

Ricardo German Serrano Rodríguez

0192775

Joaquín Miranda Castro

0192493

Juan Carlos Patrón Ruano

0194229

Pruebas automatizadas

The following document explains the different folders, libraries and methods which are used within the automation framework.

Folders

* docs: This directory contains all of the documentation for the automation framework. Including the business case (architecture within document, release notes, setup to execute scripts, environment etc.
* qa: This folder contains documents related to QA which include test plan (test strategy), test cases, traceability matrix, reports and execution logs.
* src: This directory contains the scripts which are divided into different folders:
  + lib: contains different libraries which house the methods utilized in the scripts
  + scripts: contains the scrips which contain the end to end flows for the execution of the test cases, to test the functionalities for the applications under test of the mobile device. The methods in the libraries are used in this folder.
  + Suits: Contains the definition of the test suites, which are essentially various test cases which have a common functionality, which are called in conjunction to have a more robust test.

Methods:

The following are methods defined in the different libraries.

* Logger: this library is in charge of showing the data in console and in the file:
  + begin\_log: Starts a new log in de specified file.
  + write\_log(mensaje): Writes log and in console displayed received message.
  + end\_log(): At the end writes a report with the initial and final times.
  + error\_log(mensaje): Writes the error message.
* Utils: this library handles simple methods that can be reused, but do not have such a specific task that requires a specific library to be created for each.
  + validate\_number(number): Validates the phone number being dialed. This validation recognizes the emergency number 911, national Mexican phone numbers and international numbers with the + prefix.
  + read\_json: reads any son file and converts it to a dictionary
  + get\_device\_data: determines the version of the android based on the serial number after reviewing a specific file.
  + is\_close: compares if two floats are almost identical, due to the fact that python and android sometimes return different results in decimal numbers.
* PhoneControl: This library provides the methods to connect the phone via ADB and UI Automator. It also has the tool necessary to execute a variety of actions within the mobile device.
  + read\_serial: Identifies the dies serial number within the adb list.
  + unlock\_phone: Unlocks the cellphone as long as there is no security measure such as a password.
  + click\_home: Simulates pressing the home button.
  + switch\_button(texto): Executes a click to a switch which contains the text passed as a paramater and which className is android.widget.Switch.
  + click\_button(texto): Executes a click to a switch which contains the text passed as a paramater and which className is android.widget.TextView
  + longclick\_button(texto): Executes a long click on a button which contains the text passed as a parameter and whose className is android.widget.TextView.
  + button\_exists(texto, className): Identifies if a button that contains the text and className currently exists.
  + click\_detailed\_button(className, packageName, description): Executes a click to a button that contains the three parameters which are passed
  + longclick\_detailed\_button(className, packageName, description): Executes a long click to a button that contains the three parameters which are passed
  + detailed\_button\_exists(className, packageName, description): Identifies if the a button that includes the three parameters which are passed exists.
  + set\_text\_textfield(packageName, contenido): modifies the android.widget.EditText which contains the packageName received as a parameter with the content passed in as well.
* Calculator: This library provides the methods regarding the calculator application, validating the results and calculating the desired operations.
  + Calculate: executes any of the 4 operations, the amount of digits can be specified, only 2 numbers can be used for a calculation
  + validate\_digits: for every number it validates that it cannot contain more than 15 digits, and no more than 10 decimal numbers.
  + validate\_number: validates the numbers that are inputted.
  + Division: executes the division operation
  + Addition: executes the sum operation
  + Subtraction: executes the subtraction operation
  + Multiplication: executes the multiplication operation
  + validate\_result: comprares the result of ui automator with the expected result that was executed in python with a certain margin of error to accommodate for the different android based calculations