

Fundación Universitaria del Área Andina
Facultad de Ingenierías
Ingeniería de Sistemas - Modalidad Virtual
Pasantías Empresariales

**AUTOMATIZACIÓN DE CASOS DE PRUEBA, PARA
ASEGURAMIENTO DE CALIDAD EN EL DESARROLLO DE
SOFTWARE, EN APLICACIÓN WORKMAGAER ED CLOUD®**

Presentado Por:

Pablo Camilo Vásquez Segura

Colombia, Bogotá D.C., 2021

Contenido

1. Empresa Estrategias Documentales SAS.....	3
1.1 Software WM ED CLOUD®	3
1.2 Misión	3
1.3 Visión.....	3
1.4 Ubicación.....	4
1.5 Resumen empresarial.....	4
2. Introducción.....	4
3. Objetivos	4
3.1 Objetivo General	5
3.2 Objetivo Especifico.....	5
4. Proyecto Basado en la Automatización de Pruebas con Selenium y lenguaje de Programación Python.....	5
5. Introducción a las pruebas del software.....	5
6. Elección de los navegadores.....	6
7. Herramientas para pruebas funcionales web.....	7
7.1 Configuración de Entorno de Trabajo.....	7
7.2 Instalación de Herramientas	7
8. Actividad 1 Automatización de Funcionalidad Login.....	10
8.1 Diseño de Casos de Pruebas por Funcionalidades	10
8.2 Scripts Desarrollados por Escenarios de Pruebas.....	10
8.3 Resultado Esperado.....	13
10. Actividad 2 Automatización de Funcionalidad Creación de Formularios.....	14
10.1 Scripts Desarrollados por Escenarios de Pruebas.....	14
10.2 Creación de Capos por medio de la Creación de Formularios.....	16
10.3 Creación de Campos por medio de la Consulta de Formularios	18
10.4 Resultado esperado	19
12. Actividad 4 Automatización de Funcionalidad Radicación de Formularios	19
12.1 Scripts Desarrollados por Escenarios de Pruebas.....	20
12.2 Resultado Esperado.....	21
13. Conclusiones	22
14. Recomendaciones	22
15. Glosario	23
Bibliografía	23

PROPUESTA DE AUTOMATIZACIÓN DE CASOS DE PRUEBA PARA ASEGURAMIENTO DE CALIDAD EN EL DESARROLLO DE SOFTWARE

1. Empresa Estrategias Documentales SAS

1.1 Software WM ED CLOUD®

es un sistema de gestión de documentos electrónicos de archivo (SGDEA) desplegado en la nube, cuyo propósito es capturar, almacenar, acceder, gestionar y organizar la información física y electrónica de una organización, cumpliendo con la normativa archivística vigente y los requisitos técnicos y funcionales de la MOREQ. Cuenta con una interfaz de usuario moderna, intuitiva y de fácil manejo que se adapta fácilmente a cualquier dispositivo (computador de escritorio, computador portátil, tableta o celular inteligente) facilitando el ingreso desde diferentes lugares.

WM ED Cloud, se encuentra desarrollado bajo la modalidad de servicio SaaS y cuenta con el acceso mediante suscripción mensual o anual.

Los términos y/o conceptos acá presentados, están tanto en español como en inglés, dado que la corriente anglosajona es la de mayor uso en esta disciplina y para facilitar su consulta y entendimiento, los mismos no son los únicos en el tema, solo fueron elegidos porque son los que más escuchamos, pero existen muchos otros que no quedaron incluidos en este glosario y lo ideal es que este documento continúe siendo alimentado con aquellos términos que consideremos se deben incluir para facilitar nuestra labor diaria dentro del software .

Esperamos que este glosario pueda aportar y aclarar algunos de los conceptos que escuchamos a diario en nuestra labor.

1.2 Misión

Es una organización dedicada a la integración y optimización de los procesos de gestión documental tales como la recepción, producción, distribución, trámite, organización, conservación, disposición final y consulta de información bajo esquemas de eficacia y eficiencia, conformada por un equipo humano competente.

1.3 Visión

Consolidarnos para el 2025 como una empresa innovadora y sostenible, garantizando la continuidad del negocio y su crecimiento en el mercado nacional e internacional, reconocida por nuestros colaboradores, aliados, clientes y proveedores en prestación de servicios y desarrollo de software de administración de documentos, información y flujos de trabajo electrónicos con altos estándares de calidad.

1.4 Ubicación

Estrategias documentales, cuenta con una sede principal ubicada en Sabaneta, Antioquia, en la dirección 47d-185 a, Cl. 80 Sur

1.5 Resumen empresarial

Estrategias documentales cuenta actualmente dentro de su portafolio, con un Software de Gestión Documental, Gestión de Información y Flujos de Trabajo. Es una plataforma integral, colaborativa y participativa que tiene como propósito distribuir y gestionar el conocimiento y la información de la empresa en formato electrónico. Con acceso remoto a través de internet, puede también desplegarse en cualquier dispositivo móvil y en todos los navegadores disponibles, actualmente se encuentra en proceso de desarrollo el software WorkManager ED Cloud® el cual ara parte de las plataformas en la nube, dejando a un lado las instalaciones de escritorio.

2. Introducción

La automatización de pruebas de software es una de las opciones más fascinantes para enfrentar un equilibrio habitual de entregas al cliente sin comprometer la calidad del producto software; esto se hace más necesario cuando ya se tiene un producto en un ambiente de producción y con una gran cantidad de usuarios que pueden verse afectados por un fallo. Existen diversos tipos y niveles de pruebas, entre ellos las pruebas de aceptación las cuales tienen especial importancia de cara a la conformidad del comportamiento externo del producto según las expectativas del cliente.

El análisis se realizó para la aplicación WorkManager ED Cloud®, la cual, dentro de su proyecto de desarrollo, cuenta con procesos definidos y documentados para el desarrollo de funcionalidades y la ejecución de pruebas de calidad sobre dicho software. Esa ejecución de pruebas de calidad es un proceso que se ejecuta manualmente haciendo que se tome mayor tiempo y que exista una mayor posibilidad de que se cometan errores durante su ejecución. Además del análisis realizado, para el desarrollo de la propuesta de ejecución de casos de pruebas automáticas, se pretende realizar simulación de ejecución automática de casos reales buscando involucrar las herramientas que fueron identificadas inicialmente.

La automatización puede prometer grandes beneficios siempre y cuando recordemos que “pruebas” no solo significa probar lo mismo una y otra vez hasta el agotamiento: esto conlleva a evaluar los resultados con respecto a las diferentes entradas para así ver que pruebas dar valor agregado y saber que herramienta a emplear ya que esta nunca hará el trabajo solas sin un buen entendimiento.

3. Objetivos

3.1 Objetivo General

Diseñar una propuesta para la automatización y ejecución de pruebas, sobre las funcionalidades desplegadas para la aplicación web WorkManager ED Cloud®, con el fin de reducir tiempo en su ejecución y logrando una mejor respuesta para la ejecución continua de pasos repetidos en las pruebas, despliegues de funcionalidades y mejoras.

3.2 Objetivo Especifico

Analizar el proceso de pruebas llevados hoy en día con relación al despliegue de funcionalidad que se aplican para el aplicativo WorkManager ED Cloud®, con el fin de identificar posibles mejoras mediante la automatización de casos de prueba. Emplear herramientas de automatización como lo es Selenium, bajo el lenguaje de programación Python., posterior a ello plantear una propuesta de ejecución de casos de prueba automatizados en los desarrollos de WorkManager ED Cloud®, para que sea más eficiente en el proceso de aceptación y aprobación de los desarrollos realizados partiendo de Historias de Usuarios.

4. Proyecto Basado en la Automatización de Pruebas con Selenium y lenguaje de Programación Python

En este capítulo se presenta como funcionalidades a ser automatizadas de forma inicial, aquellos desarrollos a los cuales se les ha venido realizando un proceso de pruebas, integraciones con micro servicios desarrollados para la aplicación WorkManager ED Cloud.®.

Teniendo en cuenta lo descrito anteriormente, la automatización de los desarrollos estará basada en las funcionalidades actualmente desarrolladas:

- Inicio de Sesión en la aplicación WorkManager ED Cloud®
- Visualización de Entorno de trabajo de la aplicación WorkManager ED Cloud®
- Creación de Formularios en el aplicativo WorkManager ED Cloud®
- Creación de Campos en la Creación de Formularios en el aplicativo WorkManager ED Cloud®
- Radicación de Información en formularios creados
- Cargue de documentos con vinculación a la TRD.

5. Introducción a las pruebas del software

En este capítulo se describe el proceso de automatización de pruebas utilizando la propuesta de herramientas como Selenium y lenguaje de programación Python, el proceso comienza con definir que funcionalidades del proyecto WorkManager ED Cloud® serán las tenidas en cuenta para realizar el proceso de automatización, se debe tener en cuenta que las automatizaciones son basadas en pruebas funcionales, teniendo en cuenta los criterios de aceptación indicados en las historias de usuario desarrolladas por el PO del proyecto, estas historias de usuario han sido

aprobadas y validadas por el equipo scrum y sobre estas se realizaran las automatizaciones, las funcionalidades elegidas para el desarrollo de automatizaciones, corresponden a las funcionalidades de Login de usuario, contemplando sus diferentes validaciones a nivel de frontend , así como también posterior al Login el paso siguiente la visualización del dash board principal de la aplicación, seguido la funcionalidad generada para usuarios administradores correspondiente a creación de formularios, creación de campos en formularios creados y finalmente la radicación de estos formularios, para estas automatizaciones las funcionalidades nombradas anteriormente se validarían escenarios de pruebas sujetos a criterios de aceptación.

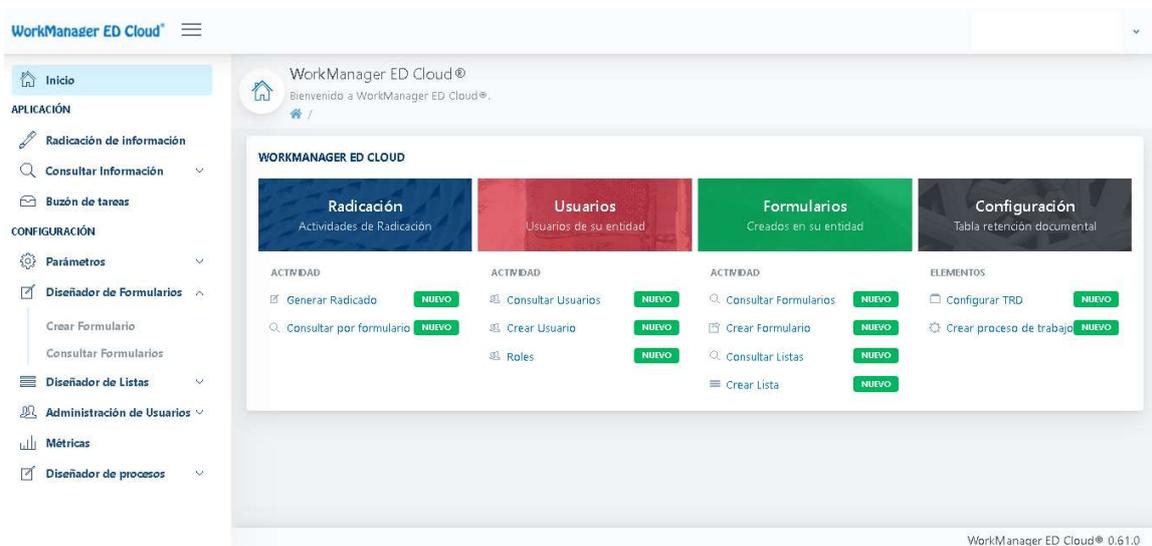


Ilustración 1 Dash Board Principal Aplicativo WorkManager ED Cloud®-Fuente Propia

6. Elección de los navegadores

Una de las principales preocupaciones para los desarrolladores WEB es el poder realizar desarrollos los cuales sean compatibles y realicen un correcto funcionamiento en el máximo número de navegadores web, de la misma manera la implementación de pruebas automatizadas busca la correcta ejecución de los script basándose en la implementación de extensiones que permiten a esta funcionalidad, poder ejecutarse de la mejor manera, por tal razón y teniendo encuesta lo anteriormente descrito se ha decidido y de acuerdo a los análisis de concurrencia en navegadores, implementar la automatización en el navegador Google Chrome. Nuestras pruebas de automatización correrán bajo el navegador Chrome contando con la extensión Selenium IDE y el driver ChromeDriver - WebDriver for Chrome.

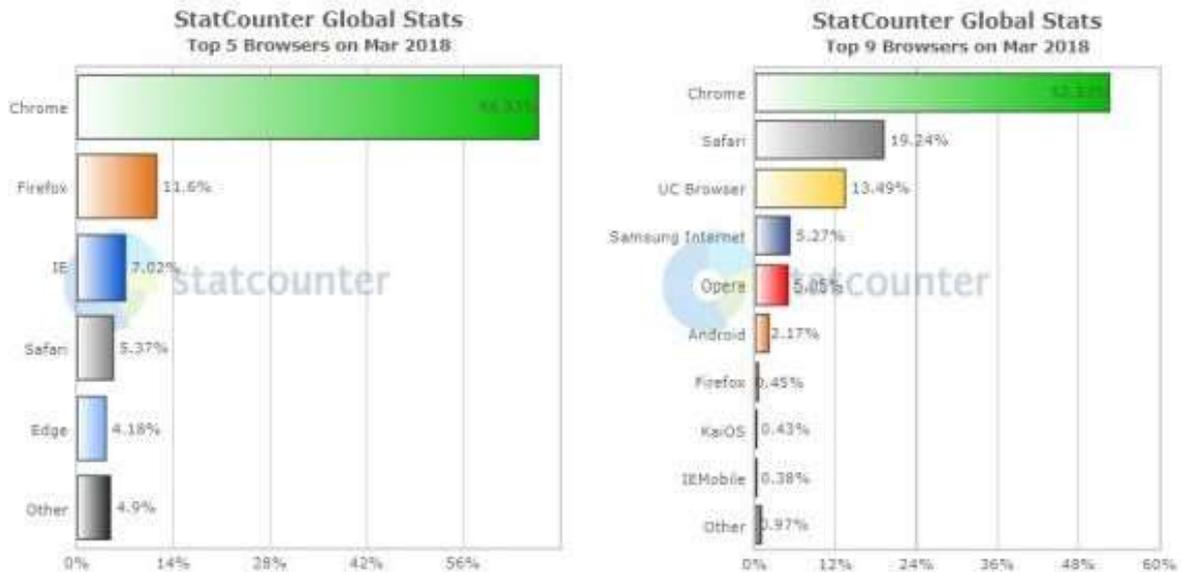


Ilustración 2: Estadísticas de uso de navegadores en el escritorio (izquierda) y móvil (derecha) a marzo de 2018, en porcentaje de usuarios. [13]

7. Herramientas para pruebas funcionales web

Como ya lo hemos mencionado, nuestro proyecto de automatización está basado en Selenium bajo el lenguaje de programación Python, para ello es necesario configurar nuestro entorno de trabajo, teniendo en cuenta esto se procederá a realizar las indicaciones correspondientes a la configuración pertinente.

7.1 Configuración de Entorno de Trabajo

Con el fin de realizar una instalación limpia de las herramientas requeridas para la elaboración y ejecución de script automatizados con Selenium y Python, es necesario instalar lo siguiente:

- Python en una versión superior a 3.6
- Selenium
- PyUnitReport (Librería la cual permite generar reportes en formato HTML.)
- Descarga de Driver de Selenium compatible con Google Chrome.
-

7.2 Instalación de Herramientas

- 1- Lo primero que se debe realizar es validar versión de Python que tenga instalada el equipo del usuario, para ello se debe ingresar a la consola de Windows y posteriormente escribir: Python -versión, de esta manera sabremos la versión que tenemos instalada, en caso de ser inferior a 3.6 o no contar con ella, se

puede realizar descarga e instalación de la misma, directamente de la página de Python : <https://www.python.org/downloads/>



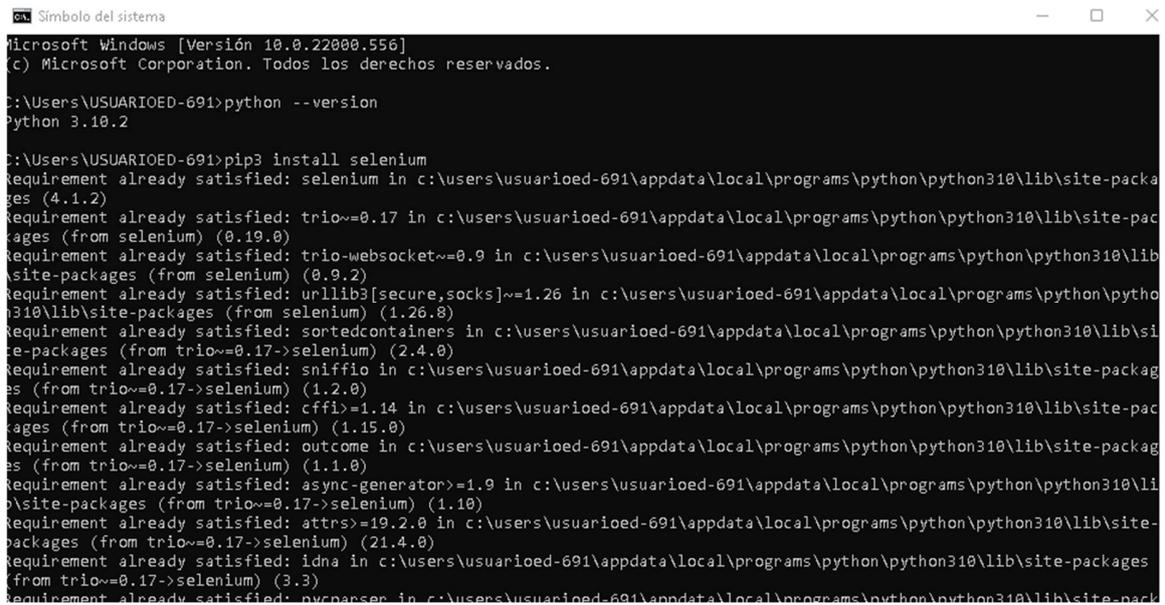
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22000.556]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIOED-691>python --version
Python 3.10.2

C:\Users\USUARIOED-691>
```

Ilustración 3 Verificación Versión Python - Fuente Propia

2- Realizar instalación de Selenium en nuestro equipo, para ello se debe ingresar a la consola de comandos de nuestro navegador, posterior mente escribir lo siguiente: pip3 install Selenium



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22000.556]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIOED-691>python --version
Python 3.10.2

C:\Users\USUARIOED-691>pip3 install selenium
Requirement already satisfied: selenium in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (4.1.2)
Requirement already satisfied: trio<=0.17 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from selenium) (0.19.0)
Requirement already satisfied: trio-websocket<=0.9 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from selenium) (0.9.2)
Requirement already satisfied: urllib3[secure,socks]<=1.26 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from selenium) (1.26.8)
Requirement already satisfied: sortedcontainers in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from trio<=0.17->selenium) (2.4.0)
Requirement already satisfied: sniffio in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from trio<=0.17->selenium) (1.2.0)
Requirement already satisfied: cffi<=1.14 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from trio<=0.17->selenium) (1.15.0)
Requirement already satisfied: outcome in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from trio<=0.17->selenium) (1.1.0)
Requirement already satisfied: async-generator<=1.9 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from trio<=0.17->selenium) (1.10)
Requirement already satisfied: attrs<=19.2.0 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from trio<=0.17->selenium) (21.4.0)
Requirement already satisfied: idna in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from trio<=0.17->selenium) (3.3)
Requirement already satisfied: pyparsing in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-pack
```

Ilustración 4 Instalación de Selenium- Fuente Propia

- 3- Por último, se debe realizar la instalación de la librería para generar reportes con Selenium, por tal razón y de la misma manera que la verificación de Python y la instalación de Selenium, ejecutamos nuestra consola de Windows y escribimos lo siguiente: pip3 install pyunitreport

```

C:\Users\USUARIOED-691>pip3 install pyunitreport
Collecting pyunitreport
  Using cached pyunitreport-0.1.4.tar.gz (10 kB)
Requirement already satisfied: Jinja2 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from pyunitreport) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\usuarioed-691\appdata\local\programs\python\python310\lib\site-packages (from Jinja2->pyunitreport) (2.0.1)
Using legacy 'setup.py install' for pyunitreport, since package 'wheel' is not installed.
Installing collected packages: pyunitreport
  Running setup.py install for pyunitreport ... done
Successfully installed pyunitreport-0.1.4
WARNING: You are using pip version 21.2.4; however, version 22.0.4 is available.
You should consider upgrading via the 'C:\Users\USUARIOED-691\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.

C:\Users\USUARIOED-691>
  
```

Ilustración 5 Instalación Librería de Reportes Selenium-Fuente Propia

- 4- Por último, paso, debemos realizar descarga del driver el cual permitirá la ejecución automática de nuestro script con el navegador, por tal razón se debe ingresar al sitio oficial de descarga: <https://chromedriver.chromium.org/downloads> posterior a ello, seleccionar la versión mas reciente y descargar el driver de acuerdo a nuestro sistema operativo.

Index of /101.0.4951.15/

Name	Last modified	Size	ETag
Parent Directory	-	-	-
chromedriver_linux64.zip	2022-04-01 07:53:29	6.28MB	26b17dda1bc1175fa3d3e78506a562b
chromedriver_mac64.zip	2022-04-01 07:53:32	7.96MB	0c0e2932391fa49521b488e0ce4dfe1e
chromedriver_mac64_ml.zip	2022-04-01 07:53:34	7.19MB	e18cf3711957ca4210802a065e334d67
chromedriver_win32.zip	2022-04-01 07:53:37	6.05MB	fe5b321d4b97907c89dc164c6c99785c
notes.txt	2022-04-01 07:53:42	0.00MB	446e4ff3aba3adbceda721a038451202

Ilustración 6 Versiones actualmente disponibles -Fuente Propia

Nota: Posterior a la descarga del Driver es necesario alojarlo en una ruta la cual podamos recordar, esto dado a que, para las automatizaciones, esta ruta se llamara por medio de comentarios para la ejecución del driver.

8. Actividad 1 Automatización de Funcionalidad Login

Una vez finalizado la instalación y configuración de nuestro entorno de trabajo, se procede a realizar la documentación correspondiente a levantamiento de información por medio de Historias de usuarios, escenarios de pruebas y elaboración de scripts.

8.1 Diseño de Casos de Pruebas por Funcionalidades

Para la funcionalidad correspondiente a Login de usuarios, se desarrollo por parte del Product Owner del proyecto la HU-005_Login, en la cual se evidencian 06 criterios de aceptación y precondiciones o requisitos dispuestos para que el desarrollador de la funcionalidad tuviera en cuenta en el momento de abordarla, posterior al análisis de la historia de usuario, se procede a realizar el documento correspondiente a la elaboración de casos de pruebas, la documentación elaborada para el desarrollo de script, será entregada a la empresa en un repositorio indicado por ellos mismos, teniendo en cuenta que es información confidencial, no es posible visualizarla detallada mente en este documento.

Nombre	Fecha de modificación	Tipo	Tamaño
24-11-2021 Casos de Prueba Login	1/12/2021 12:06 p. m.	Hoja de cálculo d...	75 KB
Anexo1_HU-005	2/07/2021 1:21 p. m.	Documento de Mi...	35 KB
HU-005_Login	7/09/2021 1:23 p. m.	Documento de Mi...	13 KB
Proceso	21/01/2022 9:32 a. m.	Archivo PNG	54 KB

Ilustración 7 Documentación Historia de usuarios y criterios de aceptación Automatizados-Fuente Propia

Nombre	Fecha de modificación	Tipo	Tamaño
Casos de Prueba HU-05 Login	12/04/2022 1:09 p. m.	Carpeta de archivos	
Diagramas de Flujo	25/11/2021 4:51 p. m.	Carpeta de archivos	
Resultados Test Login	23/03/2022 10:28 a. m.	Carpeta de archivos	
Scripts Beta	3/02/2022 2:45 p. m.	Carpeta de archivos	
TestSuity_2	3/02/2022 2:45 p. m.	Archivo PY	2 KB
Validar Login de Usuario_Full	1/03/2022 1:27 p. m.	Archivo PY	5 KB

Ilustración 8 Documentación General Historia de usuarios y criterios de aceptación Automatizados-Fuente Propia

8.2 Scripts Desarrollados por Escenarios de Pruebas

La estructura que tiene actualmente el script desarrollado para lograr la correcta automatización de los escenarios de pruebas para el proceso de Login de usuario es la siguiente:

AREANDINA

Fundación Universitaria del Área Andina

Línea(s) de código	Significado
<pre>import HtmlTestRunner import unittest from selenium import webdriver from HtmlTestRunner import HTMLTestRunner from selenium.webdriver.common.by import By from selenium.webdriver.common.keys import Keys from selenium.webdriver.support import expected_conditions as EC import time</pre>	<p>el primer paso es la Importacion de librerias de selenium y python con el fin de hacer uso de las clases, métodos y atributos que las conforman.</p>
<pre>from selenium.webdriver.support.wait import WebDriverWait</pre>	<p>El segundo paso es crear una instancia de Chrome que luego se puede controlar con comandos de Selenium. Hay previsto un tiempo de espera en segundos para que arranque el navegador.</p>
<pre>class WorkManagerTest(unittest.TestCa se): def setUp(self): self.driver = webdriver.Chrome(executable_pat h=r"P:\Curso Selenium\chromedriver")</pre>	<p>El script llamará la ruta en la cual se encuentra alojado nuestro driver para ejecución del script en el navegador Chrome de forma automática</p>
<pre>class WorkManagerTest(unittest.TestCa se): def setUp(self): self.driver.get("http://dev.workmana gercloud.com/")</pre>	<p>El script llamará la página http://dev.workmanagercloud.com, en la cual se realizarán las automatizaciones o validaciones de los criterios de aceptación.</p>
<pre>search_field.send_keys("Término de búsqueda")search_field.submit() ldef test_01_search(self): pass def test_02_validate_form_login(self): def tearDown(self): self.driver.close()</pre>	<p>A continuación, se introduce el texto del término de búsqueda con el método submit().</p> <p>Estas líneas de código, corresponden a las validaciones de prueba que se ejecutaran, se nombran con el nombre del test que se realizará la ejecución</p>
<pre>if __name__ == '__main__': unittest.main(testRunner=HtmlTest Runner.HTMLTestRunner(output=" Resultados Test Login"))</pre>	<p>El código indica a la instancia del navegador que debe cerrarse.</p> <p>La línea final del código indica al script la creación del reporte correspondiente a los resultados de la automatización.</p>

NOTA: es necesario tener en cuenta que la complejidad o la cantidad de los test, varían de acuerdo a los criterios de aceptación tenidos en cuenta por la historia de usuarios y los escenarios de pruebas indicados en la documentación para ser desarrollados y ejecutados.

Ilustración 9 Descripción general de la estructura del script de automatización

```
import HtmlTestRunner
import unittest
from selenium import webdriver
from HtmlTestRunner import HTMLTestRunner
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as EC
import time

from selenium.webdriver.support.wait import WebDriverWait

class WorkManagerTest(unittest.TestCase):

    def setUp(self):
        #Se indica al escript la ruta en el driver para la ejecucion del navegador de forma automatica
        self.driver = webdriver.Chrome(executable_path="P:\\Curso Selenium\\chromedriver")

    # Se indica al escript la ruta en el navegador que debe abrir
    self.driver.get("http://dev.workmanagercloud.com/")

    # Inicio de los test a nivel de criterios de aceptacion...

    def test_01_search(self):
        pass

    def test_02_validate_form_login(self):

# se indica al script el cierre del navegador

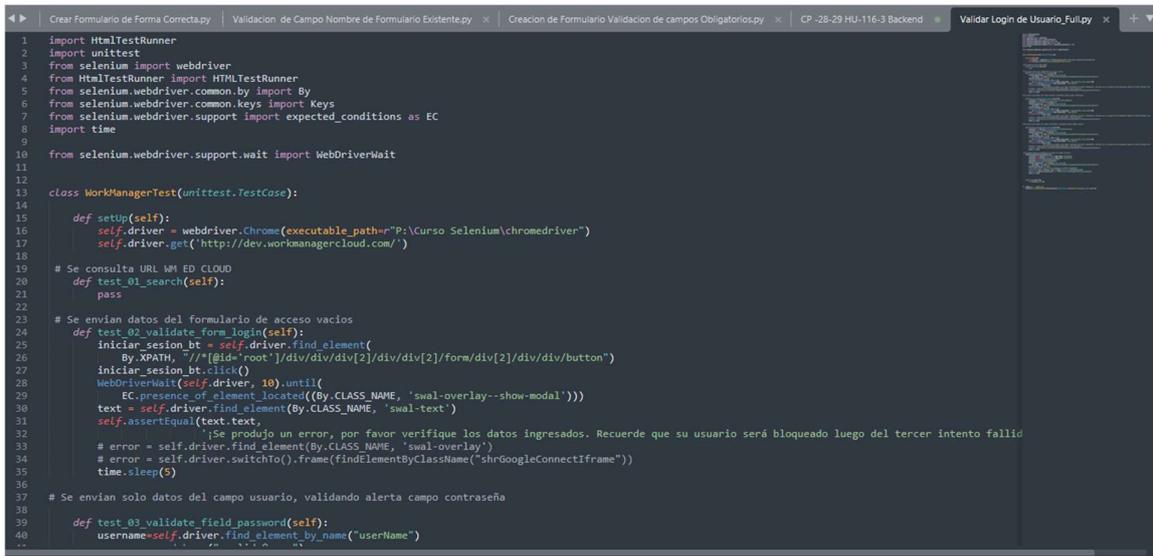
    def tearDown(self):
        self.driver.close()

# codigo para generacion de reporte por medio de pyunitreport
if __name__ == '__main__':
    unittest.main(testRunner=HtmlTestRunner.HTMLTestRunner(output="Resultados Test Login"))
```

Ilustración 10 Arquitectura de Código Script automatizaciones-Fuente Propia

Teniendo en cuenta el orden de los escenarios de pruebas identificados en el documento de pruebas, se realiza la automatización de los siguientes escenarios:

- def test_01_search
- def test_02_validate_form_login
- def test_03_validate_field_password
- def test_04_validate_field_username
- def test_05_login_success



```
1 import HtmlTestRunner
2 import unittest
3 from selenium import webdriver
4 from HtmlTestRunner import HTMLTestRunner
5 from selenium.webdriver.common.by import By
6 from selenium.webdriver.common.keys import Keys
7 from selenium.webdriver.support import expected_conditions as EC
8 import time
9
10 from selenium.webdriver.support.wait import WebDriverWait
11
12
13 class WorkManagerTest(unittest.TestCase):
14
15     def setUp(self):
16         self.driver = webdriver.Chrome(executable_path="C:\\Curso Selenium\\chromedriver")
17         self.driver.get('http://dev.workmanagercloud.com/')
18
19     # Se consulta URL WM ED CLOUD
20     def test_01_search(self):
21         pass
22
23     # Se envian datos del formulario de acceso vacios
24     def test_02_validate_form_login(self):
25         iniciar_session_bt = self.driver.find_element(
26             By.XPATH, "//*[@id='root']/div/div/div[2]/div/div[2]/form/div[2]/div/div/button")
27         iniciar_session_bt.click()
28         WebDriverWait(self.driver, 10).until(
29             EC.presence_of_element_located((By.CLASS_NAME, 'swal-overlay--show-modal')))
30         text = self.driver.find_element(By.CLASS_NAME, 'swal-text')
31         self.assertEqual(text.text,
32             "¡Se produjo un error, por favor verifique los datos ingresados. Recuerde que su usuario será bloqueado luego del tercer intento fallido")
33         # error = self.driver.find_element(By.CLASS_NAME, 'swal-overlay')
34         # error = self.driver.switchTo().frame(findElementByClassName("shrGoogleConnectIframe"))
35         time.sleep(5)
36
37     # Se envian solo datos del campo usuario, validando alerta campo contraseña
38
39     def test_03_validate_field_password(self):
40         username=self.driver.find_element_by_name("userName")
```

Ilustración 11 Script de Automatización Login de Usuario-Fuente Propia

La anterior imagen muestra el script desarrollado para realizar la automatización del proceso de Login, teniendo en cuenta los escenarios de pruebas expuestos en la documentación desarrollada.

8.3 Resultado Esperado

9. Posterior al desarrollo del código correspondiente a cada criterio de aceptación, se debe realizar ejecución del mismo, con el fin de validar en los resultados del test, que cada criterio o ejecución de prueba de forma automatizada, estén acorde a lo estipulado en cada línea de código, es preciso indicar que quien interactúe con el script desarrollado debe contar con algún conocimiento en programación, adicional a esto entender la documentación a nivel de Historia de usuario y documento de análisis y diseño de casos de pruebas, esto con el fin pueda entender lo que aquí se realiza, por otra parte estas ejecuciones automáticas, ahorraran tiempo en pruebas de regresión sobre la funcionalidad.

```
1 import HtmlTestRunner
2 import unittest
3 from selenium import webdriver
4 from HtmlTestRunner import HTMLTestRunner
5 from selenium.webdriver.common.by import By
6 from selenium.webdriver.common.keys import Keys
7 from selenium.webdriver.support import expected_conditions as EC
8 import time
9
10 from selenium.webdriver.support.wait import WebDriverWait
11
12
13 class WorkManagerTest(unittest.TestCase):
14
15     def setUp(self):
16         self.driver = Service=webdriver.Chrome("P:\Curso Selenium\chromedriver")
17         self.driver.get('http://dev.workmanagercloud.com/')
18
19 # Se consulta URL WM ED CLOUD
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661

```

- Validación de caracteres especiales
- Validación de listas
- Validación de alertas

■ Creacion de Formulario Validacion de campos Obligatorios	19
■ Crear Formulario de Forma Correcta	11
■ Se diligencia solo el campo ESTADO	19
■ Se diligencia solo el campo NOMBRE DEL FORMULARIO	19
■ Validacion de Campo Nombre de Formulario Existente	19
■ Validacion de Campo Unico	19
■ Validacion de Caracteres No Permitidos Campo Nombre del Formulario	19
■ Validacion de Caracteres Permitidos Campo Nombre del Formulario	19
■ Validacion de Campos Mensaje Campo estado	19
■ Validacion de Campos Mensaje Campo Nombre del Formulario	19
■ Validacion Longitud Minima de Caracteres Campo Nombre del Formulario	19
■ Verificacion lista desplegable Campo Estado	19

Ilustración 13 Script Automatizados Creación de Formularios Fuente Propia

En el desarrollo del script para la creación de formularios, se contempla la estructura mencionada anteriormente para la validación de Login, se trabaja en la misma estructura de desarrollo con Selenium.

```

Crear Formulario de Forma Correcta.py
1 import HtmlTestRunner
2 import unittest
3 from selenium import webdriver
4 from HtmlTestRunner import HTMLTestRunner
5 from selenium.webdriver.common.by import By
6 from selenium.webdriver.common.keys import Keys
7 from selenium.webdriver.support.ui import Select
8 from selenium.webdriver.support import expected_conditions as EC
9 import time
10
11 from selenium.webdriver.support.wait import WebDriverWait
12
13
14 class WorkManagerTest(unittest.TestCase):
15
16
17     def setUp(self):
18         self.driver = Service(webdriver.Chrome("P:Curso Selenium\chromedriver"))
19         self.driver.get('https://dev.workmanagercloud.com')
20
21
22     # Crear Formulario de Forma Correcta
23
24     def test_01_create_form_success(self):
25         username=self.driver.find_element(By.NAME,"userName")
26         password=self.driver.find_element(By.NAME,"password")
27         username.send_keys("****")
28         password.send_keys("****")
29         iniciar_sesion_bt = self.driver.find_element(
30             By.XPATH, "//*[@id='root']/div/div/div[2]/div/div[2]/form/div[2]/div/div/button")
31         iniciar_sesion_bt.click()
32
33     # Se espera hasta que la URL contenga la palabra 'dashboard'
34     WebDriverWait(self.driver, 10).until(EC.url_contains('dashboard'))
35     assert self.driver.current_url == 'https://dev.workmanagercloud.com/dashboard'
36
37     # Seleccionar opción Crear Formulario
38     WebDriverWait(self.driver, 20)\
39         .until(EC.element_to_be_clickable((By.XPATH,
40             '/html/body/div/div/div[3]/div[3]/div[1]/div/div[2]/div/div/div/div/div[2]/div[3]/ul/ul/li[3]/a/span')))\
41         .click()
42     time.sleep(3)
43
44     #Diligenciaiento campos del formulario
45     nombre_del_formulario=self.driver.find_element(By.NAME,"name")
46     descripcion=self.driver.find_element(By.CSS_SELECTOR,"textarea")
47     estado=self.driver.find_element(By.XPATH, "//*[@id='status']")

```

Ilustración 14 Automatización Creación de Campos de Forma Correcta

Nota: como se evidencia en la imagen anterior, cada paso de la automatización es comentada, esto con el fin de ser de fácil entendimiento de quien pueda ejecutar las pruebas, por otro lado, se debe tener en cuenta que existen datos únicos los cuales deben ser remplazados en caso de correr el script, como por ejemplo datos de usuario de acceso, nombres ya existentes de formularios etc.

Para las automatizaciones de la HU-081 correspondiente a la creación de campos en formularios, se toma en cuenta dos escenarios:

- 1- Creación de Campos por medio de la Creación de Formularios
- 2- Creación de Campos Por medio de la Consulta sobre Formularios

10.2 Creación de Capos por medio de la Creación de Formularios

Para la creación de campos, realizando el proceso de crear formulario y seguido los campos sobre el formulario, se desarrollaron 10 script, en los cuales se logra la automatización de:

- Botones de continuidad de procesos
- Validación de Tipos de Datos
- Validación de listas desplegadas
- Validación de Longitud de caracteres

- Crear Validacion Dato Campo Longitud máxima y Minima de Forma Correcta
- Crear Form Validacion Dato Campo Longitud máxima de caracteres
- Crear Form Seleccion Dato tipo Cadena de Texto Control Caja de texto
- Crear Form Validacion Seleccion de Dato_Accion Boton Siguiente
- Crear Form Validacion Boton Siguiente
- Crear Form Validacion Seleccion Tipo Control
- Crear Form Validacion tipo de dato Fecha
- Crear Form Validacion tipo de dato Entero
- Crear Form Validacion tipo de dato Cadena de texto
- Crear Form Visualizacion Lista Desplegable Con Tipo de Dato

Ilustración 15 Script desarrollador para el proceso de creación de campos sobre formularios

Nota: En el desarrollo del script para la creación de campos de formularios, se contempla la estructura mencionada anteriormente para la validación de Login y la creación de formularios, se trabaja en la misma estructura de desarrollo con Selenium, contrariando líneas de código para fácil entendimiento de quien ejecuta y exportando las librerías de Selenium.

```

Crear Form Validacion tipo de dato Entero.py x
'/html/body/div/div/div[3]/div[3]/div[1]/div/div/div[2]/div/div/div/div[2]/div[3]/ul/li[3]/a/span'))\
.click()
time.sleep(3)
#Diligenciamiento campos del formulario
nombre_del_formulario=self.driver.find_element(By.NAME,"name")
descripcion=self.driver.find_element(By.CSS_SELECTOR,"textarea")
estado=self.driver.find_element(By.XPATH,"//*[@id='status']")
nombre_del_formulario.send_keys("Pasantias ED Cloud v18")
time.sleep(3)
descripcion.send_keys("prueba")
time.sleep(3)
# Verificacion opciones de la lista
opcion_estado.find_elements(By.TAG_NAME,"option")
for option in opcion:
    print("Los valores son: %s" % option.get_attribute("value"))
    option.click()
    time.sleep(2)
# seleccionar estado de la lista con ID 2 (activo)
seleccionar_estado=Select(self.driver.find_element(By.XPATH,"//*[@id='status']"))
seleccionar_estado.select_by_value("2")
time.sleep(2)
# Dar clic en boton Crear
crear_sesion_bt = self.driver.find_element(By.XPATH,"//*[@id='root']/div/div[3]/div[3]/div[1]/div/div[2]/div/div/form/div[4]/button")
crear_sesion_bt.click()
time.sleep(3)
# Validar mensaje creacion de Campos
WebDriverWait(self.driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, 'swal-overlay')))
text = self.driver.find_element(By.CLASS_NAME, 'swal-text')
self.assertEqual(text.text,'¡Por favor cree los campos para el formulario!')
time.sleep(3)
# Dar clic en boton Crear campo
crear_campo_bt = self.driver.find_element(By.XPATH,"//*[@id='panel-0']/div/div/div[1]/div/div[2]/div/div/button")
crear_campo_bt.click()
time.sleep(3)
# Dar Seleccionar Dato Entero
Select(self.driver.find_element_by_id("type_data")).select_by_visible_text("Entero")
time.sleep(3)
# Visualizacion de Control de Datos
self.driver.find_element_by_id("control_data").click()
time.sleep(3)
Select(self.driver.find_element_by_id("control_data")).select_by_visible_text(u"Caja de texto numérica")
time.sleep(3)

```

Ilustración 16 Validación de Capos desde la Creación de Formularios Fuente propia

10.3 Creación de Campos por medio de la Consulta de Formularios

Para la creación de campos, realizando el proceso de consultar un formulario y seguido los campos sobre el formulario, se toma como parámetros de búsqueda y de pruebas el formulario ya creado en la aplicación WorkManager ED Cloud®, para ello se utiliza el Xpath del formulario deseado, con el fin de que la búsqueda del mismo sea exacta, teniendo en cuenta lo anterior, se desarrollan 11 script, en los cuales se logra la automatización de:

- Botones de continuidad de procesos
- Validación de Tipos de Datos
- Validación de listas desplegables
- Validación de Longitud de caracteres

- Consul Validacion Dato Campo Longitud máxima y Minima de Forma Correcta
- Consul Validacion Dato Campo Longitud máxima de caracteres
- Consul Diligenciar Nombre De Campo Con Caracteres Especiales No permitidos
- Consul Seleccion Dato tipo Cadena de Texto Control Caja de texto
- Consul Seleccion de Tipo de Dato y control de Forma Correcta
- Consul Validacion Seleccion de Dato_Accion Boton Siguiente
- Consul Validacion Boton Siguiente
- Consul Validacion Seleccion Tipo Control
- Consul Validacion tipo de Dato Entero
- Consul Validacion tipo de Dato Fecha
- Consul Validacion tipo de dato Cadena de texto

Ilustración 17 Script desarrollador para el proceso de creación de campos sobre la consulta de formularios

Nota: En el desarrollo del script para la creación de campos de formularios, se contempla la estructura mencionada anteriormente para la validación de Login y la creación de formularios, se trabaja en la misma estructura de desarrollo con Selenium, contrariando líneas de código para fácil entendimiento de quien ejecuta y exportando las librerías de Selenium.

```

84 # Dar click en el boton Crear Formulario
85 time.sleep(3)
86 # Seleccionar opcion Consultar Formulario
87 self.driver.find_element_by_link_text(u"Diseñador de Formularios").click()
88 time.sleep(3)
89 self.driver.find_element_by_link_text("Consultar Formularios").click()
90 time.sleep(3)
91 # Seleccionar Un Formulario
92 self.driver.find_element_by_xpath("//div[@id='root']/div/div[3]/div/div[2]/div/div[2]/div/div/div/div/div/div[3]/div/div[6]/div/div/button").click()
93 time.sleep(3)
94 self.driver.find_element_by_xpath("//*[id='root']/div/div[3]/div[3]/div[1]/div/div[2]/div/div/div/div/div/div[1]/div[3]/div[1]/div/div[6]/div/div/button").click()
95 time.sleep(3)
96 # Dar click en Boton Crear Campo
97 self.driver.find_element_by_xpath("//*[id='panel-0']/div/div/div[1]/div/div[2]/div/div/button/span").click()
98 time.sleep(3)
99 # Dar Seleccionar Datos Cadena de Texto
100 select(self.driver.find_element_by_id("type_data")).select_by_visible_text("Cadena de texto")
101 self.driver.find_element_by_id("control_data").click()
102 time.sleep(3)
103 # Visualizar Controles
104 select(self.driver.find_element_by_id("control_data")).select_by_visible_text("Caja de Texto")
105 self.driver.find_element_by_id("control_data").click()
106 time.sleep(2)
107 # Mover scroll hacia el final
108 deslizar_hacia_abajo=self.driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
109 time.sleep(3)
110 # Dar click en boton Siguiente
111 crear_sesion_bt = self.driver.find_element(By.XPATH, "//*[id='root']/div/div[3]/div[3]/div[1]/div/div[2]/div/div/div/div/div[4]/div/button[2]")
112 crear_sesion_bt.click()
113 # Diligenciar Campo Nombre Del Campo
114 self.driver.find_element_by_id("label").send_keys("ciudad")
115 time.sleep(10)
116 self.driver.find_element_by_xpath("//div[@id='root']/div/div[3]/div/div[2]/div/div/div/div/div[4]/div/button[2]").click()
117 time.sleep(10)
118 self.driver.find_element_by_id("len_max_chars").send_keys("4")
119 time.sleep(5)
120 self.driver.find_element_by_id("len_min_chars").send_keys("10")
121 time.sleep(10)
122 # Validar mensaje Validación de campos
123 self.driver.find_element_by_xpath(u"(//*[normalize-space(text()) and normalize-space(.)='La longitud menor de caracteres no puede superar la mayor!'])[1]/following").click()
124 time.sleep(3)
125

```

Ilustración 18 Automatización validación de longitud de caracteres Fuente Propia

10.4 Resultado esperado

11. Posterior al desarrollo del código correspondiente a cada criterio de aceptación, se debe realizar ejecución del mismo, con el fin de validar en los resultados del test, que cada criterio o ejecución de prueba de forma automatizada, estén acorde a lo estipulado en cada línea de código, es preciso indicar que quien interactúe con el script desarrollado debe contar con algún conocimiento en programación, adicional a esto entender la documentación a nivel de Historia de usuario y documento de análisis y diseño de casos de pruebas, esto con el fin pueda entender lo que aquí se realiza, por otra parte estas ejecuciones automáticas , ahorraran tiempo en pruebas de regresión sobre la funcionalidad.

12. Actividad 4 Automatización de Funcionalidad Radicación de Formularios

Con relación al proceso de automatización, correspondiente a la funcionalidad de Radicación de formularios, se aborda la Historia de usuario HU-057, en la cual se definen aquellos criterios de aceptación tenidos en cuenta para la realización del desarrollo a nivel de Backend y frontend y posterior mente y con el objetivo de esta pasantía, realizar la automatización de las pruebas.

Teniendo en cuenta lo anteriormente descrito, se realiza de la misma manera que las funcionalidades automatizadas anteriormente, el análisis y desarrollo de casos de pruebas, en esta funcionalidad se debe tener en cuenta que no todos los formularios desarrollados por un usuario o varios usuarios contienen la misma estructura, por tal razón para el proceso de radicación, se validara criterios tales como la existencia de la funcionalidad, la visualización de formularios, en vio de

proceso de radicación, entre otros criterios de aceptación relacionadas a la Historia de usuarios.

12.1 Scripts Desarrollados por Escenarios de Pruebas

Para el proceso de automatización de radicación, se continúa manejando la misma estructura de desarrollo, la cual es la importación de librerías, creación de etiquetas de pruebas, comentario de líneas o acciones dentro del script, para esta funcionalidad y teniendo en cuenta que su nivel de complejidad de acuerdo a la historia de usuario desarrollada, se crearon 5 scripts los cuales abordan los 4 criterios de aceptación planteados en la HU, se desarrollan de acuerdo a la existencia de:

- Existencia de la funcionalidad de radicación
- Botones para realizar procesos de radicación
- Validación dentro de la radicación
- Detalle del proceso o radicado.

-  Casos de Prueba Radicacion de Informacion
-  Resultados Test Crear Formularios
-  Radicar Formulario de Forma Correcta
-  Seleccionar Formulario
-  Seleccionar Funcionalidad Radicacion de Informacion
-  Validacion de Campos
-  Validacion detalle Radicado

Como se indicaba anterior mente, la estructura del script contempla librerías que Selenium requiere para la correcta ejecución con relación al lenguaje de programación Python, así como comentario de líneas de código, para mayor entendimiento de quien realice la ejecución del script.

```
1 import HtmlTestRunner
2 import unittest
3 from selenium import webdriver
4 from HtmlTestRunner import HTMLTestRunner
5 from selenium.webdriver.common.by import By
6 from selenium.webdriver.common.keys import Keys
7 from selenium.webdriver.support.ui import Select
8 from selenium.webdriver.support import expected_conditions as EC
9 import time
10
11 from selenium.webdriver.support.wait import WebDriverWait
12
13
14 class WorkManagerTest(unittest.TestCase):
15
16
17     def setUp(self):
18         self.driver =s=Service=webdriver.Chrome("P:\Curso Selenium\chromedriver")
19         self.driver.get('https://dev.workmanagercloud.com')
20
21
22     # Crear Formulario de Forma Correcta
23
24     def test_01_create_form_success(self):
25         username=self.driver.find_element(By.NAME,"userName")
26         password=self.driver.find_element(By.NAME,"password")
27         username.send_keys("*****")
28         password.send_keys("*****")
29         iniciar_sesion_bt = self.driver.find_element(
30             By.XPATH, "//*[@id='root']/div/div/div[2]/div/div[2]/form/div[2]/div/div/button")
31         iniciar_sesion_bt.click()
32     # Se espera hasta que la URL contenga la palabra 'dashboard'
33     WebDriverWait(self.driver, 10).until(EC.url_contains('dashboard'))
34     assert self.driver.current_url == 'https://dev.workmanagercloud.com/dashboard'
35     time.sleep(5)
36     # Seleccionar opcion Radicacion Informacion
37     WebDriverWait(self.driver, 20)\
38     .until(EC.element_to_be_clickable((By.XPATH,
39         "//*[@id='root']/div/div[3]/div[2]/div[4]/div[1]/div[2]/ul/li/a ")))\
40     .click()
41     time.sleep(3)
42
43
44     def tearDown(self):
```

Ilustración 19 Automatización Existencia Funcionalidad de Radicacion_Fuente Propia

Nota: Cada script desarrollado cuenta con la misma estructura de código y lenguaje de programación.

12.2 Resultado Esperado

Posterior al desarrollo del código correspondiente a cada criterio de aceptación, se debe realizar ejecución del mismo, con el fin de validar en los resultados del test, que cada criterio o ejecución de prueba de forma automatizada, estén acorde a lo estipulado en cada línea de código, es preciso indicar que quien interactúe con el script desarrollado debe contar con algún conocimiento en programación, adicional a esto entender la documentación a nivel de Historia de usuario y documento de análisis y diseño de casos de pruebas, esto con el fin pueda entender lo que aquí se realiza, por otra parte estas ejecuciones automáticas , ahorraran tiempo en pruebas de regresión sobre la funcionalidad.

13. Conclusiones

La etapa de pasantías, ha representado en mi desarrollo intelectual y personal un fundamento indispensable para el proceso de aprendizaje, esto dado a que se me ha permitido aumentar por medio de la experiencia, conocer los diferentes contextos de la empresa Estrategias Documentales S.AS, los cuales se deben analizar y tener en cuenta en los procesos de desarrollo de software, lenguajes de programación y procesos de automatizaciones a nivel de QA para la ejecución de pruebas funcionales.

Las actividades anteriormente descritas, las funcionalidades automatizadas, de acuerdo al plan de trabajo y propuestas de automatización aprobadas, han sido cumplidas en su totalidad y de manera satisfactoria, por ello se puede enfatizar que el proceso de pasantías ha sido provechoso al máximo para todos los involucrados, por ejemplo y hablando personalmente, como alumno me siento con conocimientos adquiridos en terrenos cuya existencia de vacíos en meses anteriores eran algo grandes, a la fecha y con la finalización de las pasantías se han adquirido conocimientos grandes, haciendo en mi una mano de obra capacitada y de calidad, por otro lado la universidad, quien una vez ha cumplido su visión de forma exitosa, la empresa Estrategias Documentales SAS, al haber obtenido los servicios y aportes dados por el pasante.

14. Recomendaciones

A la Empresa Estrategias Documentales SAS, seguir ofreciendo oportunidades de capacitar la mano de obra de alumnos de las universidades, instituciones educativas del país, por medio del proceso de pasantías, por otro lado recomendar se continúe con el proceso de automatización de funcionalidades, esto dado a que según la experiencia en lo trabajado, este proceso genera pruebas de regresión lo cual en caso de existir fallas en procesos se identifican a tiempo o se puede determinar un plan de acción. Otra recomendación y aspecto a tener en cuenta, es que quien ejecute o interactúe con los script desarrollados en este procesos de automatización de funcionalidades, cuente con algún tipo de conocimiento en el procesos de desarrollo, esto teniendo en cuenta que los drivers de automatización que interactúan con Selenium y el navegador, se vuelven obsoletos mes a mes con relación a las actualizaciones que el proveedor de cada navegador despliegue, así como también el lenguaje de programación puede surgir cambios con relación a las actualizaciones de Selenium, por tl razón existe el riesgo que al momento de ejecutarse alguno de los scripts puedan generar error con el paso del tiempo.

A la Universidad, continuar esforzándose cada día mas para ofrecer a los estudiantes una educación de mayor calidad, teniendo en cuenta que el ámbito laboral y de tecnología esta en constante cambios, mantenerse a la vanguardia en cuanto a las nuevas tecnologías que han surgido posterior al tema mundial de la pandemia COVID-19, continuar con la búsqueda de la calidad y cantidad de conocimientos tecnológicos, que ayuden a mejorar o afianzar el tema de soporte y

cultura general en los alumnos de la universidad del área andina.

15. Glosario

- **Selenium:** Conjunto de utilidades que facilita la labor de obtener juegos de pruebas para aplicaciones web. Para ello nos permite grabar, editar y depurar casos de prueba, que podrán ser ejecutados de forma automática e iterativa
- **Python:** Lenguaje de programación multi plataforma de código abierto y gratuito.
- **ChromeDriver:** Es un archivo .exe que utiliza su interfaz WebDriver para iniciar el navegador Google **Chrome**
- **HU:** Siglas de Historia de Usuario
- **Historia de Usuario:** Documento en el cual se levanta información correspondiente a un requerimiento o solicitud de mejora, en el cual se estable aquellos criterios de aceptación tenidos en cuenta para la correcta ejecución de la funcionalidad requerida.
- **Script:** secuencia de comando.
- **Librería:** Conjunto de archivos que se utiliza para desarrollar software. Suele estar compuesta de código y datos, y su fin es ser utilizada por otros programas de forma totalmente autónoma

Bibliografía

- @ElPSF. (Agosto de 2009). *Fundación de software de Python*. Obtenido de <https://www.python.org/doc/versions/>
- Alvarez, N. (7 de Agosto de 2021). *Curso Python con Selenium*. Obtenido de <https://www.youtube.com/playlist?list=PLas3od-GGNa2UW9-1H-NCNrUocvWD9eyh>
- Juarez, C. L.-G. (20 December 2018). *Automatización de Pruebas con Selenium y Python*.