EJERCICIO 2

La página https://petstore.swagger.io/ proporciona la documentación sobre apis de una “PetStore”. Utilizando un software para pruebas de servicios REST realizar las siguientes pruebas, identificando las entradas, capturando las salidas, test, variables, etc, en cada uno de los siguientes casos:

- Crear un usuario
- Buscar el usuario creado
- Actualizar el nombre y el correo del usuario
- Buscar el usuario actualizado
- Eliminar el usuario

Desarrollo

- Configuración a nivel de cabecera

```
Feature: PetStore
  Background: Cabecera
    #Definimos url base
    Given url 'https://petstore.swagger.io/v2'

    #Siguiendo la pagina del "https://petstore.swagger.io/v2" identificamos el request
    #analizamos el request, que parametros definir y agregar para crear un usuario
    #una vez definida y replicada en la pagina consumimos el metodo (post) con status 200.
```

- Crear un usuario

```
@test
Scenario: Crear un usuario
  #Identificamos el path
  Given path '/user'
  And request
  """
    {
      "id": 500,
      "username": "Luis",
      "firstName": "Suarez",
      "lastName": "Rojas",
      "email": "luis.rojas@email.com",
      "password": "erme&12LINDA",
      "phone": "987654321",
      "userStatus": 101
    }
  """
  When method post
  Then status 200

  #Siguiendo la pagina del "https://petstore.swagger.io/v2" identificamos metodo a consumir
  #(GET) para la busqueda del usuario con username, referimos el username del metodo anterior para
  # realizar la consulta correspondiente y visualizar que el usuario fue agrega correctamente.
```

- Buscar el usuario creado

```
@test
Scenario:  Buscar el usuario creado
  Given path 'user/Luis'
  When method get
  Then status 200

#Siguiendo la pagina del "https://petstore.swagger.io/v2" identificamos metodo a consumir
  #(PUT) status 200 la cual nos sirve para actulizar data.
  #En este caso nos pide actualizar el username y email del usuario.
```

- Actualizar el nombre y el correo del usuario

```
@test
Scenario: Actualizar el nombre y el correo del usuario
  #Identificamos el path
  Given path '/user/Luis'
  And request
  """
  {
    "id": 500,
    "username": "Mario",
    "firstName": "Suarez",
    "lastName": "Rojas",
    "email": "mario.rojas@email.com",
    "password": "erme&12LINDA",
    "phone": "987654321",
    "userStatus": 201
  }
  """

  When method put
  Then status 200

  #Siguiendo la pagina del "https://petstore.swagger.io/v2" identificamos metodo a consumir
  #(GET) para la busqueda del usuario con username, referimos el username del metodo anterior para
  # realizar la consulta correspondiente y visualizar que el usuario fue actualizado correctamente.
```

- Buscar el usuario actualizado

```
@test
Scenario:  Buscar el usuario actualizado
  Given path 'user/Mario'
  When method get
  Then status 200

  #Siguiendo la pagina del "https://petstore.swagger.io/v2" identificamos metodo a consumir
  #(DELETE) para eliminar un usuario, referimos en el path el username a eliminar.
  # realizar el delete correspondiente y visualizar que la mascota fue eliminada correctamente.
```

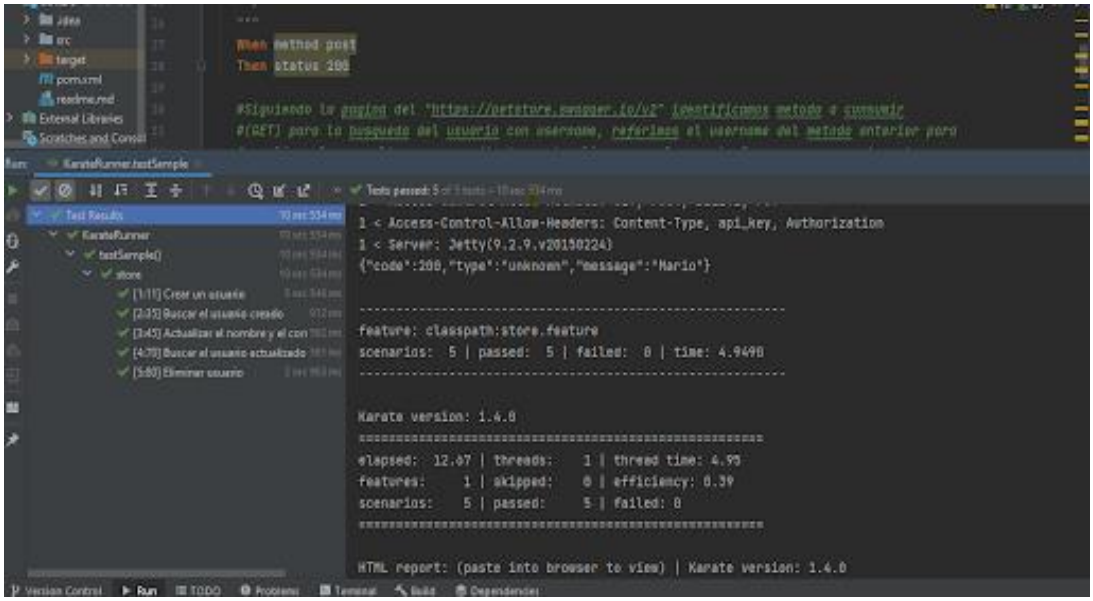
- Eliminar el usuario

```
@test
Scenario:  Eliminar usuario
  Given path 'user/Mario'
  When method delete
  Then status 200
```

- Se creo una clase KarateRunner para ejecutar el feature y todos los escenarios implementados.

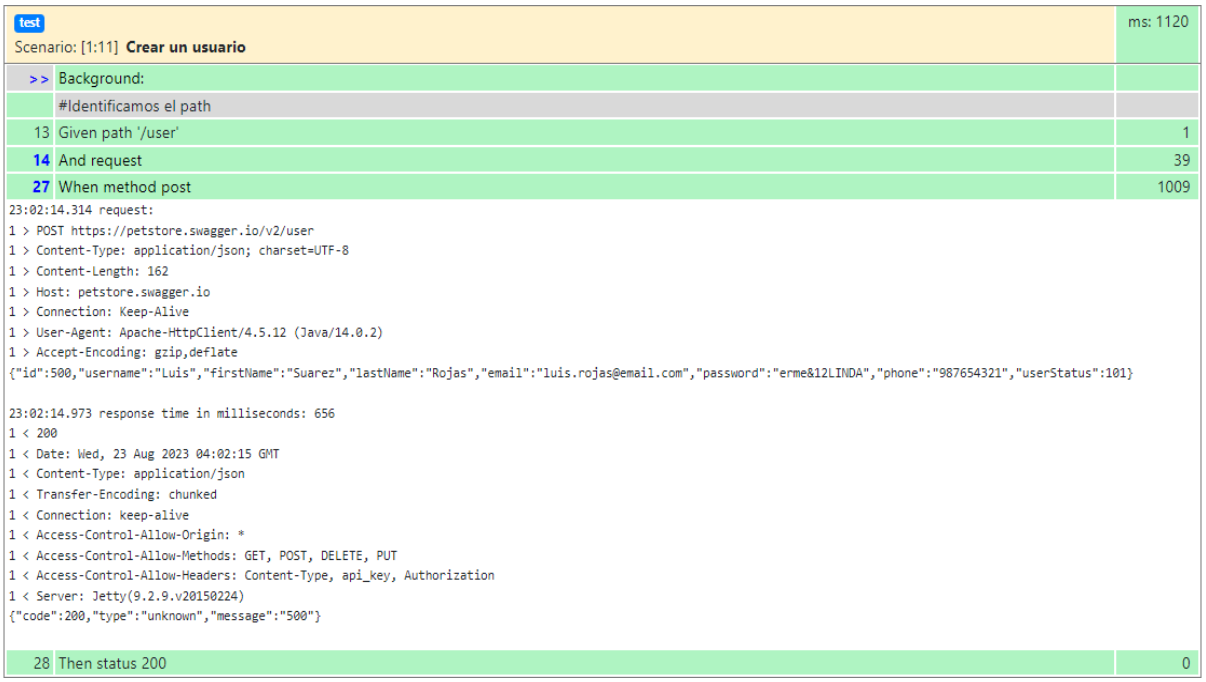
```
import com.intuit.karate.junit5.Karate;


public class KarateRunner {
    @Karate.Test
    Karate testSample() { return Karate.run("classpath:store.feature"); }
}
```

- Se muestra la ejecución de la clase java KarateRunner, con los casos mencionados con respuesta exitosa como se muestra en la imagen.



Reporte de los métodos ejecutados con Karate a nivel web (Chrome)

- Crear un usuario



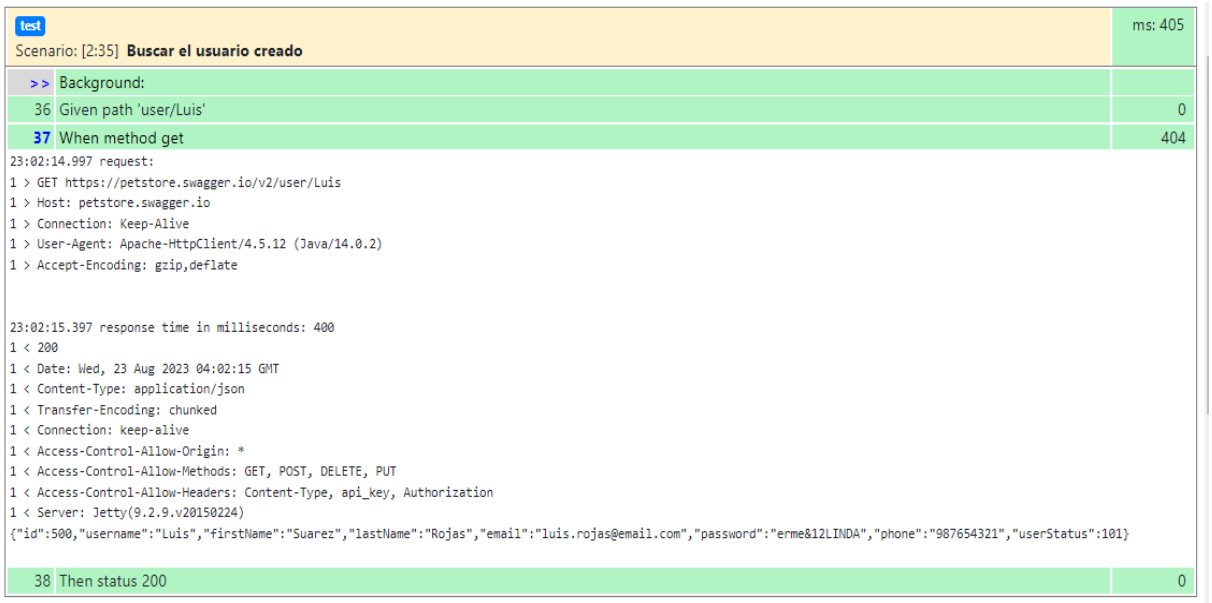
test
Scenario: [1:11] **Crear un usuario**
ms: 1120

>>	Background:	
	#Identificamos el path	
13	Given path '/user'	1
14	And request	39
27	When method post	1009

```
23:02:14.314 request:
1 > POST https://petstore.swagger.io/v2/user
1 > Content-Type: application/json; charset=UTF-8
1 > Content-Length: 162
1 > Host: petstore.swagger.io
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.12 (Java/14.0.2)
1 > Accept-Encoding: gzip,deflate
{"id":500,"username":"Luis","firstName":"Suarez","lastName":"Rojas","email":"luis.rojas@email.com","password":"erme&12LINDA","phone":"987654321","userStatus":101}

23:02:14.973 response time in milliseconds: 656
1 < 200
1 < Date: Wed, 23 Aug 2023 04:02:15 GMT
1 < Content-Type: application/json
1 < Transfer-Encoding: chunked
1 < Connection: keep-alive
1 < Access-Control-Allow-Origin: *
1 < Access-Control-Allow-Methods: GET, POST, DELETE, PUT
1 < Access-Control-Allow-Headers: Content-Type, api_key, Authorization
1 < Server: Jetty(9.2.9.v20150224)
{"code":200,"type":"unknown","message":"500"}
```

28	Then status 200	0

- Buscar el usuario creado



test
Scenario: [2:35] **Buscar el usuario creado**
ms: 405

>>	Background:	
36	Given path 'user/Luis'	0
37	When method get	404

```
23:02:14.997 request:
1 > GET https://petstore.swagger.io/v2/user/Luis
1 > Host: petstore.swagger.io
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.12 (Java/14.0.2)
1 > Accept-Encoding: gzip,deflate


23:02:15.397 response time in milliseconds: 400
1 < 200
1 < Date: Wed, 23 Aug 2023 04:02:15 GMT
1 < Content-Type: application/json
1 < Transfer-Encoding: chunked
1 < Connection: keep-alive
1 < Access-Control-Allow-Origin: *
1 < Access-Control-Allow-Methods: GET, POST, DELETE, PUT
1 < Access-Control-Allow-Headers: Content-Type, api_key, Authorization
1 < Server: Jetty(9.2.9.v20150224)
{"id":500,"username":"Luis","firstName":"Suarez","lastName":"Rojas","email":"luis.rojas@email.com","password":"erme&12LINDA","phone":"987654321","userStatus":101}
```

38	Then status 200	0

- Actualizar el nombre y el correo del usuario

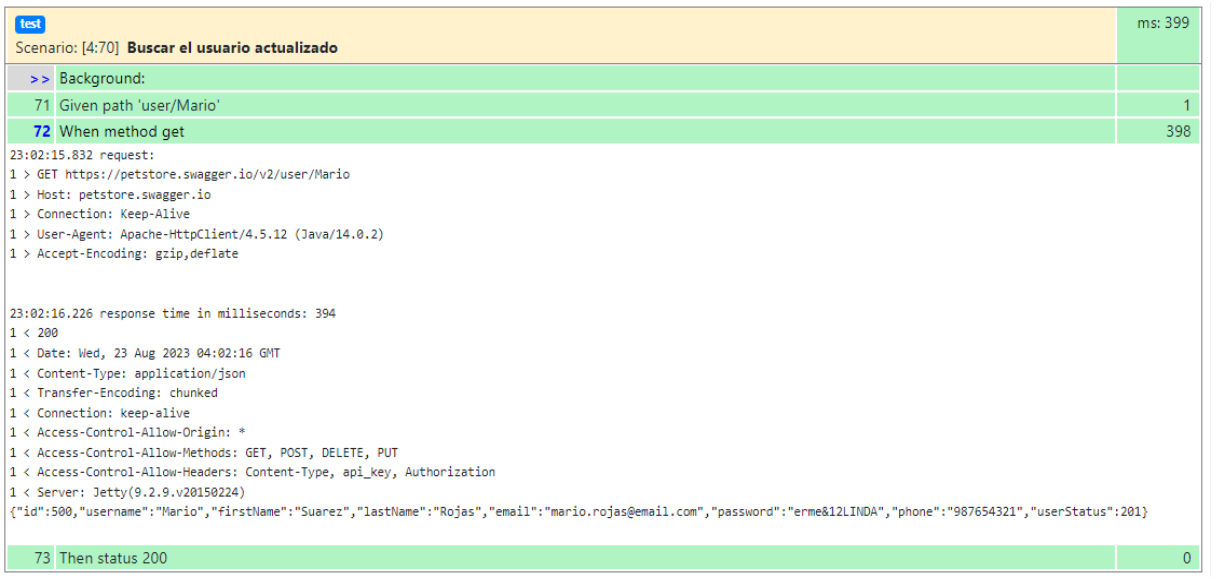
test
Scenario: [3:45] **Actualizar el nombre y el correo del usuario** ms: 411

>>	Background:	
	#Identificamos el path	
47	Given path '/user/Luis'	0
48	And request	0
62	When method put	410

```
23:02:15.412 request:
1 > PUT https://petstore.swagger.io/v2/user/Luis
1 > Content-Type: application/json; charset=UTF-8
1 > Content-Length: 164
1 > Host: petstore.swagger.io
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.12 (Java/14.0.2)
1 > Accept-Encoding: gzip,deflate
{"id":500,"username":"Mario","firstName":"Suarez","lastName":"Rojas","email":"mario.rojas@email.com","password":"erme&12LINDA","phone":"987654321","userStatus":201}

23:02:15.819 response time in milliseconds: 407
1 < 200
1 < Date: Wed, 23 Aug 2023 04:02:16 GMT
1 < Content-Type: application/json
1 < Transfer-Encoding: chunked
1 < Connection: keep-alive
1 < Access-Control-Allow-Origin: *
1 < Access-Control-Allow-Methods: GET, POST, DELETE, PUT
1 < Access-Control-Allow-Headers: Content-Type, api_key, Authorization
1 < Server: Jetty(9.2.9.v20150224)
{"code":200,"type":"unknown","message":"500"}
```

63	Then status 200	0

- Buscar el usuario actualizado



test
Scenario: [4:70] **Buscar el usuario actualizado** ms: 399

>>	Background:	
71	Given path 'user/Mario'	1
72	When method get	398

```
23:02:15.832 request:
1 > GET https://petstore.swagger.io/v2/user/Mario
1 > Host: petstore.swagger.io
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.12 (Java/14.0.2)
1 > Accept-Encoding: gzip,deflate


23:02:16.226 response time in milliseconds: 394
1 < 200
1 < Date: Wed, 23 Aug 2023 04:02:16 GMT
1 < Content-Type: application/json
1 < Transfer-Encoding: chunked
1 < Connection: keep-alive
1 < Access-Control-Allow-Origin: *
1 < Access-Control-Allow-Methods: GET, POST, DELETE, PUT
1 < Access-Control-Allow-Headers: Content-Type, api_key, Authorization
1 < Server: Jetty(9.2.9.v20150224)
{"id":500,"username":"Mario","firstName":"Suarez","lastName":"Rojas","email":"mario.rojas@email.com","password":"erme&12LINDA","phone":"987654321","userStatus":201}
```

73	Then status 200	0

- Eliminar el usuario

test

Scenario: [5:80] **Eliminar usuario** — ms: 434

>>	Background:	
81	Given path 'user/Mario'	0
82	When method delete	433

```
23:02:16.240 request:
1 > DELETE https://petstore.swagger.io/v2/user/Mario
1 > Host: petstore.swagger.io
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.12 (Java/14.0.2)
1 > Accept-Encoding: gzip,deflate


23:02:16.670 response time in milliseconds: 429
1 < 200
1 < Date: Wed, 23 Aug 2023 04:02:16 GMT
1 < Content-Type: application/json
1 < Transfer-Encoding: chunked
1 < Connection: keep-alive
1 < Access-Control-Allow-Origin: *
1 < Access-Control-Allow-Methods: GET, POST, DELETE, PUT
1 < Access-Control-Allow-Headers: Content-Type, api_key, Authorization
1 < Server: Jetty(9.2.9.v20150224)
{"code":200,"type":"unknown","message":"Mario"}
```

83	Then status 200	0

Imagen con todos los métodos ejecutados correctamente.