

CORE

El presente documento describe la organización de los archivos y carpetas empleadas para el desarrollo framework **LATA** tanto como la descripción de sus elementos, métodos y funciones.

ROOT

El framework consta de los siguientes archivos y carpetas:

→ **lata**

- ◆ output_test
 - log.txt
- ◆ screenshots
 - file_name.png
 - file_name.uix
 - ...
- ◆ auida.py
- ◆ log_record.py
- ◆ main.py

output test:

En este directorio es donde se contendrán los diferentes logs que el framework genere y/o actualice conforme el desarrollo de este evolucione.

screenshots:

Este directorio contiene los archivos generados por el método '**uiaviewer_generator**' de la clase '**AndroidDevice**', cual se describirán más adelante. Estos archivos son utilizados por herramienta **uiautomatorviewer** para realizar una inspección más detallada de las diferentes pantallas, como la descripción y posición de los elementos, del dispositivo Android.

auida.py:

Contiene los scripts y métodos necesarios para realizar los procesos para realizar una llamada y encender/apagar el WiFi mediante adb shell y uiautomator.

`select_device`

Descripción:

- Verifica si hay un dispositivo conectado, de ser así regresa el serial de dicho dispositivo. Por defecto regresa el primer dispositivo conectado.

Parámetros:

- **device**: posición del dispositivo de la lista de dispositivos conectados o disponibles.

Retorna:

- Retorna el serial del dispositivo en un string.

`check_list`

Descripción:

- Por comando de adb, verifica si hay dispositivos disponibles o conectados.

Retorna:

- Si existen dispositivos conectados, retorna un lista con dichos dispositivos.
- Retorna None cuando no hay dispositivos conectados.

`dial_number`

Descripción:

- Realiza una llamada al número dado por comando **adb**.

Parámetros:

- **number**: número al cual se realizará la llamada

Retorna:

`hang_up`

Descripción:

- Finaliza la llamada si existe una en curso.

`adb_open_settings`

Descripción:

- Abre el menu de configuracion por comando **adb**.

`turn_on_wifi`

Descripción:

Enciende el WiFi del dispositivo por comando adb.

`turn_off_wifi`

Descripción:

Apaga el WiFi del dispositivo por comando adb.

`adb_calling_test`

Descripción:

- Realiza una llamada al número dado y la finaliza después de un tiempo especificado.

Parámetros:

- **number**: número al cual se realizará la llamada
- **delay**: tiempo especificado entre el evento de realizar la llamada y finalizar llamada. Por defecto es **5**, es decir, 5 segundos.

`adb_wifi_test`

Descripción:

- Según el estado dado, enciende o apaga el WiFi del dispositivo por comando **adb**.

Parámetros:

- **on_off**: estado el cual procederá a encender o apagar el WiFi (ON/OFF)

`uiaviewer_generator`

Descripción:

- Realiza una captura de la pantalla en la cual se encuentra el dispositivo. Genera un archivo *.png y otro *.uix con el mismo nombre, para ser utilizadas por la herramienta uiautomatorviewer para su inspección más a detalle.

`click_espanish_button`

Descripción:

- Busca el elemento especificado según el UIMatcher, label y className especificado. Para aumentar cobertura sobre dispositivos que estén configurados en inglés y otros en español. Esta función pregunta si existe el elemento en español, sino existe, intenta buscarlo en inglés. Permite realizar el proceso anterior mediante un segundo UIMatcher.

Parámetros:

- **matcher**: UIMatcher para realizar la búsqueda
- **spanish**: Label del elemento traducido al español
- **english**: Label del elemento traducido al inglés
- **matcher**: Segundo UIMatcher para realizar la búsqueda. Solo realizará la búsqueda si no se encuentra el primer matcher y este es especificado.

`initial_state`

Descripción:

- Establece al dispositivo en la pantalla de HOME.

`type_number`

Descripción:

Itera y pulsa cada uno de los elementos del string **number**.

Parámetros:

- **number**: número en string el cual se realizará la llamada.

`uia_calling_test`

Descripción:

- Establece el dispositivo en la pantalla de HOME.
- Busca y pulsa el botón de **Teléfono**.
- Busca y pulsa el botón de **teclado**
- Llama la función **type_number** con el número dado por el usuario
- Busca y pulsa el botón para **marcar**
- Espera el tiempo especificado, por defecto 5 segundos.
- Busca y pulsa el botón para **finalizar llamada**

Parámetros:

- number: número al cual se realizará la llamada
- delay: tiempo de espera entre **marcar** y **finalizar llamada**

`quick_turn_on_wifi`

Descripción:

- Abre el menú de configuraciones rápidas
- Verifica si el botón de WiFi está apagado
 - Si está apagado, pulsa el botón para encenderlo
 - Si está encendido, no realiza nada

`quick_turn_off_wifi`

Descripción:

- Abre el menú de configuraciones rápidas
- Verifica si el botón de WiFi está encendido
 - Si está encendido, pulsa el botón para apagarlo
 - Si está apagado, no realiza nada

`uia_quick_wifi_test`

Descripción:

- Según el estado dado, enciende o apaga el estado del WiFi en el menú de configuraciones rápidas

Parámetros:

- **on_off**: estado el cual procederá a encender o apagar el WiFi (ON/OFF)

`setting_turn_on_wifi`

Descripción:

- Verifica si el botón de WiFi del menú de configuraciones no está encendido
 - Si no está encendido, pulsa el botón
 - Si está encendido, no realiza nada

`setting_turn_off_wifi`

Descripción:

- Verifica si el botón de WiFi del menú de configuraciones está encendido
 - Si está encendido, pulsa el botón
 - Si no está encendido, no realiza nada

`settings_wifi_test`

Descripción:

- Abre el menú de configuraciones o ajustes del dispositivo
- De acuerdo al estado dado, realiza lo siguiente:
 - **ON**: Llama a la función para encender el WiFi
 - **OFF**: Llama a la función para apagar el WiFi
- Cierra el menú de ajustes estableciendo la pantalla de HOME

log_record.py:

Contiene los métodos necesarios para llevar un registro de los eventos, status y errores durante la ejecución de los procesos.

set_serial

Descripción:

- Guarda en una variable de instancia el serial del dispositivo seleccionado.

Parámetros:

- **serial:** ID del dispositivo seleccionado.

set_wifi_status

Descripción:

- Guarda en una variable de instancia el estado del WiFi.
- Imprime en pantalla que se realizó la operación.

Parámetros:

- **status:** Estado del WiFi a guardar

set_call

Descripción:

- Guarda en una variable de instancia el número a llamar.
- Imprime en pantalla que se realizó la operación.

Parámetros:

- **number:** número al cual se realizará la llamada.

set_event

Descripción:

- Guarda en una variable de instancia el momento que inicia y finaliza un proceso de acuerdo al parámetro **event** (START/END)
- Imprime en pantalla que se realizó la operación.

Parámetros:

- **event:**
 - START: guarda la marca de tiempo de cuando inició el proceso.
 - END: guarda la marca de tiempo de cuando finalizó el proceso.

add_log

Descripción:

De acuerdo a la acción (**action**) especificada, guarda en un archivo **log.txt** en el directorio **output_test** la información del proceso que se realizó.

Parámetros:

- **action:**

- CALL: escribe en el archivo log.txt la información guardada para el proceso de llamada.
- WIFI: escribe en el archivo log.txt la información guardada para el proceso de encender/apagar el WiFi.

main.py:

Archivo principal para la ejecución del framework **LATA**. Muestra al usuario un menú para llevar a cabo los distintos procesos, además de proveer métodos para validar las entradas del usuario.

`is_valid_number`

Descripción:

- Valida si el número proporcionado es válido para realizar una llamada telefónica.

Parámetros:

number: número ingresado por el usuario para realizar una llamada.