Universidad Tecnológica Nacional
Facultad Regional Concepción del Uruguay

Ingeniería en Sistemas de Información

Tecnologías para la Automatización

Trabajo práctico Nº2: RPA

Grupo 1. Integrantes:

- Basgall, Juan Ignacio
- Ingani, Juan Cruz
- Riedel, Manuel Ángel
- Silva, Akeem

Profesores:

- Ing. Mauro Sander
- Ing. Jaime Piperno

Índice

Herramienta de RPA seleccionada

El presente trabajo tiene como objetivo analizar y evaluar herramientas de automatización utilizadas en el ámbito de la ingeniería en sistemas, con especial énfasis en su aplicación dentro de procesos repetitivos y validación de funcionalidades en entornos web.

En esta oportunidad se ha seleccionado Cypress, una herramienta de automatización moderna centrada en pruebas end-to-end (E2E) para aplicaciones web. Si bien Cypress no pertenece al grupo de herramientas de RPA clásicas como UiPath o Power Automate, su arquitectura innovadora y su enfoque en la automatización de flujos dentro del navegador permiten abordar con eficacia muchos de los desafíos propios de la automatización de procesos, especialmente en sistemas interactivos y basados en la web.

El análisis se estructura en torno a sus características técnicas, ventajas, desventajas y casos de uso representativos, con el propósito de comprender su alcance y su aplicabilidad tanto en contextos académicos como profesionales. Asimismo, se busca destacar el valor de Cypress como herramienta de apoyo en la mejora continua de la calidad del software, la eficiencia operativa y la reducción de errores humanos en tareas repetitivas.

Instructivo de instalación de la herramienta

Crear un directorio

```
mkdir cypress-qa
```

Entrar en el directorio recién creado

```
cd cypress-qa
```

Iniciar un proyecto de Node.js con configuración por defecto

```
npm init -y
```

Instalar Cypress como una dependencia de desarrollo

```
npm install cypress --save-dev
```

Dentro de la carpeta `cypress/e2e` crear un archivo `form_test.cy.js`

Descripción del proceso a automatizar

Para la implementación práctica del presente trabajo se ha seleccionado como entorno de pruebas la aplicación web SauceDemo (https://www.saucedemo.com), una plataforma diseñada específicamente para la validación de herramientas de testing automatizado.

El proceso a automatizar corresponde al flujo completo de un usuario en un e-commerce, desde el inicio de sesión hasta la finalización de una compra. Se emplearán los distintos perfiles de usuario provistos por la plataforma para simular escenarios variados que se pueden presentar, incluyendo tanto el flujo ideal como rutas alternativas o problemáticas.

Implementación de la automatización

Abrir Cypress

```
npx cypress open
```

Código:

```
const users = [
  { name: 'standard_user', shouldPass: true },
  { name: 'locked_out_user', shouldPass: false },
  { name: 'problem_user', shouldPass: false },
  { name: 'performance_glitch_user', shouldPass: false },
  { name: 'error_user', shouldPass: false },
  { name: 'visual_user', shouldPass: false }
]

describe('Flujo de compra en SauceDemo con distintos usuarios', () => {

  users.forEach(user => {
    it(`Compra con ${user.name}`, () => {
      cy.visit('https://www.saucedemo.com/')
      cy.get('#user-name').type(user.name)
      cy.get('#password').type('secret_sauce')
      cy.get('#login-button').click()
      cy.get('#add-to-cart-sauce-labs-backpack').click()
      cy.get('.shopping_cart_link').click()
      cy.get('#checkout').click()
      cy.get('#first-name').type('Manu')
      cy.get('#last-name').type('Riedel')
      cy.get('#postal-code').type('3260')
      cy.get('#continue').click()
      cy.get('#finish').click()

      cy.contains('h2', 'Thank you for your order!').should(
        user.shouldPass ? 'be.visible' : 'not.exist'
```

```
      )
    })
  })
})
```

satandard_user

Es el usuario de referencia que representa el camino feliz de la aplicación. Permite loguearse, navegar por los productos, agregarlos al carrito y completar el flujo de compra sin inconvenientes. Se utiliza como baseline para validar que la aplicación funciona correctamente bajo condiciones normales.

locked_out_user

Este usuario está bloqueado y no puede iniciar sesión. Al intentar acceder aparece un mensaje de error que indica la restricción. Sirve para probar cómo el sistema maneja cuentas sin acceso y que los mensajes de validación se muestren correctamente.

problem_user

Con este usuario la aplicación muestra errores visuales, principalmente en las imágenes de los productos que aparecen rotas o repetidas. Es útil para simular fallos de UI y comprobar si las pruebas automatizadas o manuales detectan inconsistencias en la presentación de los datos.

performance_glitch_user

Está diseñado para reproducir problemas de rendimiento: el login y la carga de la lista de productos son mucho más lentos que lo normal. Permite verificar si la aplicación mantiene su funcionalidad bajo condiciones de lentitud y si las pruebas automatizadas pueden manejar tiempos de espera prolongados.

error_user

Este usuario genera fallos funcionales intermitentes dentro de la aplicación. Algunos botones como “Add to cart” o “Checkout” no responden correctamente, lo que rompe el flujo de compra. Su uso permite validar que la automatización evidencian errores en acciones clave del proceso.

visual_user

Al ingresar con este usuario, la interfaz se muestra con errores de estilo y diseño: fuentes alteradas, tamaños incorrectos o layout desordenado. Está pensado para

exponer problemas de presentación y sirve para validar pruebas relacionadas con la experiencia visual y la consistencia de la UI