**Plan de Pruebas: Automatización App Movil Booking.com**

**Objetivo**

Verificar y asegurar el correcto flujo de reserva de alojamiento en la app de Booking.com, para detectar errores funcionales.

**1. Alcance de las Pruebas**

**- Niveles de Prueba**

* Pruebas de Integración y Pruebas Funcionales

**- Tipos de Prueba**

* Pruebas Funcionales y Pruebas de Regresión

**2. Enfoque Técnico**

**- Herramientas de Automatización**

* **Aplicación Móvil:** Appium/UIAutomator2

**- Framework de Automatización**

* Lenguaje de Programación: Python
* Framework BDD: Behave

**- Integración Continua**

* Plataforma: Jenkins

**3. Metodología**

* Implementación de pruebas basadas en comportamiento (BDD) para alinear los casos de prueba con los requisitos del negocio
* Ejecución regular de pruebas de regresión para garantizar la estabilidad de las funcionalidades existentes
* Actualización continua de los scripts de prueba para reflejar los cambios en la aplicación

**4. Entregables**

* Scripts de prueba automatizados
* Informes detallados de ejecución de pruebas
* Documentación de los casos de prueba en formato Gherkin

Esta estrategia de pruebas está diseñada para asegurar una cobertura exhaustiva y eficiente de las funcionalidades críticas de la aplicación móvil de Booking.com, facilitando la detección temprana de defectos y manteniendo un alto estándar de calidad del producto.

5. Configuración del Entorno

* Instalar Appium Server
* Configurar el dispositivo móvil o emulador Android
* Instalar las dependencias necesarias (Python, Appium-Python-Client, uiAutomator2)

6. Casos de Prueba

Happy Paths

1. Búsqueda y Reserva Exitosa
   * Ingresar destino (Cusco)
   * Seleccionar fechas (14/02/2023 - 28/02/2023)
   * Configurar huéspedes (1 habitación, 2 adultos, 1 niño de 5 años)
   * Realizar búsqueda
   * Seleccionar el segundo resultado
   * Elegir habitación y opción de reserva
   * Llenar datos de reserva
   * Verificar datos y costo
   * Seleccionar método de pago (tarjeta de crédito)
   * Verificar monto total
2. Verificación de Costos
   * Comprobar que el costo se muestra correctamente en cada paso
   * Verificar que el monto total coincide con el costo mostrado inicialmente

Unhappy Paths

1. Búsqueda sin Resultados
   * Ingresar un destino inexistente
   * Verificar mensaje de error o falta de resultados
2. Fechas Inválidas
   * Intentar seleccionar fechas pasadas
   * Verificar que no se pueden seleccionar
3. Datos de Reserva Incompletos
   * Omitir campos obligatorios en el formulario de reserva
   * Verificar mensajes de error
4. Límite de Huéspedes
   * Intentar agregar más huéspedes de los permitidos
   * Verificar restricciones o mensajes de advertencia

7. Implementación

* Crear una clase base para configurar el driver de Appium
* Implementar métodos auxiliares para interactuar con elementos comunes
* Crear clases de prueba separadas para cada funcionalidad principal
* Utilizar decoradores de pytest para parametrizar pruebas cuando sea posible

8. Ejecución y Reportes

* Configurar la ejecución de pruebas
* Generar reportes detallados de las pruebas ejecutadas
* Implementar capturas de pantalla en caso de fallos

9. Mantenimiento y Actualización

* Revisar y actualizar las pruebas regularmente
* Mantener un registro de cambios en la aplicación que puedan afectar las pruebas
* Actualizar los localizadores de elementos si cambian en nuevas versiones de la app

**10. Riesgos y Mitigaciones**

**10.1 Cambios en la Interfaz de Usuario Durante el Desarrollo de las Pruebas**

* **Mitigación**: Mantener una comunicación constante con el equipo de desarrollo de la aplicación móvil y actualizar las pruebas según sea necesario.

**10.2 Falsos Positivos Debido a la Inestabilidad del Entorno de Pruebas Móviles**

* **Mitigación**: Implementar reintentos, tiempos de espera dinámicos y mecanismos de manejo de errores en los scripts de prueba para adaptarse a diferentes velocidades de red y rendimiento de dispositivos.

**10.3 Tiempo Insuficiente para Automatizar Todos los Casos de Prueba**

* **Mitigación**: Priorizar los casos de prueba más críticos para el flujo de reserva y planificar iteraciones adicionales para aumentar la cobertura de funcionalidades secundarias.

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

**Ingeniero de Automatización de Pruebas Móviles**

* Desarrolla y mantiene scripts de prueba automatizados utilizando Appium y Python.
* Implementa casos de prueba para escenarios Happy y UnHappy Paths en el flujo de reserva de alojamiento.
* Optimiza scripts para mejorar rendimiento y eficiencia en diferentes dispositivos móviles.

**Analista de Calidad (QA) Móvil**

* Ejecuta pruebas automatizadas y manuales en diversos dispositivos móviles.
* Analiza resultados y reporta defectos específicos de la aplicación móvil.
* Genera informes de resultados y cobertura de pruebas para la aplicación de Booking.com.

**Líder de Calidad de Aplicaciones Móviles**

* Supervisa la ejecución del plan de pruebas en el ecosistema móvil.
* Coordina actividades entre equipos de desarrollo móvil y QA, gestionando recursos.
* Comunica progreso y resultados a stakeholders, enfocándose en la calidad de la experiencia móvil.

**12. Aprobaciones**

* Ingeniero de Automatización de Pruebas Móviles
* Analista de Calidad (QA) Móvil
* Líder de Calidad de Aplicaciones Móviles