



# E-book

## 10 Comandos Cypress

<https://cypress.qazando.com.br/>

## Introdução

O Cypress é uma ferramenta de teste de ponta a ponta (end-to-end) que oferece uma interface de linha de comando (CLI) fácil de usar e uma biblioteca JavaScript poderosa. Ele é capaz de testar tudo o que um usuário pode fazer em um aplicativo da web, desde clicar em botões até preencher formulários e interagir com elementos da interface. Além disso, o Cypress é rápido, confiável e fácil de aprender.

Neste E-book, vamos dar uma olhada nos 10 comandos mais utilizados do Cypress. Esses comandos são a base do teste de ponta a ponta com o Cypress e cobrem grande parte das necessidades dos testadores. Vamos lá!

## 1. cy.visit()

O comando `cy.visit()` é usado para abrir uma página da web. Ele recebe como argumento uma URL e navega até ela. Aqui está um exemplo:

arduinoCopy code

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

Esse comando é útil para iniciar um teste, pois ele carrega a página que você deseja testar.

## 2. cy.get()

O comando `cy.get()` é usado para selecionar um elemento na página. Ele recebe como argumento um seletor CSS, um seletor jQuery ou um objeto que representa um elemento DOM. Aqui está um exemplo:

csharpCopy code

```
cy.get('#nome-de-usuario')
```

Esse comando é útil para interagir com elementos na página, como clicar em botões, preencher formulários ou verificar se um elemento existe.

## 3. cy.contains()

O comando `cy.contains()` é usado para localizar um elemento com base no seu conteúdo de texto. Ele recebe como argumento um texto e seleciona o primeiro elemento na página que contém esse texto. Aqui está um exemplo:

arduinoCopy code

```
cy.contains('Entrar')
```

Esse comando é útil para encontrar botões ou links com base no seu texto.

## 4. `cy.click()`

O comando `cy.click()` é usado para clicar em um elemento na página. Ele recebe como argumento um seletor CSS, um seletor jQuery ou um objeto que representa um elemento DOM. Aqui está um exemplo:

csharpCopy code

```
cy.get('#botao-de-acao').click()
```

Esse comando é útil para interagir com botões ou links na página.

## 5. `cy.type()`

O comando `cy.type()` é usado para inserir texto em um campo de entrada na página. Ele recebe como argumento uma string com o texto que deve ser digitado. Aqui está um exemplo:

bashCopy code

```
cy.get('#campo-de-texto').type('Olá, mundo!')
```

Esse comando é útil para preencher formulários ou campos de entrada.

## 6. `cy.wait()`

O comando `cy.wait()` é usado para pausar a execução do teste por um determinado período de tempo. Ele recebe como argumento um número que representa o tempo em milissegundos. Aqui está um exemplo:

scssCopy code

```
cy.wait(1000)
```

Esse comando é útil para aguardar a conclusão de uma ação assíncrona, como o carregamento de uma página ou a exibição de uma animação.

## 7. `cy.url()`

O comando `cy.url()` é usado para obter a URL atual da página em que o teste está sendo executado. Ele não recebe argumentos e retorna a URL como uma string. Aqui está um exemplo:

rustCopy code

```
cy.url().should('include', 'https://www.exemplo.com')
```

Esse comando é útil para verificar se o teste está na página correta ou se a navegação para outra página foi bem-sucedida.

## 8. `cy.getCookie()`

O comando `cy.getCookie()` é usado para obter um cookie da página. Ele recebe como argumento o nome do cookie e retorna um objeto que representa o cookie. Aqui está um exemplo:

arduinoCopy code

```
cy.getCookie('sessao').should('have.property', 'value', 'abc123')
```

Esse comando é útil para verificar se um cookie foi definido corretamente ou para usar informações de cookies em testes subsequentes.

## 9. `cy.intercept()`

O comando `cy.intercept()` é usado para interceptar solicitações de rede na página. Ele recebe como argumento uma URL ou um padrão de URL e permite manipular ou bloquear solicitações. Aqui está um exemplo:

csharpCopy code

```
cy.intercept('GET', '/api/dados').as('dados')
```

```
cy.wait('@dados')
```

Esse comando é útil para testar a interação da página com serviços ou APIs externas.

## 10. `cy.expect()`

O comando `cy.expect()` é usado para fazer afirmações no teste. Ele recebe como argumento um valor ou uma expressão e um ou mais métodos de assunto que especificam as condições que devem ser atendidas. Aqui está um exemplo:

csharpCopy code

```
cy.get('#nome-de-usuario').should('have.value', 'fulano')
```

Esse comando é útil para verificar se o comportamento da página é o esperado, como a presença ou ausência de um elemento ou o valor de um campo de entrada.

## Conclusão

Esses foram os 10 comandos mais utilizados do Cypress. Com esses comandos, é possível realizar testes de ponta a ponta em uma página da web de forma rápida, confiável e fácil. Há muitos outros comandos disponíveis no Cypress, mas esses são um ótimo ponto de partida para começar a testar com essa ferramenta poderosa. Espero que este E-book tenha sido útil!