Pruebas de Aceptación: Python - Robot Framework

Willy Maykros Romero Naula 6668

Ingeniería en Software, Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo

SOFIP46: Validación y Verificación de Software
Ing. Raúl Rosero

11 de julio 2022

1. Pruebas de Aceptación.

En el proceso de validación y verificación de software es necesario el uso de varios tipos de pruebas, sin embargo, entre los diferentes tipos de pruebas, las pruebas de aceptación son las que tienen una mayor relación con los requisitos. Se utilizan para probar las necesidades, los requisitos y los procesos comerciales del usuario, y se llevan a cabo para determinar si un sistema cumple o no los criterios de aceptación y para permitir que los usuarios, clientes u otras entidades autorizadas determinen si aceptarán o no el sistema.

Al igual que los demás tipos de pruebas las pruebas de aceptación tienen que especificar casos de prueba, sin embargo, aporta dos facilidades, por un lado, la especificación de casos de prueba, en este caso requerirá menos esfuerzo manual ya que se definen o generan automáticamente a partir de la especificación de requisitos. Por otro lado, la propia especificación de requisitos terminará teniendo una mayor calidad debido al uso de un lenguaje más estructurado, reduciendo problemas típicos como la ambigüedad, la inconsistencia y la incorrección.

Para mejorar las actividades de pruebas de aceptación y especificación de requisitos, puede ser ventajoso realizar estas actividades en paralelo, lo que, además de aumentar la calidad de los requisitos, también permite reducir el tiempo y los recursos dedicados a ellos. Aunque comenzar las actividades de prueba al comienzo del proyecto cuando se obtienen los requisitos se considera una buena práctica, no siempre es así porque la obtención de requisitos y las pruebas están separadas en los procesos de desarrollo tradicionales.

2. Robot Framework.

El marco de Robot se destaca por su poderoso lenguaje basado en palabras clave que incluye bibliotecas listas para usar. Robot no requiere ningún tipo de implementación, ya que es posible utilizar palabras clave con implementaciones implícitas (con el uso de bibliotecas específicas como Selenium2). Robot es un framework de código abierto, relacionado con el desarrollo dirigido por pruebas de aceptación (ATDD), que es independiente del sistema operativo y se implementa de forma nativa en Python y Java, y se puede ejecutar en Jython (JVM) o IronPython (.NET).

La estructura que maneja es simple y se puede dividir en cuatro secciones:

1. Sección 1, Configuración, *Settings*, donde se configuran las rutas a los archivos auxiliares y bibliotecas utilizadas.
2. Sección 2, Variables, *Variables*, especifica la lista de variables que se utilizan, así como los valores asociados.
3. Sección 3, Casos de prueba, *Test Cases*, donde se definen los casos de prueba.
4. Sección 4, Palabras clave, *Keywords*, define palabras clave personalizadas para implementar los casos de prueba descritos en la sección Casos de prueba.

A continuación, se muestra un ejemplo del uso de Robot.

```
*** Settings ***
Documentation     Web Store Acceptance Test
Library           Selenium2Library

*** Variables ***
${product} Blouse

*** Test Cases ***
Login
    Open the browser on <www.http://
        automationpractice.com>
    Input Text id=searchBar ${product}
...
*** Keywords ***
Open the browser on <$(url)>
    Open Browser $(url)
```

3. Metodología.

El enfoque metodológico tiene los siguientes pasos:

1. Especificación de requisitos.
2. Especificación de casos de prueba.
3. Refinación de casos de prueba por el evaluador.
4. Elaboración de scripts de prueba.
5. Asociación con la GUI.
6. Ejecución de scripts de casos de prueba y generación de reporte de pruebas.

3.1. Especificación de requisitos.

La primera tarea es la definición de los requisitos, que normalmente implica la intervención de los ingenieros de requisitos, las partes interesadas y, finalmente, los evaluadores.

3.2. Especificación de casos de prueba.

Las pruebas de casos de uso se derivan de los diversos flujos de proceso expresados por un caso de uso RSL. Cada prueba contiene múltiples escenarios de prueba.

Un escenario de prueba abarca un grupo de pasos de prueba y debe ser ejecutado por un actor. La construcción UseCaseTest comienza definiendo el conjunto de prueba, incluido el ID, el nombre y el tipo de caso de uso. Luego engloba las referencias claves [Use Case] indicando el Use Case en el que se está realizando la prueba y [DataEntity] haciendo referencia a una posible entidad de datos que se gestiona.

3.3. Refinación de casos de prueba por el evaluador.

Los casos de prueba generados pueden estar sujetos a refinamientos, dando como resultado otros casos de prueba. Las DataEntities y las Variables temporales son fundamentales para la transmisión de datos entre los TestSteps involucrados en la prueba y se definen dentro de TestScenarios. Los valores de DataEntities y Variables se definen en un formato tabular con varias filas. De esta forma, cuando un atributo está asociado a

N valores, significa que este escenario puede ejecutarse N veces, una vez por cada valor de la tabla. Cada escenario también está formado por TestSteps. Un TestStep puede tener cuatro tipos (Actor PrepareData, Actor CallSystem, System Execute, System ReturnResult) y varias OperationExtensions. Es en los casos de prueba que se introducen los conceptos fundamentales para la generación de scripts de prueba, que incluyen escenarios de prueba y pasos de prueba.

3.4. Elaboración de scripts de prueba.

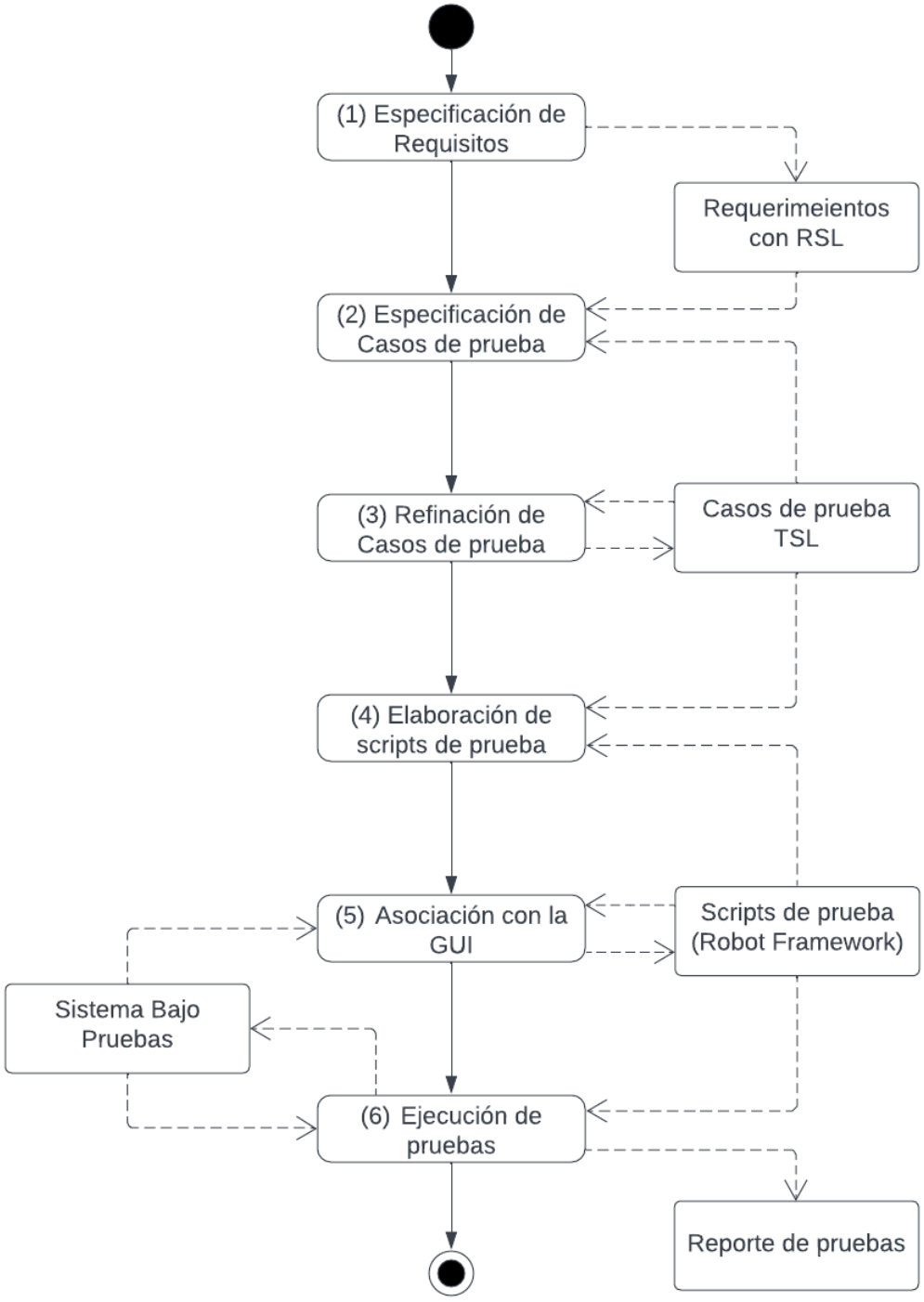
Una vez completada la especificación, sigue la generación de los scripts de prueba para la herramienta Robot. Este proceso de generación se basa en las relaciones que se establecen entre la especificación RSL y la sintaxis de Robot Framework.

3.5. Asociación con la GUI.

En esta etapa, es necesario completar los scripts de prueba generados en la fase anterior con los localizadores utilizados para seleccionar los elementos de GUI de destino.

3.6. Ejecución de scripts de casos de prueba y generación de reporte de pruebas.

Se ejecutan las pruebas y se muestran los resultados de las pruebas.


(1) Especificación de Requisitos
Requerimeientos con RSL
(2) Especificación de Casos de prueba
(3) Refinación de Casos de prueba
Casos de prueba TSL
(4) Elaboración de scripts de prueba
(5) Asociación con la GUI
Scripts de prueba (Robot Framework)
Sistema Bajo Pruebas
(6) Ejecución de pruebas
Reporte de pruebas

4. Ejecución.

4.1. Paso 1: Especificación de requisitos.

RQNF-1: Cuando la cámara no se active se envía un mensaje al servidor indicando que la cámara no fue activada.

RQF-1: Como usuario debo ver la etiqueta "Desconocido" cuando no sea reconocido.

RQF-2: Como usuario debo ver la etiqueta con mi nombre cuando sea reconocido.

RQF-3: Como usuario debo ver la etiqueta "Desconocido" cuando no sea reconocido, además el sistema debe permitirme presionar la tecla "escape" para terminar el reconocimiento.

RQF-4: Como usuario debo ver la etiqueta con mi nombre cuando sea reconocido, además el sistema debe permitirme presionar la tecla "escape" para terminar el reconocimiento.

Historia de Usuario	
Número: 1	**Usuario:** Cliente
Nombre historia:	
Prioridad en negocio: Alta	Riesgo en el desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 6
Programador responsable: Willy Romero	
Descripción: Cuando la cámara no se active se envía un mensaje al servidor indicando que la cámara no fue activada.	
Validación: Cuando la cámara del cliente no se activa, se envía un mensaje al servidor notificando que la cámara del cliente no fue iniciada.	

Historia de Usuario	
Número: 2	**Usuario:** Cliente
Nombre historia:	
Prioridad en negocio: Alta	Riesgo en el desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 6
Programador responsable: Willy Romero	
Descripción: Como usuario debo ver la etiqueta "Desconocido" cuando no sea reconocido.	
Validación: Cuando el usuario no es reconocido (el valor probabilístico de diferencia es del 70% o mayor) se muestra la etiqueta “Desconocido” sobre el rostro del usuario,	

Historia de Usuario	
Número: 3	**Usuario:** Cliente
Nombre historia:	
Prioridad en negocio: Alta	Riesgo en el desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 6
Programador responsable: Willy Romero	
Descripción: Como usuario debo ver la etiqueta con mi nombre cuando sea reconocido.	
Validación: Cuando el usuario es reconocido (el valor probabilístico de diferencia está por debajo del 70%) se muestra la etiqueta con el nombre del usuario sobre su rostro.	

Historia de Usuario	
Número: 4	**Usuario:** Cliente
Nombre historia:	
Prioridad en negocio: Alta	Riesgo en el desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 6
Programador responsable: Willy Romero	
Descripción: Como usuario debo ver la etiqueta "Desconocido" cuando no sea reconocido, además el sistema debe permitirme presionar la tecla "escape" para terminar el reconocimiento.	
Validación: Cuando el usuario no es reconocido (el valor probabilístico de diferencia es del 70% o mayor) se muestra la etiqueta “Desconocido” sobre el rostro del usuario, además de esto el usuario debe ser capaz de salir de la interfaz de reconocimiento presionando la tecla “escape”.	

Historia de Usuario	
Número: 3	**Usuario:** Cliente
Nombre historia:	
Prioridad en negocio: Alta	Riesgo en el desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 6
Programador responsable: Willy Romero	
Descripción: Como usuario debo ver la etiqueta con mi nombre cuando sea reconocido, además el sistema debe permitirme presionar la tecla "escape" para terminar el reconocimiento.	
Validación: Cuando el usuario es reconocido (el valor probabilístico de diferencia está por debajo del 70%) se muestra la etiqueta con el nombre del usuario sobre su rostro, además de esto el usuario debe ser capaz de salir de la interfaz de reconocimiento presionando la tecla “escape”.	

4.2. Paso 2: Especificación de casos de prueba.

Test Suite
Test Case No Camera Started "No se inició la cámara"
Test Case Unrecognized Face "Usuario no reconocido"
Test Case Recognized Face "Usuario reconocido"
Test Case Unrecognized Face Key "Usuario no reconocido, usa escape"
Test Case Recognized Face Key "Usuario reconocido, usa escape"

Test Case
Test Case No Camera Started "No se inició la cámara"
variable probability=00, name=willy
"Se intenta encender la cámara"
"La cámara no enciende"
"Se envía mensaje al servidor"

Test Case
Test Case Unrecognized Face "Usuario no reconocido"
variable probability=70, name=willy
"Se intenta encender la cámara"
"La cámara enciende"
"Se busca el modelo matemático"
"Se intenta reconocer al usuario"
"Evaluar probabilidad"
"Evaluar probabilidad (70<70)"
"Muestra etiqueta Desconocido"
"Se envía mensaje al servidor (No es la persona registrada)"
"Se detiene el reconocimiento"

Test Case
Test Case Unrecognized Face "Usuario no reconocido"
variable probability=60, name=willy
"Se intenta encender la cámara"
"La cámara enciende"
"Se busca el modelo matemático"
"Se intenta reconocer al usuario"
"Evaluar probabilidad"
"Evaluar probabilidad (60<70)"
"Muestra etiqueta con el nombre del usuario"
"Se envía mensaje al servidor (Persona detectada)"
"Se detiene el reconocimiento"

Test Case
Test Case Unrecognized Face "Usuario no reconocido"
variable probability=70, name=willy
"Se intenta encender la cámara"
"La cámara enciende"
"Se busca el modelo matemático"
"Se intenta reconocer al usuario"
"Evaluar probabilidad"
"Evaluar probabilidad (70<70)"
"Muestra etiqueta Desconocido"
"Se envía mensaje al servidor (No es la persona registrada)"
"El usuario presiona la tecla escape (detiene el reconocimiento)"
"Se detiene el reconocimiento"

Test Case
Test Case Unrecognized Face "Usuario no reconocido"
variable probability=60, name=willy
"Se intenta encender la cámara"
"La cámara enciende"
"Se busca el modelo matemático"
"Se intenta reconocer al usuario"
"Evaluar probabilidad"
"Evaluar probabilidad (60<70)"
"Muestra etiqueta con el nombre del usuario"
"Se envía mensaje al servidor (Persona detectada)"
"El usuario presiona la tecla escape (detiene el reconocimiento)"
"Se detiene el reconocimiento"

4.3. Paso 3: Refinación de casos de prueba por el evaluador.

```
*** Test Suite ***
Test Case  No Camera Started              “No se inició la cámara”                  [
           variable              probability              name
]
Test Case  Unrecognized Face              “Usuario no reconocido”                   [
           variable              probability              name
]
Test Case  Recognized Face                “Usuario reconocido”                      [
           variable              probability              name
]
Test Case  Unrecognized Face Key          “Usuario no reconocido, usa escape”       [
           variable              probability              name
]
Test Case  Recognized Face Key            “Usuario reconocido, usa escape”          [
           variable              probability              name
]
```

```
*** Test Case ***
Test Case  No Camera Started              “No se inició la cámara”                  [
           variable    probability withData [00]          name withData [“willy”]
           step s1 (System:Execute:Turn On Camera)              “Se intenta encender la cámara”
           step s2 (System:Return Result: Turn On Camera: Fail) “La cámara no enciende”
           step s3 (App:Execute:Send Message To Server)         “Se envía mensaje al servidor”
]
```

```
*** Test Case ***
Test Case  Unrecognized Face              “Usuario no reconocido”                   [
           variable    probability withData [70]          name withData [“willy”]
           step s1 (System:Execute:Turn On Camera)              “Se intenta encender la cámara”
           step s2 (System:Return Result: Success)              “La cámara enciende”
           step s3 (App:Execute:Search Model)                   “Se busca el modelo matemático”
           step s4 (App:Execute:Recognize:Start)                “Se intenta reconocer al usuario”
           step s5 (App:Execute:Evaluate Probability)           “Evaluar probabilidad”
           step s6 (App:Execute:Evaluate Probability:Fail)      “Evaluar probabilidad (70<70)”
           step s7 (App:Execute:Show Label: Desconocido)        “Muestra etiqueta Desconocido”
           step s8 (App:Execute:Send Message To Server)         “Se envía mensaje al servidor (No es la persona registrada)”
           step s9 (App:Execute:Recognize:End)                  “Se detiene el reconocimiento”
]
```

```
*** Test Case ***
Test Case  Unrecognized Face              “Usuario no reconocido”                   [
           variable    probability withData [60]          name withData [“willy”]
           step s1 (System:Execute:Turn On Camera)              “Se intenta encender la cámara”
           step s2 (System:Return Result: Success)              “La cámara enciende”
           step s3 (App:Execute:Search Model)                   “Se busca el modelo matemático”
           step s4 (App:Execute:Recognize:Start)                “Se intenta reconocer al usuario”
           step s5 (App:Execute:Evaluate Probability)           “Evaluar probabilidad”
           step s6 (App:Execute:Evaluate Probability:Success)   “Evaluar probabilidad (60<70)”
           step s7 (App:Execute:Show Label: Desconocido)        “Muestra etiqueta con el nombre del usuario”
           step s8 (App:Execute:Send Message To Server)         “Se envía mensaje al servidor (Persona detectada)”
           step s9 (App:Execute:Recognize:End)                  “Se detiene el reconocimiento”
]
```

```
*** Test Case ***
Test Case  Unrecognized Face              "Usuario no reconocido"                [
           variable    probability withData [70]        name withData ["willy"]
           step s1 (System:Execute:Turn On Camera)              "Se intenta encender la cámara"
           step s2 (System:Return Result: Success)              "La cámara enciende"
           step s3 (App:Execute:Search Model)                   "Se busca el modelo matemático"
           step s4 (App:Execute:Recognize:Start)                "Se intenta reconocer al usuario"
           step s5 (App:Execute:Evaluate Probability)           "Evaluar probabilidad"
           step s6 (App:Execute:Evaluate Probability:Fail)      "Evaluar probabilidad (70<70)"
           step s7 (App:Execute:Show Label: Desconocido)        "Muestra etiqueta Desconocido"
           step s8 (App:Execute:Send Message To Server)         "Se envía mensaje al servidor (No es la persona registrada)"
           step s9 (App:Execute:Press Key:Scape)                "El usuario presiona la tecla escape (detiene el reconocimiento)"
           step s10 (App:Execute:Recognize:End)                 "Se detiene el reconocimiento"
]
```

```
*** Test Case ***
Test Case  Unrecognized Face              "Usuario no reconocido"                [
           variable    probability withData [60]        name withData ["willy"]
           step s1 (System:Execute:Turn On Camera)              "Se intenta encender la cámara"
           step s2 (System:Return Result: Success)              "La cámara enciende"
           step s3 (App:Execute:Search Model)                   "Se busca el modelo matemático"
           step s4 (App:Execute:Recognize:Start)                "Se intenta reconocer al usuario"
           step s5 (App:Execute:Evaluate Probability)           "Evaluar probabilidad"
           step s6 (App:Execute:Evaluate Probability:Success)   "Evaluar probabilidad (60<70)"
           step s7 (App:Execute:Show Label: Desconocido)        "Muestra etiqueta con el nombre del usuario"
           step s8 (App:Execute:Send Message To Server)         "Se envía mensaje al servidor (Persona detectada)"
           step s9 (App:Execute:Press Key:Scape)                "El usuario presiona la tecla escape (detiene el reconocimiento)"
           step s10 (App:Execute:Recognize:End)                 "Se detiene el reconocimiento"
]
```

4.4. Paso 4: Elaboración de scripts de prueba.

Se crean funciones que permitan acceder a la funcionalidad, teniendo una clara ocurrencia enmarcada en cada uno de los casos de prueba.

```
import cv2
import recognize.src.acceptancetest.recognizeFaceTest as rf1
import recognize.src.acceptancetest.recognizeFaceTest2 as rf2
import recognize.src.acceptancetest.recognizeFaceTest3 as rf3
import socketio

socket = socketio.Client()

def test_case_1(probability, name):
    cap = cv2.VideoCapture()
    faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
    socket.connect("http://localhost:3000/")
    response = rf1.recognize(int(probability), cap, faceClassif, socket, name)
    socket.disconnect()
    return response

def test_case_2(probability, name):
    cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
    faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
    socket.connect("http://localhost:3000/")
    response = rf2.recognize(int(probability), cap, faceClassif, socket, name)
    socket.disconnect()
    return response

def test_case_3(probability, name):
    cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
    faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
    socket.connect("http://localhost:3000/")
    response = rf2.recognize(int(probability), cap, faceClassif, socket, name)
    socket.disconnect()
    return response


def test_case_4(probability, name):
    cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
    faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
    socket.connect("http://localhost:3000/")
    response = rf3.recognize(int(probability), cap, faceClassif, socket, name)
    socket.disconnect()
    return response


def test_case_5(probability, name):
    cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
    faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
    socket.connect("http://localhost:3000/")
    response = rf3.recognize(int(probability), cap, faceClassif, socket, name)
    socket.disconnect()
    return response
```

Se define una librería de pruebas, esta actúa como middleware entre las funciones de los casos de prueba y el script de robot. La nominación de las funciones es importante.

```
import robot.api.logger
import robot.utils.asserts
import sys
sys.path.append('../')
import test_cases as ft

_test_case_response = {'rtc1': None, 'rtc2': None, 'rtc3': None, 'rtc4': None, 'rtc5': None}

def camera_no_started(probability, name):
    robot.api.logger.info('Test case 1 tested with probability of %f' % float(probability))
    _test_case_response['rtc1'] =ft.test_case_1(probability, name)

def check_camera_no_started(expected):
    _actual = _test_case_response['rtc1']
    robot.utils.asserts.assert_equal(expected, _actual)

def unrecognized_face(probability, name):
    robot.api.logger.info('Test case 2 tested with probability of %f' % float(probability))
    _test_case_response['rtc2'] =ft.test_case_2(probability, name)
```

```
def check_unrecognized_face(expected):
    _actual = _test_case_response['rtc2']
    robot.utils.asserts.assert_equal(expected, _actual)

def recognized_face(probability, name):
    robot.api.logger.info('Test case 3 tested with probability of %f' % float(probability))
    _test_case_response['rtc3'] =ft.test_case_3(probability, name)

def check_recognized_face(expected):
    _actual = _test_case_response['rtc3']
    robot.utils.asserts.assert_equal(expected, _actual)

def unrecognized_face_key(probability, name):
    robot.api.logger.info('Test case 4 tested with probability of %f' % float(probability))
    _test_case_response['rtc4'] =ft.test_case_4(probability, name)

def check_unrecognized_face_key(expected):
    _actual = _test_case_response['rtc4']
    robot.utils.asserts.assert_equal(expected, _actual)

def recognized_face_key(probability, name):
    robot.api.logger.info('Test case 5 tested with probability of %f' % float(probability))
    _test_case_response['rtc5'] =ft.test_case_5(probability, name)

def check_recognized_face_key(expected):
    _actual = _test_case_response['rtc5']
    robot.utils.asserts.assert_equal(expected, _actual)
```

Se desarrolla el script en robot que haga uso del middleware para así acceder a cada uno de los casos de prueba, con los cuales se valide el cumplimiento de lo especificado en los requisitos cumpliendo así las pruebas de aceptación.

```
| *** Settings ***    |
| Documentation       | This is test suite for acceptance test of functionality 3.2. Recognize
| Metadata            | Version | 1.0.0
| Metadata            | More Info | You can acces to the source code here: https://github.com/willyromero/acceptancetest-
robotframework
| Metadata            | Executed At | ${HOST}
| Force Tags          | Functionality 3.2 Recognize
| Default Tags        | owner-willy_romero | 6668
| Resource            | Keywords.resource
| Library             | String
| Library             | ../TestLibrary/RecognizeLibrary.py

| *** Variables ***        |
| &{DIC_PARAMETERS_TC1}    | probability=00 | name=willy
| &{DIC_PARAMETERS_TC2}    | probability=70 | name=willy
| &{DIC_PARAMETERS_TC3}    | probability=60 | name=willy
| &{DIC_PARAMETERS_TC4}    | probability=70 | name=willy
| &{DIC_PARAMETERS_TC5}    | probability=60 | name=willy

| ${RQ1 F 3 2}             | Cuando la cámara no se active se envía un mensaje al servidor indicando que la cámara no fue
activada.
| ${RQ2 F 3 2}             | Como usuario debo ver la etiqueta “Desconocido" cuando no sea reconocido.
| ${RQ3 F 3 2}             | Como usuario debo ver la etiqueta con mi nombre cuando sea reconocido.
| ${RQ4 F 3 2}             | Como usuario debo ver la etiqueta “Desconocido" cuando no sea reconocido, además el sistema debe
permitirme presionar la tecla "escape" para terminar el reconocimiento.
| ${RQ5 F 3 2}             | Como usuario debo ver la etiqueta con mi nombre cuando sea reconocido, además el sistema debe
permitirme presionar la tecla "escape" para terminar el reconocimiento.

| ${HOST}                  | Local.

| *** Test Cases ***       |
| No Camera Started        | [Documentation]       | ${RQ1 F 3 2}
|                          | Verify Test Case 1    | ${DIC_PARAMETERS_TC1}[probability] | ${DIC_PARAMETERS_TC1}[name]

| Unrecognized Face        | [Documentation]       | ${RQ2 F 3 2}
|                          | Verify Test Case 2    | ${DIC_PARAMETERS_TC2}[probability] | ${DIC_PARAMETERS_TC2}[name]

| Recognized Face          | [Documentation]       | ${RQ3 F 3 2}
|                          | Verify Test Case 3    | ${DIC_PARAMETERS_TC3}[probability] | ${DIC_PARAMETERS_TC3}[name]

| Unrecognized Face Key    | [Documentation]       | ${RQ4 F 3 2}
|                          | Verify Test Case 4    | ${DIC_PARAMETERS_TC4}[probability] | ${DIC_PARAMETERS_TC4}[name]

| Recognized Face Key      | [Documentation]       | ${RQ5 F 3 2}
|                          | Verify Test Case 5    | ${DIC_PARAMETERS_TC5}[probability] | ${DIC_PARAMETERS_TC5}[name]

| *** Keywords ***      |
| Verify Test Case 1    | [Documentation]               | Camera of device was not started.
|                       | [Arguments]                   | ${probability} | ${name}
|                       | Camera No Started             | ${probability} | ${name}
|                       | Check Camera No Started       | cp1

| Verify Test Case 2    | [Documentation]               | The person scanned was unrecognized.
|                       | [Arguments]                   | ${probability} | ${name}
|                       | Unrecognized Face             | ${probability} | ${name}
```

```
|                       | Check Unrecognized Face          | cp2

| Verify Test Case 3    | [Documentation]                  | The person scanned was recognized.
|                       | [Arguments]                      | ${probability} | ${name}
|                       | Recognized Face                  | ${probability} | ${name}
|                       | Check Recognized Face            | cp3

| Verify Test Case 4    | [Documentation]                  | The person scanned was unrecognized and press key scape.
|                       | [Arguments]                      | ${probability} | ${name}
|                       | Unrecognized Face Key            | ${probability} | ${name}
|                       | Check Unrecognized Face Key      | cp4

| Verify Test Case 5    | [Documentation]                  | The person scanned was recognized and press key scape.
|                       | [Arguments]                      | ${probability} | ${name}
|                       | Recognized Face Key              | ${probability} | ${name}
|                       | Check Recognized Face Key        | cp5
```

4.5. Paso 5: Asociación con la GUI.

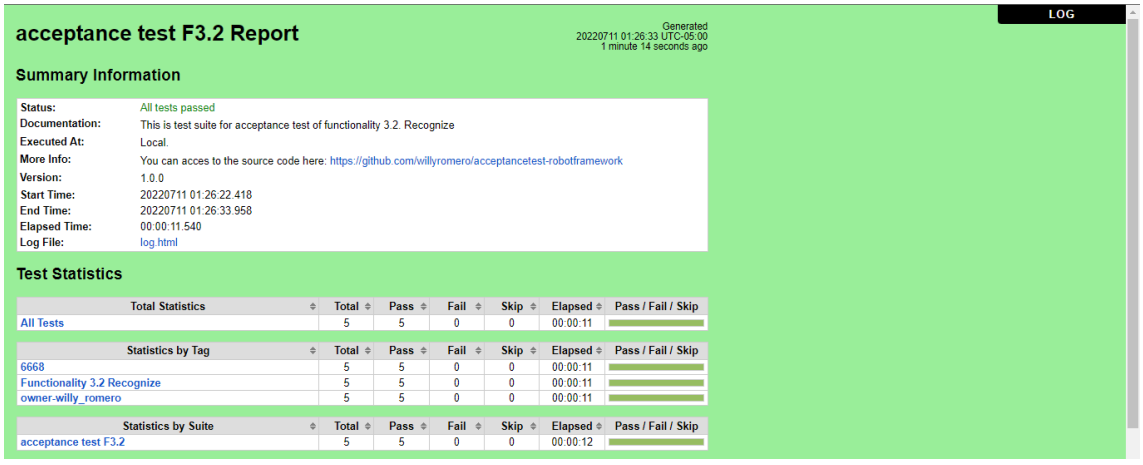
La funcionalidad en la cual se desarrolla el reconocimiento de los usuarios no tiene Interfaz de Usuario Gráfica, por tanto, se omite este paso

4.6. Paso 6: Ejecución de scripts de casos de prueba y generación de reporte de pruebas.

La ejecución de las pruebas se realiza satisfactoriamente.

```
PS D:\aceptance-test\atrfrecognize\robot> robot .\acceptance_test_F3.2.robot
==============================================================================
acceptance test F3.2 :: This is test suite for acceptance test of functiona...
==============================================================================
No Camera Started :: Cuando la cámara no se active se envía un men... | PASS |
------------------------------------------------------------------------------
Unrecognized Face :: Como usuario debo ver la etiqueta "Desconocid... | PASS |
------------------------------------------------------------------------------
Recognized Face :: Como usuario debo ver la etiqueta con mi nombre... | PASS |
------------------------------------------------------------------------------
Unrecognized Face Key :: Como usuario debo ver la etiqueta "Descon... | PASS |
------------------------------------------------------------------------------
Recognized Face Key :: Como usuario debo ver la etiqueta con mi no... | PASS |
------------------------------------------------------------------------------
acceptance test F3.2 :: This is test suite for acceptance test of ... | PASS |
5 tests, 5 passed, 0 failed
==============================================================================
Output:  D:\aceptance-test\atrfrecognize\robot\output.xml
Log:     D:\aceptance-test\atrfrecognize\robot\log.html
Report:  D:\aceptance-test\atrfrecognize\robot\report.html
```

Los reportes son:



LOG

acceptance test F3.2 Report

Generated
20220711 01:26:33 UTC-05:00
1 minute 14 seconds ago

Summary Information

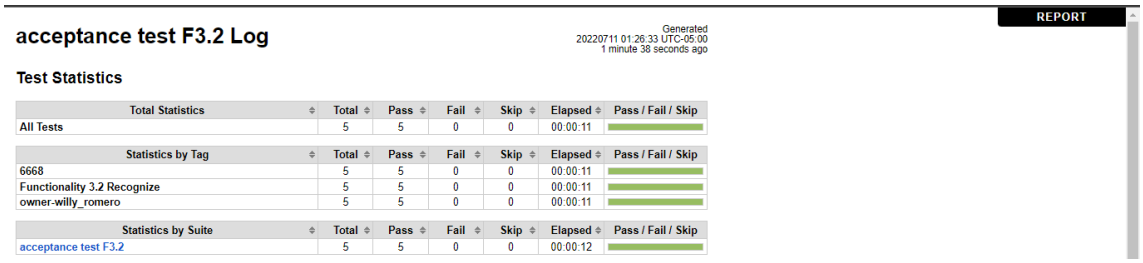
Status:	All tests passed
Documentation:	This is test suite for acceptance test of functionality 3.2. Recognize
Executed At:	Local
More Info:	You can acces to the source code here: https://github.com/willyromero/acceptancetest-robotframework
Version:	1.0.0
Start Time:	20220711 01:26:22.418
End Time:	20220711 01:26:33.958
Elapsed Time:	00:00:11.540
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	5	5	0	0	00:00:11	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
6668	5	5	0	0	00:00:11	
Functionality 3.2 Recognize	5	5	0	0	00:00:11	
owner-willy_romero	5	5	0	0	00:00:11	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
acceptance test F3.2	5	5	0	0	00:00:12	



REPORT

acceptance test F3.2 Log

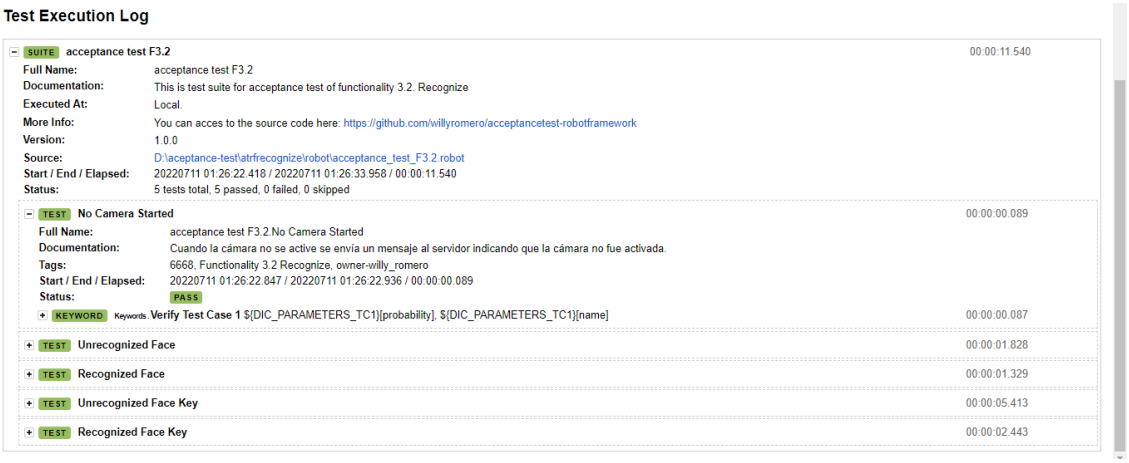
Generated
20220711 01:26:33 UTC-05:00
1 minute 38 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	5	5	0	0	00:00:11	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
6668	5	5	0	0	00:00:11	
Functionality 3.2 Recognize	5	5	0	0	00:00:11	
owner-willy_romero	5	5	0	0	00:00:11	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
acceptance test F3.2	5	5	0	0	00:00:12	


Test Execution Log
SUITE acceptance test F3.2 00:00:11.540
Full Name: acceptance test F3.2
Documentation: This is test suite for acceptance test of functionality 3.2. Recognize
Executed At: Local.
More Info: You can acces to the source code here: https://github.com/willyromero/acceptancetest-robotframework
Version: 1.0.0
Source: D:\aceptance-test\atrfrecognize\robot\acceptance_test_F3.2.robot
Start / End / Elapsed: 20220711 01:26:22.418 / 20220711 01:26:33.958 / 00:00:11.540
Status: 5 tests total, 5 passed, 0 failed, 0 skipped
TEST No Camera Started 00:00:00.089
Full Name: acceptance test F3.2.No Camera Started
Documentation: Cuando la cámara no se active se envía un mensaje al servidor indicando que la cámara no fue activada.
Tags: 6668, Functionality 3.2 Recognize, owner-willy_romero
Start / End / Elapsed: 20220711 01:26:22.847 / 20220711 01:26:22.936 / 00:00:00.089
Status: PASS
KEYWORD Keywords. Verify Test Case 1 ${DIC_PARAMETERS_TC1}[probability], ${DIC_PARAMETERS_TC1}[name] 00:00:00.087
TEST Unrecognized Face 00:00:01.828
TEST Recognized Face 00:00:01.329
TEST Unrecognized Face Key 00:00:05.413
TEST Recognized Face Key 00:00:02.443