Módulo 11 cypress e UI

Para começarmos, vamos baixar o nodejs no site nodejs.org

Abrir o VSCode, abrir terminal.

No github criar um novo repositório de nome teste-ebac-ui (adicionar o arquivo readme.me e o .gitignore), também escolher a opção de tema node.

No VSCode no terminal digite

Git clone https://github.com/SimoneSS15/teste-ebac-ui.git

E clone o repositório para a sua máquina

No terminal digite :

Cd teste-ebac-ui - para entrar na pasta

Agora digite:

Node –version - para verificar a instalação do nodejs

Npm init -y - para inicializar o node dentro da pasta teste-ebac-ui esse comando cria um arquivo de nome package.json que controla tudo.

Agora instale o cypress:

Npm install cypress -D - para instalar o cypress (vai ser instalada a última versão), caso queira versões anteriores a atual basta digitar:

Npx install cypress@9.7.0

Npm audit fix – para resolver problemas de vulnerabilidade

Para desinstalar uma versão do cypress basta digitar:

Npx uninstall cypress

Vai ser criada uma pasta de nome node_modules com todas as bibliotecas a serem usadas, essa pasta fica ignorada e não vai para o github.

Agora digite:

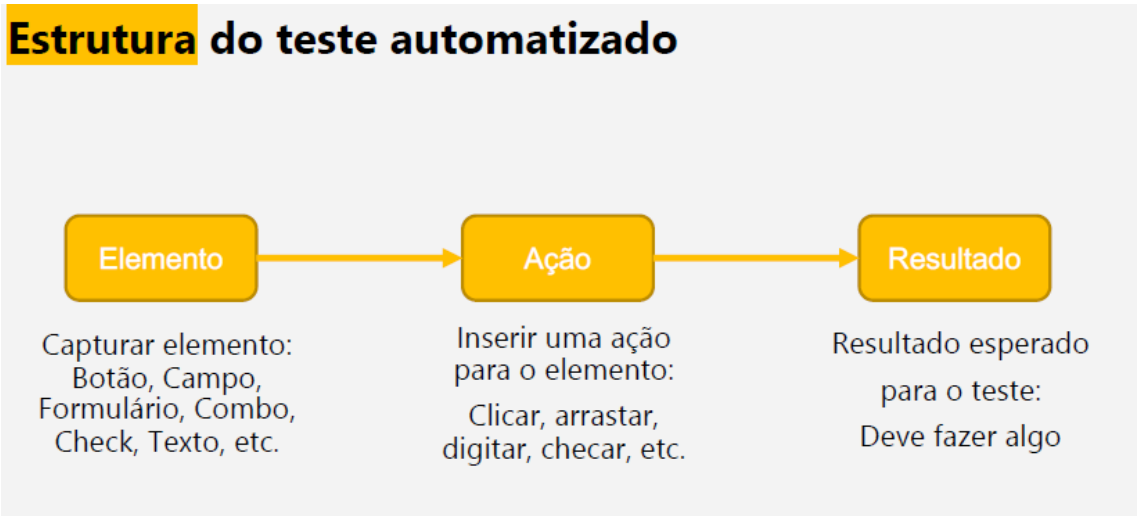
Npx cypress open – irá abrir o cypress

Agora volte para o VSCode e instale algumas extensões:

Plugins recomendados:

- **ES6 Mocha**
- **Bracket Pair Colorizer – para colorir o código.**

- **Cypress Snippets**
- **Material Icon Theme – perfumaria.**
- **Visual Studio IntelliCode – para auto completar os códigos.**



Vamos criar nossa primeira automação

Crie um arquivo dentro da pasta e2e de nome login.spec.js (nome do arquivo e a extensão de arquivos de testes em cypress), se estiver utilizando cypress versão 10 ou acima o arquivo deve ter o nome de login.cy.js

No arquivo digitar a linha abaixo:

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

Trata-se de uma referência do cypress dentro do arquivo, para ele funcionar.

Digite todo o código abaixo:

```
/// <reference types="cypress"/>
context('funcionalidade Login', ()=> { //declaração do contexto
    it('Deve fazer login com sucesso', ()=> {// dentro desse bloco será digitado os comandos dos testes

    })

    it('Deve exibir uma mensagem de erro ao inserir usuario ou senha inválidos', ()=>{

    })
})
```

Por enquanto existem duas estruturas de testes, que são os cenários, o caminho feliz e o caminho negativo, como uma boa prática sempre crie os cenários primeiro antes de sair digitando os códigos de teste.

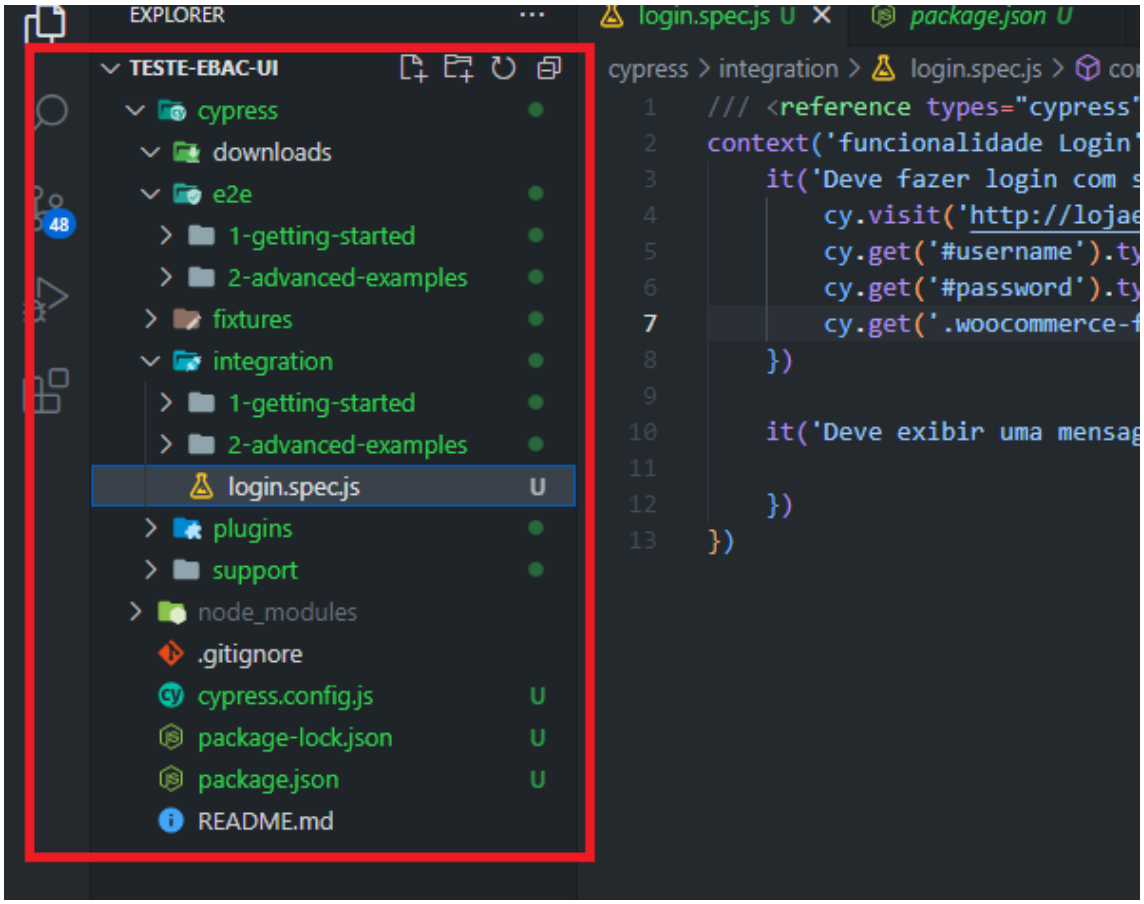
Acesse o site http://lojaebac.ebaconline.art.br/ e fazer o teste manual de login (sempre fazer o teste manual primeiro) assim saberá fazer os casos de testes necessários para a automação.

Logar com os dados:

Aluno_ebac@teste.com

Senha: teste@teste.com

Observação importante:



Se estiver usando o cypress versão 12.2.0 o arquivo de teste deve ter a extensão .cy.js (login.cy.js), se estiver usando uma versão antiga, antes da versão 10 o arquivo deve ter a extensão .spec.js (login.spec.js) e esse arquivo deve estar dentro da pasta e2e na versão 10 ou 12.2.0, e na versão anterior a 10 tem que estar dentro da pasta Integration como mostra a imagem acima. Fiz a maior confusão com isso!

Adicionando a linha abaixo no código, o teste fica completo, pois até aqui houve uma captura de elementos, uma ação mas não teve um resultado, adicionando o código abaixo, haverá uma validação de que o login foi efetuado com sucesso, isso pode ser feito capturando qualquer elemento da tela, no caso do exemplo foi o texto "Minha conta" que deve (should) contain (conter) o texto "minha conta" na tela, observe o código abaixo:

```
  cy.get('.page-title').should('contain', 'Minha
conta')
```

mais uma linha de Código que identifica um texto apresentado na tela, apenas para validar o login efetuado com sucesso:

```
 cy.get('.woocommerce-MyAccount-content > :nth-
child(2)').should('contain', 'Olá, aluno_ebac (não é aluno_ebac? Sair')
```

Agora implemente o teste onde o login e a senha apresentem erros, segue código abaixo:

```
it('Deve exibir uma mensagem de erro ao inserir usuario ou senha
inválidos', ()=>{
        cy.visit('http://lojaebac.ebaconline.art.br/minha-conta/')
        cy.get('#username').type('_ebac@teste.com')
        cy.get('#password').type('teste@teste')
        cy.get('.woocommerce-form > .button').click()
    })
```

Agora separe os dois testes de acesso, primeiro erro no usuário e depois erro na senha, conforme código abaixo, basta efetuar algumas alterações:

```
  it('Deve exibir uma mensagem de erro ao inserir usuario inválido',
()=>{
        cy.visit('http://lojaebac.ebaconline.art.br/minha-conta/')
        cy.get('#username').type('_ebac@teste.com')
        cy.get('#password').type('teste@teste.com')
        cy.get('.woocommerce-form > .button').click()
    })
    it('Deve exibir uma mensagem de erro ao inserir senha inválida',
()=>{
        cy.visit('http://lojaebac.ebaconline.art.br/minha-conta/')
        cy.get('#username').type('aluno_ebac@teste.com')
        cy.get('#password').type('teste@teste')
        cy.get('.woocommerce-form > .button').click()
```

```
    })
```

Se digitar ".only" (apenas, somente) depois do It ele executa apenas o teste especifico, veja o código abaixo:

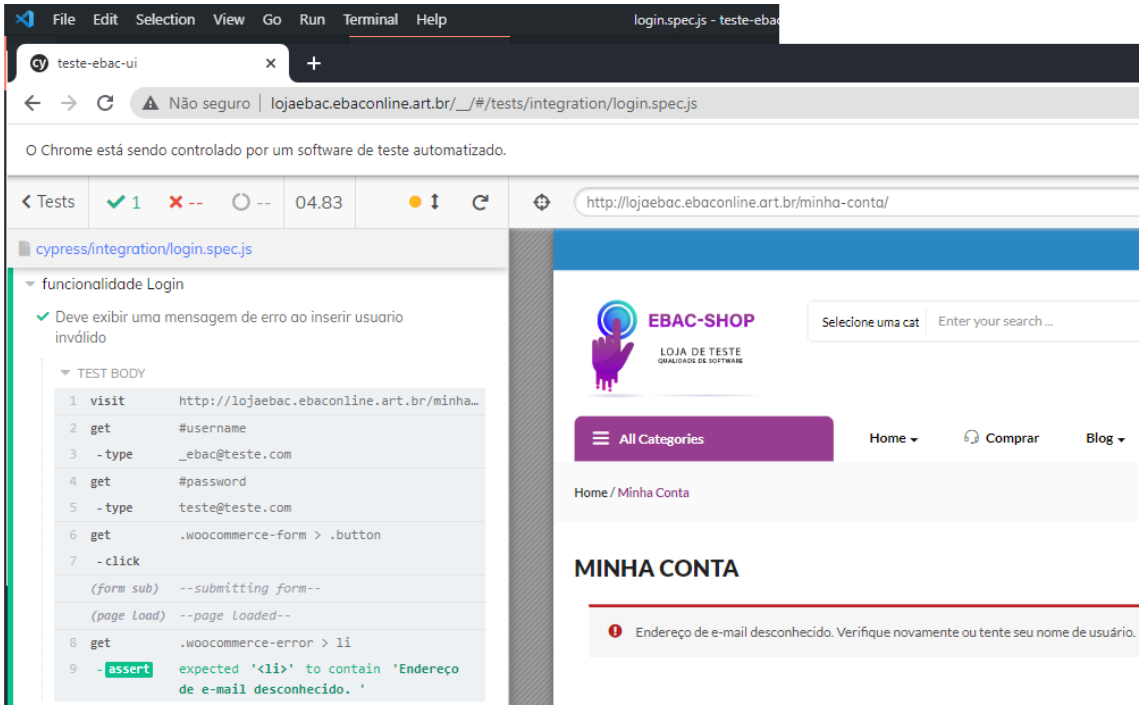
```
 it.only('Deve exibir uma mensagem de erro ao inserir usuario
inválido', ()=>{
        cy.visit('http://lojaebac.ebaconline.art.br/minha-conta/')
        cy.get('#username').type('_ebac@teste.com')
        cy.get('#password').type('teste@teste.com')
        cy.get('.woocommerce-form > .button').click()
    })
```

Ainda neste teste-, acrescente a linha abaixo para validar a mensagem de erro apresentada sobre o usuário inválido:

```
 cy.get('.woocommerce-error > li').should('contain', 'Endereço de e-mail
desconhecido. ')
```

observação: trata-se de um teste negativo, mas o cypress tem que passar o teste de validação da mensagem de erro!

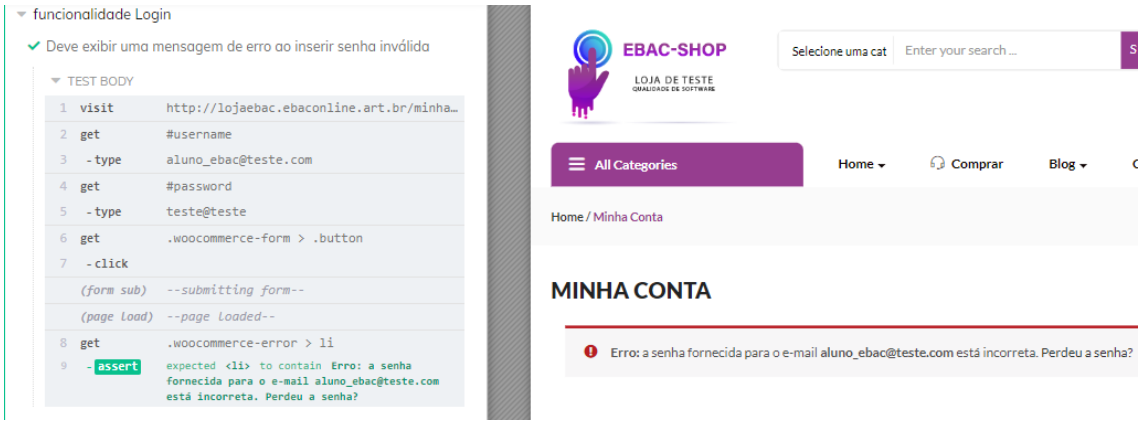
Veja o teste no cypress:



Agora tire o “Only” do teste de erro do usuário e coloca no teste de erro da senha, execute, capture a mensagem de erro e adicione no código, conforme mostrado abaixo:

```
 it.only('Deve exibir uma mensagem de erro ao inserir senha inválida',
()=>{
        cy.visit('http://lojaebac.ebaconline.art.br/minha-conta/')
        cy.get('#username').type('aluno_ebac@teste.com')
        cy.get('#password').type('teste@teste')
        cy.get('.woocommerce-form > .button').click()
        cy.get('.woocommerce-error > li').should('contain', 'Erro: a
senha fornecida para o e-mail aluno_ebac@teste.com está incorreta. Perdeu
a senha?')
    })
```

Acontecerá o mesmo que aconteceu com o teste anterior, a mensagem de erro sobre a senha inválida será validada pelo cypress como mostra imagem abaixo:



Para finalizar execute totós os testes (tire o , only).

Suba tudo para o github:

Git status – visualizer o status da pasta

Git add . – adicionar os arquivos no Git

Git commit -m “Meu commit de teste cypress” – efetuar o commit (commitar)

Git push origin main – empurrar os arquivos para o repositório remoto Github