# Unidade 7

# BANCO DE DADOS NÃO ESTRUTURADO



Nessa unidade você conhecerá os conceitos associados as características entre dados estruturados, semiestruturados e não estruturados. Conhecerá sobre os bancos de dados que trabalham com dados não estruturados, denominados NoSQL, os tipos e características de cada modelo.

# UNIDADE 7 – BANCO DE DADOS NÃO ESTRUTURADO

## 7.1 Introdução

Antes de falarmos de **dados não estruturados** vamos rever o conceito dos **dados estruturados** e os **semiestruturados**, pois entender a diferença entre eles pode ser determinante para seu projeto. De antemão vale registrar que há discussões sobre a validade dessa classificação, pois mesmo os dados ditos não estruturados possuem algum tipo de estruturação própria.

## 7.2 Dados Estruturados

O dado é dito estruturado quando contém uma organização para serem recuperados. São como etiquetas, linhas e colunas que identificam pontos sobre aquela informação que facilitam o trabalho sobre eles. São utilizados pela maioria das empresas e embora os dados produzidos ao longo do tempo sejam maiores do que os dados que são representados, certamente são eles os utilizados para a tomada de decisão em muitas organizações há muitos anos. Uma forma de organizar os dados é através de uma estrutura que se assemelha a uma planilha Excel, na forma de linhas e colunas, mas pode variar de acordo com a fonte de dados.

Esses dados possuem estruturas bem definidas, rígidas, pensada antes da existência do conteúdo que irá povoar aquela estrutura, conforme pode ser observado na Figura 7.1. Por serem pré-definidas, cada dado só consegue armazenar um conteúdo compatível com a estrutura definida para ele, por exemplo, um dado definido como numérico não aceitará conteúdo que fira essa propriedade, tal como um texto.

A análise dos dados estruturados possui uma facilidade implícita na sua existência, já que a sua estrutura não muda com frequência e que os dados carregados seguem padrões predeterminados. Dessa maneira a análise não requer avançadas técnicas de interpretação ou conhecimentos estatísticos. No entanto, isso não quer dizer que tais dados não tenham complexidade no seu conteúdo. Alguns exemplos de dados estruturados são as planilhas eletrônicas (Excel), banco de dados, arquivo XML e CSV.

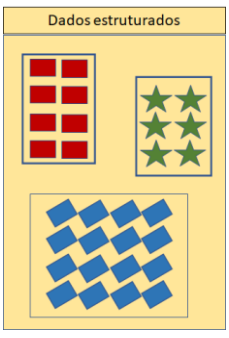

Figura 7.1 – Dados estruturados.

Dados estruturados geralmente residem em *Relational Data Base System* – RDBMS – e são gerados tantos por humanos quanto por máquinas desde que sejam criados dentro de uma estrutura relacional. Para manusear esses dados, utiliza-se *Structured Query Language* – SQL.

## 7.3 Dados semiestruturados

Um meio termo entre estruturados e não estruturados, os semiestruturados apresentam representação estrutural heterogênea, não sendo nem estritamente tipados, nem completamente não-estruturados. Nesses dados são chamados de auto-descritivos, pois o esquema de representação está presente de forma explícita ou implícita juntamente com o dado. Para que a estrutura seja identificada e extraída, primeiramente deve-se fazer uma análise do dado.

Como principais características de dados semiestruturados temos:

1. **Definição a posteriori:** A definição do esquema é feita depois que os dados são analisados, tomando por base uma investigação de suas estruturas particulares e das semelhanças e diferenças entre eles.
2. **Estrutura irregular:** Como não existe um esquema padrão, pode-se encontrar coleções extensas de dados semanticamente semelhantes, porém organizados de maneiras diferentes, podendo algumas ocorrências terem informações incompletas ou adicionais em relação as outras. Um *currículum vitae* é um exemplo.
3. **Estrutura implícita:** Muitas vezes existe uma estrutura básica, mas que está implícita na forma como os dados são representados e com isso é necessário realizar uma computação para obter a estrutura.
4. **Estrutura parcial:** Apenas parte dos dados pode ter uma estrutura, seja implícita ou explícita. Por exemplo, componentes de objetos que são arquivos *bitmaps* são não-estruturados, já os dados pessoais podem ter uma estrutura implícita e explícita.
5. **Estrutura extensa:** a ordem da magnitude de uma estrutura para esses dados é grande, uma vez que eles são muito heterogêneos. Por exemplo, supondo diferentes formatos para um *currículum vitae*, uma união dos atributos relevantes para cada formato pode gerar um esquema muito extenso.
6. **Estrutura evolucionária:** tanto os dados quanto sua a estrutura, se modificam com muita frequência.
7. **Estrutura descritiva e não prescritiva:** também chamada de estrutura indicativa, indica que não é possível prescrever esquemas fechados e muitas restrições de integridade com relação à semântica dos atributos.
8. **Distinção entre estrutura e dados não é clara:** a distinção lógica entre **estrutura** e **valor** não é sempre clara, já que a estrutura está embutida na descrição do dado.

Como exemplo de dados semiestruturados temos:

1. **Linguagem de marcação XML**: conjunto de regras de codificação de documentos, orientado por tags, que define um formato legível tanto para humanos quanto para máquinas.
2. **JSON (*Java Script Object Notation*) de padrão aberto:** formato de intercâmbio de dados semiestruturado reconhecido não só pelo Java. Está estruturado em pares nome/valor (ou objeto / tabela hash) e uma lista de valores ordenada (ou matriz, sequência, lista). Por ser uma estrutura intercambiável entre as linguagens, o JSON se destaca na transmissão de dados entre aplicativos da Web e servidores.
3. **Dados semiestruturados NoSQL:** nesses bancos de dados não há separação da organização (esquema) dos dados o que o torna a melhor alternativa para armazenar informações que não cabem facilmente no formato de registro e tabela como textos com comprimento variável.

A Figura 7.2 compara características entre dados estruturados e semiestruturados.

| | Dados estruturados | Dados semiestruturados |
|---|---|---|
| Esquema | Pré-definido | Nem sempre há um esquema pré-definido |
| Estrutura | Regular | Irregular |
| | Independente do dado | Embutida no dado |
| | Reduzida | Extensa |
| | Fracamente evolutiva | Fortemente evolutiva |
| | Prescritiva | Descritiva |
| Distinção entre estrutura e dado | Clara | Não é clara |

Figura 7.2 – Dados estruturados e semiestruturados
Fonte: Adaptado de https://www.ime.usp.br/~jef/semi-estruturado.pdf

## 7.4 Dados não estruturados

Nem todas as informações possíveis podem estar contidas dentro dos bancos de dados. Além disso, os dados estruturados precisam ser preenchidos para que o uso deles se dê de forma automatizada. No entanto, documentos de texto, não são vistos na plenitude. Muitas informações contidas nesses documentos podem não estar sendo devidamente analisados por falta de condições de percorrer o seu conteúdo, justamente pela dificuldade que seria classificar cada palavra do texto e relacioná-las com contextos, momentos, pessoas ou citações. Pense em todas as palavras contidas em um bloco de notas, um e-mail ou um documento de um editor de texto e verá a dificuldade de se relacioná-las a algum contexto significativo. Soma-se a isso a produção oriunda das redes sociais, onde as próprias emoções estão contidas nas manifestações de seus usuários. Essa incapacidade é ainda maior em dados contidos em vídeos ou áudios.

Nesse contexto fica evidente que esses dados não estão estruturados, conforme ilustra a Figura 7.3. Possuem estruturas indefinidas, desalinhadas, não padronizadas, podendo ser compostos por diversos elementos diferentes. São gerados e serão consumidos de maneira diferente daqueles utilizados em dados estruturados.

Dados estruturados estão atrelados a um contexto o que facilita a extração de informações, porém, somente 10% dos dados gerados no mundo são estruturados. O problema reside em como extrair informações em dados não estruturados, sem um contexto específico ou com estruturas conhecidas. Analisar dados não estruturados requer um esforço maior, pois são mais complexos, naturalmente. Requer ensinar a máquina (*machine learning*) a compreender, interpretar e calcular as características ou padrões que se deseja encontrar. E é na aprendizagem da máquina de transformar uma equação matemática em algo interpretável, replicável e com acurácia suficiente, que reside a complexidade de analisar os dados.

Nesse universo de análise de dados há uma máxima que diz que 80% dos dados tanto nos nossos dispositivos pessoas quanto em soluções empresariais, são não estruturados, ou seja, há uma predominância desses dados, gerados por todos nós, em nosso cotidiano, de análise complexa o que os leva a serem considerados ativos valiosos para as empresas.

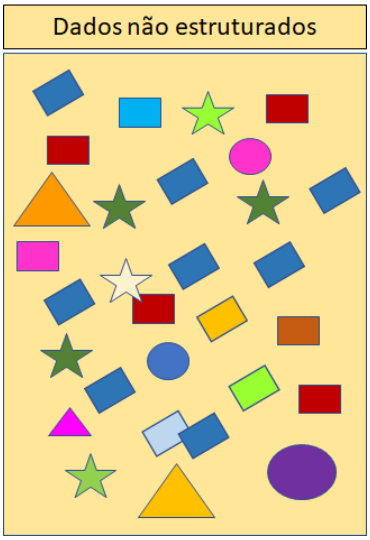


Figura 7.3 – Dados não estruturados.

Os dados podem ser armazenados em um Sistema de Banco de Dados não Relacional, como o NoSQL, que será abordado em mais detalhes na seção **7.5 – NoSQL**. Assim como os dados estruturados, os dados não estruturados podem ser gerados por humanos ou por máquinas.

Os dados não estruturados gerados por humanos são:

1. **Textos:** logs, documentos de texto, planilhas, apresentações;
2. **E-mail:** Devido a certa estrutura interna proveniente dos seus metadados, as vezes são considerados semiestruturados. No entanto, como seu campo de mensagem não é estruturado e difícil de ser analisado pelas ferramentas analíticas tradicionais, ele é então considerado não estruturado;

3. **Dados móveis:** mensagens de texto e de localização;
4. **Mídia social:** dados do Facebook, LinkedIn, Twitter;
5. **Site:** YouTube, Instagram, Pinterest, site de compartilhamento de fotos;
6. **Mídia:** vídeo, Áudio, MP3, fotos digitais;
7. **Comunicações:** mensagens instantâneas, gravações de telefone, software de colaboração, bate-papo.

Os dados não estruturados gerados por máquinas são:

1. **Imagens de satélite:** dados meteorológicos, formas terrestres, movimentos militares, relevo;
2. **Dados científicos:** exploração de petróleo e gás, exploração espacial, imagens sísmicas, dados atmosféricos.
3. **Vigilância digital:** tráfego, clima, sensores oceanográficos, segurança.

A Figura 7.4 faz uma comparação entre dados estruturados e não estruturados.

| | Dados estruturados | Dados não estruturados |
|---|---|---|
| Características | - Modelo de dados pré-definido.<br>- Usualmente somente texto<br>- Fácil de consultar | - Modelo de dados não pré-definido.<br>- Podem ser texto, vídeo, som, imagens ou outros formatos.<br>- Dificuldade na pesquisa. |
| Armazenamento | - Banco de Dados Relacional<br>- *Data Warehouses* | - Aplicações<br>- Banco de Dados NoSQL<br>- *Data Warehouses*<br>- *Data Lakes* |
| Gerados por | - Humanos e máquinas | - Humanos e máquinas |
| Aplicação típica | - Sistema de reserva de passagens.<br>- Controle de estoque.<br>- Sistemas CRM.<br>- Sistemas ERP. | - Processamento de palavras.<br>- Software de apresentação.<br>- e-mail de clientes<br>- Ferramentas de visualização ou edição de mídias. |
| Exemplos | - Datas<br>- Numero de telefone<br>- Número do seguro social<br>- Número do cartão de crédito<br>- Nome do cliente<br>- Endereço<br>- Número e nome do produto<br>- Informações de transações | - Arquivos de texto<br>- Relatórios<br>- Mensagens de e_mail<br>- Arquivos de áudio<br>- Arquivos de vídeo<br>- Imagens<br>- Imagens de vigilância |

Figura 7.4 – Dados estruturados e não estruturados
Fonte: Traduzido de https://www.datamation.com/big-data/structured-vs-unstructured-data.html.

Hoje, acredita-se que 80% de todos os dados relevantes para os negócios são não estruturados, e dentre eles, em especial os de texto, o que se explica pelo crescente interesse de se analisar esse tipo de dado. Parece natural que um percentual muito alto dos dados não seja estruturado, mas o que nos leva a acreditar que um percentual grande desses é proveniente de textos. Isso se explica naturalmente se pararmos para analisar o tempo que dedicamos produzindo ou lendo e-mails, relatórios, artigos ou similares, ouvindo áudio ao vivo ou gravado.

O trabalho relacionado à linguística computacional através da qual cientistas da computação que trabalham com Inteligência Artificial tem como alvo a linguagem natural existe há décadas, mas somente da década de 90 a

mineração de textos começou a emergir como disciplina acadêmica. Somente após o potencial apresentado pelo sistema Eliza em 1964, de Joseph Weisenbaunn, que aplicou correspondência de padrões básicos e regras linguísticas para imitar um diálogo entre um psicoterapeuta e seu paciente de forma inteligente, é que as tecnologias de texto em nível comercial surgiram. No entanto, deve-se aplicar técnicas de extração de informações de forma a extrair conteúdos úteis, pois nesse montante de textos há uma quantidade significativa de lixo linguístico.

## 7.5 NoSQL

Devido as demandas por escalabilidade cada vez mais frequentes e, também, pela natureza estruturada do Modelo Relacional, os desenvolvedores começaram a perceber as dificuldades em se organizar os dados desse modelo em um sistema distribuído trabalhando com particionamento de dados. Nesse ponto as soluções não-relacionais ganham foco. Uma emergente e popular classe de bancos de dados não-estruturados chamados *not only SQL* – NoSQL.

Bancos de dados não relacionais possuem as seguintes características em comum:

1. **Escalabilidade:** refere-se à capacidade de gravar dados em múltiplos armazéns de dados simultaneamente sem levar em conta a limitação física da infraestrutura.
2. **Modelo de consulta e dados:** no lugar de linhas, colunas, chaves, bancos não relacionais usam frameworks especializadas para armazenar dados com um conjunto de APIs para acessar os dados de maneira inteligente.
3. **Design de persistência:** a persistência ainda é um elemento crítico. Devido à alta velocidade, variedade e volume do *big data*, esses bancos de dados usam diferentes mecanismos para a persistência de dados.
4. **Diversidade de interface:** a maioria suporta APIs RESTful como o ponto de partida de interface, além de oferecer uma variedade de mecanismos de conexão para programadores e administradores incluindo ferramentas de análise, relatórios e visualizações.
5. **Consistência eventual:** enquanto os RDBMS usam ACID (*Atomicity, Consistency, Isolation, Durability*) como mecanismo de garantia de consistência nos dados, bancos não relacionais usam BASE (*Basically Available, Soft state, and Eventual Consistency)*. Destes a Consistência Eventual é mais importante porque é responsável pela resolução de conflitos quando dados estão em movimento entre nós de uma aplicação distribuída.

O modelo de dados dominante nas últimas décadas é o modelo relacional, que é visualizado como um conjunto de tabelas, composto de linhas, que representa uma entidade de interesse, cuja descrição é feita pelas características definidas através de colunas. Cada coluna é de um tipo único, armazena um único valor e pode conter um conteúdo que propicia se relacionar com outra tabela. No entanto, quando se fala em NoSQL, há um distanciamento do modelo relacional, pois cada solução possui um modelo diferente do outro. As 4 principais

categorias de modelos mais adotados no ecossistema NoSQL são: chave-valor, documento, famílias de colunas e grafos. As três primeiras compartilham uma característica comum em seus modelos de dados o que os denomina como “orientação agregada”.

A **orientação agregada** surge da percepção de que o usuário deseja, frequentemente, trabalhar com dados na forma de unidades com uma estrutura mais complexa do que um conjunto de linhas. Tipo um registro mais complexo que permita que listas e outras estruturas de dados sejam aninhadas dentro dele. Essa característica pode ser observada em bancos de dados chave/valor, de documento e família de colunas. Como não existe um consenso para identificar esse tipo de registro, vamos chamá-lo de **agregado** que vem do *Domain-Driven Design* (Projeto Orientado a Domínio), que considera o agregado como um conjunto de objetos relacionados que é tratado como uma unidade. Isso permite definir a unidade de manipulação e de consistência desse dado.

1. **Banco de dados de par chave/valor:** são de longe os bancos de dados mais simples. Não requerem um esquema como os RDBMS o que oferece grande flexibilidade e escalabilidade. Por outro lado, não oferece suporte as propriedades ACID e por isso requerem que os desenvolvedores pensem em posicionamento de dados, replicação e tolerância a falhas, uma vez que não são expressamente controladas pela tecnologia. Como os dados não são tipados, a maior parte dos dados são armazenados como *string*. Nesse banco de dados o agregado é opaco, sendo apenas um amontoado de bits sem significado, porém, justamente essa opacidade permite armazenar qualquer coisa no agregado, porém o acesso é dependente da sua chave.

   Esses bancos de dados, buscando diminuir a opacidade sobre os seus dados, podem permitir estruturas de dados, como faz o Riak, através do qual é possível adicionar metadados a agregados para indexação e conexões entre interagregados. Já o Redis permite que o agregado seja dividido em listas e conjuntos. Também poderá suportar consultas integrando ferramentas de pesquisa como o Solr.

   Exemplo desse tipo de banco de dados: ***Riak**, Redis, Mencached DB, Berkeley DB, HamsterDB, Amazon DynamoDB (não é open source), Project Voldemort (open source do AmazonDynamoDB)*.

2. **Banco de dados orientado a documento:** existem dois tipos de banco de dados de documento. O **primeiro** é frequentemente descrito como um repositório de conteúdo de estilo de documento completo (arquivos Word, páginas Web completas). O **segundo** é um banco de dados para armazenar componentes de documentos para o armazenamento permanente como uma entidade estática ou para montagem dinâmica das partes do documento. A estrutura do documento e suas partes é fornecido por *JavaScript Object Notation* (JSON) and/or *Binary JSON* (BSON).

   Bancos de dados de documentos são mais comumente utilizados quando você tem que produzir muitos relatórios que precisam ser montados dinamicamente a partir de elementos que mudam frequentemente. Um bom exemplo é o preenchimento completo dos documentos da saúde, onde a

composição do conteúdo irá variar em função do perfil do sujeito (idade, residência, nível de renda), plano de saúde e elegibilidade de programas governamentais.

Diferente do banco de dados orientado de par por chave/valor, nesse banco de dados é possível ver uma estrutura no agregado o que impõe limites sobre o que podemos inserir nele, definindo quais as estruturas e os tipos permitidos. O que torna esse banco mais restritivo proporciona maior flexibilidade de acesso. Diferente do Chave/Valor cujo acesso só pode ser feito pela chave, no banco de dados de documento pode-se submeter consultas baseadas nos campos do agregado, podendo recuperar parte do agregado em vez dele todo e ainda com a possibilidade de criar índices baseando-se no conteúdo do agregado.

Exemplo desse tipo de banco de dados: *CouchDB, **MongoDB**, Terrastore, OrientDB, RavenDB e o LotusNotes*.

3. **Banco de dados orientado a colunas:** banco de dados relacional é orientado a linha uma vez que os dados em cada linha de uma tabela são armazenados juntos. Em um banco de dados orientado a colunas os dados são armazenados em linhas. Embora possa parecer trivial, essa é característica subjacente mais importante dos banco de dados orientado a colunas. É muito fácil adicionar colunas e elas podem ser adicionadas linha a linha, oferecendo grande flexibilidade, performance e escalabilidade. Quando você tem volume e variabilidade de dados, você pode usar esse tipo de banco de dados.

   Exemplo desse tipo de banco de dados são: ***Cassandra**, HBase, Hypertable e AmazonDynamoDB, BigTable*.

Ainda no contexto de banco de dados NoSQL, mas que não são considerados agregados, tem-se:

1. **Banco de dados baseado grafos:** A estrutura fundamental para banco de dados de grafos é chamada de relacionamento de no (*node-relationship*), que é mais utilizada quando você deve lidar com dados altamente conectados. Nodes e relacionamentos suportam propriedades, um par de chave-valor onde o dado é armazenado. O banco de dados é navegado seguindo os relacionamentos. Esse tipo de armazenamento e navegação não é possível num RDBMS devido a rígida estrutura da tabelas e a inabilidade de seguir as conexões entre os dados.

   Exemplo desse tipo de banco de dados são: ***Neo4J,** InfoGrid, FlockDB, HyperGraphDB, InfiniteGraph, OrientDB*.

2. **Banco de dados espacial:** Interage-se com dados espaciais todo dia. Se você usa um smartphone ou *Global Positioning System* (GPS) para obter instruções sobre um lugar específico, ou se você procura pela localização de um restaurante próximo a um endereço físico ou marco, você está usando

aplicações com dados espaciais. Os próprios dados espaciais são padronizados pelos esforços do *Open Geospatial Consortium* (OGC) que estabele o *OpenGIS* (*Geographic Information System*) e uma série de outros padrões para dados espaciais.

Isso é importante, porque os bancos de dados espaciais são implementados do padrão OGC e, uma empresa pode ter necessidades específicas que podem (ou não) ser atendidas pelo padrão. Um banco de dados espacial tornou-se importante quando as empresas começam a usar várias dimensões diferentes de dados para tomar decisões. Por exemplo, um meteorologista fazendo pesquisa pode querer armazenar e analisar dados de um furacão, incluindo temperatura, velocidade do vento e humidade do ar, e o modelar esses resultados em 3 dimensões (3D).

Banco de dados espaciais armazenam dados em objetos de 2, 2,5 e 3 dimensões (2D, 2,5D e 3D respectivamente). Estamos familiarizados com dados em 2D (cumprimento e largura) e em 3D (cumprimento, largura e profundidade). Não estamos familiarizados com uma dimensão 2,5D, pois são tipos especiais de dados espaciais. São objetos 2D com uma elevação extra de 'metade' da dimensão. A maioria dos bancos de dados espaciais 2.5D contém informação de mapeamento e são frequentemente referenciados como *Geographic Information System* (GIS).

Um elemento atômico de um banco de dados espacial são linhas, pontos e polígonos.

3. **Persistência poliglota:** Esse termo quer definir um conjunto de aplicativos que usam várias tecnologias de banco de dados, e esse é o resultado mais provável do seu planejamento de implementação em *Big Data*. Vai ser difícil escolher um estilo de persistência, não importa quão estreita seja sua abordagem para *Big Data*. Um banco de dados de persistência poliglota é usado quando é necessário resolver problemas complexos quebrando esse problema em segmentos e aplicando diferentes modelos de banco de dados.

## Bibliografia

PRAMOD, J. Sadalage, FOWLER, Martin. **NoSQL.** Um Gruia Conciso para o Mundo Emergente da Persistência Poliglota. São Paulo: Novatec, 2013.

HURWITZ, Judith, NUGENT, Alan, HALPER, Dr. Fern, KAUFMAN, Marcia. **Big Data for dummies.** New Jersey: John Wiley & Sons, 2013.

SABHARWAL, Navin, GUPTA EDWARD, Shakuntala. **Big Data, NoSQL. Architecting MongoDB.** Lavergne, TN: 2015.

https://pt.wikibooks.org/wiki/SQL/Dados_Estruturados,_Semi-Estruturados_e_N%C3%A3o_Estruturados

https://www.digitalhouse.com/br/blog/diferenca-dados-estruturados-e-nao-estruturados

http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/

https://www.datamation.com/big-data/structured-vs-unstructured-data.html

https://document.onl/documents/no-sql-558930b5f1ecc.html