**Texto

Descrição gerada automaticamenteUma imagem contendo Logotipo

Descrição gerada automaticamente**

**INSTITUTO FEDERAL DO PARANÁ – IFPR**

*Campus* Cascavel

Curso: TADS

Discente: Ana Caroline Pedrosa e Silva

Componente Curricular: Projeto Integrador - PI

Docente: Edmar André Bellorini

Data: 28 - 09 -2024

**Selenium**: **por dentro das Automações Web**

Esse material tem como função descrever os conceitos que serão abordados durante a apresentação do seminário.

1. **AUTOMAÇÃO WEB**

A automação web refere-se ao processo de usar ferramentas e scripts para executar tarefas em aplicações web de forma automatizada, em vez de fazê-las manualmente. Isso pode incluir ações como clicar em botões, preencher formulários, navegar entre páginas e verificar resultados.

1.2 OBJETIVOS DA AUTOMAÇÃO WEB

* **Eficiência**: Reduzir o tempo e o esforço necessário para realizar tarefas repetitivas;
* **Consistência**: Garantir que as tarefas sejam executadas da mesma forma toda vez, minimizando erros humanos;
* **Testes Automatizados**: Facilitar a execução de testes em aplicações web, permitindo verificar se as funcionalidades estão funcionando conforme o esperado após alterações no código.

1. **SELENIUM**

O Selenium é um conjunto de ferramentas que ajuda a automatizar navegadores da web. Em essência, ele permite que você controle o navegador de forma remota, simulando como um usuário interage com ele. Isso significa que você pode fazer coisas como clicar em botões, preencher formulários e navegar em sites automaticamente, em vez de fazer isso manualmente.

1. **WEB DRIVER**

O WebDriver é uma API, compacta orientada a objetos, que permite automatizar a interação com navegadores web. Ele fornece uma interface que os desenvolvedores usam para controlar um navegador, permitindo a execução de ações como clicar em botões, preencher formulários e navegar entre páginas.

3.1 DRIVERS

Cada navegador (como Chrome, Firefox, Safari, etc.) possui uma implementação específica do WebDriver, conhecida como **driver**. Esses drivers são programas que atuam como intermediários entre o Selenium e o navegador.

* **Função**: Ao enviar um comando para o Selenium, ele não se comunica diretamente com o navegador. Em vez disso, o Selenium envia esse comando para o driver correspondente ao navegador que você está usando. Dessa maneira, ele consegue traduzir os comandos do Selenium em instruções que o navegador entende. Isso inclui manipular a interface do usuário, acessar elementos na página e realizar ações. Depois da execução dos comandos, o navegador pode enviar uma resposta de volta para o driver, e esse repassa para o selenium. Assim como o ilustrado na Figura 01.

3.2 ELEMENTOS WEB - ESTRATÉGIAS DE LOCALIZAÇÃO

Um **localizador** é uma forma de identificar elementos em uma página, utilizado como argumento nos métodos de busca de elementos. O selenium oferece 8 formas diferentes de localizar os elementos web.

3.3 ESTRATÉGIAS DE ESPERA

Um dos principais desafios na automação de testes com Selenium é garantir que a página da web esteja pronta antes que você tente realizar uma ação. Isso é chamado de ***race condition***.

* ***Race Condition*:** Em alguns momentos, o navegador pode estar carregado e pronto, mas os elementos que você deseja interagir (como botões ou campos de texto) ainda podem não estar disponíveis. Isso acontece porque a página pode carregar lentamente, ou porque alguns elementos são adicionados ou alterados por JavaScript depois que a página já foi carregada.
* **Comportamento Variável:** Em algumas execuções de teste, tudo funciona como deveria, e os elementos estão prontos. Em outras, o Selenium tenta interagir com elementos que ainda não foram carregados na página, resultando em falhas. Isso faz com que os testes se tornem "instáveis", ou seja, falhem em algumas execuções e passem em outras.
* **Espera implícita:** O selênio tem uma maneira interna de esperar automaticamente por elementos chamados de implicitly\_wait(). A espera implícita é uma configuração que você pode definir para que o Selenium aguarde um certo tempo sempre que estiver tentando encontrar um elemento na página.

Sempre que o Selenium tenta localizar um elemento, ele vai esperar até o tempo definido antes de falhar. Se o elemento aparecer antes do tempo acabar, o Selenium irá continuar a execução imediatamente.

* **Espera explícita:** A espera explícita é uma forma de aguardar que uma condição específica seja atendida antes de prosseguir com a execução do código. Isso é útil em situações onde os elementos da página podem demorar a carregar ou mudar de estado. A espera explícita funciona como um loop que verifica repetidamente uma condição (como a visibilidade de um elemento) até que ela seja verdadeira ou até que um tempo limite seja alcançado.

1. **ATIVIDADE:** automatizar o processo de cadastro em um site.

REFERÊNCIAS:

Selenium. **Selenium Documentation**. Disponível em: <https://www.selenium.dev/pt-br/documentation/overview/>. Acesso em: 29 set. 2024.

Selenium. **WebDriver Documentation**. Disponível em:<https://www.selenium.dev/pt-br/documentation/webdriver/>. Acesso em: 29 set. 2024.

Selenium**.** **Selenium WebDriver**: Waits. Disponível em:<https://www.selenium.dev/documentation/webdriver/waits/>. Acesso em: 29 set. 2024.