

Crear un servicio en donde se permita el listado de todas las mascotas a través del método GET, regresar el código 200.

The image shows a Postman interface with a GET request to `http://192.168.1.106:5000/mascotas` successfully executed. The response is a JSON array with two pet objects. Below the Postman window, a terminal window shows the command `python -u "c:\Users\Usuario\OneDrive\Documentos\Programacion de Redes\jmax\max.py"` being run, with output indicating the server is running on `http://127.0.0.1:5000` and `http://192.168.1.106:5000`.

Postman Request Details:

- Method: GET
- URL: `http://192.168.1.106:5000/mascotas`
- Status: 200 OK
- Time: 18 ms
- Size: 614 B

Response Body (JSON):

```
1 {
2   "M001": {
3     "alergias_conocidas": "Ninguna",
4     "edad_mascota": "2 años",
5     "nombre_mascota": "Rocky",
6     "notas_adicionales": "Le gusta jugar con pelotas",
7     "sexo_mascota": "Macho",
8     "tipo_raza": "Bulldog"
9   },
10  "M002": {
11    "alergias_conocidas": "Pollo",
```

Terminal Output:

```
[Running] python -u "c:\Users\Usuario\OneDrive\Documentos\Programacion de Redes\jmax\max.py"
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.1.106:5000
Press CTRL+C to quit
* Restarting with stat
```

Crear un servicio en donde se permita la búsqueda de una mascota mediante el nombre o la raza mediante el método GET, regresar el código 200

The screenshot shows the Postman interface with a new HTTP request configured. The URL is `http://192.168.1.106:5000/mascotas/buscar?raza=Persa`. The method is GET. The query parameter `raza` is set to `Persa`. The response is a JSON object with the following data:

```
{  "alergias_conocidas": "Pollo",  "edad_mascota": "4 años",  "nombre_mascota": "Nina",  "notas_adicionales": "Muy tranquila",  "sexo_mascota": "Hembra",  "tipo_raza": "Persa"}
```

The status of the response is 200 OK, with a time of 5 ms and a size of 383 B. The bottom panel shows the terminal output of the request:

```
192.168.1.106 - - [20/Nov/2024 18:45:58] "GET /mascotas HTTP/1.1" 200 -
192.168.1.106 - - [20/Nov/2024 18:46:09] "GET /mascotas HTTP/1.1" 200 -
192.168.1.106 - - [20/Nov/2024 18:46:38] "GET /mascotas/buscar?raza=Persa HTTP/1.1" 200 -
```

Agregar una mascota al diccionario y validar que los datos sean válidos o no sean vacíos,

método POST, regresar el código 201. Valor 0.5 puntos

. Verificar que no exista la mascota con el mismo nombre.

. Hacer manejos de errores para los tipos de datos

The screenshot shows the Postman interface with a POST request to `192.168.1.106:5000/mascotas`. The request body is a JSON object with the following fields:

```
1 {
2   "id_mascota": "M004",
3   "nombre_mascota": "Max",
4   "edad_mascota": "2 años",
5   "tipo_raza": "Labrador",
6   "alergias_conocidas": "Ninguna",
7   "sexo_mascota": "Macho",
8   "notas_adicionales": "Muy amigable"
9 }
10
```

The response status is **201 CREATED** with a response time of 13 ms and a body size of 245 B. The response body is a JSON object:

```
1 {
2   "estado": "exitoso",
3   "mensaje": "Mascota registrada correctamente"
4 }
```

Home

Workspaces

API Network

Search Postman

Invite

Upgrade

Collections

Environments

History

GET 192.168.1.106:5000/mas

POST 192.168.1.106:5000/ma

POST 192.168.1.106:5000/ma

+

No environment

HTTP

192.168.1.106:5000/mascotas

Save

Share

POST

192.168.1.106:5000/mascotas

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Beautiful

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "id_mascota": "M001",
3   "alergias_conocidas": "Ninguna",
4   "edad_mascota": "2 años",
5   "nombre_mascota": "Rocky",
6   "notas_adicionales": "Le gusta jugar con pelotas",
7   "sexo_mascota": "Macho",
8   "tipo_raza": "Bulldog"
9 }
```

Body

Cookies

Headers (5)

Test Results

400 BAD REQUEST

5 ms

256 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "estado": "error",
3   "mensaje": "Ya existe una mascota con ese ID o nombre"
4 }
```

Online

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

Modificar mascota del diccionario y validar que los datos sean correctos o no sean vacíos, método PUT, regresar el código 201.

The screenshot displays the Postman interface for a PUT request. The URL bar shows the endpoint `192.168.1.106:5000/mascotas/M002`. The request method is set to **PUT**. The request body is a JSON object with the following fields:

```
1 {
2   "nombre_mascota": "Nina",
3   "edad_mascota": "5 años",
4   "tipo_raza": "Persa",
5   "alergias_conocidas": "Pollo",
6   "sexo_mascota": "Hembra",
7   "notas_adicionales": "Le encanta dormir"
8 }
```

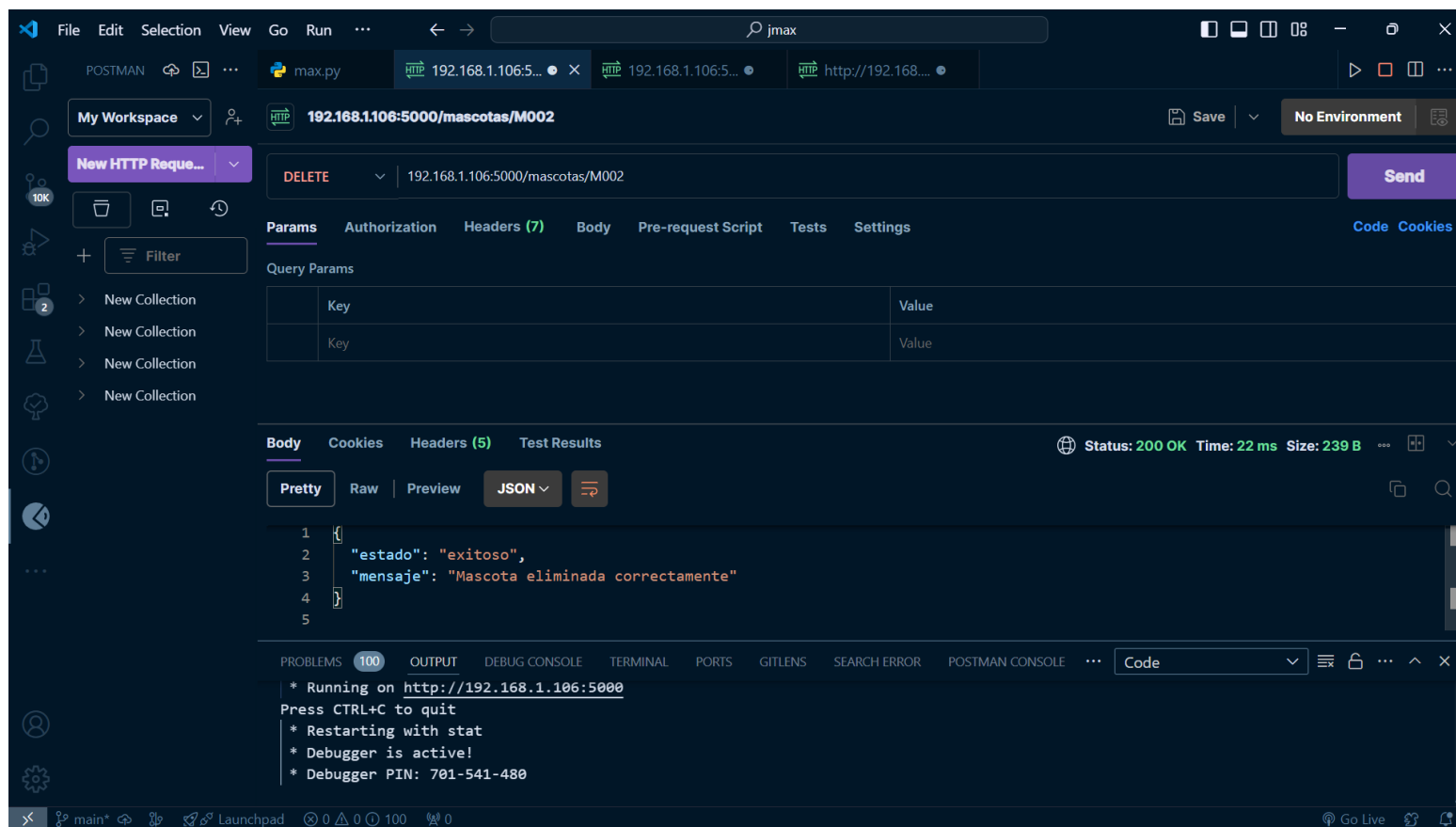
The response status is **201 OK**, with a response time of 10 ms and a size of 240 B. The response body is a JSON object:

```
1 {
2   "estado": "exitoso",
3   "mensaje": "Datos de la mascota actualizados"
4 }
```

The interface includes a sidebar with navigation options (Home, Workspaces, API Network), a top bar with search and user settings, and a bottom status bar with various tool icons.

Eliminar una mascota mediante el método DELETE

i. Verificar que exista la mascota para su eliminación o en su caso enviar un mensaje de error personalizado.



POSTMAN

File Edit Selection View Go Run ...

max.py 192.168.1.106:5... 192.168.1.106:5... 192.168.1.106:5...

My Workspace

New HTTP Reque...

10K

Filter

New Collection

New Collection

New Collection

New Collection

192.168.1.106:5000/mascotas/M002

DELETE 192.168.1.106:5000/mascotas/M002

Params

Query Params

Key Value

Key Value

Body

Status: 200 OK Time: 22 ms Size

Pretty Raw Preview JSON

```
1 {
2   "estado": "exitoso",
3   "mensaje": "Mascota eliminada correctamente"
4 }
```

192.168.1.106:5000/mascotas/M002

DELETE 192.168.1.106:5000/mascotas/M002

Params

Query Params

Key Value

Key Value

Body

Status: 404 Not Found Time: 12 ms Size

Pretty Raw Preview JSON

```
1 {
2   "estado": "error",
3   "mensaje": "Mascota no encontrada"
4 }
```

PROBLEMS 100 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH ERROR POSTMAN CONSOLE ... Code

```
* Running on http://192.168.1.106:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 701-541-480
```