

## Desafio de Automação como parte do processo seletivo para QA Test da Bionexo

Ferramentas utilizadas:

- Visual Studio Code – Ambiente de desenvolvimento
- Cmder – Prompt de comando Linux, no Windows

Linguagem de programação utilizada.

- ruby 2.5.1p57

Framworks utilizados

- Cucumber – BDD
- Capybara – **O Capybara** é um software de automação de testes baseado na web que simula cenários para histórias de usuários e automatiza testes de aplicativos da web para desenvolvimento de software orientado a comportamento. Está escrito na linguagem de programação Ruby.
- Rspec – DLS Ruby, para testar código Ruby.
- SitePrism - DSL (Domain Specific Language) criada para facilitar a criação de page objects para testes automatizados em Ruby, utilizando o Capybara.

Links úteis

<https://www.guru99.com/alert-popup-handling-selenium.html>

<https://makandracards.com/makandra/1549-accept-or-deny-javascript-confirmation-dialogs-in-capybara-selenium>

<https://imasters.com.br/agile/3-dicas-para-documentacao-de-teste-em-uma-equipe-agil>

Documentação utilizada:

BDD e2e – Desenvolvimento Orientado ao Comportamento e2e (end two end)

Para melhor compreensão dos envolvidos no projeto de desenvolvimento de um software, faz-se necessário o uso de uma documentação de fácil manutenção, e eficiente que possa ser escrita, lida e de fácil entendimento. Quando a **user stories** é escrita e aprovada pelo cliente, a mesma servirá de auxílio aos **desenvolvedores** no desenvolvimento da feature, porém para a equipe de desenvolvimento ela não anula os requisitos, mas servirá como um agregador. (a não ser que a equipe prefira não utilizar requisitos). Os **gerentes, team líder e cliente**, servirá como documentação funcional do comportamento de uma determinada funcionalidade. E ao **tester engineer**, servirá como guia para o desenvolvimento dos testes automáticos.

## Estrutura do Projeto

**[specs]** - Os cenários ou casos de testes criados foram especificados através do Cucumber e podem ser visualizados em: **bioQA\tests\features\specs**

**[pages]** – O padrão de projeto Page Object, foi utilizado afim de mapear cada tela em separado. Os métodos podem ser encontrados em: **bioQA\tests\features\pages**. A inicialização da instância dos métodos pode ser encontrada no arquivo **page\_helper.rb** em: C:\Projects\automation\challenges\bioQA\tests\features\support. A page é responsável principal por identificar os elementos da interface.

**[step\_definitions]** – Os passos de cada passo especificado na **specs**, são definidos aqui no **step\_definitions**, que interage com os métodos (**pages**)

**[support/data]** – Caso tenha outro ambiente para testes (homologação, produção, teste) – podem ser definidos através de um **arquivo.yml**

**[support/helper.rb]** – Método de auxilio, responsável por tirar screenshots e embutir no relatório html.

**[support/hooks.rb]** – Método de auxilio, responsável por tirar um screenshots seja com resultado positivo ou negativo, para cada cenário do specs.

**[support/env.rb]** – Principal arquivo de configuração. Destaque -> configurado nele o conceito de teste independente, onde o browser seja ele qual for precisa que sua sessão seja fechada e aberta para cada cenário realizado, tornando assim os testes independentes e mais confiáveis.

**[results]** – Pasta dos resultados – relatório.html e pasta com os resultados que passaram e falharam, identificando pela data que o teste foi realizado.

### Resultados da Automação

USER STORY	CRITÉRIO DE ACEITE	RESULTADO
<b>UserStory1</b> bundle exec cucumber -t @cadastrar	Cadastro de Medicamento	PASS
	Cadastro de Material Médico	PASS
	Todos os campos são obrigatórios.	FAIL
	Todos os campos são obrigatórios (Campo Fabricante)	FAIL
	O campo tipo deve aceitar apenas Medicamentos e Materiais Médicos.	PASS
<b>UserStory2</b> bundle exec cucumber -t @listar	A listagem deve ser ordenada pelo campo nome.	FAIL
	A listagem deve ser paginada pelo total de 10 resultados.	PASS
	As informações da listagem deve ser: Nome, Tipo, Fabricante e Status.	PASS
<b>UserStory3</b> bundle exec cucumber -t @editar	Devo acessar o produto pela listagem	PASS
	Todos os campos devem poder ser editados	PASS
	Após editar deve aparecer uma mensagem de sucesso	PASS
	Deve conter um link de volta a listagem	PASS
<b>Executar Todos</b>	bundle exec cucumber	

#Bug UserStory1 – reportado – Nem todos os campos estão como obrigatórios, contradizendo assim o critério de aceite. Para mais informações consulte o link abaixo: <http://qainterview.qa.cloud.bionexo.com.br/bugs/236/edit>. Quando o time de desenvolvimento corrigir o bug e torna-lo obrigatório ou o time de aceite (PO ou cliente) podem modificar o critério de aceite e aceitar o campo “descrição” sem obrigatoriedade. Porém atualmente é uma falha.

#Bug UserStory1 – reportado. Campo Fabricante, apresenta falha na mensagem de obrigatoriedade. Para mais informações, consulte o link abaixo: <http://qainterview.qa.cloud.bionexo.com.br/bugs/238/edit>

#Bug UserStory2 – reportado. Para mais informações consulte o link abaixo: <http://qainterview.qa.cloud.bionexo.com.br/bugs/239/edit>. Quando o time de desenvolvimento corrigir o bug e ordena-lo por ordem alfabética, ao executar a automação novamente o resultado será PASS.

Alguns passos para configurar o projeto.

1. Pré-Condições:
  - a. Variáveis de ambiente Java configuradas corretamente
  - b. Efetue download da versão mais recente do Ruby com Devkit.  
<https://github.com/oneclick/rubyinstaller2/releases/download/rubyinstaller-2.4.5-1/rubyinstaller-devkit-2.4.5-1-x64.exe>
  - c. Coloque na pasta Windows o arquivo gecko e chromedriver correspondente a versão do seu navegador  
<https://github.com/mozilla/geckodriver/releases>  
<http://chromedriver.chromium.org/downloads>
2. Descompacte o arquivo **bioQA.rar** em seu diretório de desenvolvimento.  
Ou efetue o clone do projeto no github  
**[git@github.com:IsaiaSilva/bioQA.git](https://github.com/IsaiaSilva/bioQA.git)**
3. Abra no cmd ou outro prompt de comando no diretório **tests** e execute o projeto de automação.  
bundle install  
Bundle exec cucumber

Atenção

No diretório **...tests\features\support\data\homolog.yml**

**Altere a URL para o caminho onde o arquivo foi extraído em sua máquina local.**

Referências:

Isaias Santos Silva – [isaiasilva.info@gmail.com](mailto:isaiasilva.info@gmail.com)

(71) 99171 - 5923