

Universidade Federal de Sergipe
Departamento de Computação
Disciplina: Banco de Dados 1

Relatório de Banco de Dados: Parte 2.2

Projeto Lógico NoSQL (MongoDB)

Alunos: Rafael Gomes Oliveira Santos
Alícia Vitória Sousa Santos

Professor: André Britto

São Cristóvão
2026

1 Pesquisa e Estudo de SGBDs NoSQL

A transição do modelo relacional (PostgreSQL) para o modelo não relacional exige uma mudança de paradigma: do foco na normalização para a performance de leitura. O **MongoDB** foi selecionado como a solução NoSQL orientado a documentos.

1.1 Paridade de Impedância e Atributos Multivalorados

A decisão de utilizar o tipo **ARRAY** para idiomas na Parte 2.1 eliminou a necessidade de uma tabela associativa no SQL, permitindo um mapeamento direto para o MongoDB. Isso demonstra uma **paridade de impedância**, onde a estrutura lógica do dado informativo é preservada entre os paradigmas.

2 Mapeamento para a Estrutura NoSQL

O mapeamento consolida as **16 tabelas** originais em **4 coleções**, otimizando a localidade dos dados.

Tabela SQL (Origem)	Coleção NoSQL (Destino)	Estratégia
Usuario, Carteira, Biblioteca	usuarios	Embedding (1:1 e 1:N)
Produto, Jogos, Software, DLC	produtos	Polimorfismo / Unificação
Genero, Jogos_has_Genero	produtos	Denormalização (Array)
Conquista, RequisitoSistema	produtos	Embedding (Atomicidade)
Desenvolvedora, Publicadora	produtos	Extended Reference (Nomes)
Compra, Nota_Fiscal	compras	Embedding 1:1 e Linking
Avaliacao	avaliacoes	Referencing (ID Linking)

Tabela 1: Rastreabilidade das 16 Tabelas Relacionais.

3 Integridade e Restrições de Esquema (\$jsonSchema)

A integridade é mantida via **\$jsonSchema**, garantindo as regras de nulidade (como os campos opcionais de *Avaliação* e *Biblioteca*) e o uso de chaves naturais.

3.1 Coleção: Usuarios

```
1 {
2   "$jsonSchema": {
3     "bsonType": "object",
4     "required": ["nome_completo", "email", "carteira", "biblioteca"],
5     "additionalProperties": false,
6     "properties": {
7       "_id": { "bsonType": "objectId" },
```

```

8     "email": { "bsonType": "string", "pattern": "^.+@.+$" },
9     "carteira": { "bsonType": "object", "required": ["moeda", "
      saldo_atual"] },
10    "biblioteca": {
11      "bsonType": "array",
12      "items": {
13        "bsonType": "object",
14        "required": ["produto_id"],
15        "properties": {
16          "tempo_jogado": { "bsonType": ["date", "null"] }
17        }
18      }
19    }
20  }
21 }
22 }

```

3.2 Coleção: Produtos (Modelo Polimórfico)

Implementa a unificação de especializações e o uso de nomes como chaves naturais para agentes do mercado.

```

1 {
2   "$jsonSchema": {
3     "bsonType": "object",
4     "required": ["tipo", "titulo", "preco", "desenvolvedora", "
      publicadora"],
5     "additionalProperties": false,
6     "properties": {
7       "tipo": { "enum": ["Jogo", "Software", "DLC"] },
8       "desenvolvedora": { "bsonType": "string" },
9       "publicadora": { "bsonType": "string" },
10      "idiomas": { "bsonType": "array", "items": { "bsonType": "string"
        } },
11      "conquistas": {
12        "bsonType": "array",
13        "items": {
14          "bsonType": "object",
15          "required": ["titulo", "descricao", "imagem_icone"]
16        }
17      }
18    }
19  }
20 }

```

3.3 Coleção: Avaliacoes

Mapeada para suportar a nulidade de texto e data, refletindo a PK composta usuário-jogo.

```
1 {  
2   "$jsonSchema": {  
3     "bsonType": "object",  
4     "required": ["id_usuario", "id_produto", "voto"],  
5     "properties": {  
6       "voto": { "bsonType": "int", "minimum": 0, "maximum": 5 },  
7       "texto_analise": { "bsonType": ["string", "null"] },  
8       "data_postagem": { "bsonType": ["date", "null"] }  
9     }  
10  }  
11 }
```

4 Conclusão

O modelo documental proposto eliminou a necessidade de *Joins* complexos, integrando as 16 tabelas em 4 coleções de alta performance e mantendo a fidelidade às restrições do MySQL Workbench.