INTRODUÇÃO À DADOS

AUTOR: **Renan Moreira da Silva Apolinario**

Revisado por: **Fernandão**

OSASCO – SP

2025

OSASCO – SÃO PAULO

2025

O presente documento tem como intuito a abordagem inicial para aqueles que tem interesse nos estudos como engenheiro de dados. Nesta documentação serão abordados temas categorizados como básicos para aqueles que já exercem a profissão. Caso encontre algum conteúdo incorreto, desatualizado ou incompleto, fique à vontade para me contatar que realizarei a retificação.

É importante lembrar que o mundo da tecnologia está em constante evolução, em decorrência deste fato, os métodos, ferramentas e conhecimento se atualizam a uma velocidade impressionante, não se surpreenda caso leia este documento e, poucos meses depois, encontre informações obsoletas.

(Documentação ainda em construção)

O que são dados?

Pode parecer óbvio, mas a maior arma de um desenvolvedor é conhecer a fundo a base de seu trabalho.

Como um engenheiro de dados, assim como o nome já diz, utilizaremos todas as fontes para coleta de informações, e realizaremos a extração, categorização e estruturação destes, para que tenham valor em seus devidos usos.

Tudo que se pode armazenar são considerado dados:

- Textos
- Números
- Vídeos
- Tweets
- Reels

Quaisquer fontes que tragam algum valor a determinado nicho, podem ser considerados dados e armazenados, este é nosso trabalho, iremos aprender como.

É importante reconhecer que, tudo na rede é dado, mas nem todo dado é valioso ou necessário, precisamos saber separar e categorizar de acordo com a nossa necessidade, pois além do uso destes, como desenvolvedores, é necessário balancear performance e armazenamento. Sabendo disto, realizaremos a separação a partir de:

- Dados Estruturados
- Dados Não estruturados
- Semi estruturados

Cada qual com sua característica, finalidade e tratativa, que veremos nos próximos tópico.

Dados estruturados

Estes são os dados que se encontram em formatação padrão, e o que se espera quando falamos sobre dados.

Pense numa planilha de Excel, temos linhas e colunas que utilizamos como guias para nos auxiliar na organização e visualização destes, como exemplificado abaixo:

NOME	IDADE
Renan	22
Rafael	30
Chico	17

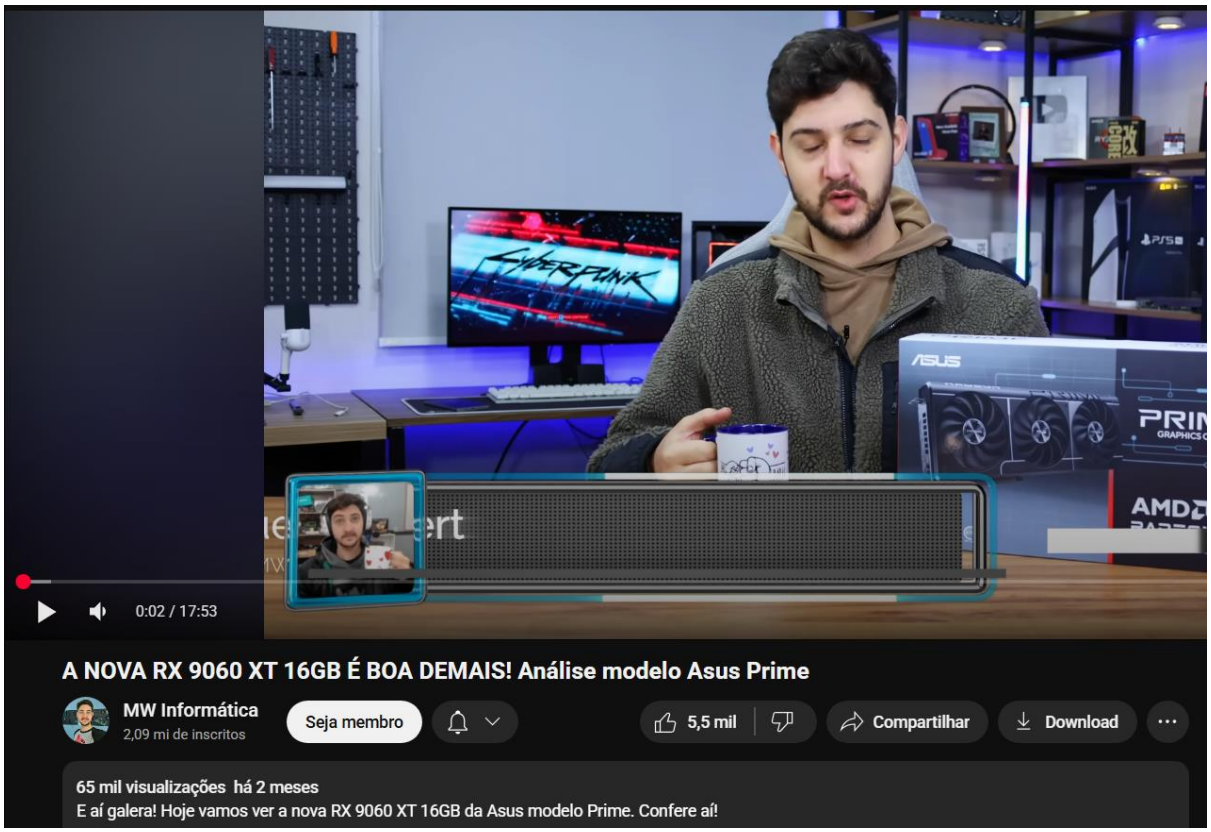
Neste exemplo utilizamos uma estrutura tabular, isto é, linhas e colunas, armazenando nome e idade de possíveis clientes dentro de um sistema. Nesta situação, nos basearemos principalmente nas colunas como marcadores para o que cada operação necessita, seja nome, CPF, idade e afins...

Dados estruturados são ótimos pois estão em um formato de fácil entendimento para quem olhe, facilita referenciação e filtragem de acordo com as necessidades do negócio.

Dados não estruturados

Diferente dos estruturados, onde temos uma clara delimitação de Colunas e linhas para nos basear, os dados não estruturados são aqueles que não possuem necessariamente um “guia”, onde cada informação extraída individualmente tem seu valor, e é necessário um cuidado especial do desenvolvedor para categorizar.

Um ótimo exemplo destes são os vídeos, aqui está um exemplo de como podemos trabalhar com este tipo de dado:



Nesta extração, temos um vídeo relacionado a uma análise de uma placa de vídeo, supondo que estamos trabalhando com uma distribuidora de eletrônicos, e precisamos decidir que tipo de produto iremos disponibilizar, quais destas informações presentes na tela seriam úteis?

Não há necessariamente uma resposta 100% correta para esse tipo de pergunta, mas temos algumas informações valiosíssimas dentre os presentes:

- Número de visualizações
- Número de curtidas
- Quando o vídeo foi postado
- Os comentários

Sim, até os comentários são utilizados aqui, nada é desperdiçado. Para extrairmos este tipo de informações, temos algumas alternativas como web scrapping, API e outras ferramentas, mas isto fica para outro tópico. É importante compreender que mesmo um vídeo possui informações a serem utilizadas no contexto de atribuição de valor.

Dados Semiestruturados

Esta categoria de dados se encontra no limiar entre os últimos dois tópicos apresentados. Podemos dizer que os semiestruturados são dados que possuem um indicador (Como as colunas dos estruturados), mas não são de simples visualização.

Podemos encaixar nessa categoria alguns tipos de arquivos como:

- JSON
- XML
- HTML (Extraindo de páginas)

Utilizando a estrutura de um JSON como exemplo, categorizamos os dados na estrutura de chave – valor dentro de listas, como no exemplo abaixo:

```
{
"Copo": [
    {"Capacidade": "500 ML"},
    {"Cor": "Transparente"},
    {"Material": "Vidro"},
    {"Fabricante": "XXX"}
        ],

"Prato": [
    {"Tamanho": "20 cm"},
    {"Cor": "Branco"},
    {"Material": "Porcelana"},
    {"Fabricante": "YYY"}
        ]
}
```

A estrutura “Chave: Valor” implica que a variável Chave conteria a referência para o valor descrito do objeto, simplificando, considere a chave como as Colunas e valor como as linhas.

O adendo deste tipo de dado, é que diferente dos estruturados, se observar com atenção, nosso objeto "Copo” tem a chave “Capacidade”, enquanto o “Prato” possui

“Tamanho”, com a mesma finalidade, mas com nomes diferentes, é necessário um certo cuidado no acesso e no tratamento destes dados.

Banco de dados

É fundamental que o desenvolvedor tenha um certo domínio quando se trata de armazenamento de dados, por questões de segurança, organização e acesso.

Mas, afinal, o que é banco de dados? Parafraseando o pai dos burros (Google) "é uma coleção organizada de informações estruturadas, armazenadas eletronicamente em um sistema de computador, e é usado para armazenar, gerenciar, recuperar e atualizar dados de maneira eficiente", isto é, o local onde passaremos grande parte do tempo configurando, acessando e armazenando nossas tabelas.

Os SGBD (Sistema de Gerenciamento de Banco de Dados) entre os mais escolhidos e utilizados estão:

- Oracle;
- MySQL;
- PostgreSQL;
- SQL Server.

Estes quem, provavelmente você leitor já deva ter ouvido falar, dentro desses SGBDS podemos adicionar, remover e alterar tabelas e colunas.

Você pode estar se perguntando para que preciso de um banco de dados, se normalmente já armazeno em algum local?

E a resposta é bem mais simples do que parece, vamos imaginar que você tenha 50 planilhas mensais acumulada ao longo dos anos dentro de uma empresa, e seu chefe pede que você registre todas as vendas feitas para uma somatória de valores acumulados.

Um cenário catastrófico, considerando dados armazenados dentro de vários arquivos excel onde você teria que acessar um por um e anotar manualmente. Atividade esta que é totalmente trivializada dentro de um SGBD, onde todos os registros estão armazenados dentro de uma única tabela, sendo possível realizar essa extração com poucas linhas de comando.

Com essa ideia em mente, é importante saber que mesmo dentro das SGBDS existe uma separação entre Bando de Dados Relacional (SQL) e Banco de Dados não relacionais (NoSql), vamos abordar individualmente cada tipo no próximo tópico.

Banco de dados Relacional

Um banco de dados **relacional** (também conhecido como SQL) é como uma agenda telefônica ou uma planilha do Excel muito bem estruturada.

Características Principais:

- **Estrutura Rígida:** Antes de adicionar qualquer contato, você define colunas fixas: `Nome`, `Telefone`, `Endereço` e `Aniversário`. Todos os contatos *precisam* ter informações que se encaixem nessas colunas.

- **Tudo se Conecta:** Você pode ter uma segunda "planilha" apenas para endereços e conectar as duas através de um código de identificação único para cada amigo. Isso garante que não haja repetição de dados e que tudo esteja interligado de forma lógica. Se o endereço de um amigo mudar, você altera em um só lugar.

- **Ordem e Consistência:** Esse método garante que os dados sejam consistentes e confiáveis. É muito difícil inserir uma informação no lugar errado.

Em resumo, o banco de dados relacional preza pela estrutura, pela organização e pelas conexões (relações) claras entre as informações. Pense em sistemas que não podem ter erros, como caixas de banco, sistemas de controle de estoque ou registros de matrículas de alunos.

Banco de dados Não Relacional

Um banco de dados **não relacional** (também conhecido como NoSQL) é como uma gaveta de documentos ou um arquivo.

Características Principais:

- **Estrutura Flexível:** Dentro dessa gaveta, você pode guardar diferentes tipos de "documentos" para cada amigo. Para um amigo, você pode guardar um post-it com o telefone. Para outro, um cartão de visita com e-mail e empresa. Para um terceiro, uma ficha completa com foto, redes sociais e endereço. Não há uma estrutura fixa obrigatória.
- **Independência:** Cada "documento" é independente. A informação de um amigo está toda contida em seu próprio registro. Não há uma necessidade rígida de conectar informações de diferentes "pastas" ou "documentos".
- **Velocidade e Escalabilidade:** Como não precisa se preocupar com uma estrutura rígida e múltiplas conexões, é muito mais rápido e fácil adicionar novas informações ou grandes volumes de dados de formatos variados. É como simplesmente jogar um novo documento na gaveta.

Em resumo, o banco de dados não relacional preza pela flexibilidade, velocidade e pela capacidade de lidar com grandes volumes de dados variados. Pense em plataformas que precisam lidar com uma avalanche de informações diferentes, como redes sociais (posts, fotos, vídeos, perfis de usuários), catálogos de produtos de grandes lojas online ou dados de aplicativos de celular.

Big data

Imagine que, em vez de apenas guardar os contatos dos seus amigos, você precise armazenar todas as interações que acontecem em uma rede social como o Instagram em um único minuto: milhões de fotos postadas, vídeos, stories, curtidas, comentários, mensagens diretas etc.

Isso é Big Data. Não se trata apenas de um grande volume de dados, mas também da sua complexidade e velocidade. Podemos resumi-lo em três "Vs":

- Volume: Quantidade gigantesca de dados.
- Velocidade: Os dados são gerados e precisam ser processados em altíssima velocidade, muitas vezes em tempo real.
- Variedade: Os dados não são organizados e padronizados. Vêm em múltiplos formatos: textos, fotos, vídeos, áudios, dados de sensores, cliques em um site, etc.

Agora, vamos ver como cada tipo de banco de dados lida com esse cenário.

1. Banco de Dados Relacional (A Agenda Telefônica) e o Desafio do Big Data

A nossa "agenda telefônica" super organizada começa a ter problemas sérios quando tenta lidar com o Big Data.

Problema com a Variedade: Como você armazena uma foto, um vídeo e uma localização de GPS nas colunas rígidas de Nome, Telefone e Endereço? A estrutura fixa do banco relacional não foi feita para dados não estruturados ou semiestruturados. Seria como tentar forçar o encaixe de objetos de formatos diferentes em buracos quadrados.

Problema com a Velocidade e o Volume: A necessidade de garantir consistência e verificar as relações entre tabelas a cada nova informação torna o processo mais lento. Em um cenário com milhões de novos dados por segundo, essa rigidez vira um gargalo. Além disso, escalar um banco de dados relacional (deixá-lo mais potente) geralmente significa comprar um servidor maior e mais caro (escalabilidade vertical), o que tem um limite e um custo muito alto.

2. Banco de Dados Não Relacional (A Gaveta de Documentos) e a Solução para o Big Data

A nossa "gaveta de documentos" foi praticamente criada para resolver os desafios do Big Data.

Solução para a Variedade: A flexibilidade é sua maior força. Você pode simplesmente "jogar" qualquer tipo de dado na gaveta — um documento de texto para um post, um arquivo de imagem para uma foto, um conjunto de coordenadas para um mapa. Cada registro é autônomo e não precisa seguir um molde rígido.

Solução para a Velocidade e o Volume: Bancos não relacionais são projetados para escalar horizontalmente. Isso significa que, em vez de comprar um servidor gigante, você pode simplesmente adicionar mais servidores comuns e mais baratos à rede para distribuir a carga. É como ter várias gavetas trabalhando juntas. Essa arquitetura é perfeita para processar um fluxo imenso e contínuo de informações de forma rápida e eficiente.