

TAREA

Evaluación de Sistema QA
Semana 2



Nombre: Gestión de entorno y datos de prueba

1. Instrucciones de la Tarea

Estimado(a) estudiante:

A continuación, te presentamos la actividad evaluativa de la semana. En primer lugar, encontrarás una situación problemática o caso relacionado con el mundo laboral, la que deberás resolver respondiendo las preguntas que se presentan. Esto tiene como propósito evidenciar el logro del aprendizaje propuesto para esta semana.

Antes de comenzar considera lo siguiente:

- Esta actividad tiene un puntaje máximo de 9 puntos. Una vez que la entregues, el docente la revisará empleando la rúbrica disponible en el aula, la que contiene los indicadores de evaluación propuestos para esta semana. Por ello, te invitamos a revisarla antes de comenzar.
- Las respuestas deben ser una elaboración propia. Te puedes apoyar en los objetos virtuales de aprendizaje de la semana y en otros textos, pero siempre debes identificar a los autores y citar las fuentes.
- Debes agregar la bibliografía, en formato APA, al final de la tarea. Si necesitas recordar cómo se realiza, puedes buscar el Manual sobre Normas APA disponible en la colección CREDITI de la Biblioteca Virtual de IACC. Cuida tu redacción y ortografía.
- Si tienes dudas, puedes realizar consultas al docente en el Foro de Interacción de la semana.
- La tarea debe ser desarrollada en la plantilla establecida por IACC, disponible para ser descargada desde la plataforma de la asignatura junto a estas instrucciones.
- Envía el documento creado con tu nombre y apellido (Nombre_Apellido_Tarea2) **en formato PDF**.

2. Desarrollo de la actividad

Caso Práctico: Automatización de Pruebas para una Aplicación de Cálculo de Fuerza

En el contexto de una empresa de desarrollo de software, se ha desarrollado una aplicación *web* que calcula la fuerza de acuerdo con la fórmula de la Ley de Newton: **Fuerza = masa x aceleración**. Se le ha asignado al equipo responsable de pruebas la tarea de diseñar pruebas automatizadas para garantizar el correcto funcionamiento de esta aplicación.

Estas pruebas deben demostrar si la funcionalidad de cálculo de fuerza produce resultados precisos y coherentes.

El equipo de pruebas ha recibido el código de la aplicación, implementada en Python con Flask, y se les ha proporcionado un ejemplo de cómo interactuar con la interfaz de usuario utilizando Selenium para realizar pruebas automatizadas. Se espera que el equipo utilice este marco de trabajo para diseñar casos de prueba que evalúen el resultado obtenido frente al resultado esperado, determinando así si la prueba fue exitosa o fallida.

La aplicación web cuenta con una interfaz de usuario simple que permite al usuario ingresar la masa y la aceleración, y luego muestra el resultado del cálculo de fuerza. El equipo de pruebas debe asegurarse de que la aplicación maneje correctamente diferentes combinaciones de valores de masa y aceleración, y que genere el resultado esperado de acuerdo con la fórmula establecida.

El desafío radica en diseñar casos de prueba efectivos que cubran una amplia gama de escenarios posibles y que detecten posibles errores o anomalías en el cálculo de la fuerza. Además, el equipo debe establecer una estrategia para la gestión de datos de prueba, agregando tantas pruebas y datos como sean necesarios para validar la robustez y la precisión de la aplicación.

No olviden consultar la guía de ejercicios, donde encontrarán material práctico fundamental para abordar el ejercicio.

A continuación, se muestran los códigos fuentes entregados al equipo de prueba y algunos aspectos de interés:

1. Código fuente completo de `calculadora_fuerza.py`

`\tarea1\calculadora_fuerza.py`

```
from flask import Flask, render_template, request

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/calcular', methods=['POST'])
def calcular():
    masa = float(request.form['masa'])
    aceleracion = float(request.form['aceleracion'])
    fuerza = masa * aceleracion
    return render_template('index.html', fuerza=fuerza)

if __name__ == '__main__':
    app.run(debug=True)
```

2. Código fuente completo de la plantilla HTML (index.html)

\tarea1\templates\index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculadora de Fuerza</title>
</head>
<body>
    <h1>Calculadora de Fuerza</h1>
    <form action="/calcular" method="post">
        <label for="masa">Masa (Kg):</label><br>
        <input type="text" id="masa" name="masa"><br><br>
        <label for="aceleracion">Aceleración (m/s^2):</label><br>
        <input type="text" id="aceleracion" name="aceleracion"><br><br>
        <input type="submit" value="Calcular">
    </form>
    {% if fuerza %}
    <h2>Fuerza (N):</h2>
    <p>{{ fuerza }}</p>
    {% endif %}
</body>
</html>
```

Este programa en Python con Selenium permita realizar pruebas automatizadas de la aplicación. Es importante para el equipo, evaluar el resultado obtenido versus el resultado esperado y determinar si la prueba fue exitosa o fallida.

3.Código Python – Selenium:

```
import time
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

# Inicializar el navegador web
driver = webdriver.Chrome()

# Función para ejecutar una prueba orbital
def probar_fuerza(masa, aceleracion, resultado_esperado):
    driver.get("http://localhost:5000/") # URL de la aplicación Flask
    time.sleep(2) # Esperar a que la página se cargue completamente

    input_masa = driver.find_element("id", "masa")
    input_masa.clear()
    input_masa.send_keys(masa)

    input_aceleracion = driver.find_element("id", "aceleracion")
    input_aceleracion.clear()
    input_aceleracion.send_keys(aceleracion)

    # Hacer clic en el botón "Calcular"
    btn_calcular = driver.find_element("xpath", "//input[@type='submit']")
    btn_calcular.click()

    # Obtener el resultado de la página
    resultado_pagina = driver.find_element("xpath", "//p").text

    # Comparar el resultado obtenido con el esperado
    if resultado_pagina == resultado_esperado:
        print(f"Prueba exitosa para masa {masa} y aceleración {aceleracion}")
    else:
        print(f"La prueba falló para masa {masa} y aceleración {aceleracion}")

probar_fuerza(2, 9, "18.0")
probar_fuerza(4, 8, "32.0")
probar_fuerza(8, 10, "80.0")

# Cerrar el navegador
driver.quit()
```

Aspectos de interés:

Configuración de los datos de prueba: Los datos de prueba se reciben en la función `probar_fuerza`, los cuales consisten en los parámetros relativos a masa y aceleración, y el resultado de fuerza esperado. *Se recomienda al equipo que realice las pruebas, que agregue tantas pruebas y datos como considere.*

Salida:

```
Prueba exitosa para masa 2 y aceleración 9
Prueba exitosa para masa 4 y aceleración 8
Prueba exitosa para masa 8 y aceleración 10
```

A continuación, responde las siguientes preguntas:

1. ¿Cuáles son los componentes principales que debemos considerar al examinar los frameworks de automatización de pruebas, especialmente en el contexto de la aplicación de cálculo de fuerza?
2. ¿Cuáles son las características clave de Selenium IDE que la convierten en una herramienta eficaz para la detección de errores del software en el caso práctico de pruebas automatizadas?
3. ¿Cómo podemos aplicar los principios de desarrollo de scripts y pruebas automatizadas para mejorar la calidad del sistema de cálculo de fuerza, teniendo en cuenta los requisitos específicos y los escenarios de prueba identificados en nuestro caso práctico? Finalmente, incluye en el documento una o varias capturas de pantalla que muestren la salida después de ejecutar el programa con los datos de prueba utilizados.

Indicadores de Evaluación
Examina los frameworks de automatización de pruebas, considerando los componentes principales.
Reconoce las características de Selenium IDE, considerando su uso como herramienta de prueba automatizada que permita la detección de errores del software.
Usa los principios de desarrollo de scripts y pruebas automatizadas para mejorar la calidad del sistema.