

PYTHON PARA QAs



REVISÃO AULA 8

- O que é Business Intelligence (BI)?
- Sincronização de dados JSON ↔ Google Sheets
- Introdução ao Google Data Studio (Looker)

MÓDULO 5 - INTRODUÇÃO AO SELENIUM

1. **Fundamentos do Selenium WebDriver e configuração inicial.**
2. **Inspeção de elementos DOM e criação de scripts básicos de automação.**
3. **Relação entre HTML, CSS e JavaScript no contexto de automação**

AULA 9 - INTRODUÇÃO AO SELENIUM

Objetivos da aula:

- Compreender o que é automação de testes com Selenium e seus benefícios.
- Configurar corretamente o Selenium WebDriver para diferentes navegadores.
- Criar scripts de automação para acessar páginas e interagir com elementos web.
- Inspecionar elementos HTML e utilizar estratégias avançadas de localização.
- Aprender a lidar com diferentes tipos de elementos (inputs, botões, links, dropdowns, checkboxes, alerts, iframes).
- Executar testes sincronizados com waits explícitos e implícitos.

SELENIUM

- O que é Selenium?
 - Selenium WebDriver é uma ferramenta de automação para navegadores.
 - Amplamente usado em testes funcionais e regressivos.
 - Usado para testes de interface web (front-end).
 - Suporta diversos navegadores: Chrome, Firefox, Edge, Safari.
 - Integração com frameworks de testes como pytest e unittest.
- Por que usar Selenium?
 - Simula ações reais de usuários (cliques, digitação, scroll).
 - Reduz tempo e esforço em testes manuais repetitivos.
 - Detecta falhas rapidamente em diferentes ambientes e navegadores.
 - Facilita a automação de fluxos de teste críticos.

SELENIUM

- Onde o Selenium NÃO funciona?
 - Não é recomendado para automação de testes mobile (para isso, usa-se Appium).
 - Não é ideal para testes unitários, pois é mais pesado que frameworks como pytest.
- Exemplo de caso real:
 - Automação de testes em uma plataforma de e-commerce, verificando se a compra funciona corretamente em diferentes navegadores.

CONFIGURAÇÃO DO SELENIUM WEBDRIVER

- Instalar o Selenium via pip
- Baixar o WebDriver do Chrome
 - Acesse: <https://chromedriver.chromium.org/downloads>
 - Escolha a versão compatível com seu navegador.
 - Adicione o caminho do WebDriver ao PATH do sistema.
- Baixar o WebDriver do Firefox, baixe o GeckoDriver:
 - Acesse: <https://github.com/mozilla/geckodriver/releases>.

INSPECIONANDO ELEMENTOS NA PÁGINA

- Como encontrar elementos em uma página web?
 - Usando o DevTools do navegador (F12 → Inspeccionar Elemento).
 - Principais estratégias de localização:
 - ID: `find_element(By.ID, "campo")`
 - Class Name: `find_element(By.CLASS_NAME, "classe")`
 - Name: `find_element(By.NAME, "nome")`
 - XPath: `find_element(By.XPATH, "//tag[@atributo='valor']")`
 - CSS Selector: `find_element(By.CSS_SELECTOR, "seletor-css")`

INTERAGINDO COM ELEMENTOS WEB

- Interações possíveis com Selenium:
 - Digitar em campos de input
 - Clicar em botões e links
 - Selecionar opções em dropdowns
 - Marcar checkboxes e radio buttons
 - Lidar com alertas e pop-ups
- Desafio:
 - Criar um teste automatizado que preencha um formulário e clique em um botão.
 - Rodar o script e verificar os resultados na página web.

ESPERAS E SINCRONIZAÇÃO NO SELENIUM

- Por que precisamos de esperas?
 - Algumas ações ocorrem antes que o elemento esteja carregado, causando falhas nos testes.
 - Para evitar isso, usamos waits implícitos e explícitos.
- Desafio:
 - Criar um script que aguarde um elemento carregar antes de interagir com ele.

ATIVIDADE

- Criar um teste automatizado completo;
 - Automação de um formulário no site <https://the-internet.herokuapp.com/login>
 - O usuário insere um nome e email em um formulário de contato e clica em "Enviar";
 - O Selenium verifica se uma mensagem de sucesso aparece na tela.
- Passos:
 - Abrir o navegador com Selenium;
 - Acesse um formulário;
 - Preencha os campos necessários;
 - Envie os dados;
 - Valide o resultado da submissão.

REVISÃO

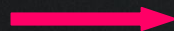
- O que aprendemos?
 - O que é Selenium WebDriver e como ele funciona;
 - Como instalar e configurar o Selenium no ambiente de desenvolvimento;
 - Como localizar e interagir com elementos web;
 - Como evitar falhas com espera implícita e explícita;
 - Criamos nossos primeiros testes automatizados.

DESAFIO PARA CASA

- Objetivo:
 - Criar um teste automatizado que preencha e interaja com um formulário no <https://the-internet.herokuapp.com/login>
 - Validar a execução correta da automação;
- Descrição da Tarefa:
 - Criar um script Selenium que:
 - Acesse a página de Login.
 - Preencha os campos de mensagem com um texto personalizado.
 - Clique no botão "Login".
 - Capture e valide a mensagem exibida.
- Requisitos:
 - O script deve executar as ações automaticamente.
 - O aluno deve validar se a mensagem foi exibida corretamente.
- Dicas:
 - Use as estratégias de localização aprendidas na aula (By.ID, By.NAME, By.XPATH).
 - Tente usar esperas (waits) se necessário para garantir que o elemento seja carregado antes da interação.

Podemos contar com o seu feedback?

Escaneie o QR Code ao lado e responda nossa Pesquisa de Avaliação.



OBRIGADO



/apsferreira



/apsferreira_



alura + FIA/P

PARA EMPRESAS

Copyright © 2019 | Professor (a) Nome do Professor

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.