**Pruebas de calidad**

Jorman Patiño

Maicol Zapata

Edwin Perez

**SENA**

Ficha : 1828182

**Tabla de Contenidos**

[1.](#bookmark=id.1r7znmsd5us3) Descripción 3

[2.](#bookmark=id.104f0mlj6tij) Prueba caja blanca del módulo Donar 3

[2.1](#bookmark=id.z0mxr7gvq8us) Descripción 3

[2.2](#bookmark=id.qame5c3ecbgc) Condiciones de ejecución 3

[2.3](#bookmark=id.298bkk23q4ud) Entrada 4

[2.4](#bookmark=id.c4qzxlbtnzxu) Resultado esperado 4

[2.5](#bookmark=id.5f9yv31qwqgi) Evaluación de la Prueba 4

[3.](#bookmark=id.3ktk3xmsut1t) Prueba de caja negra del módulo Donar 4

[3.1](#bookmark=id.voukcwiw2v60) Descripción 4

[3.2](#bookmark=id.glbs1e1aqrdf) Condiciones de ejecución 4

[3.3](#bookmark=id.kypcxshf6xmz) Entrada 5

[3.4](#bookmark=id.5nyhfy4oqshk) Resultado esperado 5

[3.5](#bookmark=id.rkyi6bwynsrl) Evaluación de la Prueba 5

[4.](#bookmark=id.c1uvx9nkyfbh) Prueba automatizada del módulo Donar 5

[4.1](#bookmark=id.edp7socmlft3) Descripción 5

[4.2](#bookmark=id.lf8pir9ljqy) Condiciones de ejecución 5

[4.3](#bookmark=id.edow0idpqgwx) Script 5

[4.4](#bookmark=id.r8y6xb2fu5lr) Resultado esperado 6

[4.5](#bookmark=id.i5m47k7q8nnn) Evaluación de la Prueba 6

5[.](#bookmark=id.npe5kxhyg35i) Prueba de caja negra del módulo Iniciar Sesión 6

5[.1](#bookmark=id.8h02stutnj3e) Descripción 6

5[.2](#bookmark=id.ru7een1ejxvi) Condiciones de ejecución 6

5[.3](#bookmark=id.2s2vl7vw64pu) Entrada 6

5[.4](#bookmark=id.uzff8xv0b62o) Resultado esperado 7

5[.5](#bookmark=id.859a3r7jxv5v) Evaluación de la Prueba 7

6[.](#bookmark=id.qwyln8n73e83) Prueba automatizada del módulo Iniciar Sesión 7

6[.1](#bookmark=id.atfzmogylszo) Descripción 7

6[.2](#bookmark=id.ru7een1ejxvi) Condiciones de ejecución 7

6[.3](#bookmark=id.u6fkxfijk37u) Script 7

6[.4](#bookmark=id.ulu95qc4edq) Resultado esperado 8

6[.5](#bookmark=id.jjjbbm8ptdgb) Evaluación de la Prueba 8

1. **Descripción**

En este documento vamos a poder apreciar el proceso de realización de pruebas de calidad del proyecto Coink.

Evaluaremos la calidad de los módulos principales del proyecto COINK utilizando diferentes pruebas manuales (Pruebas de caja blanca, pruebas de caja negra ) y pruebas automatizadas (Utilizando Katalon).

Las pruebas realizadas son:

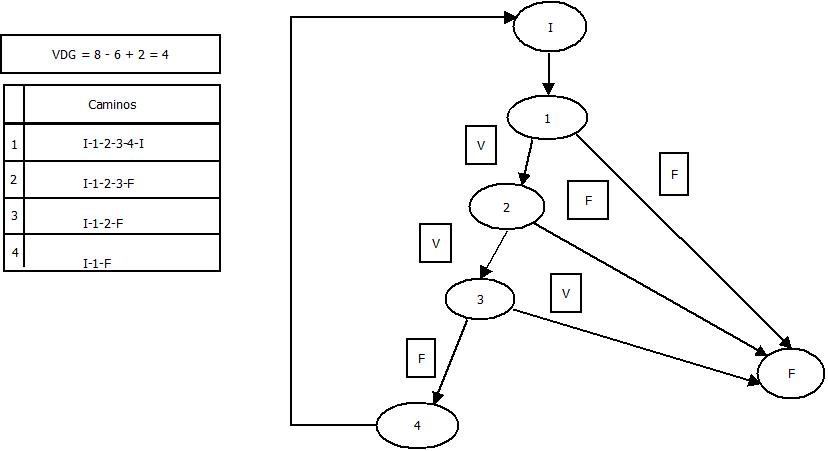
* Prueba de caja blanca del módulo donar.
* Prueba de caja negra del módulo donar.
* Prueba automatizada del módulo donar.
* Prueba de caja negra del módulo Iniciar Sesión.
* Prueba automatizada del módulo Iniciar Sesión.
* Prueba de caja blanca del módulo reportar.
* Prueba de caja negra del módulo reportar.
* Prueba automatizada del modulo reportar.

El entorno del cual partiremos para realizar la prueba será el formulario de entrada de la aplicación.

1. **Prueba de caja blanca del módulo Donar.**

* 1. **Descripción**

Comprobaremos que la lógica al momento de donar este correcta.



* 1. **Condiciones de ejecución**

Las condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba:

-Se debe iniciar sesión.

-Se debe de encontrar un proyecto con estado activo.

* 1. **Entrada**

Descripción paso a paso de la ejecución del caso de prueba.

* I : Inicio del algoritmo.
* 1: Confirma si hay una sesión activa.
* 2: Verifica que el valor a donar sea mayor a 2000.
* 3: Verifica que el usuario tenga saldo.
* 4: Le pide al usuario que recargue.
* F: Final del algoritmo.

* 1. **Resultado esperado**

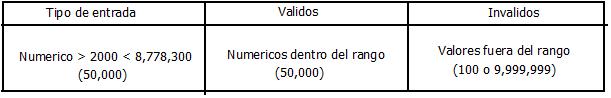
Se espera que el sistema recorra todos los nodos sin excepción dependiendo del camino que se tome.

* 1. **Evaluación de la Prueba**

La prueba se realizó con satisfacción, mostrando los resultados esperados.

1. **Prueba de caja Negra del módulo Donar.**
   1. **Descripción**

Comprobaremos la reacción del módulo a diferentes tipos de datos de entrada.

**

* 1. **Condiciones de ejecución**

condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba:

-Se debe iniciar sesión.

-Se debe de encontrar un proyecto con estado activo.

* 1. **Entrada**

Descripción paso a paso de la ejecución del caso de prueba.

El sistema pide un dato numérico mayor a 2000 y menor a 8,778,300, ingresamos los siguientes datos para ver la reacción del sistema:

* 50,000.
* 100.
* 9,999,999.
* Hola.

* 1. **Resultado esperado**

Se espera que el sistema no acepte nada que no esté dentro del rango que exige, Y que muestre mensajes según el resultado.

* 1. **Evaluación de la Prueba**

La prueba se realizó con satisfacción, mostrando los resultados esperados, aceptando los números dentro del rango y rechazando texto y números fuera del rango.

1. **Prueba automatizada del módulo Donar.**
   1. **Descripción**

Realizaremos el script de la prueba automatizada con la herramienta katalon y luego probaremos con diferentes datos para ver las reacciones del sistema

* 1. **Condiciones de ejecución**

condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba:

-Se debe iniciar sesión.

-Se debe de encontrar un proyecto con estado activo.

* 1. **Script**

import { browser, by, element, logging } from 'protractor';

describe('Donar', () => {

beforeAll(async () => { });

beforeEach(async () => { });

it('should do something', async () => {

await browser.get('http://localhost:8080/coinkgg10x/ver.php?pro=446');

await element(by.name("donar")).click();

await element(by.name("dinero")).click();

await element(by.name("dinero")).sendKeys('2000');

await element(by.xpath("//input[@name='donar']")).click();

});

afterEach(async () => {

// Assert that there are no errors emitted from the browser

const logs = await browser.manage().logs().get(logging.Type.BROWSER);

expect(logs).not.toContain(jasmine.objectContaining({

level: logging.Level.SEVERE,

}));

});

});

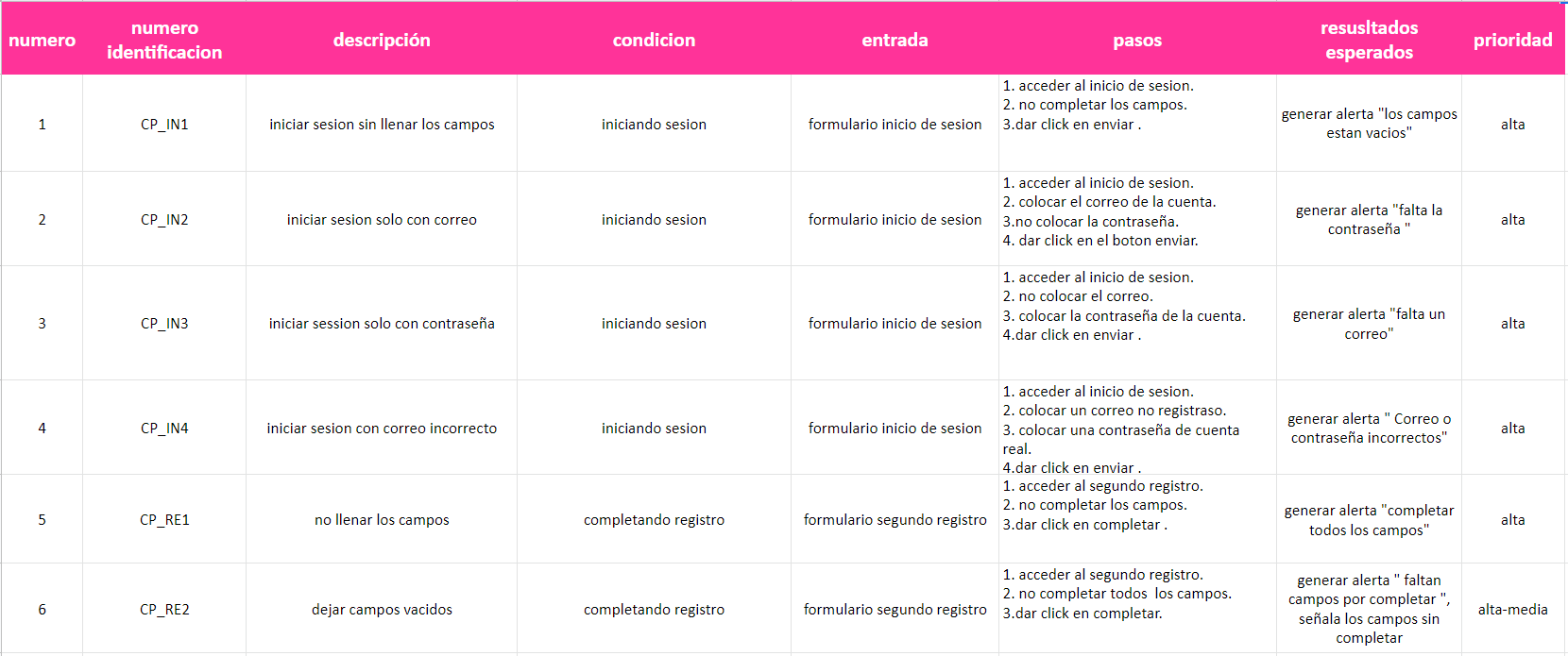
* 1. **Resultado esperado**

Se espera que el sistema no acepte nada que no esté dentro del rango que exige, Y que muestre mensajes según el resultado.

* 1. **Evaluación de la Prueba**

Los procesos se han ejecutado con éxito, obteniendo los resultados obtenidos y con la rapidez y facilidad que ofrecen las pruebas automatizadas

1. **Prueba de caja negra del módulo Iniciar Sesión.**
   1. **Descripción**
   2. Comprobaremos la reacción del módulo a diferentes tipos de datos de entrada.

**

* 1. **Condiciones de ejecución**

Condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba:

-No se requiere ninguna ejecución específica antes de iniciar sesión, esta acción se puede realizar desde cualquier página del proyecto.

* 1. **Entrada**

Descripción paso a paso de la ejecución del caso de prueba.

El sistema pide datos de tipo texto y tipo password mayores a 6 caracteres y menores a 16 caracteres, ingresamos los siguientes datos para ver la reacción del sistema:

* Datos vacíos.
* Solo ingresar un dato.
* Correo no registrado.
* Contraseña incorrecta.
* Datos con más de 16 caracteres.
* Datos con menos de 6 caracteres.

* 1. **Resultado esperado**

Se espera que el sistema no acepte datos sin contenido, que confirme que estén todos los datos necesarios,

que los datos no superen los 16 dígitos y menos de 6 dígitos, y para realizar la acción de iniciar sesion los

datos deben de estar registrados.

* 1. **Evaluación de la Prueba**

El sistema rechaza los campos vacíos y los datos no registrados, no deja enviar formularios incompletos, pero permite datos menores a 6 caracteres y mayores a 16 caracteres.

1. **Prueba automatizada del módulo Iniciar Sesión.**
   1. **Descripción**

Realizaremos el script de la prueba automatizada con la herramienta katalon y luego probaremos con diferentes datos para ver las reacciones del sistema.

* 1. **Condiciones de ejecución**

Condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba:

-No hay ninguna condición para realizar esta prueba ya que se puede iniciar sesion en cualquier parte del

sistema.

* 1. **Script**

import { browser, by, element, logging } from 'protractor';

describe('Login', () => {

beforeAll(async () => { });

beforeEach(async () => { });

it('should do something', async () => {

await browser.get('http://localhost:8080/coinkgg10x/index.php');

await element(by.linkText("Iniciar sesión")).click();

await element(by.id("inputEmail4")).click();

await element(by.id("inputEmail4")).sendKeys('admin');

await element(by.id("inputPassword4")).sendKeys('admin');

await element(by.name("iniciar")).click();

});

afterEach(async () => {

// Assert that there are no errors emitted from the browser

const logs = await browser.manage().logs().get(logging.Type.BROWSER);

expect(logs).not.toContain(jasmine.objectContaining({

level: logging.Level.SEVERE,

}));

});

});

* 1. **Resultado esperado**

Se espera que el sistema no acepte datos sin contenido, que confirme que estén todos los datos necesarios,

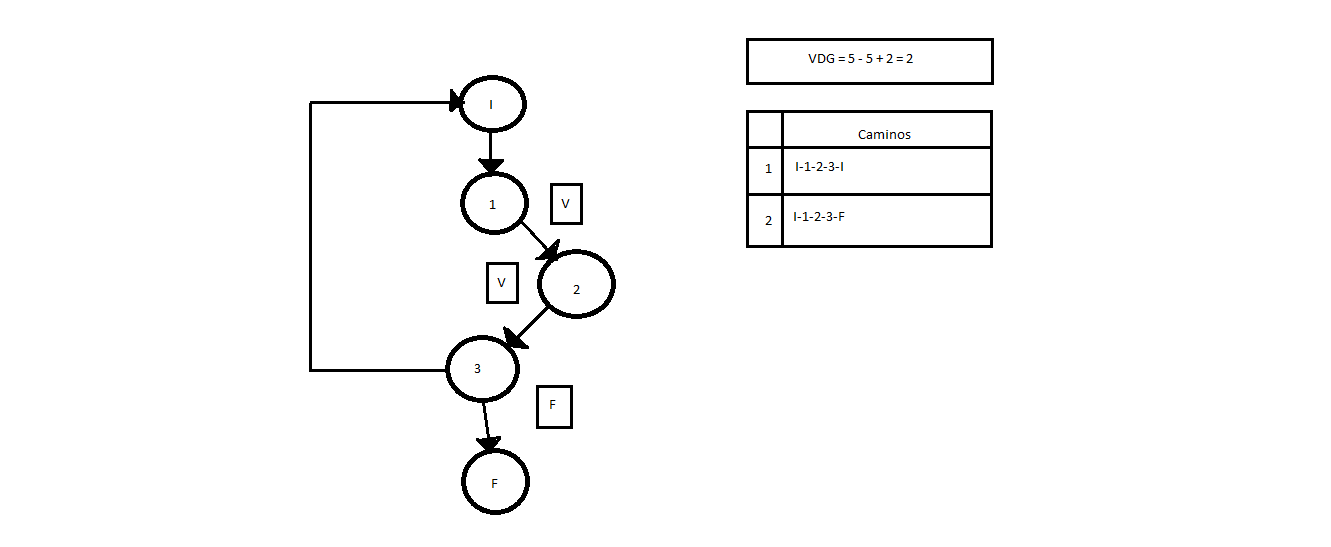
que los datos no superen los 16 dígitos y menos de 6 dígitos, y para realizar la acción de iniciar sesion los

datos deben de estar registrados.

* 1. **Evaluación de la Prueba**

El sistema rechaza los campos vacíos y los datos no registrados, no deja enviar formularios incompletos, pero permite datos menores a 6 caracteres y mayores a 16 caracteres.

1. **Prueba de caja blanca del módulo Reportar**

**7.1 Descripción**  
Comprobaremos que la lógica al momento de reportar este correcta.  
 ****

**7.2 Condiciones de ejecución**

Las condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba:

-Se debe iniciar sesión.

-Se debe encontrar un proyecto en estado activo.

**7.3 Entrada**

Descripción paso a paso de la ejecución del caso de prueba.

* I: Inicio del algoritmo.
* 1: Se confirma que existe un usuario.
* 2: Se trae el dato del usuario.
* 3: Se retorna el dato.
* F: Final del algoritmo.

**7.4 Resultado esperado**

Se espera que el sistema recorra todos los nodos sin excepción dependiendo del camino que se tome.

**7.5 Evaluación de la prueba**

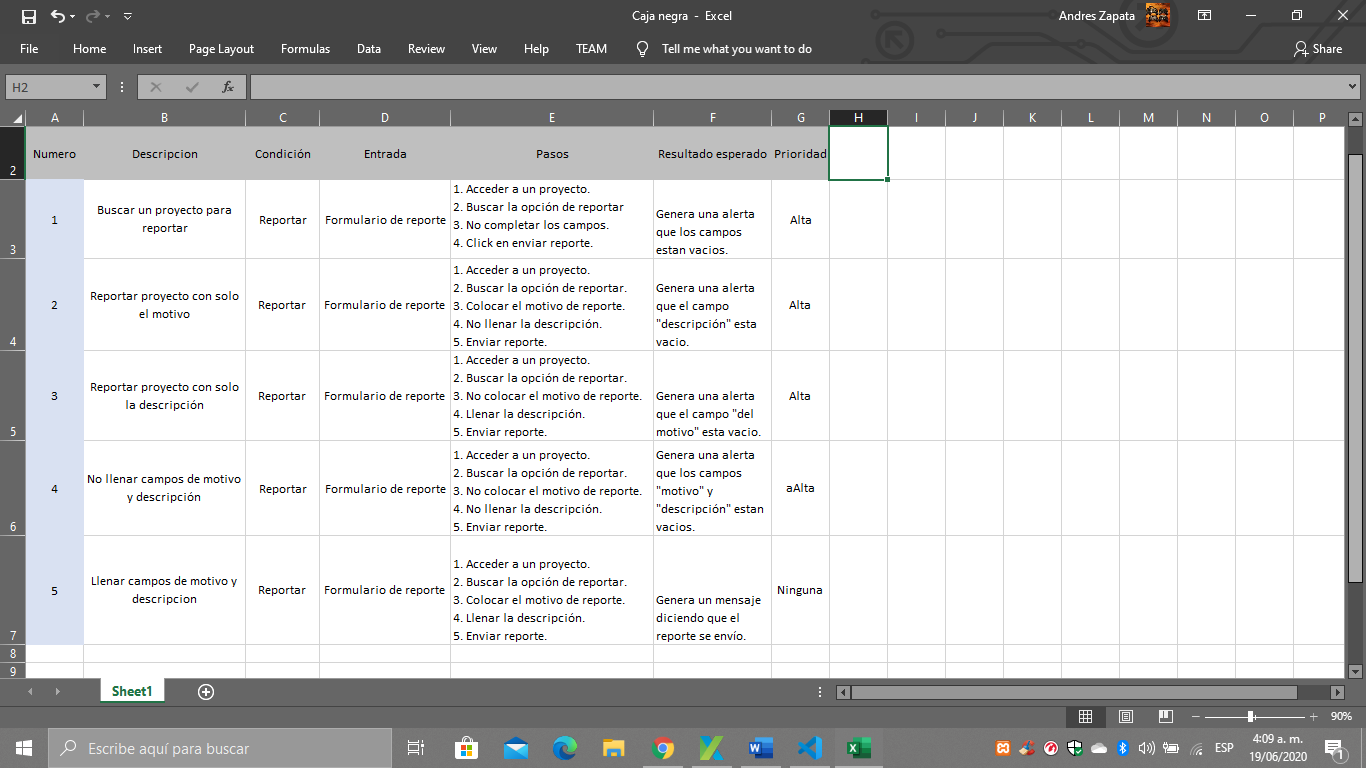
La prueba se realizó con satisfacción, mostrando los resultados esperados.

1. **Prueba de caja Negra del módulo Reportar.**

**8.1 Descripción**

Comprobaremos la reacción del módulo a diferentes tipos de datos de entrada.

**8.2**



**8.3 Condiciones de ejecución**

Condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba:

-Se debe iniciar sesión.

-Se debe de encontrar un proyecto con estado activo.

**8.4 Entrada**

Descripción paso a paso de la ejecución del caso de prueba.

El sistema pide datos de tipo selección múltiple y tipo texto mayores a 100 caracteres, ingresamos los datos para ver la reacción del sistema:

* Datos vacíos.
* Solo ingresar un dato.
* Tipo de motivo.
* Faltan campos por llenar.

**8.5 Resultado esperado**

Se espera que el sistema no acepte los campos sin contenido, que confirme que estén todos los datos necesarios, que los datos estén seleccionados de una manera adecuada, y para realizar la acción de reportar el usuario debe estar registrado.

**8.6 Evaluación de la prueba**

El sistema rechaza los campos vacíos, por lo tanto, no deja enviar formularios vacíos.

1. **Prueba automatizada del módulo Reportar.**
   1. **Descripción**

Realizaremos el script de la prueba automatizada con la herramienta katalon y luego probaremos con diferentes datos para ver las reacciones del sistema.

* 1. **Condiciones de ejecución**

Condiciones de ejecución que se deben cumplir antes de iniciar con el caso de prueba:

-No hay ninguna condición para realizar esta prueba ya que se puede iniciar sesión en cualquier parte del sistema.

* 1. **Script**

import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint

import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase

import static com.kms.katalon.core.testdata.TestDataFactory.findTestData

import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject

import static com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject

import com.kms.katalon.core.checkpoint.Checkpoint as Checkpoint

import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as CucumberKW

import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile

import com.kms.katalon.core.model.FailureHandling as FailureHandling

import com.kms.katalon.core.testcase.TestCase as TestCase

import com.kms.katalon.core.testdata.TestData as TestData

import com.kms.katalon.core.testobject.TestObject as TestObject

import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS

import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows

import internal.GlobalVariable as GlobalVariable

import org.openqa.selenium.Keys as Keys

WebUI.openBrowser('')

WebUI.navigateToUrl('http://localhost/Coink-1828182/ver.php?pro=454')

WebUI.click(findTestObject('Object Repository/Page\_Coink Colombia Viva/a\_Iniciar sesin'))

WebUI.setText(findTestObject('Object Repository/Page\_Coink Colombia Viva/input\_Correo\_correo'), 'maicol@gmail.com')

WebUI.setEncryptedText(findTestObject('Object Repository/Page\_Coink Colombia Viva/input\_Contrasea\_contra'), '4nvbrPglk7k=')

WebUI.click(findTestObject('Object Repository/Page\_Coink Colombia Viva/input\_Contrasea\_iniciar'))

WebUI.click(findTestObject('Object Repository/Page\_Coink Colombia Viva/input\_Comentarios\_reporte'))

WebUI.selectOptionByValue(findTestObject('Object Repository/Page\_Coink Colombia Viva/select\_Contenido inapropiado Proyecto \_a0bd2c'),

'Proyecto falso', true)

WebUI.setText(findTestObject('Object Repository/Page\_Coink Colombia Viva/textarea\_Descripcin\_descripcion'), 'malo')

WebUI.click(findTestObject('Object Repository/Page\_Coink Colombia Viva/input\_Salir\_enviar'))

WebUI.closeBrowser()

* 1. **Resultado esperado**

Se espera que el sistema no acepte los campos sin contenido, que confirme que estén todos los datos necesarios, que los datos estén seleccionados de una manera adecuada, y para realizar la acción de reportar el usuario debe estar registrado.

* 1. **Evaluación de la prueba**

El sistema rechaza los campos vacíos, por lo tanto, no deja enviar formularios vacíos.