



---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

**VITOR BORTOLOTO**

**MONGO DB, UMA ALTERNATIVA AOS SGBDS RELACIONAIS**

**Americana, SP**

**2022**



---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

**VITOR BORTOLOTO**

**MONGO DB, UMA ALTERNATIVA AOS SGBDS RELACIONAIS**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Clerivaldo José Roccia.

Área de concentração: Banco de dados.

**Americana, SP**

**2022**



---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

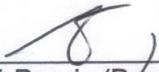
Vitor Bortoloto

**MONGODB - UMA ALTERNATIVA AOS SGBDS RELACIONAIS**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pelo CEETEPS/Faculdade de Tecnologia – Fatec/ Americana.  
Área de concentração: Banco de dados

Americana, 21 de Junho de 2022.

**Banca Examinadora:**

  
\_\_\_\_\_  
Clerivaldo José Roccia (Presidente)  
Mestre  
Fatec Americana

  
\_\_\_\_\_  
Eduardo Antonio Vicentini (Membro)  
Mestre  
Fatec Americana

  
\_\_\_\_\_  
Lucas Serafim Parizotto (Membro)  
Especialista  
Fatec Americana

## **RESUMO**

Atualmente há um aumento muito grande na quantidade de informações que são geradas e armazenadas eletronicamente. A criação dos bancos de dados relacionais foi uma revolução, pois apresentou uma maneira mais fácil e rápida de se armazenar os dados, porém com esse aumento no número de dados armazenados, os bancos de dados relacionais vêm perdendo sua força para atender às necessidades do mercado atual. Para atender essa demanda, surgiram os bancos de dados NoSQL, os quais vem conquistando cada vez mais o seu espaço no mercado nas aplicações mais modernas. Este trabalho tem como objetivo demonstrar a utilização do Mongo BD como sendo uma alternativa aos bancos de dados relacionais. O mongo DB é um software de banco de dados orientado a documentos livres, que possui código aberto e é multiplataforma. O MongoDB é escrito em linguagem C++ e é classificado como um programa de bancos de dados NoSQL, que utiliza documentos semelhantes ao JSON. A sua utilização dispensa as estruturas dos bancos de dados relacionais e faz uso de esquemas dinâmicos, tornando dessa forma a integração de dados em determinados tipos de aplicativos mais fácil e rápida. Dito isso, este trabalho demonstrará através de um estudo como a utilização do MongoDB pode substituir os bancos de dados relacionais.

**Palavras chave:** Mongo DB, Bancos de dados relacionais, NoSQL, JSON.

**ABSTRACT**

Nowadays there is an increase in the amount of information that are generated electronically. The creation of the relational databases was a revolution, because it has presented an easier and faster way to store data, although with this growth of the amount of data stored, the relational databases are losing their strength to meet the necessities of the current market. To fill this demand the NoSQL databases were created and are conquering more and more space in the market in modern applications. This monography's goal is to demonstrate the MongoDB's usage as an alternative to relational databases. MongoDB is a document-oriented database software, with open-source code and it is multiplatform. It is written in C++ language and it is classified as a NoSQL database that uses documents similar to JSON. Its usage dispenses the schema of relational databases and utilizes dynamic schemas that make this data integration form in some kinds of applications easier and faster. That said, this monography will show through a study how the MongoDB usage can replace the relational databases.

**Keywords:** Mongo DB, Relational databases, NoSQL, JSON.

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

**LISTA DE FIGURAS**

Figura 1 - Banco de Dados .....	12
Figura 2 - Sistema Gerenciador de Bancos de Dados .....	13
Figura 3 - Modelo Relacional.....	15
Figura 4 - NoSQL .....	16
Figura 5 - MongoDB .....	17
Figura 6 - Modelo Relacional.....	19
Figura 7 - Atributos.....	20
Figura 8 - Chave Estrangeira .....	22
Figura 9 - Modelo Relacional.....	23
Tabela 1 - Modelo Relacional X NoSQL.....	26
Figura 10 - Escalonamento Horizontal .....	28
Figura 11 - API .....	29
Figura 12 - Replicação de Dados.....	30
Figura 13 – Banco de alunos no MongoDB.....	33
Figura 14 – Tabela Pessoas no MySQL.....	34
Figura 15 – Exemplo de documento no MongoDB.....	35
Figura 16 – Tabela Habilidades no MySQL.....	35
Figura 17 – Resultado do join no MySQL.....	36
Figura 18 – Botão Export Collection.....	37
Figura 19 – Escolha do conteúdo.....	38
Figura 20 – Escolha dos campos .....	39
Figura 21 – Escolha do tipo de arquivo e diretório .....	39
Figura 22 – Importando uma base de dados.....	40
Figura 23 – Escolhendo o arquivo.....	40
Figura 25 – Base importada .....	41
Figura 26 – Exemplo de validação .....	43
Figura 27 – Documento validado X não validado .....	44
Figura 28 – Exemplo de análise do Compass .....	45

**LISTA DE TABELAS**

Tabela 1 - Modelo Relacional X NoSQL.....	26
---	----

## SUMÁRIO

1. INTRODUÇÃO.....	9
2. DESENVOLVIMENTO .....	11
2.1. Banco de dados .....	11
2.1.1. Sistema Gerenciador de Banco de Dados .....	12
2.1.2. Modelo de Dados .....	13
2.1.3. Modelo Relacional.....	14
2.1.4. NoSQL.....	15
2.1.5. MONGODB .....	17
2.2. BANCO DE DADOS RELACIONAL .....	18
2.2.1. Chaves Primárias e Estrangeira .....	21
2.2.2. Limitações do Modelo Relacional.....	22
2.3. MONGODB X MODELO RELACIONAL .....	24
2.3.1. Propriedades do NoSQL .....	27
2.4. MongoDB utilizado como alternativa ao Banco de Dados Relacional .....	31
2.4.1. Funcionamento do MongoDB.....	32
2.4.2. Casos de uso para utilização do MongoDB em substituição ao BD Relacional.....	32
2.4.3. Exemplo prático do MongoDB x MySQL .....	32
3. CONCLUSÃO.....	45
4. BIBLIOGRAFIA .....	47

## 1. INTRODUÇÃO

A quantidade de informações nas empresas tem crescido de forma rápida, desta forma os bancos de dados mais utilizados como os bancos de dados relacionais, tornam-se cada vez menos eficientes, já que esses bancos de dados são baseados em estruturas computacionais que utilizam da troca ou da adição de processadores e adição de memória.

Dessa forma, as empresas necessitam possuir uma estrutura mais escalável e mais dinâmica para que assim possam armazenar os dados e processá-los. De acordo com Sadalage (2013), os bancos de dados NoSQL tem ganhado destaque. Os bancos de dados NoSQL são projetados em arquiteturas distribuídas, permitindo então que eles possam processar os grandes volumes de dados que possuem, com disponibilidade e escalabilidade.

Existem diversos sistemas gerenciadores de bancos de dados, mais comumente chamados de SGBD. No caso desse trabalho, abordaremos os SGBD não relacionais, os quais existem diferentes tipos e cada um tem as suas particularidades. Por exemplo, existem SGBDs não relacionais do tipo orientado a documentos, chave-valor, colunas e grafos. O MongoDB que será abordado nesse estudo se encaixa nos SGBDs orientados a documentos e atualmente seu uso está em constante crescimento no mercado.

Esse trabalho tem como objetivo principal verificar a possibilidade de substituição dos bancos de dados relacionais pelo Mongo DB e como objetivos específicos conceituar o Mongo DB, demonstrar a utilização dos bancos de dados relacionais e verificar as vantagens e as desvantagens do Mongo DB em relação aos bancos de dados relacionais.

A metodologia utilizada para a realização desse trabalho é a pesquisa bibliográfica, que de acordo com Fachim (2010), a pesquisa bibliográfica é a mais

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

importante no trajeto da pesquisa e que constitui o ato de ler, selecionar, fichar, organizar e compreender.

O trabalho foi estruturado em três capítulos. O primeiro capítulo conceituará Banco de Dados e o Mongo DB, o segundo capítulo demonstrará a utilização dos bancos de dados relacionais, as vantagens e desvantagens na utilização do mongo DB e a possibilidade de substituição dos bancos de dados relacionais pelo Mongo DB.

Baseando-se nas informações a partir do estudo realizado, o terceiro e último capítulo será utilizado para as considerações finais sobre o trabalho que foi feito.

## **2. DESENVOLVIMENTO**

### **2.1. Banco de dados**

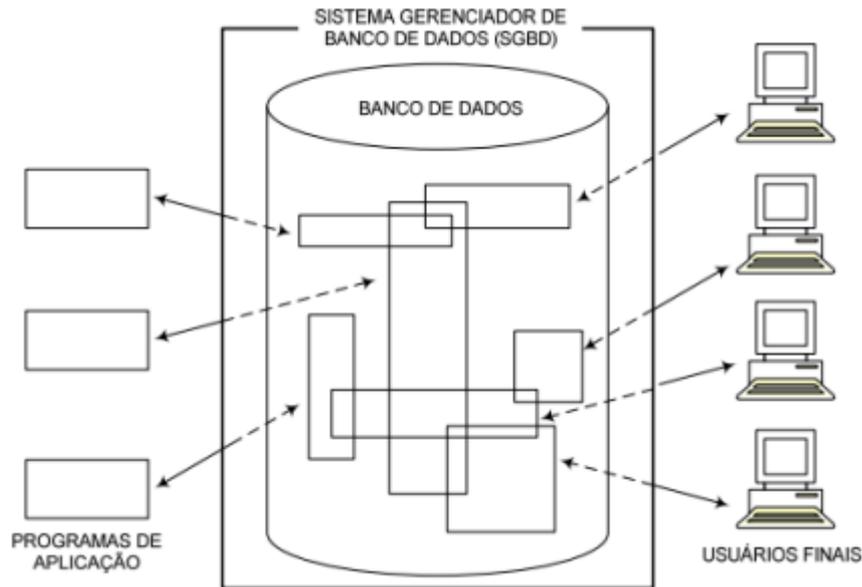
Banco de dados pode ser definido como uma coleção organizada de informações ou dados que são armazenados de forma eletrônica em um sistema computacional. De acordo com Date (2004, p.6), "um Banco de Dados é uma coleção de dados persistentes e estas informações são utilizadas pelos sistemas de aplicação de uma empresa".

Assim, os bancos de dados podem ser definidos como um local onde são armazenados todos os dados que são necessários para a manutenção das atividades cotidianas de uma determinada empresa e que possui vários níveis de interação como o mundo real e com todos os interessados em seus conteúdos. Os componentes dos bancos de dados são:

- O próprio banco de dados; e
- O software gerenciador do banco de dados.

A figura 1 apresenta um esquema de como é a composição de um banco de dados.

**Figura 1 - Banco de Dados**



Fonte: Date (2004)

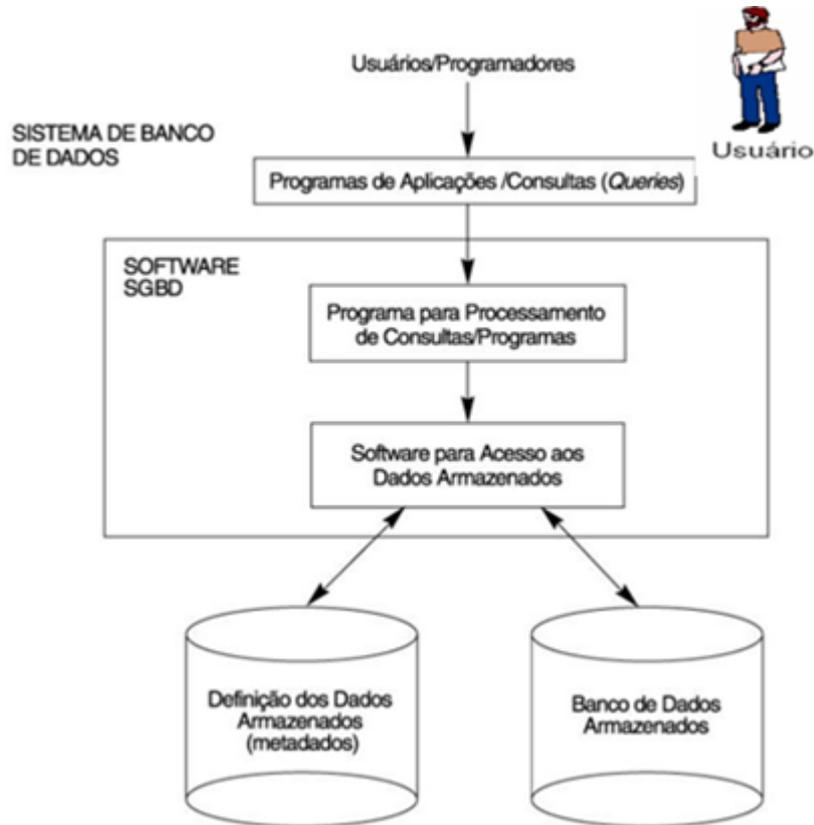
Uma outra definição de banco de dados que podemos citar, de acordo com Elmasri e Navathe é:

Um banco de dados representa um determinado aspecto do mundo real, chamado de minimundo ou de universo de discurso. As mudanças no minimundo são refletidas no Banco de Dados. Um banco de dados é uma coleção logicamente coerente de dados com um significado inerente. Uma variedade aleatória dos dados não pode ser chamada de bancos de dados. Um banco de dados geralmente é projetado, construído com dados para uma finalidade específica. Ele possui um grupo definido de usuários e possui algumas aplicações previamente concebidas nas quais os usuários estão interessados. (ELMASRI E NAVATHE, 2011, P.3).

### **2.1.1. Sistema Gerenciador de Banco de Dados**

A gestão dos bancos de dados é feita através do Sistema Gerenciador de Banco de Dados (SGBD), que tem como principal objetivo proporcionar um ambiente eficiente para a recuperação e para o armazenamento das informações do banco de dados. (SILBERSCHATZ; KORTH; SUDARSHAN, 1999, P.1).

Figura 2 - Sistema Gerenciador de Bancos de Dados



Fonte: Emerson Fender (2012, p.2)

O gerenciador de banco de dados disponibiliza os recursos para definição, construção, manipulação, compartilhamento, proteção e para manter o banco de dados. Portanto, ele pode ser entendido como um conjunto de programas responsáveis pelo gerenciamento do banco de dados, tendo em vista que o mesmo fornece suporte aos métodos de acesso, ou seja, as formas de armazenar e de recuperar as informações de forma apropriada. (ELMASRI; NAVATHE, 2011).

### 2.1.2. Modelo de Dados

O modelo de dados pode ser definido como uma coleção de conceitos que podem ser utilizados para realizar a descrição da estrutura do banco de dados. Então, ele pode ser entendido como uma maneira de descrever os seus

---

## Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

dados, relacionamentos, semântica e restrições de consistência. O modelo de dados demonstra o projeto do banco de dados em sua forma física, lógica e de visão. (SILBERSCHATZ, 2006). Todos os sistemas gerenciadores de bancos de dados seguem um modelo de banco de dados para diferenciar estes dados de acordo com a sua representação.

### 2.1.3. Modelo Relacional

O modelo relacional refere-se aos três principais aspectos dos dados, sendo eles: a estrutura de dados, a integridade dos dados e a manipulação dos dados. O modelo relacional representa os dados como um conjunto de relações que é similar a uma tabela de valores, na qual as linhas são denominadas tuplas, as colunas são denominadas de atributos e a tabela é denominada de relação. (ELMASRI; NAVATHE, 2011).

As tabelas são constituídas de vários elementos os quais possuem características comuns, conforme vemos na figura 3. Esses elementos são denominados de "atributos". Pode-se verificar na figura 3 o conceito de atributo, sendo que cada um destes atributos indicam características que estão associadas a tabela. Os seguintes atributos estão presentes no exemplo: *id*, *description*, *data*, *type*, *unit*, *variable*, *type* e *symbol*. Além disso, temos o "domínio", que são os possíveis valores dos atributos, e as tuplas, que indicam as linhas existentes na relação.

Figura 3 - Modelo Relacional

variable (relação)

id	description	data_type	unit	variable_type	symbol
[PK] integer	character varying(200)	character(1)	integer	integer	character varying(50)
1	Temperatura	3	1	1	temp
2	Umidade Relativa	3	2	1	umidrel
3	Fracao de Cobertura de Nuvens Baixas	3	1	1	cfrac1
4	Fracao de Cobertura de Nuvens Medias	3	1	1	cfracm
5	Fracao de Cobertura de Nuvens Altas	3	1	1	cfrach
6	Precipitacao	3	1	1	ttprec
7	Radiacao de Onda Curta	3	1	1	aswin
8	Pressao Atmosferica	3	1	1	pressatcoef
9	Velocidade Rajada do Vento	3	1	1	velrajvento
10	Direcao Vento	3	1	1	direvento
11	Radiacao Solar	3	1	1	radso1ar
12	Ponto de Orvalho	3	1	1	porvalho
13	Latitude	3	1	1	Latitude
14	Longitude	3	1	1	Longitude
15	Altitude	3	1	1	Altitude
17	Temperatura do Solo	3	1	1	tempolo
18	Umidade do solo	3	1	1	umiso1o
19	Nivel do tanque	3	1	1	niveltan
20	Saldo Rad Cor	3	1	1	saldo1cor
21	Umid Rad Cor	3	1	1	umid1cor

T  
U  
P  
L  
A  
S

Fonte: Autor

#### 2.1.4. NoSQL

O modelo NoSQL foi utilizado pela primeira vez no ano de 1998, quando Carlo Strozzi baseando-se no modelo relacional, desenvolveu um modelo relacional sem interface SQL, criando um termo genérico para uma classe de banco de dados que rompe os paradigmas dos bancos de dados baseados nos modelos relacionais (SUISSA, 2010). Todos os bancos de dados NoSQL foram criados baseados nas necessidades do desenvolvimento de *websites*. Eles foram projetados para promoção de alternativas de armazenamento com velocidades e disponibilidade elevadas.

A figura abaixo mostra alguns logos de banco de dados não relacionais:

Figura 4 - NoSQL



Fonte: Juan Benitez (2012, p .1)

Os bancos de dados NoSQL são classificados conforme a sua maneira de armazenamento de dados, sendo eles:

- **Chave-valor:** são armazenamentos semelhantes a utilização de mapas ou de dicionários, todos os dados são endereçados por uma chave única que permite aos clientes colocar e solicitar valores através de chaves;
- **Orientado a colunas:** este modelo armazena todo o seu conteúdo de maneira inversa aos bancos de dados que são orientados a linhas;
- **Grafos:** é fundamentado na teoria dos grafos que consiste em nós, propriedades e arestas;
- **Orientado a documento:** este modelo é baseado no armazenamento de dados dentro de documentos que são semelhantes a JSON. Este modelo armazena, recupera e gerencia dados semiestruturados. O MongoDB é um modelo deste tipo de NoSQL. O modelo orientado a documentos permite a recuperação de documentos através de uma API ou de uma linguagem de consulta que é disponibilizada pelo banco de dados. Todos

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

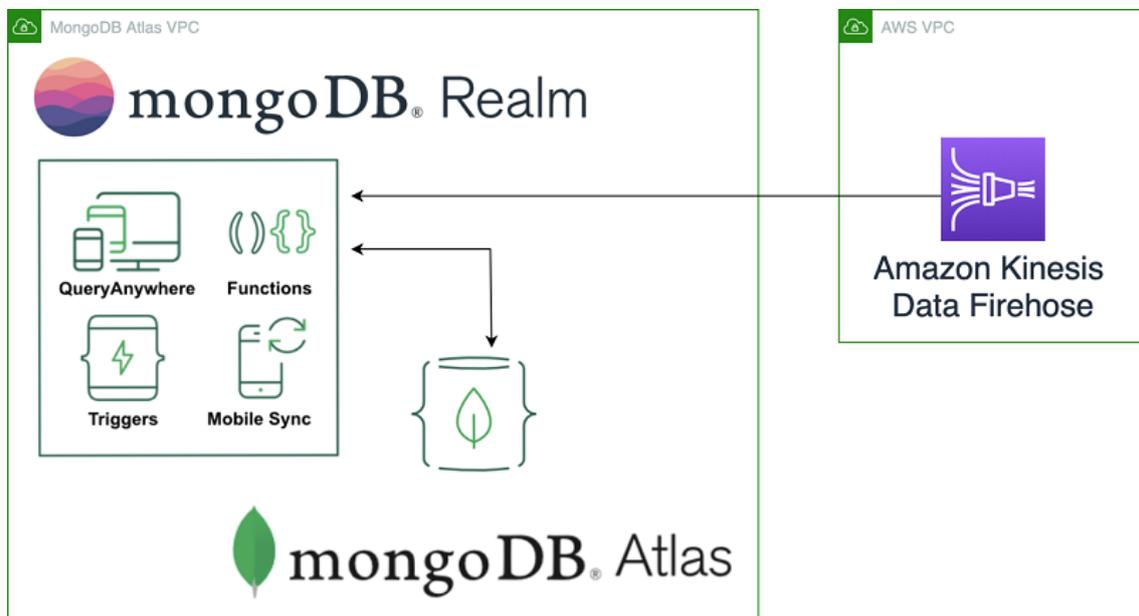
os bancos de dados atuais têm a capacidade de fornecer suporte aos documentos que estão em formato JSON (LEAVITT, 2010).

**2.1.5. MONGODB**

O MongoDB foi desenvolvido como um componente de serviço, pela empresa chamada de 10gen, em outubro do ano de 2007 e em 2009 passou a ser um software open source. O nome MongoDB vem do inglês *humongous*, que significa gigantesco. Ele é uma aplicação de código aberto, possui alta performance, dispensa a existência de esquemas e é orientado a documentos.

A figura 5 mostra algumas funcionalidades do MongoDB, como por exemplo a conexão com o serviço de computação em nuvem AWS, sincronia com dispositivos móveis e a criação de funções e gatilhos.

**Figura 5 - MongoDB**



Fonte: Anusha Dharmalingam and Igor Alekseev (2020, p.1)

O MongoDB é um banco de dados bem diferente dos tradicionais bancos de dados relacionais, já que ele armazena os dados em documentos definidos, enquanto os bancos relacionais armazenam os dados em estruturas rígidas, tais como tabelas (CUNHA, 2011).

---

## **Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

Os documentos utilizados no MongoDB são unidades de dados armazenáveis e possuem uma estrutura comparável a um documento em XML ou em JSON. Esses documentos são denominados BSON (Binary encoded Javascript Object Notation), que é bastante similar ao JSON. Devido a maneira de armazenamento dos dados, o MongoDB é considerado um banco de dados orientado a documentos e é um dos projetos que mais se destaca no NoSQL, tendo em vista que a sua linguagem de consulta é de alta performance. Por ser baseado em documentos o MongoDB facilita a migração das bases de dados e suas consultas são fáceis e rápidas de realizar.

O MongoDB pode ser gerenciado através da utilização de uma interface gráfica denominada "MongoDB Compass", que pode ser utilizada para gerar buscas, agregações e analisar os dados da sua base de dados de uma maneira rápida e fácil. O MongoDB Compass é gratuito e pode ser utilizado tanto no Windows quanto no Linux e macOS.

## **2.2. BANCO DE DADOS RELACIONAL**

Os bancos de dados relacionais surgiram devido as seguintes necessidades:

- Aumento da independência de dados nos sistemas gerenciadores de bancos de dados;
- Para promover um conjunto e funções que estão apoiadas na álgebra relacional, para realizar o armazenamento e recuperação dos dados;
- Para permitir o processamento ad hoc.

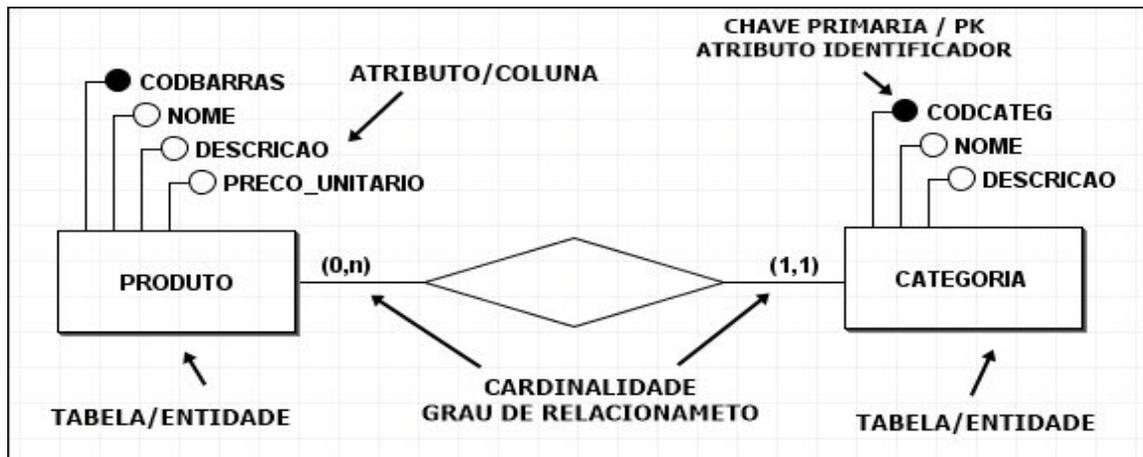
O modelo relacional foi o resultado de estudos teóricos realizados por Codd (1970), através da teoria dos conjuntos e da álgebra relacional. Esse modelo foi apresentado em um artigo publicado no ano de 1970, no entanto foi no ano de 1980 que o modelo foi implementado de fato.

## Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

O modelo relacional é o mais flexível e mais adequado para solucionar diversos tipos de problemas a níveis de concepção e de implementação de base de dados. A estrutura fundamental do modelo relacional é a relação e essa relação é construída por um ou mais atributos, que são os campos. Os campos têm a função de traduzir o tipo dos dados e de armazená-los.

A figura 6 demonstra um exemplo da estrutura de um banco de dados relacional:

**Figura 6 - Modelo Relacional**



Fonte: Luiz Santana (2019, p.2)

Podemos observar na figura acima que uma tabela é constituída por atributos, os quais podem traduzir todos os tipos de dados e então armazená-los nessas tabelas. No modelo relacional as implementações das estruturas de dados são organizadas de forma que haja relações coerentes entre as tabelas e essas relações são definidas pela cardinalidade. Como é mostrado na figura 6, a cardinalidade define se uma tabela pode ter um, vários ou nenhum registro na tabela ligada a ela para cada registro na própria tabela. Para trabalhar com tabelas foram desenvolvidas algumas restrições para evitar aspectos indesejáveis no modelo relacional, tais como:

- Repetição de informações;

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

- Incapacidade de representação das informações e
- Perda de informações.

Estas restrições são:

- Integridade referencial;
- Chaves;
- Integridade de junção destas relações.

Para Elmasri e Navathe (2005), o modelo relacional representa o banco de dados como uma coleção das relações, assim cada uma dessas relações se assemelha a uma tabela de valores.

Portanto, pode-se definir um modelo de dados como sendo uma combinação de um conjunto de estruturas de dados; da tabela ou relação; conjunto de operadores que são responsáveis pela manipulação do banco de dados e regras de integridade que são definidas no esquema do banco de dados. Essas regras são aplicadas nas instâncias dos bancos de dados.

No modelo relacional, cada linha é denominada de tupla, os títulos das colunas são os atributos e a tabela é chamada de relação. Os tipos de dados descrevem os tipos de valores que podem aparecer em cada uma das colunas e são denominados de domínio. A figura 7 demonstra essa terminologia.

Fonte: Autoria própria

**Figura 7 - Atributos**

#IDPedido	#IDCliente	DataPedido	LocalEntrega	Valor
1	5	5/12/06	Lavras	R\$ 530,00

○ Domínio      ○ Relação      ■ Tupla

### **2.2.1. Chaves Primárias e Estrangeira**

Os bancos de dados relacionais são compostos de diversas tabelas as quais armazenam todos os dados operacionais dos sistemas que estão em funcionamento no dia-a-dia de uma organização. Dependendo do tipo e da quantidade de dados que são armazenados, são desenvolvidos diversos bancos de dados, sendo um para cada domínio das aplicações. As restrições tanto da integridade do tipo de identidade e do referencial são mantidas através da utilização de diversos tipos de chaves.

Para o estabelecimento das restrições de integridade e para o estabelecimento de relacionamentos entre as tabelas, são utilizados campos que são identificados como chaves e são denominadas de chaves primárias e estrangeiras. As chaves primárias correspondem a uma ou a várias colunas que não possuem valores duplicados dentro de uma tabela. Já a chave estrangeira é aquela correspondente a uma ou a diversas colunas em que seus valores estejam identificados como chaves primárias de outra tabela. A chave estrangeira pode ser denominada como o mecanismo que define todos os relacionamentos de um banco de dados relacional. (DATE, 2000).

A chave primária é escolhida a partir de um conjunto de chaves candidatas possíveis para uma determinada entidade. Da mesma maneira que as chaves candidatas representam uma entidade, as chaves primárias representam um valor único e não nulo. (DATE,2000).

As chaves estrangeiras são um conjunto de atributos de uma relação que possuem valores que devem ser correspondentes a valores de uma determinada chave candidata de outra relação. As chaves estrangeiras são consideradas ferramentas poderosas para auxílio na manutenção da integridade do banco de dados. A figura 8 abaixo demonstra um modelo de chave estrangeira.

Figura 8 - Chave Estrangeira

#IDCliente	Nome
5	Maria

Cliente

#IDPedido	#IDCliente
1	5

Pedido

Fonte: Autor

### 2.2.2. Limitações do Modelo Relacional

Devido ao avanço das novas aplicações, as quais foram impulsionadas pelo mercado de desenvolvimento de sistemas, novas características encontraram o seu lugar nessas aplicações. Dessa forma, surgiu a necessidade de poder-se criar uma representação dos tipos de dados, suporte para transações longas, suporte ao processamento de conhecimentos com tratamento de restrições e de gatilhos, contando com um aumento no grau de inter-relacionamentos entre os dados. (CARDOSO, 2003).

Bancos de dados relacionais funcionam como simples dados que envolvem poucas associações entre as relações, no entanto elas apresentam deficiências quando suas aplicações são mais complexas e necessitam ser projetadas e implementadas. As limitações referentes a ausência de tipos que são definidos pelos usuários são:

- Dificuldade de representação de objetos do mundo real em suas aplicações;
- Incompatibilidade com linguagens de programação;
- Divergências entre os objetos da linguagem e as estruturas das entidades do modelo relacional;
- Suporte a grandes bancos de dados;

---

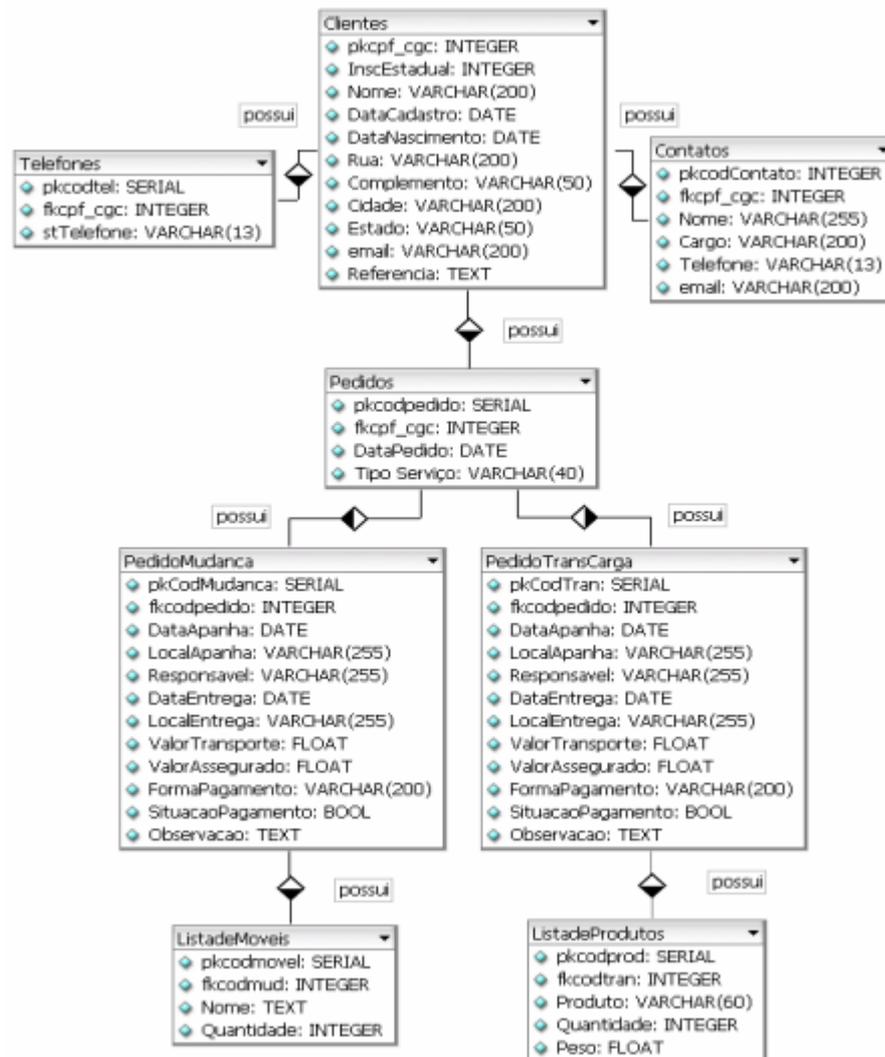
### **Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

- Suporte a transações longas; e
- Suporte a processamento de conhecimentos.

Os bancos de dados orientados a objetos foram propostos para fazer face a todas as necessidades de aplicações que são mais complexas, ou seja, que possuem muitos atributos e uma grande quantidade de dados. O método de orientação a objetos oferece muita flexibilidade para lidar com exigências sem limites, sendo uma solução para esses problemas. Então, a partir dessas limitações, surgiram os SGBBOO (Banco de dados NoSQL orientados a documentos).

A figura 9 demonstra um exemplo de sistema baseado no modelo relacional e que possui oito entidades. Quanto maior o número de entidades, mais complexo o banco fica e a utilização de um banco NoSQL fica cada vez mais vantajosa.

**Figura 9 - Modelo Relacional**



Fonte: Autor

### 2.3. MONGODB X MODELO RELACIONAL

O Banco de Dados Relacional é uma tecnologia capaz de armazenar uma grande quantidade de dados, no entanto ele possui uma estrutura rígida, não permitindo desta forma uma alta escalabilidade. De acordo com Lóscio et al., (2011), a tecnologia relacional surgiu de uma proposta na década de 70.

Os bancos de dados atuais possuem muitas restrições relacionadas a escalabilidade, porque eles utilizam o escalonamento vertical nos servidores. Assim, quanto mais dados maior a necessidade de espaço no servidor e na

---

### **Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

memória. A diferença do banco de dados relacional para o Mongo DB que é um NoSQL, destaca-se porque ele atende aos requisitos de alta escalabilidade que são necessários para o gerenciamento de grande quantidade de dados, e por garantir a alta disponibilidade dos mesmos, características que são muito desejadas atualmente.

A tecnologia NoSQL propõe novos modelos de organização de dados, já que eles permitem que as aplicações possuam uma estrutura de dados mais flexível, possibilitando que as bases de dados sejam altamente escaláveis e permitem a manipulação livre das regras. Cada um dos SGBD seguem um tipo de organização de dados diferente.

A tabela abaixo demonstra uma análise comparativa entre as classes dos bancos de dados.

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

**Tabela 1 - Modelo Relacional X NoSQL**

	<b>Modelo Relacional</b>	<b>NoSQL</b>
Escalabilidade	Possível, mas complexo. Devido à natureza estruturada do modelo, a adição de forma dinâmica e transparente de novos nós no grid não é realizada de modo natural.	Uma das principais vantagens desse modelo. Por não possuir nenhum tipo de esquema pré-definido, o modelo possui maior flexibilidade o que favorece a inclusão transparente de outros elementos
Consistência	Ponto mais forte do modelo relacional. As regras de consistência presentes propiciam um maior grau de rigor quanto à consistência das informações.	Realizada de modo eventual no modelo: só garante que, se nenhuma atualização for realizada sobre o item de dados, todos os acessos a esse item devolverão o último valor atualizado.
Disponibilidade	Dada a dificuldade de se conseguir trabalhar de forma eficiente com a distribuição dos dados, esse modelo pode não suportar a demanda muito grande de informações do banco.	Outro fator fundamental do sucesso desse modelo. O alto grau de distribuição dos dados propicia que um maior número de solicitações aos dados seja atendido por parte do sistema e que o sistema fique menos tempo não disponível.

Fonte Brito (2010, p.5)

A classe de banco de dados do MongoDB é o NoSQL, que não tem como objetivo a substituição do modelo relacional, mas sim o de atender as demandas do mercado em situações em que exista a necessidade de se trabalhar de forma mais flexível e com cargas de dados que são semiestruturadas ou não estruturadas, bem como em sistemas que apresentam grandes volumes de dados e que necessitam de disponibilidade para seus usuários. (ALMEIDA e BRITO, 2012).

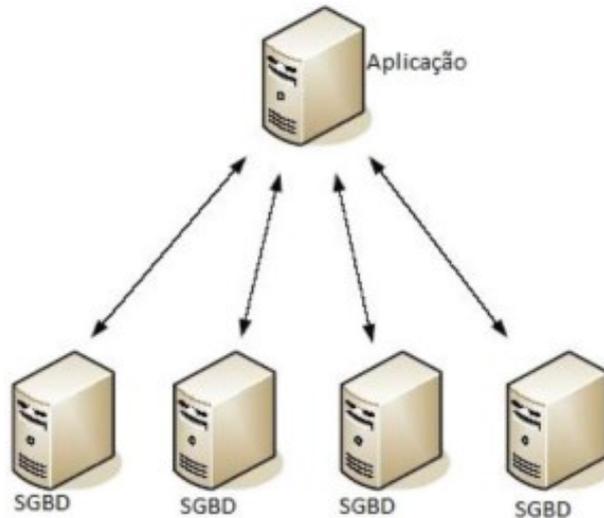
### **2.3.1. Propriedades do NoSQL**

Os bancos de dados NoSQL, apresentam determinadas propriedades fundamentais que os diferem dos sistemas de bancos de dados relacionais, tornando-os desta forma mais adequados para o armazenamento de grandes volumes de dados que não são estruturados ou que são semiestruturados. Os NoSQL possuem as seguintes características: escalabilidade, API's simples de acesso aos dados, ausência de esquemas ou esquemas flexíveis e suporte negativo a replicações. A seguir, algumas dessas características serão explicadas:

- **Escalabilidade:** de acordo com Delgado (2011, p.30), os bancos de dados NoSQL tem como principal característica a capacidade de oferecer a uma aplicação uma estrutura de dados flexíveis que possibilitam que seus dados sejam manipulados livres de regras e que seus bancos de dados sejam distribuídos em um maior nível do que seria possível se fosse seguida uma estrutura relacional ou que fosse pouco flexível.

A escalabilidade de um banco de dados é determinada pela forma da divisão do banco de dados, tendo em vista que para cada petição existe um limite de dados que é suportado. Para demonstrar isso, a figura 10 abaixo ilustra uma certa quantidade de servidores que são utilizados para uma única aplicação, solução que é conhecida como escalonamento horizontal e tem sido a mais utilizada nas camadas de aplicações, principalmente quando se trata de aplicações web.

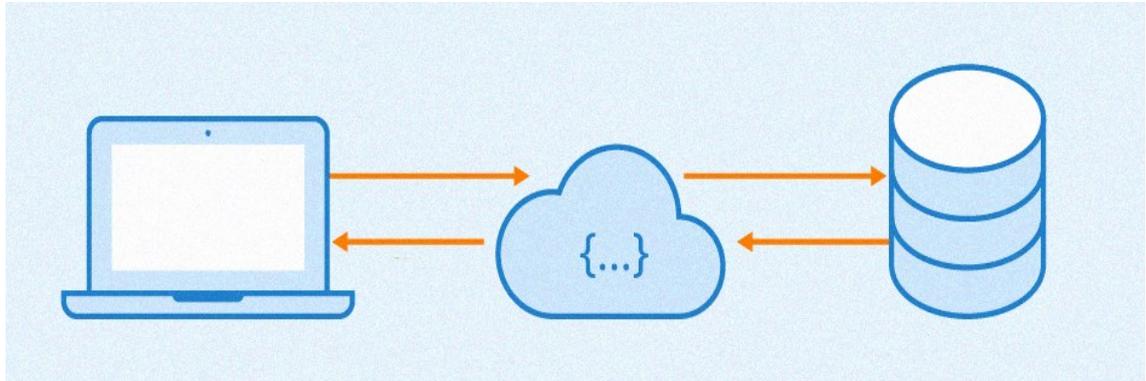
Figura 7 - Escalonamento Horizontal



Fonte: Delgado (2011)

- **API simples de acesso aos dados:** o NoSQL tem como objetivo prover uma maneira eficiente de acesso aos dados para oferecer alta disponibilidade e escalabilidade, ou seja, o foco não está no fato de como os dados são armazenados e sim como estes poderão ser recuperados de uma maneira eficiente. Para isso é necessário que uma API seja desenvolvida para facilitar o acesso a estas informações, permitindo desta forma que qualquer aplicação possa utilizar os dados do banco de forma mais rápida e eficiente. A figura 11 mostra a movimentação entre uma aplicação e um banco de dados, tendo uma API como intermediário.

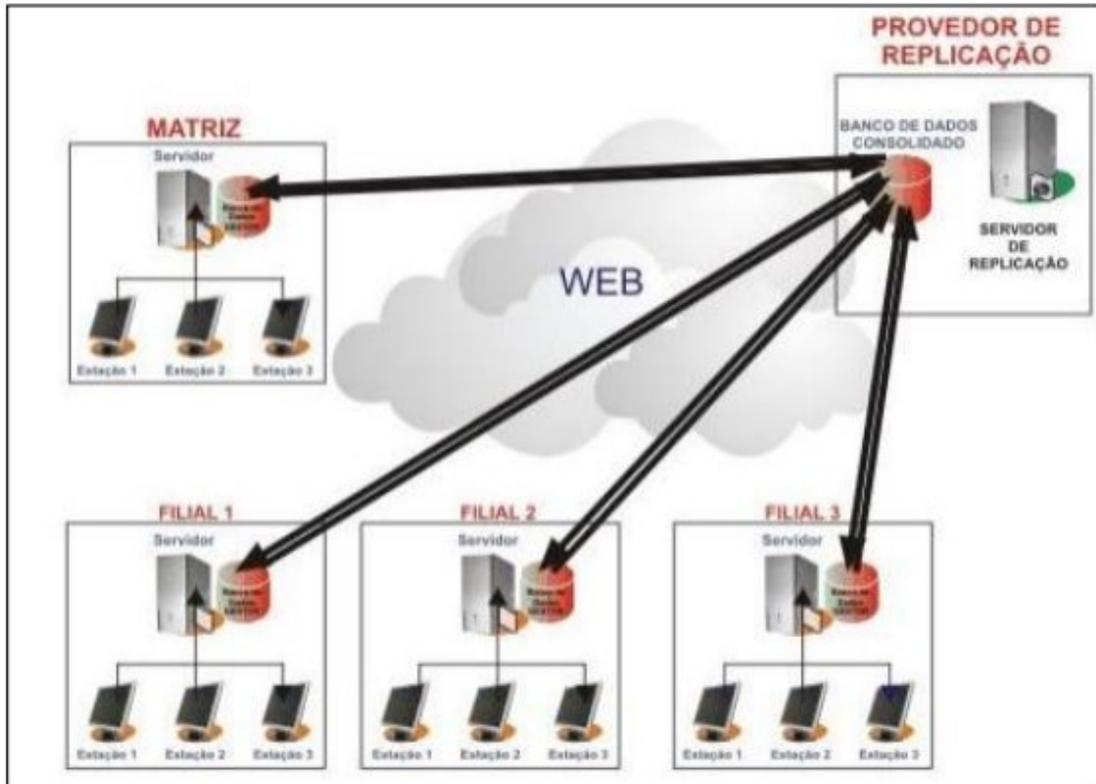
Figura 8 - API



Fonte: Ivan de Souza

- **Ausência de esquemas ou esquemas flexíveis:** os bancos de dados NoSQL possuem a característica de ausência completa ou quase completa do esquema que define a estrutura de dados modelados. Assim essa ausência de esquema é uma facilidade tanto para a escalabilidade quanto para contribuir para aumentar a disponibilidade do sistema. Em contrapartida, não existem garantias da integridade dos dados, fato que acontece nos bancos de dados relacionais, devido a sua rígida estrutura.
- **Suporte nativo a replicação:** a replicação é uma outra forma de prover a escalabilidade, pois ela consiste no armazenamento de um mesmo conjunto de dados em vários servidores distintos de uma forma nativa, diminuindo assim o tempo gasto para recuperação das informações. A figura 12 abaixo mostra um exemplo de como seria a replicação dos dados de uma empresa.

Figura 9 - Replicação de Dados



Fonte: Delgado (2001, p.35)

De acordo com Brito (2010 apud Delgado 2011, p.36), a utilização do suporte nativo a replicação permite que a leitura seja potencializada, já que a realização do balanceamento distribui as requisições de leitura para todas as máquinas. Dessa maneira a operação da leitura não é realizada de uma forma concentrada em um único servidor, mas sim distribuída entre todos os existentes nesta aplicação.

- **Taxonomia do modelo de dados NoSQL:** a taxonomia descreve os tipos de bancos de dados NoSQL que são baseados em chave-valor, orientados a documentos, banco de dados de família, de coluna e banco de dados de grafos. Para Sadalage e Fowler (2013, p.41) o modelo de dados na maioria das vezes serve para falar o modelo através do qual o gerenciador do banco de dados organiza estes dados, que é denominado

---

## **Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

de metamodelo. Através do surgimento da tecnologia NoSQL, cada uma das soluções possui modelos diferentes, que são divididos em quatro categorias. São elas: chave-valor, documento, família de colunas e grafos. As três primeiras possuem características comuns em seu modelo de dados, denominada de orientação agregada, que é um conjunto de objetos relacionados que são tratados como uma unidade.

### **2.4. MongoDB utilizado como alternativa ao Banco de Dados Relacional**

A utilização de Bancos de Dados relacionais nem sempre é a melhor opção para os projetos, já que eles possuem uma grande complexidade e por serem de difícil escalabilidade, esta forma de armazenamento de dados é ineficiente em algumas situações.

Quando existe um grande volume de dados ou quando existe a necessidade de uma baixa latência no acesso ou na extração das informações, torna-se mais vantajosa a utilização dos bancos de dados NoSQL (que significa "*Not Only SQL*"). Os bancos de dados não relacionais fogem do modelo tradicional de tabelas para realizar a implementação de soluções que são mais eficientes na manipulação de grandes massas de dados, e portanto, deveriam ser preferidos nessas ocasiões.

De acordo com uma pesquisa realizada pelo site DB-Engines o MongoDB é o primeiro NoSQL mais utilizado no mundo e é o quinto mais utilizado no ranking geral, ficando atrás somente dos SQLs mais consolidados, como o Oracle e MySQL. Ele é um banco de dados gratuito, de código aberto e que adota o modelo baseado em documentos e armazena os dados em arquivos denominados de BSON, que é como se fosse uma extensão do JSON. Ele oferece formas para o escalonamento, para replicação e para a realização de backups dos dados. O MongoDB disponibiliza também suporte a queries

---

## **Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

dinâmicas nos documentos utilizando uma linguagem poderosa. Outro destaque que o MongoDB apresenta é a conectividade com as linguagens de programação. Como o Python, por exemplo, basta instalar a biblioteca PyMongo que então será possível utilizar os comandos do MongoDB dentro da linguagem Python.

### **2.4.1. Funcionamento do MongoDB**

O MongoDB guarda os dados de uma determinada entidade em um arquivo que é formatado em uma sintaxe JSON-like. Ele é autossuficiente e contém todas as informações relacionadas a uma entidade, evitando dessa forma a necessidade de "JOINS" entre as tabelas para fazer as buscas no banco.

O MongoDB possui um método denominado de Sharding. Esse método possibilita o armazenamento de dados ao longo de múltiplas máquinas, permitindo dessa forma que haja uma escalabilidade horizontal. No MongoDB todas as coleções estão contidas em um banco, assim ele é denominado como sendo um banco de dados schemaless, ou seja, ele não força nenhuma estrutura definida para os documentos, assim uma coleção pode possuir diversos documentos diferentes e cada um conter campos distintos entre si.

### **2.4.2. Casos de uso para utilização do MongoDB em substituição ao BD Relacional**

O MongoDB utilizado em substituição ao Banco de Dados Relacional é uma excelente ferramenta para quem trabalha com BigData, tendo em vista que ele permite guardar qualquer tipo de dado. Desta forma os arquivos binários, tais como vídeos e imagens, podem ser armazenados em documentos. Essa é uma excelente forma de guardar os dados que não são estruturados para serem manipulados ou analisados.

---

## **Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

O MongoDB é capaz de fazer queries mais velozes e desta forma fornecer informações que são solicitadas em baixa latência, ele é recomendado para aplicações tanto mobile como web com requisições simultâneas e que possuem a necessidade de apresentar os dados que foram salvos em diversas instancias. Dessa maneira ele permite que as informações de uma determinada página sejam reunidas em um único documento e não em várias tabelas como no banco de dados relacional.

O MongoDB garante a alta disponibilidade através da replicação. O servidor MongoDB pode realizar copias duplicadas em mais de dois servidores de seus bancos de dados. Essas copias são conhecidas como um conjunto de réplicas que são organizados através de um processo de eleição onde os membros da réplica realizam a votação de qual servidor se torna o principal e quais atuam como secundários.

### **2.4.3 Exemplo prático do MongoDB x MySQL**

Como foi visto anteriormente, o MongoDB permite a criação de documentos que não precisam necessariamente ter uma estrutura definida como no MySQL, o que permite que eles possam ter dados diferentes, como por exemplo, na figura 13 vemos que nem todas as entradas possuem o campo notas.

**Figura 10 – Banco de alunos no MongoDB**

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

#	Alunos	nome String	_id ObjectID	data_nascimento String	curso Object	habilidades Mixed	notas String
1	"Felipe"	ObjectID("61e08745525fe7e350..")	"1994/02/26"	{ } 1 fields	[ ] 3 elements	"[10.0, 9.0, 4.5]"	
2	"Julio"	ObjectID("61e08c3d525fe7e350..")	"1972/09/30"	{ } 1 fields	[ ] 1 elements	No field	
3	"Alberto"	ObjectID("61e09083525fe7e350..")	"1972/10/30"	{ } 1 fields	[ ] 1 elements	No field	
4	"Daniela"	ObjectID("61e09085525fe7e350..")	"1995/09/30"	{ } 1 fields	[ ] 1 elements	No field	
5	"Jonas"	ObjectID("6294bf51f93ae1867b4..")	"1990/05/12"	{ } 1 fields	[ ] 3 elements	"[10.0, 10.0, 8.7, 9.2]"	
6	"Adriana"	ObjectID("6294c338f93ae1867b4..")	"1999/12/01"	{ } 1 fields	[ ] 2 elements	"[8.0, 9.0]"	
7	"Marshall"	ObjectID("6294c432f93ae1867b4..")	"1991/03/30"	{ } 1 fields	[ ] 3 elements	"[10.0, 10.0, 10.0, 5.0]"	
8	"Bruna"	ObjectID("6294fb09f93ae1867b4..")	"1998/07/15"	{ } 1 fields	[ ] 2 elements	No field	
9	"Emerson"	ObjectID("6294fbb2f93ae1867b4..")	"1998/11/05"	{ } 1 fields	**	"[6.0, 9.0, 7.0, 5.0]"	
10	"Vanessa"	ObjectID("6294fd18f93ae1867b4..")	"1998-02-10"	{ } 1 fields	[ ] 1 elements	No field	

Fonte: Autoria própria

Para esse mesmo banco de dados, porém feito no MySQL, vemos na figura seguinte que todas as pessoas possuem o dado "notas", entretanto alguns deles estão em branco.

**Figura 11 – Tabela Pessoas no MySQL**

	id_pessoa	nome	data_nascimento	curso	notas
▶	1	Felipe	1994-02-26	Sistemas da Informação	[10, 9, 4.5]
	2	Daniela	1995-09-30	Moda	
	3	Jonas	1990-05-12	Engenharia Aeronautica	[10, 10, 8.7, 9.2]
	4	Adriana	1999-12-01	Advocacia	[8, 9]
	5	Marshall	1991-03-30	Música	[10, 10, 10, 5]
	6	Bruna	1998-07-15	Jogos Digitais	
	7	Emerson	1998-11-05	Enfermagem	[6, 9, 7, 5]
	8	Vanessa	1998-02-10	Biologia	
	9	Alberto	1972-10-30	Engenharia Química	
	10	Julio	1972-09-30	Medicina	

Fonte: Autoria própria

Outra diferença que podemos citar é que no MongoDB a base de dados foi feita com o campo "curso" e "habilidades" que estão definidos como *arrays*, ou seja, eles têm um conjunto de dados dentro deles. Isso não é possível de ser feito no banco de dados relacional de uma maneira direta, para representar um *array* no MySQL temos que criar mais de uma tabela, relacionando com a tabela principal (que no caso do exemplo desse trabalho é a tabela Pessoas) através de uma chave estrangeira e posteriormente executar uma query com o comando join para juntar as tabelas.

No exemplo criado para esse trabalho, podemos ver na figura seguinte que o campo "habilidades" possui dois dados, o nome e o nível.

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

**Figura 12 – Exemplo de documento no MongoDB**

```
_id: ObjectId('6294c432f93ae1867b437390')
nome: "Marshall"
data_nascimento: "1991/03/30"
v curso: Object
  nome: "Música"
v habilidades: Array
  v 0: Object
    nome: "inglês"
    nível: "nativo"
  v 1: Object
    nome: "rap"
    nível: "profissional"
  v 2: Object
    nome: "piano"
    nível: "profissional"
notas: "[10.0, 10.0, 10.0, 5.0]"
```

---

Fonte: Autoria própria

Para conseguir replicar isso no MySQL, além da tabela Pessoas, foram criadas mais uma tabela denominada Habilidades.

**Figura 16 – Tabela Habilidades no MySQL**

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

	id_habilidade	habilidade	nivel	id_pessoa
▶	1	inglês	avançado	1
	2	taekwondo	básico	1
	3	guitarra	profissional	1
	4	alemão	básico	2
	5	alemão	avançado	3
	6	natação	avançado	3
	7	piano	profissional	3
	8	inglês	fluyente	4
	9	yoga	avançado	4
	10	inglês	nativo	5
	11	rap	profissional	5
	12	piano	profissional	5
	13	design digital	avançado	6
	14	javascript	profissional	6
	15	botânica	profissional	8
	16	inglês	intermediário	9
	17	inglês	avançado	10

Fonte: Autoria própria

Como podemos ver, foi preciso entrar com uma linha de dados para cada habilidade de cada pessoa, contendo o nome da habilidade, o nível e a chave estrangeira "id\_pessoa" para podermos relacionar as habilidades às pessoas. Em seguida, para conseguirmos um resultado que mostre todos os dados como na figura 17, devemos executar o seguinte comando de busca no MySQL:

```
"select Pessoas.id_pessoa, nome, data_nascimento, curso, notas, habilidade, nivel from Pessoas left join Habilidades on Habilidades.id_pessoa = Pessoas.id_pessoa;"
```

Após a execução dessa busca, que irá "juntar" as tabelas, devemos obter o seguinte resultado:

**Figura 17 – Resultado do join no MySQL**

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

id_pessoa	nome	data_nascimento	curso	notas	habilidade	nivel
1	Felipe	1994-02-26	Sistemas da Informação	[10, 9, 4.5]	inglês	avançado
1	Felipe	1994-02-26	Sistemas da Informação	[10, 9, 4.5]	taekwondo	básico
1	Felipe	1994-02-26	Sistemas da Informação	[10, 9, 4.5]	guitarra	profissional
2	Daniela	1995-09-30	Moda		alemão	básico
3	Jonas	1990-05-12	Engenharia Aeronautica	[10, 10, 8.7, 9.2]	alemão	avançado
3	Jonas	1990-05-12	Engenharia Aeronautica	[10, 10, 8.7, 9.2]	natação	avançado
3	Jonas	1990-05-12	Engenharia Aeronautica	[10, 10, 8.7, 9.2]	piano	profissional
4	Adriana	1999-12-01	Advocacia	[8, 9]	inglês	fluente
4	Adriana	1999-12-01	Advocacia	[8, 9]	yoga	avançado
5	Marshall	1991-03-30	Música	[10, 10, 10, 5]	inglês	nativo
5	Marshall	1991-03-30	Música	[10, 10, 10, 5]	rap	profissional
5	Marshall	1991-03-30	Música	[10, 10, 10, 5]	piano	profissional
6	Bruna	1998-07-15	Jogos Digitais		design digital	avançado
6	Bruna	1998-07-15	Jogos Digitais		javascript	profissional
7	Emerson	1998-11-05	Enfermagem	[6, 9, 7, 5]	NULL	NULL
8	Vanessa	1998-02-10	Biologia		botânica	profissional
9	Alberto	1972-10-30	Engenharia Química		inglês	intermediário
10	Julio	1972-09-30	Medicina		inglês	avançado

Fonte: Autoria própria

Apesar de obtermos os dados que desejávamos, vemos que muitas informações se repetem, como por exemplo o nome, data de nascimento, curso e as notas das pessoas, algo que pode tornar o entendimento da tabela mais difícil e pesado.

Além disso, como foi dito mais cedo nesse trabalho, a utilização da interface gráfica do Mongo, o MongoDB Compass, traz uma facilidade para salvar a base de dados, replicá-la, fazer análises rápidas do tipo e quantidade de dados presente na base de dados e também é possível criar validações. As validações são uma funcionalidade que pode limitar os dados que podem ser inseridos na base de dados, tornando-a mais estruturada e próxima de um banco de dados relacional.

A sequência das figuras 18 a 25 demonstram no passo a passo de como fazer para salvar a base de dados do MongoDB em um único documento JSON que pode ser exportado e reutilizado em outra base de dados.

**Figura 18 – Botão Export Collection**

## Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

MongoDB Compass - localhost:27017/TCC.Alunos

Connect View Help

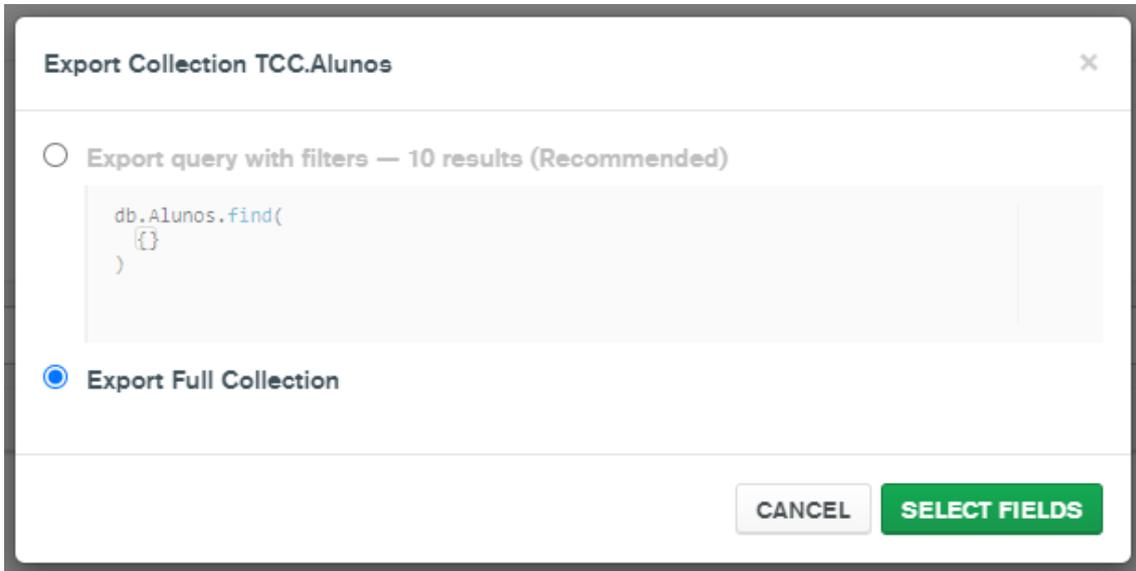
The screenshot shows the MongoDB Compass interface for the 'TCC.Alunos' collection. The left sidebar displays the database structure, including 'localhost:27017', '8 DBS', '9 COLLECTIONS', and a tree view with 'TCC' expanded to show 'Alunos'. The main area shows the 'Documents' tab with three documents:

```
{ "_id": ObjectId('61e087455525fe7e350df488'),  
  "nome": "Felipe",  
  "data_nascimento": "1994/02/26",  
  "curso": Object,  
  "habilidades": Array,  
  "notas": "[10.0, 9.0, 4.5]" }  
  
{ "_id": ObjectId('61e08c3d5525fe7e350df48e'),  
  "nome": "Julio",  
  "data_nascimento": "1972/09/30",  
  "curso": Object,  
  "habilidades": Array }  
  
{ "_id": ObjectId('61e090035525fe7e350df48f'),  
  "nome": "Alberto",  
  "data_nascimento": "1972/10/30",  
  "curso": Object,  
  "habilidades": Array }
```

Fonte – Autoria Própria

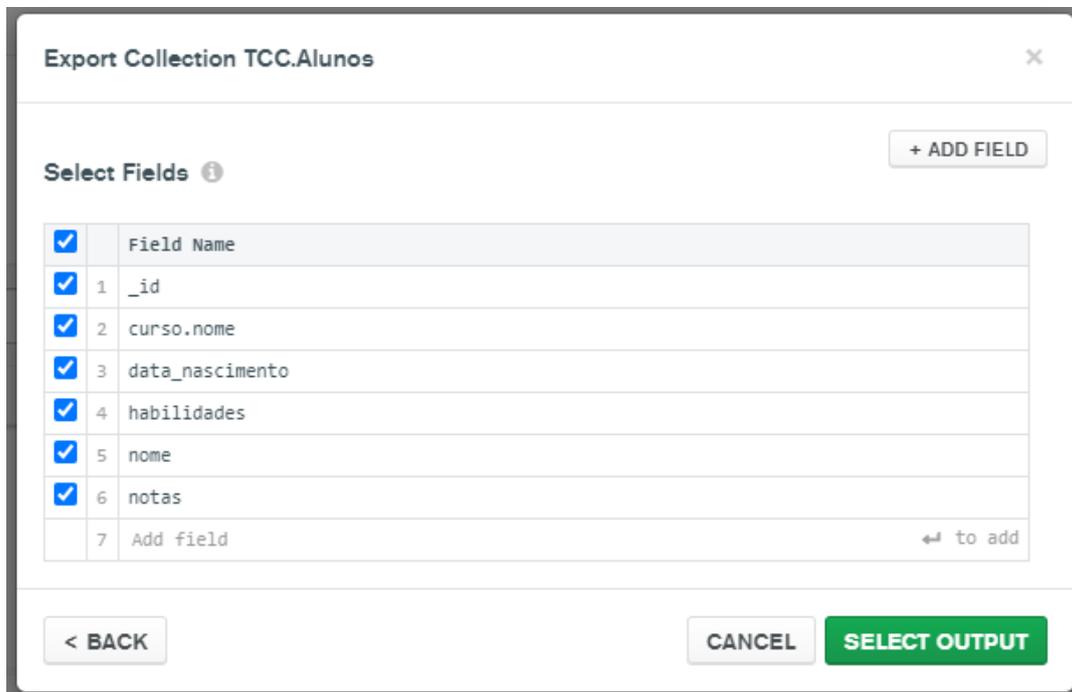
Figura 19 – Escolha do conteúdo

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**



Fonte – Autoria Própria

**Figura 20 – Escolha dos campos**



Fonte – Autoria Própria

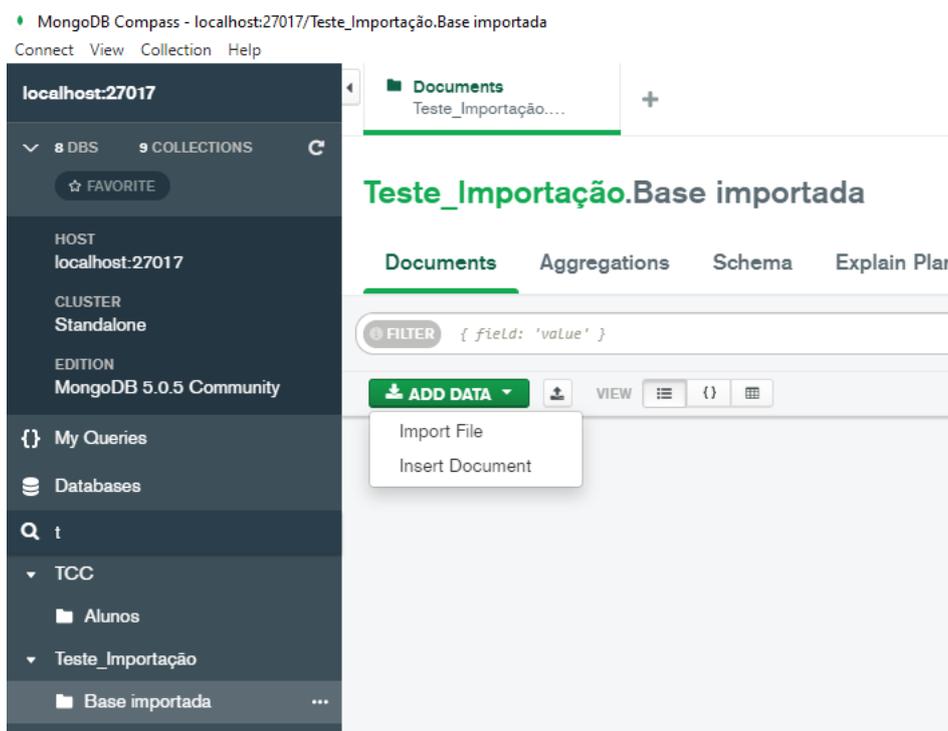
**Figura 21 – Escolha do tipo de arquivo e diretório**

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**



Fonte – Autoria Própria

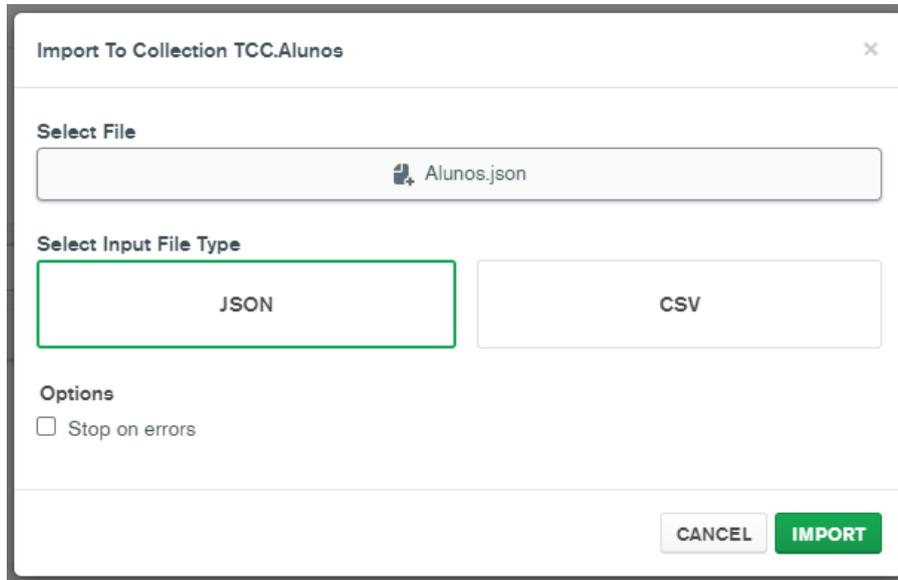
**Figura 22 – Importando uma base de dados**



Fonte – Autoria Própria

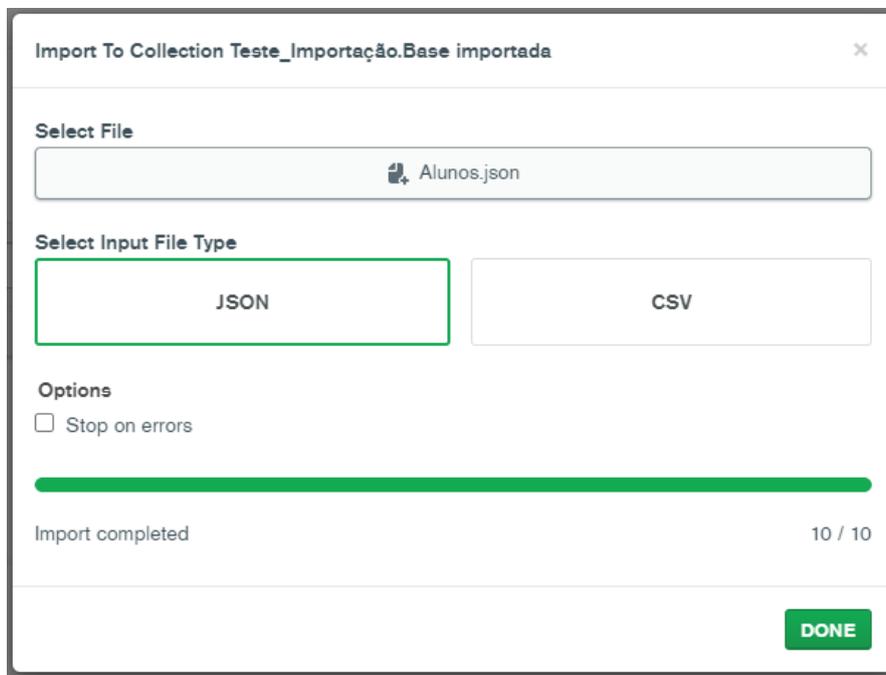
**Figura 23 – Escolhendo o arquivo**

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**



Fonte – Autoria Própria

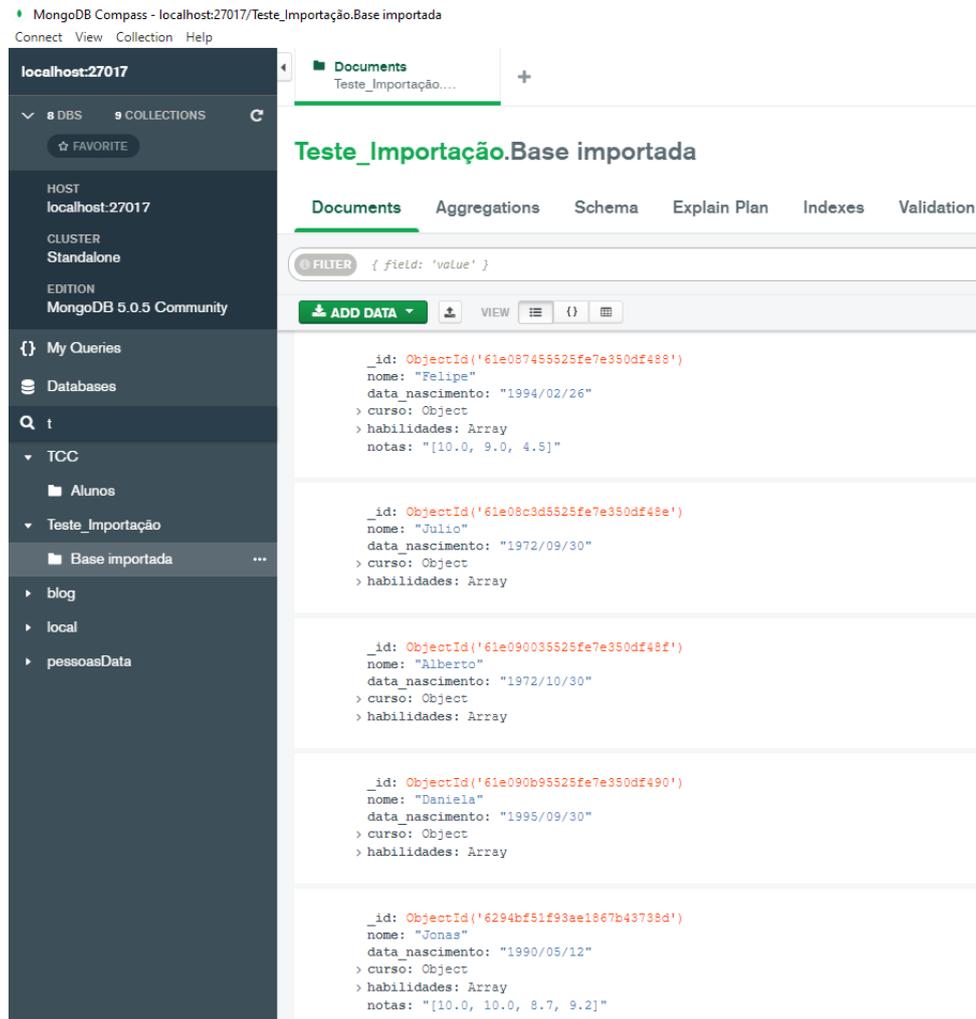
**Figura 24 – Importação completa**



Fonte – Autoria Própria

**Figura 25 – Base importada**

## Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"



MongoDB Compass - localhost:27017/Teste\_Importação.Base importada

Connect View Collection Help

localhost:27017

DBS COLLECTIONS

FAVORITE

HOST  
localhost:27017

CLUSTER  
Standalone

EDITION  
MongoDB 5.0.5 Community

My Queries

Databases

t

TCC

Alunos

Teste\_Importação

Base importada

blog

local

pessoasData

Documents  
Teste\_Importação...

### Teste\_Importação.Base importada

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

```
{
  "_id": ObjectId("61e087455525fe7e350df488"),
  "nome": "Felipe",
  "data_nascimento": "1994/02/26",
  "curso": Object,
  "habilidades": Array,
  "notas": "[10.0, 9.0, 4.5]"
}
```

```
{
  "_id": ObjectId("61e08c3d5525fe7e350df49e"),
  "nome": "Julio",
  "data_nascimento": "1972/09/30",
  "curso": Object,
  "habilidades": Array
}
```

```
{
  "_id": ObjectId("61e09035525fe7e350df48f"),
  "nome": "Alberto",
  "data_nascimento": "1972/10/30",
  "curso": Object,
  "habilidades": Array
}
```

```
{
  "_id": ObjectId("61e090b95525fe7e350df490"),
  "nome": "Daniela",
  "data_nascimento": "1995/09/30",
  "curso": Object,
  "habilidades": Array
}
```

```
{
  "_id": ObjectId("6294bf51f93ae1867b43738d"),
  "nome": "Jonas",
  "data_nascimento": "1990/05/12",
  "curso": Object,
  "habilidades": Array,
  "notas": "[10.0, 10.0, 8.7, 9.2]"
}
```

Fonte – Autoria Própria

Para exportar a base de dados, vemos que basta clicar no botão export collection que irá abrir uma nova janela na qual devemos escolher quais documentos queremos exportar através de uma busca ou se desejamos exportar toda a base. Em seguida podemos escolher quais os campos dos documentos escolhidos vamos exportar, o que nos leva a tela seguinte, na qual deve-se escolher o tipo do documento (JSON ou CSV) e o local no qual deseja salvar o arquivo. Então basta clicar no botão “export” e a base de dados estará salva em um arquivo no formato escolhido.

## Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

Feito isso, para importar a base que foi salva em um outro banco, deve-se selecionar o banco desejado no menu esquerdo e depois clicar no botão "Add Data" e selecionar "Import File". Em seguida basta escolher o arquivo que se deseja importar e clicar no botão "Import". Caso não haja nenhum erro, a importação estará completa.

Utilizando a base de dados de alunos que foi mostrada na figura 13, podemos criar uma validação para que futuras adições a base de dados sejam feitas igualmente aos dados já existentes. Um exemplo de validação pode ser visto na figura 23 a seguir:

**Figura 26 – Exemplo de validação**

### TCC.Alunos

Documents Aggregations Schema Explain Plan Indexes **Validation**

Validation Action **ERROR** Validation Level **STRICT**

```
1 {  
2   $jsonSchema: {  
3     required: ["nome", "data_nascimento", "curso", "habilidades", "notas"],  
4     properties: {  
5       nome: {  
6         bsonType: "string",  
7         description: "deve ser uma string"  
8       },  
9       data_nascimento: {  
10        bsonType: "string",  
11        description: "deve ser uma string"  
12      },  
13      notas: {  
14        bsonType: "string",  
15        description: "deve ser uma string"  
16      }  
17    }  
18  }  
19 }
```

Fonte – Autoria Própria

## Faculdade de Tecnologia de Americana "Ministro Ralph BIASI"

Nesse exemplo, tornamos necessário que todos os documentos que forem inseridos na base de dados contenham os campos nome, data\_nascimento, curso, habilidades e notas. Além disso o conteúdo dos campos nome, data\_nascimento e notas devem ser do tipo string. Ao escrever o código de validação, o Compass já mostra dentro da sua base de dados um exemplo de um documento que passou pela validação e um que não passou, como mostra a figura 27 a seguir:

**Figura 27 – Documento validado X não validado**

✔ **Sample Document That Passed Validation**

```
_id: ObjectId('61e087455525fe7e350df488')
nome: "Felipe"
data_nascimento: "1994/02/26"
> curso: Object
> habilidades: Array
notas: "[10.0, 9.0, 4.5]"
```

✘ **Sample Document That Failed Validation**

```
_id: ObjectId('61e08c3d5525fe7e350df48e')
nome: "Julio"
data_nascimento: "1972/09/30"
> curso: Object
> habilidades: Array
```

Fonte – Autoria Própria

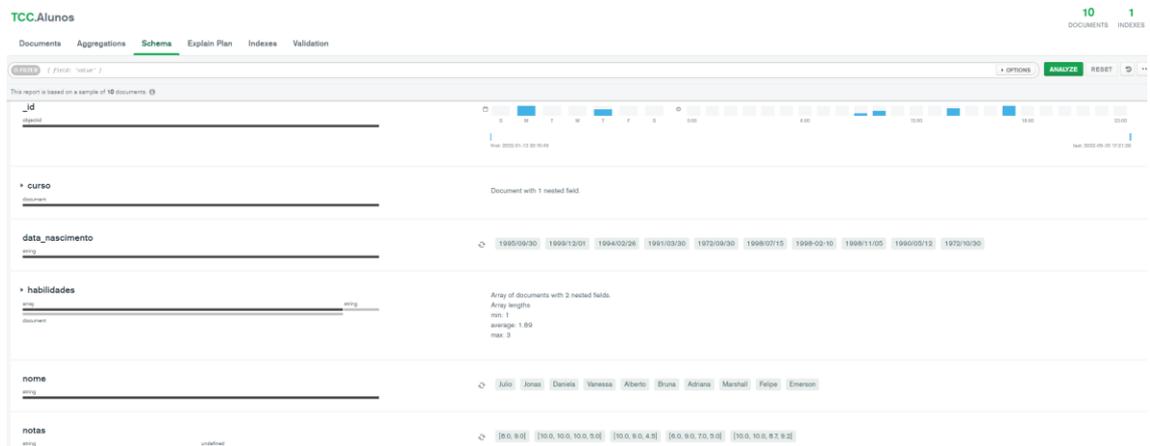
Nesse caso, vemos que um dos documentos da base de dados em questão não foi validado pois não apresenta o campo notas.

Dessa maneira, com poucas linhas de código conseguimos garantir uma estrutura a nossa base de dados bem similar aos bancos de dados relacionais.

## Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

Outra funcionalidade do Compass é a análise que ele apresenta das *collections* presentes no seu banco de dados. A figura 25 mostra a página de análise do Compass.

**Figura 28 – Exemplo de análise do Compass**



Fonte – Autoria Própria

Essa análise traz todos os campos presentes na base de dados tanto quanto o tipo de variável e no lado direito também podemos ver que ele mostra algumas entradas de dados dos campos correspondentes. Além disso, na parte superior direita o Compass mostra alguns gráficos de barras que mostram em quais dias o banco foi utilizado e em quais horários. Essa funcionalidade é bem interessante e é algo que não encontramos nos bancos de dados tradicionais.

### 3. CONCLUSÃO

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

A utilização do MongoDB em substituição do Banco de Dados Relacional é uma alternativa a qual oferece diversas vantagens tendo em vista que este modelo não relacional é um banco de dados orientado a documentos, livre de estruturas, possui código aberto, é multiplataforma e é escrito em linguagem C++.

Por ser um banco de dados NoSQL, ele permite a utilização de documentos que são semelhantes ao JSON para armazenar os dados e suas características permitem que todas as aplicações possam modelar informações de forma muito mais natural, tendo em vista que seus dados são alinhados em hierarquias complexas que podem continuar a serem indexadas e tornam as informações fáceis de buscar. Essas características favorecem a utilização do MongoDB na área de ciência de dados e Big Data, que trabalham com uma quantidade muito grande de dados e, portanto, se beneficiam da velocidade de busca do Mongo e da falta de uma estrutura definida para cada documento.

O Mongo DB oferece também o recurso da replicabilidade. Como todos os dados são armazenados em documentos, fica fácil de criar cópias da base de dados para garantir que a informação fique segura caso algo aconteça com a base principal. Os dados são gravados em todas as bases em tempo real e o próprio sistema determina qual se comporta como base primária e quais são as secundárias.

Enfim, pode-se dizer que o objetivo do estudo foi atingido, pois pudemos verificar que há muitas vantagens na utilização do MongoDB como alternativa aos bancos de dados tradicionais, mas que essas vantagens também são situacionais e que para cada projeto que se for desenvolver, deve ser ponderado qual o tipo de banco de dados mais apropriado para a ocasião e, portanto, concluímos que o MongoDB não substituí completamente a utilização de bancos de dados relacionais.



---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

#### **4. REFERÊNCIAS BIBLIOGRÁFICAS**

BSON. Binary **JSON**. Disponível em: <http://bsonspec.org/>.

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**

DATE, C. J.. **Introdução a Sistemas De Bancos De Dados**. 8. ed. Rio de Janeiro: Elsevier, 2003.

ELMASRI, RAMEZ; NAVATHE, SHAMKANT B.. **Sistemas de banco de dados**. 6. ed. São Paulo: Addison Wesley, 2011.

FACHIN, ODÍLIA. **Fundamentos de Metodologia**. 3. ed. São Paulo: Saraiva, 2001

GAUDÊNCIO, EMERSON. **Conceituando banco de dados e SGBD**. Disponível em: <https://cooperati.com.br/2012/08/banco-de-dados-conceituando-banco-de-dados-e-sgbd/>.

GUIMARÃES, CÉLIO (2003) **Fundamentos de Bancos de Dados: Modelagem, Projeto e Linguagem SQL**. Editora UNICAMP, 1a.

JSON. **Introducing JSON**. Disponível em <http://www.json.org/>.

MONGODB. **MongoDB** Disponível em: <https://www.mongodb.com/>.

SADALAGE, PRAMOD J. **NoSQL: um guia conciso para o mundo emergente da persistência poliglota**. Novatec. Junho de 2013.

SILBERSCHATZ, ABRAHAM; KORTH, HENRY F.; SUDARSHAN, S.. **Sistema De Banco De Dados**. 3. ed. São Paulo: Makron Books, 1999.

SUISSA, J. **Introdução ao NoSQL**. NoSQL Br. 2010. Disponível em: <http://www.nosqlbr.com.br/introducao-ao-nosql.html>.

RANKING DOS SGBDS MAIS UTILIZADOS. Disponível em <https://db-engines.com/en/ranking>.

CONECTIVIDADE PYTHON-MONGODB. Disponível em: <https://ronanlopes.me/python-com-mongodb-nosql-getting-started/#:~:text=Novamente%2C%20para%20estabelecer%20uma%20conex%C3%A3o%20com%20o%20MongoDB,2%20from%20pymongo%20import%20MongoClient%20client%20%3D%20MongoClient%28%22mongodb%3A%2F%2Flocalhost%3A27017%2F%22%29>