Atividade prática de teste funcional

Alunos: João Fernando Ehlers 23317130-2, Bruno Valério Abrahim 23000333-2, Hugo Vinícius Fonseca Zuin 23000248-2 , Henrique Pacheco Alves 23293915-2

Instalação:

Instalar Playwright:

```
pip install playwright pytest
playwright install
```

Instalar Selenium:

```
pip install selenium
```

Casos de teste:

Plataforma 1 - SauceDemo

| Caso | Fluxo | Passos | Dados | Resultado Esperado

| CT01 | Login válido | Acessar página → Inserir `standard_user` / `secret_sauce` → Clicar Login | Credenciais válidas | Página de inventário carregada |

| CT02 | Login inválido | Acessar página → Inserir `user_errado` / `senha_errada` → Clicar Login | Credenciais inválidas | Mensagem de erro exibida

| CT03 | Logout | Fazer login válido → Abrir menu → Clicar Logout | - | Volta para tela de login|

Plataforma 2 - Herokuapp

| Caso | Fluxo| Passos| Dados| Resultado Esperado |

| CT04 | Login válido | Acessar página → Inserir `tomsmith` / `SuperSecretPassword!` → Clicar Login | Credenciais válidas | Mensagem de sucesso exibida |

| CT05 | Login inválido | Acessar página → Inserir `user_errado` / `senha_errada` → Clicar Login | Credenciais inválidas | Mensagem de erro exibida |

| CT06 | Logout | Fazer login válido → Clicar Logout | – | Volta para tela de login|

Plataforma 3 - Practice Test Automation

| Caso | Fluxo | Passos| Dados | Resultado Esperado |

| CT07 | Login válido | Acessar página → Inserir `student` / `Password123` → Clicar Login | Credenciais válidas | Página de “Logged In Successfully” carregada |

| CT08 | Login inválido | Acessar página → Inserir `usuarioX` / `senhaErrada` → Clicar Login | Credenciais inválidas | Mensagem “Your username is invalid!” exibida |
| CT09 | Logout | Fazer login válido → Clicar Logout | – Retorna para página de login |

Plataforma 4 - OrangeHRM Demo

| Caso | Fluxo | Passos| Dados | Resultado Esperado |

| CT10 | Login válido | Acessar página → Inserir `Admin` / `admin123` → Clicar Login | Credenciais válidas | Dashboard carregado |

| CT11 | Login inválido | Acessar página → Inserir `userErrado` / `senhaErrada` → Clicar Login | Credenciais inválidas | Mensagem de erro exibida |

| CT12 | Logout | Fazer login válido → Abrir menu do usuário → Logout | – | Volta para tela de login |

Implementação no Playwright:

SauceDemo:

```
# test_saucedemo_playwright.py
import pytest
from playwright.sync_api import sync_playwright

def test_login_valido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://www.saucedemo.com")

        page.fill("#user-name", "standard_user")
        page.fill("#password", "secret_sauce")
        page.click("#login-button")

        assert "inventory" in page.url

        # Logout
        page.click("#react-burger-menu-btn")
        page.click("#logout_sidebar_link")
        assert "saucedemo.com" in page.url
        browser.close()

def test_login_invalido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://www.saucedemo.com")

        page.fill("#user-name", "usuario_errado")
        page.fill("#password", "senha_errada")
        page.click("#login-button")

        assert "Epic sadface" in page.text_content("h3")
        browser.close()
```

Herokuapp:

```
# test_herokuapp_playwright.py
import pytest
from playwright.sync_api import sync_playwright

def test_login_valido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://the-internet.herokuapp.com/login")

        page.fill("#username", "tomsmith")
        page.fill("#password", "SuperSecretPassword!")
        page.click("button[type='submit']")

        assert "You logged into a secure area!" in page.text_content("#flash")

        # Logout
        page.click("a[href='/logout']")
        assert "Login Page" in page.text_content("h2")
        browser.close()

def test_login_invalido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://the-internet.herokuapp.com/login")

        page.fill("#username", "user_errado")
        page.fill("#password", "senha_errada")
        page.click("button[type='submit']")

        assert "Your username is invalid!" in page.text_content("#flash")
        browser.close()
```

Practice Test Automation:

```
# test_practicetest_playwright.py
import pytest
from playwright.sync_api import sync_playwright

def test_login_valido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://practicetestautomation.com/practice-test-login/")

        page.fill("#username", "student")
        page.fill("#password", "Password123")
        page.click("button[type='submit']")

        assert "Logged In Successfully" in page.text_content("h1")

        # Logout
        page.click("a[href='/practice-test-login/']")
        assert "Test login" in page.text_content("h2")
        browser.close()

def test_login_invalido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://practicetestautomation.com/practice-test-login/")

        page.fill("#username", "usuarioX")
        page.fill("#password", "senhaErrada")
        page.click("button[type='submit']")

        assert "Your username is invalid!" in page.text_content("#error")
        browser.close()
```

OrangeHRM Demo:

```
# test_orangehrm_playwright.py
import pytest
from playwright.sync_api import sync_playwright

def test_login_valido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://opensource-demo.orangehrmlive.com/")

        page.fill("input[name='username']", "Admin")
        page.fill("input[name='password']", "admin123")
        page.click("button[type='submit']")

        assert "Dashboard" in page.text_content("h6")

        # Logout
        page.click(".oxd-userdropdown-tab")
        page.click("a[href='/web/index.php/auth/logout']")
        assert "OrangeHRM" in page.title()
        browser.close()

def test_login_invalido():
    with sync_playwright() as p:
        browser = p.chromium.launch(headless=True)
        page = browser.new_page()
        page.goto("https://opensource-demo.orangehrmlive.com/")

        page.fill("input[name='username']", "userErrado")
        page.fill("input[name='password']", "senhaErrada")
        page.click("button[type='submit']")

        assert "Invalid credentials" in page.text_content(".oxd-alert-content")
        browser.close()
```

Implementação no Selenium:

SauceDemo:

```
# test_saucedemo_selenium.py
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By
```

```
def test_login_valido():
    driver = webdriver.Chrome()
    driver.get("https://www.saucedemo.com")

    driver.find_element(By.ID, "user-name").send_keys("standard_user")
    driver.find_element(By.ID, "password").send_keys("secret_sauce")
    driver.find_element(By.ID, "login-button").click()

    assert "inventory" in driver.current_url

    driver.find_element(By.ID, "react-burger-menu-btn").click()
    driver.find_element(By.ID, "logout_sidebar_link").click()

    assert "saucedemo.com" in driver.current_url
    driver.quit()

def test_login_invalido():
    driver = webdriver.Chrome()
    driver.get("https://www.saucedemo.com")

    driver.find_element(By.ID, "user-name").send_keys("usuario_errado")
    driver.find_element(By.ID, "password").send_keys("senha_errada")
    driver.find_element(By.ID, "login-button").click()

    erro = driver.find_element(By.TAG_NAME, "h3").text
    assert "Epic sadface" in erro
    driver.quit()
```

Herokuapp:

```
# test_herokuapp_selenium.py
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By

def test_login_valido():
    driver = webdriver.Chrome()
    driver.get("https://the-internet.herokuapp.com/login")

    driver.find_element(By.ID, "username").send_keys("tomsmith")
    driver.find_element(By.ID, "password").send_keys("SuperSecretPassword!")
    driver.find_element(By.CSS_SELECTOR, "button[type='submit']").click()

    assert "You logged into a secure area!" in driver.find_element(By.ID, "flash").text

    driver.find_element(By.CSS_SELECTOR, "a[href='/logout']").click()
    assert "Login Page" in driver.find_element(By.TAG_NAME, "h2").text
    driver.quit()

def test_login_invalido():
    driver = webdriver.Chrome()
    driver.get("https://the-internet.herokuapp.com/login")

    driver.find_element(By.ID, "username").send_keys("user_errado")
    driver.find_element(By.ID, "password").send_keys("senha_errada")
    driver.find_element(By.CSS_SELECTOR, "button[type='submit']").click()

    assert "Your username is invalid!" in driver.find_element(By.ID, "flash").text
    driver.quit()
```

Practice Test Automation:

```
# test_practicetest_selenium.py
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By

def test_login_valido():
    driver = webdriver.Chrome()
    driver.get("https://practicetestautomation.com/practice-test-login/")

    driver.find_element(By.ID, "username").send_keys("student")
    driver.find_element(By.ID, "password").send_keys("Password123")
    driver.find_element(By.ID, "submit").click()

    assert "Logged In Successfully" in driver.find_element(By.TAG_NAME, "h1").text

    driver.find_element(By.LINK_TEXT, "Log out").click()
    assert "Test login" in driver.find_element(By.TAG_NAME, "h2").text
    driver.quit()

def test_login_invalido():
    driver = webdriver.Chrome()
    driver.get("https://practicetestautomation.com/practice-test-login/")

    driver.find_element(By.ID, "username").send_keys("usuarioX")
    driver.find_element(By.ID, "password").send_keys("senhaErrada")
    driver.find_element(By.ID, "submit").click()

    assert "Your username is invalid!" in driver.find_element(By.ID, "error").text
    driver.quit()
```

OrangeHRM Demo:

```python
# test_orangehrm_selenium.py
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By

def test_login_valido():
    driver = webdriver.Chrome()
    driver.get("https://opensource-demo.orangehrmlive.com/")

    driver.find_element(By.NAME, "username").send_keys("Admin")
    driver.find_element(By.NAME, "password").send_keys("admin123")
    driver.find_element(By.CSS_SELECTOR, "button[type='submit']").click()

    assert "Dashboard" in driver.find_element(By.TAG_NAME, "h6").text

    driver.find_element(By.CLASS_NAME, "oxd-userdropdown-tab").click()
    driver.find_element(By.LINK_TEXT, "Logout").click()

    assert "OrangeHRM" in driver.title
    driver.quit()

def test_login_invalido():
    driver = webdriver.Chrome()
    driver.get("https://opensource-demo.orangehrmlive.com/")

    driver.find_element(By.NAME, "username").send_keys("userErrado")
    driver.find_element(By.NAME, "password").send_keys("senhaErrada")
    driver.find_element(By.CSS_SELECTOR, "button[type='submit']").click()

    erro = driver.find_element(By.CLASS_NAME, "oxd-alert-content").text
    assert "Invalid credentials" in erro
    driver.quit()
```