

Curso de Cypress Básico

Abaixo o código de uma página de login feita pelo professor com acesso local

<http://127.0.0.1:5500/CursoAutomacao/login.html>.

```
/// <reference types="cypress"/>

describe('Testar a página de login', () => {
  it('Tela de login', () => {
    cy.visit('/CursoAutomacao/login.html')
    cy.get('span[class="login-form-tittle"]')
      .contains('Faça o Login').should('contain', 'Faça o Login')

    cy.get('input[class="input-form"]').eq(0) //eq pega o primeiro
    elemento
      .should('be.visible') // mostra que o elemento está visível

    cy.get('input[class="input-form"]').eq(0).click().type('Automação
de testes')
      .should('have.value', 'Automação de testes') // verifica que
possui o valor Automação de testes

    cy.get('input[class="input-form"]').eq(1) // eq pega o segundo
    elemento
      .should('be.visible') // mostra se está visível

    cy.get('input[class="input-form"]').eq(1).click().type('123456')
    // clica e digita a senha
      .should('have.value', '123456') // verifica se possui o valor
senha 123456

    cy.get('img[src="./images/login.png"]')
      .should('be.visible') // verifica se existe uma imagem na
tela

    cy.get('img[class="margin-left-50"]')
      .should('be.visible') // verifica a existencia da classe da
imagem

    cy.get('button[class="login-form-btn"]').contains('Login') //
valida se contém a palavra Login no botão
      .should('be.visible') // mostra se o botão está visível

    cy.get('button[class="login-form-btn"]').click() // clica no
botão Login
  });
});
```

```
});
```

O nome do arquivo é `Telalogin.spec.js` estou utilizando o Cypress 9.7.0 pois as versões 12 e 13 estavam dando problemas na execução.

Fiz umas configurações no arquivo `cypress.json`:

```
{
  "baseUrl": "http://127.0.0.1:5500/CursoAutomacao/login.html",
  "chromeWebSecurity": false,
  "viewportWidth": 1400,
  "viewportHeight": 900,
  "defaultCommandTimeout": 10000,
  "video": true,
  "screenshots": true,
  "pageLoadTimeout": 50000,
  "responseTimeout": 50000,
  "numTestsKeptInMemory": 50
}
```

`baseUrl` para tornar a URL global nos testes;

`chromeWebSecurity` para eliminar problemas de segurança no google chrome;

`viewportWidth` e `viewportHeight` se referem ao tamanho da tela;

`defaultCommandTimeout` tempo padrão para acessar um elemento na página;

`video` quando executamos os testes em modo texto e ocorre um erro, é gerado um vídeo;

`screenshots` tira print da tela;

`pageLoadTimeout` define um tempo para a página ser executada;

`responseTimeout` tempo de execução de uma API;

`numTestsKeptInMemory` tamanho de registro de memória;

Próximo passo é criar um arquivo de nome `Telaindex.spec.js` e realizar os testes abaixo:

```
/// <reference types="cypress"/>

describe('Validar botões na tela', () => {
  beforeEach(function(){
    cy.visit('/CursoAutomacao/Index.html')
  })
  it('Validar Título na página', () => {
    cy.get('h1')
```

```

        .contains('Curso Automação de Testes
Básico')).should('have.text', 'Curso Automação de Testes Básico')
    });

    it('Validando os três botões', () => {
        cy.get('a[href="/login.html"]').as('btnVoltar')
            .should('be.visible')
        cy.log('Botão voltar visível')
        // cy.pause() comando para pausar o teste até aqui

        cy.get('a[href="texto.html"]').as('BtnTexto')
            .should('be.visible')
        cy.log('Botão texto está visível')

        cy.get('a[href="imagem.html"]').as('BtnImagem')
            .should('be.visible')
        cy.log('Botão imagem está visível')

        cy.get('input[type="submit"]').as('BtnCadastrar')
            .should('be.visible').should('have.value', 'Cadastrar')
        cy.log('O botão cadastrar está visível')
    });

    it('Validar os campos', () => {
        cy.get('input[id="txtNome"]').should('be.visible')

        cy.get('input[id="numero"]').should('be.visible')

        cy.get('input[id="dtpDataNascimento"]').should('be.visible')
    });

    it('Validar checkbox', () => {
        cy.get('label[for="rdoMasculino"]').should('be.visible')

        cy.get('label[for="rdoFeminino"]').should('be.visible')
    });

    it('Validar o texto Tabela e a tabela', () => {
        cy.get('h2')
            .contains('Tabela').should('have.text', 'Tabela') //tesxto na
tela

        cy.get('tr').find('th').eq(0) // todos os 6 elementos (coluna) da
tabela

```

```

        .should('have.text', 'Id')

    cy.get('tr').find('th').eq(1)
        .should('have.text', 'Nome')

    cy.get('tr').find('th').eq(2)
        .should('have.text', 'Telefone')

    cy.get('tr').find('th').eq(3)
        .should('have.text', 'Nascimento')

    cy.get('tr').find('th').eq(4)
        .should('have.text', 'Sexo')

    cy.get('tr').find('th').eq(5)
        .should('have.text', 'Última Atualização')
    });
});

```

Agora falta a parte 2 para concluirmos os testes. Ainda na página Index precisamos fazer o cadastro de usuário, excluir e Alterar, segue abaixo o código comentado:

```

/// <reference types="cypress"/>

describe('Testando a tela inicial do site', () => {
    beforeEach(function(){
        cy.visit('/CursoAutomacao/Index.html')
    })

    context('Validação de elementos na página inicial', () => {
        it('Validar Título na página', () => {
            cy.get('h1')
                .contains('Curso Automação de Testes
Básico').should('have.text', 'Curso Automação de Testes Básico')
        });

        it('Validando os três botões', () => {
            cy.get('a[href="./login.html"]').as('btnVoltar')
                .should('be.visible')
            cy.log('Botão voltar visível')
            // cy.pause() comando para pausar o teste até aqui

            cy.get('a[href="texto.html"]').as('BtnTexto')
                .should('be.visible')
            cy.log('Botão texto está visível')

            cy.get('a[href="imagem.html"]').as('BtnImagem')
                .should('be.visible')

```

```

        cy.log('Botão imagem está visível')

        cy.get('input[type="submit"]').as('BtnCadastrar')
            .should('be.visible').should('have.value', 'Cadastrar')
        cy.log('O botão cadastrar está visível')
    });

    it('Validar os campos', () => {
        cy.get('input[id="txtNome"]')
            .should('be.visible')

        cy.get('input[id="numero"]')
            .should('be.visible')

        cy.get('input[id="dtpDataNascimento"]')
            .should('be.visible')
    });

    it('Validar checkbox', () => {
        cy.get('label[for="rdoMasculino"]')
            .should('be.visible')

        cy.get('label[for="rdoFeminino"]')
            .should('be.visible')
    });

    it('Validar o texto Tabela e a tabela', () => {
        cy.get('h2')
            .contains('Tabela').should('have.text', 'Tabela')
        //tesxto na tela

        cy.get('tr').find('th').eq(0) // todos os 6 elementos
        (coluna) da tabela
            .should('have.text', 'Id')

        cy.get('tr').find('th').eq(1)
            .should('have.text', 'Nome')

        cy.get('tr').find('th').eq(2)
            .should('have.text', 'Telefone')

        cy.get('tr').find('th').eq(3)
            .should('have.text', 'Nascimento')

        cy.get('tr').find('th').eq(4)
            .should('have.text', 'Sexo')

        cy.get('tr').find('th').eq(5)
            .should('have.text', 'Ultima Atualização')
    });

```

```

    });

});

context('Cadastrando usuário', () => {
  it('Cadastrando um novo usuário', () => {
    cy.get('input[id="txtNome"]')
      .type('Simone dos Santos Silva')

    cy.get('input[id="numero"]')
      .type('11973571191')

    cy.get('input[id="dtpDataNascimento"]')
      .type('1981-08-15')

    cy.get('label[for="rdoFeminino"]')
      .click()

    cy.get('input[id="btnCadastrarSalvar"]')
      .click()

    cy.get('button').contains('Alterar').should('be.visible')//
    botão Alterar
    //cy.get('button').contains('Excluir').should('be.visible').c
    lick() // uma maneira diferente de capturar o botão

    //Cadastrar outro usuário para efetuar a alteração

    cy.get('input[id="txtNome"]')
      .type('Sofia Santos de Oliveira')

    cy.get('input[id="numero"]')
      .type('11973571191')

    cy.get('input[id="dtpDataNascimento"]')
      .type('2010-10-22')

    cy.get('label[for="rdoFeminino"]')
      .click()

    cy.get('input[id="btnCadastrarSalvar"]')
      .click()

    cy.get('button').contains('Alterar').click()

    cy.get('input[id="numero"]').clear()

    cy.get('input[id="btnCadastrarSalvar"]')
      .click()
  })
});

```

```

    });
  });

  context.only('Validar dados da tabela', () => {
    it('Fazer cadastro e depois validar os Dados', () => {

      cy.get('input[id="txtNome"]')
        .type('Sofia Simone de Oliveira')

      cy.get('input[id="numero"]')
        .type('11973571191')

      cy.get('input[id="dtpDataNascimento"]')
        .type('1966-09-23')

      cy.get('label[for="rdoFeminino"]')
        .click()

      cy.get('input[id="btnCadastrarSalvar"]')
        .click()

      cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').eq(0).invoke('text').then((PegaId) =>{
        cy.wrap(PegaId).as('Id da tabela') // função que pega o
        Id na tabela, cy.wrap pra pegar a variável, as cria um alias
        cy.get('tbody[id="tbodyResultados"]').find("tr").find('td')
        '.eq(0)
          .should('have.text', PegaId) // validação da variável
        PegaId que contem o Id da tabela
      })

      cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').eq(1)
        .should('have.text', 'Sofia Simone de Oliveira')

      cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').eq(2)
        .should('have.text', '11973571191')

      cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').eq(3)
        .should('have.text', '23/09/1966')

      cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').eq(4)
        .should('have.text', 'F')

    });
  });

```

```
});  
  
});
```

Agora falta customizar isso tudo, deixar mais enxuto o código. Antes disso, ainda mexendo na página Index vamos fazer testes clicando nos botões que estão no canto superior esquerdo (interagindo com eles).

Observação: No vídeo o professor coloca os dados verdadeiros da tela de Login, para acessar o site, que é usuário: admin e senha: admin (corrigi isso no arquivo TelaLogin.spec.js).

Crie um arquivo de nome Telainicial.spec.js para começarmos a fazer os testes, segue abaixo o código:

```
/// <reference types="cypress"/>  
  
describe('Testes na tela index', () => {  
  beforeEach(() => {  
    cy.visit('/CursoAutomacao/Index.html')  
  });  
  
  it('Textar botão de texto', () => {  
    cy.get('button')// procura o botão  
      .contains('Texto').should('be.visible').click() // verifica  
se o botão contém a palavra Texto  
  
    //Verificar texto da página  
    cy.get('div[class="all"]').find('h1[class="title"]')// acha a  
classe e depois procura pela classe do texto h1  
      .should('have.text', 'Curso Automação de Testes Básico')  
  
    cy.get('h2')// texto  
      .contains('Automação de testes') // texto  
  
    cy.get('div[class="all"]').find('p') // acha a classe e procura  
pelo elemento paragrafo p  
      .should('contain', 'A automação de testes é')  
  
    cy.get('button')// encontra o botão  
      .contains('Voltar Ao início').click() // afirma que contem o  
texto Voltar ao início e clica no botão  
  });  
  
});
```

Todo esse código, faz o teste de clicar no botão 'Texto' vai para a página que contém um texto sobre automação, valida o título da página h1, o subtítulo h2 e depois o texto, em seguida clica no botão 'Voltar Ao início'.

Próximo passo é instalar o faker (para usarmos dados aleatórios, fictícios), no site <https://fakerjs.dev/guide/> tem o passo a passo pra instalação:

```
npm install @faker-js/faker --save-dev -
```

Basta digitar o comando acima no terminal para fazer a instalação.

Criei uma cópia do arquivo Telaindex.spec.js ficou Telaindex.spec copy.js e coloquei o .only no contexto de cadastro de usuário assim:

```
context.only('Cadastrando usuário', () => {
```

Na parte superior do código temos que adicionar a linha de comando abaixo:

```
import {faker} from '@faker-js/faker';
```

E agora para cada campo é necessário adicionar um código com uma variável correspondente para o faker funcionar, por exemplo: para o faker gerar um nome feminino aleatório adiciona-se a linha abaixo:

```
/// <reference types="cypress"/>
import {faker} from '@faker-js/faker';
const randomNomeFeminino = faker.person.firstName('female')
```

Note que criei uma variável de nome 'randomNomeFeminino, precisamos adicionar a variável no teste, ficará assim:

```
context.only('Cadastrando usuário', () => {
  it('Cadastrando um novo usuário', () => {
    cy.get('input[id="txtNome"]')
      .type(randomNomeFeminino)
```

Ao executar será gerado um nome feminino aleatório.

Abaixo, segue todo o código com o uso do faker:

```
/// <reference types="cypress"/>
import {faker} from '@faker-js/faker';
const randomNomeFeminino = faker.person.firstName('female')
const numeroRandom = faker.phone.number('#####') // 'ESSE
ASTERISTICO SIGNIFICA A QUANTIDADE DE NUMEROS QUE VOCE VAI QUERE GERAR E
O FORMATO TBM'
const date = faker.date.future(); // Isso gera uma data
const formattedDate = `${date.getFullYear()}-${(date.getMonth() +
1).toString().padStart(2, '0')}-${date.getDate().toString().padStart(2,
'0')}`;
// o código acima tranforma a data em uma string
describe('Testando a tela inicial do site', () => {
  beforeEach(function () {
    cy.visit('/CursoAutomacao/Index.html')
  })

  context('Validação de elementos na página inicial', () => {
    it('Validar Título na página', () => {
```

```

        cy.get('h1')
            .contains('Curso Automação de Testes
Básico').should('have.text', 'Curso Automação de Testes Básico')
    });

    it('Validando os três botões', () => {
        cy.get('a[href="/login.html"]').as('btnVoltar')
            .should('be.visible')
        cy.log('Botão voltar visível')
        // cy.pause() comando para pausar o teste até aqui

        cy.get('a[href="texto.html"]').as('BtnTexto')
            .should('be.visible')
        cy.log('Botão texto está visível')

        cy.get('a[href="imagem.html"]').as('BtnImagem')
            .should('be.visible')
        cy.log('Botão imagem está visível')

        cy.get('input[type="submit"]').as('BtnCadastrar')
            .should('be.visible').should('have.value', 'Cadastrar')
        cy.log('O botão cadastrar está visível')
    });

    it('Validar os campos', () => {
        cy.get('input[id="txtNome"]')
            .should('be.visible')

        cy.get('input[id="numero"]')
            .should('be.visible')

        cy.get('input[id="dtpDataNascimento"]')
            .should('be.visible')
    });

    it('Validar checkbox', () => {
        cy.get('label[for="rdoMasculino"]')
            .should('be.visible')

        cy.get('label[for="rdoFeminino"]')
            .should('be.visible')
    });

    it('Validar o texto Tabela e a tabela', () => {
        cy.get('h2')
            .contains('Tabela').should('have.text', 'Tabela')
        //tesxto na tela
    });

```

```

        cy.get('tr').find('th').eq(0) // todos os 6 elementos
(coluna) da tabela
        .should('have.text', 'Id')

        cy.get('tr').find('th').eq(1)
        .should('have.text', 'Nome')

        cy.get('tr').find('th').eq(2)
        .should('have.text', 'Telefone')

        cy.get('tr').find('th').eq(3)
        .should('have.text', 'Nascimento')

        cy.get('tr').find('th').eq(4)
        .should('have.text', 'Sexo')

        cy.get('tr').find('th').eq(5)
        .should('have.text', 'Ultima Atualização')
    });

});

context.only('Cadastrando usuário', () => {
    it('Cadastrando um novo usuário', () => {
        cy.get('input[id="txtNome"]')
            .type(randomNomeFeminino)

        cy.get('input[id="numero"]')
            .type(numeroRandom) /* VEJA BEM O PROBLEMA É QUE VOCE
COLOCOU ASPAS NO MEIO DA VARIVEL*/

        cy.get('input[id="dtpDataNascimento"]')
            .type(formattedDate)

        cy.get('label[for="rdoFeminino"]')
            .click()

        cy.get('input[id="btnCadastrarSalvar"]')
            .click()

        cy.get('button').contains('Alterar').should('be.visible')//
botão Alterar
        //cy.get('button').contains('Excluir').should('be.visible').c
lick() // uma maneira diferente de capturar o botão

        //Cadastrar outro usuário para efetuar a alteração

        cy.get('input[id="txtNome"]')
            .type('Sofia Santos de Oliveira')
    });
});

```

```

        cy.get('input[id="numero"]')
            .type('11973571191')

        cy.get('input[id="dtpDataNascimento"]')
            .type('2010-10-22')

        cy.get('label[for="rdoFeminino"]')
            .click()

        cy.get('input[id="btnCadastrarSalvar"]')
            .click()

        cy.get('button').contains('Alterar').click()

        // cy.get('input[id="numero"]').clear() /*TEM ESSE CLEAR AQUI
        QUE ESTA APAGANDO O CAMPO */

        cy.get('input[id="btnCadastrarSalvar"]')
            .click()
    });
});

context('Validar dados da tabela', () => {
    it('Fazer cadastro e depois validar os Dados', () => {

        cy.get('input[id="txtNome"]')
            .type('Sofia Simone de Oliveira')

        cy.get('input[id="numero"]')
            .type('11973571191')

        cy.get('input[id="dtpDataNascimento"]')
            .type('1966-09-23')

        cy.get('label[for="rdoFeminino"]')
            .click()

        cy.get('input[id="btnCadastrarSalvar"]')
            .click()

        cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').eq(0).invoke('text').then((PegaId) => {
            cy.wrap(PegaId).as('Id da tabela') // função que pega o
            Id na tabela, cy.wrap pra pegar a variável, as cria um alias
            cy.get('tbody[id="tbodyResultados"]').find("tr").find('td')
            '.eq(0)

            .should('have.text', PegaId) // validação da variável
            PegaId que contem o Id da tabela

```

```
    })  
  
    cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').e  
q(1)      .should('have.text', 'Sofia Simone de Oliveira')  
  
    cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').e  
q(2)      .should('have.text', '11973571191')  
  
    cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').e  
q(3)      .should('have.text', '23/09/1966')  
  
    cy.get('tbody[id="tbodyResultados"]').find("tr").find('td').e  
q(4)      .should('have.text', 'F')  
  
    });  
  });  
});
```