Tabla de contenido

[Especificación de Herramientas para Pruebas de Automatización con Appium en Booking 3](#_Toc164197007)

[Objetivo 3](#_Toc164197008)

[Herramientas Utilizadas 3](#_Toc164197009)

[Técnicas de Pruebas Utilizadas 4](#_Toc164197010)

[Detalles Adicionales 4](#_Toc164197011)

[Arquitectura del Script 5](#_Toc164197012)

[Esquema de arquitectura del proyecto 5](#_Toc164197013)

[Modo de Ejecución: 5](#_Toc164197014)

[Ubicarse en la carpeta del proyecto 5](#_Toc164197015)

[Reportes 6](#_Toc164197016)

[Conclusiones 6](#_Toc164197017)

**Técnicas de Pruebas Utilizadas** 4

**Especificación de Herramientas para Pruebas de Automatización con Appium en Booking**

**Objetivo**

El objetivo de esta especificación es detallar las herramientas utilizadas para implementar pruebas de automatización con Appium en la aplicación móvil de Booking, permitiendo la realización de reservas de alojamientos y viajes de manera eficiente y efectiva.

**Herramientas Utilizadas**

1. **Lenguaje de Programación:**
   * Java 11
2. **Gestión de Proyectos:**
   * Maven 3.8.5
3. **Automatización de Pruebas:**
   * Appium version 2.0.0-beta.71
4. **Entorno de Desarrollo Integrado (IDE):**
   * Android Studio
   * Intellij Idea
5. **Emulador de Dispositivos Android:**
   * Pixel 6 API 30 con Android 11
6. **Inspector de Appium:**
   * Appium Inspector 2023.4.3
7. **Entorno de Ejecución de JavaScript:**
   * Node 16.14.1
8. **Drivers de Automatización:**
   * UIAutomator 2.25.1
9. **Framework de Automatización:**
   * SerenityBDD con Screenplay como patrón de diseño

**Técnicas de Pruebas Utilizadas**

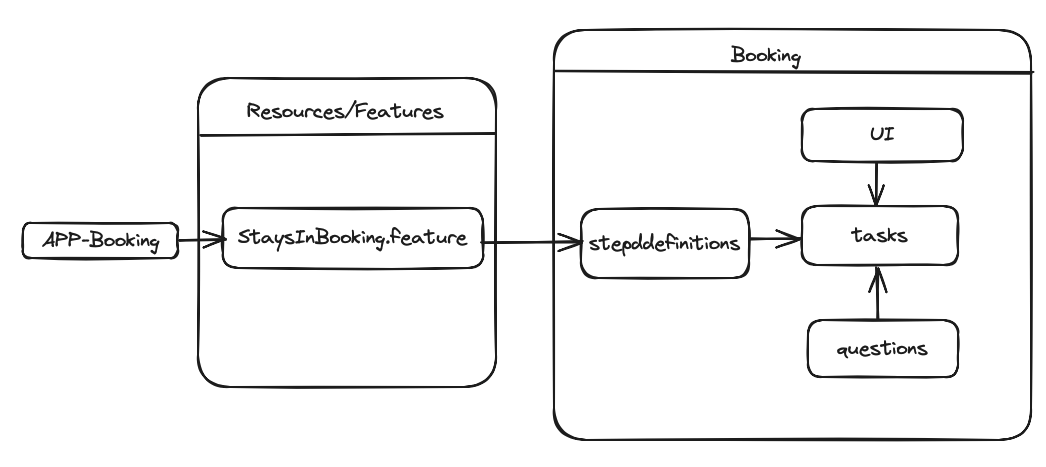
* **Pruebas de Funcionalidad:** Se verifican las funciones principales de la aplicación, como la búsqueda y reserva de alojamientos y viajes, para asegurar que se comporten según lo esperado.
* **Pruebas de Interfaz de Usuario (UI):** Se realizan pruebas para garantizar que la interfaz de usuario sea intuitiva, accesible y estéticamente agradable para el usuario final.
* **Pruebas de Integración:** Se comprueba la integración entre los diferentes componentes de la aplicación, como la autenticación de usuarios, la búsqueda de alojamientos y la realización de reservas.
* **Pruebas de Compatibilidad:** Se realizan pruebas en diferentes dispositivos Android para garantizar que la aplicación funcione correctamente en una variedad de configuraciones de hardware y software.
* **Pruebas de Rendimiento:** Se evalúa el rendimiento de la aplicación en términos de velocidad de carga, capacidad de respuesta y consumo de recursos para asegurar una experiencia fluida para el usuario.

**Detalles Adicionales**

* **Java 11:** Se utiliza como el lenguaje de programación principal para el desarrollo de las pruebas de automatización debido a su robustez y amplio soporte en el ecosistema de pruebas de software.
* **Maven 3.8.5:** Maven se emplea para la gestión de dependencias y la construcción del proyecto de automatización, facilitando la configuración y el mantenimiento del mismo.
* **Appium version 2.0.0-beta.71:** Appium se utiliza como el framework de automatización de pruebas móviles, permitiendo la interacción con la aplicación de Booking a través de scripts de prueba escritos en Java.
* **Android Studio e Intellij Idea:** Ambos IDE se utilizan como entornos de desarrollo integrados para la creación y el mantenimiento del proyecto de automatización, proporcionando herramientas específicas para el desarrollo de aplicaciones Android y una experiencia de desarrollo fluida.
* **Emulador de Dispositivos Android:** Se emplea un emulador de Pixel 6 con Android 11 para simular el entorno de prueba, permitiendo ejecutar las pruebas de automatización en un ambiente controlado y reproducible.
* **Appium Inspector:** Appium Inspector se utiliza para inspeccionar los elementos de la interfaz de usuario de la aplicación de Booking, facilitando la identificación de los elementos necesarios para las pruebas de automatización.
* **Node 16.14.1:** Node.js se utiliza como el entorno de ejecución de JavaScript para la ejecución de Appium Server, permitiendo la comunicación entre el servidor de Appium y los scripts de prueba.
* **UIAutomator 2.25.1:** Se utilizan los drivers de automatización de UIAutomator para interactuar con los elementos de la interfaz de usuario de la aplicación de Booking en dispositivos Android, garantizando una integración fluida con Appium.
* **SerenityBDD con Screenplay:** Se emplea SerenityBDD con el patrón de diseño Screenplay para organizar y estructurar las pruebas de automatización, facilitando la escritura de pruebas claras, mantenibles y escalables.

**Arquitectura del Script**

Esquema de arquitectura del proyecto



**Modo de Ejecución:**

Ubicarse en la carpeta del proyecto

Maven:

mvn clean verify

Genera un reporte resumen para la alta dirección:

mvn clean verify serenity:reports -Dserenity.reports=serenity-single-page-report

**Reportes**

La importancia de toda prueba automatizada radica en un buen reporte en donde se visualice el resultado de las pruebas y ayude a la trazabilidad de las mismas como también a la toma de decisiones

Reporte Serenity: target/site/serenity/index.html

Reporte Serenity Resumen: target/site/serenity/serenity-summary.html

**Conclusiones**

La combinación de estas herramientas y técnicas de pruebas proporciona un entorno robusto y eficiente para la implementación de pruebas de automatización con Appium en la aplicación móvil de Booking. Esto permite garantizar la calidad del software mediante la detección temprana de errores y la validación de las funcionalidades clave de la aplicación en diferentes aspectos.