compensar | fundación universitaria

Fecha: 17 de Abril del 2025

PROYECTO FINAL SEGUNDO CORTE

DOCUMENTACIÓN DEL PROYECTO

Estudiante

Steven Alejandro Sandoval Cardozo

Docente:

Julian David Gomez Perez

Documentación del Proyecto: Automatización de Pruebas con Selenium

Guía de Instalación y Ejecución

1. Instalar ChromeDriver si aún no está instalado en el sistema.
2. Crear un nuevo proyecto en PyCharm y configurar el entorno de desarrollo.

Descripción del Código

Modo Headless

En este proyecto, se utiliza el modo Headless para ejecutar las pruebas sin necesidad de abrir una ventana del navegador. Esto permite realizar las pruebas de manera más rápida y eficiente.

```
options = Options()
options.add_argument("--headless")
options.add_argument("--window-size=1920,1080")
driver = webdriver.Chrome(options=options)
```

Prueba 1: Registro de Usuario

Esta prueba simula el llenado de un formulario de registro de usuario en la página https://demoqa.com/automation-practice-form. Se ingresan datos como nombre, apellido, correo electrónico, número de teléfono, fecha de nacimiento, entre otros. Después de enviar el formulario, se toma una captura de pantalla para confirmar que se ha completado correctamente.

```
driver.get("https://demoqa.com/automation-practice-form")
driver.find_element(By.ID, "firstName").send_keys("Steven Alejandro")
...
```

Prueba 2: Carga de Archivos

En esta prueba, se simula la carga de un archivo desde el equipo del usuario en la página Upload and Download. El archivo se selecciona y sube al sistema. Al finalizar la carga, se toma una captura de pantalla para confirmar que el archivo ha sido correctamente subido.

```
driver.get("https://demoqa.com/upload-download")
driver.find_element(By.ID,
"uploadFile").send_keys(os.path.abspath(r"C:\Users\steve\Pictures\Python.jpg"))
```

Prueba 3: Descarga de Archivos

Se simula la descarga de un archivo llamado sampleFile.jpeg en la misma sección de Upload and Download. Luego, se verifica si el archivo realmente se ha descargado en la carpeta predeterminada de descargas del usuario.

```
driver.find_element(By.ID, "downloadButton").send_keys(Keys.ENTER)
time.sleep(5)
carpeta_usuario = os.path.expanduser("~")
ruta_archivo = os.path.join(carpeta_usuario, "Downloads", "sampleFile.jpeg")
if os.path.exists(ruta_archivo):
 print("Archivo encontrado en la carpeta: ", ruta_archivo)
else:
 print("Archivo no encontrado")
```

Prueba 4: Interacción con Alertas

En esta parte del código, se interactúa con alertas de tipo confirmación y prompt en la página Agregar algo de texto. Primero, se acepta una alerta de confirmación, y luego se captura el texto de una alerta de tipo prompt, enviando un mensaje antes de aceptarla.

```
driver.find_element(By.ID, "confirmButton").send_keys(Keys.ENTER)
WebDriverWait(driver, 10).until(EC.alert_is_present())
alerta = driver.switch_to.alert
alerta.accept()
...
```

Capturas de Pantalla

Para realizar un seguimiento visual de cada paso en las pruebas, se ha implementado una función que toma capturas de pantalla después de cada acción importante. Esto asegura que se pueda validar visualmente si el proceso ha sido completado correctamente.

```
def tomar_captura(driver, nombre_paso):
 ruta_captura = f"Screenshots/{nombre_paso}.png"
 driver.save_screenshot(ruta_captura)
 print(f"Captura Guardada en: {ruta_captura}")
```

Generación de Reportes

Este proyecto no genera un reporte HTML automáticamente, pero he utilizado capturas de pantalla tomadas durante la ejecución de las pruebas. Estas capturas pueden ser utilizadas como base para crear un reporte visual que muestre los resultados de las pruebas realizadas.

Manejo de Descargas

Las descargas se gestionan de manera automática en las carpetas de descargas predeterminadas del sistema operativo del usuario, ya sea en Windows (C:/downloads) o en Mac/Linux (~/downloads).

```
# Verifica que el archivo descargado esté en la carpeta correspondiente
carpeta_usuario = os.path.expanduser("~")
ruta_archivo = os.path.join(carpeta_usuario, "Downloads", "sampleFile.jpeg")
```

Modo Headless

El modo Headless permite que las pruebas se realicen sin abrir el navegador visualmente. Esto es útil para entornos donde no es necesario ver la interfaz gráfica del navegador y ayuda a ahorrar recursos.

```
options.add_argument("--headless")
options.add_argument("--window-size=1920,1080")
```

Video Demostración

Se incluyen dos videos como parte de la entrega: uno con el modo Headless activado y otro sin él, para que se pueda visualizar el funcionamiento del código tanto en segundo plano como de forma interactiva.

En ambos videos se muestra claramente la ejecución completa de cada una de las pruebas: el llenado del formulario de registro, la carga y descarga de archivos, así como la interacción con las alertas del sitio. Además, se puede observar cómo se generan y guardan las capturas de pantalla en cada paso importante del proceso.