "message": "string"}  </pre></div>

## LISTAR

The screenshot shows a REST client interface with the following details:

- Curl:**

```
curl -X 'GET' \
  'http://localhost:8080/mascotas/rs/mascotas/list' \
  -H 'accept: application/json'
```
- Request URL:** `http://localhost:8080/mascotas/rs/mascotas/list`
- Server response:**

Code	Details
200	<p><b>Response body</b></p> <pre>{   "raza": "Labrador",   "especie": "Perro",   "id": 1,   "nombre": "Rex",   "raza": "Labrador" }, {   "especie": "Perro",   "id": 2,   "nombre": "Rex",   "raza": "Labrador" }, {   "especie": "Perro",   "id": 3,   "nombre": "Rex",   "raza": "Labrador" }, {   "especie": "Perro",   "id": 4,   "nombre": "Rex",   "raza": "Labrador" } }</pre>

The response body contains a JSON array of four pet objects, all of which are Labradors named Rex.

## ACTUALIZAR

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://localhost:8080/mascotas/rs/mascotas`
- Server response:**

Code	Details
200	<p><b>Response body</b></p> <pre>{   "especie": "Gato",   "id": 1,   "nombre": "Maik",   "raza": "Siames" }</pre> <p><b>Response headers</b></p> <pre>content-length: 57 content-type: application/json</pre>
- Responses:**

Code	Description
200	<p>Operación exitosa</p> <p>Media type: <input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value   Schema</p> <pre>{   "nombre": "Rex",   "especie": "Perro",   "raza": "Labrador" }</pre>
406	Error al actualizar la mascota

Vemos que se edito

Request URL: `http://localhost:8080/mascotas/rs/mascotas/list`

Server response

Code	Details
200	<p>Response body</p> <pre>[{"raza": "Labrador", "especie": "Perro", "id": 1, "nombre": "Rex", "raza": "Labrador"}, {"especie": "Perro", "id": 2, "nombre": "Rex", "raza": "Labrador"}, {"especie": "Perro", "id": 3, "nombre": "Rex", "raza": "Labrador"}, {"especie": "Gato", "id": 4, "nombre": "Maik", "raza": "Siames"}]</pre> <p>Response headers</p> <pre>content-length: 539 content-type: application/json</pre>

Responses

Code	Description	Links
------	-------------	-------

Eliminar

**DELETE** /mascotas Borrar una mascota

Borrar una mascota de la base de datos.

Parameters

Name	Description
<b>id</b> * required integer (query)	ID de la mascota a borrar

2

Execute Clear

Responses

Curl

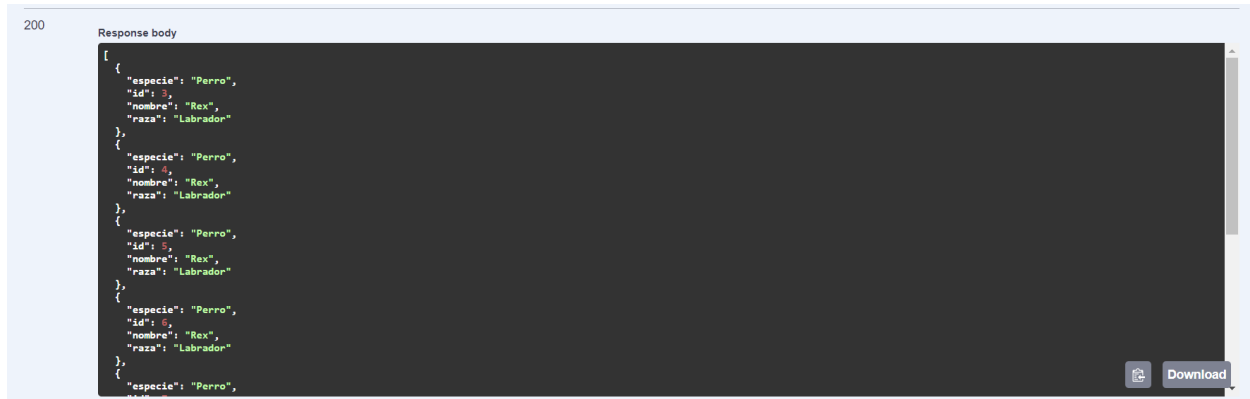
```
curl -X 'DELETE' \
'http://localhost:8080/mascotas/rs/mascotas?id=2' \
-H 'accept: application/json'
```

Request URL

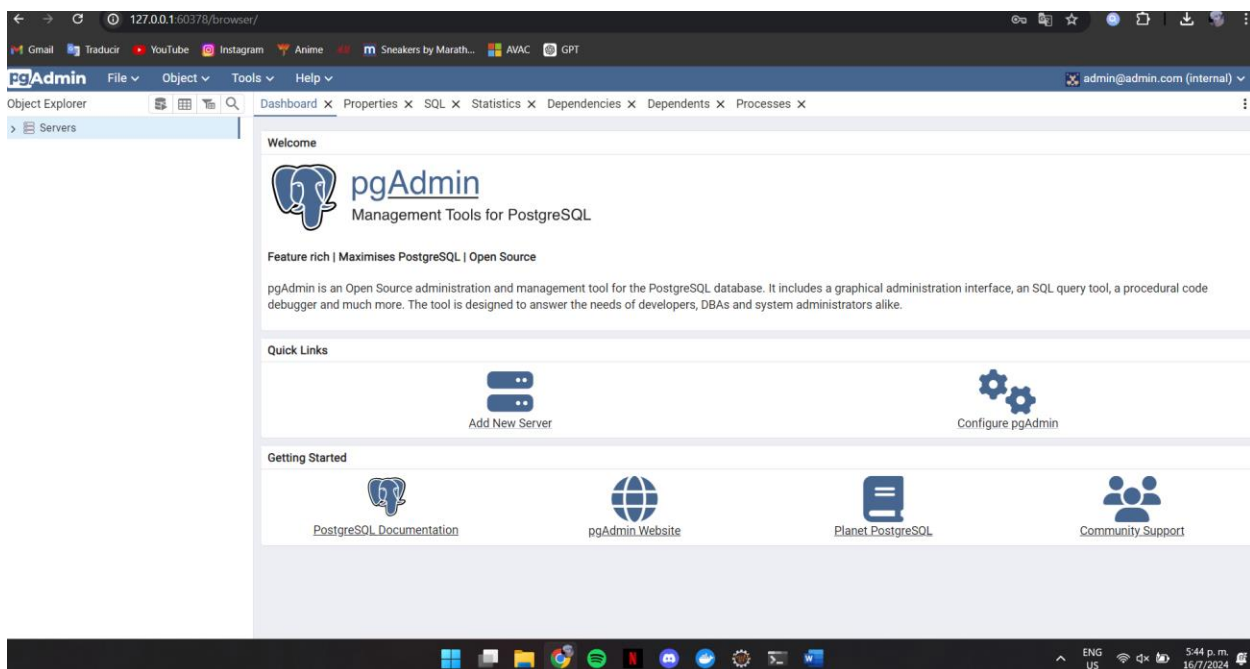
```
http://localhost:8080/mascotas/rs/mascotas?id=2
```

Server response

Al listar vemos que se elimina



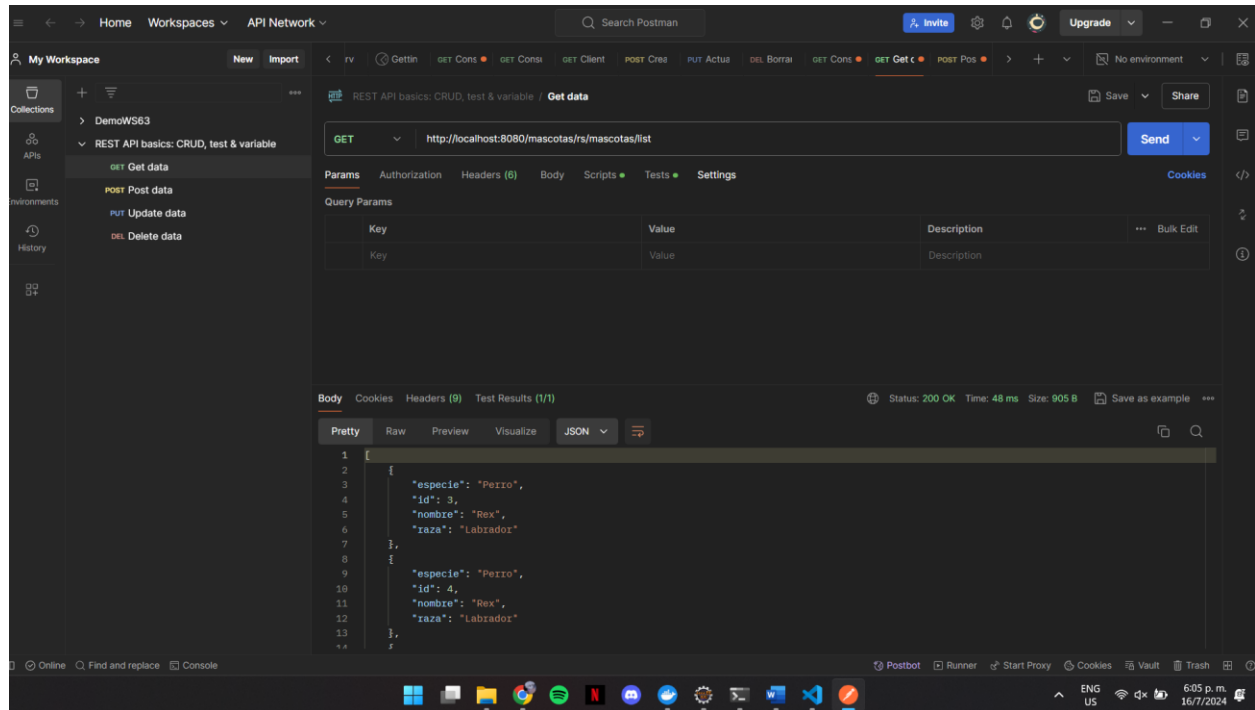
## Base de datos



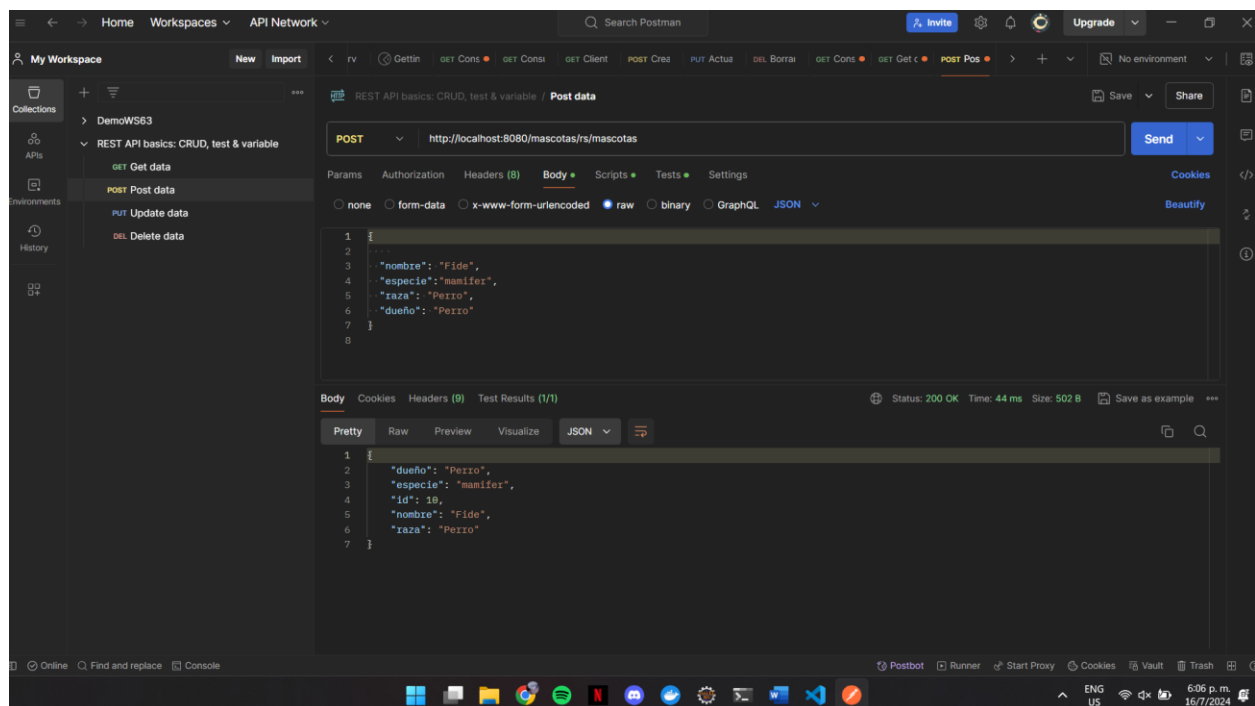


PotMan

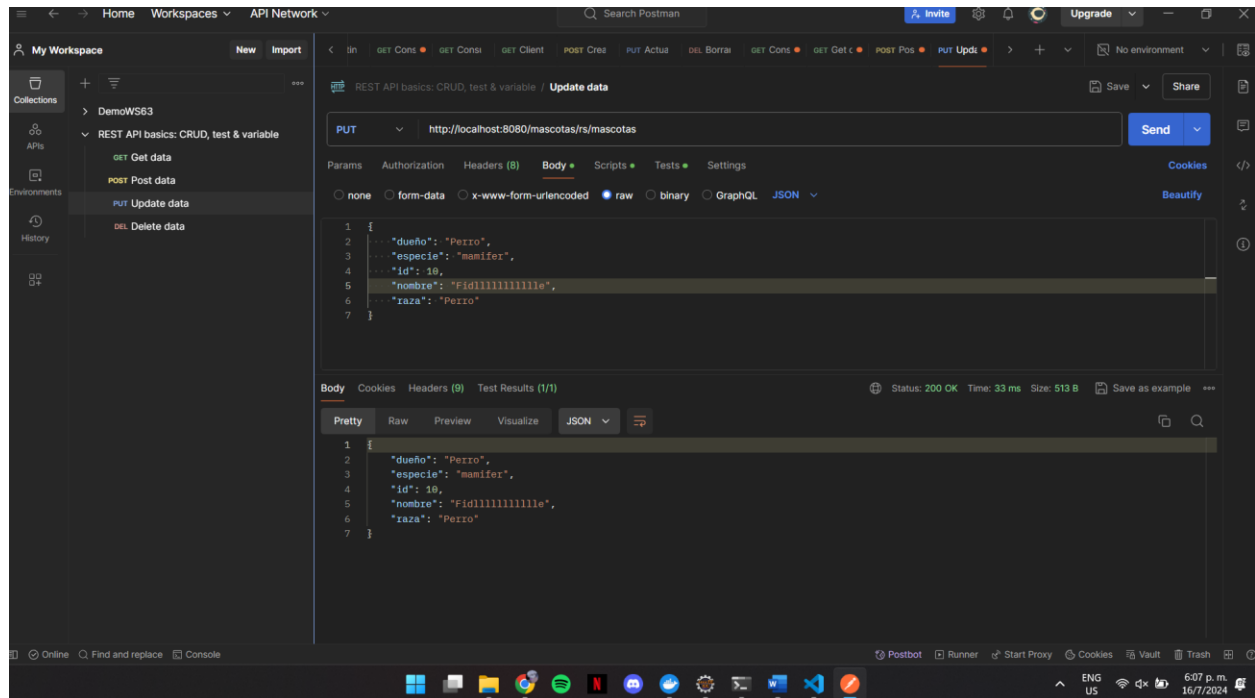
Listar



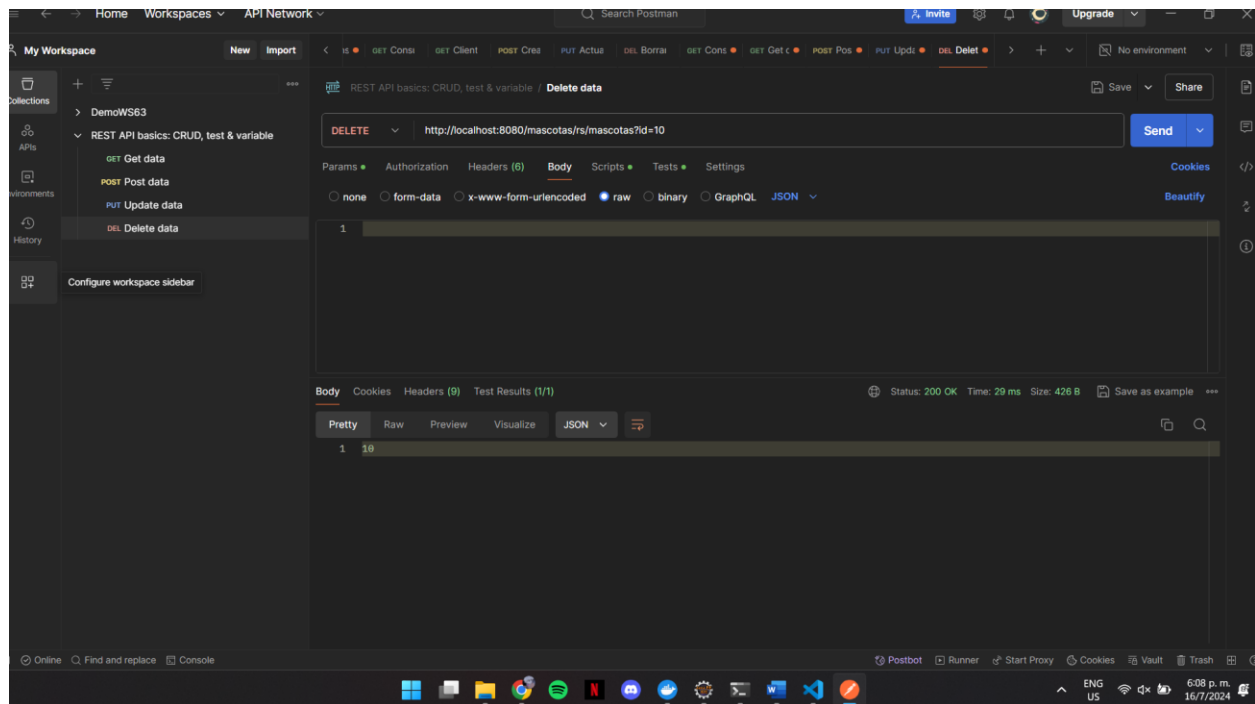
Agregar



## Actualizar

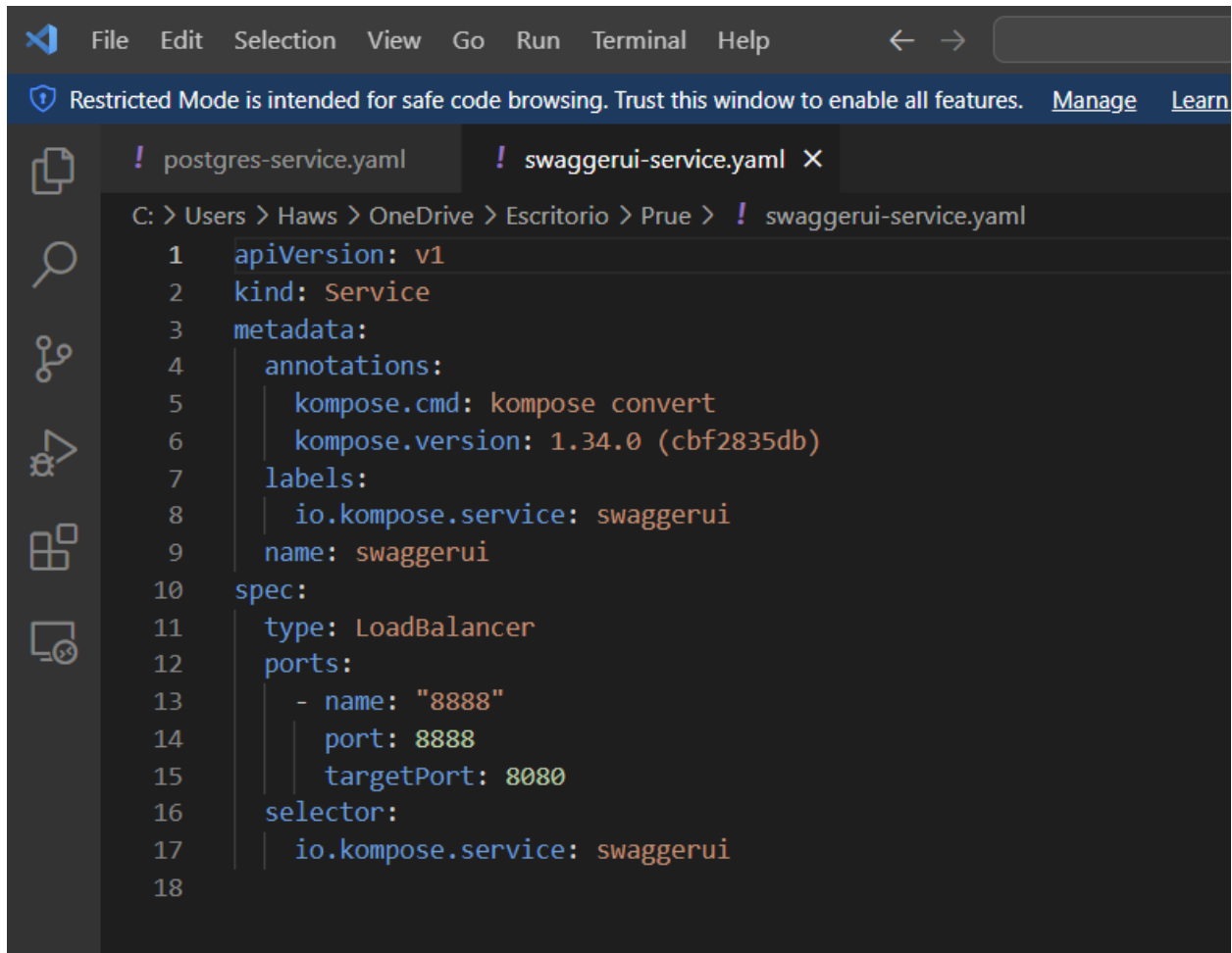


## Borrar



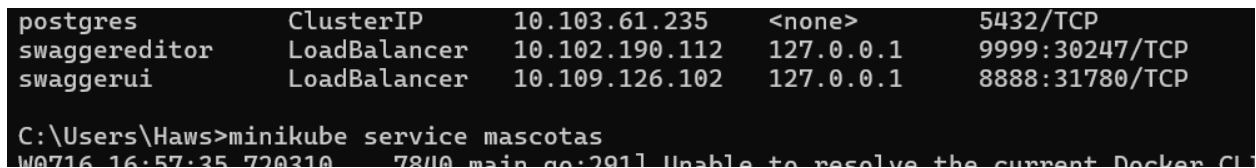
### Tarea 3: Despliegue en Kubernetes (40 minutos)

Cree un archivo YAML llamado `microservice-service.yaml` para un Service de tipo `LoadBalancer` que exponga el microservicio en el puerto 8080.



```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   annotations:
5     kompose.cmd: kompose convert
6     kompose.version: 1.34.0 (cbf2835db)
7   labels:
8     io.kompose.service: swaggerui
9   name: swaggerui
10 spec:
11   type: LoadBalancer
12   ports:
13     - name: "8888"
14       port: 8888
15       targetPort: 8080
16   selector:
17     io.kompose.service: swaggerui
18
```

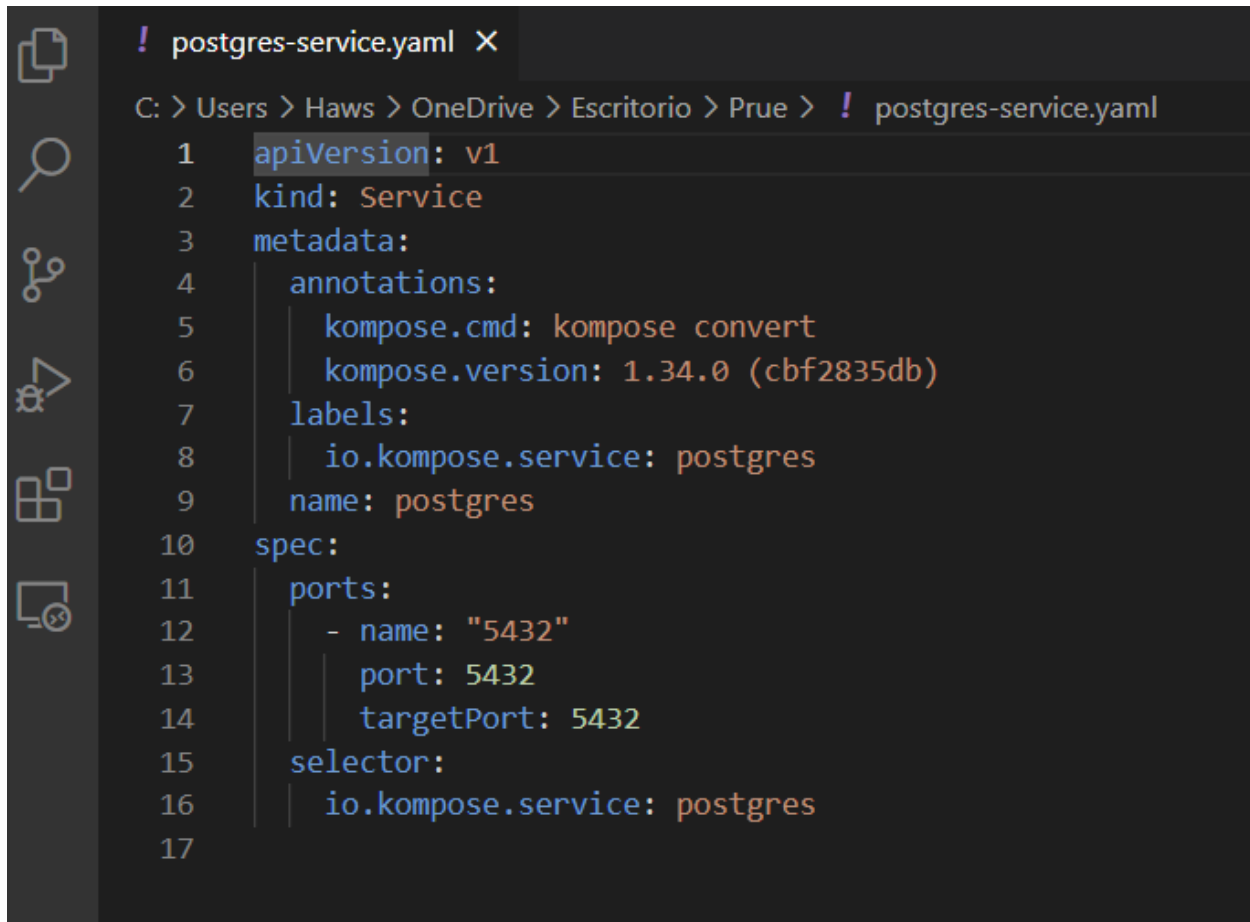
Puesto de tipo `LoadBalancer`



```
postgres      ClusterIP   10.103.61.235   <none>      5432/TCP
swaggereditor  LoadBalancer 10.102.190.112  127.0.0.1   9999:30247/TCP
swaggerui      LoadBalancer 10.109.126.102  127.0.0.1   8888:31780/TCP

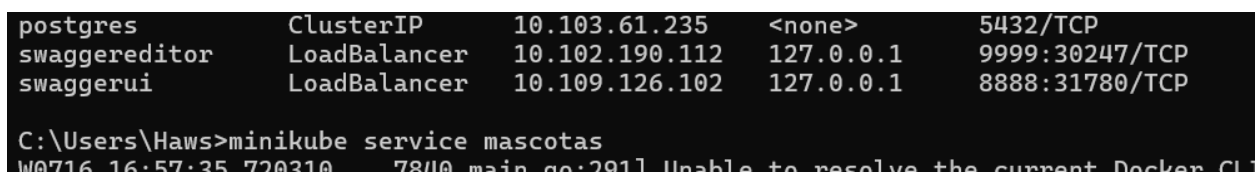
C:\Users\Haws>minikube service mascotas
W0716 16:57:35.720210 7840 main.go:2911 Unable to resolve the current Docker CL
```

Cree un archivo YAML llamado database-service.yaml para un Service de tipo ClusterIP que permita al microservicio comunicarse con la base de datos.



```
! postgres-service.yaml X
C: > Users > Haws > OneDrive > Escritorio > Prue > ! postgres-service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    annotations:
5      kompose.cmd: kompose convert
6      kompose.version: 1.34.0 (cbf2835db)
7    labels:
8      io.kompose.service: postgres
9    name: postgres
10 spec:
11   ports:
12     - name: "5432"
13       port: 5432
14       targetPort: 5432
15   selector:
16     io.kompose.service: postgres
17
```

Puesto Como cluster



postgres	ClusterIP	10.103.61.235	<none>	5432/TCP
swaggereditor	LoadBalancer	10.102.190.112	127.0.0.1	9999:30247/TCP
swaggerui	LoadBalancer	10.109.126.102	127.0.0.1	8888:31780/TCP

C:\Users\Haws>minikube service mascotas  
W0716 16:57:35.720310 [7840 main go:281] Unable to resolve the current Docker CL

Escalado de las replicas

## Escalar un recurso

deployment swaggerui will be updated to reflect the desired replicas count.

3

### Réplicas actuales

**i** Esta acción es equivalente a: `kubectl scale -n default deployment swaggerui --replicas=3`

Pods	Fecha de creación	↑
1 / 1	19.days.ago	⋮
1 / 1	19.days.ago	⋮
1 / 1	19.days.ago	⋮
1 / 1	19.days.ago	⋮
3 / 3	19.days.ago	⋮
1 / 1	19.days.ago	⋮
1 / 1	19.days.ago	⋮

● swaggerui swaggerapi/swagger-ui io.kompose.service: swaggerui 3 / 3 19 days ago ⋮