Informe de pruebas Api con Rest-Assured

Diseño de casos de prueba

Link de notion: Informe de pruebas Api con Rest-Assured

Como se indica en los requerimientos del trabajo, se seleccionan 4 servicios, dos de la pagina https://jsonplaceholder.typicode.com/ y otros dos de la página http://reqres.in/, entonces a partir de la selección de estos servicios se generan los casos de prueba para cada uno:

Delete Json

Para este servicio se generan dos casos de prueba, uno en donde se intenta eliminar un post con un ID adecuado como lo sería un ID de 1, y otro con ID que no tenga mucho sentido que sería un ID negativo, como -1.

```
Feature: Post delete of JSON api
  Me as a user of the json service
  want to handle the posts of the site
  for make a correct handle of the posts
  #Juan Garcia

  @validID
  Scenario: Deleting a post
    Given the JSONPlaceholder API endpoint for posts is accessible
    And there exists a post with id 1
    When sends a DELETE request to the post endpoint with id 1
    Then the response delete status code must be 200 deleting the user
  @NoValidID
  Scenario: Deleting a post with invalid ID
    Given the JSONPlaceholder API endpoint for posts is accessible
    And there exists no post with id -1
    When sends a DELETE request to the post endpoint with id -1
    Then the response delete status code must be 404 indicating resource not found
```

Get Users of JSON API

Para este servicio se genera un caso de prueba en el cual se intenta obtener la lista de usuarios de la API JSON.

```
Feature: Get users of JSON API
  Me as a user of the JSON service
  want to retrieve the list of users from the site
  for interaction with user data effectively
  #Juan Garcia

  Scenario: Retrieving users
    Given the JSONAPI endpoint for users is accessible
    When sends a GET request to the users endpoint
    Then the response status code must be 200
    And the response should contain at least one user with a valid name
```

List Users

Este servicio ofrece la posibilidad de gestionar la lista de usuarios, con casos de prueba que abarcan diferentes aspectos de esta funcionalidad.

```
Feature: List users
  Me as a user of the get service
  want to handle the list of users
  for manage his information

  Background:
  Given the reqres API endpoint for users is accessible

  @GetAll
  Scenario: Getting users from page 2
    When sends the query Get to the system with page parameter set to 2
    Then the response status code should be 200
    And the response should contain user data

  @VerifyInfo
  Scenario: Verifying user information
    When sends the query Get to the system with page parameter set to 2
    Then the response status code should be 200
    And each user in the response should have a valid id
    And each user in the response should have a valid email

  @EmptyData
  Scenario: Requesting users with an invalid page number
    When sends the query Get to the system with page parameter set to 5
    Then the response status code should be 200
    And the response body should not contain user data
```

User Register

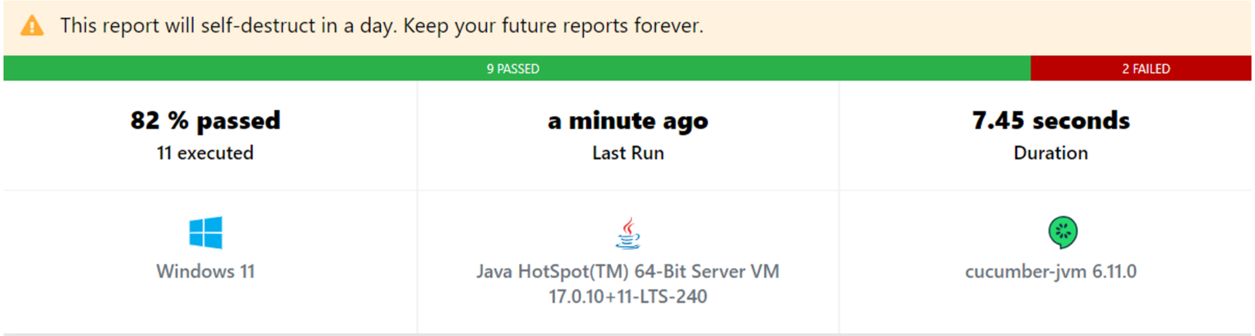
Este servicio permite a los usuarios crear nuevas cuentas en la aplicación Reqres, con casos de prueba que abarcan diferentes situaciones al crear una cuenta.

```
Feature: User register
  As user of of the reqres app
  want to create a new account
  For access to the platform.

  Background:
    Given the user is currently on the reqres.in platform.

  @CompleteInfo
  Scenario Outline: Successful Account Creation
    When the user correctly enters their details
      | name   | job   |
      | <name> | <job> |
    And send the request for the user creation
    Then they should be able to view their information along
with a generated ID
    And the response should indicate a valid status code of 2
01
    Examples:
      | name                  | job   |
      | juanito@gmail.com     | ladron |
      | pedro@gmail.com       | navaja |
      | carlitos123@gmail.com | rugrats |
      | hola_@gmail.com       | correo |



    @BadRequest
  Scenario: Attempt to Create Account with Only Job Specified
    When the user attempts to create an account with only job
specified
      | name | job     |
      |      | boss  |
    And send the request for the user creation
    Then they should receive a response with a status code of
400 indicating bad request
```

Resultados

Resumen de la Ejecución

- **Fecha de la Ejecución:** 07/05/2024
- **Duración de la Ejecución:** 7.45 segundos
- **Entorno de Ejecución:** Windows 11
- **Versión de la Aplicación:** Java HotSpot(TM) 64-Bit Server VM 17.0.10+11-LTS-240
- **Número de Escenarios Ejecutados:** 11
- **Resultados de la Ejecución:** 82% de los casos pasaron



⚠ This report will self-destruct in a day. Keep your future reports forever.

9 PASSED 2 FAILED

82 % passed 11 executed	**a minute ago** Last Run	**7.45 seconds** Duration
Windows 11	Java HotSpot(TM) 64-Bit Server VM 17.0.10+11-LTS-240	cucumber-jvm 6.11.0

Detalles de la Ejecución

Escenarios Ejecutados

Escenarios Exitosos

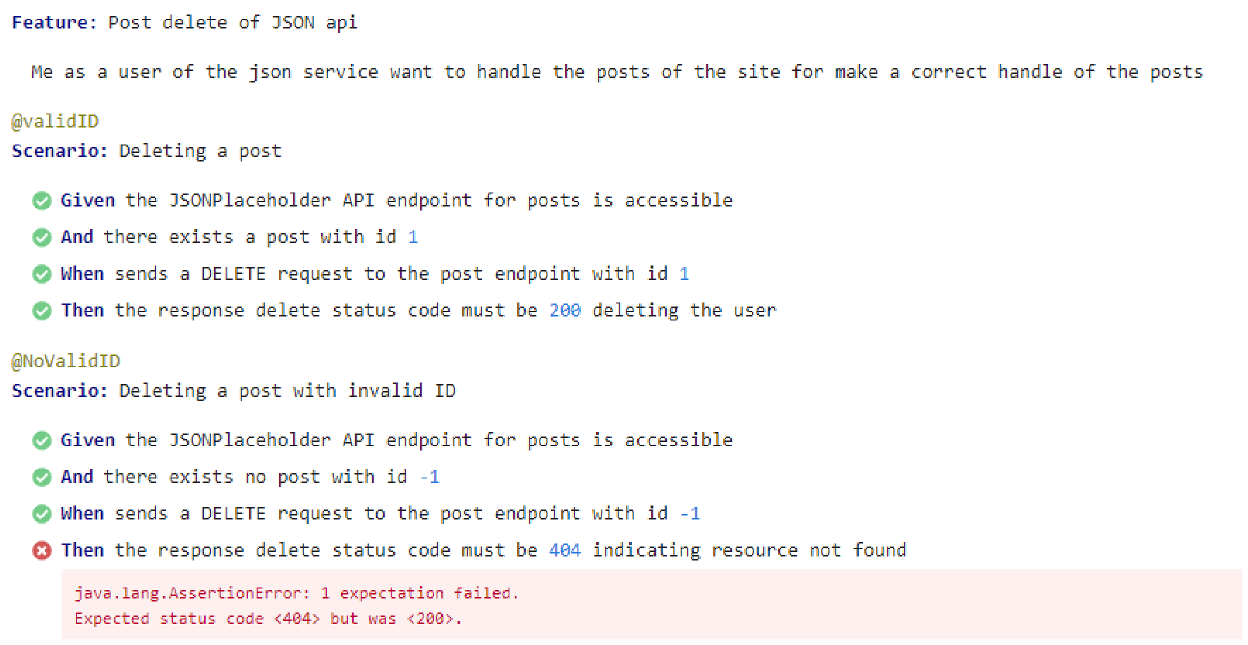
- En total fueron 9 escenarios exitosos, a continuación el nombre de los escenarios exitosos:
  - **Scenario:Deleting a post**
  - **Scenario:Getting users from page 2**
  - **Scenario:Verifying user information**
  - **Scenario:Requesting users with an invalid page number**
  - **Scenario Outline:Successful Account Creation**
  - **Scenario:Retrieving users**

Escenarios Fallidos

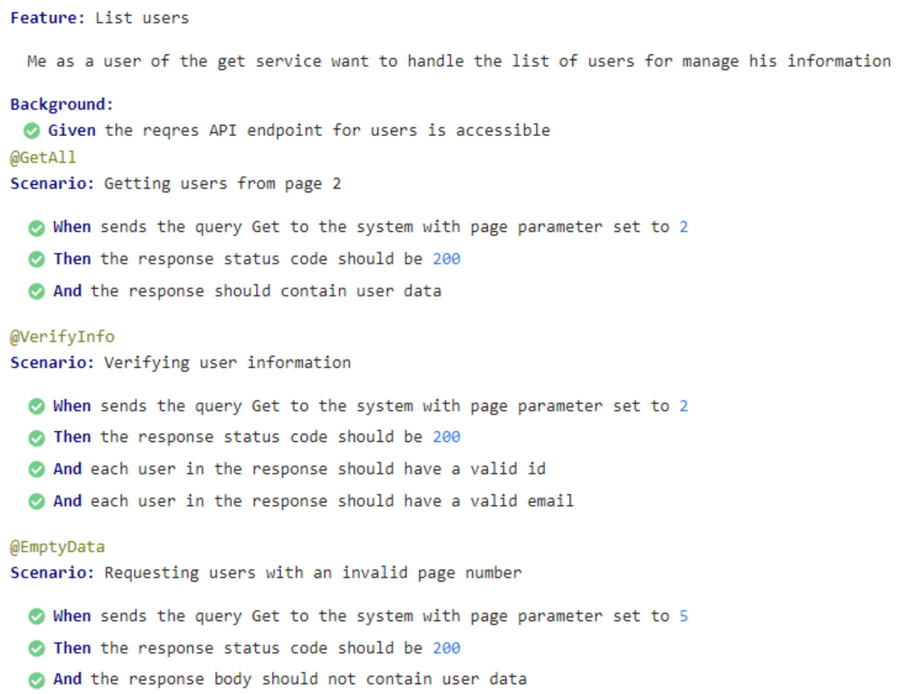
- Fueron 2 escenarios que no pasaron, a continuación el nombre de escenarios fallidos
    - **Scenario: Attempt to Create Account with Only Job Specified**
    - **Scenario: Deleting a post with invalid ID**

Evidencias

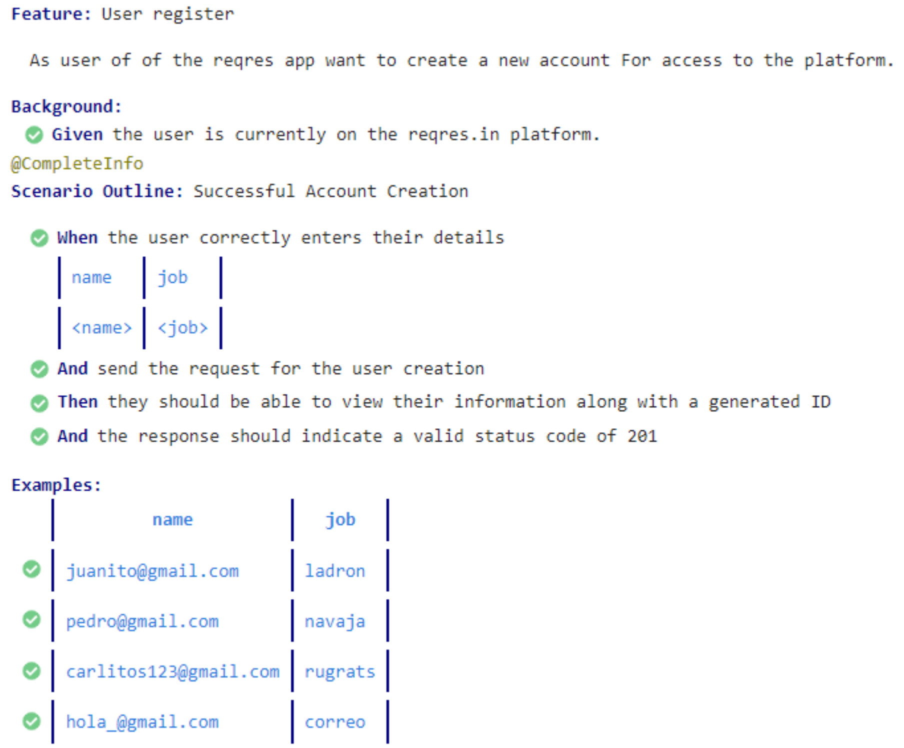
Fuente: reporte generado por cucumber, al ejecutar el runner AllTest, se pueden correr todos los test del proyecto, y por último se podrá generar el mismo informe utilizado para estas evidencias:



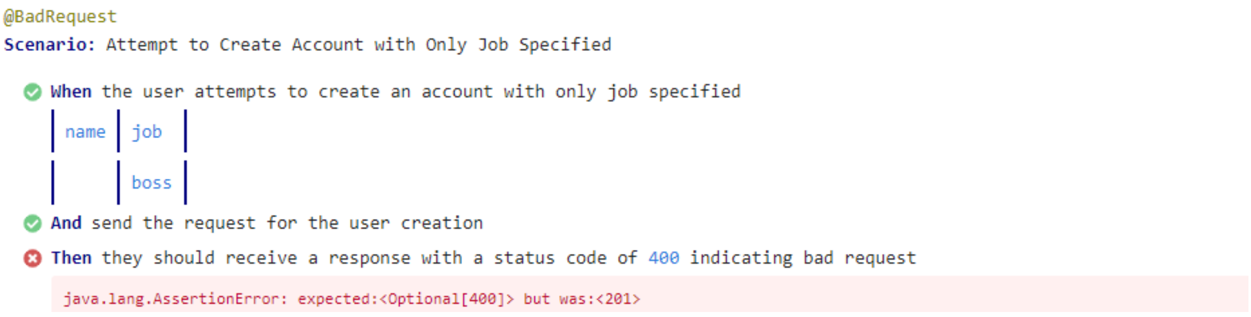
```
 Feature: Post delete of JSON api

   Me as a user of the json service want to handle the posts of the site for make a correct handle of the posts

 @validID
 Scenario: Deleting a post

   ✓ Given the JSONPlaceholder API endpoint for posts is accessible
   ✓ And there exists a post with id 1
   ✓ When sends a DELETE request to the post endpoint with id 1
   ✓ Then the response delete status code must be 200 deleting the user

 @NoValidID
 Scenario: Deleting a post with invalid ID

   ✓ Given the JSONPlaceholder API endpoint for posts is accessible
   ✓ And there exists no post with id -1
   ✓ When sends a DELETE request to the post endpoint with id -1
   ✗ Then the response delete status code must be 404 indicating resource not found
       java.lang.AssertionError: 1 expectation failed.
       Expected status code <404> but was <200>.
```



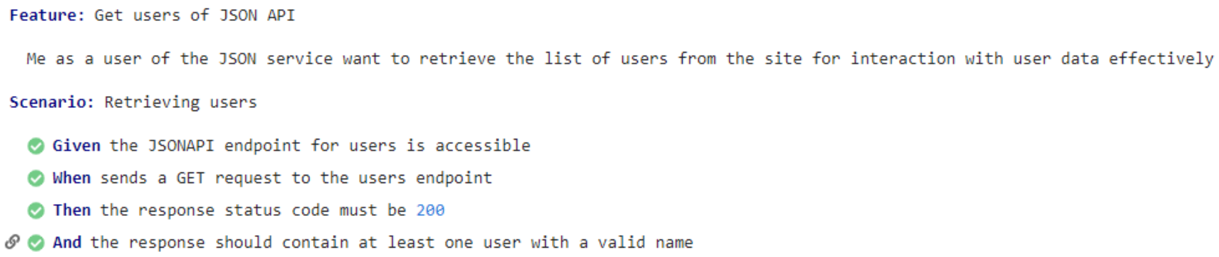
```
Feature: List users

  Me as a user of the get service want to handle the list of users for manage his information

Background:
 ✓ Given the reqres API endpoint for users is accessible
@GetAll
Scenario: Getting users from page 2

  ✓ When sends the query Get to the system with page parameter set to 2
  ✓ Then the response status code should be 200
  ✓ And the response should contain user data

@VerifyInfo
Scenario: Verifying user information

  ✓ When sends the query Get to the system with page parameter set to 2
  ✓ Then the response status code should be 200
  ✓ And each user in the response should have a valid id
  ✓ And each user in the response should have a valid email

@EmptyData
Scenario: Requesting users with an invalid page number

  ✓ When sends the query Get to the system with page parameter set to 5
  ✓ Then the response status code should be 200
  ✓ And the response body should not contain user data
```



```
Feature: User register

  As user of of the reqres app want to create a new account For access to the platform.

Background:
 ✓ Given the user is currently on the reqres.in platform.
@CompleteInfo
Scenario Outline: Successful Account Creation

  ✓ When the user correctly enters their details
     | name   | job   |
     | <name> | <job> |
  ✓ And send the request for the user creation
  ✓ Then they should be able to view their information along with a generated ID
  ✓ And the response should indicate a valid status code of 201

Examples:
     |         name          |   job   |
   ✓ | juanito@gmail.com     | ladron  |
   ✓ | pedro@gmail.com       | navaja  |
   ✓ | carlitos123@gmail.com | rugrats |
   ✓ | hola_@gmail.com       | correo  |
```



```
@BadRequest
Scenario: Attempt to Create Account with Only Job Specified

  ✓ When the user attempts to create an account with only job specified
     | name | job  |
     |      | boss |
  ✓ And send the request for the user creation
  ✗ Then they should receive a response with a status code of 400 indicating bad request
     java.lang.AssertionError: expected:<Optional[400]> but was:<201>
```



```
Feature: Get users of JSON API

  Me as a user of the JSON service want to retrieve the list of users from the site for interaction with user data effectively

Scenario: Retrieving users

  ✓ Given the JSONAPI endpoint for users is accessible
  ✓ When sends a GET request to the users endpoint
  ✓ Then the response status code must be 200
  ✓ And the response should contain at least one user with a valid name
```

Detalles de los Casos de Prueba fallidos

Escenario: Deleting a post with invalid ID

En este escenario, se intentó eliminar un post con un ID inválido mediante una solicitud DELETE a la API de JSONPlaceholder. Se esperaba que la API respondiera con un código de estado 404 (indicando que el recurso no se encontró), pero en cambio, la respuesta fue un código de estado 200 (indicando una solicitud exitosa). Esto indica que la API no manejó correctamente la solicitud de eliminación de un post con un ID inválido.

Escenario: Attempt to Create Account with Only Job Specified

En este escenario, se intentó crear una cuenta especificando solo el trabajo (job) y dejando el campo de nombre (name) en blanco. Se esperaba que la API respondiera con un código de estado 400 (indicando una solicitud incorrecta), pero en cambio, la respuesta fue un código de estado 201 (indicando una creación exitosa). Esto indica que la API no manejó correctamente la solicitud de creación de cuenta y no validó los datos de entrada de manera adecuada.