Desafio
QA

Elizabeth Poliani Fidencio Nunes
Framework : Cypress
BDD: Cucumber

Como foi construído automação para o desafio

A primeira etapa foi escrever os testes em BDD, conforme especificação do desafio.

```
Feature: Cadastro de usuário

  Scenario: Cadastrar usuário
    Given Acessar página inicial
    When  Verificar componentes presentes
    And   Salvar usuário sem preencher nenhum campo
    Then Deverá exibir mensagem de campos obrigatórios


    When Realizar cadastro do usuário com dados válidos
    Then Exibirá mensagem de usuário cadastro com sucesso

    When Verificar usuário cadastrado no grid
    Then Acessar e editar usuário com nome inválido
    And  Ao salvar irá exibir mensagem de dados inválidos
    And  Inserir nome válido na edição e salvar
    Then Dados serão alterados e salvos com sucesso


    When Selecionar usuário cadastrado e apagar o cadastro
    Then Usuário será excluído com sucesso
```

Após a escrita dos testes, foi separado todos os elementos da tela no arquivo na pasta e2e/elements

```
const usuarioElements = {

    inputNome: '[id="name"]',
    inputEmail: '[id="email"]',
    inputPassword: '[id="password"]',
    btnCadastrar: '[class="btn btn-primary mt-3"]',
    feedbackSucesso: '[class="alert alert-success"]',
    feedbackCamposObrigatorios: '[class="form-text text-danger"]',
    btnAcoes: 'table > tbody > tr:nth-last-child(1) > td:nth-last-child(1) > div > button',
    optionsAcoes: '[class="dropdown-menu"]',
    optionBtnEditar: '.dropdown-menu > a.dropdown-item:nth-child(1)',
    optionBtnExcluir: '.dropdown-menu > a.dropdown-item:nth-last-child(1)',

    btnConfirmarExclusao: '.modal-dialog > .modal-content > .modal-footer > .btn-danger',
    btnFecharModal: '[class="btn btn-secondary"]',
    feedbackRemovidoSucesso: '[class="alert alert-success"]',

    modalEditar: {
        nome: 'input[id^=e_name]',
        email: 'input[id^=e_email]',
        btnSalvar: '[class="btn btn-primary"]',
        feedbackNomeinvalido: '[class="form-text text-danger"]',

    } ,

   tableGrid: 'table > tbody > tr',



}
export default usuarioElements
```

Foi criado também e2e/elements/pages arquivo dos *pageobjects* mais utilizados, nesse caso foi criado da home para não ficar se repetindo no código.

```
const Home = {
  visit() {
    cy.visit("https://qa-test.ticto.io/");
  }
}

export default Home;
```

Agora dentro do e2e/usuario temos o arquivo .js da construção dos testes em si. Aqui inicia-se os *imports* que a tela do cadastro de usuário precisa.

```
cypress > e2e > Usuario > JS usuario.cy.js > ...
1   import { faker } from "@faker-js/faker";
2   import { Given, When, And, Then } from "cypress-cucumber-preprocessor/steps";
3
4   import usuarioElements from "../../support/elements/usuarioElements.po";
5
6   // pageObjects
7   import Home from "../../support/elements/pages/Home";
8
```

Após isso foi adicionado a função do faker para geração de dados aleatórios, como nomes, e-mails, senhas, endereços, etc.
No caso dos testes em si, foram adicionadas as funções de geração e nome completo, e-mail e senha.

```
const person = {
  fullName: faker.name.fullName(),
  email: faker.internet.email(),
  password: faker.internet.password(),
};
```

Foi necessário adicionar uma *exception* na tela, pois na url de testes aparece um erro de *exception* em função de mau funcionamento de código, o que estava impedindo de executar os testes automatizados na tela de editar usuário.

```
Cypress.on("uncaught:exception", (err, runnable) => {
  return false;
});
```

Começando a execução dos testes.
Abaixo temos o acesso à página inicial e ao carregar a verificação dos componentes presentes, para validar carregamento da tela.

```
Given("Acessar página inicial", () => {
  Home.visit();
  cy.wait(2000);
});

When('Verificar componentes presentes', () => {
    cy.get(usuarioElements.inputNome).should('be.visible');
    cy.get(usuarioElements.inputEmail).should('be.visible');
    cy.get(usuarioElements.btnCadastrar).should('be.visible');
})
```

Assim que validado os componentes executei a ação de salvar sem preencher nenhum campo, para checar se o sistema irá permitir salvar dados vazios e validação dos campos obrigatórios.
Quando a ação de salvar é realizada deve exibir as mensagens de alerta para o usuário sobre a obrigatoriedade dos campos.

```
And ('Salvar usuário sem preencher nenhum campo', () =>{
    cy.get(usuarioElements.btnCadastrar).should('be.visible').click();
})

Then ('Deverá exibir mensagem de campos obrigatórios', () =>{
    cy.get(usuarioElements.feedbackCamposObrigatorios).contains('O campo Nome é obrigatório.');
    cy.get(usuarioElements.feedbackCamposObrigatorios).contains('O campo Email é obrigatório.');
    cy.get(usuarioElements.feedbackCamposObrigatorios).contains('O campo Password é obrigatório.');
})
```

Checando essas informações irá realizar o cadastro com sucesso, usando o faker para preencher essas informações e salvar. Ao salvar deverá checar mensagem de cadastro realizado com sucesso.

```
When ('Realizar cadastro do usuário com dados válidos', () => {
    cy.get(usuarioElements.inputNome).should('be.visible').type(person.fullName);
    cy.get(usuarioElements.inputEmail).should('be.visible').type(person.email);
    cy.get(usuarioElements.inputPassword).should('be.visible').type(person.password);
    cy.get(usuarioElements.btnCadastrar).should('be.visible').click();
})

Then ('Exibirá mensagem de usuário cadastro com sucesso', () => {
    cy.wait(1000);
    cy.get(usuarioElements.feedbackSucesso).contains('Usuário cadastrado com sucesso.');
})
```

Ao checar a mensagem de sucesso o teste irá percorrer no grid de usuários cadastrados e chegar que o usuário de fato foi cadastrado, buscando-o no grid.

```
When("Verificar usuário cadastrado no grid", () => {
  cy.get(usuarioElements.tableGrid).last().contains(person.fullName);
  cy.get(usuarioElements.tableGrid).last();
});
```

Após a checagem no grid, vamos acessar esse usuário e editá-lo inserindo nome inválido,o campo nome verifica nome e sobrenome e no teste inserimos na edição apenas o primeiro nome. Adicionamos o primeiro nome usando o *faker* e salvamos, a tela deverá exibir o alerta para usuário que é necessário nome e sobrenome válidos.

```
Then("Acessar e editar usuário com nome inválido", () => {
  cy.get(usuarioElements.btnAcoes)
    .click()
    .get(usuarioElements.optionBtnEditar)
    .should("be.visible")
    .last()
    .click();

  cy.get(usuarioElements.modalEditar.nome)
    .should("be.visible")
    .last()
    .clear()
    .type(faker.name.firstName());
});

And("Ao salvar irá exibir mensagem de dados inválidos", () => {
  cy.get(usuarioElements.modalEditar.btnSalvar)
    .should("be.visible")
    .last()
    .click();

  cy.wait(1000);

  cy.get(usuarioElements.modalEditar.feedbackNomeinvalido)
    .should("be.visible")
    .contains("Insira um Nome e Sobrenome válido.");
});
```

Validando as informações, adicionamos novamente o nome completo usando o faker e salvamos com sucesso a edição do usuário. Irá exibir mensagem de alerta que usuário foi salvo com sucesso.

```
And("Inserir nome válido na edição e salvar", () => {
  cy.get(usuarioElements.modalEditar.nome)
    .should("be.visible")
    .last()
    .clear()
    .type(person.fullName);

  cy.get(usuarioElements.modalEditar.btnSalvar)
    .should("be.visible")
    .last()
    .click();
});

Then("Dados serão alterados e salvos com sucesso", () => {
  cy.wait(1000);
  cy.get(usuarioElements.feedbackSucesso).contains(
    "Usuário salvo com sucesso."
  );
});
```

Após a edição, vamos acessar novamente o cadastro desse usuário e excluí-lo, após a exclusão foi validado a exclusão do usuário e checado o alerta de retorno de usuário removido com sucesso.

```
When("Selecionar usuário cadastrado e apagar o cadastro", () => {
  cy.get(usuarioElements.btnAcoes)
    .click()
    .get(usuarioElements.optionBtnExcluir)
    .last()
    .click();
  cy.wait(300);
  cy.get(usuarioElements.btnConfirmarExclusao)
    .last()
    .should("be.visible")
    .click();
});

Then("Usuário será excluído com sucesso", () => {
  cy.wait(300);
  cy.get(usuarioElements.feedbackRemovidoSucesso).contains(
    "Usuário removido com sucesso."
  );
});
```