**Architecture**

Testing suites:

* Dial Number ADB
* Dial Number UI
* Wifi ON
* Wifi OFF

Testing libraries:

* Functions
* Execution

Open Source Libraries :

* ADB shell
* UI automator
* Python libraries

**Core**

The libraries classes1 and Scripts where created.

The classes1 testing suite holds the following files:

* dial\_number\_adb.py
* dial\_number\_ui.py
* wifi\_off.py
* wifi\_on.py

Each of these files hold functions which call their respective scripts/methods that where created.

Scripts houses two files:

* Execution.py
* Functions.py

Inside Functions.py various methods where created to achieve the project goals which are: Turning the Wi-Fi on or off and making a call to a phone number using ADB shell and/or Ui Automator.

The execution.py file consists as the project´s main, where the testing suites inside classes1 are called based on the desired test that the user wants to run and is executed.

**Methods**

The following methods are located inside functions.py

* read\_serial() :
  + Function that returns the serial number of device in the first position
* serial\_number\_device():
  + Function that returns the serial number and device for future assignment
* phone\_target():
  + Function which receives the name of the target app to open it on the device
* input\_phone\_number():
  + Function that receives the desired phone number to call from the user.
* number\_validation():
  + Function that validates each phone number with a regex and a 15 max length which is the length limit for a phone number around the world and with a min length of 3 for emergency numbers.  
    Returns true if number is validated
* Converter():
  + Function that returns a string corresponding to each number or character inside

the phone number, to be later appended in the resourceId inside dial\_number() so these can be identified.

* Dial\_number():
  + Function that uses input\_phone number, number\_validation and converter to dial the desired number
* Dial\_adb():
  + Function that asks for the device, inputs phone number, validates it and finally calls the number using adb
* wifi\_on\_func():
  + Function that turns the wifi setting on
* wifi\_off\_func():
  + Function that turns the wifi setting off

**Set Up**

1. User must have installed the following

* Uiautomator
* Adb shell
* Android SDK tools
* Java 7
* Python 2.7

1. User must have android smart phone to run project on (Phone must have developer options on and the USB configuration in Media Transfer Protocol)
2. Clone repository

<https://github.com/charliepatronr/Proyecto-SQA-II.git>

1. Run execution.py

\*Note\*

Due to the Samsung Android phone that was used during the development of this framework not all ui automator objects present in the scripts will match other android phone objects. Remove “Samsung” from source-id´s, package names, class names, etc. when present.

**Environment**

Samsung Galaxy Grand Prime+

Model number: SM-6532M

Android Version: 6.0.1