



Proyecto Screenplay Rest paginas reqres.in y jsonplaceholder.typicode.com

Plan de pruebas automatizadas

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | Julian Alberto Avila Navarro | Documento plan de pruebas | Enero 05 de 2022 |

Tabla de Contenidos

[1.Introducción](#_Toc92297162)

[2. Alcance](#_Toc92297163)

[3. Roles y Responsabilidades](#_Toc92297164)

[5. Diseño de casos de prueba](#_Toc92297165)

[6. Ambiente y Herramientas de Pruebas](#_Toc92297166)

[6.1 Herramientas de Pruebas](#_Toc92297167)

[6.2 Ambiente de Pruebas](#_Toc92297168)

[7. Criterios de Entrada y Salida](#_Toc92297169)

[7.1 Criterios de Entrada](#_Toc92297170)

[7.2 Criterios de Salida](#_Toc92297171)

[8. Planificación de ejecución de las pruebas](#_Toc92297172)

[9. Reporte de Pruebas](#_Toc92297173)

# 1.Introducción

En esta Estrategia para la realización de pruebas automatizadas se describe el alcance de las pruebas, el ambiente de pruebas, los recursos necesarios, las herramientas a utilizar y los riesgos de las pruebas del proyecto Screenplay Rest paginas reqres.in y jsonplaceholder.typicode.com.

El objetivo de este trabajo es generar un plan de pruebas automatizada para los metodos Put y Delete de la página web <https://reqres.in/> y los metodos Put y Get de la página web <https://jsonplaceholder.typicode.com/> por medio de Serenity BDD- REST. Dicho proyecto se trabajará bajo la metodología ágil scrum y contará con 2 Sprints, cada uno de dos semanas.

# 2. Alcance

Se realizarán pruebas automatizadas a las funcionalidades seleccionadas durante la planificación de cada sprint usando Serenity BDD- REST.

**Alcance Sprint 1:**

Las pruebas del primer Sprint incluyen la verificación de la siguiente historia de usuario HU001 – Probar el método Put con el recurso api/users/2.

\*\*Criterios de Aceptación 1: \*\*

1. Verificar que se pueda realizar un Put al recurso /api/users/2.
2. Verificar el código de estatus http.
3. Verificar que la respuesta corresponda al recurso /api/users/2.

Las pruebas del primer Sprint incluyen la verificación de la siguiente historia de usuario HU002 – Probar el método Delete con el recurso api/users/2.

\*\*Criterios de Aceptación 2: \*\*

1. Verificar que se pueda realizar un Delete al recurso /api/users/2.
2. Verificar el código de estatus http.
3. Verificar que la respuesta corresponda al recurso /api/users/2.

**Alcance Sprint 2:** Las pruebas del primer Sprint incluyen la verificación de las siguientes historias de usuarios:

Las pruebas del primer Sprint incluyen la verificación de la siguiente historia de usuario HU003 – Probar el método Put con el recurso /posts/1.

\*\*Criterios de Aceptación 3: \*\*

1. Verificar que se pueda realizar un Put al recurso /posts/1.
2. Verificar el código de estatus http.
3. Verificar que la respuesta corresponda al recurso /posts/1.

Las pruebas del primer Sprint incluyen la verificación de la siguiente historia de usuario HU004 – Probar el método Get con el recurso /posts/1.

\*\*Criterios de Aceptación 4: \*\*

1. Verificar que se pueda realizar un Get al recurso posts/1.
2. Verificar el código de estatus http.
3. Verificar que la respuesta corresponda al recurso /posts/1.

# 3. Roles y Responsabilidades

|  |  |
| --- | --- |
| **Roles** | **Responsabilidades** |
| Manager de QA | Verificar que lo realizado corresponda. |
| Ingeniero QA de Automatización/ Analista QA | Planificación y monitoreo de las pruebas automatizadas  Reporte de Defectos  Reporte de progreso de las pruebas  Diseño e implementación de las pruebas.  Ejecución de las pruebas automatizadas.  Reporte de resultados de las pruebas. |

4. Riesgos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CA** | **Historia de usuario 1** | **Probabilidad de Ocurrencia**  **(1-4)** | **Impacto**  **(1-4)** | **Riesgo** |
| 1 | Verificar que se pueda realizar un Put al recurso /api/users/2 | 1 | 2 | 2 |
| 2 | Verificar el código de estatus http. | 1 | 4 | 4 |
| 3 | Verificar que la respuesta corresponda al recurso /api/users/2. | 2 | 2 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CA** | **Historia de usuario 2** | **Probabilidad de Ocurrencia**  **(1-4)** | **Impacto**  **(1-4)** | **Riesgo** |
| 1 | Verificar que se pueda realizar un Delete al recurso /api/users/2 | 1 | 2 | 2 |
| 2 | Verificar el código de estatus http. | 1 | 4 | 4 |
| 3 | Verificar que la respuesta corresponda al recurso /api/users/2. | 2 | 2 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CA** | **Historia de usuario 3** | **Probabilidad de Ocurrencia**  **(1-4)** | **Impacto**  **(1-4)** | **Riesgo** |
| 1 | Verificar que se pueda realizar un Get al recurso /posts/1. | 2 | 2 | 4 |
| 2 | Verificar el código de estatus http. | 1 | 4 | 4 |
| 3 | Verificar que la respuesta corresponda al recurso /posts/1. | 2 | 2 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CA** | **Historia de usuario 4** | **Probabilidad de Ocurrencia**  **(1-4)** | **Impacto**  **(1-4)** | **Riesgo** |
| 1 | Verificar que se pueda realizar un Put al recurso /posts/1. | 1 | 2 | 2 |
| 2 | Verificar el código de estatus http. | 1 | 4 | 4 |
| 3 | Verificar que la respuesta corresponda al recurso /posts/1. | 2 | 2 | 4 |

# 5. Diseño de casos de prueba

A continuación, se enuncian los diseños de casos de prueba HU001 y HU002:

**Feature:** Actualizar un recurso y borrar datos

como un usuario registrado del sistema quiero actualizar un recurso del sistema y luego borrar un registro

con el fin de saber de verificar los servicios del sistema

**Scenario:** Actualizar un recurso

**Given** el usuario está en la página de búsqueda y desea actualizar la información de una persona

**When** el usuario actualiza el recurso que desea y obtiene una respuesta

**Then** el usuario podrá ver la confirmación de la actualización

**Scenario:** Borrar un registro

**Given** el usuario está en la página y desea borrar un registro

**When** el usuario escribe la instrucción de borrar el registro

Then el usuario obtiene una la respuesta de borrado

A continuación, se enuncian los diseños de casos de prueba HU003 y HU004:

**Feature:** Actualizar y Obtener información en la página jsonplaceholder

Como usuario registrado del sistema

quiero actualizar y ver mi información

para poder corroborarla

**Scenario:** Actualizar información

**Given** el usuario se encuentra en la página y desea actualizar su información

**When** el usuario utiliza la función put

**Then** el usuario verá que su información fue actualizada

**Scenario:** Obtener información

**Given** el usuario está en la página y desea ver su información

**When** el usuario utiliza la función get

**Then** el usuario vera que su información

# 6. Ambiente y Herramientas de Pruebas

## 6.1 Herramientas de Pruebas

|  |  |
| --- | --- |
| **Herramienta** | **Función** |
| Serenity BDD | Es una librería de código abierto que ayuda a escribir pruebas de aceptación automatizadas de mayor calidad y más rápido |
| JUnit testing framework | Ejecución y Reporte de las pruebas |
| Gradle | Creación de la estructura de proyectos y uso e importación de librerías |
| Rest | REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP (uno de los protocolos más antiguos) y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON. |
| Cucumber | Gestor de features en leguaje Gherkin |

## 6.2 Ambiente de Pruebas

|  |  |
| --- | --- |
| Navegadores | Chrome |
| Sistemas Operativos | Windows |

# 7. Criterios de Entrada y Salida

## 7.1 Criterios de Entrada

* Las funcionalidades deben estar desplegadas en el ambiente de QA y haber sido probadas manualmente.
* El framework de pruebas está instalado y listo para la ejecución
* El ambiente de QA está disponible.

## 7.2 Criterios de Salida

* Ejecución de todos los casos de pruebas automatizados
* Se ha logrado la suficiente cobertura de los requerimientos y funcionalidades bajo pruebas
* Ningún defecto de severidad alta se encuentra abierto.

# 8. Planificación de ejecución de las pruebas

Lista de funcionalidades a ser automatizadas por Sprint

|  |  |  |
| --- | --- | --- |
| Sprint | Funcionalidades | Comentarios |
| 1 | Probar los metodos Put y Delete con el recurso api/users/2. | La validación se realiza con el nombre de la persona. |
| 2 | Probar los metodos Put y Get con el recurso /posts/1. | La validación se realiza con el nombre del título. |

Es necesario que las funcionalidades a automatizar se desarrollen, implementen y prueben manualmente para que tengan un nivel determinado de estabilidad cuando comienzan las tareas de automatización.

# 9. Reporte de Pruebas

El Reporte automático de pruebas se obtendrá a través de SerenityBDD. Este Reporte informará sobre los resultados de la ejecución de la automatización. Incluirá las pruebas que pasaron y las que fallaron, los errores encontrados, la tasa de éxito y el tiempo transcurrido.





