

1º Cenário

Você está trabalhando como Analista de Qualidade em um projeto de integração de marketplaces de terceiros, como Amazon e Mercado Livre, em um grande e-commerce. A equipe de desenvolvimento concluiu a implementação da integração e você é responsável por realizar os testes para garantir que a funcionalidade esteja funcionando corretamente. Um detalhe importante, nessa integração, temos integração de estoque, anúncios, faturamento, pedidos e preço.

Descrever detalhadamente o plano de testes que você implementaria para validar a integração com o marketplace. Inclua no seu plano:

1. Documentação e Materiais de Apoio:

- **Identificação da documentação:** Para ter um bom norte e entendimento sobre o funcionamento dos sistemas, eu consultaria as documentações, buscando informações básicas e técnicas para identificar as funcionalidades, quais as URLs das APIs, se possuem ambiente de testes, como funcionam as requisições, quais os endpoints existentes e se existem regras de negócio aplicadas
- **Análise da documentação:** Após conhecer as funcionalidades e entendê-las um pouco melhor, acredito que já seria possível ter uma base de o que testar, porque testar é o mais importante: como realizar os testes. Em muitos casos as documentações já trazem alguns exemplos e isso acaba auxiliando bastante no processo
- **Mapeamento dos requisitos:** Separando por funcionalidades imagino que seja a melhor forma, dentro de cada funcionalidade, pode-se definir quais requisitos são necessários e sabendo os requisitos, é possível guiar os testes que serão feitos, sejam eles manuais ou automatizados
- **Utilização de ferramentas:** Na análise da documentação e gestão dos requisitos eu utilizaria a ferramenta Jira, pois acredito que tenha uma simples utilização e funcionalidades que podem auxiliar muito nas demandas da equipe. Para tomar nota de forma mais rápida e prática, penso que o Notion seria um ótimo aliado

2. Abrangência dos Testes:

- **Funcionalidades:**

- Cadastro de anúncios: verificar se os anúncios estão de acordo, com as informações corretas (inclusive o preço)
- Pedidos: validar o fluxo completo de um pedido garantindo atualização do status, desde o momento em que o cliente compra, passando pelos meios de pagamento e também entrega
- Gerenciamento de estoque: confirmar se o estoque está sendo consumido conforme o esperado
- Erros: identificar se existe um tratamento para telas de erro e como o sistema se portará frente a um erro na integração (instabilidade, endpoint retornando erro 500, entre outras)
- Envio de e-mail: garantir que todos os processos que possuem gatilhos de envio de e-mail estejam funcionais
- Faturamento: s

Estas funcionalidades foram selecionadas devido a influência que elas possuem no fluxo de utilização da integração, e pensando também na validação do funcionamento da mesma.

- **Casos de uso:**

Cadastro de anúncios	
Sucesso	Anúncios sendo listados com as informações corretas
Falha	Anúncio com erro 404
Carga	Cadastro de um número alto de produtos simultaneamente

Pedidos	
Sucesso	Pedido recebido com todos os dados corretos
Falha	Pedido com dados faltando

Gerenciamento de estoque	
Sucesso	Pedido finalizado com y quantidade e o estoque contendo x-y
Falha	Pedido finalizado sem consumir o estoque

Erros	
Sucesso	Endpoint retornando erro 500 na api, na interface uma mensagem amigável
Falha	Sistema tenta realizar uma requisição com token desatualizado, então retorna erro 401 na api, na interface mensagem genérica de erro

Envio de e-mail	
Sucesso	E-mail enviado ao finalizar pedido
Falha	E-mail não enviado por dados faltando

- **Priorização dos testes:** Todos os testes são importantes e devem cumprir o seu papel, porém, os testes priorizados serão sempre àqueles que impactam e/ou impedem o uso do sistema, porque estão relacionados diretamente a satisfação do cliente e consequentemente aos resultados obtidos

3. Execução dos Testes:

- **Ambiente de teste:** No mínimo seria necessário um ambiente de homologação. Em homologação serão lançados todos os testes nas fases iniciais e intermediarias do desenvolvimento, para que não haja nenhuma alteração nos dados reais dos clientes. O cenário ideal seria um ambiente de homologação para cada time de desenvolvimento para evitar conflitos nos releases
- **Dados de teste:** Utilizaria alguma biblioteca semelhante a faker-js, pois é muito prática, útil e de simples utilização

- **Ferramentas de automação:** Para os testes de API, utilizaria o Postman, pois ele mesmo possibilita a automatização dos testes em endpoints. Nos testes funcionais utilizaria o Cypress pela velocidade que a ferramenta entrega
- **Registro de resultados:** Como estamos falando sobre os testes automatizados, aliando o uso do Cypress, optaria pelo Cypress Cloud para rastrear falhas, comparar execuções anteriores e facilitar a análise da evolução dos testes

2º Cenário

Você está trabalhando como Analista de Qualidade em um projeto de integração com a ferramenta de gerenciamento de estoque Bling em um grande e-commerce. A equipe de desenvolvimento concluiu a implementação da integração e você é responsável por realizar os testes para garantir que a funcionalidade esteja funcionando corretamente.

Descrever detalhadamente o plano de testes que você implementaria para validar a integração com a ferramenta Bling. Inclua no seu plano:

1. Documentação e Materiais de Apoio:

- **Identificação da documentação:** Por ser uma integração parecida com a do primeiro cenário, existem muitas semelhanças, uma delas, seria na busca por informações consistentes e detalhadas sobre o produto que está sendo integrado. Inicialmente eu visitaria o site da bling para entender como funciona a ferramenta e quais as funcionalidades disponíveis, depois disso, olharia a documentação para pegar os detalhes técnicos e método de funcionamento da integração e da sua API. Acredito que desta forma teria um conteúdo para definir quais testes serão realizados e de que forma
- **Mapeamento dos requisitos:** Conforme mencionei no 1º cenário, eu faria da mesma forma: dividindo os requisitos por funcionalidades, e

definindo se são funcionais ou não funcionais. Como já sei que uma das formas de controlar o estoque na Bling seria por meio de notas fiscais, um exemplo de requisito seria: O sistema deve atualizar o estoque ao entrar uma nota fiscal

- **Utilização de ferramentas:** Utilizaria as ferramentas Jira e Notion, o Jira por ser um grande aliado e facilitador na gestão de projetos. Já o Notion, seria necessário para me auxiliar nas anotações, tópicos necessários e pontos de atenção de uma forma simples, para tomar nota de tudo e posteriormente transferir todas as informações da forma mais clara e objetiva possível para o plano de testes

2. Abrangência dos Testes:

- **Funcionalidades:**
 - Cadastro de produtos: Para garantir que as informações estão disponíveis e está sendo possível manuseá-las
 - Notas fiscais: Devido a importância que elas possuem para as vendas, quanto na gestão de estoques
 - Gerenciamento do estoque: Ter a certeza de que as quantidades dos produtos estão sendo atualizadas a cada nova movimentação no estoque
 - Pedidos: Garantir que todo o processo que envolve os pedidos estão em pleno funcionamento (importante)
 - Sincronização de dados: Avaliar se as informações que estão sendo atualizadas estão sendo refletidas via integração
- **Priorização dos testes:** Os principais critérios serão o impacto da funcionalidade no uso da integração, se essa funcionalidade pode ou não gerar riscos relacionados à segurança e também a importância do valor que a funcionalidade irá agregar no contexto geral

3. Execução dos Testes:

- **Dados de teste:** Acredito que o uso de ferramentas que gerem JSON com dados fictícios para testes seria muito interessante, porém, teria também a possibilidade de buscar estes dados nas APIs dos sistemas integrados, o que acaba facilitando o processo. Em alguns casos as APIs têm um GET nos seus endpoints apontando para os dados de homologação
- **Ferramentas de automação:** Para os testes de API, utilizaria o Postman, pois ele mesmo possibilita a automatização dos testes em endpoints. Nos testes funcionais utilizaria o Cypress pela velocidade que a ferramenta entrega
- **Registro de resultados:** Como estamos falando sobre os testes automatizados, aliando o uso do Cypress, optaria pelo Cypress Cloud

3º Cenário

O cliente entrou em contato devido a uma questão relacionada aos seus anúncios. Quando os itens ficam sem estoque, em vez de serem marcados como "Pausado (sem estoque)", eles estão apenas sendo marcados como "Pausado", o que faz com que os anúncios permaneçam disponíveis para venda no Mercado Livre, mesmo sem estoque real. Isso está resultando em pedidos sendo feitos para itens sem estoque disponível, o que prejudica a reputação do cliente, pois ele precisa cancelar as vendas devido à falta de estoque.

Atualmente, o cliente possui 38 anúncios com a situação "Pausado", e ele relata que não os marcou manualmente como pausados. Quando enviamos uma atualização manual de estoque, o anúncio muda para a situação "Pausado sem estoque" e fica indisponível para venda no Mercado Livre. Ao verificar o log dos anúncios, não encontramos registros de envio de informação de estoque igual a zero quando o estoque se esgota no Magazord. Um exemplo de anúncio afetado é o seguinte: MLB3097510082. Envio em anexo um print do anúncio e um print do log, mostrando que não foram encontradas informações registradas na abertura da atividade.

Para o caso acima, descreva quais seriam os passos para análise e identificação do problema tendo em vista a documentação do próprio ML. Visto que não possuo acesso ao painel seller do cliente e nem ao painel do mesmo no ERP Magazord, quais seriam as possíveis hipóteses do erro.

Proposta de Solução

Inicialmente eu testaria o mesmo cenário em ambiente de homologação (se houver) no ambiente do cliente que relatou o problema e também em outro cliente. Identificaria um produto com um item em estoque, ou então alteraria o estoque de determinado produto para ser 1. Após ajustar o estoque, realizaria uma venda deste produto para validar como o sistema está se portando para ambos os clientes, desta forma conseguiria ter certeza se é um comportamento global ou se ocorre somente com este cliente.

Tendo esta resposta eu partiria para as parametrizações do sistema, para confirmar se não se trata de uma configuração específica que está ocasionando tal comportamento. Para me auxiliar no entendimento das parametrizações e das regras de negocio envolvidas eu consultaria a documentação do Mercado Livre, pois imagino que lá estarão disponíveis todas as informações essenciais para uso da integração e suas particularidades. Verificaria se há algum evento assíncrono, fila ou hook responsável por enviar o estoque para o ML, o que poderia estar falhando ou sendo ignorado.

Acredito que seria válido também tentar entender com os devs do projeto se existe algum gatilho que faz com que o anúncio tenha a situação atualizada e como esse fluxo funciona.

Causas prováveis:

- Falta de parametrização no cadastro do cliente
- Integração não se portando da forma esperada
- Problema nas requisições que realizam esta alteração
- Alteração na estrutura da API do ML sem aviso prévio

- A ausência de envio de estoque 0 pode indicar falha no disparo automático do evento de sincronização

4º Cenário

Você está encarregado de testar a validação de dados cadastrais em um sistema de gerenciamento de usuários. O sistema possui os seguintes campos de cadastro:

- Nome completo
- E-mail
- Número de telefone
- Data de nascimento
- Endereço (com campos para rua, cidade, estado e CEP)

Houve alterações nos campos mencionados acima no cadastro do cliente, quais testes você faria para validar que o(s) campo(s) esta(ão) funcionando conforme esperado?

Casos de Teste

CT001 - Cadastro com dados pessoais válidos	
Objetivo	Validar se o cadastro ocorre com todos os campos preenchidos corretamente
Pré-condições	Ter permissão para realizar o cadastro
Passos	1- Acessar a tela de cadastro de pessoa 2- Preencher o campo nome com Gustavo Peres 3- Preencher o campo e-mail com gustavorperes@gmail.com 4- Preencher o campo numero de telefone com (53) 98100-9180

	5- Preencher o campo data de nascimento 06/12/1995 6- Preencher o campo rua com Rua Olegario Mariano 7- Preencher o campo cidade com Pelotas 8- Preencher o campo estado com RS 9- Preencher o campo cep com 96065-390 10- Clicar no botão “Gravar”
Resultado Esperado	Sistema deve exibir a mensagem “Usuário cadastrado com sucesso” e salvar os dados no banco de dados

CT002 - Cadastro com e-mail e data de nascimento inválidos	
Objetivo	Validar se o sistema impede o cadastro com os campos E-mail e Data de Nascimento preenchidos incorretamente
Pré-condições	Ter permissão para realizar o cadastro
Passos	1- Acessar a tela de cadastro de pessoa 2- Preencher o campo nome com Gustavo Peres 3- Preencher o campo e-mail com gustavorperes.br 4- Preencher o campo numero de telefone com (53) 98100-9180 5- Preencher o campo data de nascimento 99/99/0000 6- Preencher o campo rua com Rua Olegario Mariano 7- Preencher o campo cidade com Pelotas

	8- Preencher o campo estado com RS 9- Preencher o campo cep com 96065-390 10-Clicar no botão “Gravar”
Resultado Esperado	Sistema deve exibir a mensagem “Campos inválidos” e destacar em vermelho quais são os campos

CT003 - Cadastro com telefone e CEP inválidos	
Objetivo	Validar se o sistema impede o cadastro com os campos Telefone e CEP preenchidos incorretamente
Pré-condições	Ter permissão para realizar o cadastro
Passos	1- Acessar a tela de cadastro de pessoa 2- Preencher o campo nome com Gustavo Peres 3- Preencher o campo e-mail com gustavorperes@gmail.com 4- Preencher o campo numero de telefone com (53) 98100-918 5- Preencher o campo data de nascimento 06/12/1995 6- Preencher o campo rua com Rua Olegario Mariano 7- Preencher o campo cidade com Pelotas 8- Preencher o campo estado com RS 9- Preencher o campo cep com 96065-3AB 10-Clicar no botão “Gravar”
Resultado Esperado	Sistema deve exibir a mensagem “Campos inválidos” e destacar em vermelho quais são os campos