

Tutorial de Cypress:

passo a passo para iniciantes

Sumário

1. Sobre a ferramenta
2. Tutorial de uso
3. Links de referência
4. Contatos do grupo
5. Objetivo do tutorial

1. Sobre a ferramenta

Cypress é um framework de testes automatizados end-to-end, que usa a linguagem JavaScript e roda em vários navegadores, como Chrome, Firefox, Edge, entre outros.

Com o Cypress, é possível automatizar os testes de qualquer aplicação que possa ser acessada via navegador, seja ela uma API REST, um site, um aplicativo web ou um sistema.

Algumas das vantagens de usar o Cypress para automatizar os testes dos produtos digitais são:

- Instalação e configuração rápida e fácil, sem a necessidade de dependências externas ou configurações complexas;
- Atuação no mesmo loop de execução do navegador, permitindo acesso a comandos nativos e interação com o tráfego da rede, podendo ler e manipular os dados recebidos e enviados;
- Debug dos testes em tempo real, com a possibilidade de selecionar qualquer estado da execução do teste, retornar para a página que a aplicação estava localizada e interagir com os elementos naquele instante;
- Captura de elementos com um simples clique, retornando ao usuário um seletor para ser utilizado na automação;
- Geração de relatórios detalhados, com screenshots e vídeos dos testes, facilitando a análise e a correção dos erros.

2. Tutorial de uso

- **Processo de instalação**

Para criar o seu primeiro projeto de testes automatizados com Cypress, é necessário que você tenha um editor de código em seu computador (recomendamos o [Visual Studio Code](#)) e que abra o mesmo para então realizar as configurações necessárias.

Antes de iniciar a instalação em seu projeto, é necessário confirmar que você possui o *Node.js* instalado. Para isto, basta checar em um terminal com o seguinte comando:

```
# npm -v
```

Se você já possuir o *Node.js* instalado, irá aparecer no terminal o número da versão que você possui, do contrário, o terminal acusará com uma mensagem do tipo “Comando não encontrado”. Então, você terá que baixar e instalar o [Node.js](#) com a versão mais atual e instável disponível, basta clicar no link referenciado e seguir os passos de instalação da ferramenta.

Antes de partir para a instalação do Cypress, criaremos um diretório para estruturar os testes. Então é necessário você abrir o editor de código e na opção terminal, localizada no menu da ferramenta, clicar em Novo terminal. Se preferir, você pode executar os comandos posteriores no terminal do próprio computador.

No terminal, execute o comando:

```
# mkdir seu-projeto-cypress
```

Com este comando você irá criar uma pasta onde ficarão todos os arquivos e informações do seu projeto, depois é só entrar na pasta criada com o comando:

```
# cd seu-projeto-cypress
```

Com o *Node.js* e um editor de código devidamente instalados e com a pasta do projeto de testes criada, então é possível realizar a instalação do Cypress. Ainda no terminal, considerando que já esteja na pasta do projeto, execute os comandos:

```
# npm init -y
```

```
# npm install cypress --save-dev
```

O primeiro, inicializa um novo projeto *Node.js* com um arquivo *package.json* usando as configurações padrão. O segundo é usado para instalar o Cypress como uma dependência de desenvolvimento em um projeto *Node.js*.

- **Configuração do ambiente**

Após a instalação, a configuração do ambiente é o próximo passo. Você pode inicializar o Cypress, pelo terminal, com o comando:

```
# npx cypress open
```

Este comando abre a interface gráfica do Cypress. Você verá duas opções principais de testes:

- E2E Testing: Para testes End-to-End.
- Component Testing: Para testes de componentes individuais.

Clique em E2E Testing para iniciar a configuração e, depois, selecione o navegador que desejar:

Welcome to Cypress!

[Review the differences between each testing type →](#)



E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

● Configured



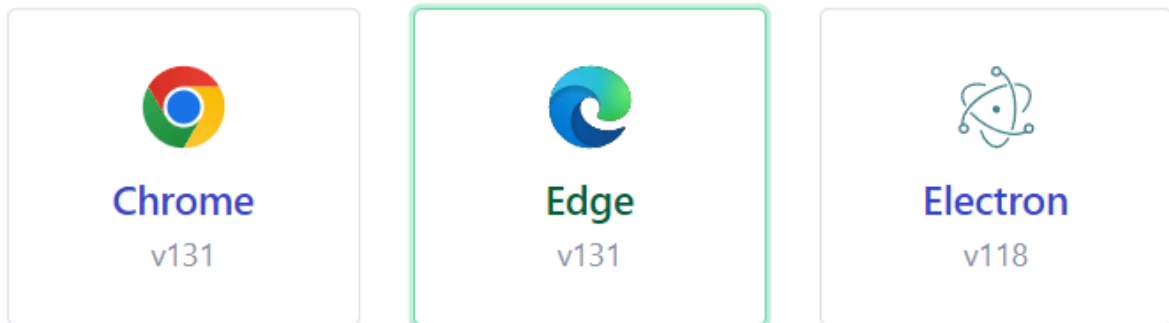
Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

● Not Configured

Choose a browser

Choose your preferred browser for E2E testing.



Por padrão, na primeira execução, são gerados alguns arquivos e pastas com a seguinte estrutura:

- `cypress/` : Diretório principal para os testes.
 - `e2e/` : Contém os testes End-to-End (E2E).
 - `support/` : Contém arquivos para personalizar os comportamentos globais e comandos customizados.
 - `fixtures/` : Contém arquivos de dados usados nos testes.
- `cypress.config.js` : Arquivo de configuração do Cypress.

• Implementando o primeiro teste

Após os passos acima, clique em “Create new spec”, para criar seu primeiro arquivo de teste. Coloque o nome de sua preferência, por exemplo: `primeiro_teste.cy.js`. O Cypress executará um teste de exemplo.

Create your first spec

Since this project looks new, we recommend that you use the specs and tests that we've written for you to get started.



Scaffold example specs


We'll generate several example specs to help guide you on how to write tests in Cypress.



Create new spec

We'll generate a template spec file which can be used to start testing your application.

If you feel that you're seeing this screen in error, and there should be specs listed here, you likely need to update the spec pattern.

 [View spec pattern](#)

primeiro_teste.cy.js

Abra o referido arquivo no editor de código, insira os comandos abaixo, salve e em seguida volte ao navegador em que o Cypress está aberto.

```
describe('Testes simples com Cypress', () => {

  // Cenário 1

  it('Verifica título da página', () => {

    cy.visit('https://www.google.com');

    cy.title().should('include', 'Google');

  });

  // Cenário 2

  it('Verifica título da página com falha', () => {
```

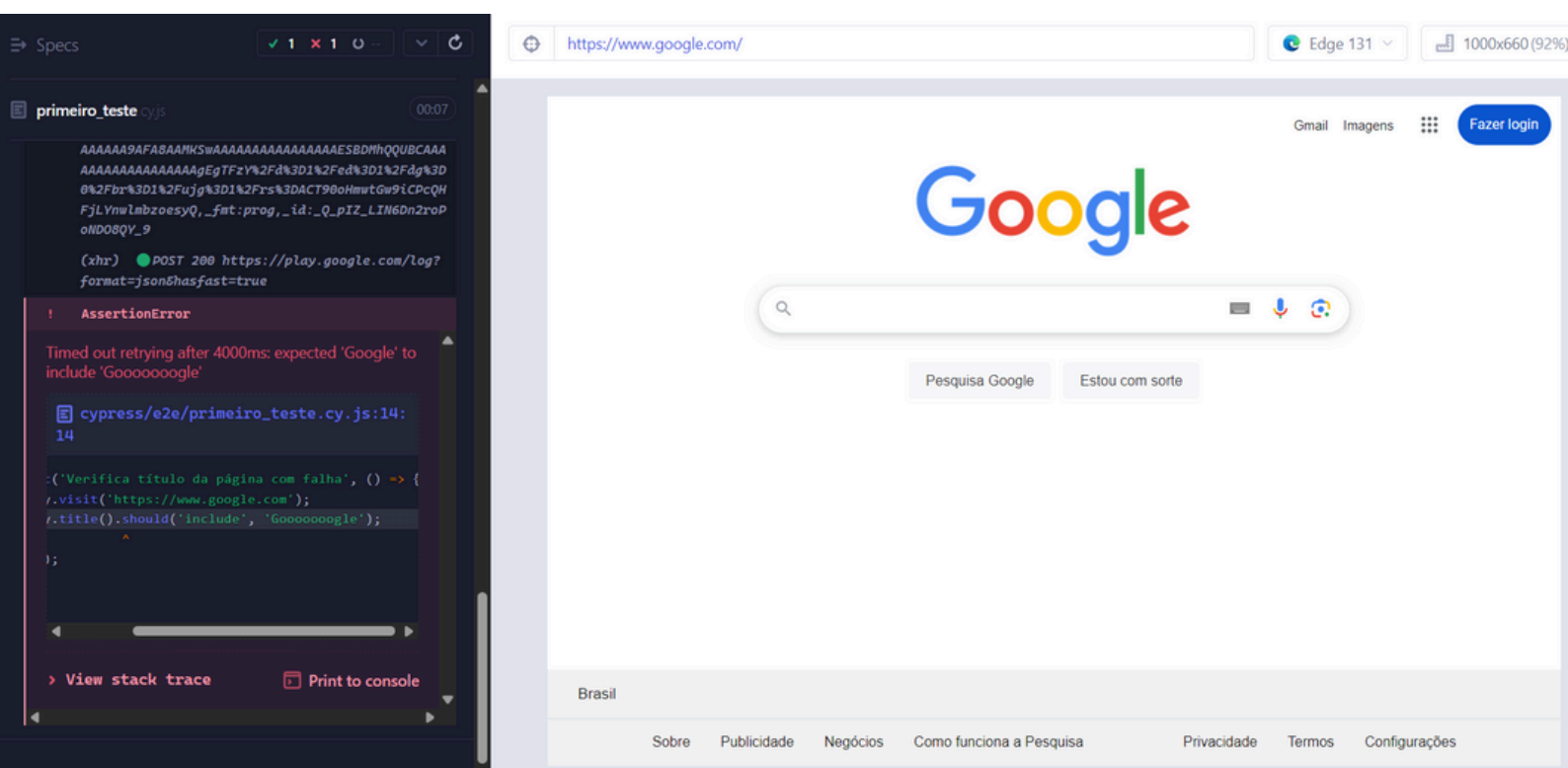
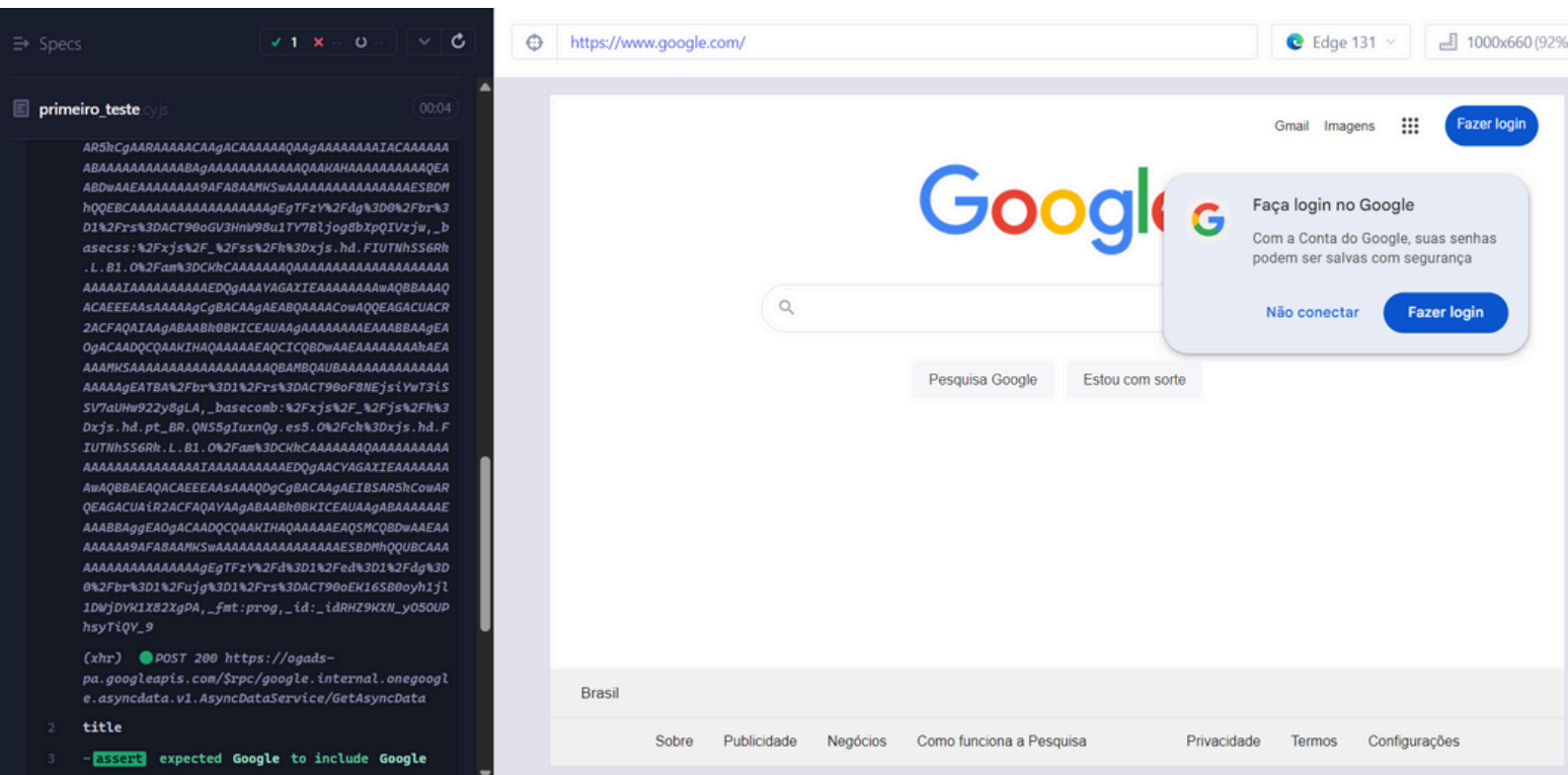
```
cy.visit('https://www.google.com');

cy.title().should('include', 'Gooooooooogle');

});

});
```

Você verá que os testes acima foram executados e, após finalizar a execução, terá no log do Cypress o resultado do teste (assert), algo como demonstrados nas imagens abaixo:



No primeiro cenário, o resultado obtido foi de sucesso porque o título da página é Google, enquanto que no segundo resultado tivemos uma falha devido a não encontrar o título da página conforme passado no código.

3. Links de referência

Automação de Testes com Cypress. Zup, 2024. Disponível em: <https://zup.com.br/blog/automacao-de-testes-cypress/>. Acesso em: 22 e 27 nov. 2024.

Testes End-to-End com Cypress: Guia Completo. Escola DNC, 2024. Disponível em: <https://www.escoladnc.com.br/blog/testes-end-to-end-com-cypress-guia-completo/>. Acesso em: 22 e 27 nov. 2024.

Iniciando com Testes E2E no Cypress. Rocketseat, 2024. Disponível em: <https://blog.rocketseat.com.br/iniciando-com-testes-e2e-no-cypress/>. Acesso em: 22 e 27 nov. 2024.

Cypress: Passo a Passo para Começar a Usar. SoftDesign, 2024. Disponível em: https://softdesign.com.br/blog/cypress_passo_a_passo_para_comecar_a_usar/. Acesso em: 22 e 27 nov. 2024.

Por que Cypress? Blog Cypress, 2024. Disponível em: <https://docs.cypress.io/app/get-started/why-cypress/>. Acesso em: 28 nov 2024.

4. Contatos do grupo

GRUPO 8



Júlia Orsi

<https://www.linkedin.com/in/júlia-orsi>



Lucas Santana

<https://www.linkedin.com/in/lucasantanas>



Ranielle Sousa

<https://www.linkedin.com/in/ranielle-sousa-0345b733>



Rayra Bulcão

<https://www.linkedin.com/in/rayrabulcao>



Ruan Osório

<https://www.linkedin.com/in/ruanosorio>

5. Objetivo do tutorial

Este tutorial tem como objetivo apresentar uma introdução prática ao uso do **Cypress**, destacando desde sua instalação e configuração até a implementação de testes simples e avançados. Por meio de exemplos claros e estruturados, busca-se capacitar desenvolvedores e QAs a realizar automação de testes end-to-end, garantindo maior qualidade em aplicações web de forma eficiente e acessível.