**Cenários de teste**   
Funcionalidade: Compra de Imóveis

Como um cliente interessado em comprar um imóvel, eu quero poder pesquisar, visualizar e solicitar informações sobre imóveis disponíveis, para que eu possa encontrar e comprar o imóvel ideal para mim.

CT-P: Pesquisar imóveis

Dado que estou na página inicial do site de busca de imóveis

Quando eu insiro os critérios de pesquisa, como localização, preço máximo, tipo de imóvel, etc.

E eu clico no botão de pesquisa

Então deve ser exibido uma lista de imóveis que correspondam aos meus critérios de pesquisa

CT-P: Visualizar detalhes do imóvel

Dado que estou visualizando uma lista de imóveis

Quando eu clico em um imóvel específico

Então deve ser exibido os detalhes completos do imóvel, incluindo preço, localização, características, fotos, etc.

CT-P: Solicitar informações sobre o imóvel

Dado que estou visualizando os detalhes de um imóvel,

Quando eu preencho o formulário de solicitação de informações, incluindo meu nome, e-mail e telefone,

E eu clico no botão de enviar solicitação,

Então sistema deve enviar uma confirmação de que minha solicitação foi enviada com sucesso.

CT-P: Adicionar imóvel ao carrinho

Dado que estou visualizando os detalhes de um imóvel,

Quando eu clico no botão "Adicionar ao Carrinho",

Então o imóvel deve ser adicionado ao meu carrinho de compras.

CT-P: Finalizar compra com sucesso

Dado que estou no carrinho de compras com pelo menos um imóvel adicionado,

Quando eu clico no botão "Finalizar Compra",

Então eu devo ser redirecionado para o processo de checkout,

E após completar as etapas de pagamento com sucesso,

Então sistema deve exibir uma confirmação de que minha compra foi realizada com sucesso.

CT-N: Compra com estoque esgotado

Dado que estou no processo de checkout,

Quando eu tento finalizar a compra de um imóvel que está com o estoque esgotado,

Então o sistema deve exibir uma mensagem de erro indicando que o imóvel não está disponível para compra.

CT-N: Solicitar informações com campos obrigatórios em branco

Dado que estou visualizando os detalhes de um imóvel,

Quando eu tento enviar uma solicitação de informações com campos obrigatórios em branco,

Então o sistema deve exibir mensagens de erro indicando quais campos precisam ser preenchidos.

CT-N: Acessar página restrita sem estar logado

Dado que estou tentando acessar uma página restrita, como o histórico de compras,

Quando eu tento acessar sem ter feito login,

Então o sistema deve redirecionar-me para a página de login.

CT-P: Integração com sistemas externos

Dado que o sistema está integrado com sistemas externos de bancos e imobiliárias,

Quando um cliente solicita informações sobre um imóvel,

Então o sistema deve enviar as informações relevantes para o sistema externo correspondente

E registrar a interação no sistema interno para rastreamento e análise.

CT-P: Testar a escalabilidade do sistema

Dado que o sistema está operando em todo o território brasileiro,

Quando o sistema experimenta um pico de tráfego, simulando um alto volume de operações,

Então o sistema deve lidar eficientemente com o aumento de carga sem falhas ou degradação significativa no desempenho.

**Estratégia de testes**

**Envolvimento Precoce dos Testadores:**

Introduziria testadores na fase inicial do desenvolvimento, permitindo que eles revisem os requisitos, as especificações e os designs antes da implementação. Isso ajuda a identificar potenciais problemas desde o início e a alinhar as expectativas em relação aos testes.

**Revisão de Código Colaborativa:**

Encorajaria revisões de código colaborativas, onde os testadores podem fornecer feedback sobre a adequação dos testes unitários e a qualidade geral do código antes mesmo de serem integrados ao repositório principal.

**Desenvolvimento Orientado a Testes (TDD):**

Incentivaria os desenvolvedores a seguir a abordagem do Desenvolvimento Orientado a Testes, escrevendo testes unitários antes de implementar o código. Isso garante que o código seja testável e ajuda a identificar problemas de design precocemente.

**Automação de Testes Unitários:**

Integraria a automação de testes unitários ao processo de integração contínua (CI), permitindo que os testes sejam executados automaticamente sempre que o código é alterado. Isso ajuda a identificar rapidamente problemas de regressão.

**Testes Exploratórios Antecipados:**

Iniciaria testes exploratórios antecipados, onde os testadores podem explorar o sistema em desenvolvimento para identificar problemas de usabilidade, design e funcionalidade.

**Integração Contínua e Entrega Contínua (CI/CD):**

Implementaria um pipeline de CI/CD para automatizar o processo de integração, teste e entrega do software. Isso permite que as alterações sejam testadas e validadas automaticamente em um ambiente controlado antes de serem implantadas em produção.

**Abordagens de teste**

**Testes de Unidade:**

Para garantir que cada componente do sistema funcione corretamente individualmente. Isso é essencial especialmente para microsserviços, onde cada serviço deve ser testado isoladamente.

**Testes de Integração:**

Para verificar a interação entre os diferentes componentes do sistema, garantindo que eles se comuniquem corretamente e que as integrações com sistemas externos funcionem conforme o esperado.

**Testes de Regressão:**

Para garantir que as alterações recentes no código não introduziram regressões, ou seja, que as funcionalidades existentes continuem funcionando conforme o esperado após as modificações.

**Testes Funcionais:**

Para validar se o sistema atende aos requisitos de negócios e às expectativas do cliente. Isso inclui testes de caixa-preta, onde os testes são realizados com base nas especificações, sem conhecimento do código interno.

**Testes de Aceitação:**

Para garantir que o sistema atenda às necessidades e expectativas do usuário final.

**Testes de Desempenho e Carga:**

Para avaliar o desempenho do sistema sob diferentes cargas de usuários e condições de uso, garantindo que o sistema possa lidar com o volume esperado de operações sem degradação significativa no desempenho.

**Testes de Segurança:**

Para identificar e mitigar possíveis vulnerabilidades de segurança no sistema, protegendo os dados dos usuários e garantindo a conformidade com regulamentações de segurança.

**Testes de Usabilidade:**

Para avaliar a facilidade de uso e a experiência do usuário do sistema, garantindo que seja intuitivo e eficiente para os usuários finais.

**Testes de Resiliência:**

Para verificar como o sistema responde a falhas e interrupções, garantindo que ele possa se recuperar adequadamente e manter a integridade dos dados.

**Testes de Regressão Automatizados:**

Para automatizar a execução de testes de regressão, permitindo uma rápida validação de alterações no código e garantindo a estabilidade do sistema ao longo do tempo.

**Tipos/níveis de teste**

**Testes Unitários (Componente):**

Seriam realizados pelos próprios desenvolvedores com auxílio dos QA’s para testar unidades individuais de código, como funções, métodos e classes. A cobertura de código seria monitorada para garantir que uma porcentagem significativa do código seja testada.

**Testes de Integração (Componente):**

Seriam realizados para verificar a interação entre os diferentes componentes do sistema, incluindo microsserviços, APIs e módulos, utilizando massa de dados de teste realistas para simular o comportamento do sistema em cenários reais de uso, e garantiria que as integrações com sistemas externos, como bancos de dados, serviços de terceiros e sistemas de pagamento, funcionem conforme o esperado.

**Testes Funcionais (Sistema):**

Seriam realizados para verificar se o sistema atende aos requisitos funcionais e comportamentais especificados. Utilizando cenários de teste abrangentes, incluindo casos de teste positivos e negativos, para garantir que todas as funcionalidades do sistema se comportem conforme esperado. A automação de testes funcionais seria implementada sempre que possível para garantir uma alta cobertura.

**Testes de Aceitação do Usuário (UAT) (Nível de Sistema):**

Seriam realizados em colaboração com o cliente do time de desenvolvimento(PO). Incluíria testes exploratórios conduzidos pelo PO para identificar problemas de usabilidade, fluxos de trabalho e experiência do usuário.

**Testes de Desempenho e Carga (Sistema):**

Seriam realizados para avaliar o desempenho, escalabilidade e capacidade de resposta do sistema sob diferentes cargas de trabalho. Usaria ferramentas de teste de desempenho, como JMeter, para medir métricas de desempenho.

**Testes de Segurança (Sistema):**

Seriam realizados para identificar e mitigar possíveis vulnerabilidades de segurança no sistema. Incluíria testes de penetração, análise estática de código, verificação de certificados SSL e outras técnicas para garantir a segurança do sistema e dos dados dos usuários.

**Testes de Regressão Automatizados (Sistema):**

Seriam implementados para garantir que as alterações no código não introduzam regressões ou quebras em funcionalidades existentes. Sendo incrementada de acordo com a evolução do sistema

**Ferramentas propostas**

**Testes Unitários e de Integração:**

Pytest.

**Testes Funcionais e de Regressão:**

Robot Framework.

**Testes de Desempenho e Carga:**

JMeter, Locust.

**Testes de Segurança:**

OWASP ZAP, Burp Suite.

**Gerenciamento de Testes e Colaboração:**

Azure DevOps , Jira.

**Como você se imagina atuando nessa equipe?**

Considero que minha atuação nessa equipe seria primariamente no desevolvimento de testes automatizados, implementação de CI/CD para realização de testes continuos, realização de testes manuais, escrita e manutenção de planos de teste, tentaria aplicar práticas do “Shif-left testing” para tentar encontrar e mitigar custos o mais cedo possível, e tentaria levar um pouco dessa cultura de qualidade para todo o time. Auxiliando os desenvolvedores a escreverem testes unitários, me aproximando do PO para analisar e previnir erros na documentação de requisitos e escrita da US