

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

LUIGI WASCHENSHIKY LUZ

ESTUDO DE CASO COM MONGODB

**CAMPOS DO JORDÃO
2024**

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

ESTUDO DE CASO COM MONGODB

Entrega do projeto de NoSQL da disciplina de Banco de Dados II apresentado ao Instituto Federal de São Paulo (IFSP), em cumprimento a exigência da disciplina de Banco de Dados II, do curso de Análise e Desenvolvimento de Sistemas.

PROFESSOR: Paulo Giovani de Faria Zeferrine

**CAMPOS DO JORDÃO
2024**

RESUMO

Neste estudo de caso, será explorada a utilização do MongoDB, um banco de dados não relacional, no desenvolvimento de um sistema. Primeiramente, discutiremos o conceito de bancos de dados não relacionais, seus modelos dentro dessa classificação e suas principais aplicações. Em seguida, focaremos no MongoDB, sua implementação no estudo de caso e os resultados obtidos, culminando em uma conclusão sobre a eficácia do MongoDB na resolução do problema proposto. A utilização do MongoDB será aplicada em um sistema de gerenciamento ferroviário já criado de autoria própria, demonstrando como as características deste banco de dados podem ser aproveitadas para aprimorar a eficiência e a flexibilidade do sistema.

Palavras-Chave: Estudo de Caso; MongoDB; Banco de Dados; Não Relacionais; Flexibilidade;

ABSTRACT

In this case study, the use of MongoDB, a non-relational database, will be explored in the development of a system. First, we will discuss the concept of non-relational databases, their models within this classification and their main applications. Next, we will focus on MongoDB, its implementation in the case study and the results obtained, culminating in a conclusion about the effectiveness of MongoDB in solving the proposed problem. The use of MongoDB will be applied to an already created railway management system, demonstrating how the characteristics of this database can be used to improve the efficiency and flexibility of the system.

Keywords: Case study; MongoDB; Database; Non-Relational; Flexibility;

LISTA DE ALGORITMOS

ALGORITMO 1 – Comparação entre SQL e BSON	16
ALGORITMO 2 – Sistema de Gerenciamento Ferroviário	18

LISTA DE SIGLAS

IFSP	Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
SQL	Structured Query Language
BSON	Binary JavaScript Object Notation
NoSQL	No only Structured Query Language
CRUD	Create, Read, Update e Delete
JSON	JavaScript Object Notation
SGBD	Sistema Gerenciador de Banco de Dados
ACID	Atomicidade, Consistência, Isolamento e Durabilidade

SUMÁRIO

1	INTRODUÇÃO	9
1.1	<i>Objetivos</i>	9
1.2	<i>Justificativa</i>	10
1.3	<i>Aspectos Metodológicos</i>	10
1.4	<i>Aporte Teórico</i>	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	<i>Documentação do MongoDB</i>	11
2.2	Big Data: Técnicas e tecnologias para extração de valor dos dados	11
2.3	NoSQL como armazenar os dados de uma aplicação moderna	11
3	PROJETO PROPOSTO	12
3.1	<i>Metodologias de Desenvolvimento</i>	12
3.2	<i>Considerações Iniciais</i>	12
3.3	<i>O que são Banco de Dados Não Relacionais</i>	13
3.4	<i>Tipo de Banco de Dados que operam no modelo NoSQL</i>	13
3.5	<i>Descrição do SGBD utilizado no estudo de caso</i>	14
3.6	<i>Características do MongoDB</i>	14
4	RESULTADOS OBTIDOS	16
4.1	RESULTADOS	16
4.2	COMPARAÇÃO ENTRE MODELOS RELACIONAL E NÃO RELACIONAL	16
4.3	SISTEMA DESENVOLVIDO COM O MONGODB	17
5	CONCLUSÃO	20

1 INTRODUÇÃO

Os bancos de dados desempenham um papel fundamental no armazenamento e gerenciamento de informações em sistemas de software. Eles são a base para muitas aplicações, desde sistemas de gerenciamento de estoque até redes sociais. Existem dois tipos principais de bancos de dados: relacionais e não relacionais. Os bancos de dados relacionais, populares e tradicionalmente mais utilizados, organizam os dados em tabelas e colunas, permitindo relações estruturadas entre diferentes conjuntos de dados.

No entanto, os bancos de dados não relacionais, que são relativamente novos, vêm ganhando força devido à sua flexibilidade e escalabilidade. Diferente dos relacionais, os bancos de dados não relacionais não seguem um formato de tabelas, permitindo uma estrutura de dados mais dinâmica e adaptável. Neste trabalho, será explorada a utilização do MongoDB, um banco de dados não relacional.

1.1 Objetivos

Este trabalho tem por objetivo conhecer, se aprofundar e entender as principais ferramentas, aplicações e características do MongoDB a partir da criação de um sistema de gerenciamento ferroviário. Através desse estudo, será possível identificar os pontos positivos e negativos do MongoDB, avaliando sua eficácia e adequação para este tipo de aplicação.

Para a consecução deste objetivo foram estabelecidos os objetivos específicos:

- **Explorar as Ferramentas e Funcionalidades do MongoDB:** Investigar e compreender as principais ferramentas e funcionalidades oferecidas pelo MongoDB, incluindo sua estrutura de documentos, índices, agregações e recursos de replicação e sharding.
- **Identificar Vantagens e Desvantagens do MongoDB:** Comparar o MongoDB com bancos de dados relacionais e outros bancos de dados não relacionais, destacando suas vantagens e desvantagens específicas para o contexto de um sistema de gerenciamento ferroviário.

- **Desenvolver um Sistema de Gerenciamento Ferroviário:** Projetar e implementar um sistema de gerenciamento ferroviário utilizando MongoDB.

1.2 Justificativa

A utilização do MongoDB é justificada por sua flexibilidade, escalabilidade e desempenho. O modelo de dados baseado em documentos do MongoDB permite armazenar informações complexas sem a rigidez dos esquemas relacionais. Sua escalabilidade horizontal facilita o crescimento do sistema, enquanto seus recursos de indexação e agregação garantem consultas rápidas e eficientes. Além disso, a alta disponibilidade e replicação do MongoDB asseguram a continuidade e resiliência do sistema, tornando-o ideal para atender às necessidades de um gerenciamento ferroviário robusto e eficiente.

1.3 Aspectos Metodológicos

O presente estudo adotou uma abordagem metodológica que integrou duas principais metodologias: a exploratória, a bibliográfica.

1.4 Aporte Teórico

A pesquisa fundamentou-se em algumas bases teóricas essenciais para o desenvolvimento do sistema de registro e gerenciamento ferroviário:

- Documentação do MongoDB
- Big Data: Técnicas e tecnologias para extração de valor dos dados
- **NoSQL** Como armazenar os dados de uma aplicação moderna

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção será apresentada uma revisão de textos, artigos, livros, periódicos, enfim, todo o material pertinente à revisão da literatura que será utilizada no desenvolvimento do trabalho.

2.1 Documentação do MongoDB

A documentação oficial do MongoDB fornece diretrizes detalhadas sobre a modelagem de dados, operações CRUD, agregações e uso avançado de índices. Este recurso é essencial para garantir a implementação eficaz de um banco de dados NoSQL que suporte as complexas necessidades do sistema ferroviário integrado.

2.2 Big Data: Técnicas e tecnologias para extração de valor dos dados

O livro cobre as principais fases de um projeto de Big Data, desde a captura, armazenamento, processamento, análise até a visualização de dados. Embora não seja especificamente sobre NoSQL, o livro contém um capítulo que aborda este tópico de forma aprofundada, destacando sua relevância no contexto de Big Data.

2.3 NoSQL como armazenar os dados de uma aplicação moderna

Neste livro, é apresentado os bancos de dados não relacionais, discutindo suas vantagens e desvantagens, o ajudando a escolher a melhor opção para determinada situação. Através de um exemplo real, é aprendido a usar bancos de dados NoSQL na prática, aplicando soluções mais elegantes, práticas, performáticas e escaláveis, ampliando suas opções na criação de projetos.

3 PROJETO PROPOSTO

3.1 Metodologias de Desenvolvimento

A metodologia bibliográfica foi fundamental para revisar e analisar a literatura existente relacionada ao banco de dados não relacional, utilizando como base a documentação do MongoDB, bem como os livros "Big Data: Técnicas e tecnologias para extração de valor dos dados" e "NoSQL: Como armazenar os dados de uma aplicação moderna". Essa revisão da literatura contribuiu para a fundamentação teórica do projeto, permitindo uma contextualização sólida.

A metodologia qualitativa foi aplicada durante a criação e implementação do sistema utilizando o MongoDB. Após a implementação, foi realizada uma análise detalhada, permitindo identificar melhorias e ajustes necessários. Esta abordagem mista garantiu uma base sólida e prática para o desenvolvimento eficiente e adaptável do sistema ferroviário integrado.

3.2 Considerações Iniciais

Este projeto começou a partir do sistema de gerenciamento ferroviário de autoria própria, originalmente utilizando um banco de dados relacional. A conversão para um banco de dados não relacional, como o MongoDB, foi realizada com o objetivo de aprofundar o conhecimento, ganhar familiaridade com a nova tecnologia, utilizar suas ferramentas e comparar as diferenças entre um modelo relacional e um não relacional. Este sistema relacional foi muito bem fundamentado em suas regras de negócios, objetivos e captação de requisitos, servindo como um excelente exemplo para a implementação no MongoDB.

3.3 O que são Banco de Dados não relacionais

Banco de dados NoSQL, ou "Not Only SQL", é uma categoria de sistemas de gerenciamento de banco de dados que difere dos tradicionais bancos de dados relacionais. Enquanto os bancos de dados relacionais seguem um modelo de dados tabular com esquemas fixos, os bancos de dados NoSQL oferecem uma abordagem mais flexível e escalável para armazenar e recuperar dados. Um banco de dados não relacional, ou NoSQL, difere dos tradicionais bancos de dados relacionais, pois não utiliza um esquema de tabela fixo. Em vez disso, adota uma estrutura mais flexível, como documentos JSON, pares chave-valor ou grafos, permitindo armazenar informações específicas de forma eficiente.

A principal vantagem do NoSQL é a liberdade que oferece para construir sistemas modernos, pois permite armazenar dados de forma mais adequada às necessidades específicas de cada aplicação. Isso é especialmente útil em cenários como redes sociais, onde é possível vincular documentos, como fotos, diretamente aos perfis dos usuários, em vez de exigir conexões complexas com toda a rede de contatos. Outra característica importante é que, ao contrário dos bancos de dados relacionais que utilizam a linguagem SQL para manipulação de dados, os bancos de dados NoSQL são acessados e gerenciados através de linguagens de programação, como JavaScript, o que pode aumentar a produtividade dos desenvolvedores.

3.4 Tipos de Banco de Dados que operam no modelo NoSQL

Bancos de Dados de Documentos: Os dados são armazenados em documentos semelhantes a JSON ou BSON (Binary JSON). Cada documento pode conter informações heterogêneas e não precisa seguir um esquema rígido. Exemplos populares são o MongoDB e o Couchbase.

Bancos de Dados de Chave-Valor: Os dados são armazenados como pares de chave e valor, onde a chave é um identificador único para o valor associado. São altamente eficientes em termos de acesso rápido aos dados. Exemplos incluem o Redis e o Amazon DynamoDB.

Bancos de Dados de Família de Colunas: Organizam os dados em famílias de colunas, permitindo que cada linha contenha diferentes colunas. São adequados para

cenários em que os dados têm estruturas complexas. O Cassandra é um exemplo popular dessa categoria.

Bancos de Dados de Grafos: São otimizados para armazenar e recuperar dados interconectados. Eles utilizam a teoria dos grafos para representar e processar relações entre os dados. O Neo4j é um exemplo conhecido de banco de dados de grafos.

3.5 Descrição do SGBD utilizado no estudo de caso

No estudo de caso, utilizamos o MongoDB, um Sistema de Gerenciamento de Banco de Dados (SGBD) não relacional, de código aberto e orientado a documentos para processar e armazenar diversos tipos de dados. Como uma solução NoSQL, o MongoDB não exige um sistema de gerenciamento de banco de dados relacional, oferecendo um modelo de armazenamento de dados elástico que permite aos usuários armazenar e consultar diversos tipos de dados com facilidade. Isso simplifica o gerenciamento do banco de dados para os desenvolvedores e cria um ambiente altamente escalável para aplicativos e serviços multiplataforma.

3.6 Características do MongoDB

Modelo de Dados Flexível: Utilizamos o modelo de dados baseado em documentos do MongoDB, onde os dados são armazenados em documentos BSON, permitindo estruturas complexas sem a necessidade de um esquema rígido, muito semelhante ao JSON a principal diferença é que possuem extensões nos quais permitem a representação de dados derivados, por exemplo um tipo Date, ObjectId .

Escalabilidade Horizontal: Implementamos a escalabilidade horizontal, distribuindo o banco de dados em vários servidores para aumentar a capacidade de armazenamento e processamento conforme necessário, usando o conceito de sharding para dividir os dados entre diferentes nós.

Alta Performance: Tiramos proveito da alta performance do MongoDB em operações de leitura e escrita, utilizando indexação para acelerar consultas e uma arquitetura interna eficiente para gerenciar grandes volumes de dados.

Replicação: A replicação automática do MongoDB é utilizada para espelhar dados em vários servidores, melhorando a disponibilidade do sistema e proporcionando tolerância a falhas, permitindo a recuperação de dados de réplicas em caso de falha.

Consultas Avançadas: As consultas avançadas do MongoDB, que suportam operações de filtragem, projeção, ordenação e agregação, permitindo extrair e manipular dados de maneira eficiente.

Indexação: Criação de índices em campos específicos dos documentos para acelerar consultas e melhorar o desempenho geral do sistema.

Transações ACID: Em operações críticas, utilizamos o suporte a transações ACID do MongoDB, garantindo atomicidade, consistência, isolamento e durabilidade.

4 RESULTADOS OBTIDOS

Nesta seção serão apresentados os resultados deste trabalho e uma descrição sobre o modelo criado em MongoDB.

4.1 Resultados

O resultado deste trabalho, a partir da análise e estudos sobre os modelos NoSQL, especificamente o MongoDB, culminou no desenvolvimento de um sistema de gerenciamento ferroviário. Este sistema aproveitou as vantagens do Mon-

goDB, como maior flexibilidade na modelagem de dados, escalabilidade e eficiência no processamento de grandes volumes de dados.

4.2 Comparação entre Modelos Relacional e Não Relacional

Para melhor compreensão, neste capítulo, abordamos a criação da entidade "Viagens" utilizando dois paradigmas distintos de banco de dados: o modelo relacional e o modelo não relacional (NoSQL) com BSON, especificamente no MongoDB. A seguir, apresentamos as duas implementações para facilitar a compreensão das diferenças estruturais e funcionais entre ambos.

<pre> /*Viagens*/ //criando uma collection de forma implicita db.Viagens.insertOne(["id": 1, "DataSaida": "2024-09-24", "DataChega": "2024-09-25", "HorarioSaida": "01:00:00", "HorarioChegada": "05:00:00", "Estacoes": { "Saida": { "Nome": "Estação da Luz", "Endereco": { "Rua": "Newton de Vasconcellos", "Bairro": "Jardim Esperança", "Numero": 27 }, "Cidade_id": 2 }, "Chegada": { "Nome": "Estação da Mantiqueira", "Endereco": { "Rua": "José Fragozzo", "Bairro": "Jardim Inglaterra", "Numero": 156 }, "Cidade_id": 3 }, "CidadeDestino_id": 2, "Trens_id": 3 }]) </pre>	<pre> ----- -- Viagens ----- CREATE TABLE Viagens (ID_Viagem INT NOT NULL, Data_Saida DATE NOT NULL, Data_Chegada DATE NOT NULL, Horario_de_Partida TIME NOT NULL, Horario_de_Chegada TIME NOT NULL, ID_Estacao_Saida INT NOT NULL, ID_Estacao_Chegada INT NOT NULL, ID_Cidade_Destino INT NOT NULL, ID_Trem INT NOT NULL, PRIMARY KEY (ID_Viagem), FOREIGN KEY (ID_Trem) REFERENCES Trens (ID_Trem), FOREIGN KEY (ID_Cidade_Destino) REFERENCES Cidades (ID_Cidade), FOREIGN KEY (ID_Estacao_Saida) REFERENCES Estacoes (ID_Estacao), FOREIGN KEY (ID_Estacao_Chegada) REFERENCES Estacoes (ID_Esta- cao)); </pre>
---	---

Algoritmo 1: Comparação entre SQL e BSON

No modelo relacional, a tabela "Viagens" é definida com várias colunas representando os atributos da viagem, como data de saída, data de chegada, horários, e chaves

estrangeiras para referenciar outras tabelas relacionadas. A estrutura segue uma rigidez típica dos bancos de dados relacionais, com restrições de integridade e relações bem definidas. No modelo não relacional, a entidade "Viagens" é representada como um documento BSON. Esta abordagem permite maior flexibilidade na estrutura dos dados, incluindo subdocumentos e arrays, como demonstrado na inclusão de informações detalhadas sobre as estações de saída e chegada diretamente no documento da viagem e também existe a referência para Cidades de Partida e Chegada no qual não são uma chave estrangeira, mas fazem referência a entidade Cidades no qual possui como chave primaria estes valores.

4.3 Sistema Desenvolvido com o MongoDB

Neste capítulo, será apresentado o sistema de gerenciamento ferroviário desenvolvido com MongoDB em BSON. O sistema inclui todas as entidades necessárias para a operação do serviço ferroviário, utilizando uma das principais características do MongoDB: a incorporação. Os dados de telefone e email foram incorporados dentro das collections "Maquinistas" e "Clientes". Além disso, a entidade "Categorias" foi incorporada dentro da collection "Trens". O sistema também utiliza referências, como observado na collection "Viagens", onde o campo "Estacoes" contém IDs referentes às cidades.

```

/*Clientes*/

[{
  "_id": 3,
  "nome": "Marcos Silva",
  "email": "Marcos@gmail.com",
  "telCliente": {
    "DDD": 12,
    "Tel": 992039122
  },
  "endereco": {
    "Rua": "25 de Julho",
    "CEP": "12018221",
    "Bairro": "Independencia",
    "Numero": 212
  },
  "DatadeRegistro": "01-05-2024"
},
{
  "_id": 2,
  "nome": "Ana Matos",
  "email": "ana@gmail.com",
  "telCliente": {
    "DDD": 12,
    "Tel": 991456811
  },
  "endereco": {
    "Rua": "Alemanha",
    "CEP": "12061001",
    "Bairro": "Jardim das Nações",
    "Numero": 123
  },
  "DatadeRegistro": "22-03-2024"
}]

/*Passagens*/

[{
  "_id": {
    "$oid":
"667c21ed10376b898b073007"
  },
  "StatusPassagem": "Confirmada",
  "Poltrona": 23,
  "DatadaCompra": "2024-06-18",
  "Valor": 78.8,
  "Clientes_id": 1,
  "Viagens_id": 1
}]

/*Trens*/

/*Cidades*/

[{
  "_id": 2,
  "NomeCidade": "Taubaté",
  "Estado": "São Paulo"
},
{
  "_id": 3,
  "NomeCidade": "Gonçalves",
  "Estado": "Minas Gerais"
}]

/*Maquinistas*/

[{
  "_id": {
    "$oid":
"667b90e610376b898b073004"
  },
  "nome": "Junior Nascimento",
  "email": "JN@gmail.com",
  "telCliente": {
    "DDD": 11,
    "Tel": 998258718
  },
  "RG": "573482348",
  "endereco": {
    "Rua": "Das Palmeiras",
    "CEP": "12032181",
    "Bairro": "Quinta da Frutas",
    "Numero": 85
  },
  "Trens_id": [1,2],
  "DatadeRegistro": "12-10-2022"
}]

/*Viagens*/

[{
  "_id": 1,
  "DataSaida": "2024-09-24",
  "DataChega": "2024-09-25",
  "HorarioSaida": "01:00:00",
  "HorarioChegada": "05:00:00",
  "Estacoes": {
    "Saida": {
      "Nome": "Estação da Luz",
      "Endereco": {
        "Rua": "Newton de Vasconcellos",
        "Bairro": "Jardim Esperança",
        "Numero": 27
      },
      "Cidade_id": 2
    }
  }
}]

```

```

[ {
  "_id": 1,
  "StatusTrem": "Ativo",
  "Capacidade": 110,
  "Categoria": "Carga"
},
{
  "_id": 2,
  "StatusTrem": "Ativo",
  "Capacidade": 50,
  "Categoria": "Intermunicipal"
} ]

},
"Chegada": {
  "Nome": "Estação da Mantiqueira",
  "Endereco": {
    "Rua": "José Fragozzo",
    "Bairro": "Jardim Inglaterra",
    "Numero": 156
  },
  "Cidade_id": 3
},
"CidadeDestino_id": 2,
"Trens_id": 3
} ]

```

Algoritmo 2: Sistema de Gerenciamento Ferroviário

É importante destacar que, embora o MongoDB possua a possibilidade do uso de referências, é recomendado utilizar a incorporação sempre que possível. O MongoDB possui uma regra de ouro que diz "Dados que são acessados juntos devem estar armazenados juntos", garantindo assim a atomicidade. Portanto, as referências devem ser usadas com moderação.

5 CONCLUSÃO

Após a realização deste estudo de caso, foi possível adquirir um aprofundamento significativo sobre modelos de bancos de dados não relacionais, com foco específico no MongoDB. Durante o projeto, exploramos as características distintivas do MongoDB, sua sintaxe e práticas recomendadas, culminando na criação de um sistema de gerenciamento ferroviário robusto e eficiente utilizando um banco de dados não relacional.

O objetivo principal de criar um sistema utilizando um banco de dados não relacional foi concluído com sucesso, evidenciando a flexibilidade e eficiência deste tipo de banco de dados em comparação com os modelos relacionais tradicionais. No entanto, este estudo também destacou áreas para futuras melhorias e aprofundamento. Entre as possíveis melhorias estão:

- **Aprimoramento da Performance Utilizando Índices:** Aprender mais sobre a criação e otimização de índices em MongoDB para melhorar a performance das consultas e operações.
- **Modelagem de Dados no MongoDB:** Aprofundar o conhecimento sobre técnicas avançadas de modelagem de dados para maximizar a eficiência e a escalabilidade do banco de dados.
- **Utilização de Referências e Incorporação:** Compreender melhor as melhores práticas para decidir entre o uso de referências ou a incorporação de dados, garantindo uma estrutura de dados balanceada que atenda às necessidades específicas do sistema.

Essas melhorias e aprendizados contínuos garantirão que o sistema de gerenciamento ferroviário evolua e se adapte às demandas futuras, proporcionando uma base sólida para a expansão e a inovação no uso de bancos de dados não relacionais.

REFERÊNCIAS

A. LIVROS:

Paniz, Daivid. **NoSQL**: Como armazenar os dados de uma aplicação moderna. São Paulo: Casa do Código, 2020.

Marquesone, Rosangela. **Big Data**: Técnicas e tecnologias para extração de valor dos dados. São Paulo: Casa do Código, 2016.

B. ONLINE:

Docs.MONGODB Disponível em: <<https://www.mongodb.com/pt-br/docs/>>. Acesso em: 25 jun 2024.

Rdbms-mongodb-migration-guide.MONGODB Disponível em: <<https://www.mongodb.com/resources/solutions/use-cases/rdbms-mongodb-migration-guide>>. Acesso em: 25 jun 2024.

GLOSSÁRIO

SQL: O SQL é uma linguagem padrão para manipulação de registros em bancos de dados relacionais. A sigla SQL vem dos termos em inglês “Structured Query Language”, que podem ser traduzidos para o português como “Linguagem de Consulta Estruturada”.

BSON: O BSON é um formato de serialização de documentos em binário, projetado para armazenar e transmitir dados de forma eficiente e estruturada. Sua sigla, “Binary JSON,” indica sua relação com o formato **JSON (JavaScript Object Notation)**, mas com um toque especial: ele é binário.

Sharding: Sharding é um conceito originário do gerenciamento de banco de dados tradicional. Ele se refere ao processo de divisão de um banco de dados maior em partes menores e mais gerenciáveis, chamadas shards. A sua aplicação em blockchain visa melhorar a escalabilidade, mantendo o princípio da descentralização.

CRUD: CRUD é uma abreviação de **Create, Read, Update e Delete**. Essas quatro operações são comuns em muitos sistemas de banco de dados, e a maioria dos desenvolvedores as usa para criar aplicativos que precisam armazenar, ler, atualizar e excluir dados.

NoSQL: NoSQL, também conhecido como “não apenas SQL”, é uma abordagem de design de banco de dados que permite o armazenamento e consulta de dados fora das estruturas tradicionais encontradas em bancos de dados relacionais.

Big Data: Big Data é um processo de **coleta, armazenagem, organização, análise e interpretação de grandes volumes de dados de uma empresa ou mercado de atuação**. Ele serve para direcionar as companhias em processos de tomada de decisão, resultando em ações mais estratégicas e assertivas.

SGBD: SGBD é o nome dado para qualquer software que controla e gerencia dados de um banco de dados por meio de uma interface gráfica.

Transações ACID: ACID é uma sigla para as quatro principais características que definem uma transação: **Atomicidade, Consistência, Isolamento e Durabilidade**. Quando uma operação de banco de dados possui essas propriedades ACID, ela pode ser chamada de transação ACID.