



Framework.

Historial de versiones

| Versión | Fecha | Autor |
|-------------|----------------|--------------------------------|
| LFRM-001-00 | 17 - 04 - 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

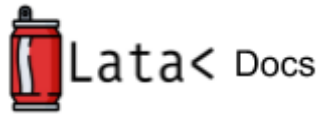
Java Development Kit (JDK) permite que se pueda ejecutar y compilar código Java, ya que es indispensable ya que los dispositivos Android están basados en este lenguaje, además de proveer herramientas como **Android Debug Bridge** (ADB) y **uiautomatorviewer** lo permitirá el desarrollo del framework, 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

Descargar o clonar el repositorio de GitHub, desde la carpeta raíz del proyecto abra una terminal y ejecute lo siguiente comando:

\$ python main.py

A continuación debe de mostrar en terminal el menú para la ejecución de los diferentes procesos, cada proceso puede ser llevado a cabo usando **adb** o **uiautomator**:

```
Call a number
    1) by adb shell
    2) by uiautomator
Turn ON Wifi
    3) by adb shell
    4) by uiautomator
Turn OFF Wifi
    5) by adb shell
    6) by uiautomator
7) Exit
Select an operation: █
```

Para un ejemplo seleccionamos la **opción 2**, llamar a un número usando uiautomator. Al seleccionar la opción 2, solicita al usuario el número el cual desea llamar:

```
2) by uiautomator
Turn ON Wifi
3) by adb shell
4) by uiautomator
Turn OFF Wifi
5) by adb shell
6) by uiautomator
7) Exit
Select an operation: 2
Enter the number to call: 4492846104
DEVICE: emulator-5554 CALLING 4492846104
START at 2020-04-17 22:34:21.564640
END at 2020-04-17 22:34:39.275745
Call a number
1) by adb shell
2) by uiautomator
Turn ON Wifi
3) by adb shell
4) by uiautomator
Turn OFF Wifi
5) by adb shell
6) by uiautomator
7) Exit
Select an operation: █
```

La ejecución del proceso debe de verse reflejado en el dispositivo, en caso contrario, tenemos un bug. El framework mostrará las marcas de tiempo de cuando inicia el proceso y de cuando finaliza, además de la información y estado del mismo.

Cuando finaliza el proceso, LATA vuelve a preguntar al usuario si desea continuar con el ejecución de otro proceso o simplemente salir del framework.