

|                              |   |
|------------------------------|---|
| Javier Alberto Romero Oviedo | Ingeniero de sistemas   |
| Ejercicio práctico- Postman  | Api <a href="https://rickandmortyapi.com/documentation">https://rickandmortyapi.com/documentation</a> |

1. Hacer un listado de todos los personajes, y hacer un test con la información de un personaje que aparezca en esa respuesta (Tener en cuenta que deben hacer una búsqueda entre los diferentes objetos que hay embebidos en la respuesta).

The screenshot shows the Postman interface with a GET request to the endpoint `endpointCharacte / Busqueda de todos los personajes`. The URL is `{{urlMorty}}/character`. The tests section shows two tests: "El código de estado es 200" and "Realiza búsqueda de personaje por nombre y verifica su Species", both of which passed.

```

1 pm.test("El código de estado es 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Realiza búsqueda de personaje por nombre y verifica su Species", function () {
5   var res = pm.response.json();
6   var name = res.results.find(n => n.name == pm.globals.get('name'))
7   pm.expect(name.species).to.eql("Human")
8 });

```

Test Results (2/2):

- PASS El código de estado es 200
- PASS Realiza búsqueda de personaje por nombre y verifica su Species

2. Hacer un listado de personajes por página (en la documentación se explica cómo hacer paginación), y hacer un test por aparición de personaje en esta página.

The screenshot shows the Postman interface with a GET request to the endpoint `endpointCharacte / Paginación`. The URL is `{{urlMorty}}/character?page={{numPage}}`. The tests section shows two tests: "El código de estado es 200" and "Personajes obtenidos por página", both of which passed.

```

1 pm.test("El código de estado es 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Personajes obtenidos por página", function () {
5   const idsPersonajesObtenidosPorPagina = pm.response.json().results;
6
7   idsPersonajesObtenidosPorPagina.forEach(function (idPersonajePorPagina) {
8     console.log(idPersonajePorPagina);
9     pm.expect(idsPersonajesObtenidosPorPagina).to.include(idPersonajePorPagina);
10  });
11 });
12
13
14

```

Test Results (2/2):

- PASS El código de estado es 200
- PASS Personajes obtenidos por página

|                              |   |
|------------------------------|---|
| Javier Alberto Romero Oviedo | Ingeniero de sistemas   |
| Ejercicio práctico- Postman  | Api <a href="https://rickandmortyapi.com/documentation">https://rickandmortyapi.com/documentation</a> |

### 3. Traer un personaje por id, y hacer un test correspondiente

endpointCaracte

- GET Búsqueda de todos los persona...
- GET Paginación
- GET Personaje por Id
- GET Personajes por Id
- GET idNoExistentesMultiples

GET `{{urlMorty}}/character/{{idPersonaje}}` **Send**

Params Authorization Headers (7) Body Pre-request Script **Tests** Settings Cookies

```

1 pm.test("El código de estado es 200", function () {
2   pm.response.to.have.status(200);
3 });
4 const response=pm.response.json();
5 const personID=Object.keys(response).includes('id');
6
7 pm.test('validar id', ()=>{
8   pm.expect(true).to.eql(personID);
9 })
10
11
12
13

```

Body Cookies Headers (12) **Test Results (2/2)** Status: 200 OK Time: 614 ms Size: 782 B **Save Response**

All Passed Skipped Failed

- PASS** El código de estado es 200
- PASS** validar id

orking with your team 67%

Next: Invite at least 1 person. [Invite](#)

### ➤ Traer un personajes por id, y hacer un test correspondiente

endpointCaracte / Personajes por Id

GET `{{urlMorty}}/character/{{idMul}}` **Send**

Params Authorization Headers (7) Body Pre-request Script **Tests** Settings Cookies

```

1 pm.test("Verificar código de estado 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Validar varios ID", function () {
5   var ids= pm.globals.get('idMul').split(',').map(Number);
6   var idsObtenidos= pm.response.json().map(id=>id.id);
7   pm.expect(ids.length).to.eql(idsObtenidos.length);
8   ids.forEach(function(idPer) {
9     pm.expect(ids).to.include(idPer);
10  });
11  console.log("ID's Obtenidos: "+idsObtenidos);
12 });
13
14

```

Body Cookies Headers (14) **Test Results (2/2)** Status: 200 OK Time: 586 ms Size: 1.73 KB **Save Response**

All Passed Skipped Failed

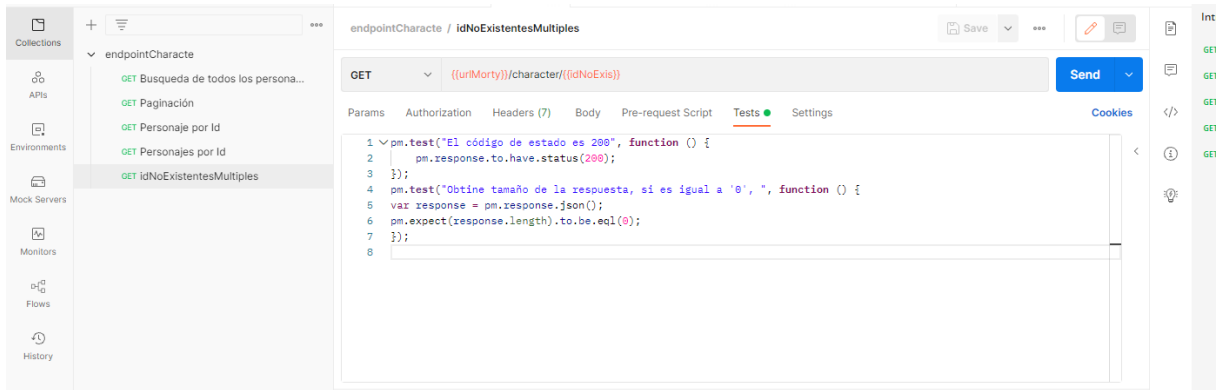
- PASS** Verificar código de estado 200
- PASS** Validar varios ID

Start working with your team 67%

Next: Invite at least 1 person. [Invite](#)

|                              |   |
|------------------------------|---|
| Javier Alberto Romero Oviedo | Ingeniero de sistemas   |
| Ejercicio práctico- Postman  | Api <a href="https://rickandmortyapi.com/documentation">https://rickandmortyapi.com/documentation</a> |

4. Al momento de buscar múltiples Id, ingresando id's que no existen nos da una respuesta, la cual es un array vacío, hacer un test en donde se valide que se tiene una respuesta vacía



Documentación Postman:

<https://documenter.getpostman.com/view/19965346/Uz5ArJiK>

URL-Ejercicio: <https://www.postman.com/universal-equinox-765712/workspace/rickandmortyapi-com/overview>