



---

# GUIA DE INSTALACION Y EJECUCION

---

Para Tricount



17 DE SEPTIEMBRE DE 2025

PUNTOS COLOMBIA

Bogota

## Tabla de contenido

1. Descarga del Proyecto desde GitHub .....	2
2. Requisitos Previos .....	2
3. Configuración de Variables de Entorno.....	2
4. Instalación y Configuración de Appium .....	2
5. Configuración de Dispositivo Android.....	3
6. Configuración de Capabilities (serenity.conf / serenity.properties).....	3
7. Instalación de Dependencias del Proyecto .....	4
8. Ejecución de Appium Server .....	4
9. Ejecución de las Pruebas Automatizadas .....	4
10. Visualización de Reportes .....	4
11. Reporte de BUGs .....	5
12. Creación de HUs .....	6
13. Creación de casos de pruebas o features .....	8
14. Análisis de requerimiento .....	10
15. Reportes .....	11
Notas: .....	14
Contacto: .....	14

## 1. Descarga del Proyecto desde GitHub

- Abre una terminal o consola de comandos.
- Ejecuta:  
`git clone https://github.com/tu-usuario/tu-repo-tricount.git`
- Ingresa al directorio del proyecto:  
`cd tu-repo-tricount`

## 2. Requisitos Previos

- Java JDK 17
- Maven 3.8+
- Node.js y npm (para Appium)
- Android Studio (incluye SDK y AVD Manager)
- Appium Studio
- Git
- Dispositivo Android físico o emulador configurado

## 3. Configuración de Variables de Entorno

- JAVA\_HOME: apunta al directorio de instalación de JDK 17
- ANDROID\_HOME: apunta al SDK de Android (ejemplo:  
`C:\Users\<usuario>\AppData\Local\Android\Sdk`)
- Agrega a tu PATH:
  - %JAVA\_HOME%\bin
  - %ANDROID\_HOME%\platform-tools
  - %ANDROID\_HOME%\tools

## 4. Instalación y Configuración de Appium

- Instala Appium globalmente:  
`npm install -g appium`
- Verifica la instalación:  
`appium -v`

- (Opcional) Instala Appium Doctor para validar tu entorno:  
npm install -g appium-doctor  
appium-doctor

## 5. Configuración de Dispositivo Android

- Conecta tu dispositivo físico o inicia un emulador desde Android Studio.
- Habilita la depuración USB en el dispositivo físico y la opción de desarrollador.
- Verifica que el dispositivo esté conectado:  
adb devices
- El udid del dispositivo debe coincidir con el configurado en serenity.conf o serenity.properties.

## 6. Configuración de Capabilities (serenity.conf / serenity.properties)

Asegúrate de que los archivos serenity.conf y/o serenity.properties tengan las siguientes capacidades:

```
serenity.driver.capabilities.platformName=Android
serenity.driver.capabilities.udid=TU_UDID
serenity.driver.capabilities.appPackage=com.tribab.tricount.android
serenity.driver.capabilities.appActivity=com.bunq.tricount.android.ui.activity.LoadingTricountStartupInfoActivity
serenity.driver.capabilities.noReset=true
serenity.driver.capabilities.fullReset=false
serenity.driver.capabilities.autoGrantPermissions=true
serenity.driver.capabilities.newCommandTimeout=120
serenity.driver.capabilities.reportDirectory=reports
serenity.driver.capabilities.reportFormat=xml
serenity.driver.capabilities.testName=Untitled
serenity.driver.capabilities.instrumentApp=true
```

## 7. Instalación de Dependencias del Proyecto

Desde la raíz del proyecto, ejecuta:

```
mvn clean install
```

Esto descargará todas las dependencias necesarias y compilará el proyecto.

## 8. Ejecución de Appium Server

En una terminal, ejecuta:

```
appium
```

Esto iniciará el servidor en <http://localhost:4723/wd/hub>

## 9. Ejecución de las Pruebas Automatizadas

En otra terminal, ejecuta:

```
mvn clean verify
```

O bien, puedes ejecutar solo el runner específico:

```
mvn test -Dtest=TricountTestRunner
```

## 10. Visualización de Reportes

Al finalizar la ejecución, los reportes Serenity estarán en:

```
target/serenity-reports/index.html
```

Ábrelos en tu navegador para ver los resultados detallados.

## 11. Reporte de BUGs

ID del Bug: TC-001

Título: [BUG][App] - El App no tiene mensajes de validación de campos obligatorios

Fecha de detección: 2025-09-16

Reportado por: QA Senior Automatizador

Severidad: Alta

Prioridad: Alta

Estado: Nuevo

Versión: 1.0-SNAPSHOT

Ambiente:

- Dispositivo: Android 12
- App: Tricount.apk
- Automatización: Serenity BDD + Appium
- Navegador: N/A

Descripción

Al crear un grupo de gastos en la aplicación Tricount, el sistema no posee mensajes de validación que indiquen al usuario un posible error como por ejemplo un campo vacío.

Pasos para reproducir

1. Iniciar sesión en la app Tricount con una cuenta válida.
2. Seleccionar la opción para crear un nuevo grupo.
3. Ingresar un nombre vacío para el grupo (por ejemplo, “\_\_\_\_\_”).
4. Confirmar la creación del grupo
5. No muestra mensaje de validación de campo vacío

Resultado actual

No muestra mensaje de validación de campo vacío u obligatorio para crear un grupo, solo hace un hover pero no indica al usuario que debe hacer.

Resultado esperado

El sistema debe mostrar un mensaje:

“Este campo es obligatorio u/o No puede estar vacío”

#### Evidencia

- Video: [Adjuntar video de la ejecución automatizada o manual]
- Logs: [Adjuntar logs de la consola o reporte de Serenity]
- Captura de pantalla:  
![bug\_sin\_participantes.png](ruta/bug\_sin\_participantes.png)

#### Impacto

- Permite ingresar solo espacios en los campos
- Puede causar errores en la visualización o interpretación del usuario
- Afecta la experiencia de usuario

## 12. Creación de HUs

Descripción: Historias de usuario y criterios de aceptación para la gestión de gastos compartidos en la app Tricount.

\*

\* Este archivo documenta las funcionalidades principales cubiertas por la automatización:

- \* - Inicio de sesión con Google (válido e inválido)
- \* - Creación de grupo de gastos
- \* - Registro de gasto compartido
- \* - Visualización y liquidación de saldos

\* Cada historia de usuario está alineada con los escenarios definidos en la feature tricount.feature.

\* Los criterios de aceptación detallan los comportamientos esperados y los mensajes clave que debe mostrar la aplicación.

\* Actualizado: 2025-09-16

\*/

## Historia 1: Inicio de sesión con Google en Android

Como usuario de Tricount

Quiero poder iniciar sesión usando mi cuenta de Google en un dispositivo Android con la app instalada

Para acceder de forma rápida y segura a la aplicación

Criterios de aceptación:

- El usuario puede iniciar sesión con una cuenta de Google válida en la app instalada en Android.
- Si la cuenta de Google es inválida, se muestra un mensaje de error: "Cuenta de Google inválida".
- El sistema debe mostrar el mensaje correspondiente tras el intento de inicio de sesión.

## Historia 2: Creación de grupo de gastos

Como usuario autenticado en Android

Quiero crear un grupo para compartir gastos

Para organizar los gastos de un evento o viaje

Criterios de aceptación:

- El usuario puede crear un grupo ingresando nombre y participantes desde la app Android.
- El grupo aparece en la lista de grupos del usuario tras la creación.
- El nombre del grupo y los participantes quedan registrados correctamente.

## Historia 3: Registro de gasto compartido

Como miembro de un grupo en Android

Quiero registrar un gasto dentro del grupo

Para que todos los miembros vean el gasto y su distribución



Criterios de aceptación:

- El usuario puede agregar un gasto indicando monto, descripción y quién pagó desde la app Android.
- El gasto se distribuye entre los participantes según la configuración.
- El saldo de cada participante se actualiza correctamente.
- El gasto aparece en la lista de gastos del grupo.

Historia 4: Visualización y liquidación de saldos

Como miembro de un grupo en Android

Quiero ver los saldos y deudas entre los participantes

Para saber quién debe pagar a quién y poder liquidar las deudas

Criterios de aceptación:

- El usuario puede ver el resumen de saldos y deudas en la app Android.
- El usuario puede registrar pagos entre participantes para liquidar deudas.
- El sistema actualiza los saldos tras cada pago.
- El sistema muestra el mensaje de liquidación de deuda correctamente.

## 13. Creación de casos de pruebas o features

Feature: Gestión de gastos compartidos en Tricount

Como usuario de Tricount

Quiero poder registrar, visualizar y liquidar gastos compartidos

Para organizar fácilmente los gastos de grupo

Background:

Given el usuario abre la aplicación Tricount en un dispositivo Android con la app instalada

Scenario Outline: Inicio de sesión con Google

When el usuario inicia sesión con la cuenta de Google "<google\_account>"

Then el usuario ve el mensaje "<resultado>"

Examples:

| google\_account      | resultado      |  
| cuenta\_invalida@gmail.com| Cuenta de Google inválida |

Scenario: Creación de grupo de gastos

Given el usuario ha iniciado sesión

When el usuario crea un grupo llamado "Viaje Cartagena" con participantes "Ana y Luis"

Scenario: Registro de gasto compartido

Given el usuario está en el grupo

When el usuario agrega un gasto con descripción Hotel pagado por Ana

Then el gasto Hotel aparece en la lista de gastos del grupo

Scenario: Visualización y liquidación de saldos

Given el usuario consulta el resumen de saldos

When el usuario ve que Luis debe a Ana

And Luis registra un pago a Ana

Then la deuda entre Luis y Ana se liquida

## 14. Análisis de requerimiento

**Aplicación:** Tricount – Mobile App (iOS/Android)

- Necesitas ejecutar pruebas E2E directamente en dispositivos móviles.

**Criterios solicitados:**

- BDD + Gherkin
- Patrón Screenplay
- Gestor de dependencias (Maven, npm, etc.)
- Living Documentation (reportes dinámicos de ejecución)
- Interactions personalizadas
- Scenario Outline
- **Reporte de Bugs**

**Restricción clave:**

Cypress y Playwright son muy fuertes para Web y Web móvil emulado, pero no tienen soporte nativo sólido para apps móviles instaladas (APK/IPA).

Para mobile testing real, el estándar es Appium.

Dado que el requerimiento es Mobile App real + Screenplay + BDD + Living Documentation,

**El framework más recomendable es:**

Serenity BDD + Appium

- Con Serenity cumples con todos los criterios del reto (BDD, Gherkin, Screenplay, Living Docs, Interactions).
- Puedes correr las pruebas en emuladores o dispositivos reales (Android/iOS).
- Los reportes de Serenity generan automáticamente Living Documentation de las historias de usuario y escenarios.

**Por este motivo se configura de la siguiente manera:**

- Se usa Serenity BDD con Appium.
- Se define las historias de usuario en Gherkin.
- Se automatiza la ruta crítica (crear un grupo, añadir gastos, dividir gastos, ver balance).
- Se implementa al menos 1 Interaction (ej. ingresar texto, scroll o swipe en la app).
- Se usa Scenario Outline para probar con diferentes tipos de gastos.
- Se genera un reporte con serenity reports

## 15. Reportes

Para este requerimiento se utiliza los reportes de serenity BDD

Serenity genera reportes detallados, visuales y navegables que muestran:

- Escenarios ejecutados: pasaron o fallaron.
- Capturas de pantalla (si activas esta opción en pruebas web o móviles).
- Steps y tareas ejecutadas con duración.
- Resumen por feature y actor.
- Historial de ejecución y métricas de cobertura.

Los reportes son HTML, fáciles de compartir y ver en cualquier navegador.

### Configuración básica

Si usas **Maven**, agrega estas dependencias en tu pom.xml:

```
<dependency>
  <groupId>net.serenity-bdd</groupId>
  <artifactId>serenity-core</artifactId>
  <version>3.9.12</version>
</dependency>
<dependency>
  <groupId>net.serenity-bdd</groupId>
  <artifactId>serenity-junit</artifactId>
  <version>3.9.12</version>
</dependency>
<dependency>
  <groupId>net.serenity-bdd</groupId>
  <artifactId>serenity-screenplay</artifactId>
  <version>3.9.12</version>
</dependency>
```

En tu serenity.properties (o serenity.conf) puedes agregar configuraciones útiles:

serenity.outputDirectory=target/serenity-reports

serenity.take.screenshots=FOR\_EACH\_ACTION

serenity.report.show.step.details=true

- take.screenshots=FOR\_EACH\_ACTION → toma screenshot en cada step.
- outputDirectory → donde se generarán los reportes.

## Ejecutar pruebas y generar reportes

### Maven

Ejecuta tus tests con Maven y Serenity genera automáticamente los reportes:

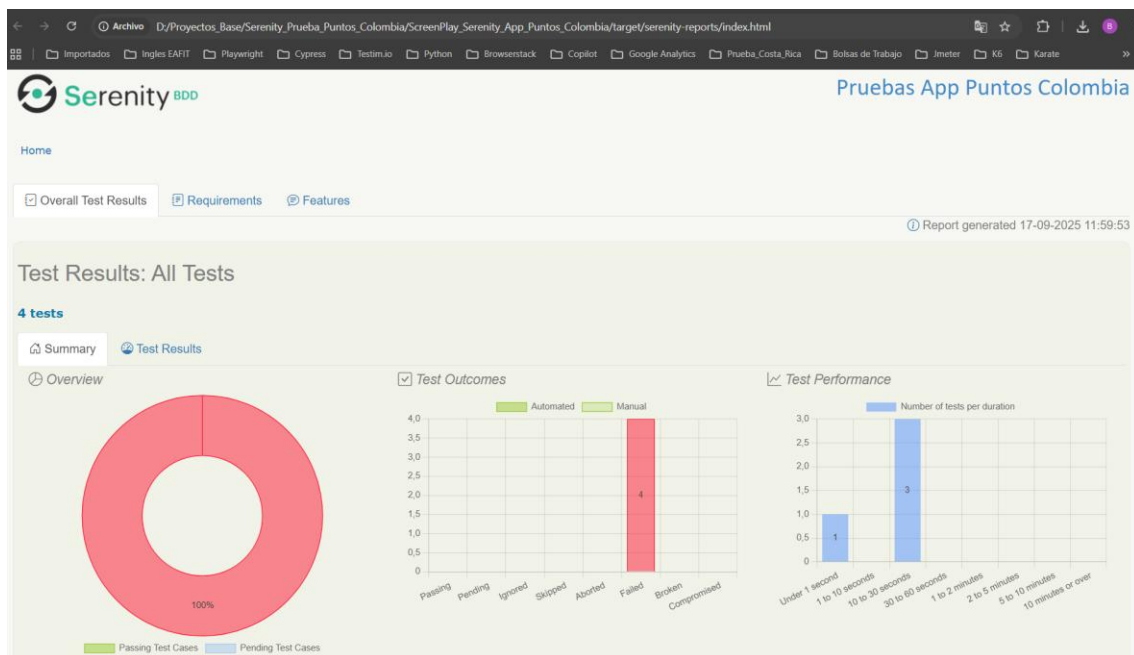
```
mvn clean verify
```

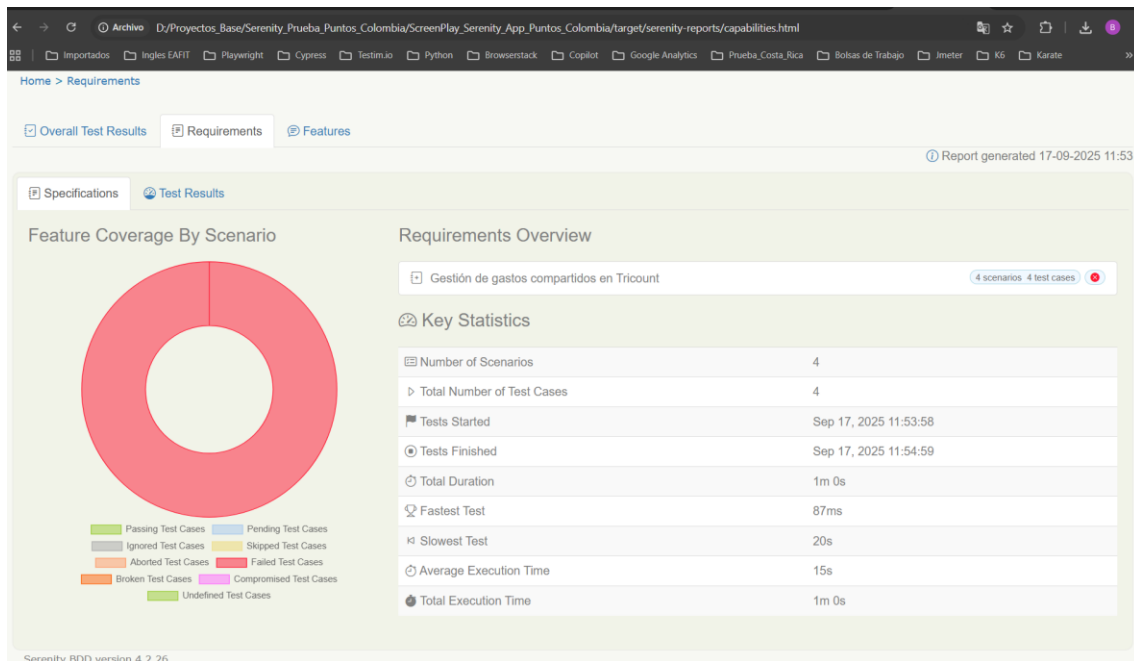
- Al finalizar, encontrarás los reportes en:

`target/serenity-reports/index.html`

- Ábrelo en cualquier navegador para ver el dashboard completo.

En el navegador se podrá ver el reporte de la siguiente manera:





The screenshot displays the 'Features' tab of a Serenity BDD report. It shows a table of test outcomes for features.

**Test Outcomes for Features**

Show 10 entries

Feature	Test Cases	Scenarios	% Pass	Result	Coverage
Gestión de gastos compartidos en Tricount	4	4	0%	Failed	0%

Showing 1 to 1 of 1 entries

Previous 1 Next

En caso de no visualizar el reporte, en la parte superior derecha del IDE, hay un icono de Maven con la letra “m”, selecciona:

- Plugins
- Serenity

Y ejecuta el plugin que dice:

- Serenity:aggregate

En consola veras el resultado y la URL para abrir el reporte en el navegador

## Notas:

- Si usas un emulador, asegúrate de que esté iniciado antes de correr las pruebas.
- Si usas un dispositivo físico, acepta cualquier permiso de depuración que solicite el sistema.
- Si necesitas cambiar el uid, modifícalo en los archivos de configuración.

## Contacto:

Para dudas o soporte, contacta al responsable del proyecto o revisa la documentación oficial de Serenity BDD y Appium.