

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

3º SEMESTRE ADS – CJOBDD2 - PROFº PAULO GIOVANI DE FARIA ZEFERINO

BRIAN GUSTAVO DE OLIVEIRA BRANDÃO

**ESTUDO DE CASO: SISTEMA PARA GERENCIAMENTO DE
HOTÉIS EM MONGODB**

CAMPOS DO JORDÃO

2025

RESUMO

Esta pesquisa trata do uso de bancos de dados que utilizam a abordagem NoSQL (isto é, não-relacionais) em aplicações reais, tendo especificamente como estudo de caso um sistema de gerenciamento de hotéis desenvolvido em trabalhos anteriores. Inicia com uma análise descritiva das funcionalidades de tais sistemas em modo geral, e finaliza com a implementação prática do modelo em MongoDB, o SGDB escolhido para a realização do projeto.

Palavras-Chave: banco de dados; NoSQL; sistema de software; gerenciamento de hotel.

1 INTRODUÇÃO

O avanço tecnológico tem transformado diversos setores da economia, e a gestão de hotéis não é exceção. A administração eficiente de um hotel exige o controle de uma grande quantidade de dados, que envolvem reservas, **check-ins** e **check-outs**, disponibilidade de quartos, preços, entre outros. Nesse contexto, os sistemas de gerenciamento se tornam ferramentas essenciais para garantir a organização, agilidade e precisão na gestão dessas informações.

Contudo, para que tais sistemas possam sequer funcionar, é necessário um banco de dados robusto, já que a área da hotelaria trabalha com um grande tráfego de pessoas, e subsequentemente, de informações a serem guardadas e processadas. Tendo isto em mente, este trabalho visa propor um projeto a ser utilizado para tal finalidade, utilizando a abordagem NoSQL para permitir maior escalabilidade.

No decorrer deste texto, serão abordados os principais conceitos e as melhores práticas na criação e implementação de um banco de dados não-relacionais para sistemas de gerenciamento de hotéis, tendo um enfoque especificamente na plataforma MongoDB.

1.1 Objetivos

Este trabalho tem por objetivo detalhar a criação de um banco de dados não-relacional para sistemas de gerenciamento de hotéis. Para a consecução deste objetivo, foram estabelecidos os objetivos específicos:

- Realizar uma investigação sobre as regras de negócio, além de demais requisitos, fundamentais para o funcionamento de um hotel;
- Encontrar um projeto de banco de dados tradicional que atenda as demandas aferidas no objetivo anterior;
- Adaptar o projeto para se adequar ao método escolhido e propô-lo como estudo de caso da viabilidade de tais sistemas.

1.2 Justificativa

Bancos de dados não-relacionais permitem muito maior escalabilidade e dinamismo do que os tradicionais, sendo assim de suma importância para gerentes de negócios que desejam evoluir suas empresas. Em Campos do Jordão especificamente, há um grande interesse no setor hoteleiro, um fator fundamental para fazer desta a área escolhida para o estudo de caso.

1.3 Aspectos Metodológicos

O presente estudo fez uso das pesquisas de natureza bibliográfica para o que remete à parte escrita e SGDBs (sistemas de gerenciamento de banco de dados) para a criação e teste do modelo físico, mais especificamente o MongoDB.

1.4 Aporte Teórico

As bases teóricas na qual esta pesquisa se fundamentou foram principalmente livros e demais fontes disponíveis na Internet, tanto pra parte do banco de dados em si quanto pras demais variáveis envolvidas no projeto a ser estudado como um todo.

2 PROJETO PROPOSTO

Nesta seção será apresentada detalhadamente a metodologia utilizada neste trabalho. Como o enfoque do projeto é a utilização dos bancos de dados NoSQL, o conceito do sistema de gerenciamento de hotéis foi adaptado a partir de trabalhos previamente existentes deste mesmo autor, no qual os modelos lógico, conceitual e físico, além de um dicionário de dados, já estão desenvolvidos.

NoSQL significa “Not Only SQL”, ou em português, “Não Apenas SQL”. Ao contrário dos bancos de dados relacionais, que são justamente conhecidos por utilizarem apenas a linguagem SQL através de ferramentas como MySQL e SQL Server, os não-relacionais mesclam essa abordagem tradicional com outras mais amplas.

Os quatro principais tipos de SGDBs NoSQL são os orientados à documentos, os orientados à grafos (como Neo4j), os orientados à colunas amplas (como Cassandra) e os orientados à pares chave-valor.

O SGDB escolhido foi o MongoDB, orientado à documentos, em decorrência de sua popularidade e relativa facilidade de uso em relação aos seus pares. As especificações mínimas para utilizar sua interface de usuário, o Compass, é ter o Windows 10 instalado, 4GB de RAM, 1GB de espaço em disco rígido e resolução de tela 1280x800; contudo, é relevante notar que o banco de dados em si possui escalabilidade horizontal, o que significa que ao invés de ser necessário aprimorar uma única máquina conforme a demanda aumenta, pode-se utilizar os recursos de várias máquinas.

3 AVALIAÇÃO

Nesta seção serão apresentados os resultados deste trabalho e uma discussão sobre eles.

3.1 Condução

Com o “esqueleto” do sistema anteriormente feito, já são conhecidas as regras de negócio e demais diretrizes necessárias para sua esquematização. O diagrama entidade-relacionamento a seguir representa o modelo conceitual do projeto, para servir de referência à implementação prática.

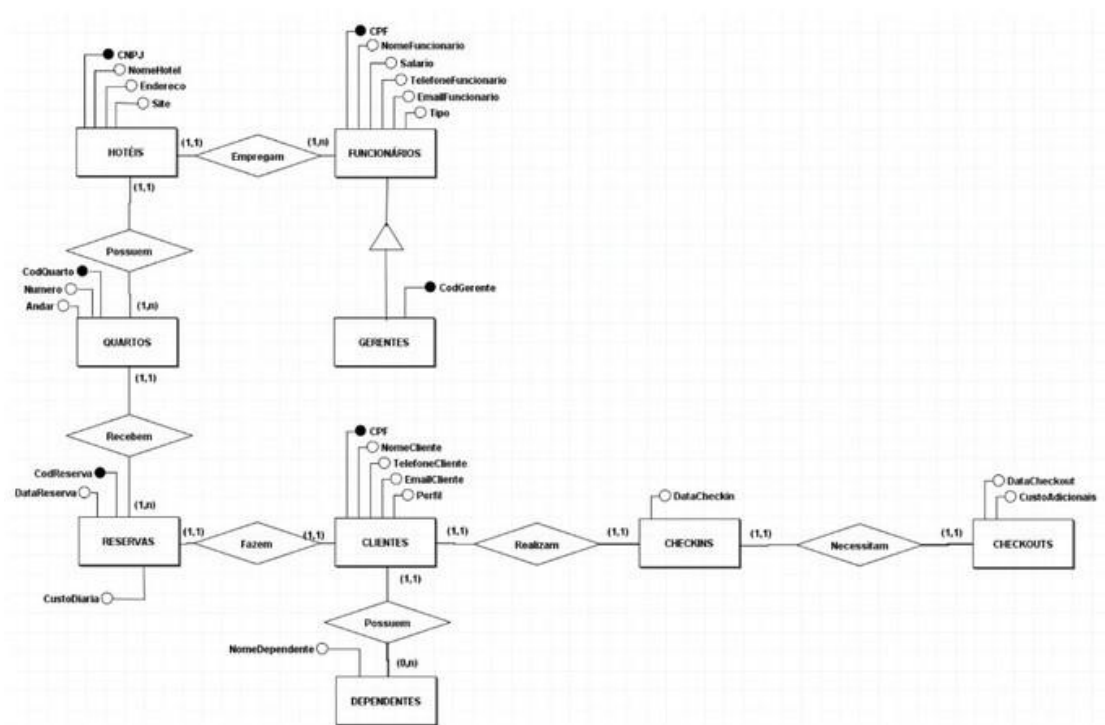


Figura 1 – Diagrama do projeto.

3.2 Resultados

A partir do diagrama acima, foi criado um banco de dados chamado SistemaHotel. Conforme a natureza do NoSQL, ao invés de tabelas, os dados são agrupados em diferentes coleções, que recebem os documentos JSON que formam a base do banco no caso do MongoDB. Foram criadas quatro: uma para os hotéis (e seus quartos), uma para os funcionários (e a especificação dos gerentes), uma para os clientes (e seus dependentes) e, por fim, uma para as reservas, além de seus respectivos *check-ins* e *check-outs*.

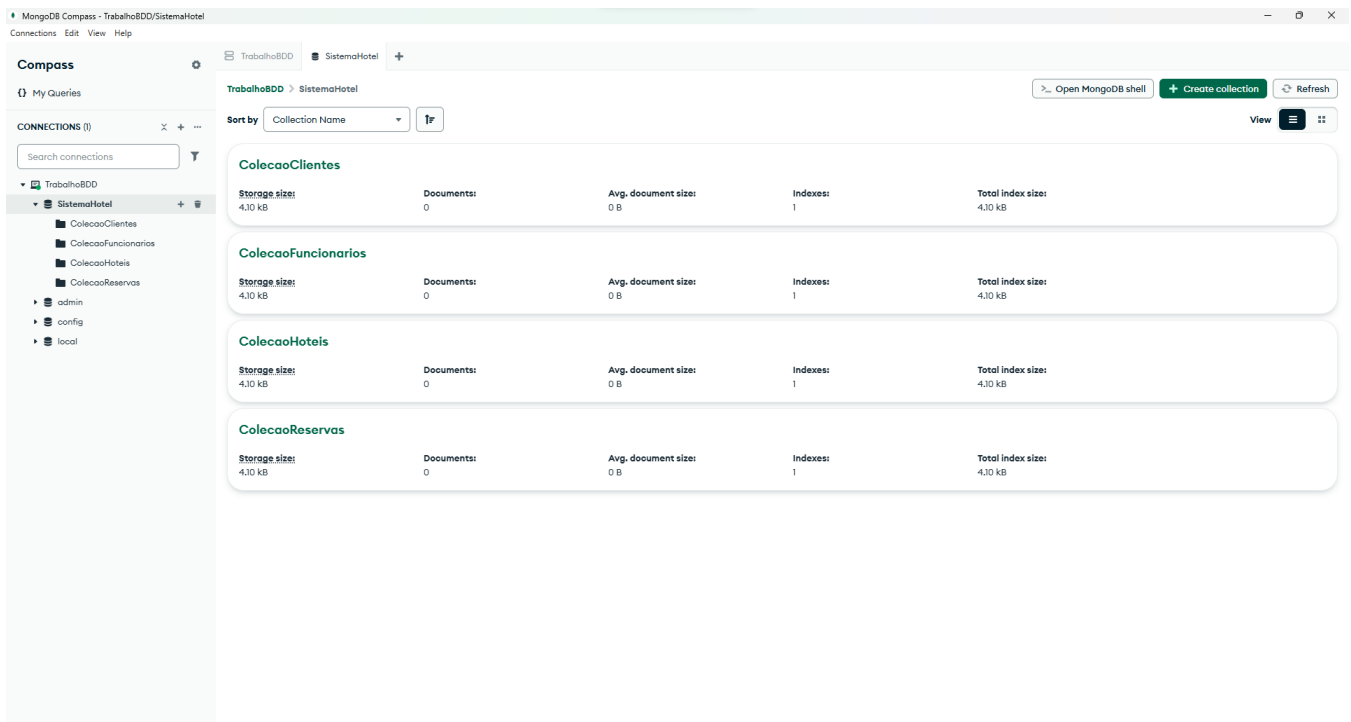


Figura 2 – Coleções do projeto.

O próximo passo foi inserir os documentos em suas respectivas coleções, com a figura a seguir apresentando um exemplo disso. Note a criação de uma ID nova para o objeto como um todo, e a forma como os *check-ins* e *check-outs*, que conceitualmente são tabelas separadas, se integram ao documento.

TrabalhoBDD > SistemaHotel > ColecaoReservas

Documents 1 Aggregations Schema Indexes 1 Validation

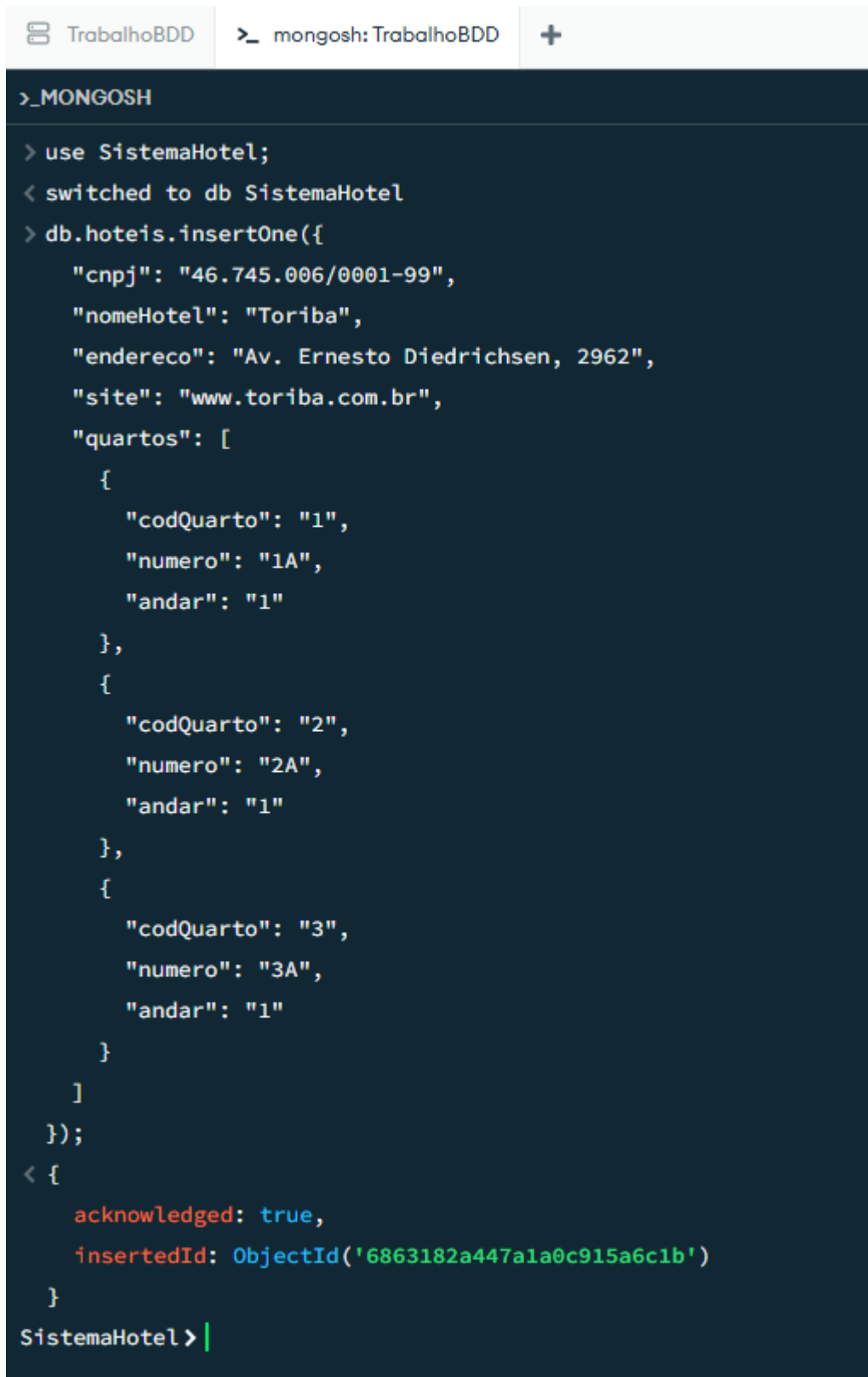
Type a query: { field: 'value' } or [Generate query](#)

+ ADD DATA EXPORT DATA UPDATE DELETE

```
{
  "_id": "<ObjectId>"
  "codReserva": "...",
  "dataReserva": "<ISODate>",
  "contaDiaria": "...",
  "cliente_id": "<ObjectId_do_cliente>",
  "hotel_id": "<ObjectId_do_hotel>"
  "quarto_info": Object
    "codQuarto": "...",
    "numero": "...",
    "andar": "...",
  "checkin": Object
    "dataCheckin": "<ISODate>",
  "checkout": Object
    "dataCheckout": "<ISODate>",
    "custosAdicionais": "...",
}
```

Figura 3 – Documento da coleção das reservas.

Para fazer a inserção de dados e consultas, necessita-se usar o *shell* do Compass, chamado de mongosh. A Figura 4 demonstra como foi o processo de adicionar um hotel e seus quartos, o que é um novo “objeto”.



```
>_MONGOSH
> use SistemaHotel;
< switched to db SistemaHotel
> db.hoteis.insertOne({
  "cnpj": "46.745.006/0001-99",
  "nomeHotel": "Toriba",
  "endereco": "Av. Ernesto Diedrichsen, 2962",
  "site": "www.toriba.com.br",
  "quartos": [
    {
      "codQuarto": "1",
      "numero": "1A",
      "andar": "1"
    },
    {
      "codQuarto": "2",
      "numero": "2A",
      "andar": "1"
    },
    {
      "codQuarto": "3",
      "numero": "3A",
      "andar": "1"
    }
  ]
});
< {
  acknowledged: true,
  insertedId: ObjectId('6863182a447a1a0c915a6c1b')
}
SistemaHotel> |
```


Figura 4 – Criação de um novo objeto.

Após povoar mais a coleção com diferentes objetos, é possível realizar consultas-exemplo, como pedir o retorno de hotéis que não possuam site:

```
> db.hoteis.find({ "site": null });
< {
  _id: ObjectId('68631c22447a1a0c915a6c1e'),
  cnpj: '21.277.360/0001-05',
  nomeHotel: 'Golden Park',
  endereco: 'Rod. Floriano Rodrigues Pinheiro, 2000',
  site: null,
  quartos: [
    {
      codQuarto: '10',
      numero: '1A',
      andar: '1'
    },
    {
      codQuarto: '11',
      numero: '1B',
      andar: '2'
    },
    {
      codQuarto: '12',
      numero: '2B',
      andar: '2'
    }
  ]
}
```

Figura 5 – Consulta aos objetos.

Como último teste de funcionalidade, foi necessário também verificar se era possível atualizar um objeto depois de sua criação. A partir do hotel que não tinha site, foi inventado um site fictício atribuído a ele, e depois se executou um comando “find” para achá-lo e exibi-lo na tela.

```

> db.hoteis.updateOne(
  {"cnpj": "21.277.360/0001-05"},
  { $set: { "site": "www.exemploupdate.com.br" } }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.hoteis.find({"cnpj": "21.277.360/0001-05"});
< {
  _id: ObjectId('68631c22447a1a0c915a6c1e'),
  cnpj: '21.277.360/0001-05',
  nomeHotel: 'Golden Park',
  endereco: 'Rod. Floriano Rodrigues Pinheiro, 2000',
  site: 'www.exemploupdate.com.br',
  quartos: [
    {
      codQuarto: '10',
      numero: '1A',
      andar: '1'
    },
    {
      codQuarto: '11',
      numero: '1B',
      andar: '2'
    },
    {
      codQuarto: '12',
      numero: '2B',
      andar: '2'
    }
  ]
}

```

Figura 6 – Atualização e exibição do objeto.

3.3 Discussão

Ao montar o diagrama, já notou-se que o caráter generalista do projeto não permitiu maior especialização, dado que diferentes hotéis podem fornecer uma vasta gama de serviços, como restaurantes, spas, recreações, dentre outros. Logo, ele atende as necessidades mais básicas, mas cada usuário do sistema iria precisar realizar suas próprias modificações a fim de atender adequadamente suas necessidades.

Tendo isto dito, as operações essenciais estão todas funcionais e a conversão para o modelo físico foi realizada com sucesso necessitando apenas de mínimos ajustes por questões de *software*, portanto o projeto ainda é uma excelente base, por mais que talvez careça de maiores detalhes em áreas específicas.

4 CONCLUSÃO

Retomando os objetivos inicialmente propostos, foi possível elaborar um sistema para gerenciamento de hotéis pensado para um banco de dados NoSQL e implementá-lo na prática com sucesso. Contudo, propor o resultado para clientes de forma que seja feito uso profissional é inviável no momento, dada a enorme variedade dos hotéis atualmente em operação.

Pesquisas futuras podem usar o modelo fornecido de base para montar sistemas mais específicos, com operações de maior complexidade, e assim potencialmente obter capacidade de adesão comercial, além de experimentar com outros SGDBs possuindo abordagens distintas (como o Neo4j ou o Cassandra).

REFERÊNCIAS

- CASTELLI, Geraldo. **Gestão hoteleira**. 2. ed. São Paulo: Saraiva, 2016. 512 p.
- VENISON, Peter. **Managing hotels**. Londres: Butterworth-Heinemann, 1983. 148 p.
- SADALAGE, Pramod; FOWLER, Martin. **NoSQL essencial: um guia conciso para o mundo emergente da persistência poliglota**. 1. ed. São Paulo: Novatec, 2013. 216 p.
- HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. 282 p.
- INSTITUTO BRASILEIRO DE TURISMO — EMBRATUR. **Regulação Geral dos Meios de Hospedagem**. 2002.
- ELMASRI, Ramez; NAVATHE, Shamkant. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Education do Brasil, 2011. 1152 p.
- ROB, Peter; CORONEL, Carlos. **Sistemas de banco de dados: projeto, implementação e gerenciamento**. São Paulo: Cengage Learning, 2011. 711 p.
- NETO, Dilson Jatahy Fonseca. **Turismo e hotelaria: comentários à legislação**. São Paulo: Liber Ars, 2016. 152 p.