

RESPOSTAS DAS QUESTÕES

Questão 01:

- História de Usuário:
 - **Como** um cliente da Amazon
 - **Quero** pesquisar pelo livro *“AI Engineering: Building Applications with Foundation Models”*
 - **Para** adicioná-lo ao carrinho e realizar a compra
- Critérios de Aceite:
 - A página da Amazon deve ser acessada com sucesso
 - O campo de busca deve aceitar o texto "AI Engineering: Building Applications with Foundation Models".
 - O resultado exibido deve conter o livro correto, com as seguintes opções:
 - Idioma: Inglês
 - O autor ser Chip Huyen
 - Ser um livro físico
 - Ser um livro novo
 - Ao clicar em “Adicionar ao carrinho”, deverá exibir a mensagem “Adicionado ao carrinho”
- Casos de Testes:

ID	Cenário	Passos	Resultado Esperado
CT01	Buscar o livro corretamente	1. Acessar https://www.amazon.com.br/ 2. Inserir o texto na busca 3. Clicar na lupa de “Pesquisar”	A página de resultados mostra o livro correto
CT02	Validar informações do livro	1. Clicar no livro “AI Engineering: Building Applications with Foundation Models” 2. Verificar idioma, autor, tipo e condição	Todos os dados estão corretos
CT03	Adicionar ao carrinho	1. Clicar em “Adicionar ao carrinho” 2. Validar mensagem exibida	Mensagem “Adicionado ao carrinho” é exibida

Questão 03:

Os testes seriam executados automaticamente a cada push ou pull request feito no repositório principal, através de um workflow configurado num Jenkins, GitLab CI/CD ou como é o caso da questão, no GitHub Actions. Esse workflow seria responsável por criar um ambiente, instalar as dependências do projeto, configurar o Playwright e o Python, e executar os testes com o pytest, com isso, qualquer modificação no código dispara a execução dos testes, realizando a integração contínua e detectando as falhas.

Poderia configurar o pipeline para interromper a execução se houvesse alguma falha, utilizando uma saída do pytest. Essas falhas apareceriam automaticamente no GitHub, marcando o commit ou o pull request como **"failed"** e armazenar o arquivo em um .xml para ser analisado.

Para executar os testes de forma mais rápida utilizando o paralelismo poderia ser usado o **matrix strategy** do GitHub Actions, separando por tipo de teste (exemplo: interface e API, como é o caso desse teste prático) ou até por navegador (Chromium, Firefox), desse jeito é permitido executar esses tipos de testes de forma paralela.

Questão 04:

1 - Quais aspectos você avaliaria ao testar uma aplicação similar ao ChatGPT (<https://chatgpt.com/>)?

Acredito que por ser um sistema com interface de usuário, os primeiros aspectos a serem testados seriam os pontos mais superficiais para a experiência do usuário como o cadastro da conta, login, os campos das perguntas, visuais dos textos das respostas, limpeza dos campos de textos, tempo de resposta das perguntas, persistência dos dados como por exemplo guardando o histórico das perguntas e respostas.

Passando desse ponto, iria para a questão da avaliação das respostas em si da IA, conferindo alguns pontos como:

- Coerência da IA, conferindo se mantém um padrão nas respostas;
- Confiabilidade e veracidade das respostas, avaliando se a IA está trazendo respostas verdadeiras e confiáveis para o usuário;
- Questões de segurança como a inserção de arquivos e textos maliciosos a serem lidos;
- Questão de memória e contexto, verificando se a IA guarda as respostas já dadas para o usuário, se consegue ser introduzido ao contexto pedido levando em consideração as respostas anteriores;
- Prompt de comando, se a IA responde de forma adequada em diferentes tipos de linguagens e formas de uso.

Outros pontos a serem testados também são as estatísticas das respostas, que poderiam ser utilizadas métricas para essa verificação como BLEU, ROUGE, entre outros, além de verificar mudanças de respostas em diferentes versões como uma avaliação A/B.

Explique se você já utilizou alguma ferramenta de IA para testes automatizados. Caso tenha experiência, responda:

1 - Qual ferramenta ou técnica de IA foi utilizada? (Ex: TestRigor, Mabl, Appliflow com IA visual, ChatGPT para geração de testes).

Para auxiliar na automação, eu utilizei somente o ChatGPT. Já cheguei a utilizar um pouco o TestRigor, mas só para fins de estudo mesmo, não foi para nenhum projeto ou experiência profissional.

2 - Para qual propósito a IA foi aplicada? (Ex: geração automática de casos de teste, detecção de falhas visuais, automação de scripts).

Utilizei o ChatGPT para criar a base do código e para a automação e para auxiliar principalmente em dúvidas obtidas no framework ou na linguagem utilizada. Utilizo bastante para me auxiliar na criação da documentação, principalmente casos e cenários de testes, onde coloco o script falando sobre a funcionalidade e ele gera a documentação, depois faço a conferência e faço a alteração de algum cenário ou caso de testes que gerou errado ou adiciono os que eu acredito que ficaram faltando.

3 - Como a IA impactou o processo de QA e os resultados obtidos?

Impactou principalmente no ganho de tempo e de performance, pois o maior tempo gasto é na estruturação e na configuração base inicial dos arquivos e dos testes, tanto na

automação em si, como na documentação, além da questão do aprendizado que consegue gerar com comentários para auxiliar na leitura e entendimento do código da automação.