1. Tipos de Testes:

**a) Teste Unitário:** são testes que os desenvolvedores fazem para verificar se partes pequenas do código, como funções ou classes, estão funcionando direito? Isso é o teste unitário. Eles são importantes para garantir que as partes individuais do software estejam corretas.

**b) Teste de Integração:** Quando queremos ter certeza de que diferentes partes do software funcionam bem juntas, usamos o teste de integração. Ele olha para a forma como essas partes se comunicam e se encaixam. É importante para ter um sistema estável, como exemplo uma tela que de perfil, geralmente há dados do usuário nesta tela, e precisa estar funcionando bem com o banco de dados CRUD, e também os dados dessa tela precisam aparecer nas páginas que exibem esses dados, como a saudação no header da página com o nome e sobrenome por exemplo.

**c) Teste de UI (User Interface):** Esse teste se concentra na aparência e na interação do usuário com o software. Basicamente, verificamos se os botões, janelas e formulários estão funcionando corretamente. É crucial para garantir uma boa experiência do usuário.

2. Importância da Qualidade de Software em um E-commerce:

A qualidade do software é uma questão gigante em praticamente tudo, mas pensando no e-commerce, se o software não estiver em boa forma, corremos sérios riscos:

Podemos perder clientes, porque ninguém gosta de lidar com um site que são lentos, trava ou tem erros.

* A segurança é crucial, pois qualquer falha pode expor informações sensíveis dos clientes.
* A concorrência está sempre à espreita, e um site com problemas pode fazer com que os clientes vão para o concorrente.
* Corrigir problemas depois do lançamento custa caro e dá muito trabalho.

3. Método "Passo a Passo" para Gherkin:

**GHERKIN**

Funcionalidade: Busca por produto

Cenário: Buscar um produto no site

Dado que eu estou na página inicial do site

Quando eu digito "Processador i9" na barra de busca e clico em pesquisar

Então eu devo ver uma listagem de produtos

4. TDD (Test Driven Development) vs. BDD (Behavior Driven Development):

**TDD** é quando os desenvolvedores escrevem testes antes de escrever o código real. Eles se concentram em aspectos técnicos.

**BDD** se concentra no comportamento do software do ponto de vista do usuário. Os testes são escritos em linguagem natural para descrever o que o software deveria fazer.

5. Page Objects:

Page Objects são uma maneira de organizar o código de automação de testes. Em vez de misturar código de automação com a interface do usuário, você cria classes separadas para representar cada página ou componente. Isso torna os testes mais fáceis de entender e manter, especialmente quando a interface do usuário muda.

6. Automação de Testes com Robot Framework e Selenium:

Desenvolvido e publicado no github, link: https://github.com/matteusbonotto/kabum-challenge