

Plano de Testes

CESLA

Automação de testes

1 Introdução

Este plano de teste foi criado para apresentar as abordagens de teste aos membros da equipe. Isso inclui os objetivos, escopo, agenda, riscos e abordagem por si só. Este documento irá identificar claramente quais serão os testes entregáveis, tanto o que compõe, quanto o que não compõe o escopo.

1.1 Objectivos

O projeto de automação de testes em questão neste documento refere-se à cobertura de testes dos endpoints da API relativa ao projeto <https://github.com/zanelladev/challenge-developer-flutter>, que deverá ser desenvolvida em Cypress, utilizando Mocha para definição dos testes e Chai para realizar as asserções, e Javascript como linguagem de programação. A execução dos testes deverá garantir que as necessidades do cliente e a qualidade foram atendidas, dada a cobertura positiva e negativa dos casos de teste. A equipe de teste é composta por um QA que desenvolverá e executará a automação, e o cliente que avaliará a sua estrutura e execução.

O projeto é de fase única.

1.2 Membros da equipe

Nome do recurso	Posição
Alex Sarubbi Heleno da Silva	Analista de Qualidade

2 Escopo

Esta única fase irá incluir os requisitos necessários para a cobertura de testes de API, bem como o desenvolvimento da automação e entrega. Para isso, o analista de qualidade deverá:

1. Criar um plano de testes
2. Criar casos de teste em BDD utilizando os critérios de aceitação e documentação, assim como o protótipo do app.
3. Criar uma automação de API com as APIs disponíveis no projeto challenge-developer-flutter utilizando Cypress
4. Elaborar uma estratégia para prevenção de bugs.

Testes de performance e capacidade não serão considerados neste escopo.

3 Suposições / Riscos

3.1 Suposições

1. Casos de teste negativos para o método POST de estudantes: API está considerando informações vazias para parâmetros obrigatórios e aceitando o envio de payload com parâmetros obrigatórios faltando. Para o último cenário, uma informação randômica é gerada e inserida no lugar do parâmetro faltante.

3.2 Riscos

Os seguintes riscos e ações apropriadas foram identificados para mitigar seu impacto no projeto. O impacto (ou severidade) do risco é baseado em como o projeto seria afetado se o risco for acionado. O gatilho é um evento causado se o risco for acionado, tornando-se um problema a ser lido.

#	Risco	Impacto	Gatilho	Plano de Mitigação
1	API com mock - POST de estudantes	Médio	Impossibilidade de executar os casos de teste negativos para este método	Implementação de obrigatoriedade dos campos exigidos pelo protótipo também na API
2	Falta de obrigatoriedade de autenticação na API	Alto	Perda de registros já inseridos para testes anteriormente por pessoas não autorizadas a testar	Implementação de obrigatoriedade de autenticação na API

4 Abordagem de Teste

4.1 Automação de Testes

Os casos de testes entregues deverão ser automatizados para a devida cobertura dos cenários de API em questão: GET/POST/PUT/DELETE. A automação deverá ser implementada em Cypress com Mocha e Chai, e entregue via GIT. A arquitetura deverá ser dividida por features e métodos, casos de teste negativos e positivos.

5 Ambiente de teste

A automação dos testes deverá ser executada de maneira local, executando todo o procedimento de instalação necessário para que seja possível rodar os testes.

5.1 Test Automation

5.1.1 Requisitos

Será necessário ter instalado o Node.js na máquina de execução dos testes

5.1.2 Execução dos testes

Para a correta execução dos testes, será necessário rodar os seguintes comandos no diretório raiz do projeto:

- **Instalação do Cypress:** npm install cypress [--force] (utilize '--force' somente no caso em que haja uma instalação prévia do Cypress em outro projeto)
- **Execução dos testes:**
 - npx cypress open (executa os testes com o browser)
 - npx cypress run (modo headless, executa os testes somente no console)

6 Entregas

6.1 Agenda de Testes

Atividade	Início	Fim	Esforço	Comentários
Test Planning	08/08/2024	09/08/2024	2 d	
Revisão dos requisitos do projeto	08/08/2024	09/08/2024	2 d	
Criação das estimativas de teste	08/08/2024	08/08/2024	1 d	
Escrita dos casos de teste	08/08/2024	08/08/2024	1 d	
Desenvolvimento do projeto de automação de testes	08/08/2024	09/08/2024	2 d	
Elaboração da estratégia para prevenção de bugs	09/08/2024	09/08/2024	1 d	

6.2 Entregáveis

Entregável	Para	Data
Plano de Teste	CESLA	09/08/2024
Casos de Teste	CESLA	09/08/2024
Estratégia para prevenção de bugs	CESLA	09/08/2024