

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

Luiz Guilherme dos Santos

CJ302492X

**PROJETO DE BANCO DE DADOS NOSQL PARA UM SISTEMA DE
GERENCIAMENTO DE RECURSOS HUMANOS**

CAMPOS DO JORDÃO

2025

RESUMO

Este relatório aborda a aplicação de bancos de dados não relacionais (NoSQL) no contexto do desenvolvimento de sistemas, com foco em um projeto de gerenciamento de recursos humanos. Inicialmente, são apresentados os conceitos fundamentais de bancos de dados NoSQL, destacando suas características, vantagens e os diversos modelos de dados existentes, como documentos, chave-valor, colunar e grafos. Em seguida, é realizada uma análise aprofundada do MongoDB, um SGBD NoSQL orientado a documentos, detalhando sua arquitetura, funcionamento e o modelo de dados flexível que o torna adequado para aplicações ágeis e escaláveis. Por fim, o trabalho descreve um estudo de caso hipotético de um sistema de gerenciamento de recursos humanos (SGRH) utilizando o MongoDB, demonstrando como os conceitos NoSQL podem ser aplicados na prática para otimizar o armazenamento e a recuperação de dados de funcionários, departamentos, projetos e históricos. A conclusão ressalta os benefícios da abordagem NoSQL para o SGRH proposto e sugere possíveis aprimoramentos.

Palavras-Chave: Banco de Dados NoSQL; MongoDB; Gerenciamento de Recursos Humanos; Modelo de Dados por Documentos; Escalabilidade.

ABSTRACT

This report addresses the application of NoSQL (Non-Relational) databases within the context of system development, focusing on a human resources management project. Initially, fundamental concepts of NoSQL databases are presented, highlighting their characteristics, advantages, and the various existing data models, such as document, key-value, columnar, and graph. Subsequently, an in-depth analysis of MongoDB, a document-oriented NoSQL DBMS, is conducted, detailing its architecture, operation, and the flexible data model that makes it suitable for agile and scalable applications. Finally, the paper describes a hypothetical case study of a Human Resources Management System (HRMS) using MongoDB, demonstrating how NoSQL concepts can be practically applied to optimize the storage and retrieval of employee, department, project, and historical data. The conclusion emphasizes the

benefits of the NoSQL approach for the proposed HRMS and suggests potential improvements.

Keywords: NoSQL Database; MongoDB; Human Resources Management; Document Data Model; Scalability.

INTRODUÇÃO

A crescente demanda por sistemas capazes de lidar com grandes volumes de dados variados (Big Data), em alta velocidade e com flexibilidade para se adaptar a requisitos em constante mudança, impulsionou o surgimento e a popularização dos bancos de dados não relacionais, conhecidos como NoSQL (Not Only SQL). Ao contrário dos bancos de dados relacionais tradicionais, que se baseiam em esquemas rígidos e tabelas predefinidas, os SGBDs NoSQL oferecem abordagens mais maleáveis para o armazenamento e manipulação de dados,

adequando-se a cenários que exigem alta performance, escalabilidade horizontal e agilidade no desenvolvimento.

Neste contexto, o presente trabalho tem como objetivo explorar o universo dos bancos de dados NoSQL, compreendendo suas características distintivas e os principais modelos de dados que os categorizam. Para aprofundar essa análise, será selecionado um SGBD NoSQL específico – o MongoDB – que se destaca por sua popularidade e aplicabilidade em diversas áreas. Serão detalhadas suas particularidades e o modelo de dados por documentos que o fundamenta. Como justificativa para este estudo, observa-se a crescente adoção de tecnologias NoSQL no mercado de trabalho e a necessidade de profissionais capacitados para projetar e implementar soluções eficientes com essas ferramentas. A metodologia empregada consistirá em pesquisa bibliográfica aprofundada sobre os conceitos de bancos de dados NoSQL e o MongoDB, seguida da concepção de um projeto hipotético para exemplificar sua aplicação prática. O aporte teórico será construído a partir de literaturas especializadas e documentações oficiais, visando fornecer uma base sólida para a compreensão do tema.

2 METODOLOGIA

A elaboração deste relatório seguiu uma metodologia de pesquisa exploratória e descritiva, combinando a revisão da literatura sobre bancos de dados não relacionais e o MongoDB com a proposição e análise de um estudo de caso prático hipotético. As etapas desenvolvidas foram as seguintes:

2.1 Considerações Iniciais sobre o Projeto

A escolha de um sistema de gerenciamento de banco de dados é um fator crítico para o sucesso de qualquer aplicação. Em contextos de desenvolvimento ágil e demandas dinâmicas, como a de um sistema de gerenciamento de recursos humanos em empresas com estruturas complexas e em constante evolução, a flexibilidade e a capacidade de escalar tornam-se requisitos primordiais. O presente trabalho buscou investigar a aplicabilidade e os benefícios dos bancos de dados NoSQL, em particular do MongoDB, para sistemas que exigem adaptabilidade na estrutura dos dados e alta performance para interações em tempo real.

2.2 Descrição do Modelo NoSQL

Os bancos de dados não relacionais, ou NoSQL (do inglês "Not Only SQL"), representam uma classe de sistemas de gerenciamento de dados projetados para lidar com desafios que os bancos de dados relacionais tradicionais (SQL) podem enfrentar, como grandes volumes de dados (Big Data), alta variabilidade de esquemas e a necessidade de escalabilidade horizontal massiva. Diferente do modelo tabular e rígido dos SGBDs relacionais, os NoSQL adotam diversas abordagens para o armazenamento e a recuperação de informações.

As características fundamentais dos bancos de dados NoSQL incluem:

- **Esquema Flexível (Schema-less):** Não exigem um esquema predefinido. Isso permite que diferentes registros (documentos, pares chave-valor, etc.) em uma mesma coleção possam ter estruturas distintas, facilitando a evolução e a adaptação do modelo de dados sem interrupções.
- **Escalabilidade Horizontal:** Projetados para distribuir a carga de trabalho e os dados entre múltiplos servidores (clusters), o que é conhecido como sharding. Isso permite aumentar a capacidade do sistema adicionando mais máquinas, em vez de depender de um único servidor mais potente.
- **Alta Disponibilidade:** Muitos SGBDs NoSQL são arquitetados para garantir a continuidade do serviço mesmo em caso de falhas de hardware ou software, através de mecanismos de replicação de dados em vários nós.
- **Performance Otimizada:** São frequentemente otimizados para tipos específicos de operações (leitura ou escrita) e modelos de dados, o que pode resultar em um desempenho superior para certas cargas de trabalho em comparação com bancos relacionais.

Dentro da vasta categoria NoSQL, os modelos de dados mais proeminentes são:

- **2.2.1 Orientado a Documentos:** Armazenam dados em documentos semi-estruturados, tipicamente no formato JSON (JavaScript Object Notation) ou BSON (Binary JSON). Cada documento é uma unidade independente e pode conter uma variedade de campos, incluindo arrays e documentos aninhados. São ideais para dados com estruturas variadas e que mudam frequentemente. Exemplos incluem MongoDB e CouchDB.
- **2.2.2 Orientado a Grafos:** Representam os dados como nós (entidades) e arestas (relacionamentos entre entidades). São otimizados para consultas que envolvem relacionamentos complexos entre entidades, como redes sociais, sistemas de recomendação e detecção de fraude. Exemplos: Neo4j e OrientDB.
- **2.2.3 Orientado a Colunas (Wide-Column Stores):** Armazenam os dados em colunas em vez de linhas. Isso os torna eficientes para lidar com grandes volumes de dados dispersos e para consultas analíticas que acessam um subconjunto específico de colunas. São comumente usados em cenários de Big Data, como análise de logs e séries temporais. Exemplos: Apache Cassandra e HBase.
- **2.2.4 Chave-Valor:** O modelo mais simples, onde cada item de dado é armazenado como um par de chave única e valor arbitrário. São extremamente rápidos para operações de leitura e escrita simples e são frequentemente utilizados para caching, gerenciar sessões de usuários ou filas de mensagens. Exemplos: Redis e Amazon DynamoDB.

2.3. Descrição do SGBD Escolhido: MongoDB

Para este estudo, o Sistema de Gerenciamento de Banco de Dados (SGBD) NoSQL selecionado é o MongoDB. Desenvolvido pela MongoDB Inc., é um dos bancos de dados orientados a documentos mais amplamente utilizados no mundo, destacando-se por sua popularidade, versatilidade e robustez para aplicações modernas.

2.3.1 MongoDB: Funcionamento e Características

O MongoDB armazena dados em documentos BSON (Binary JSON), que são uma representação binária e enriquecida do formato JSON. Essa escolha permite uma representação flexível e hierárquica dos dados, espelhando a estrutura de objetos em linguagens de programação. Os documentos são agrupados em coleções, que são análogas a tabelas em bancos de dados relacionais, mas sem a imposição de um esquema fixo para seus documentos.

As principais características que tornam o MongoDB uma escolha estratégica incluem:

- **Schema-less:** A ausência de um esquema rígido permite que a estrutura dos documentos dentro de uma coleção possa variar, facilitando a adaptação a mudanças de requisitos e a evolução do modelo de dados sem interrupções no serviço.
- **Escalabilidade Horizontal (Sharding):** Permite a distribuição de grandes volumes de dados e operações entre múltiplos servidores, aumentando a capacidade e a performance do sistema à medida que a demanda cresce.
- **Alta Disponibilidade (Replica Sets):** Garante a redundância dos dados através de conjuntos de réplicas, onde múltiplas cópias dos dados são mantidas em diferentes nós. Em caso de falha de um nó primário, um nó secundário é automaticamente promovido a primário, assegurando a continuidade operacional.
- **Linguagem de Consulta Rica (MongoDB Query Language - MQL):** Oferece um conjunto poderoso de operadores de consulta para realizar buscas complexas, incluindo filtragem, projeção, operações geoespaciais e de texto completo.
- **Framework de Agregação:** Permite processar e transformar dados de documentos para gerar relatórios e análises complexas, similar a operações de GROUP BY em SQL.
- **Índices:** Suporta a criação de diversos tipos de índices (simples, compostos, multichave, geoespaciais, de texto) para otimizar o desempenho das operações de leitura.
- **Ecossistema Amplo:** Possui drivers oficiais e comunitários para a maioria das linguagens de programação populares (Node.js, Python, Java, C#, PHP, Ruby, etc.), além de ferramentas como MongoDB Compass (interface gráfica) e MongoDB Atlas (plataforma em nuvem).

2.3.2 Operações Básicas com MongoDB

As operações fundamentais de CRUD (Create, Read, Update, Delete) no MongoDB são realizadas da seguinte forma:

Inserção (Create):

JavaScript

```
db.funcionarios.insertOne({  
  nome: "Ana Paula",  
  idade: 28,  
  cargo: "Analista de RH Júnior"  
});
```

Consulta com Filtro (Read):

JavaScript

```
db.funcionarios.find({ cargo: "Analista de RH Júnior", idade: { $lt: 30 } });
```

Atualização (Update):

JavaScript

```
db.funcionarios.updateOne(  
  { nome: "Ana Paula" },  
  { "$set": { cargo: "Analista de RH Pleno" } }  
);
```

Exclusão (Delete):

JavaScript

```
db.funcionarios.deleteOne({ nome: "Ana Paula" });
```

2.4 Estudo de Caso Selecionado: Sistema de Gerenciamento de Recursos Humanos (SGRH)

A aplicação prática descrita na seção de resultados refere-se a um projeto hipotético de Sistema de Gerenciamento de Recursos Humanos (SGRH) concebido e detalhado para este relatório. Este projeto foi desenvolvido com o objetivo de demonstrar a aplicabilidade e as vantagens do MongoDB na modelagem e gerenciamento de dados complexos em um domínio empresarial tradicionalmente associado a bancos de dados relacionais. A concepção de um SGRH como estudo de caso visa ilustrar como a flexibilidade e escalabilidade do MongoDB podem otimizar o armazenamento e a recuperação de informações de funcionários, departamentos, projetos e históricos, destacando os benefícios práticos dos conceitos estudados sobre bancos de dados NoSQL em um cenário realista de negócios.

3 Resultados Obtidos:

Nesta seção, será apresentado o detalhamento do projeto de dados para um Sistema de Gerenciamento de Recursos Humanos (SGRH) desenvolvido utilizando o MongoDB. A modelagem e as operações demonstram como o modelo de dados por documentos do MongoDB pode ser aplicado para armazenar e gerenciar informações complexas de funcionários, departamentos, projetos e outros dados relacionados a RH de forma eficiente e flexível.

3.1 Aplicação Prática: Projeto de Sistema de Gerenciamento de Recursos Humanos (SGRH) com MongoDB

O cenário para este projeto envolve uma empresa de médio a grande porte que necessita de um sistema robusto e flexível para gerenciar suas operações de RH. Este SGRH deve ser capaz de:

- Armazenar informações detalhadas de funcionários (dados pessoais, contato, cargo, salário, histórico de contratação, desempenho).
- Gerenciar departamentos e sua estrutura organizacional.
- Registrar projetos e a alocação de funcionários a esses projetos.
- Controlar histórico de promoções, treinamentos e avaliações.
- Armazenar documentos relacionados a cada funcionário (contratos, certificados).

- Permitir consultas rápidas e análises sobre a força de trabalho.

3.1.1 Modelagem de Dados no MongoDB para o SGRH

Ao invés de múltiplas tabelas normalizadas como em um banco relacional, o MongoDB permite modelar os dados de forma mais desnormalizada e focada no acesso à aplicação, consolidando informações relacionadas em um único documento ou em coleções que fazem sentido em conjunto.

Para este SGRH, propomos as seguintes coleções principais:

1. **funcionarios:** Armazena as informações principais de cada funcionário.
2. **departamentos:** Gerencia a estrutura dos departamentos da empresa.
3. **projetos:** Contém detalhes sobre os projetos em andamento ou concluídos.

3.1.1.1 Coleção funcionarios

Esta coleção seria a central para o SGRH. Cada documento representa um funcionário.

Exemplo de Estrutura de Documento funcionarios:

JSON

```
{
  "_id": ObjectId("60c72b2f9f1b2c3d4e5f6789"),
  "funcionarioid": "EMP001",
  "nomeCompleto": {
    "primeiroNome": "Mariana",
    "sobrenome": "Santos"
  },
  "dataNascimento": ISODate("1992-08-25T00:00:00Z"),
  "genero": "Feminino",
  "nacionalidade": "Brasileira",
  "documentos": {
    "cpf": "123.456.789-00",
```

```
"rg": "MG-12.345.678"

},

"contato": {

  "emailPrincipal": "mariana.santos@empresa.com",

  "telefoneCelular": "+5531998765432",

  "endereco": {

    "rua": "Rua das Laranjeiras, 45",

    "cidade": "São Paulo",

    "estado": "SP",

    "cep": "01234-567"

  }

},

"cargo": {

  "titulo": "Gerente de Projetos Sênior",

  "area": "Desenvolvimento de Software",

  "nivel": "Sênior"

},

"departamento": {

  "idDepartamento": ObjectId("60c72b3a9f1b2c3d4e5f6790"), // Referência ao _id do
departamento

  "nomeDepartamento": "Tecnologia e Inovação"

},

"salarioBase": 12000.00,

"dataContratacao": ISODate("2018-01-10T00:00:00Z"),

"statusEmprego": "Ativo",

"historicoCargos": [
```

```
{
  "titulo": "Analista de Sistemas Júnior",
  "dataInicio": ISODate("2018-01-10T00:00:00Z"),
  "dataFim": ISODate("2020-05-01T00:00:00Z")
},
{
  "titulo": "Analista de Sistemas Pleno",
  "dataInicio": ISODate("2020-05-01T00:00:00Z"),
  "dataFim": ISODate("2022-11-15T00:00:00Z")
}
],
"avaliacoesDesempenho": [
  {
    "ano": 2023,
    "pontuacao": 4.7,
    "comentarios": "Excelente liderança e entrega em projetos complexos."
  },
  {
    "ano": 2024,
    "pontuacao": 4.9,
    "comentarios": "Superou as expectativas na gestão da equipe e resultados."
  }
],
"treinamentos": [
  {
    "nome": "Certificação PMP",
```

```
"dataConclusao": ISODate("2022-03-20T00:00:00Z")
},
{
  "nome": "Gestão Ágil de Projetos",
  "dataConclusao": ISODate("2023-09-01T00:00:00Z")
}
],
"projetosAlocados": [
  {
    "idProjeto": ObjectId("60c72b4f9f1b2c3d4e5f6795"), // Referência ao _id do projeto
    "nomeProjeto": "Plataforma de E-commerce v2",
    "funcaoNoProjeto": "Gerente de Projeto",
    "dataInicioAlocacao": ISODate("2024-01-15T00:00:00Z"),
    "dataFimAlocacao": null
  }
],
"arquivosAnexos": [
  {
    "nomeArquivo": "ContratoMarianaSantos.pdf",
    "caminhoStorage": "/arquivos_rh/contratos/EMP001_Contrato.pdf",
    "tipo": "Contrato de Trabalho",
    "dataUpload": ISODate("2018-01-05T00:00:00Z")
  },
  {
    "nomeArquivo": "CertificadoPMP.pdf",
    "caminhoStorage": "/arquivos_rh/certificados/EMP001_PMP.pdf",
```

```

"tipo": "Certificado",

"dataUpload": ISODate("2022-03-25T00:00:00Z")

}

]

}

```

- **Vantagens Notáveis:** A capacidade de aninhar documentos (ex: nomeCompleto, contato, cargo), utilizar arrays de documentos (ex: historicoCargos, avaliacoesDesempenho, projetosAlocados) e incorporar arrays de valores simples permite armazenar toda a informação relevante de um funcionário em um único lugar. Isso otimiza as leituras, pois a maioria dos dados necessários para exibir um perfil completo de funcionário pode ser recuperada com uma única consulta.
- **Referências Flexíveis:** Para dados que precisam ser consultados e atualizados independentemente ou que são compartilhados por muitas entidades (como departamentos e projetos), optou-se por "denormalização de referência" (ou "referência manual"), onde o `_id` do documento referenciado é armazenado, e o nome é duplicado para facilitar as consultas.

3.1.1.2 Coleção departamentos

Esta coleção armazenaria a estrutura dos departamentos.

Exemplo de Estrutura de Documento departamentos:

JSON

```

{

  "_id": ObjectId("60c72b3a9f1b2c3d4e5f6790"),

  "nome": "Tecnologia e Inovação",

  "codigo": "TI001",

  "localizacao": "Edifício Alfa, 5º Andar",

  "telefone": "(11) 3333-4444",

  "gerente": {

    "idGerente": ObjectId("60c72b2f9f1b2c3d4e5f678A"), // Referência a um funcionário

```

```
"nomeGerente": "João Pereira"

},

"numeroFuncionariosAtivos": 150,

"subDepartamentos": [

{

  "nome": "Desenvolvimento Frontend",

  "codigo": "TI001-DF",

  "gerente": {

    "idGerente": ObjectId("60c72b2f9f1b2c3d4e5f6789"), // Mariana Santos, por exemplo

    "nomeGerente": "Mariana Santos"

  }

},

{

  "nome": "Desenvolvimento Backend",

  "codigo": "TI001-DB"

},

{

  "nome": "Infraestrutura e DevOps",

  "codigo": "TI001-ID"

}

]

}
```

3.1.1.3 Coleção projetos

Esta coleção detalha os projetos da empresa.

Exemplo de Estrutura de Documento projetos:

JSON

```
{

  "_id": ObjectId("60c72b4f9f1b2c3d4e5f6795"),

  "nomeProjeto": "Plataforma de E-commerce v2",

  "codigoProjeto": "ECOMM-V2",

  "descricao": "Desenvolvimento da próxima geração da plataforma de e-commerce da empresa, com novas funcionalidades de IA.",

  "dataInicio": ISODate("2024-01-01T00:00:00Z"),

  "dataPrevisaoFim": ISODate("2025-12-31T00:00:00Z"),

  "status": "Em Andamento",

  "prioridade": "Alta",

  "equipe": [

    {

      "idFuncionario": ObjectId("60c72b2f9f1b2c3d4e5f6789"), // Mariana Santos

      "nomeFuncionario": "Mariana Santos",

      "papel": "Gerente de Projeto"

    },

    {

      "idFuncionario": ObjectId("60c72b2f9f1b2c3d4e5f678B"), // Outro funcionário

      "nomeFuncionario": "Lucas Rocha",

      "papel": "Desenvolvedor Backend Líder"

    },

    {

      "idFuncionario": ObjectId("60c72b2f9f1b2c3d4e5f678C"), // Terceiro funcionário

      "nomeFuncionario": "Isabela Costa",

      "papel": "Designer UX/UI Sênior"

    }

  ]

}
```

```
],  
"tecnologiasUtilizadas": ["Node.js", "React", "MongoDB", "Kafka", "AWS Lambda", "Python"],  
"custoEstimado": 750000.00,  
"progressoPercentual": 45  
}
```

3.1.2 Operações Comuns e Consultas no MongoDB para o SGRH

Com esta modelagem, operações comuns no SGRH seriam altamente eficientes:

Buscar informações de um funcionário: Uma única consulta na coleção `funcionarios` retorna todos os dados pessoais, de contato, cargo, histórico, avaliações, projetos alocados e arquivos anexos.

JavaScript

```
db.funcionarios.findOne({ "funcionarioId": "EMP001" })
```

Listar todos os funcionários de um departamento:

JavaScript

```
db.funcionarios.find({ "departamento.nomeDepartamento": "Tecnologia e Inovação" })
```

Buscar funcionários com um cargo específico e salário acima de um valor:

JavaScript

```
db.funcionarios.find(  
  "cargo.titulo": "Gerente de Projetos Sênior",  
  "salarioBase": { "$gt": 10000 }  
)
```

Atualizar o cargo de um funcionário:

JavaScript

```
db.funcionarios.updateOne(  
  { "funcionarioId": "EMP001" },  
  {
```



```
"$set": { "cargo.titulo": "Diretor de TI", "cargo.nivel": "Executivo" },

"$push": {

  "historicoCargos": {

    "titulo": "Gerente de Projetos Sênior",

    "dataInicio": ISODate("2022-11-15T00:00:00Z"),

    "dataFim": ISODate("2025-06-30T00:00:00Z") // Data atual

  }

}

}
```

Adicionar um novo treinamento ao histórico de um funcionário:

JavaScript

```
db.funcionarios.updateOne(

  { "funcionarioId": "EMP001" },

  { "$push": {

    "treinamentos": {

      "nome": "Liderança Estratégica",

      "dataConclusao": ISODate("2025-05-10T00:00:00Z")

    }

  }

}
```

Consultar projetos em andamento e suas equipes:

JavaScript

```
db.projetos.find(

  { "status": "Em Andamento" },
```

```
{ "nomeProjeto": 1, "equipe.nomeFuncionario": 1, "equipe.papel": 1, "progressoPercentual": 1 }  
)
```

3.1.3 Benefícios da Abordagem MongoDB no SGRH

- **Agilidade no Desenvolvimento:** A flexibilidade do esquema do MongoDB permite que a estrutura dos dados seja modificada e evoluída sem a necessidade de migrações complexas de esquema, o que é crucial em projetos ágeis.
- **Performance para Leitura:** Ao aninhar dados relacionados, a maioria das operações de leitura de perfis completos de funcionários torna-se muito rápida, pois evita múltiplos JOINS que seriam necessários em um banco relacional.
- **Escalabilidade:** Em um cenário onde a empresa cresce e o volume de funcionários e dados aumenta, o MongoDB pode escalar horizontalmente facilmente, distribuindo a carga entre diversos servidores sem a complexidade de sharding manual de bancos relacionais.
- **Armazenamento de Dados Complexos:** A capacidade de armazenar arrays de documentos e documentos aninhados é ideal para representar dados como histórico de cargos, avaliações de desempenho e anexos.
- **Desenvolvimento Intuitivo:** A representação dos dados em JSON/BSON é natural para desenvolvedores que trabalham com linguagens modernas, reduzindo a necessidade de ORMs e simplificando o código de manipulação de dados.

4 CONCLUSÃO

A adoção de bancos de dados não relacionais representa uma transformação essencial no modo como os sistemas computacionais modernos lidam com o armazenamento e a manipulação de dados. Com o crescimento exponencial da informação, a diversidade dos formatos de dados e a necessidade de respostas ágeis, os modelos tradicionais relacionais muitas vezes se mostram limitados. Nesse contexto, os bancos NoSQL — com destaque para o MongoDB — tornaram-se protagonistas.

O MongoDB, como banco de dados orientado a documentos, oferece grande flexibilidade estrutural, permitindo que desenvolvedores construam aplicações com maior agilidade e adaptação às mudanças contínuas de requisitos. Sua capacidade de lidar com documentos aninhados, consultas avançadas e escalabilidade horizontal natural o torna ideal para sistemas centrados no usuário, com dados ricos e variados.

O estudo de caso hipotético de um Sistema de Gerenciamento de Recursos Humanos (SGRH) com MongoDB ilustrou de forma prática como a desnormalização estratégica e o uso de documentos aninhados e arrays podem otimizar as operações de leitura e simplificar a lógica

da aplicação, resultando em um sistema mais ágil e performático. A modelagem proposta permitiu consolidar dados que, em um modelo relacional, estariam dispersos em múltiplas tabelas, reduzindo a complexidade das consultas e facilitando a manutenção do código.

Em suma, a escolha do MongoDB para um SGRH, especialmente em cenários de empresas com rápido crescimento e requisitos de dados em constante evolução, oferece uma alternativa robusta e eficiente aos SGBDs relacionais. Ele possibilita um desenvolvimento mais rápido e iterativo, além de prover uma arquitetura resiliente e escalável.

4.1.Possíveis Melhorias no Projeto:

Para aprimorar o projeto do SGRH com MongoDB, sugere-se as seguintes melhorias:

1. **Implementação de Autenticação e Autorização:** Integrar um sistema de segurança robusto para controle de acesso aos dados, definindo diferentes níveis de permissão para usuários (RH, gerentes, funcionários).
2. **Uso de GridFS para Armazenamento de Documentos Grandes:** Para arquivos maiores, como currículos extensos ou vídeos de treinamento, o GridFS do MongoDB pode ser explorado para armazenamento eficiente, em vez de apenas armazenar o caminho do arquivo em um *storage* externo.
3. **Integração com Ferramentas de Análise de Dados:** Conectar o MongoDB a ferramentas de Business Intelligence (BI) e análise de dados (ex: Power BI, Tableau, ou frameworks como Apache Spark) para extrair *insights* valiosos sobre o capital humano da empresa.
4. **Otimização de Índices:** Realizar um estudo aprofundado sobre os padrões de consulta mais frequentes para criar índices otimizados, incluindo índices compostos e de texto completo, para melhorar ainda mais o desempenho das buscas.
5. **Implementação de Transações Multi-Documento:** Para operações que exigem atomicidade entre múltiplos documentos (ex: transferir um funcionário de departamento e atualizar métricas do departamento), explorar as transações ACID disponíveis no MongoDB a partir da versão 4.0.
6. **Criação de Views Materializadas:** Para consultas complexas ou relatórios que exigem agregação de dados de várias coleções, criar "views materializadas" (coleções pré-agregadas) para acelerar a geração de relatórios.
7. **Adoção de Padrões de Design para MongoDB:** Estudar e aplicar padrões de design específicos para o MongoDB, como *Embedding*, *Referencing*, *Bucket Pattern*, entre outros, para otimizar ainda mais a modelagem de dados para casos de uso específicos.
8. **Monitoramento e Otimização de Performance:** Implementar ferramentas de monitoramento de performance para o cluster MongoDB (ex: MongoDB Atlas Monitoring, Prometheus/Grafana) para identificar gargalos e otimizar a infraestrutura.

Referências Bibliográficas:

D'SOUZA, Dan; CHAUDHURI, Arun. **MongoDB: The Definitive Guide**. 3. ed. O'Reilly Media, 2019.
MONGODB INC. **MongoDB Documentation**. Disponível em: <https://www.mongodb.com/docs>

PRAMOD, J. S. **Professional NoSQL**. Wrox, 2012.

SADALAGE, Martin; FOWLER, Scott. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**. Addison-Wesley Professional, 2012.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. 6. ed. Pearson, 2013.