****

**Proyecto: Posts – jsonplaceholder.typicode.com**

**Plan de pruebas – REST API**

**Historial de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Autor(es)** | **Descripción** | **Fecha** |
| 1.0 | Antonio Coque | Creación del documento | 22 de Octubre 2021 |

Tabla de Contenido

[1. Introducción 4](#_Toc85041400)

[2. Alcance 5](#_Toc85041401)

[3. Roles y Responsabilidades 6](#_Toc85041402)

[4. Riesgos y Planes de Contingencia 7](#_Toc85041403)

[5. Ambiente y Herramientas de Pruebas 8](#_Toc85041404)

[5.1. Herramientas de Pruebas 8](#_Toc85041405)

[5.2. Arquitectura del framework de automatización 8](#_Toc85041406)

[5.3. Ambiente de Pruebas 9](#_Toc85041407)

[6. Criterios de Entrada y Salida 10](#_Toc85041408)

[6.1. Criterios de Entrada 10](#_Toc85041409)

[6.2. Criterios de Salida 10](#_Toc85041410)

[7. Planificación de ejecución de las pruebas 11](#_Toc85041411)

[7.1. Planificación de las pruebas de Regresión 12](#_Toc85041412)

[8. Reporte de Pruebas 13](#_Toc85041413)

1. Introducción

En el presente documento se encuentra la estrategia a implementar para la realización de las pruebas automatizadas del servicio **REST API** de Posts en la página **jsonplaceholder.typicode.com.** Las pruebas automatizadas se implementarán con la arquitectura **Screenplay** y las librerías de **Serenity BDD** para este servicio disponible con **REST API**, adicionalmente se describe el alcance de las pruebas, el ambiente de pruebas, los recursos necesarios, las herramientas a utilizar, los riesgos, los planes de contingencia en caso de que se materialice un riesgo y el calendario de ejecución de las pruebas automatizadas en el proyecto. En la parte final del documento se muestran los reportes generados durante la ejecución de las pruebas automatizadas.

1. Alcance

Se realizarán pruebas de caja negra (automatizadas) a las funcionalidades seleccionadas durante la planificación del sprint en el aplicativo web para el servicio de **Posts**.

Las funcionalidades a ser automatizadas serán seleccionadas utilizando los criterios de la lista de chequeo “Que casos de pruebas automatizar”, donde se valorara el impacto, la probabilidad de ocurrencia y la severidad del riesgo.

Para este Sprint solo se evaluará el servicio de **Posts**, bajo 2 escenarios de prueba. Para este desarrollo se hará uso de la arquitectura Screenplay y con Serenity BDD.

1. Roles y Responsabilidades

A continuación, se describen los roles y las responsabilidades de los miembros del equipo involucrados en el proceso de pruebas:

|  |  |
| --- | --- |
| **Roles** | **Responsabilidades** |
| Manager de QA | * Planificación y monitorio de las pruebas automatización * 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 |

1. Riesgos y Planes de Contingencia

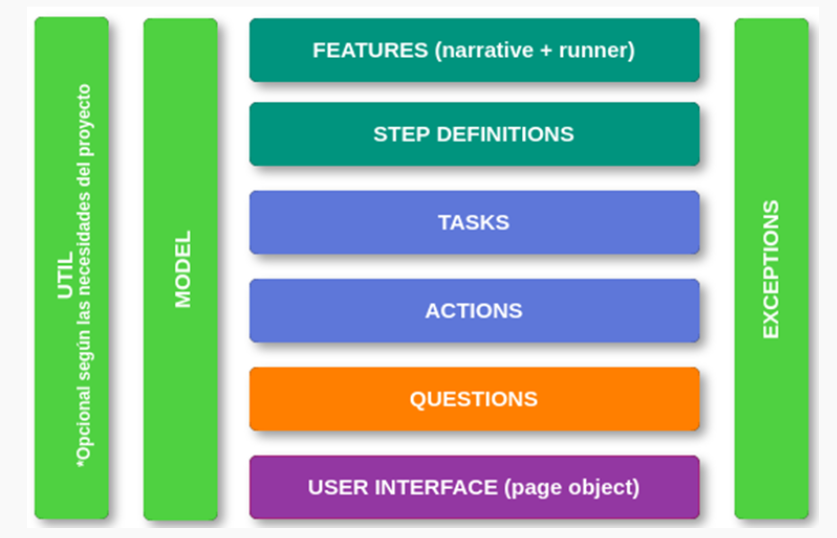
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N°** | **Riesgos** | **Probabilidad de Ocurrencia (1-5)** | **Impacto**  **(1-5)** | **Severidad (Probabilidad**  **\*Impacto)** | **Plan de Contingencia** |
| 1 | El sistema no permite la consulta de Posts existentes | 1 | 5 | 5 | Detectar la posible causa del problema, reportarlo y priorizarlo. Socializar y retroalimentar con el equipo de desarrollo. |
| 2 | El sistema no permite consultar los comentarios sobre los Posts existentes | 1 | 5 | 5 |

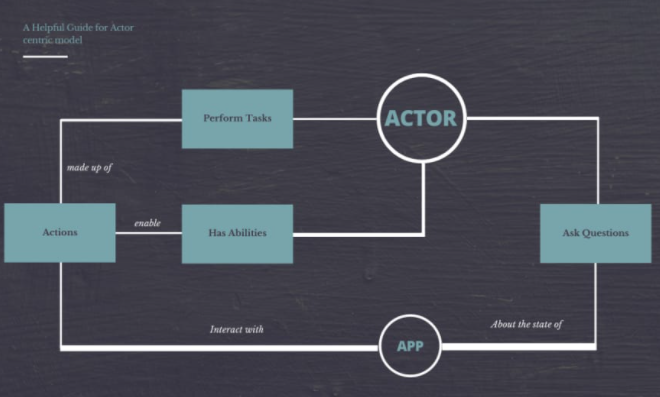
1. Ambiente y Herramientas de Pruebas
   1. Herramientas de Pruebas

|  |  |
| --- | --- |
| **Herramienta** | **Función** |
| JUnit testing framewrok | Ejecución y Reporte de las pruebas |
| Gradle | Creación de la estructura de proyectos y uso e importación de librerías |
| Cucumber for Java | Plugin que permite implementar la metodología BDD y la creación de features en el lenguaje java. |
| Log4j2 | Librería que permite escribir mensajes de registro y llevar el control de los mismos. |
| Serenity BDD | Librería que ayuda a escribir pruebas de aceptación automatizadas de mayor calidad y más rápido |
| POSTMAN | Aplicación que permite realizar pruebas API, dando la posibilidad de testear 'HTTP requests' a través de una interfaz gráfica de usuario, por medio de la cual se obtiene diferentes tipos de respuesta para posteriormente ser validadas |

* 1. Arquitectura del framework de automatización

**Arquitectura de Screenplay**





Se implementará el patrón Screenplay para escribir las pruebas automatizadas basadas en los principios de ingeniería de software tal como son el principio de responsabilidad individual (SRP) y en el principio de abrir y cerrar (Open-Closed).

Este patrón promueve los buenos hábitos de testing y suites de pruebas bien organizadas las cuales son fácil de seguir, fácil de mantener y fácil de extender; habilitando equipos a escribir más robustos y entendibles pruebas automatizadas.

En el patrón de Screenplay las pruebas describen como un usuario puede interactuar con la aplicación para alcanzar una meta. En este el usuario que interactúa con el sistema es llamado "Actor". Los Actores son la parte central del patrón Screenplay. Como Screenplay (Guion) los actores tienes uno o más "Habilidades", como son la habilidad de navegar por la web o realizar una consulta un web service, el "Actor" también puede realizar "Tareas", como pueden ser agregar o remover objetos o cosas.

Para interactuar con la aplicación, como puede ser introducir valores en los campos o dar click en botones, los actores necesitan estas interacciones, las cuales son denominadas "Acciones". Los "Actores" pueden hacer "Preguntas" acerca del estado del sistema, como por ejemplo leer un valor de un campo en la pantalla o realizando una consulta a algún web service, de tal forma es como se prueba usando el patrón de Screenplay.

* 1. Ambiente de Pruebas

|  |  |
| --- | --- |
| **Navegadores** | **Sistemas Operativos** |
| Google Chrome | Windows 10 |

1. Criterios de Entrada y Salida
   1. Criterios de Entrada

* Las funcionalidades deben estar desplegadas en el ambiente de QA (Administración de grados de pago).
* Las funcionalidades han sido probadas manualmente (Todos los escenarios de prueba seleccionados en el módulo).
* El framework de pruebas está instalado y listo para la ejecución.
* Todas las herramientas están debidamente configuradas y funcionales.
* El ambiente de QA está disponible.
* Los defectos críticos encontrados durante las pruebas manuales han sido resueltos y cerrados.
  1. Criterios de Salida
* Automatización de los escenarios de prueba.
* Ejecución de todos los casos de pruebas automatizados.
* Se ha logrado la suficiente cobertura de los requerimientos y funcionalidades bajo pruebas para el sprint.
* Ningún defecto de severidad alta de los evaluados para este sprint se encuentra abierto.
* Los reportes de las pruebas se encuentran debidamente documentados.

1. Planificación de ejecución de las pruebas

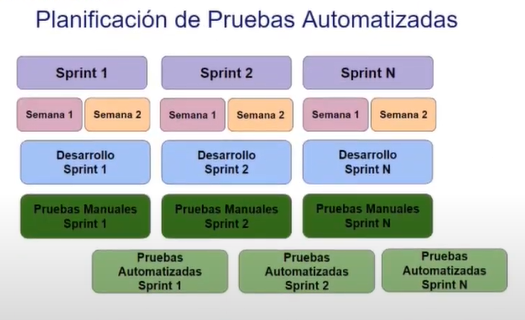
Lista de funcionalidades a ser automatizadas por Sprint

|  |  |  |
| --- | --- | --- |
| **Sprint numero** | **Funcionalidades** | **Comentarios** |
| 1 | Consulta de Posts existentes por id | Se debe verificar que se brinde información sobre el post existente y que corresponda al mismo. |
| Consultar de comentarios sobre los Posts existentes | Se debe permitir la consulta de los comentarios que se hayan realizado sobre un post |

Las pruebas de automatización comenzaran en la segunda semana del Sprint (de 2 semanas)

Las funcionalidades a automatizar se desarrollarán, implementaran y probaran manualmente para que tengan un nivel determinado de estabilidad cuando se comiencen con las tareas de automatización.

Una vez automatizadas las pruebas se ejecutarán y verificaran, para garantizar que no existen falsos positivos en los reportes generados.



Para el servicio de **Posts** seleccionado para este Sprint se ejecutarán los siguientes escenarios**:**

**Posts:** Consultas sobre Posts

**Escenario 1:**

* Se permite consulta de Posts por id

**Escenario 2:**

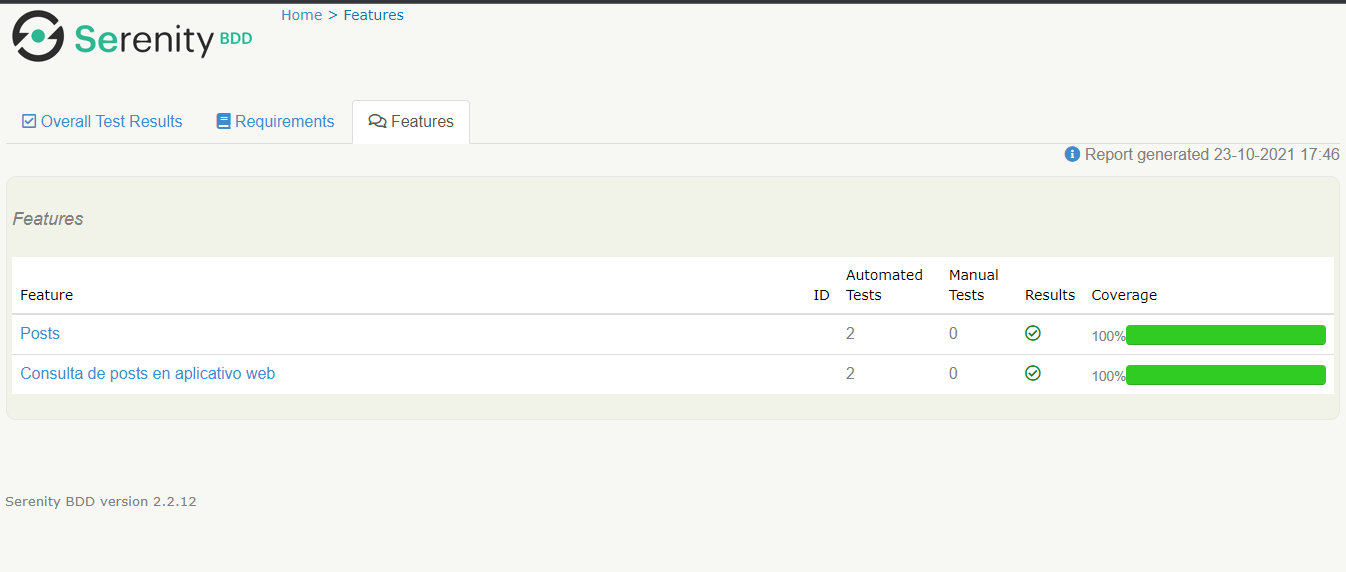
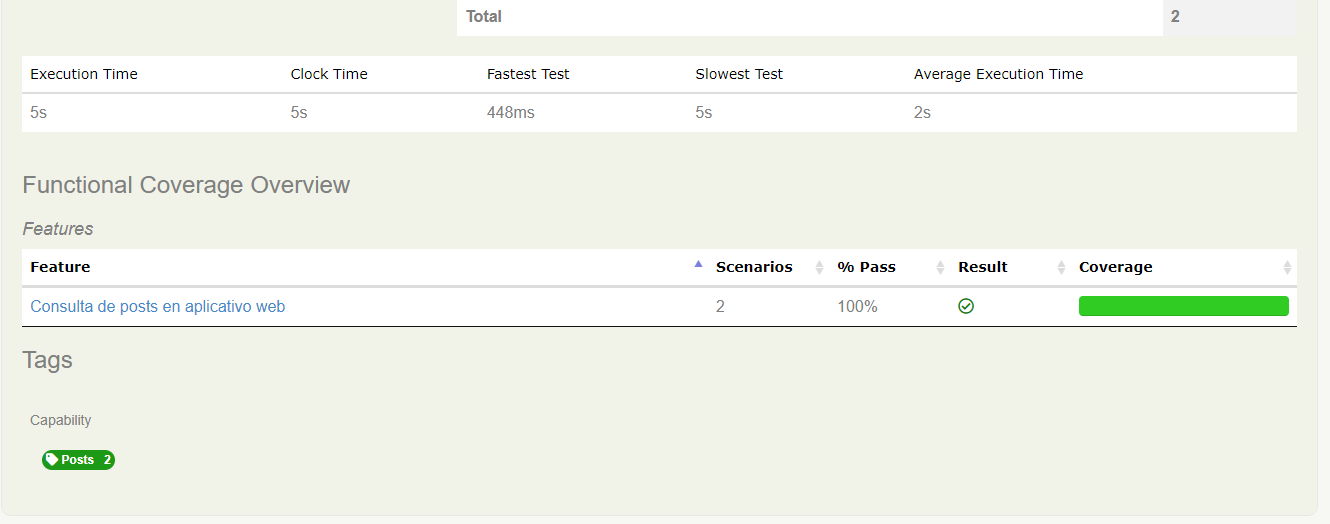
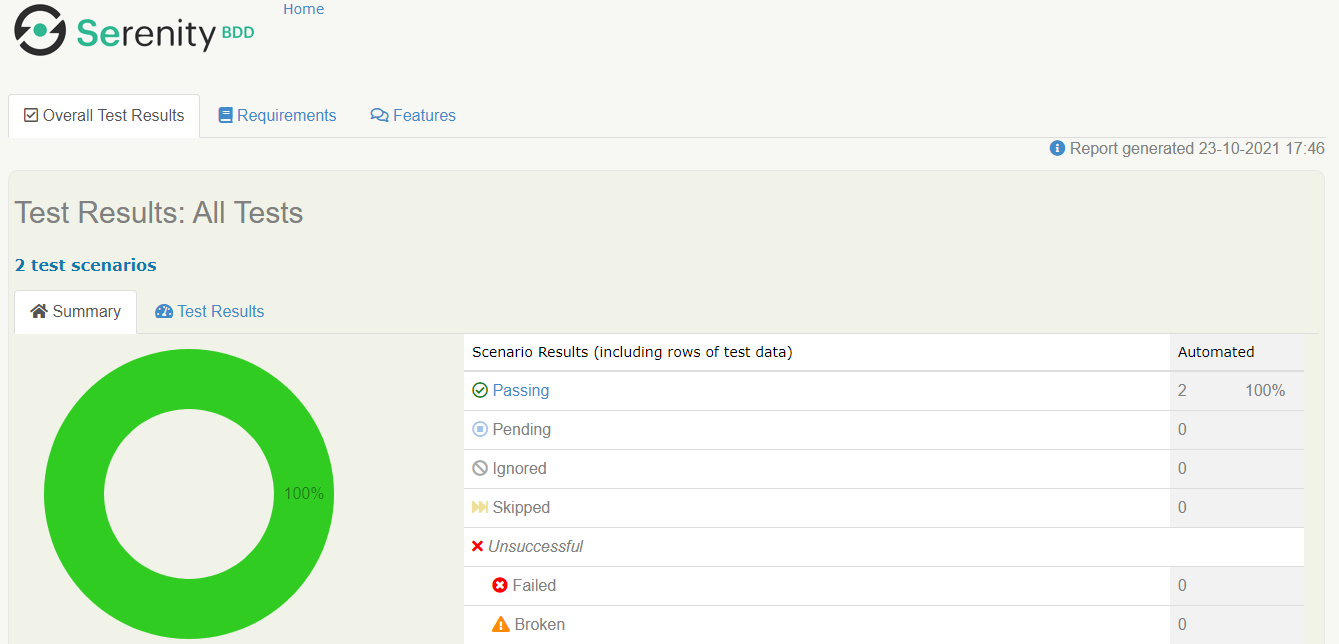
* Se permite la consulta de los comentarios de un Post.
  1. Planificación de las pruebas de Regresión

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.

1. Reporte de Pruebas

El reporte automático de las pruebas se obtendrá a través de **Serenity BDD** reports. Este reporte informara 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, el tiempo transcurrido y los escenarios testeados.

* Reporte de las pruebas automatizadas realizadas al servicio **Posts** REST API



Como se puede observar en los reportes las pruebas han resultado exitosas, para estos escenarios de prueba en un sistema operativo y en un navegador. Es recomendable realizar las pruebas en diferentes dispositivos, SO y navegadores, para garantizar que en todos los escenarios posibles el resultado es satisfactorio.