

007



QA's 007: Permissão para Testar

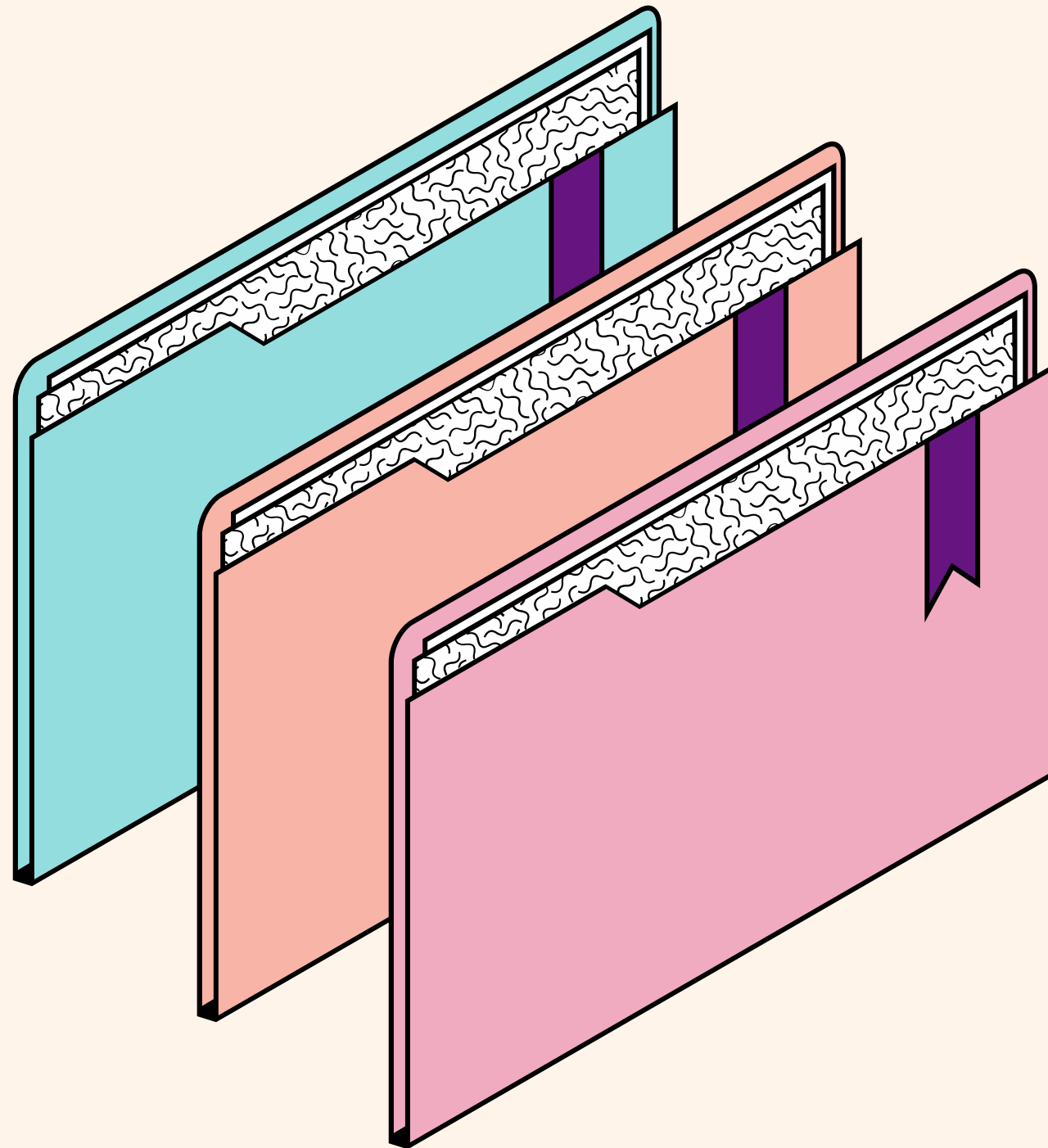


AUTOMATION EXERCISE

QA's 007: Permissão para Testar

Maria Luchecy Ribeiro de Araujo
Quézia Menezes Filadelfo
Samuel Araujo Carvalho Roch
Mauro Victor Pimentel Diasa
Yann Gabriel Pires Herval dos Anjos

Índice



1. Conhecendo o Produto
2. Objetivos do Trabalho
3. Tipos de Testes
4. Níveis de Teste
5. Metodologia de Teste
6. Especificações de Testes
7. Modelo utilizado
8. Artefatos desenvolvidos
9. Testes Automatizados
10. Estatísticas Gerais de Testes e resultados
11. Desafios e soluções
12. Aprendizados e Melhorias

Conhecendo o Produto

O Automation Exercise é uma plataforma de simulação de e-commerce, projetada para que profissionais de QA, desde iniciantes até avançados, pratiquem testes funcionais, de automação e de APIs.

Home

Products

Cart

Signup / Login

Test Cases

API Testing

Video Tutorials

Contact us

AutomationExercise

Full-Fledged practice website for Automation Engineers

All QA engineers can use this website for automation practice and API testing either they are at beginner or advance level. This is for everybody to help them brush up their automation skills.

Test Cases

APIs test for practice

WOMEN

MEN

KIDS

POLO

H&M

MADAME

MAST & HARBOUR

BABYHUG

ALLEN SOLLY JUNIOR

Blue Top

Rs. 500

Add to cart

View Product

FEATURES ITEMS

Blue Top

Man with glasses

Woman in black top

Objetivos

Este projeto foi desenvolvido para garantir a excelência da aplicação Automation Exercise, focando em:

- **Qualidade e Desempenho:** Assegurar uma aplicação completa e eficiente.
- **Identificação de Defeitos:** Detectar e classificar falhas que impactam os usuários finais.
- **Conformidade:** Cumprir rigorosamente as especificações de teste e os requisitos de qualidade.



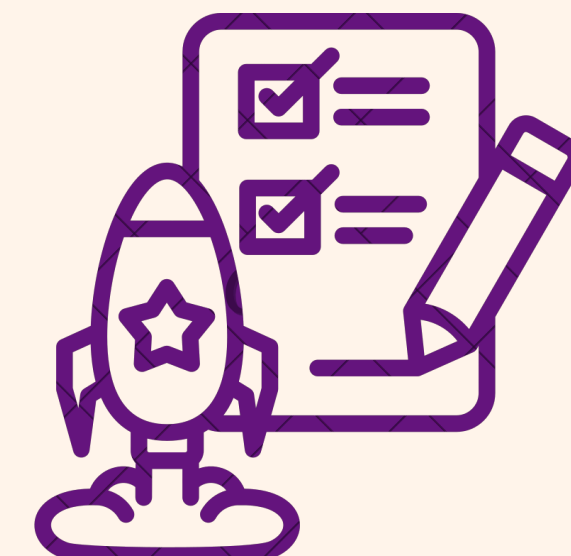
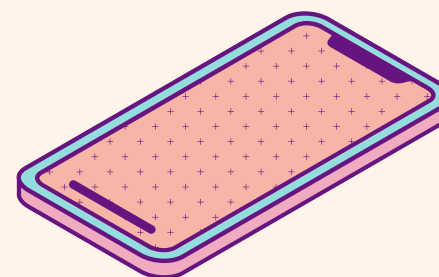
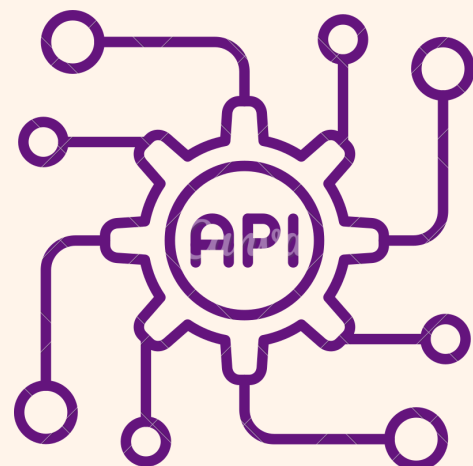


Tipos de Testes



Os tipos de teste adotados incluem:

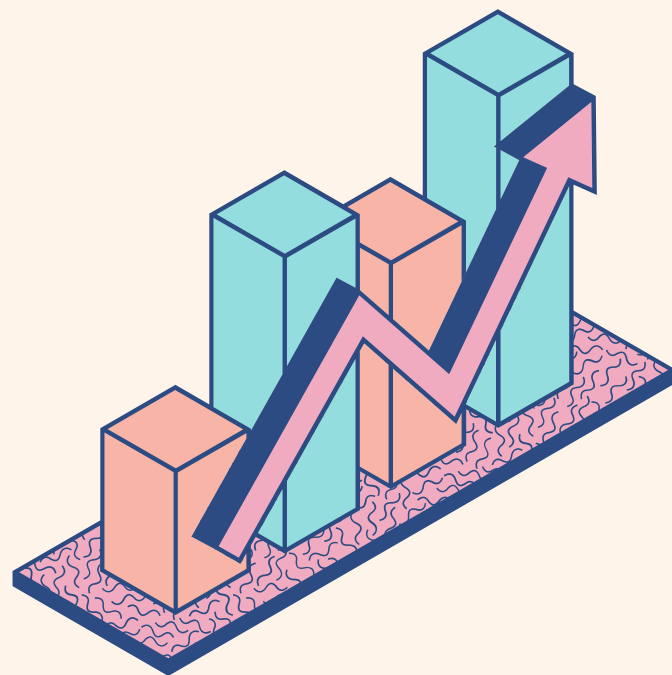
1. **Manuais:** Execução de casos de teste sem automação, focando em validações subjetivas e exploratórias.
2. **Funcionais:** Validação das funcionalidades do sistema.
3. **API:** Garantia de comunicação e segurança dos dados.
4. **Automação:** Repetição de casos de teste para maior cobertura.



Níveis de Testes

Testes de Aceitação

- Testes de Aceitação: Englobam a validação final, onde o sistema é testado contra os requisitos do usuário para garantir que está pronto para o uso real.



Testes de Sistema:

- Integração: Verificação da interação entre partes do sistema.
- Sistema: Avaliação do sistema completo.
- Baseados na Experiência: Testes exploratórios e intuitivos.

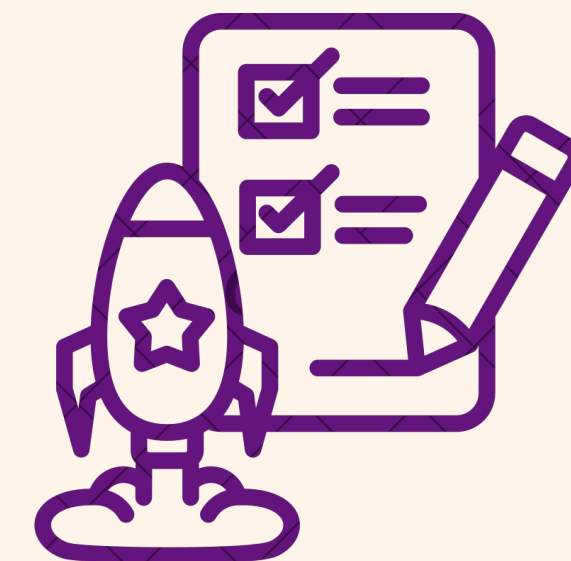
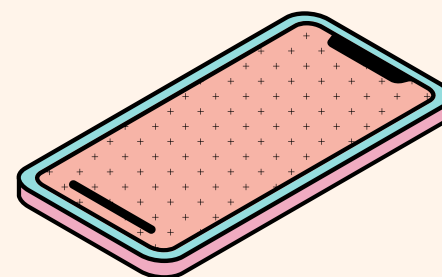


Metodologia de Testes



As metodologias de teste adotadas foram:

1. **Caixa Preta:** Foco na camada de front-end, avaliando a funcionalidade sem olhar para o código interno.
2. **Automatizados:** Execução repetitiva e consistente de casos de teste, garantindo eficiência e cobertura ampla.
3. **Testes de API:** Validação da comunicação entre sistemas e a segurança dos dados transmitidos.



Especificações de Testes

Cenários de planejamento de testes definidos para as especificações de teste.

Cenário	Descrição	Cenário	Descrição
1	Login	7	Atualizar Conta do Usuário
2	Cadastro de Usuário	8	Buscar Usuário
3	Formulário de Contato	9	Excluir Conta do Usuário
4	Produto	10	Obter detalhes da Conta do Usuário por E-mail
5	Carrinho	11	Verificação de Acessibilidade
6	Obter Lista de todos os Produtos		

Modelo Utilizado

CENÁRIO XX - Nome do Cenário	
CT ID	Nome da pessoa responsável pelo teste
Nome do CT	Uma descrição concisa e clara do objetivo do teste, detalhando o que está sendo avaliado.
Prioridade	Nível de importância do teste, classificando-o como Alta, Média ou Baixa, para ajudar na priorização dos testes.
Ambiente	Descrição do ambiente onde o teste será executado, incluindo os detalhes.
Pré -condições	Condições necessárias para que o teste possa ser executado.
Dados de Entrada	Dados necessários para a execução do teste, incluindo formatos e valores específicos.
Passo-a-passo	Sequência detalhada de ações a serem realizadas para executar o teste, incluindo cada etapa necessária.
Resultado Esperado	Descrição do comportamento ou resultado que se espera do sistema após a execução do teste.
Resultado Atual	Descrição do resultado atual do sistema.
Status	Indicação do estado do teste após a execução, que pode ser: passado, falhado, bloqueado e aguardando execução.
Observações	Notas e observações adicionais que ajudam a entender o resultado obtido.

Exemplo de Cenário de Teste

CENÁRIO 01 – Login	
CT ID	CT001
Autor(a)	Yann Gabriel
Nome do CT	Login com Credenciais Corretas
Prioridade	Alta
Ambiente	Google Chrome versão 126.0.6478.182/ Windows 11
Pré-condições	O usuário deve estar na tela de login: https://automationexercise.com/login
Dados de Entrada	Email: lidiasilva@gmail.com Senha: 30032004
Passo a Passo	<ol style="list-style-type: none">1. Preencher o campo “Email Address” com um endereço email cadastrado.2. Preencher o campo “Password” com a senha cadastrada.3. Clicar no botão “Login”.
Resultado Esperado	O sistema deve autenticar login/senha e redirecionar para a página “home”.
Resultado Atual	Teste ainda não executado.
Status	Aguardando execução.
Observação	N/A

Ferramentas

Ferramentas	
Captura de Tela	OBS Studio e Lightshot
Gestão de Tarefas	Trello
Comunicação	Discord e WhatsApp
Testes de API	Postman - Versão 11.6.2
Testes Automatizados	Cypress e Jest
Gráficos e Tabelas	Planilhas do Google Sheets e Google Docs
Teste de Acessibilidade	WAVE, Lighthouse

Artefatos Desenvolvidos



Plano de Teste

Automation Exercise

Professor: Francisco Gutenberg
Monitor: William Filho

Equipe 4 - QA's 007: Permissão para
Mauro Victor Pimentel Dias
Maria Luchecy Ribeiro de Araujo
Quêzia Menezes Filadelfo
Samuel Araujo Carvalho Rocha
Yann Gabriel Pires Herval dos Anjos

Agosto/2024



Especificação de Testes

Automation Exercise

Professor: Francisco Gutenberg
Monitor: William Filho

Equipe 4 - QA's 007: Permissão para
Mauro Victor Pimentel Dias
Maria Luchecy Ribeiro de Araujo
Quêzia Menezes Filadelfo
Samuel Araujo Carvalho Rocha
Yann Gabriel Pires Herval dos Anjos
Fiama Alves da Silva

Agosto/2024



Relatório de Testes

Automation Exercise

Professor: Francisco Gutenberg
Monitor: William Filho

Equipe 4 - QA's 007: Permissão para
Mauro Victor Pimentel Dias
Maria Luchecy Ribeiro de Araujo
Quêzia Menezes Filadelfo
Samuel Araujo Carvalho Rocha
Yann Gabriel Pires Herval dos Anjos
Fiama Alves da Silva

Agosto/2024



Relatório de Bugs

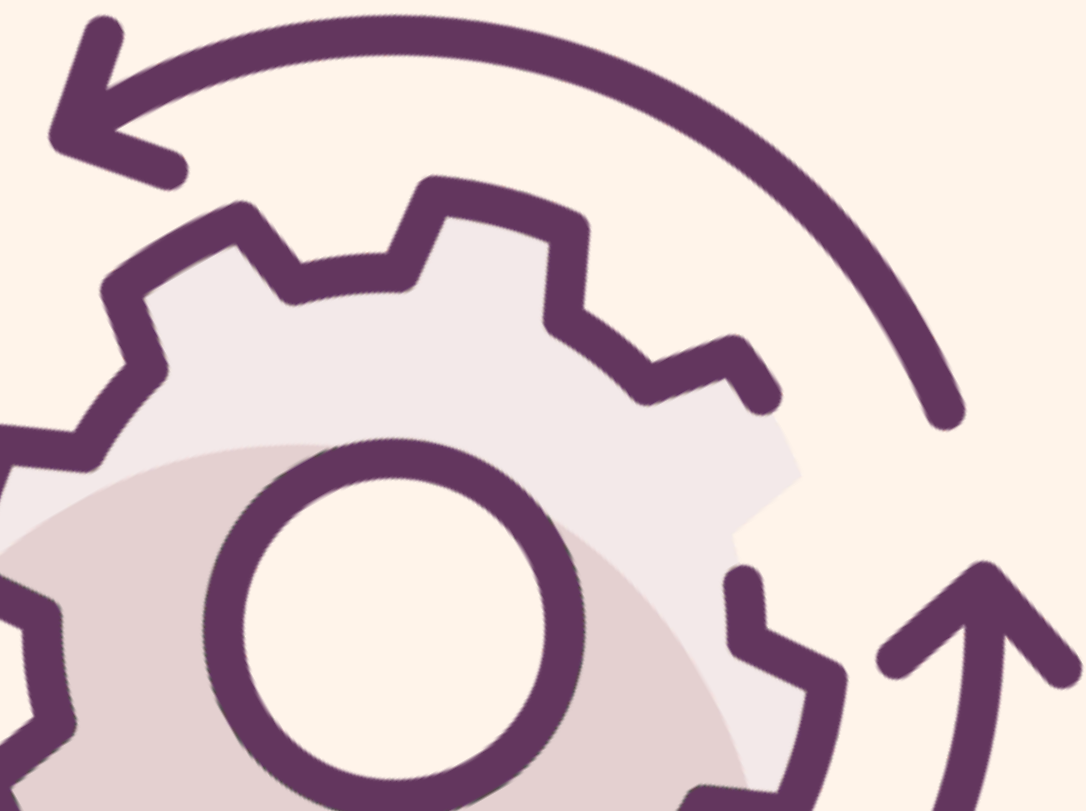
Automation Exercise

Professor: Francisco Gutenberg
Monitor: William Filho

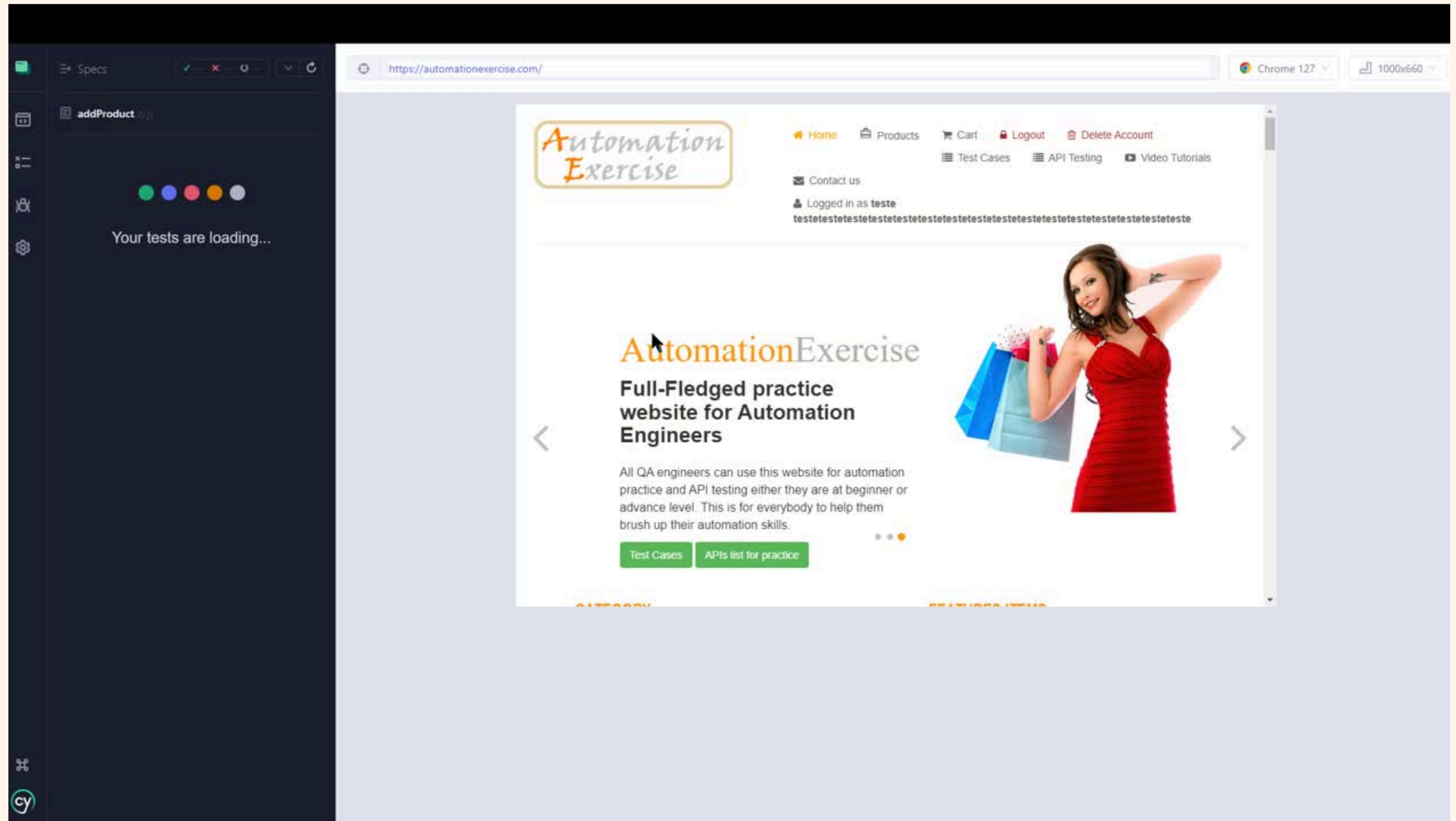
Equipe 4 - QA's 007: Permissão para Testar
Mauro Victor Pimentel Dias
Maria Luchecy Ribeiro de Araujo
Quêzia Menezes Filadelfo
Samuel Araujo Carvalho Rocha
Yann Gabriel Pires Herval dos Anjos
Fiama Alves da Silva

Agosto/2024

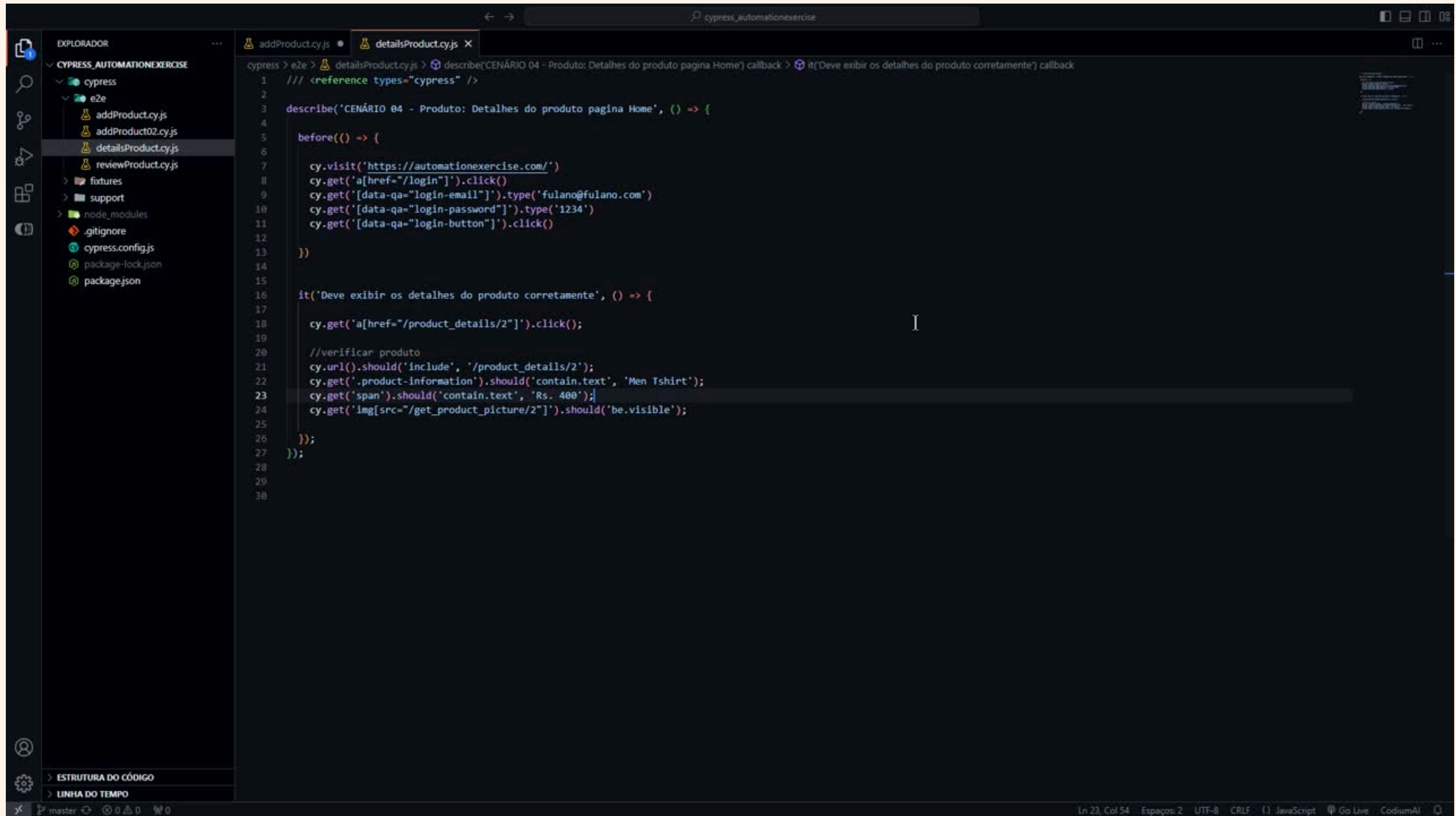
Automação



Adicionar Produto no Carrinho



Acessar Detalhes do Produto

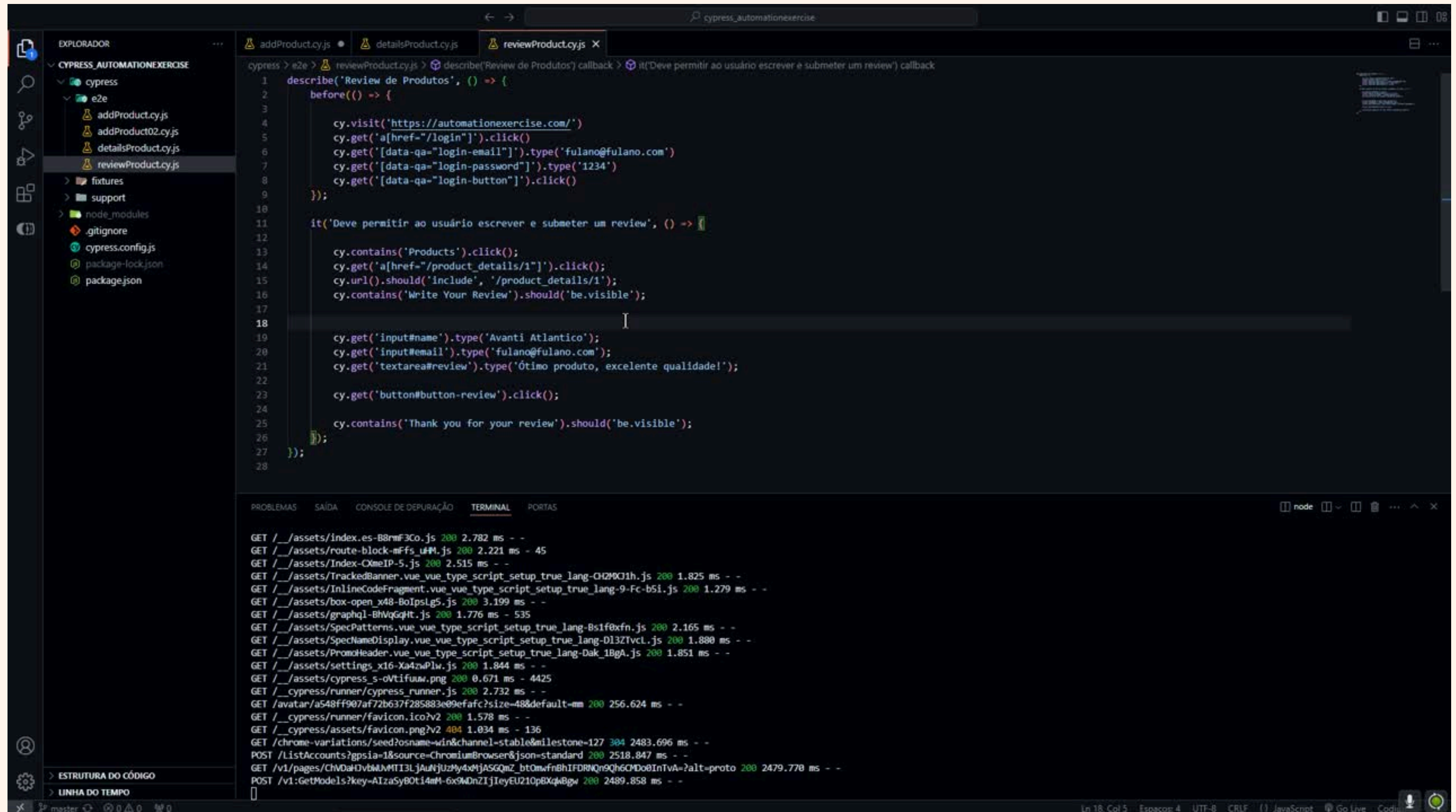


The image shows a VS Code editor with a Cypress test file named `detailsProduct.cy.js` open. The file explorer on the left shows the project structure for `CYPRESS_AUTOMATIONEXERCISE`, including `cypress`, `e2e`, `fixtures`, `support`, `node_modules`, `.gitignore`, `cypress.config.js`, `package-lock.json`, and `package.json`. The test code in the editor is as follows:

```
1  /// <reference types="cypress" />
2
3  describe('CENÁRIO 04 - Produto: Detalhes do produto pagina Home', () => {
4
5    before(() => {
6
7      cy.visit('https://automationexercise.com/')
8      cy.get('a[href="/login"]').click()
9      cy.get('[data-qa="login-email"]').type('fulano@fulano.com')
10     cy.get('[data-qa="login-password"]').type('1234')
11     cy.get('[data-qa="login-button"]').click()
12
13   })
14
15   it('Deve exibir os detalhes do produto corretamente', () => {
16
17     cy.get('a[href="/product_details/2"]').click();
18
19     //verificar produto
20     cy.url().should('include', '/product_details/2');
21     cy.get('.product-information').should('contain.text', 'Men Tshirt');
22     cy.get('span').should('contain.text', 'Rs. 400');
23     cy.get('img[src="/get_product_picture/2"]').should('be.visible');
24
25   });
26
27 });
28
29
30
```

The status bar at the bottom indicates the current file is `Ln 23, Col 54`, using `UTF-8` encoding with `CRLF` line endings, and the language is `JavaScript`.

Cenário 5 - Review de Produtos



The screenshot displays the VS Code editor with a Cypress test suite for 'Review de Produtos'. The Explorer on the left shows the project structure, including the 'reviewProduct.cy.js' file. The main editor shows the test code, which includes a 'before' hook for login and an 'it' block for the review process. The terminal at the bottom shows the network logs of the application being tested.

```
cypress > e2e > reviewProduct.cy.js > describe('Review de Produtos') callback > it('Deve permitir ao usuário escrever e submeter um review') callback
```

```
1 describe('Review de Produtos', () => {
2   before(() => {
3
4     cy.visit('https://automationexercise.com/')
5     cy.get('a[href="/login"]').click()
6     cy.get('[data-qa="login-email"]').type('fulano@fulano.com')
7     cy.get('[data-qa="login-password"]').type('1234')
8     cy.get('[data-qa="login-button"]').click()
9   });
10
11   it('Deve permitir ao usuário escrever e submeter um review', () => {
12
13     cy.contains('Products').click();
14     cy.get('a[href="/product_details/1"]').click();
15     cy.url().should('include', '/product_details/1');
16     cy.contains('Write Your Review').should('be.visible');
17
18
19     cy.get('input#name').type('Avanti Atlantico');
20     cy.get('input#email').type('fulano@fulano.com');
21     cy.get('textarea#review').type('Ótimo produto, excelente qualidade!');
22
23     cy.get('button#button-review').click();
24
25     cy.contains('Thank you for your review').should('be.visible');
26   });
27 });
28
```

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO **TERMINAL** PORTAS

```
GET /__assets/index.es-B8rmF3Co.js 200 2.782 ms - -
GET /__assets/route-block-mFfs_u4M.js 200 2.221 ms - 45
GET /__assets/Index-CKmeIP-5.js 200 2.515 ms - -
GET /__assets/TrackedBanner.vue_vue_type_script_setup_true_lang-CH2MX01h.js 200 1.825 ms - -
GET /__assets/InlineCodeFragment.vue_vue_type_script_setup_true_lang-9-Fc-b5i.js 200 1.279 ms - -
GET /__assets/box-open_x48-BoIpsLg5.js 200 3.199 ms - -
GET /__assets/graphql-BhVqGqHt.js 200 1.776 ms - 535
GET /__assets/SpecPatterns.vue_vue_type_script_setup_true_lang-Bs1f0xfn.js 200 2.165 ms - -
GET /__assets/SpecNameDisplay.vue_vue_type_script_setup_true_lang-Dl3ZTvcL.js 200 1.880 ms - -
GET /__assets/PromoHeader.vue_vue_type_script_setup_true_lang-Dak_1BgA.js 200 1.851 ms - -
GET /__assets/settings_x16-Xa4zwPlw.js 200 1.844 ms - -
GET /__assets/cypress_s-oVtifuuW.png 200 0.671 ms - 4425
GET /__cypress/runner/cypress_runner.js 200 2.732 ms - -
GET /avatar/a548ff907af72b637f285883e09efafc?size=48&default=mm 200 256.624 ms - -
GET /__cypress/runner/favicon.ico?v2 200 1.578 ms - -
GET /__cypress/assets/favicon.png?v2 404 1.034 ms - 136
GET /chrome-variations/seed?osname=win&channel=stable&milestone=127 304 2483.696 ms - -
POST /ListAccounts?gpsia=1&source=ChromiumBrowser&json=standard 200 2518.847 ms - -
GET /v1/pages/ChVdaIdvBMUMTI3LjAuNjUzMy4xMjJASGQmZ_btOmwfN8hIFDRnQn9Qh6CMD08InTvA=?alt=proto 200 2479.770 ms - -
POST /v1:GetModels?key=AIzaSyBOTi4mM-6x9WDnZIJieyEU210pBXqkBgw 200 2489.858 ms - -
```

ESTRUTURA DO CÓDIGO
LINHA DO TEMPO

Ln 18, Col 5 | Espaços: 4 | UTF-8 | CRLF | () JavaScript | Go Live | Codio

Testes de API

- Uso do Framework Jest



Jest é um poderoso Framework de Testes em JavaScript com um foco na simplicidade.

Funciona com projetos usando: [Babel](#), [TypeScript](#), [Node](#), [React](#), [Angular](#), [Vue](#) e muito mais!

Automação - Testando API

The screenshot displays the Visual Studio Code interface for a project named 'automationExerciseTests'. The Explorer sidebar on the left shows the project structure, including an 'api' folder with 'requests' and 'tests' subfolders. The 'tests' folder contains 'getProducts.test.js' and 'getUserDetails.test.js'. The main editor window shows the 'getUserDetails.test.js' file, which contains a Jest test suite. The test suite includes a 'describe' block for 'Cenário 10 - Obter Detalhes da Conta do Usuário por E-mail' and a 'test' block that checks if 'Get User Details' returns an empty response for an invalid email. The test code is as follows:

```
3 describe("Cenário 10 - Obter Detalhes da Conta do Usuário por E-mail", () => {
30   });
31
32   test("Get User Details não deve retornar nenhum dado para email invalido", async () => {
33
34     const emailInvalido = "a"
35
36     const _response = await getUserDetails(emailInvalido);
37     const _data = await _response.json();
38
39     expect(_data.user.id).not.toBeGreaterThanOrEqual(0);
40   });
41 }
```

The bottom panel shows the 'TERMINAL' output, which displays the command 'npm test' and the resulting Jest output. The output indicates that the test suite is 0 of 2 total, with 0 tests, 0 snapshots, and a time of 1 s, estimated 5 s. A progress bar is shown at the bottom of the terminal output.

PS: automationExerciseTests> npm test

> automationexercisetests@1.0.0 test

> jest --verbose

RUNS api/tests/getUserDetails.test.js

RUNS api/tests/getProducts.test.js

Test Suites: 0 of 2 total

Tests: 0 total

Snapshots: 0 total

Time: 1 s, estimated 5 s

Ln 39, Col 61 Spaces: 4 UTF-8 CRLF {} JavaScript

27°C Parc. nublado 22:08:50 23/08/2024

Automação - Resultado

```
> automationexercisetests@1.0.0 test  
> jest --verbose
```

PASS api/tests/getProducts.test.js

Cenário 06 - Obter Lista de Todos os Produtos

- ✓ Get Products deve retornar status code de 200 (40 ms)
- ✓ Get Products deve retornar responseCode de 200 (3 ms)
- ✓ Get Products deve conter uma lista de produtos (2 ms)
- ✓ Get Products deve conter uma lista de produtos não vazia (4 ms)

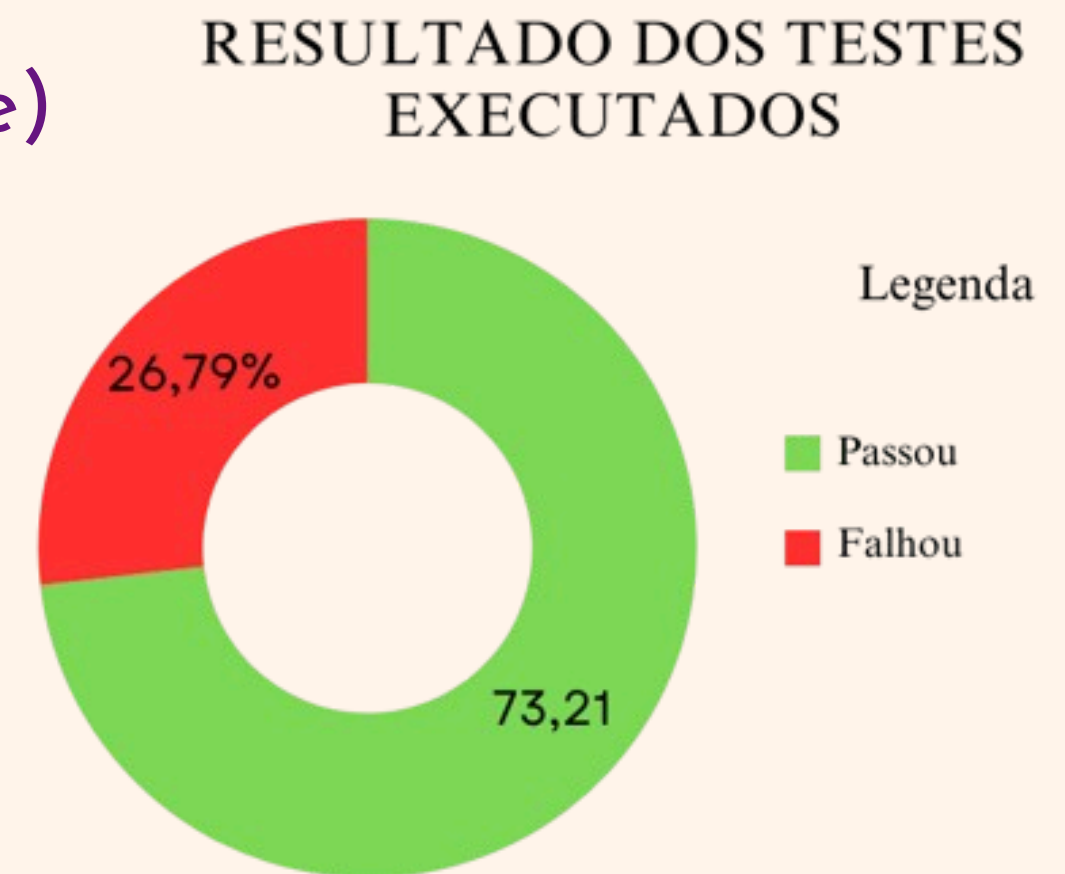
FAIL api/tests/getUserDetails.test.js

Cenário 10 - Obter Detalhes da Conta do Usuário por E-mail

- ✓ Get User Details deve retornar status code de 200 para email valido (55 ms)
- ✓ Get User Details deve retornar responseCode de 200 para email valido (3 ms)
- ✗ Get User Details deve retornar responseCode de 400 para email invalido (268 ms)
- ✗ Get User Details não deve retornar nenhum dado para email invalido (878 ms)

Estatísticas gerais de Testes e Resultados

- Total de Testes Executados: 56 em 11 cenários.
- Testes Bem-Sucedidos: 41 (73%)
- Testes com Falhas: 15 (26%)
- Cenários com Falhas: 8 dos 11 cenários testados
- Cenário mais Problemático: Cenário 11 (Acessibilidade)



Principais desafios e Soluções

- **Gestão de Tempo**
 - **Solução:**
 - **Cronograma Detalhado**
 - **Ferramentas de Gerenciamento**
- **Desistência de Membros do Time:**
 - **Solução:**
 - **Foco na comunicação constante entre o time**

Aprendizados e Melhorias finais

- **Melhorias futuras:**
 - Adaptabilidade;
 - Expansão de cobertura de teste;
- **Aprendizados :**
 - Funcionamento de processos de testes;
 - Documentação;
 - Automatização de testes;
 - Integração de novas ferramentas;
 - Colaboração em equipe.

Muito obrigada pela atenção!



Mauro Victor
Engenheiro
de Teste



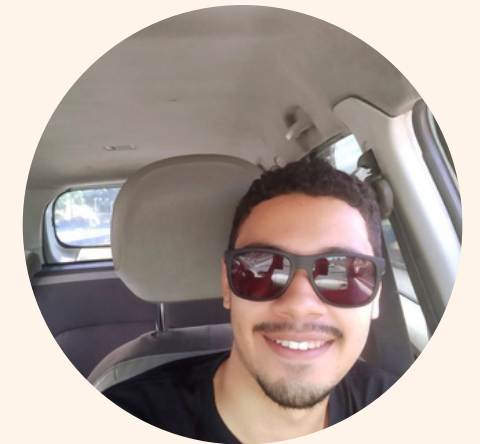
Luchecy Ribeiro
Engenheira
de Teste



Quézia Menezes
Analista de QA



Samuel Araujo
Engenheiro
de Teste



Yann Gabriel
Designer
de Teste

Você tem alguma pergunta?

Esperamos que você tenha aprendido algo novo.

