Proyecto: Servicios SOAP

Plan de pruebas de automatización

Historia de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Nilton Rodríguez	Creación del documento	7 enero de 2022

Tabla de contenido

1. Introducción

Representational State Transfet (REST) es un estilo arquitectónico para proporcionar estándares entre sistemas informáticos en la web, lo que facilita que los sistemas se comuniquen entre sí. El alcance para el actual plan de pruebas son los servicios Register y Single User de Reqres.in, una API que provee servicios y herramientas de respuesta REST. Las pruebas se desarrollarán en sistemas operativos Windows y MacOS. La automatización de las pruebas se desarrollará con Java, usando la librería Serenity BDD, el patrón Screenplay y Cucumber. Entre los riesgos se encuentra la disponibilidad de los servicios. Ante dicho riesgo no se tiene plan de contingencia, puesto que las pruebas dependen directamente de la implementación de los servicios.

2. Alcance

Se realizarán pruebas de caja negra (automatizadas) a los servicios POST Register – Successful y GET Single User. En cada uno de los servicios se probarán escenarios exitosos y la respuesta ante peticiones no soportadas.

2.1. Casos de prueba

2.1.1. Register Successful

```
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 allowed user is in the website registration page
    When the user send a registration request with the email
    "eve.holt@reqres.in" and the password "pistol"
    Then the user sees a success response code and an id with a response
    token

  Scenario: Not allowed register
    Given a not allowed user of the website attempts to register
    When the user attempt a registration request with the email
    "snow.flake@reqres.in" and the password "snowflake"
    Then the user sees a bad request response code and an error message
```

2.1.2. Single User

```
Feature: Single user listing
  AS
    administrator of the system
  I WANT TO
    list an user information
  SO THAT
    I can validate the user information

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

  Scenario: List a single user
    When the administrator sends the list request for user "2"
    Then the administrator see a success response code and the user
    information

  Scenario: List a non existent user
    When the administrator sends the list request for the non existent user
    "13"
    Then the administrator see a Not Found response code
```

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 Decisiones.

4. Riesgos y Planes de contingencia

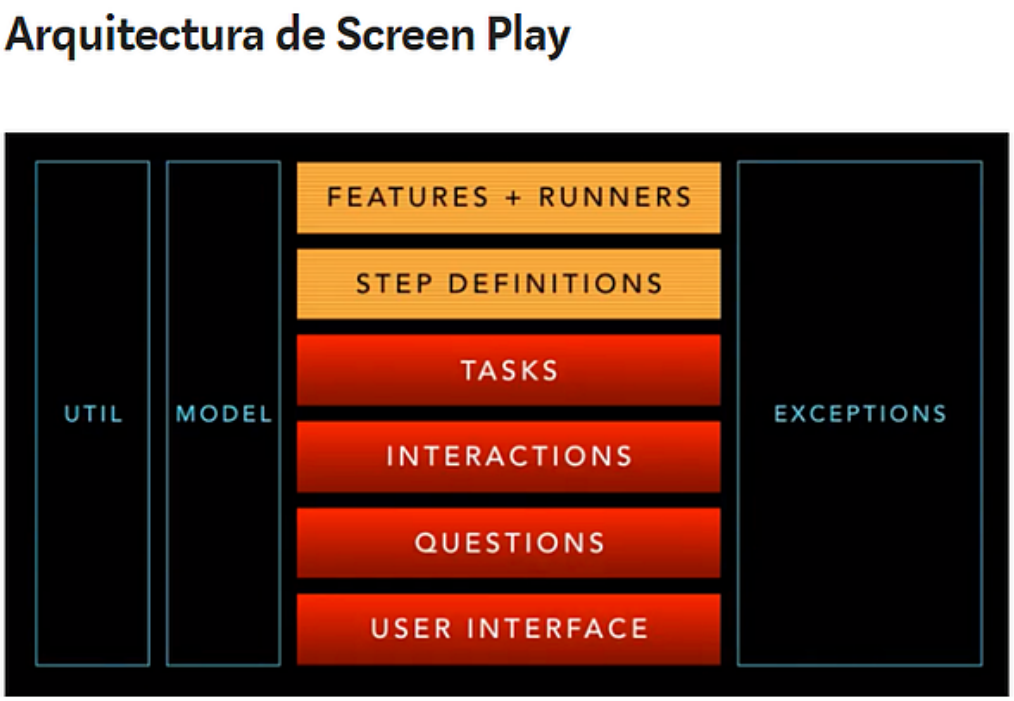
No	Riesgos	Probabilidad de ocurrencia (1-4)	Impacto (1-4)	Severidad	Plan de contingencia
1	Los servicios no se encuentran disponibles.	1	4	4	

5. Ambiente y Herramientas de prueba.

5.1. Herramientas de prueba

Herramienta	Función
Serenity BDD	Librería para escribir pruebas automatizadas y producir documentación viva.
Screenplay	Patrón para escribir pruebas de aceptación automatizadas.
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.

5.2. Arquitectura del framework de automatización



Serenity BDD es una librería open source que ayuda a construir pruebas automatizadas limpias y fáciles de mantener sin necesidad de construir un framweork propio. Serenity utiliza los resultados de las pruebas para generar informes que documentan la aplicación y su funcionamiento. Esta librería proporciona soporte para pruebas web con Selenium, pruebas API REST con RESTAssured, pruebas SOAP que son fáciles de escalar y mantener gracias al patrón Screenplay.

El patrón Screenplay se enfoca en escribir pruebas de aceptación automatizadas basadas en los principios SOLID, fomentando buenas prácticas de pruebas y conjuntos de pruebas fáciles de leer y mantener. Este patrón se centra en escribir pruebas como si se tratara de un guion de una película o una obra de teatro, donde *actores* hacen *preguntas*, ejecutan *tareas* y tienen *habilidades* que les permiten ejecutar *acciones* en su interacción con las páginas web.

Gracias a su integración con Behaviour Driven Development (BDD) Serenity se integra con otras librerías y frameworks como Selenium y Cucumber, lo cual permite que el código sea fácil de leer y pueda aplicar lenguajes como Gherkin de manera prácticamente nativa.

5.3. Ambiente de pruebas

Sistemas Operativos	Windows, MacOS.

6. Criterios de entrada y salida

6.1. Criterios de entrada

Las funcionalidades e y las funcionalidades han sido probadas manualmente.

El framework se encuentra instalado y listo para la operación.

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	Register Successful	A demás del escenario exitoso, se realizarán pruebas para comprobar el comportamiento del servicio ante peticiones no soportadas.
2	Single User	A demás del escenario exitoso, se realizarán pruebas para comprobar el comportamiento del servicio ante peticiones no soportadas.

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 realizan pruebas con peticiones no soportadas para comprobar el comportamiento de los servicios y los códigos de respuesta que generan dichas peticiones.

8. Reporte de pruebas

El reporte automático de las pruebas se obtendrá a través de SerenityBDD 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.