|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-001 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado a la solicitud de información de los usuarios registrados |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: GET Users |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo GET para listar todos los usuarios registrados en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición, así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 200 * KeyWord: Ok * Se deberá observar la lista de los usuarios registrados dentro de la API * Uno de los usuarios de la Lista lleva por nombre “George” |
| 1. **Resultado Obtenido** | Se obtuvo un status code “200” con la keyword “Ok”  Se observa la lista de usuarios registrados en la API, y uno de ellos es “George”. |
| 1. **Recomendaciones** | No aplica. |

|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-002 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado a la solicitud de información de un Usuario en específico, este usuario deberá estar registrado en el BackEnd |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: GET an User |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo GET para obtener la información de un usuario registrados en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición, así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 200 * KeyWord: Ok * Se deberá observar la información del usuario registrado dentro de la API * Validar que al menos uno de los datos obtenidos coincida con los del usuario requerido |
| 1. **Resultado Obtenido** | Se obtuvo el status code “200” con la palabra Keyword “Ok”.  Se observa la información del usuario registrado en la API.  Se obtiene el usuario buscado “Michael”. |
| 1. **Recomendaciones** | No aplica. |

|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-003 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado a la creación de un Usuario exitosamente, este usuario deberá estar registrado en el BackEnd al final de la transacción |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: POST Create an User |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo Post para la creación de un usuario en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición, así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 201 * KeyWord: Created * Se deberá observar la información del usuario registrado dentro de la API * Validar que al menos uno de los datos de la respuesta coincida con los del usuario insertado |
| 1. **Resultado Obtenido** | Se obtuvo un status code “201” con la keyword “Created”.  Se observa la información del usuario creado.  Se valida que el usuario insertado con nombre “Ivette” coincide con la respuesta. |
| 1. **Recomendaciones** | No aplica. |

|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-004 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado al Registro de un Usuario exitosamente, este usuario deberá quedar registrado en el BackEnd al final de la transacción |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: POST Register an User |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo Post para Registrar un usuario en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición, así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 200 * KeyWord: Ok * Se deberá observar la información del usuario registrado dentro de la API * Validar que en la respuesta se obtiene el id de registro |
| 1. **Resultado Obtenido** | Cuando se ejecuta una petición del tipo POST se esperaría un Status Code 201 relacionado con una solicitud de creación del registro, sin embargo, estamos recibiendo un status code 200 que no corresponde a la creación sino mas bien a una modificación de un registro |
| 1. **Recomendaciones** | Se recomienda al equipo de desarrollo poder revisar los flujos de los request de creación de nuevos registros en el BackEnd |

|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-005 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado la actualización de un Usuario exitosamente, este usuario deberá estar registrado en el BackEnd antes de realizar la transacción |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: PUT Update an User |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo PUT para Actualizar un usuario en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición, así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 200 * KeyWord: Ok * Se deberá observar la información del usuario registrado dentro de la API |
| 1. **Resultado Obtenido** | Se obtuvo un Status Code “200” y la keyword “Ok”.  Se observa la información del usuario. |
| 1. **Recomendaciones** | No aplica. |

|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-006 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado la actualización de un Recurso exitosamente, este usuario deberá estar registrado en el BackEnd antes de realizar la transacción |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: PUT Update a Resource |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo PUT para Actualizar un recurso en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 200 * KeyWord: Ok * Se deberá observar la actualización registrada dentro de la API |
| 1. **Resultado Obtenido** | Se obtuvo el status code “200” y la keyword “Ok”.  Se observa la actualización del recurso en la API. |
| 1. **Recomendaciones** | No aplica. |

|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-007 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado la eliminación de un usuario exitosamente, este usuario deberá estar registrado en el BackEnd antes de realizar la transacción |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: DELETE Delete an User |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo DELETE para Eliminar un usuario en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 204 * KeyWord: Deleted |
| 1. **Resultado Obtenido** | Se obtuvo el status code “204” y la keyword “Deleted”. |
| 1. **Recomendaciones** | No aplica. |

|  |  |
| --- | --- |
| 1. **Nombre del Analista** | Stefany |
| 1. **Caso de prueba** | ReqresAPI-008 |
| 1. **Objetivo** | Verificar el funcionamiento del EndPoint de la API Reqres relacionado la eliminación de un usuario de manera fallida, este usuario deberá estar registrado en el BackEnd antes de realizar la transacción |
| 1. **Tipo de Prueba** | Prueba Funcional (Unitaria-BackEnd) |
| 1. **Escenario** | API Request: DELETE Delete an User |
| 1. **Precondiciones** | * Acceso garantizado a la conexión de la API * Se debe tener instalado POSTMAN como herramienta de pruebas de API * IDE de Desarrollo: Intellij IDEA v2024 * Java JDK v21 * Karate-Apache v1.1.0 * Cucumber v5.8.0 |
| 1. **Paso a Paso** | 1. Abrir Postman 2. Generar una petición de tipo DELETE para Eliminar un usuario que no exista en el BackEnd, siguiendo la documentación del Swagger: <https://reqres.in/api-docs/#/default/post_register> 3. Comprobar la respuesta de la petición así como el código de estatus y el resultado de la transacción 4. Generar el escenario de automatización de este caso de prueba, empleando el Framework diseñado en Intellij 5. Validar la respuesta de la petición y comprobar que el resultado esperado (Swagger) es el mismo que el resultado obtenido de la transacción |
| 1. **Resultado Esperado** | * Status Code: 400 * KeyWord: Not Found |
| 1. **Resultado Obtenido** | El resultado obtenido es un Status Code 204, sin embargo, si el usuario no existe en la BBDD debería arrojar un error 400 “Not Found” |
| 1. **Recomendaciones** | Se recomienda revisar el flujo de borrado para que no obtengamos un falso positivo con usuarios que no existen en el BackEnd |