**Pontos Negativos do MongoDB na ZettaMarket + Soluções**

Falta de integridade relacional (sem Foreign Key reais) MongoDB não possui suporte nativo a relacionamentos fortes como os bancos relacionais. Por exemplo, não há garantia automática de integridade entre a loja, seus produtos e usuários.

Impacto na ZettaMarket:  
 Risco de dados órfãos ou inconsistentes, como um produto apontar para uma loja que não existe mais.

Solução:

* Implementar validações na aplicação (backend) para garantir integridade.
* Utilizar referências manuais e verificações periódicas com scripts de manutenção.

Limitações em transações complexas multi-documento. MongoDB só começou a suportar transações ACID com múltiplos documentos a partir da versão 4.0, e com algumas limitações de performance.

Impacto na ZettaMarket:Problemas potenciais em fluxos que envolvam múltiplas coleções ao mesmo tempo, como pedidos + pagamento + estoque.

Solução:

* Sempre que possível, usar modelagem com documentos aninhados (embutidos).
* Usar transações somente quando realmente necessário e manter as operações pequenas e rápidas.
* Testar bem as transações em ambientes de stress.

Modelagem de dados mais complexa para iniciantes. Modelar dados em NoSQL exige pensar diferente do SQL tradicional — é necessário decidir entre referenciar ou embutir dados.

Impacto na ZettaMarket:  
 Tempo maior para treinar novos desenvolvedores ou colaboradores vindos de bancos relacionais.

Solução:

* Utilizar padrões de modelagem do MongoDB (como Bucket Pattern, Outlier Pattern).
* Consultar a MongoDB University e usar guias oficiais de boas práticas.
* Criar uma documentação interna clara sobre as estruturas adotadas.

Consistência eventual por padrão. O MongoDB prioriza disponibilidade e particionamento, o que pode levar a leituras temporariamente desatualizadas em ambientes distribuídos.

Impacto na ZettaMarket:  
 Dados como status de pagamento ou estoque podem parecer incorretos por alguns segundos após uma atualização.

Solução:

* Ajustar os parâmetros de readConcern e writeConcern para garantir mais consistência onde necessário.
* Utilizar replica sets com configuração adequada para leitura preferencial do nó primário.

Consumo de armazenamento maior. Documentos no MongoDB podem ocupar mais espaço por armazenarem metadados repetidos (campos em JSON por documento).

Impacto na ZettaMarket:  
 A hospedagem em cloud (como MongoDB Atlas) pode ter custo maior com o tempo.

Solução:

* Utilizar compressão nativa do MongoDB.
* Arquivar ou excluir documentos antigos e irrelevantes periodicamente.
* Modelar dados de forma eficiente, evitando campos desnecessários.

Falta de ferramentas maduras para relátorios avançados (BI). Ferramentas tradicionais de BI (Power BI, Tableau) funcionam melhor com SQL.

Impacto na ZettaMarket:  
 Dificuldade em criar dashboards financeiros, operacionais e estratégicos diretamente com o MongoDB.

Solução:

* Usar conectores como o MongoDB BI Connector para exportar dados em formato relacional.
* Criar pipelines de agregação e views no MongoDB para simplificar consultas.

Complexidade na manutenção de dados aninhados (subdocumentos).

Atualizar ou consultar dados profundamente aninhados pode ser mais complicado e custoso.

Impacto na ZettaMarket:  
 Mais trabalho para buscar e atualizar informações como mensagens de chat, configurações específicas de produtos, etc.

Solução:

* Utilizar projeções e filtros adequados com operadores ($elemMatch, $push, $set, etc).
* Criar índices sobre campos internos quando necessário.