**Introducción**

A continuación se resumen los resultados de cuatro pruebas principales realizadas en el servicio PetStore ubicadas en dos archivos.feature:

**1. Añadir una mascota (POST).**

* Endpoint: https://petstore.swagger.io/v2/pet
* Método: POST
* Resultado: La mascota fue creada exitosamente (código 200), verifica que el response retorne name y status.

**2. Consultar mascota por ID (GET).**

* Endpoint: [https://petstore.swagger.io/v2/pet/<id](https://petstore.swagger.io/v2/pet/%3cid)> donde <id> corresponde al id anterior de la mascata creada.
* Método: GET
* Resultado: La respuesta coincidió con los datos enviados en el POST inicial se realiza el assert con el status.

**3. Actualizar mascota (PUT).**

* Endpoint: https://petstore.swagger.io/v2/pet
* Método: PUT
* Resultado: La respuesta confirmó la actualización (código 200).

**4. Consultar mascota por estado (GET).**

* Endpoint: https://petstore.swagger.io/v2/pet/findByStatus?status=sold
* Método: GET
* Resultado:Retorna todas las mascotas con el estatus sold.

**Conclusiones:**

1.- Todas las pruebas se ejecutaron correctamente, validando el flujo(post, get, put).

2.- Karate Framework demostró ser eficiente para automatizar pruebas de API, gracias a su sintaxis simple basada en Gherkin, teniendo un mejor desempeño que automatizar apis en monitors de postman o pytest de python, por la facilidad del lenguaje Gherkin.

3.- El servicio PetStore respondió como se esperaba, confirmando que los endpoints funcionan correctamente, esto agregando los match correspondientes en base a los reponse y a la documentacion del Swagger.