

[Problema](#)

[Descripción de la aplicación](#)

[Requisitos del sistema](#)

[Requisitos que cumple el software](#)

[Pruebas](#)

[Estructura de la interfaz](#)

[Flujo del software](#)

## Problema

Existen muy pocas herramientas para el testing en Android que permitan sistematizar pruebas, y las que existen tienen ciertas falencias en el rendimiento o no contienen acciones que un tester desea probar (ej. MonkeyRunner Rotation).

## Descripción de la aplicación

Solución práctica para sistematizar aplicaciones android, útil para testers que deseen ejecutar un conjunto de pruebas repetitivamente en un mismo dispositivo. En donde por medio de una lista de comando adb shell puede crear pruebas y guardarlas para poder ejecutarlas en el dispositivo mientras logra visualizar fácilmente lo que sucede en este. Interfaz intuitiva y fácil. Esto con un óptimo rendimiento gracias a rápida conexión con el adb shell.

## Requisitos del sistema

- El equipo el que se ejecute la herramienta debe tener instalada la máquina virtual de java.
- El equipo debe tener las herramientas de desarrollo de Android, específicamente la herramienta Android Debug Bridge.
- La ruta de la carpeta platform-tools del SDK de Android debe estar configurada en el sistema para poder acceder al proceso adb y ejecutar las acciones de los eventos en el dispositivo Android.
- Debe haber un dispositivo Android o un emulador conectado a la máquina en la que se ejecutarán los eventos. Debe haber máximo un dispositivo para evitar inconvenientes al ejecutar
- El dispositivo Android debe permitir la creación de la carpeta /sdcard/screenshots en el sistema para no tener inconvenientes con la visualización de la pantalla en el computador.

## Requisitos que cumple el software

1. Plataforma java
2. El usuario podrá registrar y manipular eventos por medio de una interfaz gráfica.
3. El usuario podrá guardar y abrir los comandos usando archivos.
4. El usuario podrá hallar todas las funcionalidades en una sola interfaz.
5. El usuario podrá ejecutar las pruebas desde la interfaz gráfica en el dispositivo conectado por medio del Android Debug Bridge.
6. El usuario podrá registrar eventos los siguiente eventos para el Input Manager de Android
  - a. Key Event: 1 parámetro - Un evento que indica que una pulsación de tecla se produjo en un componente, el valor del código que puede asignarse está disponible en <http://developer.android.com/reference/android/view/KeyEvent.html>.
  - b. Rotation Event: 1 parámetro - Evento en el que se rota el dispositivo según el código: 0 indica 0°, 1 indica 90°, 2 indica 180°, 3 indica 270°.
  - c. Tap: 2 parámetros - Toque en la pantalla, recibe dos parámetros para las coordenadas X y Y en la pantalla donde se debe ejecutar el evento.
  - d. Text: 1 parámetro - Ingresa una cadena de texto sin espacios.
  - e. Swipe: 5 parámetros - Ejecuta un evento de arrastrar el dedo sobre la pantalla, recibe las coordenadas X y Y de dos puntos (fuente y destino) al igual que el tiempo en milisegundos que debe tomar el evento para ejecutarse.

## Pruebas

Asociación de pruebas y requisitos:

Requisito	Tipo de Prueba	Prueba
1. Plataforma java	Aceptación	Prueba 1
2. Registro y manipulación de eventos	Caja Negra , Unidad	Prueba 2
3. Abrir y guardar archivos	Caja Negra , Unidad	Prueba 3
4. Interfaz Única	Aceptación	Prueba 4
5. Funcionamiento de la	Caja Negra, Unidad	Prueba 5

clase Factory		
6. Funcionamiento de la clase ShellCommandExecuter	Caja Negra, Unidad	Prueba 6

Prueba	1. Requisitos del sistema
Procedimiento	Ejecutar el comando “adb” en la consola de comandos
Resultados esperados	Se debe mostrar la información acerca del uso de la herramienta Android Debug Bridge sobre la consola, de esta manera se verifica que la herramienta (y por lo tanto, la máquina virtual de Java también) está instalada en el sistema.
Resultados obtenidos	Ok

Prueba	2. Registro y manipulación de eventos
Procedimiento	Crear instancias de cada una de las clases que implementan la interfaz IEvent y ejecutar el evento getCommand.
Resultados esperados	El resultado al ejecutar el evento getCommand de la clase es la instrucción usada en la herramienta Android Debug Bridge para ejecutar un evento.
Resultados obtenidos	Ok

Prueba	3. Abrir y guardar archivos.
Procedimiento	Instanciar un objeto de tipo PersistenceControl que implemente la interfaz IPersistenceControl y guardar un objeto que implemente la interfaz Serializable, luego leer el archivo.
Resultados esperados	Los datos almacenados dentro del archivo son los mismos que los datos que tenía el objeto que se usó para crear el archivo.
Resultados obtenidos	Ok

Prueba	4. Interfaz gráfica
Procedimiento	Ejecutar el software y hacer clic sobre cada uno de los botones de la interfaz.

Resultados esperados	El software hace los llamados indicados a la principal de control al hacer clic en los botones de la interfaz gráfica.
Resultados obtenidos	Ok

Prueba	5. Clase Factory
Procedimiento	Solicitar una instancia de cada uno de los tipos de objetos que provee el sistema usando el método getInstance de la clase Factory.
Resultados esperados	Se obtiene una instancia de cada una de las clases de forma correcta.
Resultados obtenidos	Ok

Prueba	6. ShellCommandExecuter
Procedimiento	Ejecutar el comando “echo hola” usando la instancia de la clase ShellCommandExecuter
Resultados esperados	La respuesta del método debe ser la cadena de texto “hola”
Resultados obtenidos	Ok

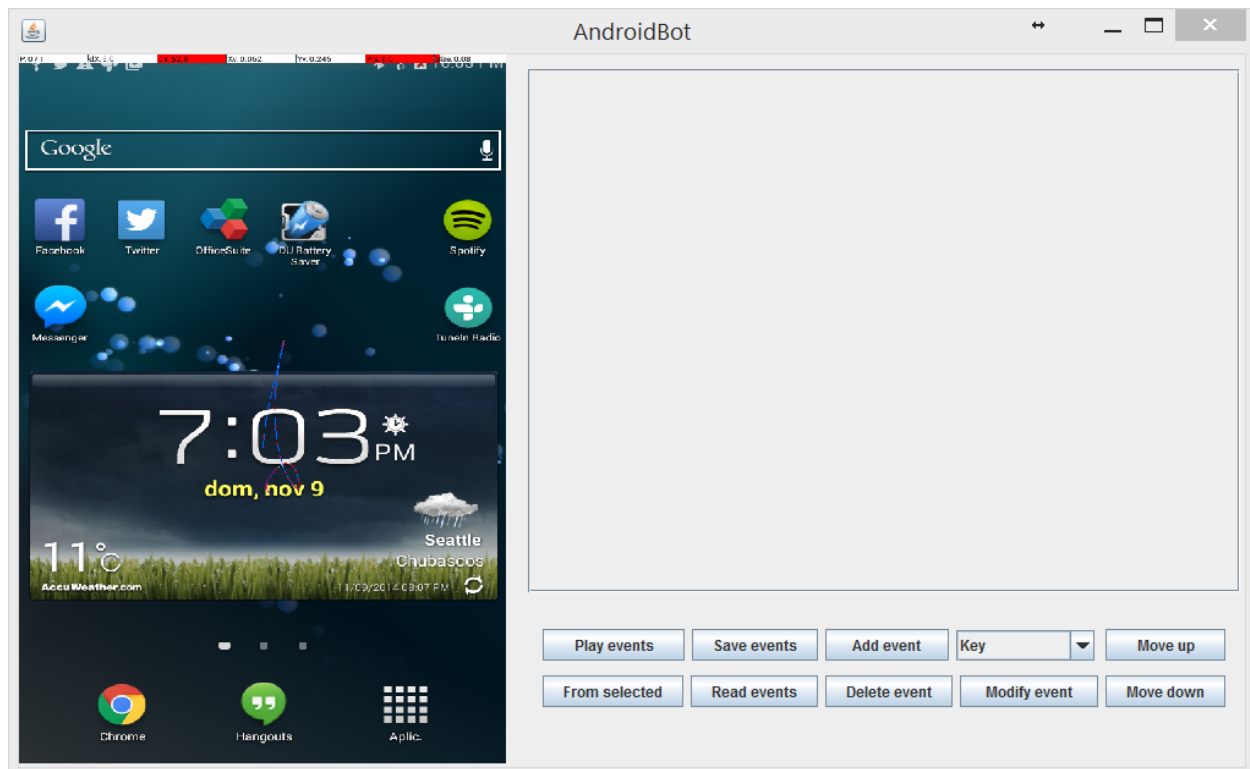
## Estructura de la interfaz

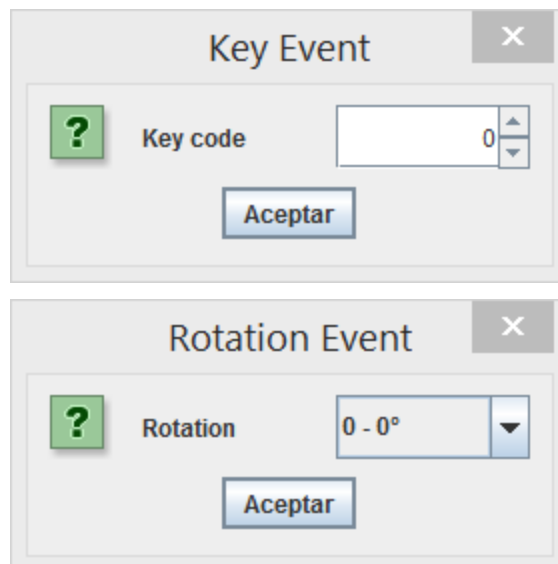
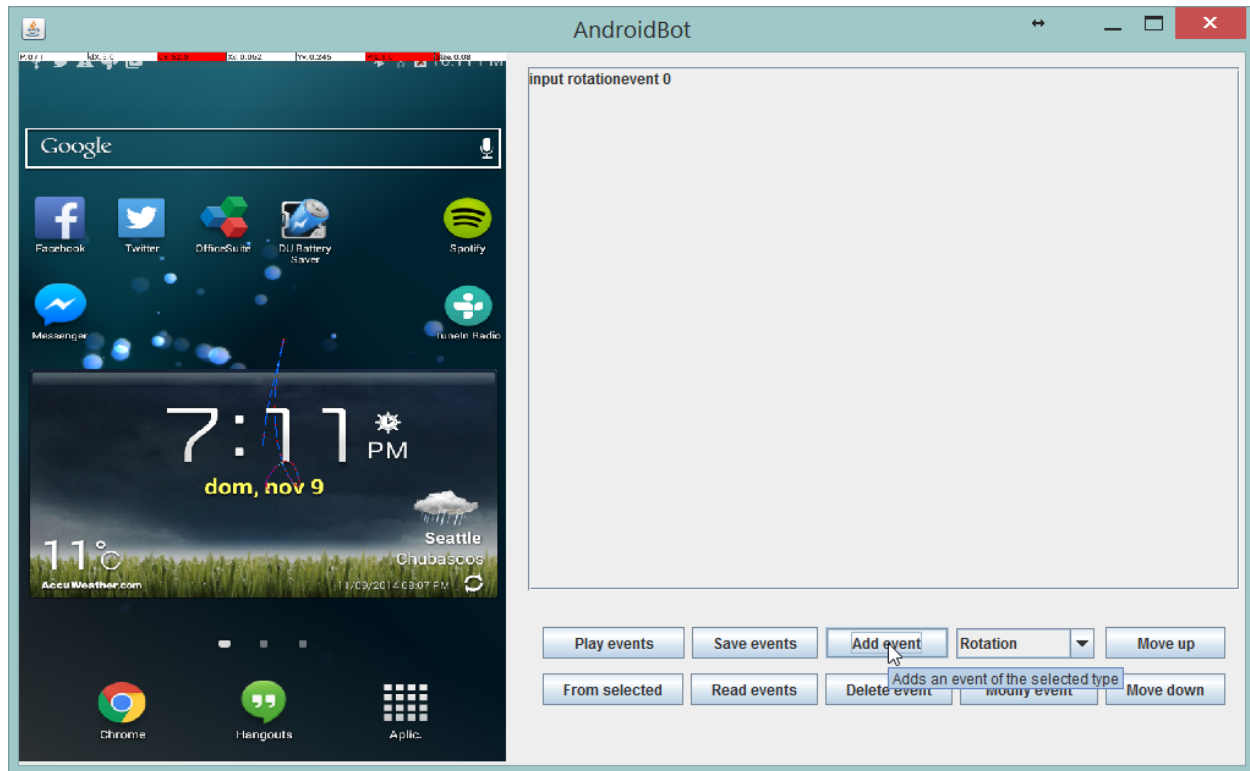
Al iniciar el software se verá la interfaz gráfica con tres zonas principales:

1. Zona de screenshot que muestra la imagen de la pantalla del dispositivo conectado actualizándola una vez por segundo. En esta zona se pueden crear eventos (ejecutándolos de una vez en el dispositivo) de tap y swipe.
2. Zona de botones en donde se pueden encontrar 10 diferentes opciones
  - a. Play Events: Se encarga de ejecutar los comandos que se encuentren en la lista de eventos.
  - b. Save Events: Se encarga de guardar la lista de eventos en un archivo, en este usted podrá elegir la ubicación y el nombre del archivo.
  - c. Add Event: Se encarga de agregar un evento a la lista (su funcionalidad esta descrita en requisitos del sistema), estos pueden ser :
    - i. KeyEvent
    - ii. Rotation Event
    - iii. Text

- iv. Swipe
  - v. Tap
  - d. MoveUp: Mueve el comando seleccionado, una posición arriba.
  - e. MoveDown : Mueve el comando seleccionado, una posición abajo.
  - f. Modify Event : Modifica el comando seleccionado, mostrando diferentes opciones según el evento.
  - g. Read Event : Despliega una interfaz para la selección de un archivo, previamente guardado. Y carga la lista de eventos en la zona de lista de eventos.
  - h. From Selected : Ejecuta una lista de eventos , donde su primer elementos es el evento seleccionado.
3. Zona de lista de eventos que se encarga de mostrar los eventos han sido programados por el usuario, esta lista permite seleccionar un evento específico (el fondo será blanco) y los eventos son representados en la interfaz por el comando que generan para ser ejecutados en el shell de Android.

## Flujo del software





Swipe Event

?

x1

y1

x2

y2

millis

0

0

100

100

1.000

+

-

+

-

+

-

+

-

+

-

Aceptar

Tap Event

?

x1

y1

0

0

+

-

+

-

Aceptar

Text Event

?

text

asdf

Aceptar

