



Proyecto: Website zonafit.com.co

Plan de pruebas de automatización

Historia de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Nilton Rodríguez	Creación del documento	2 enero de 2022

Tabla de contenido

1. Introducción	4
2. Alcance	5
2.1. Casos de prueba.....	5
3. Roles y Responsabilidades.....	6
4. Riesgos y Planes de contingencia	7
5. Ambiente y Herramientas de prueba.	8
5.1. Herramientas de prueba	8
5.2. Arquitectura del framework de automatización	8
5.3. Ambiente de pruebas	8
6. Criterios de entrada y salida.....	9
6.1. Criterios de entrada	9
6.2. Criterios de salida	9
7. Planificación de la ejecución de las pruebas.....	10
8. Reporte de pruebas	11

1. Introducción

Regres es una API que provee servicios y herramientas de respuesta REST. El alcance para el actual plan de pruebas son los servicios CREATE y SINGLE USER. Las pruebas se desarrollarán sobre el ambiente de producción en sistemas operativos Windows y MacOS. Para la automatización de las pruebas desarrollará con Java, usando la librería REST Assured y Cucumber. Entre los riesgos se encuentra que los servicios no estén implementados y que los servicios no sean accesibles. Ante dichos riesgos no se tiene plan de contingencia, puesto que las pruebas dependen directamente de la implementación de las funcionalidades.

2. Alcance

Se realizarán pruebas de caja negra (automatizadas) a los servicios CREATE y SINGLE USER.

2.1. Casos de prueba

Feature: User register.

AS

user of the system

I WANT TO

register in the system

SO THAT

I can use the system services.

Scenario: Register Successful

Given the user is in the website registration page with the email "eve.holt@reqres.in" and the password "pistol"

When the user send a registration request

Then the user sees a success response code and an id with a response token

Feature: Single user listing

AS

administrator of the system

I WANT TO

list an user information

SO THAT

I can validate the user information

Scenario: List a single user

Given an administrator of the website that wants to list a single user information

When the administrator sends the list request

Then the administrator see a success response code and the user information

3. Roles y Responsabilidades

ROLES	RESPONSABILIDADES
Manger de QA	Planificación y monitoreo de las pruebas automatizadas. Reporte de defectos
Analista QA	Diseño e implementación de las pruebas. Ejecución de las pruebas automatizadas. Reporte de resultados de las pruebas.
Product Owner / Stakeholders	Toma de Desiciones.

4. Riesgos y Planes de contingencia

No	Riesgos	Probabilidad de ocurrencia (1-4)	Impacto (1-4)	Severidad	Plan de contingencia
1	Servicios no implementados.	1	4	4	
2	Servicios no disponibles	1	4	4	

5. Ambiente y Herramientas de prueba.

5.1. Herramientas de prueba

Herramienta	Función
REST Assured	API para automatizar servicios REST.
Cucumber	Herramienta de BDD para desarrollar casos de prueba y el reporte de las pruebas.
Gradle	Creación de la estructura de proyectos y uso e importación de librerías.
Chromedriver	Crea una instancia del navegador Google Chrome.

5.2. Arquitectura del framework de automatización

REST Assured es una librería de Java que simplifica los test de servicios basados en REST y contruídos con HTTP Builder. Soporta las peticiones POST, GET, DELET, PUT, OPTIONS, PATCH y HEAD y puede ser usado para validar y verificar la respuesta de esas peticiones.

5.3. Ambiente de pruebas

Sistemas Operativos	Windows, MacOS.
---------------------	-----------------

6. Criterios de entrada y salida

6.1. Criterios de entrada

Las funcionalidades están desplegadas y han sido probadas manualmente.

El framework se encuentra instalado y listo para la operación.

Los servicios se encuentran disponibles.

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

6.2. Criterios de salida

Ejecución de todos los casos de prueba automatizados.

Se logra suficiente cobertura de los requerimientos y funcionalidades bajo las pruebas.

Ningún defecto de severidad alta se encuentra abierto.

7. Planificación de la ejecución de las pruebas

Lista de funcionalidades a ser automatizadas.

No.	Funcionalidad	Comentarios
1.1	CREATE	Se realiza la petición POST con los datos suministrados.
2.1	SINGLE USER	Se realiza la petición GET para verificar los datos suministrados.




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.

Se requiere que los servicios POST para CREATE y GET para SINGLE USER estén implementados y disponibles.

8. Reporte de pruebas

El reporte automático de las pruebas se obtendrá a través de Cucumber y será publicado en un link generado directamente por esta herramienta. El reporte informará sobre los resultados de la ejecución de cada caso de prueba, incluirá las pruebas que pasaron y las pruebas que fallaron, la tasa de éxito y el tiempo transcurrido.

1 PASSED

100% passed 1 executed	a few seconds ago Last Run	7.97 seconds Duration
 Mac OS X	 Java HotSpot(TM) 64-Bit Server VM 25.111-b14	 cucumber-jvm 6.11.0

Delete Report

✓ file:///Users/niltonrodriguez/Documents/Programming/SofkaU/Introduccio%CC%81n_a_la_automatizacio%CC%81n/RetoRestAssured/src/test/resources/features/registerSuccessful.feature

Report

Edit

Feature: User register.
AS user of the system I WANT TO register in the system SO THAT I can use the system services.
Scenario: Register Successful
✓ **Given** the user is in the website registration page with the email "eve.holt@regres.in" and the password "pistol"
✓ **When** the user send a registration request
✓ **Then** the user sees a success response code and an id with a response token