**Walmart - Projeto de Automação de Testes com Cucumber**

|  |
| --- |
| O objetivo desse desafio é avaliar o conhecimento dos candidatos e sua capacidade de realiza-lo dentro de um tempo pré-determinado. Não existe resolução certa ou errada, avaliaremos de acordo com o nível de experiência que for exigido pelas vagas disponíveis no momento.  Envie seu desafio mesmo que você não conclua todas as questões, avaliaremos tudo o que for entregue.  Atualmente, utilizamos algumas ferramentas Open Source como padrão em todos os projetos de automação de testes. Queremos saber o quanto você conhece das ferramentas e/ou o quanto você é capaz de aprender sobre elas.  O tempo **máximo** para envio do link do projeto é de uma semana corrida (claro, envie antes caso termine antes =]). |

**Instruções gerais**

|  |
| --- |
| * Crie um repositório no GitHub ou no BitBucket para compartilhar o projeto; * Crie um README.md para o projeto, explicando como preparar o ambiente e como executar os testes de cada um dos exercícios; * Use as técnicas e ferramentas que você souber ou aprender durante os estudos, e que considere que sejam adequadas aos exercícios, mesmo que não tenha sido pedido explicitamente. Se achar válido destacar, comente no README.md sobre elas e sobre sua motivação em usá-las. Tudo será avaliado; * A linguagem para realizar os desafios é Ruby. |

1. **Automação Frontend**

|  |
| --- |
| Nessa questão, utilize Cucumber e Capybara.  Na página <https://the-internet.herokuapp.com/login>, os usuários fazem login com sucesso preenchendo os campos **Username** e **Password**.  Execute os passos a seguir:   * Preencher o campo Username com o valor “tomsmith”; * Preencher o campo Password com o valor “SuperSecretPassword!”; * Clicar no botão Login; * Validar a mensagem “**You logged into a secure área!**”na tela; * Clicar no botão Logout; * Validar a mensagem “**You logged out of the secure área!**’ na tela.   **Username e Password devem ser passados via linha de comando, NÃO deixa-los *hardcoded* no projeto.**  Crie ao menos um cenário que represente esse comportamento do usuário e faça a automação do mesmo. |

1. **Automação Backend**

|  |
| --- |
| Nessa questão, utilize Cucumber e HTTParty.  Os correios possuem uma API pública que retorna os detalhes de um endereço dado um determinado CEP. Essa API é usada por diversos sistemas para preenchimento de endereço, evitando que o usuário precise digitar todos os seus dados, pois os sistemas auto completam os dados baseado apenas no CEP fornecido.  Os dados da API são:   * **URL**: <http://correiosapi.apphb.com/cep/$CEP> (substitua $CEP pelo CEP que desejar validar) * **Método**: GET * **Exemplo**:   **Request**: GET -  http://correiosapi.apphb.com/cep/13040089  **Response**:  {  "cep": "13040089",  "tipoDeLogradouro": "Rua",  "logradouro": "Manoel Sylvestre de Freitas Filho",  "bairro": "Jardim Nova Europa",  "cidade": "Campinas",  "estado": "SP"  }  Implemente cenários que representem o uso dessa API e faça a automação dos mesmos, contemplando ao menos:   * O uso da API com um CEP válido; * O uso da API com CEP inválido.   **É importante validar a API através de cada campo do JSON de resposta e de seu response code.** |