



Challange-QA

O objetivo deste desafio é avaliar as competências técnicas dos candidatos a Analista de Testes na Maior Plataforma de Educação do Brasil.

Atividade 1

1.1. Nesta primeira atividade deve ser criado um Projeto .Net para a documentação dos Cenários de testes e para a automação e2e.

Especificações Técnicas

- Projeto em .Net8.0
 - Specflow
 - Specflow Livingdoc
 - Selenium
 - XUnit
 - Fluent Assert

1.2. Explore os fluxos presentes na aplicação Subscriptions e identifique regras e requisitos.

- Acesse a aplicação [aqui](#).
- Caso encontre algo que considere bug, reporte em um arquivo chamado **BUGS.md**.

1.3. Utilizando todos os elementos do Gherkin descreva em cenários as regras e requisitos identificados no item 1.2. Os cenários devem ser versionados no projeto .Net criado no item 1.1, seguindo os padrões do Specflow.

1.4. Utilizando os cenários descritos no item 1.3 implemente a automação dos mesmos, utilizando os recursos do Specflow e Selenium.

1.5. Gere o relatório html da documentação e resultados da execução de testes com o LivingDoc e versione junto ao projeto.

1.6. Considerando que a aplicação aqui utilizada vai passar a persistir dados em banco de Dados PostgreSQL, descreva em um arquivo chamado **DATA.md** a estratégia que você utilizaria em relação a massa de dados de testes

Atividade 2

2.1. Criar ao menos 3 cenários de teste de performance simulando cargas de 100, 500 e 1000 usuários simultâneos.

Teste, no mínimo, os seguintes endpoints:

```
- /flip_coin.php  
- /my_messages.php
```

2.2. Criar um projeto de teste de carga usando k6 para avaliar as APIs disponíveis [aqui](#).

2.2.1. Crie scripts de teste com k6 para os cenários criados no item 2.1, sendo, no mínimo os testes para os seguintes endpoints:

```
- /flip_coin.php  
- /my_messages.php
```

2.2.2. Gere os resultados em html

2.2.3. Envie os scripts de teste e um relatório com os resultados, incluindo gráficos e análise de desempenho.

Utilize o mesmo repositório Git do projeto criado no item 1 para a entrega.

Critérios de avaliação

- Cenários identificados
- Qualidade da escrita e uso adequado dos elementos da sintaxe Gherkin
- Padrões de projeto e arquitetura aplicados
- Organização do projeto
- Tratamento de erros nos cenários
- Utilização do Git (quantidade e descrição dos commits, Git Flow, ...)
- Validações realizadas

Desejável

- Aplicação do BRIEF
- Aplicação de boas práticas de escrita de cenários
- Uso de todos os elementos do Gherkin
- Uso de recursos do Specflow
- Realização da atividade 1 completa
- Realização do item 2.1 da atividade 2

Diferencial

- Componentizar os elementos das páginas com Selenium e utilizar estes componentes na automação

- Execução dos testes em paralelo utilizando 3 diferentes navegadores
- Utilizar Selenium grid com docker compose
- Entregar além do item 1 todas as atividades propostas para a atividade 2

Instruções de entrega

1. Crie um fork do repositório no seu GitHub
2. Faça o push do código desenvolvido no seu Github
3. Inclua um arquivo chamado **COMMENTS.md** explicando:
 - Decisões da arquitetura utilizada
 - Lista de bibliotecas de terceiros utilizadas
 - O que você melhoraria se tivesse mais tempo
 - Quais requisitos obrigatórios que não foram entregues
4. Informe ao recrutador quando concluir o desafio junto com o link do repositório
5. Após revisão do projeto junto com a equipe de técnica deixe seu repositório privado