

Automação de Testes em Aplicações Web

Capacitar os alunos a compreender e aplicar os conceitos fundamentais e avançados relacionados à automação de testes em aplicações web, fornecendo assim uma base sólida de conhecimentos teóricos e práticos para que os alunos possam desenvolver e implementar testes automatizados de alta qualidade, eficiência e confiabilidade em projetos web.

Prof. Samuel Lucas

Ementa e Horários

Automação de Testes em Aplicações Web

Introdução a Aplicações Web
HTML para construção das páginas
CSS para estilização das páginas
Javascript para interação com usuário e servidor
A interação do Frontend com o Backend
Requisições Síncronas e Assíncronas
CSS Selector e XPath
Fundamentos e conceitos sobre Automação de Testes Web
Adicionando bibliotecas para estruturação dos testes e interação com navegadores Web
Escrevendo o primeiro teste automatizado Web
Estratégias de Identificação de Elementos
Esperas Explícitas e Implícitas
Comboboxes, Janelas e Drag and Drop
Tirando Screenshots
Execução de Testes Headless
Design Patterns para Automação Web
Configurações Avançadas
Execução de testes na nuvem
Relatórios de execução de testes
Boas práticas no uso da ferramenta

Datas:

19 e 20 de Setembro
03 e 04 de Outubro
17 e 18 de Outubro

Horários:

Sextas

19h00 - 20h30; 20h45 - 22h00

Sábados

09h00 - 10h30; 10h45 - 12h00;
13h30 - 15h00; 15h15 - 16h30;
16h45 - 18h00

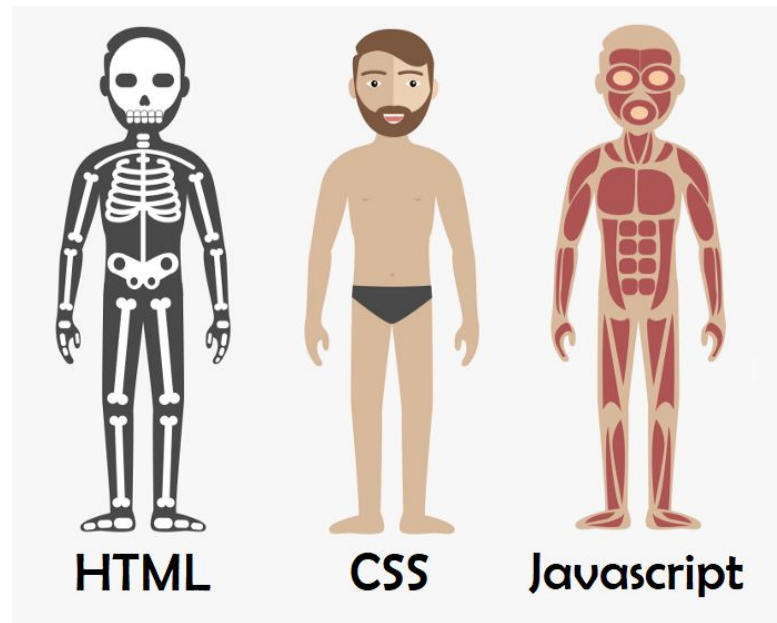
- 1 Introdução a Aplicações Web
- 2 Automação de Testes Web
- 3 Revisão do módulo
- 4 Avaliação final

Introdução a Aplicações Web

HTML, CSS e Javascript

Introdução a Aplicações Web

São as três principais linguagens utilizadas no *front-end* da nossa aplicação web. Cada uma possui uma responsabilidade específica na página.



HTML para construção de páginas

Introdução a Aplicações Web

HTML (*Linguagem de Marcação de HiperTexto*) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web.

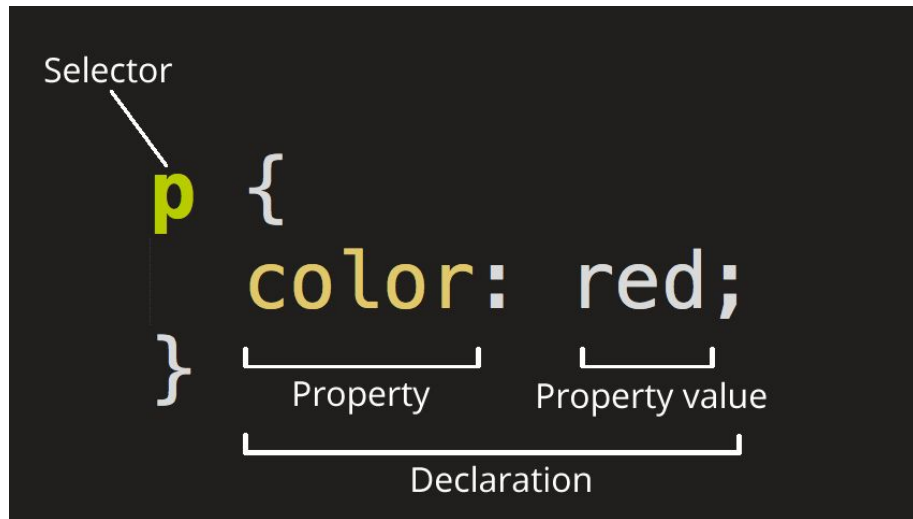
List of HTML Tags

- | | | | | |
|-----------------------------------|-----------------------------------|---|---------------------------------|---------------------------------|
| • <code><!--></code> | • <code><cite></code> | • <code><frameset></code> | • <code><object></code> | • <code><style></code> |
| • <code><!DOCTYPE></code> | • <code><code></code> | • <code><h1> to <h6></code> | • <code></code> | • <code><sub></code> |
| • <code><a></code> | • <code><col></code> | • <code><head></code> | • <code><optgroup></code> | • <code><summary></code> |
| • <code><abbr></code> | • <code><colgroup></code> | • <code><header></code> | • <code><option></code> | • <code><sup></code> |
| • <code><acronym></code> | • <code><data></code> | • <code><hr></code> | • <code><output></code> | • <code><svg></code> |
| • <code><address></code> | • <code><datalist></code> | • <code><html></code> | • <code><p></code> | • <code><table></code> |
| • <code><applet></code> | • <code><dd></code> | • <code><i></code> | • <code><param></code> | • <code><tbody></code> |
| • <code><area></code> | • <code></code> | • <code><iframe></code> | • <code><picture></code> | • <code><td></code> |
| • <code><article></code> | • <code><details></code> | • <code></code> | • <code><pre></code> | • <code><template></code> |
| • <code><aside></code> | • <code><dfn></code> | • <code><input></code> | • <code><progress></code> | • <code><textarea></code> |
| • <code><audio></code> | • <code><dialog></code> | • <code><ins></code> | • <code><q></code> | • <code><tfoot></code> |
| • <code></code> | • <code><dir></code> | • <code><kbd></code> | • <code><rp></code> | • <code><th></code> |
| • <code><base></code> | • <code><div></code> | • <code><label></code> | • <code><rt></code> | • <code><thead></code> |
| • <code><basefont></code> | • <code><dl></code> | • <code><legend></code> | • <code><ruby></code> | • <code><time></code> |
| • <code><bdi></code> | • <code><dt></code> | • <code></code> | • <code><s></code> | • <code><title></code> |
| • <code><bdo></code> | • <code></code> | • <code><link></code> | • <code><samp></code> | • <code><tr></code> |
| • <code><big></code> | • <code><embed></code> | • <code><main></code> | • <code><script></code> | • <code><track></code> |
| • <code><blockquote></code> | • <code><fieldset></code> | • <code><map></code> | • <code><section></code> | • <code><tt></code> |
| • <code><body></code> | • <code><figcaption></code> | • <code><mark></code> | • <code><select></code> | • <code><u></code> |
| • <code>
</code> | • <code><figure></code> | • <code><meta></code> | • <code><small></code> | • <code></code> |
| • <code><button></code> | • <code></code> | • <code><meter></code> | • <code><source></code> | • <code><var></code> |
| • <code><canvas></code> | • <code><footer></code> | • <code><nav></code> | • <code></code> | • <code><video></code> |
| • <code><caption></code> | • <code><form></code> | • <code><noframes></code> | • <code><strike></code> | • <code><wbr></code> |
| • <code><center></code> | • <code><frame></code> | • <code><noscript></code> | • <code></code> | |

CSS para estilização das páginas

Introdução a Aplicações Web

CSS (Folhas de Estilo em Cascata) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML.



JS para interação com usuário e servidor

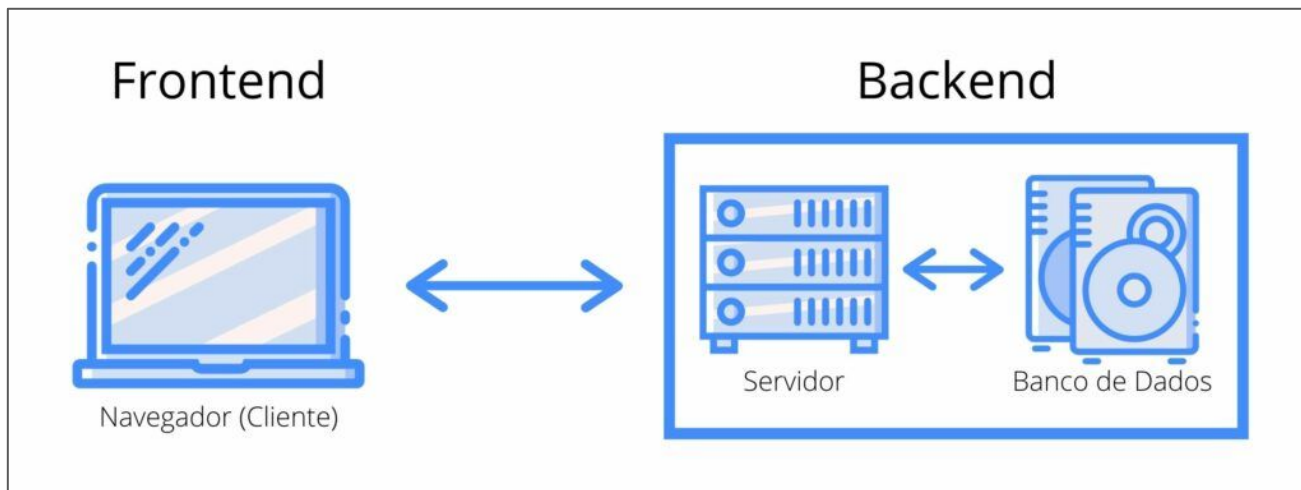
Introdução a Aplicações Web

Javascript é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Website</title>
    <script>
      function changeBackground() {
        document.body.style.backgroundColor = "blue";
      }
    </script>
  </head>
  <body onload="changeBackground()">
    <h1>Welcome to my website!</h1>
  </body>
</html>
```


A interação do *frontend* com o *backend*

Introdução a Aplicações Web



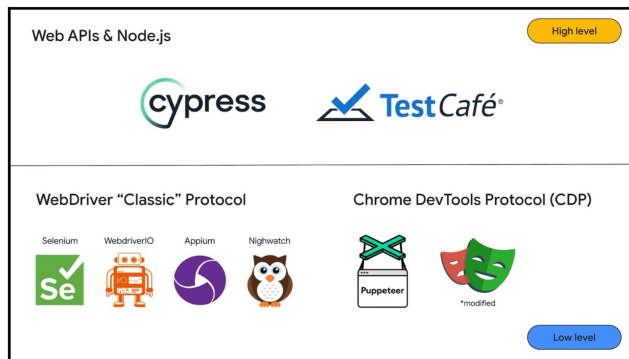
Automação de Testes Web

Fundamentos sobre Automação de Testes Web

Automação de Testes Web - Arquitetura

As ferramentas de automação de testes para aplicações web possuem, pelo menos, 3 abordagens arquiteturais para interagir com a aplicação que está executando no navegador.

Conhecer sobre a arquitetura das ferramentas nos ajuda a compreender suas virtudes e limitações, por isso, vamos conhecer cada uma com mais profundidade.



Baseado no
Protocolo WebDriver

Baseado no CDP
Chrome DevTools
Protocol

Javascript no
navegador como
aplicação

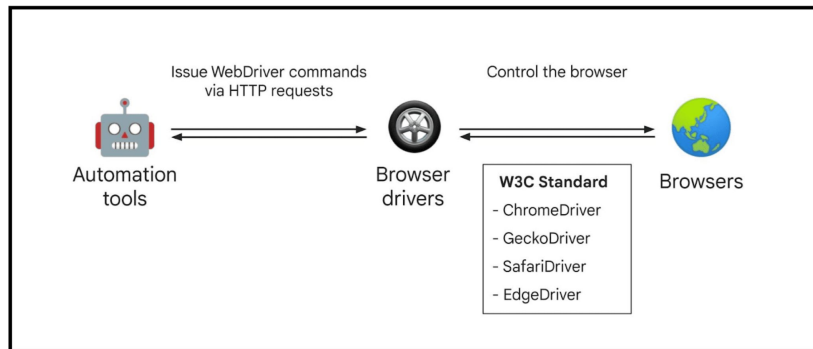
Fundamentos sobre Automação de Testes Web

Automação de Testes Web - Arquitetura

O protocolo WebDriver é um protocolo HTTP que envia comandos do *script de testes* para o navegador. Com o tempo, o WebDriver se tornou um padrão W3C.

Para usar este protocolo, você precisa de um servidor proxy que traduz os comandos e os executa no ambiente alvo, ou seja, no navegador.

Para automação web, esse proxy é o driver do próprio navegador. Cada navegador fornece um *Driver* em que o WebDriver usa para se comunicar.



Baseado no
Protocolo WebDriver

Baseado no CDP
Chrome DevTools
Protocol

Javascript no
navegador como
aplicação

Fundamentos sobre Automação de Testes Web

Automação de Testes Web - Arquitetura

O CDP usa conexão WebSocket para comunicar o script de testes ao navegador. Esse protocolo é mantido pelo Chrome e é parcialmente implementado pelo Firefox.

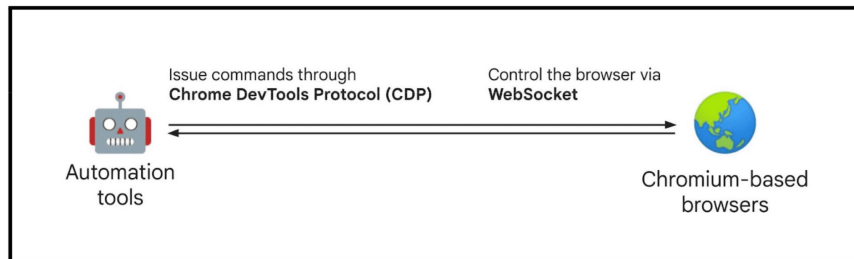
O CDP foi criado para atender a todas as necessidades de *debug* e oferece suporte a mais controles de baixo nível em relação ao WebDriver.

Ele consegue gerenciar recursos como: Interceptar solicitações de rede, modo do dispositivo, geolocalização, dentre outros.

Baseado no
Protocolo WebDriver

Baseado no CDP
Chrome DevTools
Protocol

Javascript no
navegador como
aplicação



Fundamentos sobre Automação de Testes Web

Automação de Testes Web - Arquitetura

Ferramentas que executam Javascript no navegador. Por exemplo, o Cypress utiliza APIs da Web e Node.js para executar testes diretamente no navegador.

Baseado no
Protocolo WebDriver

Baseado no CDP
Chrome DevTools
Protocol

Curiosidades:

- A primeira versão do Selenium também usou essa abordagem
- As primeiras versões do TestCafe rodavam inteiramente no navegador

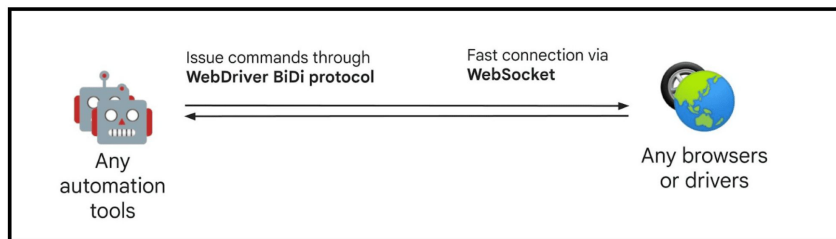
Javascript no
navegador como
aplicação

Fundamentos sobre Automação de Testes Web

Automação de Testes Web - Arquitetura

O WebDriver Bidi é um novo protocolo padrão de automação de navegador que está em desenvolvimento e busca combinar o melhor do WebDriver "clássico" do CDP.

O WebDriver BiDi promete comunicação bidirecional, tornando-a rápida por padrão, e vem com controle de baixo nível. A visão por trás do WebDriver BiDi é permitir que você escreva testes usando qualquer ferramenta, em qualquer navegador ou driver, oferecendo flexibilidade.



Baseado no
Protocolo WebDriver

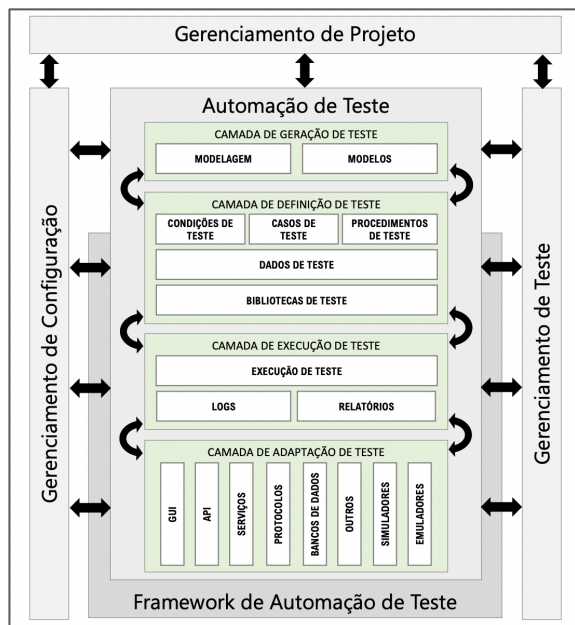
Novo protocolo
WebDriver BiDi

Baseado no CDP
Chrome DevTools
Protocol

Javascript no
navegador como
aplicação

Bibliotecas e Frameworks para Automação Web

Automação de Testes Web



CSS Selector e XPath

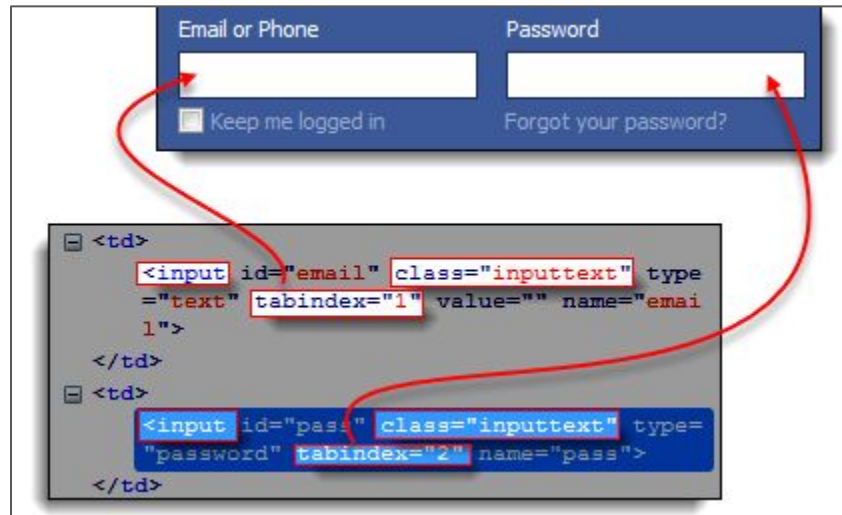
Introdução a Aplicações Web

Seletores CSS é uma maneira de definir a quais elementos um conjunto de regras CSS se aplica.

No contexto de automação web, servem para definir qual elemento um determinado comando será executado.

Seletores Básicos:

- Tag - input, button
- Classe - .primary-button
- ID - #botao-login
- Atributo - [placeholder="Entrar"]
- Universais - * (asterisco)
- Dedicados à testes - [test-id="tst-login-button"]



CSS Selector e XPath

Introdução a Aplicações Web

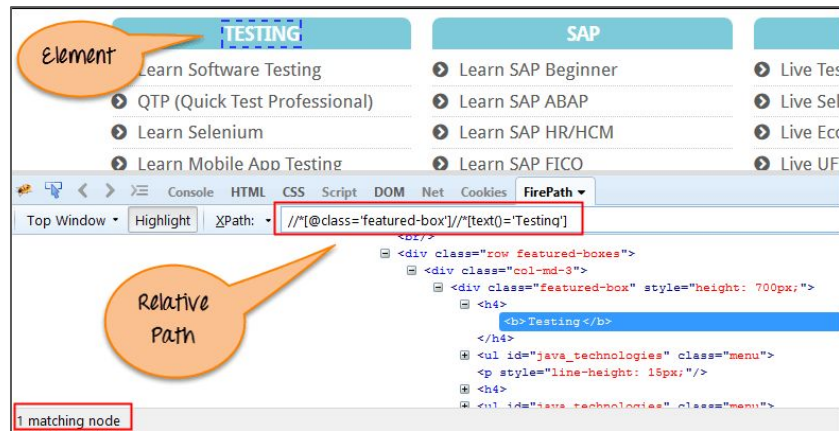
XPath (Linguagem de Caminho XML) é uma maneira de navegar na estrutura XML de um documento HTML e encontrar elementos, similar ao uso do css selector.

É particularmente útil em cenários complexos, onde recursos com id, classe ou atributos não são suficientes para identificar um elemento.

Conceitos:

- Axes; Ancestor, Parent, Child, Siblings, Ancestors, Descendants
- Absoluto, Relativo, Exato
- Funções *text()* e *contains()*

Expression	Description
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

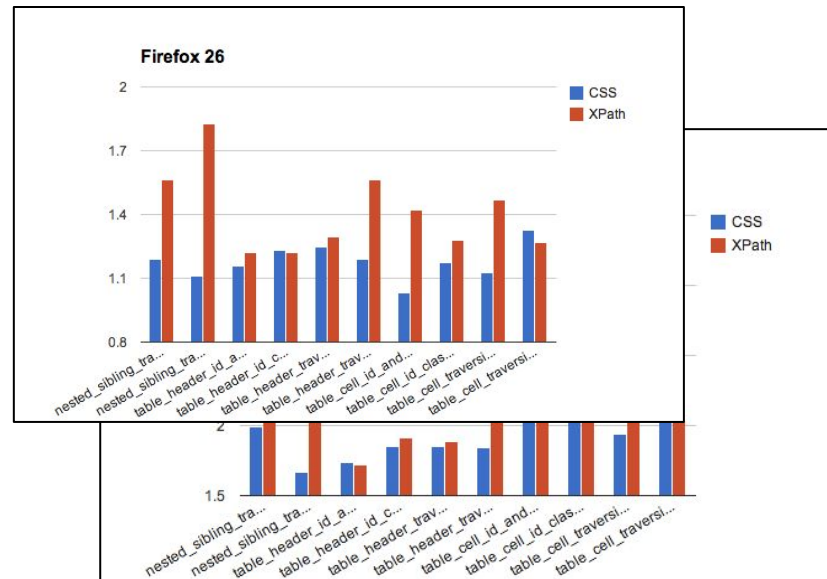


CSS Selector e XPath: qual é mais *rápido*?

Introdução a Aplicações Web

CSS na maioria das análises comparativas (benchmarks) demonstra ser mais performático que o XPath.

Entretanto, a diferença costuma ser quase que irrelevante - entre 0,1 e 0,3 milisegundos.



Seletores contextuais

Introdução a Aplicações Web

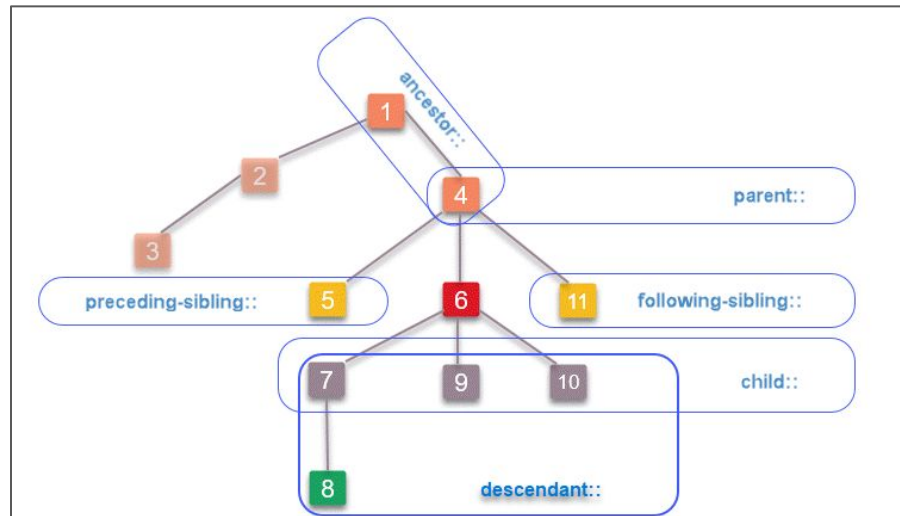
Assim como no XPath, algumas ferramentas possuem recursos para navegação na *árvore* de elementos. É possível navegar para elementos em um nível acima, abaixo, no mesmo, dentre outras variações. Normalmente, isso minimiza o uso de XPath.

Exemplos:

- Cypress - `.first`, `.last`, `.siblings`, `.parent`, `.children`
- Playwright - `.first`, `.last`,

Exemplos:

- Cypress - `first`, `last`, `siblings`, `parent`, `children`
- Playwright - `first`, `last`



Setup da Ferramenta / Hands On

Setup da ferramenta que vamos utilizar

Automação de Testes Web

Nesta disciplina, vamos utilizar o Cypress para ilustrar e exercitar os conceitos de aula.

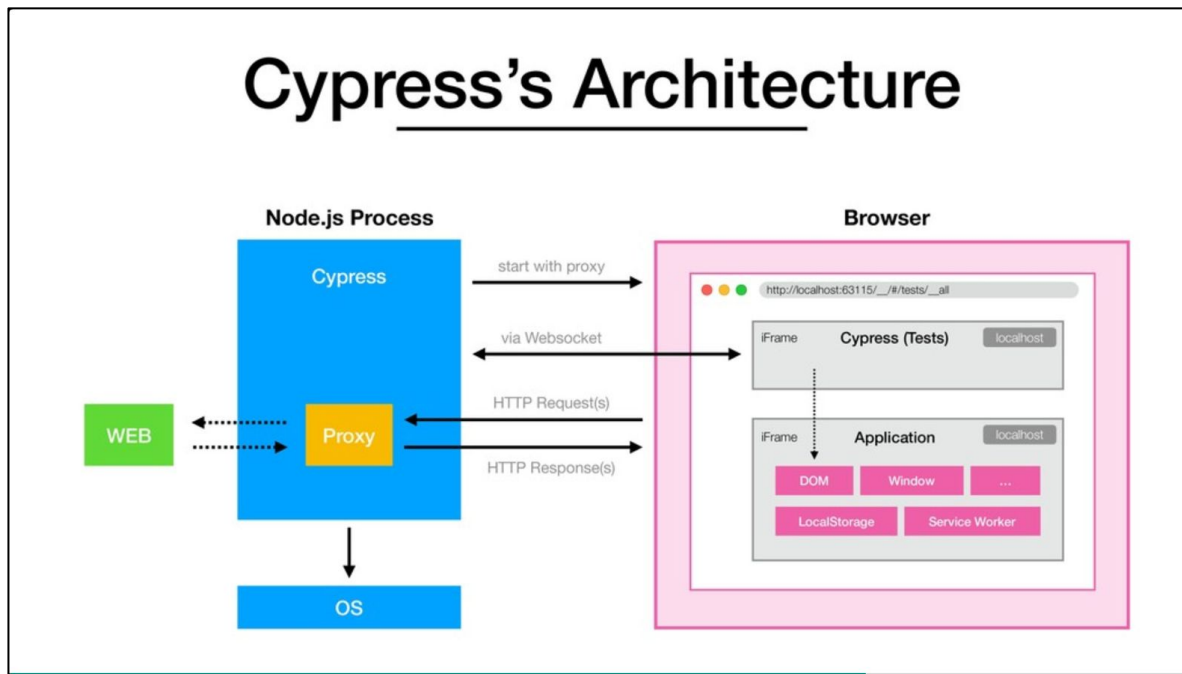
Pré-requisitos:

- Node.js 22.x ou superior
- Visual Studio Code
- Extensões do VSCode (opcional)
 - ES6 Mocha Snippets
 - Cypress Snippets (Andrew Smith)



Adicionando bibliotecas para interação com a Web

Automação de Testes Web



Adicionando bibliotecas para interação com a Web

Automação de Testes Web

Passo a Passo

- Criar nosso diretório (`mkdir` ou manual)
- Abrir no editor de textos (`code .` ou manual)
- Inicializar um projeto npm (`npm init --yes`)
- Instalar o Cypress v13 (ou última, se quiser)
 - `npm install -D cypress`
- Abrir e iniciar a config do projeto
 - `npx cypress open`
- Avançar até executar o primeiro teste



Escrevendo o primeiro testes automatizado Web

Automação de Testes Web



Hands on

Esperas Explícitas e Implícitas

Automação de Testes Web

Ambos são conceitos que vem do Selenium, uma das primeiras bibliotecas dedicadas a interação aplicações Web.

Espera Implícita é o tempo máximo pelo qual a ferramenta deve aguardar enquanto estiver executando um comando. Normalmente, é uma configuração global.

Espera Explícita é uma condição pelo qual a ferramenta deve aguardar antes de prosseguir. Pode ser de comportamento ou tempo. Exemplo, *esperar que o elemento esteja visível*; ou *esperar por 5 segundos*.

Ferramentas Modernas como Cypress e Playwright possuem também um mecanismo chamado *Espera Automática*, que executa algumas verificações de forma automática a cada comando, como visibilidade e estabilidade.

Here is the complete list of actionability checks performed for each action:

Action	Visible	Stable	Receives Events	Enabled	Editable
<code>locator.check()</code>	Yes	Yes	Yes	Yes	-
<code>locator.click()</code>	Yes	Yes	Yes	Yes	-
<code>locator.dblclick()</code>	Yes	Yes	Yes	Yes	-
<code>locator.setChecked()</code>	Yes	Yes	Yes	Yes	-
<code>locator.tap()</code>	Yes	Yes	Yes	Yes	-

```
// Set the implicit wait time to 2 seconds  
Cypress.config('defaultCommandTimeout', 2000)
```

```
// Wait until an element becomes visible  
cy.wait('#some-element').should('be.visible')
```

Screenshots

Automação de Testes Web

Na conclusão de cada ciclo de teste, os resultados do teste devem ser analisados. Durante a análise, um dos artefatos mais importantes são os screenshots (capturas de tela).

As situações mais comuns do uso são:

Durante o teste

Cypress - `cy.screenshot`
Playwright - `page.screenshot`

Quando há falha

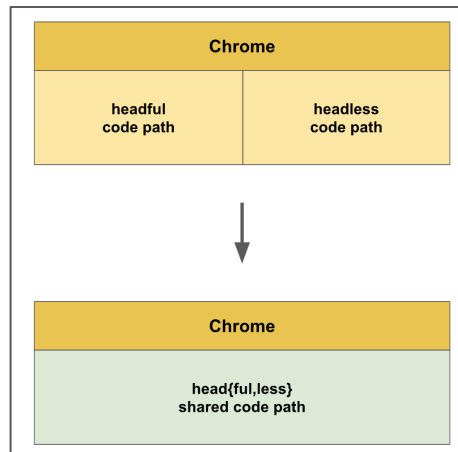
Cypress `screenshotOnRunFailure`

A cada comando

CodeceptJS `stepByStepReport`

Execução de Testes Headless

Automação de Testes Web



antigo vs novo headless no Chrome

O modo headless é, basicamente, executar o navegador sem nenhuma interface visível. No passado, usar o modo *headless* exigia alguns cuidados, dada a sua diferença em relação ao modo "headed" - com a interface visível.

Por exemplo, o antigo modo headless que o Google lançou em 2017 para o Chrome 59 era uma implementação de navegador alternativa e separada do Chrome.

A atualização de 2023 para o navegador headless do Google Chrome mesclou os caminhos de código dos dois modos, onde a única diferença é que no modo headless, o Chrome cria, mas não exibe nenhuma janela do navegador.

Execução de Testes Headless

Automação de Testes Web

npx cypress run

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Usando o projeto e conceitos que aprendemos em aula, implemente os 5 primeiros cenários da lista na página *testcases* do site usado em aula.

(não vale nota, mas, é importante para fixar)

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Analise os testes criados para uma aplicação e que estão com falhas diversas, de sintaxe, funcionalidade, seletores. Execute, Identifique a origem do erro e corrija os testes falhando.

Link:

<https://gist.github.com/samlucax/0df852881249b561cdf8888493b03125>

(não vale nota, mas, é importante para fixar)

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Instale a biblioteca *cypress-xpath* no projeto e tente adaptar os seletores para usar xpath.

Observe aspectos como legibilidade, velocidade de execução e compartilhe suas percepções na aula.

(não vale nota, mas, é importante para fixar)

Resumo

Resumo

Automação de Testes em Aplicações Web

Introdução a Aplicações Web

HTML para construção das páginas

CSS para estilização das páginas

Javascript para interação com usuário e servidor

A interação do Frontend com o Backend

Requisições Síncronas e Assíncronas

CSS Selector e XPath

Fundamentos e conceitos sobre Automação de Testes Web

Adicionando bibliotecas para estruturação dos testes e interação com navegadores Web

Escrevendo o primeiro teste automatizado Web

Estratégias de Identificação de Elementos

Esperas Explícitas e Implícitas

Tirando Screenshots

Execução de Testes Headless

Próximas aulas

Próximas aulas

Automação de Testes em Aplicações Web

Comboboxes, Janelas e Drag and Drop

Design Patterns para Automação Web

Configurações Avançadas

Execução de testes na nuvem

Relatórios de execução de testes

Boas práticas no uso da ferramenta

Trabalho de Conclusão da disciplina

Perguntas e Respostas

Até a próxima aula

Automação de Testes em Aplicações Web

Capacitar os alunos a compreender e aplicar os conceitos fundamentais e avançados relacionados à automação de testes em aplicações web, fornecendo assim uma base sólida de conhecimentos teóricos e práticos para que os alunos possam desenvolver e implementar testes automatizados de alta qualidade, eficiência e confiabilidade em projetos web.

Prof. Samuel Lucas

Ementa e Horários

Automação de Testes em Aplicações Web

Introdução a Aplicações Web
HTML para construção das páginas
CSS para estilização das páginas
Javascript para interação com usuário e servidor
A interação do Frontend com o Backend
Requisições Síncronas e Assíncronas
CSS Selector e XPath
Fundamentos e conceitos sobre Automação de Testes Web
Adicionando bibliotecas para estruturação dos testes e interação com navegadores Web
Escrevendo o primeiro teste automatizado Web
Estratégias de Identificação de Elementos
Esperas Explícitas e Implícitas
Comboboxes, Janelas e Drag and Drop
Tirando Screenshots
Execução de Testes Headless
Design Patterns para Automação Web
Configurações Avançadas
Execução de testes na nuvem
Relatórios de execução de testes
Boas práticas no uso da ferramenta

Datas:

19 e 20 de Setembro
03 e 04 de Outubro
17 e 18 de Outubro

Horários:

Sextas

19h00 - 20h30; 20h45 - 22h00

Sábados

09h00 - 10h30; 10h45 - 12h00;
13h30 - 15h00; 15h15 - 16h30;
16h45 - 18h00

Revisando os exercícios

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Usando o projeto e conceitos que aprendemos em aula, implemente os 5 primeiros cenários da lista na página *testcases* do site usado em aula.

(não vale nota, mas, é importante para fixar)

Debugging e Análise de Falhas

Debugging e Análise de Falhas

Automação de Testes Web

- Logs
 - Logs no teste (cy.log, comandos)
 - Logs no terminal (andamento do teste)
 - Logs no navegador (console.log)
- Screenshots e Vídeos
 - cy.screenshot
 - config de vídeo
 - screenshots da falha
- **StackTrace** (caminho do erro)

Principais motivos de falhas

Automação de Testes Web

- **Element Not Found** / Elemento Não Encontrado
 - **Assertion Error** / Erro de Asserção
 - **Command Timeout** / Comando atingiu o Tempo Limite
 - **Cannot Read Property of Undefined** / Não é possível ler a propriedade de algo indefinido
 - **cy.visit failed** / Falha no comando de acesso à aplicação
-
- *Extra - Selenium - Understanding Common Errors*
 - <https://www.selenium.dev/documentation/webdriver/troubleshooting/errors>

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Analise os testes criados para uma aplicação e que estão com falhas diversas, de sintaxe, funcionalidade, seletores. Execute, Identifique a origem do erro e corrija os testes falhando.

Link:

<https://gist.github.com/samlucax/0df852881249b561cdf8888493b03125>

(não vale nota, mas, é importante para fixar)

Organização e Separação de Responsabilidades

Fixtures, Dados hardcoded e Dados externos

Automação de Testes em Aplicações Web

Exercício Individual

Automação de Testes em Aplicações Web



Hands-on

Interação com *upload* de arquivos

Implemente um teste que simula o envio de um formulário de contato (Contact Us) na aplicação que estamos usando em aula, o Automation Exercise.

Você deve mapear os elementos, preencher os campos e adicionar uma interação a mais: o upload de um arquivo antes de enviar o formulário.

O comando para upload será o `.selectFile`.

Funções Auxiliares e Funções Utilitárias

Automação de Testes em Aplicações Web

Exercício Individual

Automação de Testes em Aplicações Web



Hands-on

Usando Dados Dinâmicos

Escolha e instale uma biblioteca de sua preferência para gerar dados dinâmicos durante o teste.

Avalie os testes existentes e ajuste para que use dados dinâmicos nos pontos em que fizer sentido.

Sugestões de biblioteca: *faker-js*, *chancejs*

Módulos e Páginas

Automação de Testes em Aplicações Web

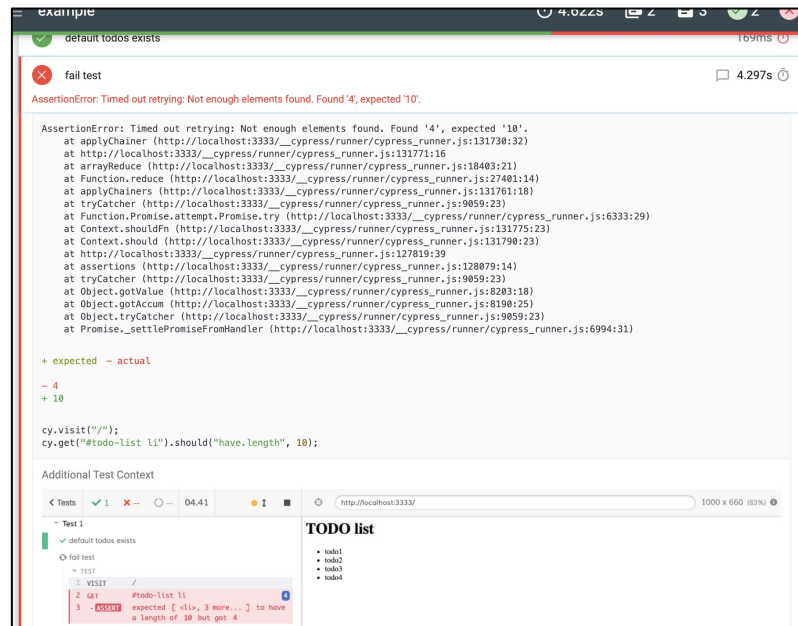
Comandos Customizados (Cypress)

Automação de Testes em Aplicações Web

Relatórios de execução de testes

Automação de Testes Web

Relatórios nos ajudam a passar a visão resumida dos resultados. Podem ser gerados no formato de logs em texto, arquivos json ou xml.



Relatórios de execução de testes

Automação de Testes Web

Hands on

`cypress-mochawesome-reporter`

Resumo

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Instale a biblioteca *cypress-xpath* no projeto e tente adaptar os seletores para usar xpath.

Observe aspectos como legibilidade, velocidade de execução e compartilhe suas percepções na aula.

Vale 1pt. Extra; Enviar apenas a spec via formulário.

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Usando o projeto e conceitos que aprendemos em aula, conclua o processo de modularização das ações feitas para os demais testes do projeto.

(não vale nota, mas, será importante para o final)

Exercício para fazer em casa

Introdução a Aplicações Web



Hands-on

Usando o projeto e conceitos que aprendemos em aula, implemente os cenários **8, 9, 10, 15 e 16** da lista na página *testcases* do site usado em aula.

(não vale nota, mas, será importante para o final)

Perguntas e Respostas

Até a próxima aula

27877678851

Automação de Testes em Aplicações Web

Capacitar os alunos a compreender e aplicar os conceitos fundamentais e avançados relacionados à automação de testes em aplicações web, fornecendo assim uma base sólida de conhecimentos teóricos e práticos para que os alunos possam desenvolver e implementar testes automatizados de alta qualidade, eficiência e confiabilidade em projetos web.

Prof. Samuel Lucas

Ementa e Horários

Automação de Testes em Aplicações Web

Introdução a Aplicações Web
HTML para construção das páginas
CSS para estilização das páginas
Javascript para interação com usuário e servidor
A interação do Frontend com o Backend
Requisições Síncronas e Assíncronas
CSS Selector e XPath
Fundamentos e conceitos sobre Automação de Testes Web
Adicionando bibliotecas para estruturação dos testes e interação com navegadores Web
Escrevendo o primeiro teste automatizado Web
Estratégias de Identificação de Elementos
Esperas Explícitas e Implícitas
Comboboxes, Janelas e Drag and Drop
Tirando Screenshots
Execução de Testes Headless
Design Patterns para Automação Web
Configurações Avançadas
Execução de testes na nuvem
Relatórios de execução de testes
Boas práticas no uso da ferramenta

Datas:

19 e 20 de Setembro
03 e 04 de Outubro
17 e 18 de Outubro

Horários:

Sextas

19h00 - 20h30; 20h45 - 22h00

Sábados

09h00 - 10h30; 10h45 - 12h00;
13h30 - 15h00; 15h15 - 16h30;
16h45 - 18h00

Contexto e Revisão

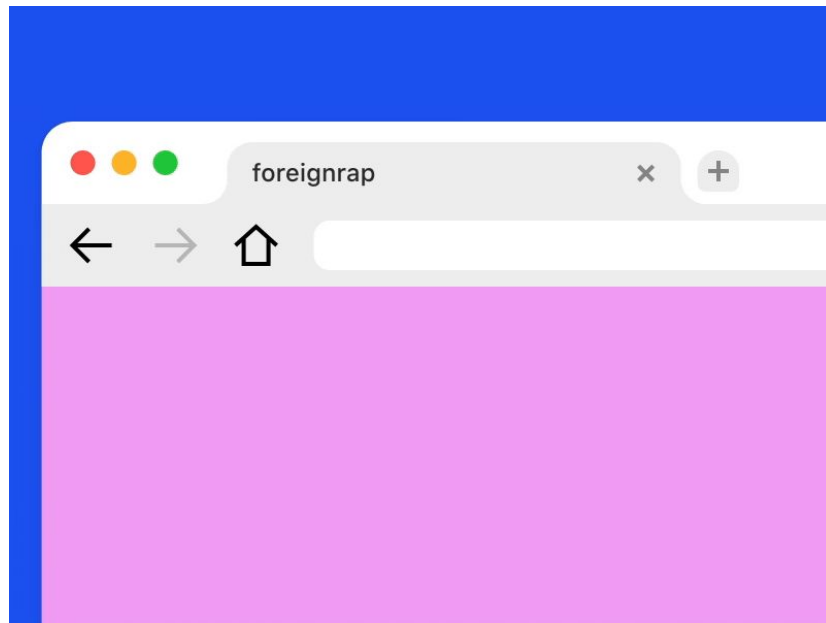
Janelas e Drag and Drop

Janelas e Drag and Drop

Automação de Testes Web

Janelas, também conhecidas como Abas, são uma forma de navegar em múltiplas instâncias dentro de um mesmo navegador.

Aplicações web costumam usar esse recurso para abrir sites externos ou recursos de pagamento e autenticação.



Janelas e Drag and Drop

Automação de Testes Web

Drag and Drop, são interações que permitem ao usuário *arrastar* um elemento de uma área para outra.

Um exemplo clássico: Trello.



Integração com Github Actions e CI/CD

Hands On

IA e Automação de Testes Web

Hands On

Recomendações e Boas Práticas

Recomendações e Boas Práticas

Automação de Testes Web

1. Usar seletores que provavelmente não serão alterados
2. Mover ações que se repetem antes ou depois de cada teste para *Hooks*
3. Definir uma URL base global para execução dos testes
4. Usar de múltiplas asserções sempre que possível
5. Mover códigos que se repetem para funções ou arquivos dedicados

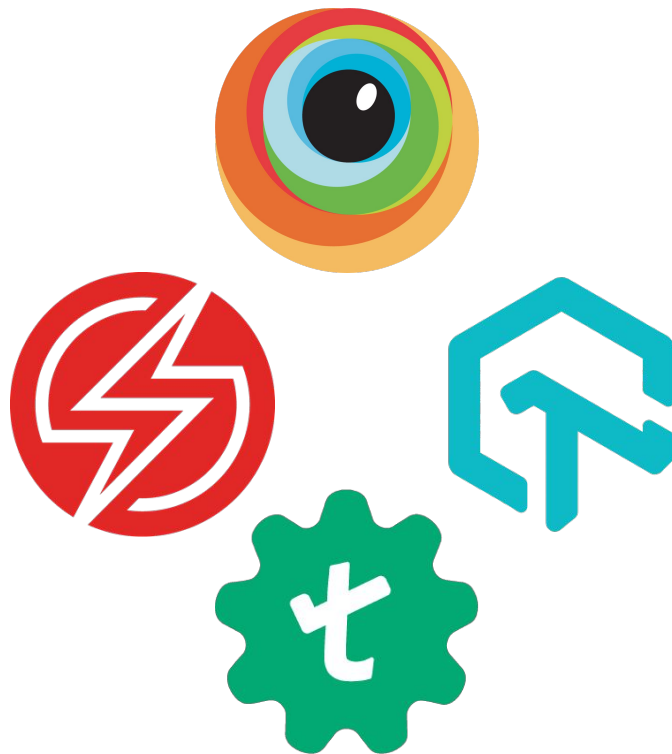
Execução de Testes na Nuvem

Execução de testes na nuvem

Automação de Testes Web

Plataformas de testes na nuvem oferecem uma infraestrutura para execução dos testes.

Em automação de testes, auxiliam em aspectos como **escalabilidade**, **fragmentação** de dispositivos e **infraestrutura** de testes.

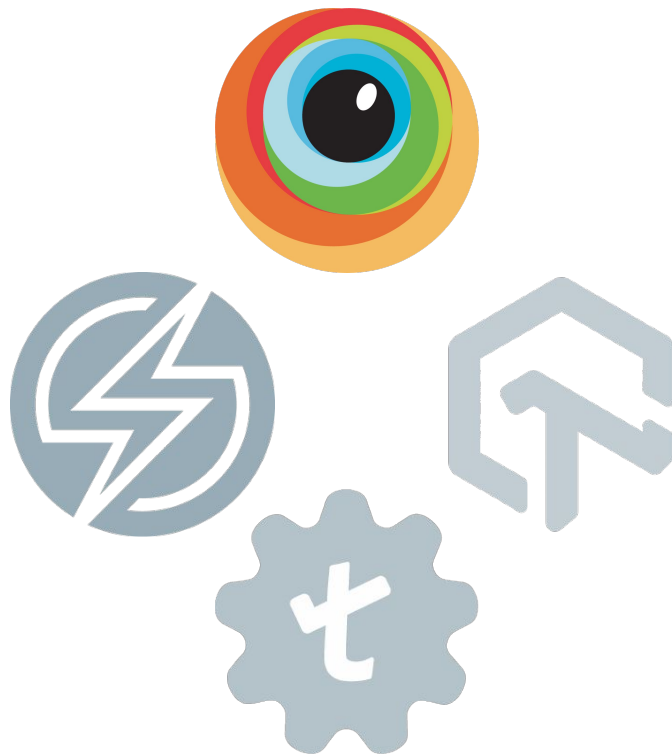


Execução de testes na nuvem

Automação de Testes Web

O Browserstack é a principal plataforma de testes na nuvem atualmente.

Podemos tanto explorar aplicações de forma manual, quanto executar testes automatizados para web e mobile.



Execução de testes na nuvem

Automação de Testes Web

Hands on

Trabalho de Conclusão da Disciplina



Descrição do Trabalho de Conclusão

Trabalho individual.

Usando a aplicação fictícia **Automation Exercise**, crie um projeto que implementa os casos descritos na página "*Test Cases*":

1, 2, 3, 4, 5, 6, 8, 9, 10 e 15



Formato de Entrega do Trabalho

O entregável deverá ser o upload do projeto com a URL para o Github, com atenção para os seguintes pontos:

1. Execução via Github Actions
2. Execução sem quebras
3. Uso adequado de seletores e padrões
4. Atenção a boas práticas
5. Relatórios

Obs.: se possível, remover a node_modules antes de fazer o zip para a entrega



Data e Hora de Entrega

Hoje às 18 horas com suporte ao vivo ou até o dia 31 de Outubro às 23:59:59 sem o suporte ao vivo.

A entrega individual deve ser feita via upload na plataforma.

Encerramento

Trabalho de Recuperação da Disciplina



Descrição do Trabalho de Conclusão

Trabalho individual.

Usando a aplicação fictícia

Automation Exercise, crie um projeto que implementa os casos descritos na página "*Test Cases*":

1, 2, 3, 4, 5, 6, 8, 9, 10 e 15



Formato de Entrega do Trabalho

O entregável deverá ser o upload do projeto com a URL para o Github, com atenção para os seguintes pontos:

1. Execução via Github Actions
2. Execução sem quebras
3. Uso adequado de seletores e padrões
4. Atenção a boas práticas
5. Relatórios

Obs.: se possível, remover a node_modules antes de fazer o zip para a entrega



Data e Hora de Entrega

Até o dia 09 de Novembro às 23:59:59 sem o suporte ao vivo.

A entrega individual deve ser feita via upload na plataforma.

Referências

Lista de Referências Utilizadas no Material

1. Hayes, Linda G. The automated testing handbook. Software Testing Institute, 2004.
2. Axelrod, Arnon. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects, 2018.