

Reporte

Caso 1.

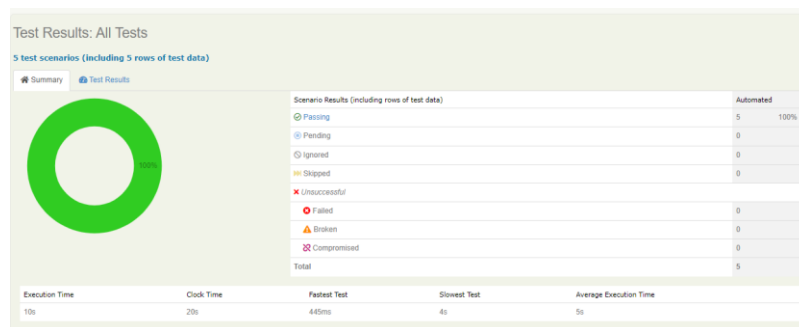
Consultar recurso por id

```
Feature:consultar recurso por id
yo como administrador del servicio REQRES
quiero realizar una peticion de consulta de usuario por id en REQRES
para ver los recursos registrados

@ConsultarUsuario
Scenario Outline:Consulta exitoso
Given que el usuario esta en la pagina de consulta de recursos
When el usuario envia una peticion get con el <id> del recurso
Then debe retornar un codigo de respuesta <codigoEstado> y name <name>
Examples:
| id | codigoEstado | name |
| "2" | 200 | "fuchsia rose" |
| "6" | 400 | "blue turquoise" |
```

Prueba realizada en Serenity

The screenshot shows the Serenity test runner interface. On the left, a tree view shows the test structure: 'ConsultarRecursoTestRunner' (5 sec 289 ms) containing 'Consulta exitoso' (4 sec 521 ms) and 'Consulta exitoso' (463 ms). The main console displays the test execution log, including the scenario 'Consulta exitoso' and its steps. The log shows the test passed with a status code of 200 and the name 'fuchsia rose'. A red box highlights the 'status code: 200' and 'nombre: fuchsia rose' in the log output. Red arrows point to these values with labels 'status' and 'name'.



The figure shows the 'Feature: Consultar Recurso Por Id' details. It includes the feature description, the scenario 'Consulta exitoso' (2 examples), and the scenario details. The scenario details show the given, when, and then clauses, and the examples table.

id	codigoEstado	name
"2"	200	"fuchsia rose"
"6"	400	"blue turquoise"

Steps

Example: {id="2", codigoEstado=200, name="fuchsia rose"}

Given que el usuario esta en la pagina de consulta de recursos

When el usuario envia una petition get con el "2" del recurso

Nevardo do get

Nevardo executes a GET on the resource api/unknown/2

GET https://reqres.in/api/unknown/2

REST Query

Response

Status code: 200

Content Type: null

Request Headers

Accept: */*

Content Body

Request Cookies

Response Headers

Date: Wed, 29 Mar 2023 19:38:05 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"0b-omada/0Y1DQhpgasVY0Rvcs"
Via: 1.1 vegur
Cache-Control: max-age=14400
CF-Cache-Status: REVALIDATED
Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?r=6Gub4u10155ASj8E1xGpL12B4Wu52F8Hf634042P4uK2uXuo0yq3Gy12jFq9X04F68u182P6L1Z2FYK000P4u8817A3L3KvAq2M6C1h4d4F32vG10UE3C"}],"group":"cf-nel","max_age":604800}
NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Vary: Accept-Encoding
Server: cloudflare
CF-RAY: 7af93729a22805-MIA
Content-Encoding: gzip

Then debe retornar un codigo de respuesta 200 y name "fuchsia rose"

***** El codigo de respuesta de la petition seria --> 200

***** El codigo de respuesta de la petition seria --> fuchsia rose

Example: {id="6", codigoEstado=400, name="blue turquoise"}

Given que el usuario esta en la pagina de consulta de recursos

When el usuario envia una petition get con el "6" del recurso

Nevardo do get

Nevardo executes a GET on the resource api/unknown/6

GET https://reqres.in/api/unknown/6

Response Body

```
{
  "data": {
    "id": 2,
    "name": "fuchsia rose",
    "year": 2001,
    "color": "#C74375",
    "pantone_value": "17-2031"
  },
  "support": {
    "url": "https://reqres.in/#support-heading",
    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

GET https://reqres.in/api/unknown/6

REST Query

Response

Status code: 200

Content Type: null

Request Headers

Accept: */*

Content Body

Request Cookies

Response Headers

Date: Wed, 29 Mar 2023 19:38:56 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"ea-Aq/Addr7g9jhZew0WuZYCuiGhc4"
Via: 1.1 vegur
Cache-Control: max-age=14400
CF-Cache-Status: REVALIDATED
Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=KoeTyq15Vau4eda3ZfUgzeH6WghxfJrxYc8h32BcnL8p6fC59HAspdgeStogh0W5Kw25Z2Ff3f3Z2fA3Zfey1JmUPJRN"}],"group":"cf-nel","max_age":604800}
NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Vary: Accept-Encoding
Server: cloudflare
CF-RAY: 7af93784e3302ba-MIA
Content-Encoding: gzip



Evidencia pruebas realizadas en postman, petición Get de recurso del servicio REQRES

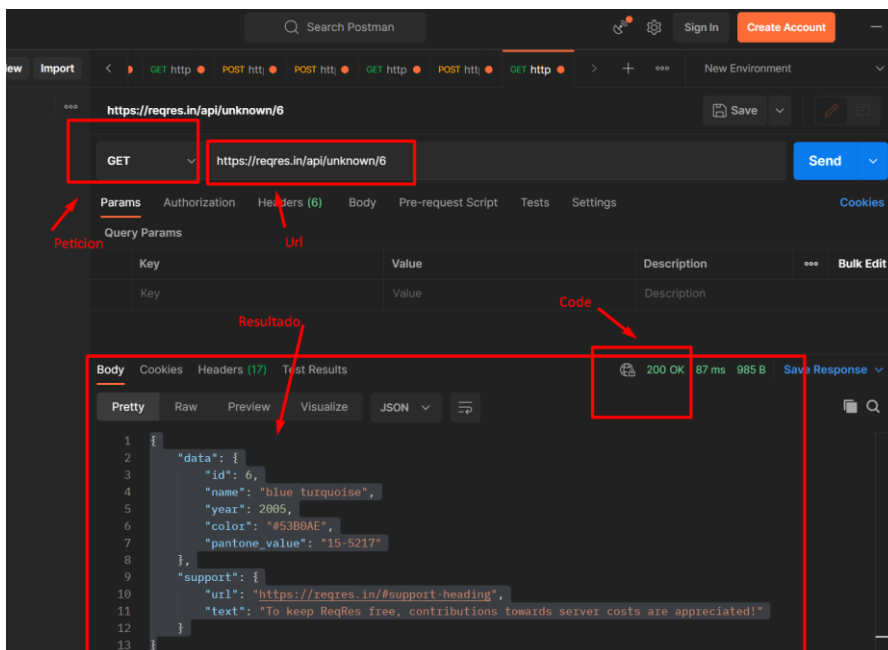
con la siguiente ruta

base

<https://reqres.in/>

recurso

api/unknown/6



Caso 2.

Registrar un usuario

```
Feature: Creacion de usuario
yo como administrador del servicio REQRES
quiero realizar una peticion de crear un usuario
para registrar un nuevo usuario

@RegisterUsuario
Scenario Outline: Registro exitoso
    Given que el usuario esta en la pagina de registro de la api
    When el usuario envia una peticion post con el name <name> y job <job>
    Then el usuario debe ver un codeStatus <code> de respuesta y el id <id>

Examples:
| name      | job  | code | id |
| "Antonio" | "QA" | 201  | 2  |
| "Maria"   | "QA" | 201  | 24 |
| "Angelica"| "QA" | 201  | 22 |
```

Prueba realizada en Serenity

Tests passed: 3 of 3 tests - 5 sec 161 ms

RegistrarUsuarioTestRunner (com.softkau.r) 5 sec 161 ms

- ✓ Creacion de usuario 5 sec 161 ms
- ✓ Registro exitoso 4 sec 257 ms
- ✓ Registro exitoso 455 ms
- ✓ Registro exitoso 449 ms

Body:

```
{
  "name": "Maria",
  "job": "QA"
}
```

Nuevo registro

HTTP/1.1 201 Created
Date: Wed, 29 Mar 2023 19:53:38 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 77
Connection: keep-alive

Feature: Creacion De Usuario

yo como administrador del servicio REQRES
quiero realizar una peticion de crear un usuario
para registrar un nuevo usuario

Specifications Test Results

Scenarios

Registro exitoso (3 examples)

Scenario details

Registro exitoso

Given que el usuario esta en la pagina de registro de la api
When el usuario envia una peticion post con el name <name> y job <job>
Then el usuario debe ver un codeStatus <code> de respuesta y el id <id>

Examples:

name	job	code	id
"Antonio"	"QA"	201	2
"Maria"	"QA"	201	24
"Angelica"	"QA"	201	22

✓ Registro exitoso

Call

Scenario Outline

Given que el usuario esta en la pagina de registro de la api
When el usuario envia una peticion post con el name <name> y job <job>
Then el usuario debe ver un codeStatus <code> de respuesta y el id <id>

Examples:

#	Name	Job	Code	Id
1	"Antonio"	"QA"	201	2
2	"Maria"	"QA"	201	24
3	"Angelica"	"QA"	201	22

Steps

- Example: {name="Antonio", job="QA", code=201, id=2}
- Given que el usuario esta en la pagina de registro de la api
- When el usuario envia una peticion post con el name "Antonio" y job "QA"
- Then el usuario debe ver un codeStatus 201 de respuesta y el id 2

POST https://reqres.in/api/users

REST Query

Response

Status code: 201

Content Type: application/json

Request Headers

Accept: /*
Content-Type: application/json; charset=utf-8

Content Body

```
{
  "name": "Antonio",
  "job": "QA"
}
```

Request Cookies

Response Headers

Date: Wed, 29 Mar 2023 10:53:37 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 79
Connection: keep-alive
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"4f-tj7X9GouDZEk9p8Kxjd1lAgOU"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Report-To: {
 "endpoints": [{
 "url": "https://a.ne1.cloudflare.com/report/v3?ts=q9t5gzycrotaeq4JL39F071h2XGqNcQj56o0s5Jbal-ruxbgnaz6T1saw9JtsqDk18AhBo0fbxqE35hLb3Jh2Tr32B0oeh5kcp9PjJkwt
 NEL: {
 "success_fraction": 0,
 "report_to": "cf-nel",
 "max_age": 604800
 }
 }
}
Server: cloudflare
CF-RAY: 7afa8fa990e67b7-MIA

Response Body

```
{
  "name": "Antonio",
  "job": "QA",
  "id": "7777",
  "createdAt": "2023-03-29T10:53:37.000Z"
}
```

Then el usuario debe ver un codeStatus 201 de respuesta y el id 2

El codigo de respuesta es: 201

Then Retorna información should be not null

Example: {name="Maria", job="QA", code=201, id=24}

Given que el usuario esta en la pagina de registro de la api

When el usuario envia una peticion post con el name "Maria" y job "QA"

Then el usuario debe ver un codeStatus 201 de respuesta y el id 2

POST https://reqres.in/api/users

REST Query

Response

Status code: 201

Content Type: application/json

Request Headers

Accept: /*
Content-Type: application/json; charset=utf-8

Content Body

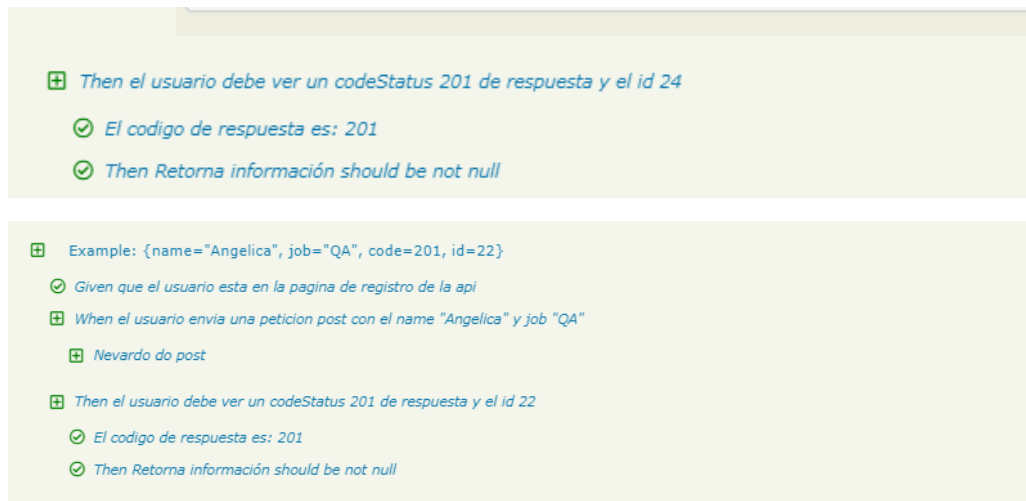
```
{
  "name": "Maria",
  "job": "QA"
}
```

Response Headers

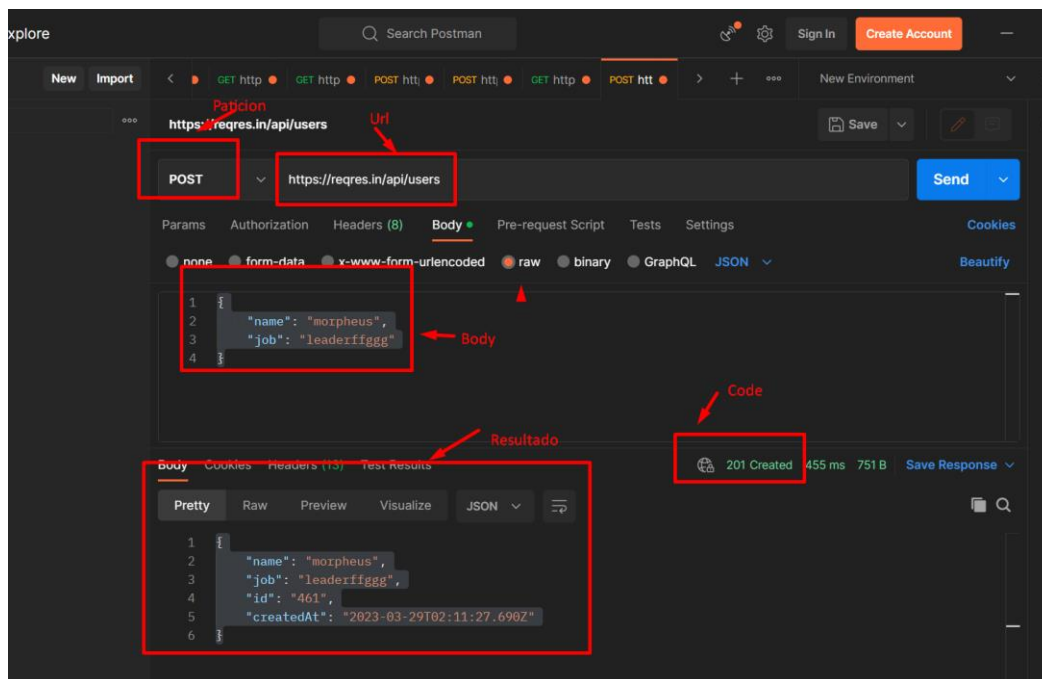
Date: Wed, 29 Mar 2023 10:53:38 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 77
Connection: keep-alive
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"4d-q24UJD3H0GekPvOLk2zaA0fHMo"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Report-To: {
 "endpoints": [{
 "url": "https://a.ne1.cloudflare.com/report/v3?ts=3C1ph4QVE32n1P4UBN5-4wZaCkBBHFKECF3Ikw3KXvc1EF9krmxH13UuQx8Z28w0XG81f7gFknzK3h1t3cJbdfC3vANS328W13rYTS
 NEL: {
 "success_fraction": 0,
 "report_to": "cf-nel",
 "max_age": 604800
 }
 }
}
Server: cloudflare
CF-RAY: 7afa8fffaacb3c8-MIA

Response Body

```
{
  "name": "Maria",
  "job": "QA",
  "id": "578",
  "createdAt": "2023-03-29T10:53:38.739Z"
}
```



Pruebas realizadas en Postman, petición Post de recurso del servicio reqre
con la siguiente ruta
base: <https://regres.in/>
recurso: api/users



Caso 1 por Karate.

Crear nuevos registros

Caso realizada en karate

```
Feature: Post usuario en regres

Background:
  * url "https://regres.in"
  * path "/api/users/"
  * request { "name": "#(name)", "job": "#(job)" }

Scenario Outline: Post un usuario
  When method post
  Then status 201
  Examples:
    | name      | job      |
    | Nevardo | QA       |
    | Antonio   | QA       |
    | Luis      | Doctor   |
```

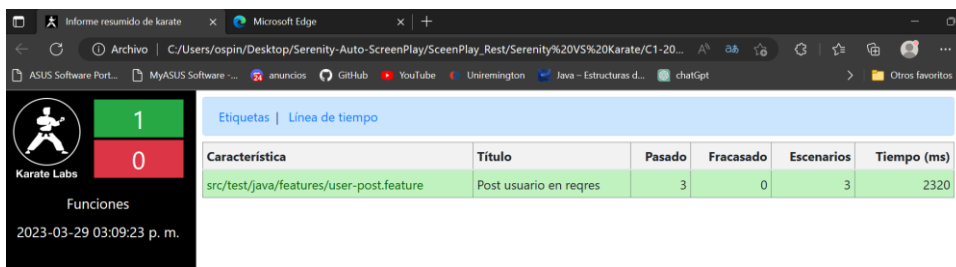
Prueba realizado en karate

```
Tests passed: 3 of 3 tests - 2 sec 318 ms

Test Results 2 sec 318 ms
1 > Content-Type: application/json; charset=UTF-8
1 > Content-Length: 30
1 > Host: regres.in
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/11.0.15)
1 > Accept-Encoding: gzip, deflate
{"name":"Luis","job":"Doctor"}

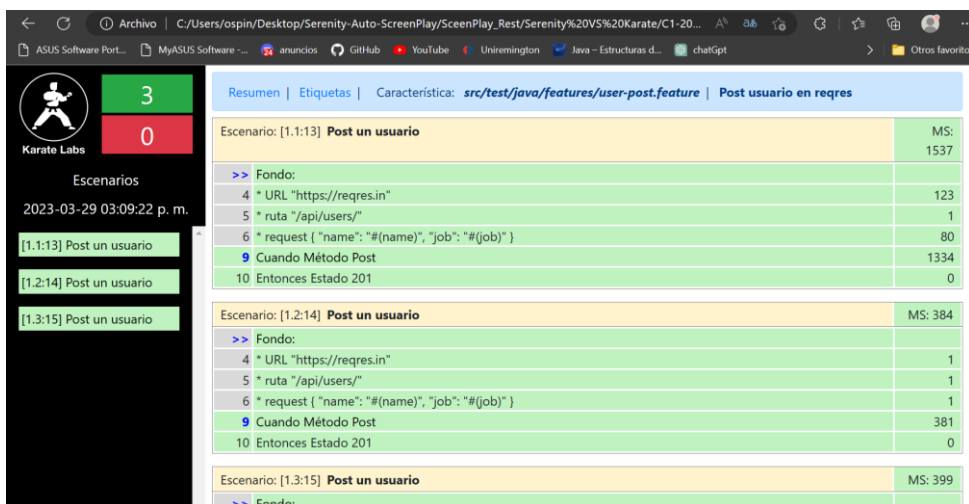
15:09:22.132 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConne
{s}->https://regres.in:443][total available: 0; route allocated: 0 of 5;
15:09:22.132 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConne
2][route: {s}->https://regres.in:443][total available: 0; route allocated
15:09:22.132 [main] DEBUG org.apache.http.impl.execchain.MainClientExec -
.in:443
15:09:22.132 [main] DEBUG org.apache.http.impl.conn.DefaultHttpClientConne
```

Reporte Karate



The screenshot shows the Karate Labs interface with a sidebar on the left containing a logo, a counter showing '1' in green and '0' in red, and the text 'Funciones' and '2023-03-29 03:09:23 p. m.'. The main area has a header with 'Etiquetas' and 'Línea de tiempo'. Below is a table with columns: Característica, Título, Pasado, Fracasado, Escenarios, and Tiempo (ms).

Característica	Título	Pasado	Fracasado	Escenarios	Tiempo (ms)
src/test/java/features/user-post.feature	Post usuario en reqres	3	0	3	2320



The screenshot shows the Karate Labs interface with a sidebar on the left containing a logo, a counter showing '3' in green and '0' in red, and the text 'Escenarios' and '2023-03-29 03:09:22 p. m.'. The main area has a header with 'Resumen', 'Etiquetas', and 'Característica: src/test/java/features/user-post.feature | Post usuario en reqres'. Below is a table with columns: Escenario, MS, and a list of steps.

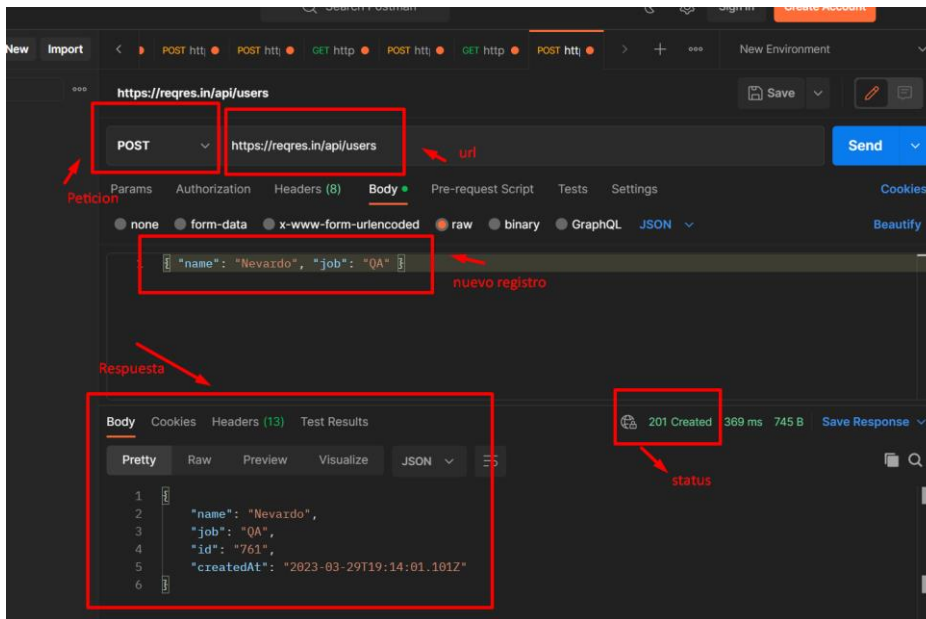
Escenario	MS
[1.1:13] Post un usuario	1537
>> Fondo:	
4 * URL "https://reqres.in"	123
5 * ruta "/api/users/"	1
6 * request ("name": "#(name)", "job": "#(job)")	80
9 Cuando Método Post	1334
10 Entonces Estado 201	0
[1.2:14] Post un usuario	384
>> Fondo:	
4 * URL "https://reqres.in"	1
5 * ruta "/api/users/"	1
6 * request ("name": "#(name)", "job": "#(job)")	1
9 Cuando Método Post	381
10 Entonces Estado 201	0
[1.3:15] Post un usuario	399
>> Fondo:	

Pruebas realizadas en Postman, petición Post de recurso del servicio REQRES

con la siguiente ruta

base: <https://reqres.in/>

recurso: api/users



Caso 2 por Karate.

Actualizar un registros

Caso realizado en karate


```
Feature: PUT usuario reqres

Scenario: Put user
  Given url "https://reqres.in" + "/api/users/" + "2"
  And request { "name": "Antonio", "job": "zion resident" }
  When method PUT
  Then status 200
  And match $.name == 'Antonio'
```

Prueba realizada en karate

```
Tests passed: 1 of 1 test - 1 sec 485 ms
Test Results 1 sec 485 ms
%2Fpt9q026LbzEXvFwfS0hrsHunM%2BEVqd0hmhZRuW%3D%3D"}], "group": "cf-nel", "max_age": 604800
1 < NEL: {"success_fraction": 0, "report_to": "cf-nel", "max_age": 604800}
1 < Vary: Accept-Encoding
1 < Server: cloudflare
1 < CF-RAY: 7afac462a84dad5-MIA
{"name": "Antonio", "job": "zion resident", "updatedAt": "2023-03-29T20:19:46.451Z"}
15:19:48.216 [main] DEBUG com.jayway.jsonpath.internal.path.CompiledPath - Evaluating p
15:19:48.296 [main] DEBUG org.thymeleaf.TemplateEngine - [THYMELEAF] INITIALIZING TEMPL
15:19:48.613 [main] DEBUG org.thymeleaf.TemplateEngine.CONFIG - Initializing Thymeleaf
...
[THYMELEAF] TEMPLATE ENGINE CONFIGURATION:
[THYMELEAF] - Thymeleaf version: 3.0.16, dialects: [html, html5, xml, xml5, text]
```

Reporte Karate




1
0

Funciones
2023-03-29 03:19:49

Etiquetas | Línea de tiempo

Característica	Título	Pasado	Fracasado	Escenarios	Tiempo (ms)
src/test/java/features/user-put.feature	PUT usuario reqres	1	0	1	1485



1
0

Escenarios
2023-03-29 03:19:48 p. m.

Resumen | Etiquetas | Característica: src/test/java/features/user-put.feature | PUT usuario reqres

Escenario: [1:3] Put user

4 Dado URL "https://reqres.in" + "/api/users/" + "2"

5 Y request { "name": "Antonio", "job": "zion resident" }

6 Cuando método PUT

15:19:47.450 Solicitud:

1 > PONER https://reqres.in/api/users/2

1 > Content-Type: application/json; charset=UTF-8

1 > Contenido-Longitud: 40

1 > Anfitrión: reqres.in

1 > Conexión: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/11.0.15)

1 > Accept-Encoding: gzip,deflate

{ "name": "Antonio", "job": "zion resident" }

15:19:48.162 Tiempo de respuesta en milisegundos: 706

1 < 200

1 < Fecha: Wed, 29 Mar 2023 20:19:46 GMT

1 < Content-Type: application/json; charset=utf-8

1 < Codificación de transferencia: fragmentado

1 < Conexión: keep-alive

1 < X-Powered-By: Express

1 < Acceso-Control-Permitir-Origen: *

1 < Etag: W/"4f-tY/1H5WavFIjRVcoRs9/t+N+pbI"

1 < Vía: 1.1 vegur

1 < CF-Cache-Status: DYNAMIC

1 < Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=kF%2FtQs0%2FMN65Vw%2BDtWUYVEGBY82zx2iiPcCs3058FpWtPpYzTR6qKD6V%2BaXfbL%2FNjofWlPouYqE8%2Fc31jf%2Fpt9q026LbzEXvFwfS0hrsHunM%2BEVqd0hmhZRuW%3D%3D"}], "group": "cf-nel", "max_age": 604800}

1 < NEL: {"success_fraction": 0, "report_to": "cf-nel", "max_age": 604800}

1 < Vary: Accept-Encoding

1 < servidor: cloudflare

1 < RAYOS CF: 7afac462a84dad5-MIA

{ "name": "Antonio", "job": "zion resident", "updatedAt": "2023-03-29T20:19:46.451Z" }

7 Entonces Estado 200

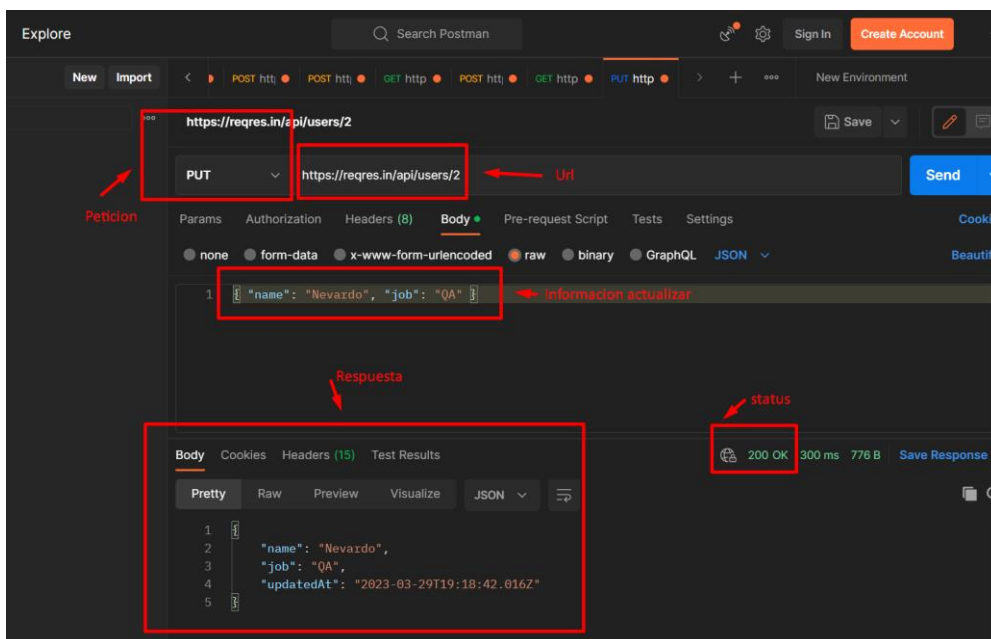
8 Y match \$.name == 'Antonio'

Pruebas realizadas en Postman, petición Put de recurso del servicio REQRES

con la siguiente ruta

base: <https://reqres.in/>

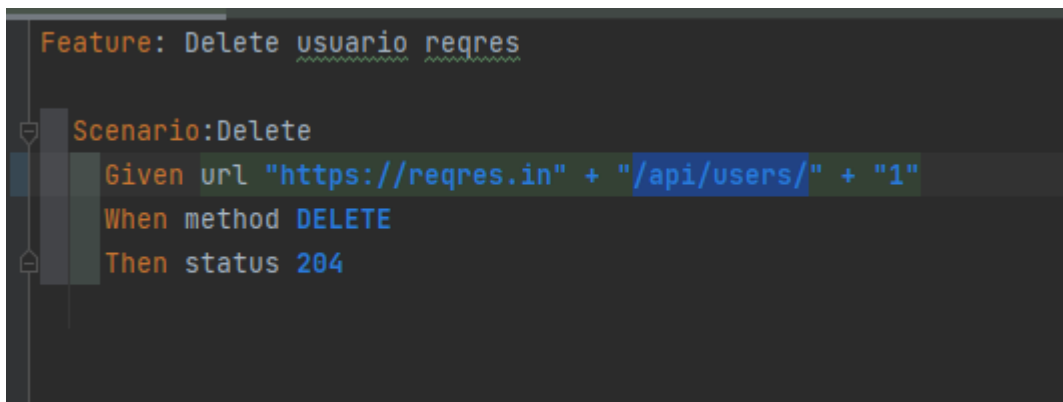
recurso: api/users/2



Caso 3 por Karate.

Eliminar un registro

Caso realizado en karate




Prueba realizada en karate

```
Tests passed: 1 of 1 test - 1 sec 328 ms
Test Results 1 sec 328 ms
15:26:14.990 [main] DEBUG com.intuit.karate - request:
1 > DELETE https://regres.in/api/users/1
1 > Host: regres.in
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/11.0.15)
1 > Accept-Encoding: gzip,deflate
```

```
1 < 204
1 < Date: Wed, 29 Mar 2023 20:26:14 GMT
1 < Content-Length: 0
1 < Connection: keep-alive
1 < X-Powered-By: Express
1 < Access-Control-Allow-Origin: *
1 < Etag: W/"2-vy6p6PvFo4RvsFtPoIWeCReyIC8"
1 < Via: 1.1 vegur
1 < CF-Cache-Status: DYNAMIC
1 < Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=DvZ4D9vvfim%2FxxF8s0grlz%2F06uwcyzbwxto2Ur1SKjHpRS%2BnxFCwhQkeMp0bKogSf0ASmuNhsP3ULFN09ySo1%2B1Q6k%2F309V7L7%2FEIZDTziRdYPa52prIhAGJWw%3D%3D"}],"group":"cf-nel","max_age":604800}}
1 < NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
1 < Server: cloudflare
1 < CF-RAY: 7afad8bc69b00355-MIA
```

Reporte Karate



1


0

Funciones

2023-03-29 03:26:16 p. m.

Etiquetas | Línea de tiempo

Característica	Título	Pasado	Fracasado	Escenarios	Tiempo (ms)
src/test/java/features/user-delete.feature	Eliminar requisitos de usuario	1	0	1	1329



1

0

Escenarios

2023-03-29 03:26:15 p. m.

[1-3] Borrar

Resumen | Etiquetas | Característica: src/test/java/features/user-delete.feature | Eliminar requisitos de usuario

Escenario: [1-3] Eliminar

MS: 1329

4 Dado URL "https://regres.in" + "/api/users/" + "1"

130

5 Cuando método DELETE

1199

```
15:26:14.990 Solicitud:
1 > SUPRIMIR https://regres.in/api/users/1
1 > Anfitrión: regres.in
1 > Conexión: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/11.0.15)
1 > Accept-Encoding: gzip,deflate

15:26:15.725 Tiempo de respuesta en milisegundos: 732
1 < 204
1 < Fecha: Wed, 29 Mar 2023 20:26:14 GMT
1 < Longitud del contenido: 0
1 < Conexión: keep-alive
1 < X-Powered-By: Express
1 < Acceso-Control-Permitir-Origen: *
1 < Etag: W/"2-vy6p6PvFo4RvsFtPoIWeCReyIC8"
1 < Via: 1.1 vegur
1 < CF-Cache-Status: DYNAMIC
1 < Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=DvZ4D9vvfim%2FxxF8s0grlz%2F06uwcyzbwxto2Ur1SKjHpRS%2BnxFCwhQkeMp0bKogSf0ASmuNhsP3ULFN09ySo1%2B1Q6k%2F309V7L7%2FEIZDTziRdYPa52prIhAGJWw%3D%3D"}],"group":"cf-nel","max_age":604800}}
1 < NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
1 < servidor: cloudflare
1 < RAYOS CF: 7afad8bc69b00355-MIA

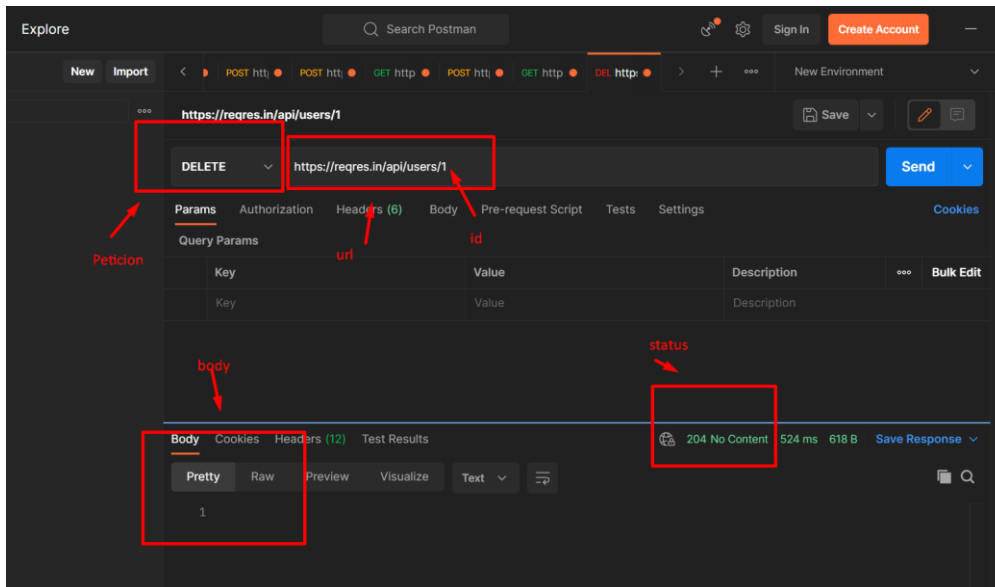
6 Entonces Estado 204
```

Pruebas realizadas en Postman, petición Delete de recurso del servicio REQRES

con la siguiente ruta

base: <https://reqres.in/>

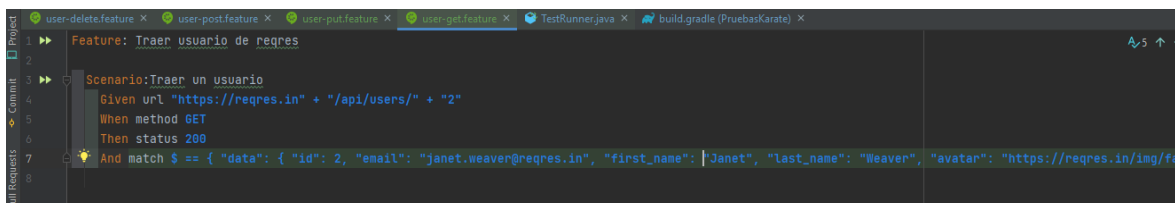
recurso: api/users/1



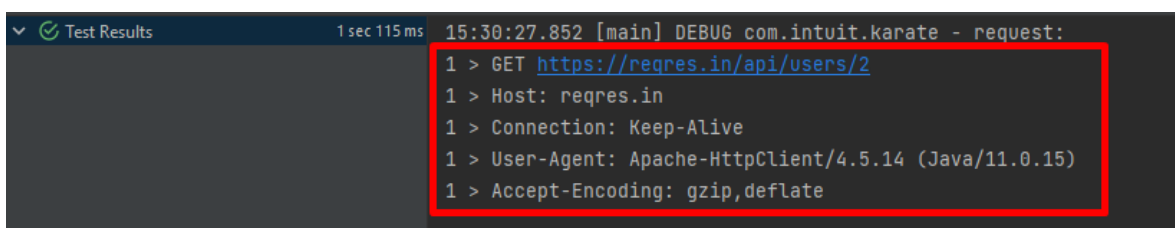
Caso 4 por Karate.

Actualizar un registros

Caso listar un usuario y comparar la informacion en karate




Prueba realizada en karate{



```
15:30:28.292 [main] DEBUG com.intuit.karate - response time in millisec
1 < 200
1 < Date: Wed, 29 Mar 2023 20:30:26 GMT
1 < Content-Type: application/json; charset=utf-8
1 < Transfer-Encoding: chunked
1 < Connection: keep-alive
1 < X-Powered-By: Express
1 < Access-Control-Allow-Origin: *
1 < Etag: W/"118-pbdwwFo9SKNhD3Lx5iHJyngpq00"
1 < Via: 1.1 vegur
1 < Cache-Control: max-age=14400
1 < CF-Cache-Status: HIT
1 < Age: 2885
1 < Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare
.com/report/v3?s=%2BUfcETSerLtIMwyQbFgeN4ETTPHHdSmEgHuidPctmL
... 15:30:28.292 [main] DEBUG com.intuit.karate - response time in millisec
```

```
1 < CF-RAY: 7afae1931c738dea-MIA
{"data":{"id":2,"email":"janet.weaver@reqres.in","first_name":"Janet","last_name":"Weaver","avatar":"https://reqres
.in/img/faces/2-image.jpg"},"support":{"url":"https://reqres.in/support-heading","text":"To keep ReqRes free,
contributions towards server costs are appreciated!"}}
```

Reporte Karate



1

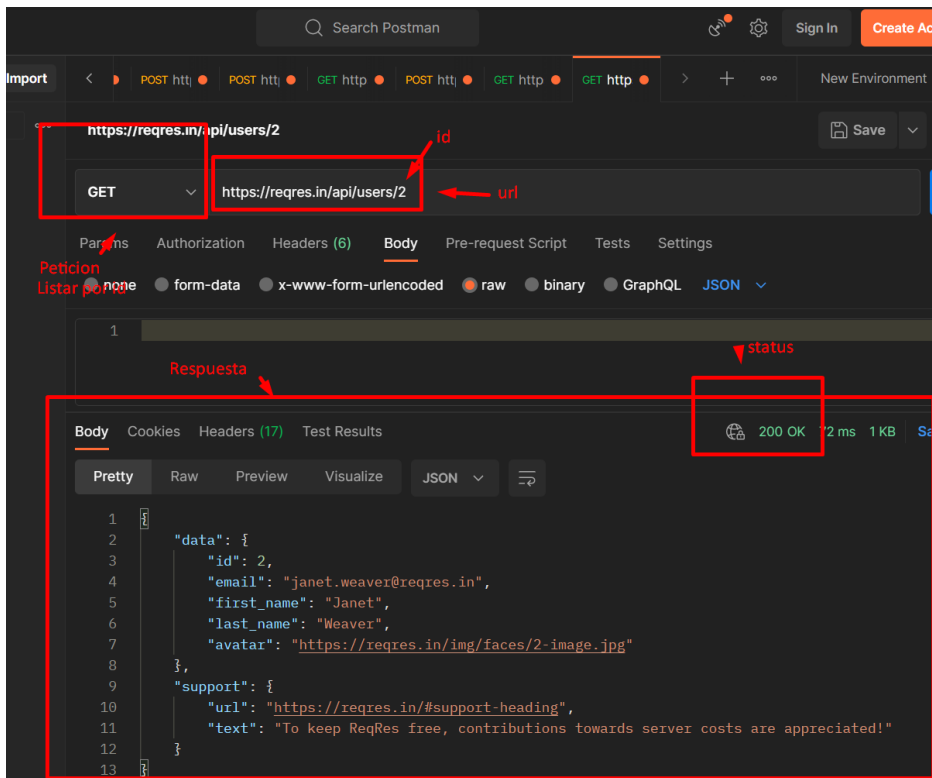
0

Funciones

2023-03-29 03:32:18 p. m.

Etiquetas | Línea de tiempo

Característica	Título	Pasado	Fracasado	Escenarios	Tiempo (ms)
src/test/java/features/user-get.feature	Traer usuario de reqres	1	0	1	1127



Conclusión sobre el taller

Serenity y Karate son diferentes herramientas utilizadas en la automatización de pruebas. Aunque ambas herramientas se pueden usar para la automatización de pruebas, difieren en su enfoque y capacidades. Serenity se enfoca en crear pruebas automatizadas de alta calidad e informes de prueba detallados, mientras que Karate está diseñado para pruebas de API y proporciona una sintaxis de prueba simple, karate no es adecuado para otros tipos de pruebas, como pruebas de rendimiento.