



Framework.

Historial de versiones

Versión	Fecha	Autor
LFRM-001-00	17 - 04 - 2020	Oscar Antonio Hernández Mojica
LFRM-001-01	10 - 05 -2020	Oscar Antonio Hernández Mojica

Instalación	3
Antes de empezar	3
Pre-requisitos	3
Información del ambiente	4
Uso	4

README

Instalación

Antes de empezar

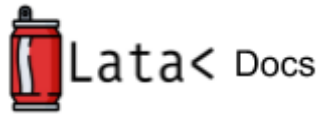
Java Development Kit (JDK) permite que se pueda ejecutar y compilar código Java, ya que es indispensable ya que los dispositivos Android están basados en este lenguaje, además de proveer herramientas como **Android Debug Bridge** (ADB) y **uiautomatorviewer** lo permitirá el desarrollo del framework, puedes descargar el JDK más adecuado al sistema operativo de tu equipo desde la [página oficial](#) de ORACLE.

Android Studio provee de las herramientas para emular un dispositivo Android en caso de no contar con alguno, puedes instalarlo desde su [página oficial](#).

Es importante establecer las variables de entorno de acuerdo al sistema operativo de tu equipo para **JDK** y **Android Studio**

Pre-requisitos

- Java Development Kit 8
 - Android Debug Bridge
- Android Studio
 - SDK Platforms
 - Instalar el API del dispositivo que se necesite, de preferencia mayor o igual a Android 7.0 (Nougat) API 24
 - SDK Tools
 - Android Emulator
 - Android SDK Platforms-Tools
 - Android SDK Tools
- Python 2.7
 - [uiautomator](#)
- Dispositivo Android
 - En ajustes se debe de habilitar el [modo desarrollador](#) para poder ejecutar los procesos desde el framework en Python.
 - **El dispositivo debe de estar conectado por medio de una cable USB al equipo antes de iniciar el framework**
 - Para emular un dispositivo Android
 - Abrir Android Studio
 - Tools -> AVD Manager
 - Create Virtual Device
 - Seleccionar el hardware y system image que usted prefiera
 - Instalará los paquetes necesarios e iniciará el emulador
 - Si ya ha creado un dispositivo virtual, ejecutar desde línea de comando:



- Para mostrar los dispositivos instalados
 - **emulator -list-avds**
- Para iniciar el dispositivo virtual:
 - **emulator -avd Nombre_del_dispositivo**

Información del ambiente

- Pixel 2 - Android 9.0 Pie API 28
- Remi Note 8 - Android 9.0 Pie API 28

Uso

1. Descargar o clonar el repositorio de GitHub.
2. Editar el archivo data-test.json:
 - a. wifies: lista que describe la secuencia de encendido (1) y apagado (0).
 - b. phones: lista números telefónicos al se realizara una llamada.
 - c. operations: lista de operaciones a evaluar, junto con el resultado esperado.
3. El dispositivo debe de estar conectado al equipo antes de comenzar el test.
4. Ejecutar en una terminal en raíz del proyecto el siguiente comando:
 - a. **\$ python main.py**
5. A continuación se mostrará en consola el resultado de los test en secuencia:
 - a. Se mostrará con un punto ".", los test que resultaron exitosos
 - b. Se mostrará con una "F", los test que resultaron con un error
6. Al ejecutar el test, se crea un archivo de texto de acuerdo a la versión del framework en el directorio **output_test**, donde podrá consultar información más detallada de la ejecución.