

Proyecto: Servicio Rest JSONPLACEHOLDER

Estrategia de Prueba Automatizadas
Con Screenplay

Historia de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Cristian Ayala	Creación del documento	Octubre 2021

Tabla de Contenidos

[1.Introducción](#)

[2. Alcance](#)

[3. Roles y Responsabilidades](#)

[5. Ambiente y Herramientas de Pruebas](#)

[5.1 Herramientas de Pruebas](#)

[5.2 Arquitectura del framework de automatización](#)

[5.3 Ambiente de Pruebas](#)

[6. Criterios de Entrada y Salida](#)

[6.1 Criterios de Entrada](#)

[6.2 Criterios de Salida](#)

[7. Planificación de ejecución de las pruebas](#)

[7.1 Planificación de las Pruebas de Regresión](#)

[8. Reporte de Pruebas](#)

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 del proyecto JSONPLACEHOLDER.

2. Alcance

Se realizarán pruebas de caja negra (automatizadas) a las funcionalidades asignadas bajo los escenarios descritos en el ámbito de este reto.

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

No	Riesgos	Probabilidad de Ocurrencia (1-5)	Impacto (1-5)	Severidad (Prob*Impacto)	Plan de Contingencia
1	Funcionalidades no terminadas en el tiempo estimado no pueden formar parte de las funcionalidades planificadas para ser automatizadas en el sprint actual	2	5	10	Re planificar las funcionalidades para ser automatizadas
2	Solicitud de cambios en aquellas funcionalidades que ya tienen casos de pruebas automatizadas. Esto ocasiona re trabajo debido a que se deben actualizar estos scripts.	3	3	9	Estimar el tiempo del cambio y volver a priorizar la lista de funcionalidades a ser automatizadas en el sprint.

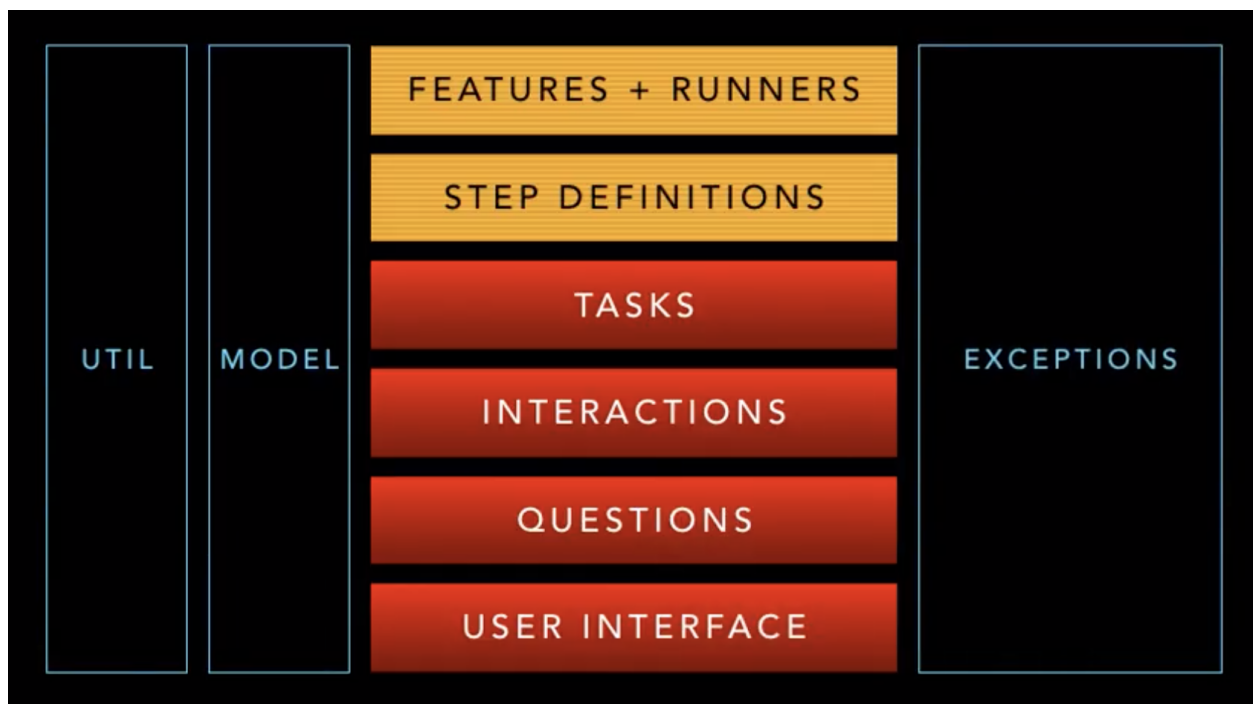
5. Ambiente y Herramientas de Pruebas

5.1 Herramientas de Pruebas

Herramienta	Función
Serenity BDD	Libreria para crear pruebas de aceptacion automatizadas
Selenium WebDriver	API para automatizar sistemas Web
Cucumber	Ejecución y Reporte de las pruebas
Gradle	Creación de la estructura de proyectos y uso e importación de librerías
Chromedriver	Crea una instancia del navegador Chrome

5.2 Arquitectura del framework de automatización

Utilizaremos el patrón Screenplay para la automatización de las pruebas, siguiendo la siguiente arquitectura:



5.3 Ambiente de Pruebas

Navegadores	Chrome
Sistemas Operativos	Windows

6. Criterios de Entrada y Salida

6.1 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.

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 en este único sprint:

Funcionalidades	Escenarios a automatizar	Historia de Usuario
Filtrar Comments	HU-001-CA-001 Consultar por campo existente	HU-001

	HU-001-CA-002 Consultar por campo inexistente	
--	--	--

Las pruebas de automatización se realizan para este primer sprint partiendo de la base de que se entregó el aplicativo con las funcionalidades a probar y se realizaron pruebas manuales sobre el mismo.

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 los escenarios de prueba para cada feature. Incluirá las pruebas que pasaron y las que fallaron, los errores encontrados, la tasa de éxito y el tiempo transcurrido.

9. Anexos

HU-001: Filtrar comments

Como usuario de la aplicacion

Necesito poder filtrar los informacion de los comments por campos para poder visualizarlos mejor.

HU-001-CA-001

Dado el usuario decidio consultar un comment por el campo ("id") con valor (5)

Cuando el usuario realiza la peticion

Entonces deberia obtener resultado exitoso con la informacion relativa al comment solicitado

HU-001-CA-002

Dado el usuario decidio consultar un comment por un campo inexistente ("noExiste") con valor (5000)

Cuando el usuario realiza la peticion

Entonces deberia obtener resultado de error