

**Plan de pruebas**

Proyecto: API RestFul Booking

Versión: 1.0

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | Steven Duque | Creación del documento | Diciembre 2022 |

Contenido

[Propósito del documento 2](#_Toc123046322)

[Contexto inicial del proyecto 2](#_Toc123046323)

[Documentación recibida para las pruebas 3](#_Toc123046324)

[Alcance 3](#_Toc123046325)

[Fuera del alcance 3](#_Toc123046326)

[Supuestos 4](#_Toc123046327)

[ El equipo de QA proporcionará el set de datos para la ejecución de los diferentes casos de prueba. 4](#_Toc123046328)

[ Establecimiento del entorno de pruebas. 4](#_Toc123046329)

[Estrategia de prueba 4](#_Toc123046330)

[Recurso Humano 4](#_Toc123046331)

[Ambientes / Entornos de prueba 5](#_Toc123046332)

[Insumos necesarios para las pruebas 5](#_Toc123046333)

[Entregables 5](#_Toc123046334)

[Criterios de Entrada y de Salida 5](#_Toc123046335)

[Criterios de Entrada: 6](#_Toc123046336)

[Criterios de Salida: 6](#_Toc123046337)

[Criterios de finalización 6](#_Toc123046338)

[Criterios de suspensión 6](#_Toc123046339)

[Criterios de reanudación 6](#_Toc123046340)

# 

# Propósito del documento

El propósito de este documento es brindar la información y las directrices requeridas para planificar y desarrollar las actividades del proceso de pruebas de la API RestFul Booking.

# Contexto inicial del proyecto

Demo Web Shop es una API diseñada para fines educativos y prácticos que simula una tienda online de libros, el cual permite al usuario poder usarla como base para realizar pruebas de software manuales y automatizadas a través de sus distintos tipos de peticiones, ya sea: crear un token de seguridad, obtener la lista de libros completa, obtener un libro por su Id, crear un nuevo registro de un libro, actualizar un registro de un libro y eliminarlo.

# Documentación recibida para las pruebas

Se reciben por parte de Sofka U las consideraciones a tener en cuenta para realizar las pruebas a la API RestFul Booking:

**Consideraciones**

* Genere un plan de pruebas con una estrategia de automatización acorde a lo ilustrado en las listas de reproducción sugeridas en el campus.
* Suba el resultado a un repositorio de código fuente **privado** como GitHub; GitLab; etc.
* Use Java como lenguaje de automatización.
* Use Cucumber en el proceso de automatización.
* Use un gestor de logs como por ejemplo log4j.
* Implemente programación orientada a objetos.
* Use buenas prácticas de programación.
* Recuerde que las automatizaciones deben ser independientes unas de otras, es decir, los escenarios no pueden tener dependencias unos de otros. Las automatizaciones siempre deben ser repetibles.
* Es necesario verificar el funcionamiento de los servicios web de la página: <https://restful-booker.herokuapp.com/apidoc/index.html>

# Alcance

Durante las pruebas se cubrirán los siguientes tipos de peticiones:

* Get Books
* Auth – Create Token
* Update Booking

# Fuera del alcance

Los siguientes tipos de peticiones y elementos no se probarán en este proyecto de prueba:

* UI
* Get Book por ID
* Create Booking
* Partial Update Booking
* Delete Booking
* HealthCheck
* Pruebas unitarias; Estas las realiza el equipo de desarrollo
* Ejecución de pruebas en sistema operativo Android y IOS

# Supuestos

# El equipo de QA proporcionará el set de datos para la ejecución de los diferentes casos de prueba.

# Establecimiento del entorno de pruebas.

# Estrategia de prueba

Se realizarán el siguiente tipo de pruebas:

**Se implementaron diferentes tipos de prueba funcionales de los siguientes tipos:**

* Pruebas de confirmación
* Pruebas de Humo

**Se implementaron diferentes tipos de prueba no funcionales de los siguientes tipos:**

* Pruebas exploratorias

Todas las pruebas se realizarán inicialmente bajo un ambiente de pruebas manual para comprender inicialmente de buena manera el aplicativo, y seguido se realizarán pruebas automatizadas en cada uno de las peticiones requeridas.

# Recurso Humano

|  |  |
| --- | --- |
| ROLES | Integrantes |
| Analista QA | Steven Duque, |

# Ambientes / Entornos de prueba

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Nombre | Versión | Descripción |
| Navegador | Chrome | 108 |  |
| Sistema operativo | Windows | 10 |  |
| Herramienta | IntelliJ |  | IDE para la elaboración de los test automatizados |
| Herramienta | Postman |  | Herramienta que permite la ejecución de pruebas exploratorias |

URL del sitio: <https://restful-booker.herokuapp.com/apidoc/index.html>

* Las credenciales de login con permisos de Admin para poder ingresar al aplicativo y probar todas las funcionalidades. El responsable de gestionar estos datos es el equipo de desarrollo.
* Datos requeridos en las peticiones Get Booking, Create Token y Update Booking, tales como: first name, last name, total price, deposit paid, booking dates (checkin y checkout), additional needs, username y password. El responsable de gestionar estos datos es el equipo de QA proporcionando datos de prueba ficticios.

# Insumos necesarios para las pruebas

|  |  |
| --- | --- |
| **Insumo** | **Descripción** |
| Computador portátil ASUS A55V | Con sistemas operativo Windows versión 8.1 o superior |

# Entregables

* Plan de pruebas
* Repositorio Git con los códigos de las pruebas realizados en IntelliJ

# Criterios de Entrada y de Salida

## Criterios de Entrada:

* El ambiente de pruebas debe de estar disponible para las pruebas
* Las herramientas de pruebas deben de estar instaladas y listas para ejecutar

## Criterios de Salida:

* No puede haber defectos críticos sin resolver o abiertos
* Se ha logrado una alta cobertura de los requerimientos de la funcionalidad del aplicativo.

# Criterios de finalización

Se aprobará el proyecto con un 100% de ejecución de las pruebas, pero con un 85% de aceptación. Será motivo de rechazo si:

* No se ha ejecutado el 100% de las pruebas.
* Cuando se ha ejecutado el 100% de las pruebas, pero el porcentaje de aceptación es menor al 85%.
* Cuando se ha ejecutado el 100% de las pruebas, se tiene un 85% de aceptación, pero existen errores de alto impacto.
* En caso de ocurrir que el proyecto no cumpla con el nivel exigido, el proyecto se rechaza por completo en su etapa de certificación.

# Criterios de suspensión

Se considera como criterio de suspensión de las pruebas que:

* La solución no cumple con las funcionalidades especificadas.
* El entorno de pruebas no sea estable y/o no retorne resultados confiables
* Se superó un porcentaje de casos de prueba fallidos sobre una funcionalidad

# Criterios de reanudación

Llegar a un acuerdo entre las partes para reanudar las pruebas o que se hayan solucionado los defectos/problemas encontrados.