Proyecto: Servicios Rest (Post - Put)
https://reqres.in/
https://jsonplaceholder.typicode.com/

Estrategia de Prueba Automatizadas

Historia de revisiones

Versión	Autor	Descripción	Fecha
1.0	Juan Esteban Pineda Angel	Creación del documento	Enero 2022

Tabla de Contenidos

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, los riesgos, planes de contingencia y el calendario de ejecución de las pruebas de servicios Rest Post y Put en la página Reqres.in y en la página JSON placeholder que en el documento llamaremos “JSON”.

Estos servicios inicialmente son evaluados en POSTMAN con el fin de explorar su comportamiento ante diferentes estímulos y conocer sus códigos de respuesta y cuerpos de respuesta.

Se plantea la selección de dos escenarios por cada petición exitosos.

2. Alcance

Se realizarán pruebas de caja negra (automatizadas) a las funcionalidades seleccionadas durante el primer sprint el cual tiene una duración de una semana .
Para probar la historia de usuario se plantea en lenguaje Gherkin

Primera historia de usuario

Feature: crear un nuevo registro
como un usuario de la página quiero crear un nuevo registro asignando un nombre de usuario y un trabajo,

Scenario: Creación de registro Reqres
Given el usuario está en la página Reqres ingresa el nombre "morpheus" y el campo trabajo "leader"
When cuando el usuario hace una petición de creación de registro
Then el usuario deberá ver un codigo de respuesta 201 y los datos creados

Scenario: Creación de registro Json
Given el usuario está en la página de creación de Json ingresa el nombre "Juanes"
When cuando el usuario hace una petición de creación post
Then el usuario deberá ver un código 201 y el dato creado

Segunda historia de usuario

Feature: actualizar datos de registro
como un usuario registrado del sistema
necesito actualizar el campo trabajo asignado a mi nombre
@actualizarReqres
Scenario: Registro actualizado completamente
Given el usuario está en la página de actualización de reqres con usuario "morpheus" y trabajo "zion resident"
When cuando el usuario hace una petición de actualizacion put
Then el usuario deberá obtener un codigo de respuesta 200 y los datos actualizados

@actualizarJson

Scenario: Registro actualizado parcialmente

Given el usuario está en la página de actualización de Json con el título "dale a tu cuerpo alegría macarena"

When cuando el usuario hace una petición de actualización

Then el usuario deberá ver un código de respuesta 200 y los datos ingresados

3. Roles y Responsabilidades

Roles	Responsabilidades
Manager de QA	Planificación y monitoreo de las pruebas automatizadas Reporte de Defectos Reporte de progreso de las pruebas
Ingeniero QA de Automatización/ Analista QA	Diseño e implementación de las pruebas. Ejecución de las pruebas automatizadas. Reporte de resultados de las pruebas.
Product Owner/Stakeholders	Toma de decisiones

4. Riesgos y Planes de Contingencia

S001	Historia de usuario 1	Probabilidad de Ocurrencia (1-5)	Impacto (1-5)	Riesgo
1	Verificar la disponibilidad del servicio POST Reqres.	2	3	6
2	Verificar que se pueda ingresar el código correcto.	3	1	3
3	Verificar la petición	2	2	4
4	Verificar el código de respuesta “201”	1	4	4
5	Verificar el resultado válido.	2	4	4

Tabla 1 Análisis de riesgo Gherkin 1

S002	Historia de usuario 1	Probabilidad de Ocurrencia (1-5)	Impacto (1-5)	Riesgo
1	Verificar la disponibilidad del servicio POST JSON.	2	3	6
2	Verificar que se pueda ingresar el código incorrecto.	3	1	3
3	Verificar la petición	2	2	4
4	Verificar el código de respuesta “201”	1	4	4
5	Verificar el mensaje de resultado no válido.	2	4	4

Tabla 2 Análisis de riesgo Gherkin 1

S001	Historia de usuario 1	Probabilidad de Ocurrencia (1-5)	Impacto (1-5)	Riesgo
1	Verificar la disponibilidad del servicio PUT Reqres.	2	3	6
2	Verificar que se pueda ingresar el código país correcto.	3	1	3
3	Verificar la petición	2	2	4
4	Verificar el código de respuesta “200”	1	4	4
5	Verificar el resultado válido.	2	4	4

Tabla 1 Análisis de riesgo Gherkin 2

S002	Historia de usuario 1	Probabilidad de Ocurrencia (1-5)	Impacto (1-5)	Riesgo
1	Verificar la disponibilidad del servicio PUT JSON	2	3	6
2	Verificar que se pueda ingresar el código incorrecto.	3	1	3
3	Verificar la petición	2	2	4
4	Verificar el código de respuesta “200”	1	4	4
5	Verificar el mensaje de resultado no válido.	2	4	4

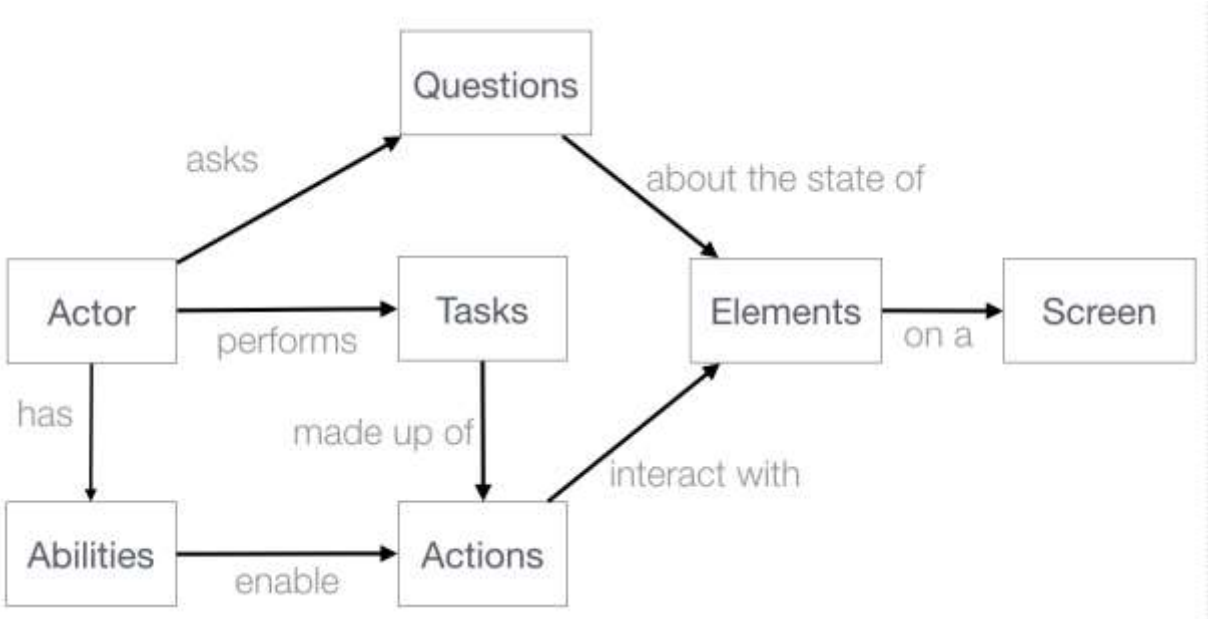
Tabla 2 Análisis de riesgo Gherkin 2

5. Ambiente y Herramientas de Pruebas

5.1 Herramientas de Pruebas

Herramienta	Function
Serenity BDD	API para automatizar peticiones de servicios con patrón ScreenPlay
JUnit testing framework	Ejecución y Reporte de las pruebas
Gradle	Creación de la estructura de proyectos y uso e importación de librerías
Hamcrest	Comparaciones personalizadas
Cucumber	Gestor de Historias de usuario en formato Gherkin
LOG4J	Impresión de mensajes por consola

5.2 Arquitectura del patrón de diseño.



Arquitectura ScreenPlay

Para hacer la verificación de servicios Rest se emplea como cliente la aplicación creada en Intellij en lenguaje Java, la cual emplea Serenity BDD para la gestión de las peticiones en post de ambos features de Gherkin.

5.3 Ambiente de Pruebas

Resultados	Consola de Intellij
Sistemas Operativos	Windows

6. Criterios de Entrada y Salida

6.1 Criterios de Entrada

Los servicios deben de estar desplegados en el dominio de la URL y se debe conocer los mensajes esperados por las peticiones, así como sus códigos de respuesta esperados.

El framework de pruebas está instalado y listo para la ejecución

El ambiente de QA está disponible.

Los defectos críticos encontrados durante las pruebas manuales han sido resueltos y cerrados.

6.2 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.

7. Planificación de ejecución de las pruebas

Lista de funcionalidades a ser automatizadas por Sprint

Sprint 1	Funcionalidades	Comentarios
1 POST	Post para la creación de registros	
2 PUT	Put para la Actualización de registros	

Las pruebas de automatización normalmente comenzarán finalizando la primera semana del Sprint.

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.

Planificación de Pruebas Automatizadas

7.1 Planificación de las Pruebas de Regresión

Las suites de regresión se ejecutarán al final de cada Sprint (antes de la Revisión del Sprint), al realizarse un cambio o por solicitud de los Clientes, Product Owner y Project Manager.

8. Reporte de Pruebas

El Reporte automático de pruebas se obtendrá a través de Serenity, este Reporte informará sobre los resultados de la ejecución de cada caso de prueba. Incluirá las pruebas que pasaron y las que fallaron, los errores encontrados, la tasa de éxito y el tiempo transcurrido (documentación viva).