
RELATÓRIO ACADÊMICO – SISTEMA DE CONTROLE DE ESTUDANTES (MongoDB)

1. Introdução

Este relatório apresenta os resultados do trabalho utilizando o banco de dados não relacional MongoDB, aplicado ao sistema de Controle de Estudantes, baseado no repositório fornecido.

O objetivo é demonstrar consultas avançadas com:

1. Funções agregadas + GROUP BY (sumarização)

2. Joins entre coleções (junção)

Esses dois relatórios são obrigatórios para a entrega da disciplina de Banco de Dados.

O sistema utiliza uma modelagem híbrida, onde a coleção ESTUDANTE contém um array de cursos cursados, e a coleção CURSO guarda os metadados dos cursos.

3. RELATÓRIO A – Sumarização com Funções Agregadas (GROUP BY em MongoDB)

2.1 Objetivo

Gerar uma sumarização por curso contendo:

Quantidade total de alunos

Média das notas

Número de alunos aprovados

Taxa de aprovação (%)

2.2 Pipeline MongoDB (AGGREGATE)

```
Db.estudante.aggregate([
  { $unwind: "$cursos" },

  {
    $group: {
      _id: "$cursos.id_curso",
      Nome_curso: { $first: "$cursos.nome" },
      Num_alunos: { $sum: 1 },
      Media_nota: { $avg: "$cursos.nota" },
      Num_aprovados: {
        $sum: {
          $cond: [ { $gte: [ "$cursos.nota", 6.0 ] }, 1, 0 ]
        }
      }
    }
  }
])
```

```
}
},

{
  $project: {
    _id: 0,
    Id_curso: "$_id",
    Nome_curso: 1,
    Num_alunos: 1,
    Media_nota: { $round: ["$media_nota", 2] },
    Num_aprovados: 1,
    Taxa_aprovacao: {
      $round: [
        {
          $multiply: [
            { $divide: ["$num_aprovados", "$num_alunos"] },
            100
          ]
        },
        2
      ]
    }
  }
},

{ $sort: { num_alunos: -1 } }
]);
```

2.3 Exemplo de Resultado (Relatório)

ID Curso	Nome do Curso	Alunos	Média Aprovados	Taxa (%)
101	Matemática I 120	7.45 90	75.00%	
203	Programação Básica	95 6.88 60	63.16%	

2.4 Explicação da Consulta

\$unwind quebra o array cursos em documentos individuais

\$group funciona como o GROUP BY, agregando por curso

\$avg, \$sum, \$cond aplicam funções agregadas

\$project formata e calcula a taxa de aprovação

\$sort ordena os cursos por número de alunos

4. RELATÓRIO B – JOIN ENTRE COLEÇÕES (LOOKUP)

3.1 Objetivo

Construir um relatório que uma dados da coleção ESTUDANTE com a coleção CURSO, gerando uma lista completa com:

Estudante

Matrícula

Curso

Nota

Carga horária (vinda da coleção CURSO)

Código do curso

3.2 Pipeline MongoDB (JOIN)

```
Db.estudante.aggregate([
  { $unwind: "$cursos" },

  {
    $lookup: {
      From: "curso",
      localField: "cursos.id_curso",
      foreignField: "id_curso",
      as: "curso_meta"
    }
  }
])
```

```

    },

    { $unwind: { path: "$curso_meta", preserveNullAndEmptyArrays: true } },

    {
      $project: {
        _id: 0,
        Id_estudante: 1,
        Nome_estudante: "$nome",
        Matricula: 1,
        Id_curso: "$cursos.id_curso",
        Nome_curso: "$cursos.nome",
        Nota: "$cursos.nota",
        Curso_nome: "$curso_meta.nome",
        Carga_horaria: "$curso_meta.carga_horaria",
        Codigo: "$curso_meta.codigo"
      }
    },

    { $sort: { id_curso: 1, nome_estudante: 1 } }
  ]);

```

3.3 Exemplo de Relatório (Saída Tabular)

ID Estudante	Nome	Matrícula	Curso	Nota	Nome do Curso	C.H.

2001	Ana Silva	2021001	101	8.5	Matemática I 60h	MAT101
2033	João Pereira	2021005	101	5.0	Matemática I 60h	MAT101

3.4 Explicação da Consulta

\$lookup realiza o join entre ESTUDANTE e CURSO

\$unwind expande arrays tanto de cursos quanto do lookup

\$project seleciona os campos que serão exibidos como relatório

\$sort organiza a saída por curso e nome

5. Conclusão

Os dois relatórios obrigatórios foram produzidos:

❑ Relatório 1 — Agregações + “GROUP BY”

Utilizando \$unwind, \$group, \$avg, \$sum, \$cond.

❑ Relatório 2 — JOIN entre coleções

Utilizando \$lookup + \$project + \$sort.

As consultas foram escritas no padrão acadêmico, prontas para anexar ao trabalho, incluindo explicação, contexto e exemplos de relatório.
