Data: 24/04

Zorder: Eduardo Sangaletti

**Resolução do Teste**

**1º CENÁRIO**

1. Documentação e Material de Apoio:
   1. **Identificação da documentação:** a documentação oficial dos marketplaces será uma boa fonte para materiais de apoio, já possuindo certos detalhes técnicos sobre APIs disponíveis, endpoints que serão utilizados entre outras informações e detalhes técnicos. Vale lembrar que, para um cenário de elaboração de testes, para que se tenha uma cobertura mais completa dos requisitos, quanto mais informações e materiais de apoio que deixem claro o funcionamento, e os detalhes técnicos, melhor.
   2. **Análise da documentação:** será realizado uma análise detalhada da documentação para entender os processos, podendo realizar primeiramente um fluxo de atividades para deixar mais visual e claro os fluxos de dados, dessa forma seria possível visualizar de maneira mais intuitiva os elementos das integrações e como interagem entre si.
   3. **Mapeamento dos requisitos:** uma matriz de rastreabilidade pode ajudar a mapear de forma bem clara e objetiva os requisitos da integração, relacionando dessa forma os requisitos não funcionais, funcionais, regras de negócio, e a partir disso começar a relacionar os casos de testes, e entender até quais são os pontos de testes que são um pouco mais críticos e que precisam de mais atenção (quando há muitos relacionamentos em uma funcionalidade específica por exemplo).
   4. **Utilização de ferramentas:** ferramentas como o Jira, seriam uma opção interessante para realizar a documentação do projeto, incluindo os diagramas de casos de uso, fluxo de atividades, e até a matriz de rastreabilidade.
2. Abrangência dos Testes:
   1. **Funcionalidades:**
      1. Integração de estoque: garantir que as informações de estoque dos produtos sejam sincronizadas corretamente entre o sistema do e-commerce e o marketplace;
      2. Integração de anúncio: garantir que os anúncios dos produtos sejam publicados corretamente no marketplace, incluindo informações precisas e completas;
      3. Integração de faturamento: garantir que as faturas para pedidos feitos no marketplace sejam geradas corretamente e enviadas aos clientes, se os valores e informações estão corretos;
      4. Integração de pedidos: garantir que os pedidos feitos no marketplace sejam corretamente importados e processados no sistema do e-commerce;
      5. Integração de preço: garantir que os preços dos produtos no marketplace sejam atualizados corretamente de acordo com as alterações feitas no sistema do e-commerce.
   2. **Casos de uso:** poderia ser utilizado o formato de casos de uso em histórias de usuário, onde é mapeado e descrito os fluxos principais, secundários, das rotinas bem como seus cenários de sucesso e falha, já aproveitando para vincular com as mensagens apresentadas em cada caso e os seus requisitos associados.
      1. Exemplo: integração de estoque poderia ser testado o fluxo principal onde seria a atualização correta do estoque quando um produto é vendido, bem como um cenário de falha onde não há produto em estoque.
   3. **Priorização dos testes:** as funcionalidades com nível de criticidade normalmente serão priorizadas, a criticidade da funcionalidade está na maioria das vezes fortemente relacionada com o impacto no negócio e riscos potenciais, por exemplo o processo de pagamento, é um processo que não há margens para erro, e qualquer problema que apresente, é crítico, pois gera um risco e grande impacto para o negócio.
3. Execução dos Testes:
   1. **Ambiente de teste:** é possível utilizar um ambiente de homologação que replicará todas as configurações do ambiente de produção que está em operação no cliente, dessa forma é possível realizar testes de maneira isolada sem impactar o cliente, bem mais segura e próxima da realidade na simulação de testes.
   2. **Dados de teste:** os dados podem ser gerados de maneira manual de acordo com a necessidade, produtos, estoque, pedidos, usuário, formas de pagamento etc. ou se possível utilizar dados reais disponibilizados pelas plataformas do ambiente de produção para o ambiente de homologação.
   3. **Ferramentas de automação:** é possível utilizar o Postman para os testes de API e Selenium para testes de funcionalidade, regressão, desempenho.
   4. **Registro de resultados:** pode ser utilizado novamente o Jira, nele pode ser centralizado todas as informações e registros dos testes executados, dessa forma há uma facilidade maior no acompanhamento dos resultados, e análise para tomadas de decisão.

**2º CENÁRIO**

1. Documentação e Materiais de Apoio:
   1. **Identificação da documentação:** a documentação utilizada será a oficial disponibilizada da Bling, onde iremos encontrar passo a passo para integração, os manuais das APIs, e todas as outras documentações relacionadas à integração. Além das documentações do Bling, também é de suma importância possuir as especificações técnicas e os requisitos do projeto.
   2. **Mapeamento dos requisitos:** é possível mapear os pontos de testes através de matriz de rastreabilidade com casos de uso, vinculando com requisitos da aplicação e extraindo da matriz, roteiros/fluxos principais de operação do sistema e dessa forma montar os roteiros e casos de testes.
   3. **Utilização de ferramentas:** a ferramenta utilizada para o gerenciamento e mapeamento pode ser feito com Jira.
2. Abrangência dos Testes:
   1. **Funcionalidades:** as funcionalidades da integração com a Bling que serão testadas incluem a atualização de estoque, processamento dos pedidos, sincronização de produtos e geração de relatórios
   2. **Priorização dos testes:** os testes serão priorizados com base na criticidade da funcionalidade, e que geralmente está relacionada com o impacto no negócio e certos riscos. Funcionalidades críticas para o funcionamento do e-commerce, como processamento de pedidos e atualização de estoque, terão prioridade mais alta.

1. Execução dos Testes:
   1. **Dados de teste:** os dados de teste serão obtidos através de uma combinação de dados sintéticos gerados para representar produtos, estoque e pedidos, e dados reais da Bling, quando disponíveis e apropriados. Isso garantirá que os testes sejam realizados em condições realistas.
   2. **Ferramentas de automação:** para executar os testes, utilizaremos ferramentas de automação específicas para cada tipo de teste:
      1. Para testes de integração e API, podemos usar o Postman, SoapUI ou Insomnia.
      2. Para testes de funcionalidade, podemos considerar ferramentas de automação de teste, como o Katalon Studio que é uma IDE bem completa e permite juntamente com outras ferramentas e tecnologias realizar muitos tipos de testes, e nesse contexto poderia ser utilizado Groovy juntamente com Selenium, para simular interações do usuário.
   3. **Registro de resultados:** os resultados dos testes serão registrados em uma ferramenta de gerenciamento de testes, como o Jira ou TestRail onde cada caso de teste terá seu status atualizado para refletir se passou ou falhou.

**3º CENÁRIO**

1. **Passo:** Verificar a documentação nos tópicos pertinentes relacionados com os atributos de um produto.
   1. Obter informações sobre como os atributos de um produto afetam sua disponibilidade para venda.
   2. Identificar os requisitos necessários para um produto estar disponível ou indisponível para venda.
2. **Passo:** Analisar como um produto fica disponível e indisponível em um anúncio.
   1. Compreender os processos envolvidos na disponibilização e na retirada de um produto de um anúncio.
   2. Explorar os critérios que determinam a disponibilidade ou indisponibilidade de um produto para venda.
3. **Passo:** Investigar os possíveis cenários em que um produto pode se encontrar.
   1. Identificar os diferentes estados em que um produto pode estar, como disponível, pausado ou sem estoque.
   2. Compreender os motivos que podem levar a um produto ser pausado, como inconsistências no cadastro ou violações de restrições.
4. **Passo:** Sintetizar as informações obtidas da documentação para formar uma compreensão abrangente do processo.
   1. Resumir os principais pontos relacionados à disponibilidade de produtos para venda.
   2. Consolidar os cenários possíveis em que um produto pode se encontrar e as ações necessárias para cada um.
5. Seguindo esses passos, será possível obter uma compreensão sólida dos atributos de um produto e dos processos envolvidos em sua disponibilização e indisponibilização para venda, otimizando assim o tempo e facilitando a análise do contexto em questão.
6. **Hipóteses do erro:**
   1. Não definição do sub\_status como "out\_of\_stock":
      1. Problema: O produto permanece disponível para venda mesmo quando o status é definido como "Pausado".
      2. Causa provável: Falta de definição do sub\_status como "out\_of\_stock" quando o status é definido como "Pausado".
      3. Impacto: O produto pode ser vendido mesmo quando está sem estoque, resultando em uma experiência do usuário inconsistente.
      4. Solução: Garantir que o sub\_status seja corretamente definido como "out\_of\_stock" sempre que o status do produto for alterado para "Pausado" por falta de estoque.
   2. Ausência de especificação do "available\_quantity" quando o estoque é zero:
      1. Problema: O anúncio não reflete corretamente a disponibilidade do produto quando o estoque é zerado.
      2. Causa provável: Não ter sido feita a especificação do "available\_quantity" quando o estoque atinge 0.
      3. Impacto: O sistema apenas pausa o anúncio sem sinalizar que o produto está fora de estoque, o que pode levar a vendas inadequadas.
      4. Solução: Certificar-se de que o "available\_quantity" seja definido como 0 quando o estoque do produto estiver zerado, para garantir uma representação precisa da disponibilidade do produto.

**4º CENÁRIO**

Para testar todos os campos que sofreram alterações é possível realizar os seguintes roteiros e casos de testes:

|  |  |  |
| --- | --- | --- |
| Roteiro de Teste 1 – Campo Nome | | |
| Objetivo | Validar o campo “Nome” com diferentes entradas de dados. | |
| Sequência de ações | 1. Preencher o campo “Nome” com:  a. Letras maiúsculas, minúsculas sem espaço;  b. Letras com caracteres especiais;  c. Letras com caracteres numéricos;  d. Deixar vazio e tentar prosseguir;  e. Quantidade de caracteres que exceda o limite do campo;  f. Letras maiúsculas nas iniciais, minúsculas no restante e com espaçamento. | |
| Pós-condição e  Resultado esperado | 1. Para os cenários inválidos, deve invalidar, não deixar prosseguir e apresentar uma mensagem para o usuário:  a. Inválido (falta sobrenome);  b. Inválido (não é permitido caracteres especiais);  c. Inválido (não é permitido caracteres numéricos);  d. Inválido (campo de preenchimento obrigatório);  e. Inválido (campo excedeu o limite máximo de caracteres permitido);  f. Válido, deve validar e permitir prosseguir. | |
|  | | |
| Caso de Teste | Entrada | Saída |
| CT01 - a | EduSangaletti | Inválida |
| CT02 - b | Edu@rdo San@l&tti | Inválida |
| CT03 - c | Edu4rd0 S4ng4l3tti | Inválida |
| CT04 - d | “ “ | Inválida |
| CT05 - e | Eduaaaardooooooooo Sangaaaalee... | Inválida |
| CT06 - f | Eduardo Sangaletti | Válida |

|  |  |  |
| --- | --- | --- |
| Roteiro de Teste 2 – Campo E-mail | | |
| Objetivo | Validar o campo “E-mail” com diferentes entradas de dados. | |
| Sequências de ações | 1. Preencher o campo “E-mail” com:  a. E-mail com espaçamento;  b. E-mail sem “@”;  c. E-mail sem o domínio (após o @);  d. E-mail com quantidade de caracteres que exceda o limite do campo;  e. E-mail iniciando com caracteres especiais e/ou terminando com caracteres especiais;  f. E-mail com letras maiúsculas, minúsculas, com caracteres especiais; | |
| Pós-condição e  Resultado esperado | 1. Para os cenários inválidos, deve invalidar, não deixar prosseguir e apresentar uma mensagem para o usuário:  a. Inválido (e-mail com espaçamento);  b. Inválido (e-mail sem “@”);  c. Inválido (e-mail sem domínio após o “@”);  d. Inválido (e-mail excedeu o limite máximo de caracteres);  e. Inválido (não é permitido caracteres especiais no início ou no fim do e-mail);  f. Válido, deve validar e permitir prosseguir. | |
|  | | |
| Caso de Teste | Entrada | Saída |
| CT01 - a | eduardo sangaletti@gmail.com | Inválida |
| CT02 - b | eduardosangalettigmail.com | Inválida |
| CT03 - c | eduardosangaletti@ | Inválida |
| CT04 - d | eduaaaaaardosangalee... @gmail.com | Inválida |
| CT05 - e | !eduardosanga@gmail.com# | Inválida |
| CT06 - f | eduardo\_Sanga@gmail.com | Válida |

|  |  |  |
| --- | --- | --- |
| Roteiro de Teste 3 – Campo Número de Telefone | | |
| Objetivo | Validar o campo “Número de Telefone” com diferentes entradas de dados | |
| Sequência de ações | 1. Preencher o campo “Número de Telefone”:  a. Com um número menor que 11 caracteres (sem código de área por exemplo, considerando um cenário nacional);  b. Com letras e/ou caracteres especiais;  c. Com um número com código de área e número totalizando 11 dígitos | |
| Pós-condição e  Resultado  esperado | 1. Para os cenários inválidos, deve invalidar, não deixar prosseguir e apresentar uma mensagem para o usuário:  a. Inválido (número está incompleto);  b. Inválido, não permite informar caracteres diferentes de números;  c. Válido, deve validar e permitir prosseguir. | |
|  | | |
| Caso de Teste | Entrada | Saída |
| CT01 - a | (47)98873 | Inválida |
| CT02 - b | (47)9a87& | Inválida |
| CT03 - c | (47) 988731617 | Válida |

|  |  |  |
| --- | --- | --- |
| Roteiro de Teste 4 – Campo Data de Nascimento | | |
| Objetivo | Validar o campo “Data de Nascimento” com diferentes entradas de dados | |
| Sequência de ações | 1. Preencher o campo “Data de Nascimento”:  a. Com uma data que supere o limite máximo/mínimo de dia/mês/ano;  b. Com letras e/ou caracteres especiais;  c. Com uma data válida dentro dos limites mínimo e máximo; | |
| Pós-condição e  Resultado  esperado | 1. Para os cenários inválidos, deve invalidar, não deixar prosseguir e apresentar uma mensagem para o usuário:  a. Inválido (data supera os limites mínimos/máximos de dia/mês/ano);  b. Inválido, não deve permitir inserir caracteres diferentes de números;  c. Válido, deve validar e permitir prosseguir | |
|  | | |
| Caso de Teste | Entrada | Saída |
| CT01 - a | 32/13/9999 e 00/00/1000 | Inválida |
| CT02 - b | e0/!2/2000 | Inválida |
| CT03 - c | 24/04/2024 | Válida |

|  |  |  |
| --- | --- | --- |
| Roteiro de Teste 5 – Campo Endereço | | |
| Objetivo | Validar o campo “Endereço” com diferentes entradas de dados | |
| Sequência de ações | 1. Preencher o campo “CEP” (considerando que o CEP não puxará todos os campos e que não tenha integração com DNE):  a. Com cep zerado;  b. Com letras e/ou caracteres especiais;  c. Com cep menor que 8 caracteres;  d. Sem cep;  e. Cep com 8 dígitos  2. Preencher os campos “Estado”, “Cidade” e “Rua”:  a. Letras maiúsculas, minúsculas com e sem espaço;  b. Letras com caracteres especiais;  c. Letras com caracteres numéricos;  d. Deixar vazio e tentar prosseguir;  e. Quantidade de caracteres que exceda o limite do campo; | |
| Pós-condição e  Resultado  esperado | 1. Para os cenários inválidos, deve invalidar, não deixar prosseguir e apresentar uma mensagem para o usuário:  a. Inválido (não existe cep 00.000-000, porém pode ser deixado para informar um endereço desconhecido, caso não seja obrigatório);  b. Inválido, não deve permitir inserir caracteres diferentes de números;  c. Inválido, deve conter exatos 8 dígitos;  d. Inválido (campo de preenchimento obrigatório);  2. Para os cenários inválidos, deve invalidar, não deixar prosseguir e apresentar uma mensagem para o usuário:  a. Válido, deve validar e permitir prosseguir;  b. Inválido (não é permitido caracteres especiais);  c. Válido, deve validar e permitir prosseguir;  d. Inválido (campo de preenchimento obrigatório);  e. Inválido (campo excedeu o limite máximo de caracteres permitido). | |
|  | | |
| Caso de Teste | Entrada | Saída |
| CT01 - 1a | 00.000-000 | Inválido |
| CT02 - 1b | &9.is0-000 | Inválida |
| CT03 - 1c | 89.161-2 | Inválida |
| CT04 - 1d | “ “ | Inválida |
| CT05 - 1e | 89.161-220 | Válida |
| CT06 - 2a | SC – Rio do Sul – Roberto Pinheiro | Válida |
| CT07 - 2b | SC – Rio do $u7 – Robert@ Pinh#eiro | Inválida |
| CT08 - 2c | SC – Rio do Sul – Rua 25 de Março | Válida |
| CT09 - 2d | “ “ | Inválida |
| CT10 - 2e | SC – Rio do Sul – Robeeeeertoooo... | Inválida |