

# Quais os principais problemas no uso de bancos de dados NoSQL?

Os bancos de dados NoSQL oferecem vantagens significativas, como escalabilidade horizontal, alta performance e flexibilidade para lidar com dados não estruturados. No entanto, eles também apresentam desafios e limitações, dependendo do caso de uso. Aqui estão os principais problemas no uso de bancos de dados NoSQL:

## 1. Consistência e Garantias ACID

- Muitos bancos NoSQL optam pelo modelo **BASE** (Basic Availability, Soft-state, Eventual consistency) em vez de ACID (Atomicidade, Consistência, Isolamento, Durabilidade).
- Isso pode levar a problemas de **consistência eventual**, onde dados atualizados em uma parte do sistema podem levar tempo para se propagar para outras partes, resultando em leituras desatualizadas.

## 2. Curva de Aprendizado

- Cada banco de dados NoSQL (como MongoDB, Cassandra, Redis ou DynamoDB) tem seu próprio modelo de dados, API e arquitetura, exigindo **expertise especializada**.
- Desenvolvedores acostumados com bancos relacionais podem ter dificuldade em adaptar-se ao novo paradigma.

## 3. Falta de Padrões

- Ao contrário dos bancos de dados relacionais, que seguem o padrão **SQL**, o ecossistema NoSQL carece de padronização universal.
- Isso dificulta a **migração** entre soluções NoSQL e pode levar a dependência de fornecedor (**vendor lock-in**).

## 4. Consulta e Indexação

- Sistemas NoSQL geralmente não possuem uma linguagem de consulta tão robusta quanto o SQL.
- Realizar consultas complexas ou criar índices personalizados pode ser mais difícil e menos intuitivo.

## 5. Modelagem de Dados

- O design do esquema nos bancos NoSQL exige que você conheça profundamente os padrões de acesso aos dados, porque a **desnormalização** (duplicação de dados) é comum.
- Alterações nos requisitos de acesso podem exigir uma reestruturação significativa dos dados.

## 6. Limitações de Transações

- Muitos bancos NoSQL oferecem **transações limitadas** ou inexistentes em comparação com bancos relacionais, o que é problemático para aplicações que exigem **consistência forte** ou operações envolvendo múltiplos documentos/tabelas.

## 7. Gerenciamento de Escalabilidade

- Apesar de serem projetados para escalar horizontalmente, a **configuração, manutenção e monitoração** de clusters NoSQL em larga escala podem ser complexas e exigir conhecimento especializado.

## 8. Segurança

- Alguns bancos NoSQL priorizam a performance e escalabilidade em detrimento de **recursos de segurança** robustos (como controle de acesso granular e criptografia por padrão).
- Isso pode expor dados sensíveis se não forem tomadas medidas adicionais.

## 9. Custo

- Embora o NoSQL seja atrativo para grandes volumes de dados, a escalabilidade horizontal pode aumentar os custos operacionais em ambientes distribuídos.
- Soluções baseadas em nuvem, como AWS DynamoDB ou MongoDB Atlas, podem gerar despesas significativas com armazenamento e tráfego de dados.

## 10. Inadequação para Certos Casos de Uso

- Para aplicações que requerem **relacionamentos complexos** entre entidades (como sistemas financeiros ou ERPs), bancos relacionais são mais apropriados.
- O uso de NoSQL em cenários inadequados pode levar a baixa performance e maior complexidade no desenvolvimento.

Se você estiver considerando usar NoSQL, é essencial analisar os requisitos específicos do seu projeto e entender as características do banco escolhido. Muitas vezes, um **modelo híbrido** (combinando NoSQL e SQL) pode oferecer o melhor dos dois mundos.