

Dado esse cenário, como você iniciaria uma estratégia de teste?

Minha estratégia para início de um plano de testes seria encontros com o cliente e time de produto para levantamento de:

- Requisitos de negócio com o cliente.
- Mapear riscos de negócio.
- Definir a estratégia de testes de usabilidade.

Então, me reuniria com o time de desenvolvimento e arquitetura para mapear:

- Requisitos não funcionais (escalabilidade, segurança e desempenho).
- Definir a estratégia de testes baseada em desenvolvimento ágil.
- Mapear os riscos técnicos .
- Identificar os pontos de integração com os microserviços e outros sistemas externos.
- Ajudar a definir critérios de DoR e DoD (definition of ready e definition of done), para facilitar o fluxo ágil de desenvolvimento.
- Incluir QA nas cerimônias ágeis.
- Garantir que o pipeline CI/CD inclua testes automatizados.

Com todos esses alinhamentos documentados, iniciaria um documento de plano de testes onde iria detalhar a estratégia de testes (incremental, exploração, funcional, etc), inserir os casos de teste (foco no comportamento com BDD, mais voltado para o time de negócio) e cenários de testes (com foco na execução e verificação) onde este será voltado ao time de QA, documentado também na ferramenta de gestão e permitirá a **reprodutibilidade**. No segmento final do plano de testes estariam todos os links para as execuções dos mesmos testes e relatório dos bugs abertos.

Quais abordagens de teste você consideraria?

Eu adotaria uma abordagem multicamadas, onde os passos de desenvolvimento seriam acompanhados por testes indicados para cada fase. As abordagens seriam:

- **Testes funcionais**, com etapas de testes end-to-end e UI, para validar os fluxos dos diferentes atores. Fluxo do cliente comprando um imóvel, do corretor de imóveis enviando uma proposta, aprovação pela imobiliária e banco avaliando e aprovando o financiamento.
- **Testes de contrato**, após os refinamentos técnicos onde foi firmado o contrato das APIs, realizar testes de contrato para garantir a compatibilidade dos microserviços.

- **Testes de API** (REST/SOAP), com foco nas entradas e saídas e status HTTP.
- **Testes de integração** entre microserviços e com APIs externas.
- **Testes de performance e carga** para avaliar a escalabilidade e latência, uma vez que o sistema terá alto tráfego e será de uso nacional.
- **Testes de segurança** para verificação de autenticação/autorização, credenciais e proteção de dados sensíveis.
- **Testes de usabilidade (UX)** com atenção especial para as interfaces de clientes e corretores.
- **Testes exploratórios** para verificar eventuais cenários alternativos não cobertos pelos roteiros tradicionais.

Como você trabalharia com diferentes tipos/níveis de teste?

Inicialmente usaria a pirâmide de testes, a base são os testes unitários com alta cobertura usando o framework necessário (XUnit, Jest, JUnit) executados a cada build.A camada do meio da pirâmide são testes de integração e API, com scripts newman ou RestAssured, validados em ambiente de homologação. Então o topo com os testes end-to-end automatizados com Cypress ou Playwright para jornadas críticas. Nos níveis transversais de testes utilizaria os testes de performance e de segurança.

Quais ferramentas você consideraria usar?

Categoria do teste	Ferramenta
Automação	Cypress, Playwright
Testes de API	Postman, Newman
CI/CD	Github Actions, Azure devops pipeline
Performance	JMeter
Contrato	Postman, Newman
Monitoramento	DataDog, New Relic
Report de bugs	Jira, Azure Devops

Como você se imagina atuando nessa equipe?

Meu perfil ágil e colaborativo me permite participar ativamente desde a fase de discovery até a entrega:

- Apoio na escrita de critérios de aceite e cenários em Gherkin.
- Desenvolvimento de automação de testes desde o início.
- Acompanhamento das entregas por épico, validando o valor entregue por release.
- Validação de integrações e dados, especialmente os sensíveis (financiamentos, dados pessoais).
- Protagonismo em montar estratégias de testes E2E e regressão com foco em riscos.
- Comunicação direta com cliente e PO para validação contínua e rápida.
- Monitoramento de métricas de qualidade e report semanal com bugs e cobertura.

Exemplos de cenários de testes

Cenário: Simular compra de imóvel financiado

Dado que o cliente acessa o portal de imóveis

Quando seleciona um imóvel com opção de financiamento

E preenche seus dados e renda

Então o sistema deve exibir as opções de financiamento disponíveis

E a simulação deve incluir taxa, valor total e parcelas

Cenário: Enviar proposta ao banco para análise de crédito

Dado que a imobiliária recebe uma proposta do cliente

E a documentação está completa

Quando a imobiliária envia para análise

Então o sistema deve registrar o envio

E o banco deve receber os dados via integração externa

Cenário: Validar comunicação entre microserviço de propostas e microserviço de financiamento

Dado que o microserviço de proposta está em execução

E o microserviço de financiamento está disponível

Quando a proposta for enviada

Então o schema do contrato deve ser compatível com o esperado

E o retorno deve conter status 201 e ID da proposta

Cenário: Realizar teste de carga em endpoint de simulação de financiamento

Dado que o sistema está em ambiente de staging

Quando 1000 simulações são realizadas em 5 minutos

Então o tempo de resposta médio deve ser menor que 2s

E nenhuma requisição deve retornar erro 5xx