

Projeto Lógico – Banco de Dados NoSQL: Gestão Esportiva

Vinícius Vasconi

9 de fevereiro de 2026

Resumo

Este relatório apresenta o projeto lógico de um sistema de gestão esportiva mapeado para um SGBD NoSQL (MongoDB). Contempla a pesquisa sobre tipos de SGBD NoSQL, a justificativa da escolha do MongoDB, a estratégia de mapeamento do modelo relacional (DER) para documentos, os schemas JSON de validação (via `$jsonSchema`) implementados e a discussão sobre como as restrições conceituais foram garantidas ou tratadas pelo projeto.

Conteúdo

1	Introdução	2
2	Pesquisa e escolha do SGBD	2
3	Estratégia de mapeamento Relacional → NoSQL	2
4	Discussão	2
5	Conclusão	3

1 Introdução

O objetivo deste trabalho é apresentar o projeto lógico, no modelo NoSQL, de um sistema de gestão esportiva originalmente modelado em DER/relacional. A migração do modelo relacional para um modelo orientado a documentos exige decisões conscientes sobre embutimento de dados, referências e o local onde garantir restrições (SGBD versus aplicação). Neste relatório descrevemos as escolhas feitas, justificativas, implementações de validação e limitações inerentes ao paradigma NoSQL.

2 Pesquisa e escolha do SGBD

Pela simplicidade e compatibilidade com JSON, escolhemos o **MongoDB** para a modelagem do nosso banco, o que posteriormente será útil no desenvolvimento da aplicação. Além disso, o fato desse SGBD ser *schema-less* favorece o aprendizado do processo de substituição de um banco relacional para um não relacional.

3 Estratégia de mapeamento Relacional → NoSQL

As decisões estratégicas adotadas foram:

- **Entidades independentes:** Entidades principais como pessoa foram traduzidas diretamente como coleções.
- **Embutir documentos:** Quando o dado só tem sentido no contexto do agregado optamos por embutir o documento (ex.: *jogos* e *classificacao* dentro de *campeonato*).
- **Referências (ObjectId):** Traduzimos a noção de chaves estrangeiras como referências nos documentos. Um exemplo claro foi o documento de penalidade, que recebe um *_id* para jogador, árbitro e jogo.
- **Regras de negócio complexas:** O MongoDB não oferece algumas funcionalidades similares ao postgres, como por exemplo, validar se a data de fim é maior que data início. Nesse caso, abstraímos para a camada de aplicação.

Devido à quantidade de arquivos para composição dos schemas lógicos do MongoDB, optamos por disponibilizá-los no repositório do projeto [1].

4 Discussão

A migração de um modelo relacional para um modelo orientado a documentos envolve trade-offs:

- **Vantagens:** consultas agregadas mais rápidas, flexibilidade de esquema.
- **Desvantagens:** Ausência de chaves primárias e identificadores, necessidade de maior controle da aplicação.

5 Conclusão

As decisões de modelagem e as validações implementadas garantem consistência estrutural próxima ao modelo relacional original, enquanto respeitam o paradigma orientado a documentos. Regras complexas de integridade e relacionamentos entre coleções foram delegadas à camada de aplicação ou podem ser gerenciadas com índices, transações e rotinas auxiliares.

Referências

- [1] Ricardo Nabucoo. Banco-de-dados-myworkbench. <https://github.com/ricardonabucoo/Banco-de-dados-myWorkbench>, 2025. Acesso em: 08 fev. 2026.