**Plan de Pruebas: Automatización de Servicios Web RESTful Booker**

**1. Introducción**

Se detalla la estrategia para la automatización de pruebas de los servicios web de la aplicación RESTful Booker. El objetivo es implementar pruebas automatizadas a nivel de integración para detectar errores funcionales de manera temprana y proporcionar feedback rápido a los desarrolladores.

**2. Alcance**

El alcance de este plan de pruebas incluye la automatización de las siguientes funcionalidades de la API RESTful Booker:

* Auth
* Booking
* Ping

Se considerarán los métodos HTTP: POST, GET, PUT, PATCH y DELETE.

**3. Objetivos**

* Implementar pruebas automatizadas para los servicios web de RESTful Booker.
* Crear escenarios de prueba tanto para Happy Paths como para UnHappy Paths.
* Lograr una cobertura de pruebas amplia y efectiva.
* Proporcionar feedback temprano a los desarrolladores sobre posibles errores funcionales.

**4. Estrategia de Pruebas**

**4.1 Pruebas Manuales Iniciales**

1. Crear una colección en Postman para realizar pruebas manuales a los servicios web.
2. Documentar los resultados de las pruebas manuales para informar la creación de casos de prueba automatizados.

**4.2 Automatización de Pruebas**

1. Implementar scripts de prueba automatizados para las funcionalidades de Auth, Booking y Ping.
2. Cubrir todos los métodos HTTP relevantes: POST, GET, PUT, PATCH, DELETE.
3. Diseñar y ejecutar escenarios de Happy Paths y UnHappy Paths.
4. Implementar aserciones exhaustivas para validar respuestas, códigos de estado, y estructura de datos.

**5. Casos de Prueba**

**5.1 Auth**

1. Generar token de autenticación válido
2. Intentar generar token con credenciales inválidas

**5.2 Booking**

1. Crear un nuevo booking (POST)
2. Obtener detalles de un booking (GET)
3. Actualizar un booking existente (PUT)
4. Actualizar parcialmente un booking (PATCH)
5. Eliminar un booking (DELETE)
6. Obtener todas los booking (GET)
7. Filtrar booking por fecha de check-in y check-out
8. Intentar crear un booking con datos inválidos

**5.3 Ping**

1. Verificar el estado de salud de la API (Healthcheck)

**6. Herramientas y Técnicas**

* Postman: Para pruebas manuales iniciales y creación de la colección de requests.
* Ejecución de scripts de automatización para comprobar los resultados.
* Jenkins: Para la integración continua y ejecución automatizada de las pruebas.
* Git: Para el control de versiones de los scripts de prueba.

**7. Entregables**

1. Colección de Postman con pruebas manuales.
2. Scripts de pruebas automatizadas.
3. Informe de cobertura de pruebas.
4. Documentación de los casos de prueba implementados.
5. Informe de resultados de las pruebas automatizadas.

**8. Implementación de Pruebas**

**8.1 Desarrollo de Scripts**

1. Creación de scripts de prueba:
   * Utilizar Python como lenguaje principal de programación.
   * Implementar la biblioteca requests para interacciones HTTP con la API.
   * Estructurar los scripts siguiendo las mejores prácticas de programación y patrones de diseño para pruebas.
2. Definición de escenarios de prueba:
   * Utilizar el formato Gherkin para describir los escenarios de prueba.
   * Implementar los escenarios utilizando el framework Behave para BDD (Behavior-Driven Development).
   * Asegurar que los escenarios cubran tanto los Happy Paths como los UnHappy Paths.

**8.2 Ejecución de Pruebas**

1. Pruebas manuales iniciales:
   * Utilizar Postman para realizar pruebas manuales exploratorias.
   * Documentar los resultados de las pruebas manuales para informar el desarrollo de pruebas automatizadas.
   * Crear y mantener una colección de Postman con todos los endpoints y casos de prueba.
2. Ejecución de pruebas automatizadas:
   * Ejecutar los scripts de prueba desarrollados en Python.
   * Verificar los resultados de las pruebas y generar informes detallados.
   * Realizar un análisis de los resultados para identificar posibles problemas o áreas de mejora.
3. Integración continua:
   * Configurar un pipeline en Jenkins para la ejecución automatizada de pruebas.
   * Programar ejecuciones periódicas de las pruebas automatizadas.
   * Configurar notificaciones para alertar al equipo sobre fallos en las pruebas.

**8.3 Mantenimiento y Mejora Continua**

1. Actualización regular de los scripts de prueba:
   * Revisar y actualizar los scripts según los cambios en la API.
   * Optimizar los scripts para mejorar el rendimiento y la eficiencia.
2. Ampliación de la cobertura de pruebas:
   * Identificar áreas adicionales para automatización.
   * Desarrollar nuevos scripts de prueba según sea necesario.
3. Revisión y refinamiento del proceso:
   * Realizar revisiones periódicas del proceso de pruebas.
   * Implementar mejoras basadas en el feedback del equipo y los resultados de las pruebas.

**9. Criterios de Éxito**

* Todos los casos de prueba automatizados se ejecutan sin errores.
* Se logra una cobertura de pruebas de las funcionalidades especificadas.
* Los escenarios de Happy Paths y UnHappy Paths están cubiertos adecuadamente.
* Las pruebas automatizadas se pueden ejecutar de manera consistente y repetible.

**10. Riesgos y Mitigaciones**

1. Cambios en la API durante el desarrollo de las pruebas
   * Mitigación: Mantener una comunicación constante con el equipo de desarrollo y actualizar las pruebas según sea necesario.
2. Falsos positivos debido a la inestabilidad del entorno de pruebas
   * Mitigación: Implementar reintentos y mecanismos de manejo de errores en los scripts de prueba.
3. Tiempo insuficiente para automatizar todos los casos de prueba
   * Mitigación: Priorizar los casos de prueba más críticos y planificar iteraciones adicionales para aumentar la cobertura.

**11. Roles y Responsabilidades del Equipo de Pruebas**

**Ingeniero de Automatización de Pruebas**

* Desarrolla y mantiene scripts de prueba automatizados.
* Implementa casos de prueba para escenarios Happy y UnHappy Paths.
* Optimiza scripts para mejorar rendimiento y eficiencia.

**Analista de Calidad (QA)**

* Ejecuta pruebas automatizadas y manuales.
* Analiza resultados y reporta defectos.
* Genera informes de resultados y cobertura de pruebas.

**Líder de Calidad**

* Supervisa la ejecución del plan de pruebas.
* Coordina actividades entre equipos y gestiona recursos.
* Comunica progreso y resultados a stakeholders.

**12. Aprobaciones**

* Ingeniero de Automatización de Pruebas
* Analista de Calidad (QA)
* Líder de Calidad