

Bloco 3 - Report de Bugs

Exemplos de Bugs Encontrados

| ID | Descrição | Severidade | Status |
|--------|---|------------|---------|
| BUG-01 | Produto inativo aparece nos resultados de pesquisa | Alta | Aberto |
| BUG-02 | Score exibido incorretamente após selecionar fornecedor | Alta | Aberto |
| BUG-03 | Campo de justificativa não obrigatório ao prosseguir com fornecedor "Risco" | Média | Aberto |
| BUG-04 | Ordenação incorreta dos fornecedores por score (ASC ao invés de DESC) | Média | Aberto |
| BUG-05 | Mensagem de erro genérica ao tentar comprar fornecedor com pendência | Baixa | Fechado |

Comunicação e Ações

- **Report formal de bugs:** criado seguindo o padrão **JIRA**, com campos de Resumo, Passos, Resultado Esperado/Obtido e Evidências.
- **Interação com o Product Owner (PO):** comunicação ativa para validação das regras de negócio (ex.: RN1 e RN4).
- **Acompanhamento técnico com Devs:** análise de logs e validação de payloads para confirmar origem das falhas.
- **Sugestão de priorização:** baseada em **severidade x impacto funcional**, garantindo foco nos erros que afetam diretamente o fluxo de compra.

Bug Report - Formato (Jira/TestLink)

ID: BUG-001

Resumo: Produto com preço e score divergentes ao adicionar ao carrinho

Status: Aberto

Severidade: Alta

Prioridade: Alta

Componente: Módulo de Pedido de Compra

Ambiente: Homologação – WebApp Systock (Frontend Vue.js / Backend Laravel)

Responsável: A definir

Descrição

Ao adicionar determinados produtos ao carrinho, os valores exibidos de **preço**, **score de análise** e **quantidade** não correspondem às informações apresentadas na listagem de produtos.

O problema ocorre de forma consistente e impacta diretamente a integridade dos pedidos de compra.

Passos para Reprodução

1. Acessar o sistema Systock com usuário autenticado.
 2. Ir até o módulo **"Produtos"**.
 3. Pesquisar o produto **"Caneta Azul – SKU 1023A"** (ativo e disponível).
 4. Selecionar o produto e clicar em **"Adicionar ao carrinho"**.
 5. Abrir o carrinho e observar as informações exibidas.
-

Resultado Esperado

- O produto deve ser exibido no carrinho com:
 - **Preço unitário** correto (R\$ 4,50);
 - **Score de análise** compatível com o da listagem ("A+");
 - **Quantidade** igual à selecionada (1 unidade).
-

Resultado Obtido

- O produto aparece no carrinho com:

- **Preço:** R\$ 0,00 (ou valor divergente);
- **Score:** "Indefinido";
- **Quantidade:** alterada automaticamente para "2".

Evidências

- **Print 01:** Produto na listagem (dados corretos).
- **Print 02:** Produto no carrinho (dados incorretos).
- **Console Log:** `score: undefined` no payload de resposta.
- **Vídeo (opcional):** Fluxo completo demonstrando o erro.

Análise Técnica Inicial

| Possível Causa | Descrição |
|--|---|
| Falha na comunicação entre front e API | Campos <code>price</code> e <code>score</code> não retornam corretamente no POST <code>/api/cart/add</code> . |
| Problema de binding no estado local | O estado <code>cartItem</code> é sobrescrito antes de receber o retorno da API. |
| Cache desatualizado | Dados antigos do produto podem estar sendo exibidos no carrinho. |

Impacto

- Afeta a **confiabilidade das informações de compra**;
- Pode gerar **pedidos incorretos** ou **valores inconsistentes**;
- Compromete o cálculo de **rentabilidade e score** dos fornecedores.

Recomendações

1. Revisar o endpoint `/api/cart/add` e validar o payload retornado.
2. Garantir que o front-end atualize o estado do carrinho após a resposta da API.
3. Adicionar testes de integração e validação de consistência de dados.
4. Implementar alerta visual quando o carrinho receber dados inválidos.

Conclusão

O bug representa uma falha crítica no fluxo de pedido de compra e deve ser tratado com **prioridade alta**.

Recomenda-se correção imediata, seguida de **testes de regressão automatizados** e **revalidação manual** do módulo de carrinho.