Fábrica de Software - QA   
Phillipe Dautro - Desafio parte 2   
  
Plano de Teste

Nome do sistema: Alurapic   
  
Introdução:

Identificar possíveis bugs que podem ser apresentados pelos usuários na utilização do site Alurapic. O problema apresentado é na entrada de dados no formulário para registrar.

Relatório de bugs:

Nos casos observados, é possível adicionar informações que não apresentam legitimidade. No primeiro campo que é o do email, é possível adicionar qualquer caractere desde que tenha um ‘@’ e outro caractere a mais. No campo seguinte do full name, também é possível acrescentar qualquer caractere, desde que seja no mínimo dois. Desse modo, possibilitaria a entrada de duplo traços “—“, vírgulas”,,”, pontos “..”, arrobas “@@” e apenas números. No terceiro campo, que é o de username, é admissível que autentique com apenas dois traços “—“, demonstrando assim uma dificuldade na identificação do usuário. Os testes foram realizados em um PC com sistema Windows 10 e naveagador Google Chrome, como também em um iphone 11 com o IOS versão 16.1.

Tipos de testes

Em relação aos tipos de testes, os testes utilizados foram os *testes manuais* e automatizados. No aspecto manual, um desenvolvedor testou os campos da área de registro dos campos do site, assim sendo, apresentando um caráter mais exploratório. Os *testes automatizados* aconteceram através de um QA. As ferramentas utilizadas para os testes foram o Visual Studio Code, node e Cypress.   
   
 Ainda, no que tange os tipos de testes utilizados, o de *validação* foi recorrente, visto que era possível preencher os campos de forma incorreta e ainda assim passaria pelo site. Outro teste válido, foi o de *usabilidade*, já que a falta de organização de dados que podem ser adicionados geraria uma maior dificuldade de compreensão no banco de dados.

Métodos de classes:

|  |  |  |
| --- | --- | --- |
| Objetivo | Identificar bugs apresentados no preenchimento de registro no site Alurapic | |
| Técnica | (x) manual | (x) automática |
| Estágio de Teste | ( ) Integração (x) Sistema | ( ) Unidade (x) Aceitação |
| Abordagem de teste | Caixa branca ( ) | Caixa preta (x) |

Severidade do Bug:

A severidade do bug apresentada pode ser caracterizada como *moderada*. Mesmo com a abundância de caracteres que podem ser adicionados nos campos, a sua funcionalidade não é afetada. Outro aspecto dessa severidade, é a ausência da mensagem de erro quando os dados são inseridos de maneira que no futuro seja de difícil de compreensão.

Códigos de testes:

1) O primeiro bug que pode ser verificado é no campo do email, sendo não necessário adicionar um email existente ou correto para que não passe pelo sistema. Com a utilização do ‘ferrari@a’, isto é, um ‘@’ e mais um caractere, o site aceita esse tipo de informação sem nenhum problema.

it.only('validaçao do email', () => {

        cy.contains('a', `Register now`).click();

        cy.contains(`button`, `Register`).click();

        cy.contains(`ap-vmessage`, `Email is required!`).should(`be.visible`);

        cy.get(`input[formcontrolname="email"]`).type(`ferrari@a`);

}

)

2) O segundo bug é caracterizado pela aceitação mínima de dois caracteres, não importando quais sejam. Atualmente digitando números, caracteres, traços e vírgulas, sucederiam por uma tranquila passagem pelo sistema. Desse modo, se faz necessário um maior controle para as informações colocadas no nome, uma das correções seria a implementação de apenas leitura de letras.

it.only('valida full name', () => {

        cy.contains('a', `Register now`).click();

        cy.contains(`button`, `Register`).click();

        cy.get(`input[formcontrolname="fullName"]`).type(`--`);

    }

    )

3) Outro bug encontrado é que precisa um maior controle de letras e números na formação de um username, até para melhor identificação, pois atualmente o sistema aceita dois traços como username “—“.

it.only('valida username', () => {

        cy.contains('a', `Register now`).click();

        cy.contains(`button`, `Register`).click();

        cy.get(`input[formcontrolname="userName"]`).type(`--`);

cy.contains(`ap-vmessage`, `User available`).should(`be.visible`);