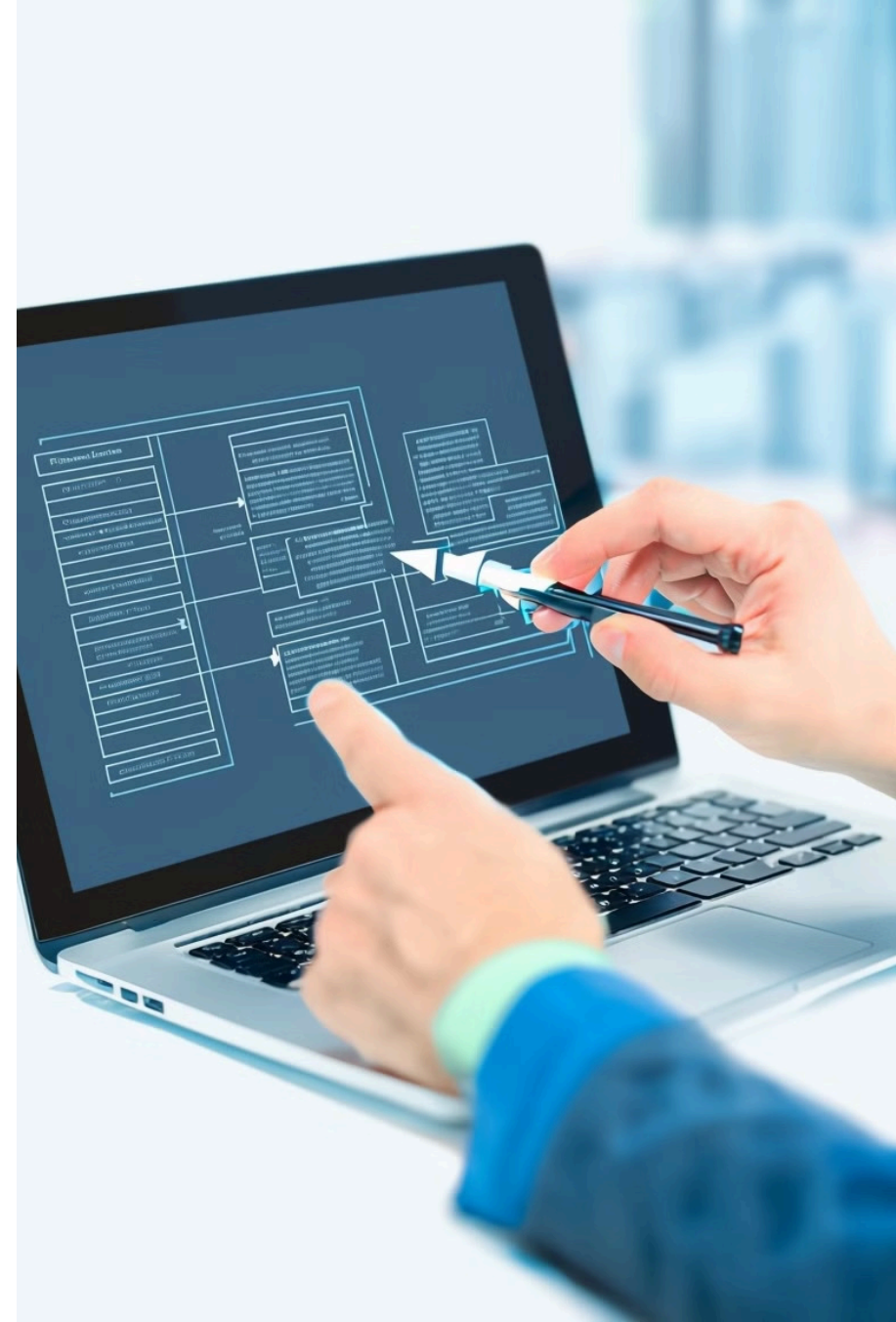


Testes End-to-End

Técnico em Desenvolvimento de Sistemas

Prof. Daniel Ferreira



O que são testes end-to-end?

1

1. Fluxo Completo

Verificam o comportamento de uma aplicação do início ao fim, desde a interface do usuário até o banco de dados.

2

2. Integração de Camadas

Testam a interação entre todas as camadas do sistema, incluindo front-end, back-end e banco de dados.

3

3. Cenários de Uso

Simulam o uso real da aplicação por um usuário, percorrendo todos os passos de um determinado fluxo.

4

4. Experiência do Usuário

Validam a experiência do usuário, garantindo que a aplicação funcione conforme o esperado e de forma intuitiva.



Vantagens dos Testes End-to-End



Confiança Aumentada

Aumentam a confiança na qualidade do software, garantindo que todos os componentes funcionem juntos como esperado.



Detecção de Falhas

Identificam problemas de integração e fluxo de dados, evitando que erros cheguem aos usuários finais.



Melhoria da Experiência do Usuário

Contribuem para uma experiência do usuário mais suave e satisfatória, garantindo que o aplicativo funcione como esperado.

Ferramentas para Testes End-to-End em Node.js



Cypress

Framework moderno e robusto com interface intuitiva, execução em tempo real e debugging fácil. Excelente para aplicações web modernas.



Selenium

Ferramenta consolidada e versátil, com suporte a múltiplos navegadores. Ideal para projetos que necessitam ampla compatibilidade.



Playwright

Desenvolvido pela Microsoft, oferece suporte a múltiplos navegadores e automação confiável. Excelente para testes entre navegadores.



Puppeteer

Ferramenta do Google focada em Chrome/Chromium, perfeita para automação, screenshots e geração de PDFs.



Introdução ao Playwright

O Playwright é um framework de automação de testes open-source que facilita a criação de testes end-to-end para aplicativos web. O Playwright suporta múltiplos navegadores, incluindo Chrome, Firefox, Safari e Edge, e oferece recursos avançados para interagir com elementos de páginas web.

Playwright foi desenvolvido pelo Microsoft, e a comunidade tem crescido rapidamente. É uma ótima opção para quem precisa de uma solução confiável e completa para automação de testes de aplicativos web.

Interagindo com elementos da página

Selecionando Elementos

O Playwright fornece métodos para localizar elementos na página web, como por meio de seletores CSS ou XPath.

Após localizar um elemento, você pode realizar ações como clicar, digitar ou verificar seu conteúdo.

Ações Comuns

Exemplos de ações comuns incluem clicar em botões, preencher campos de texto, selecionar itens em menus e verificar caixas de seleção.

O Playwright oferece uma variedade de métodos para interagir com diferentes tipos de elementos da página.

Manipulando dados de entrada



Campos de texto

Simule a digitação de texto em campos de entrada, como caixas de texto, áreas de texto e campos de pesquisa.



Seletores de data

Defina datas específicas em seletores de data, incluindo datas futuras e passadas.



Caixas de seleção

Selecione opções em listas de opções, como caixas de seleção, botões de rádio e menus suspensos.



Upload de arquivos

Envie arquivos para formulários web, simulando a experiência do usuário.



Verificando resultados

Asserções

Asserções garantem que o resultado esperado seja atingido. Bibliotecas de teste fornecem métodos para validar o conteúdo da página, elementos da interface, ou até mesmo comportamento específico.

Comparação de valores

Com asserções, podemos comparar valores de texto, atributos, estados de elementos e outros dados relevantes do aplicativo. Essa comparação garante que o aplicativo esteja funcionando conforme o esperado.

Validação de comportamento

Asserções também podem ser usadas para validar o comportamento dinâmico do aplicativo, como a resposta do servidor a uma solicitação, o redirecionamento para outras páginas ou a exibição de mensagens de erro.

Organização e estruturação dos testes

Arquivos de Teste

Organize os testes em pastas separadas para cada funcionalidade ou componente do aplicativo.

Nomeie os arquivos de teste de forma clara e descritiva, refletindo o objetivo do teste.

Suítes de Teste

Agrupe testes relacionados em suítes para facilitar a execução e organização.

Crie suítes para diferentes tipos de testes, como smoke tests, testes de integração e testes de regressão.

Relatórios e Screenshots

Os relatórios de testes end-to-end fornecem informações detalhadas sobre o andamento e os resultados de cada teste.

Imagens capturadas durante a execução dos testes podem ajudar a identificar erros visuais, como layouts quebrados ou elementos ausentes.

Playwright oferece recursos para gerar relatórios abrangentes e capturar screenshots automaticamente.





Boas práticas e dicas para testes end-to-end

Foco em cenários de usuário

Testes devem simular o comportamento real dos usuários, incluindo ações e interações comuns.

Testes específicos e concisos

Cada teste deve verificar um único fluxo de trabalho, facilitando a identificação e depuração de falhas.

Manutenção regular

Testes devem ser atualizados e mantidos para refletir mudanças no código e interface do usuário.

Uso de ferramentas adequadas

Selecionar ferramentas de teste que se integrem ao ambiente de desenvolvimento e atendam às necessidades do projeto.

Exemplos do Playwright no VS Code

Instalação e configuração do Playwright

1

Fork e clone

```
https://github.com/daniellferreira/class-e2e-test
```

2

Acessar repositório clonado e iniciar projeto npm

```
cd class-e2e-test  
npm init -y
```

3

Instalação do Playwright

==> VS Code

O Playwright pode ser instalado globalmente ou localmente no seu projeto.

A instalação inclui os navegadores que você deseja usar.

Exercício: Criando testes end-to-end para um aplicativo web

1

Escolha um aplicativo web

Selecione um aplicativo web que você conhece ou que esteja interessado em testar.

Exemplo:

the-internet.herokuapp.com

2

Defina cenários de teste

Identifique os principais fluxos de usuários no aplicativo e crie cenários de teste que cubram esses fluxos.

3

Instale o Playwright

Instale o Playwright e configure o ambiente de desenvolvimento.

4

Escreva os testes

Crie testes end-to-end para os cenários definidos, utilizando o Playwright.

5

Execute os testes

Execute os testes e verifique se eles estão funcionando conforme o esperado.

6

Analise os resultados

Analise os resultados dos testes e corrija quaisquer erros encontrados.