

# Gerenciamento de Dados para IoT

---

Disciplina: **Introdução à Internet das Coisas - IMD0902**

Prof. Heitor Florencio

[heitorm@imd.ufrn.br](mailto:heitorm@imd.ufrn.br)

# Tópicos

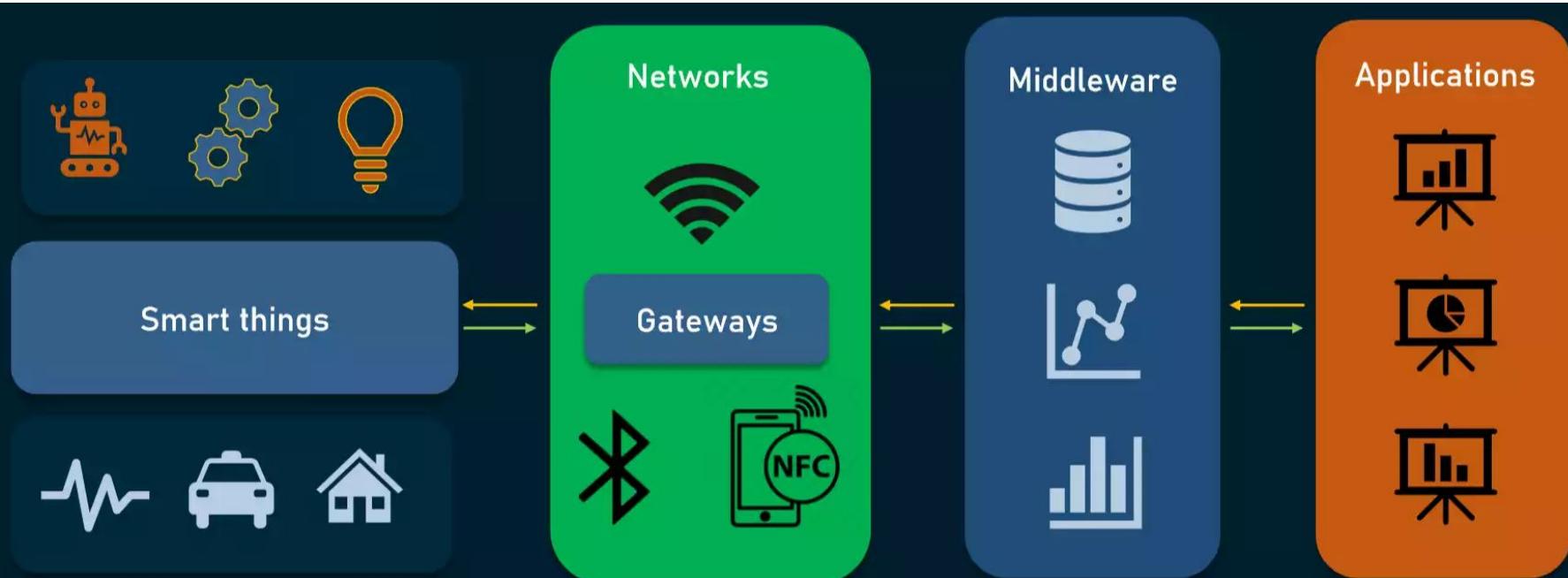
- Importância do Gerenciamento de Dados em IoT
- Características dos Dados das Coisas e Metadados
- Etapas do Gerenciamento de Dados
- Conceitos de Banco de Dados
- Banco de Dados para IoT

Aula:

## Gerenciamento de Dados para IoT

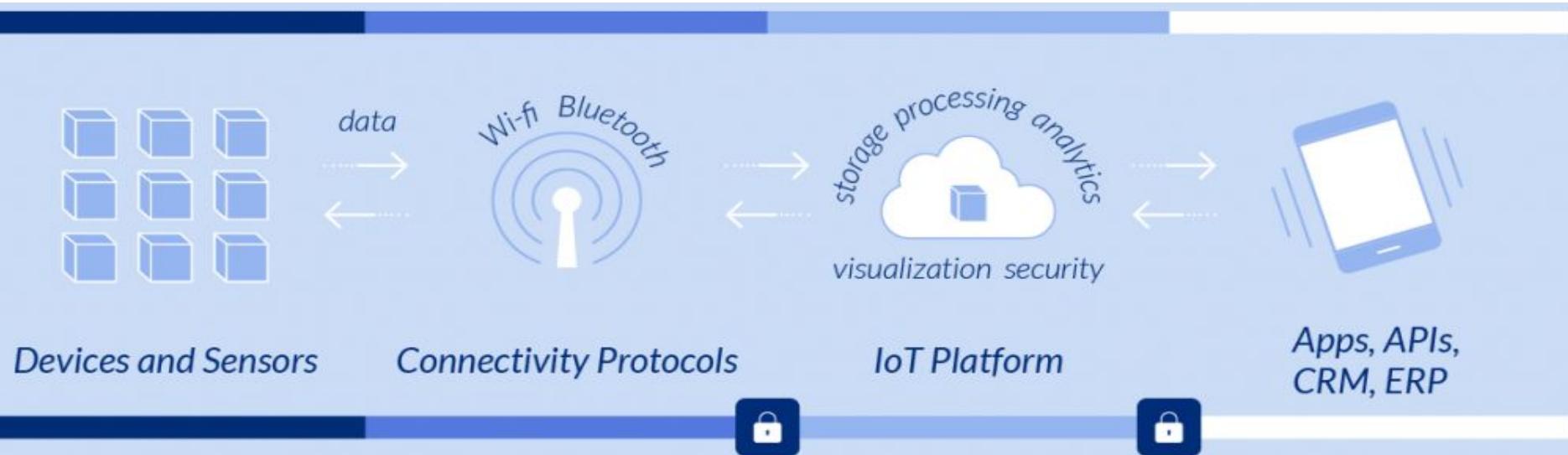
# Arquitetura de Internet das Coisas

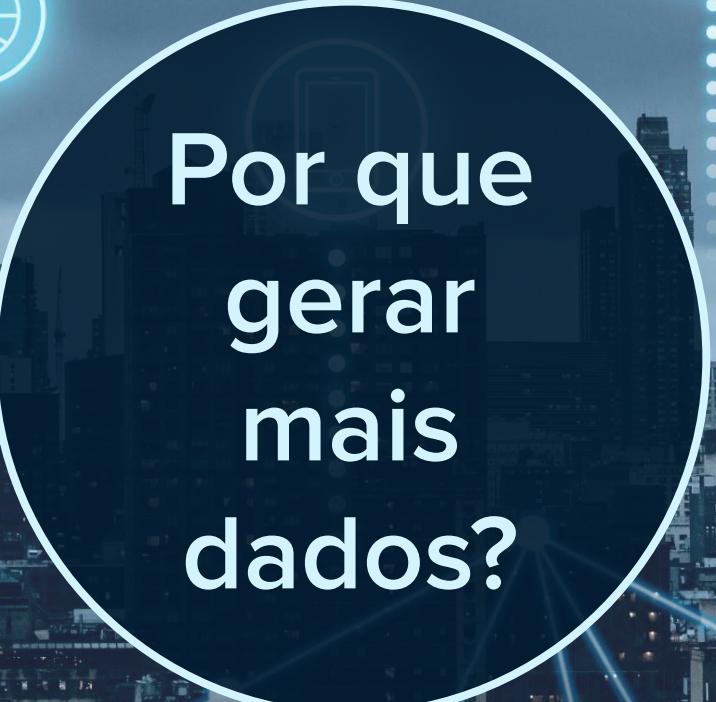
- Uma arquitetura IoT apresenta o fluxo de dados ou informações desde a coleta (sensores e dispositivos) até o armazenamento e consumo dos dados (serviços e aplicações).



# Plataformas de Internet das Coisas

- Uma etapa essencial nas soluções de IoT é o processamento dos dados.





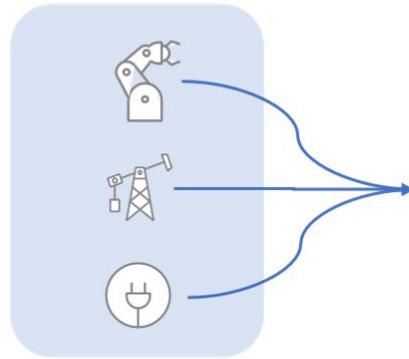
Por que  
gerar  
mais  
dados?





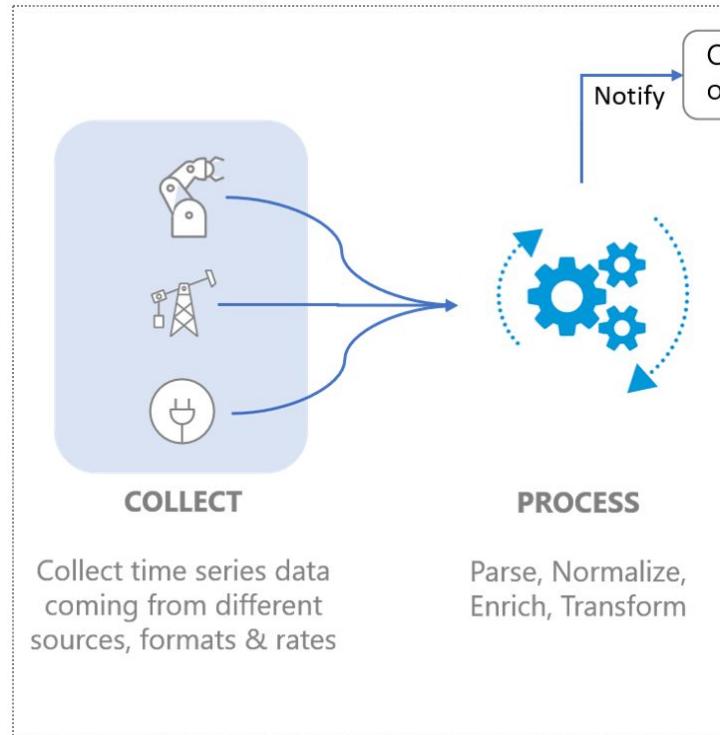
# Fluxo de dados em IoT

- **Coleta:** dispositivos e sensores embutidos nas coisas para coletar os dados.



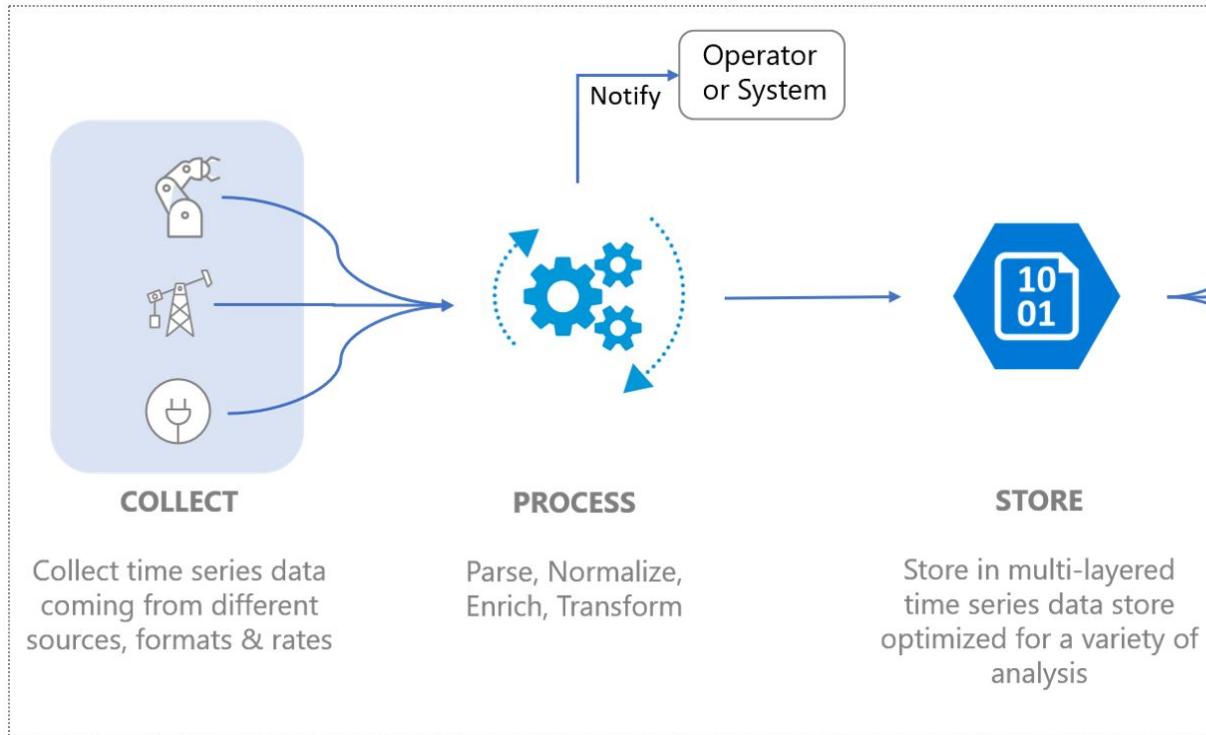
Collect time series data  
coming from different  
sources, formats & rates

# Fluxo de dados em IoT



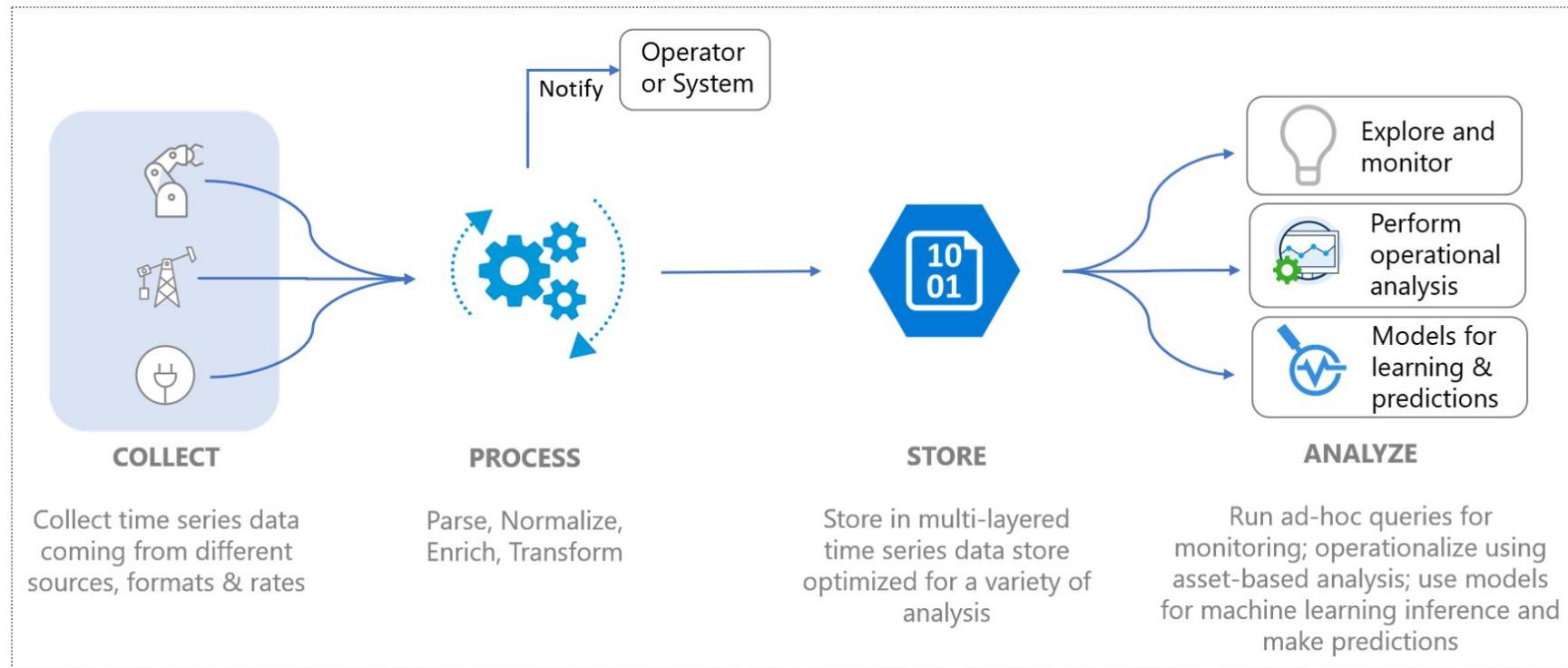
- **Coleta:** dispositivos e sensores embutidos nas coisas para coletar os dados.
- **Processamento (Pré-processamento):** limpeza, normalização, transformação e preparação dos dados.

# Fluxo de dados em IoT



- **Coleta:** dispositivos e sensores embutidos nas coisas para coletar os dados.
- **Pré-processamento:** limpeza, normalização, transformação e preparação dos dados.
- **Armazenamento:** uso adequado de banco para variedade de tipos de dados.

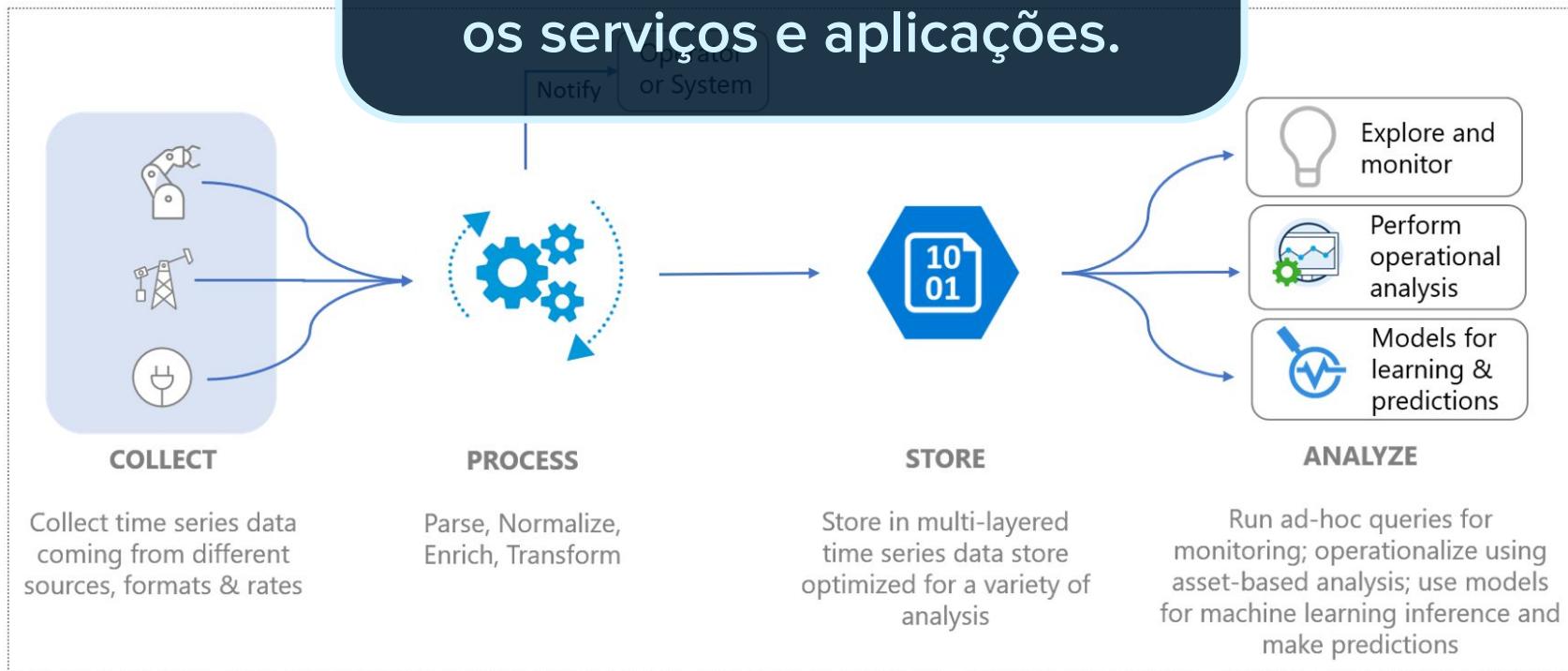
# Fluxo de dados em IoT



Fonte: <https://learn.microsoft.com/en-us/azure/time-series-insights/overview-what-is-tsi>

O objetivo é fornecer  
informações consistentes,  
abrangentes e corretas para  
os serviços e aplicações.

Notify  
Operator  
or System



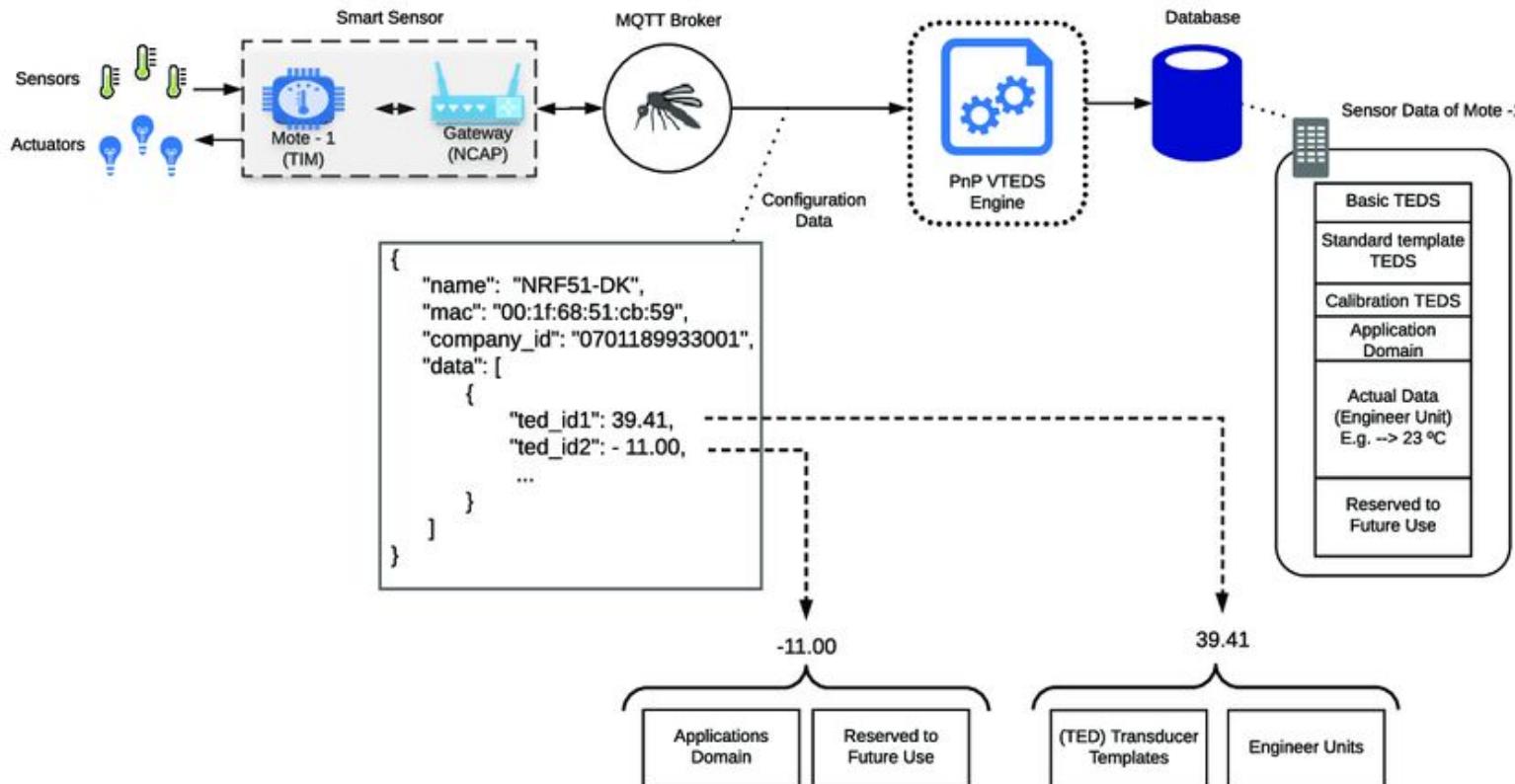
# Características dos Dados das Coisas e Metadados

---

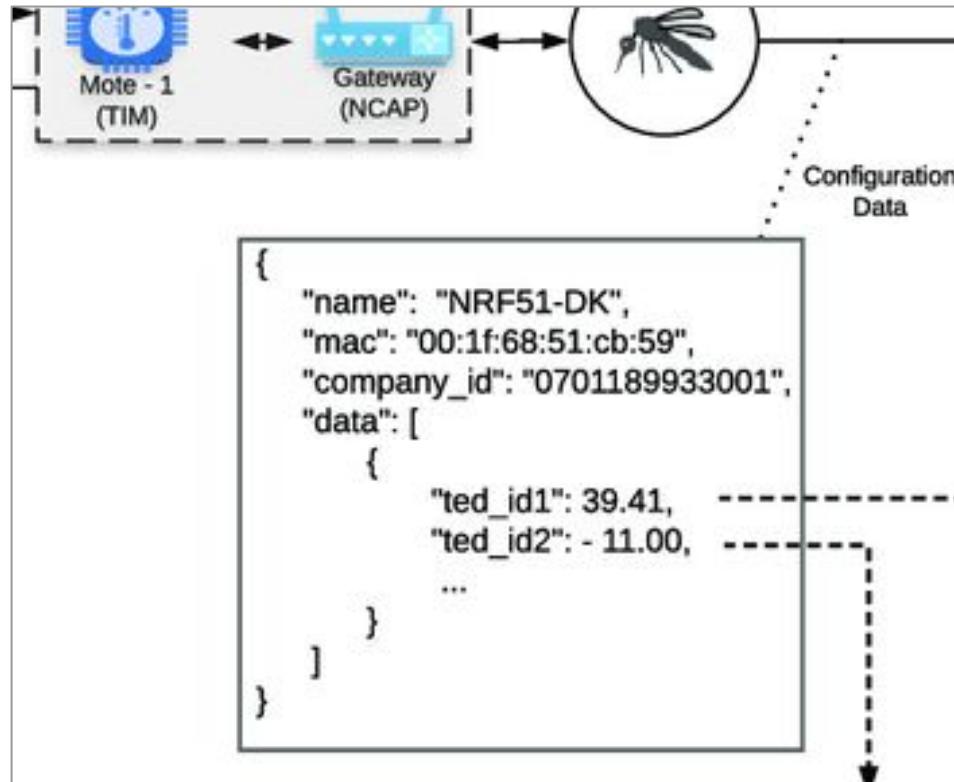
- 1. Geração contínua dos dados a partir dos sensores.**
- 2. Grande quantidade de dados.**
  - Depende da taxa de publicação.
- 3. Diferentes tipos de dados.**
  - Dados numéricos: temperatura, umidade, proximidade etc. Por ex.: dados do sensor DHT11.
  - Textos, imagens, vídeos etc. Por ex.: dados de câmera.
- 4. Ausência de sincronização dos dados.**
  - Diferentes tecnologias de comunicação → diferentes velocidade, topologias, taxas de publicação.

- **Metadados:** um **conjunto de dados** que **descrevem** e fornecem informações sobre outros dados (dados de medição).
  - Dados dos dispositivos:
    - endereço IP do dispositivo.
    - endereço MAC do dispositivo.
    - versão do firmware.
    - localização do dispositivo (ex.: quarto 1 em uma solução de Smart Home).
  - Dados para serviço/aplicação:
    - data da última publicação.
    - taxa de publicação.

# Exemplo de Metadados

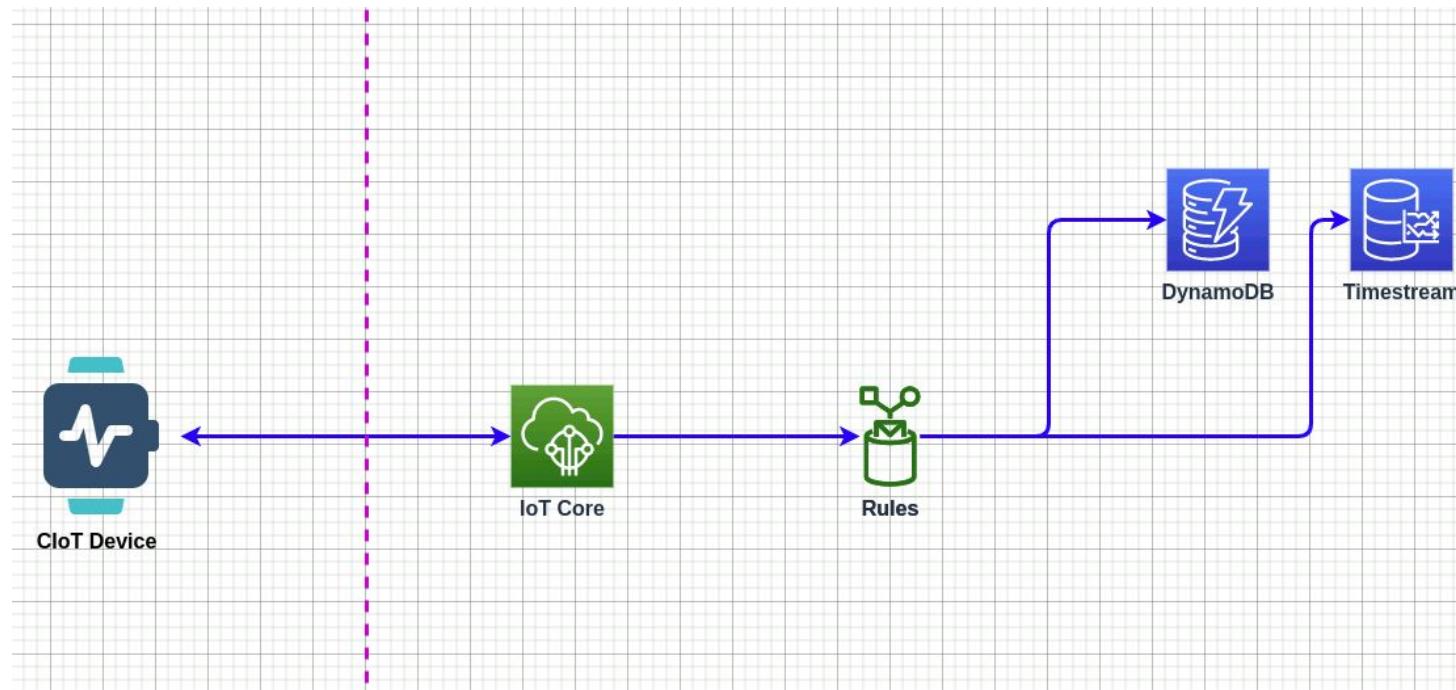


# Exemplo de Metadados



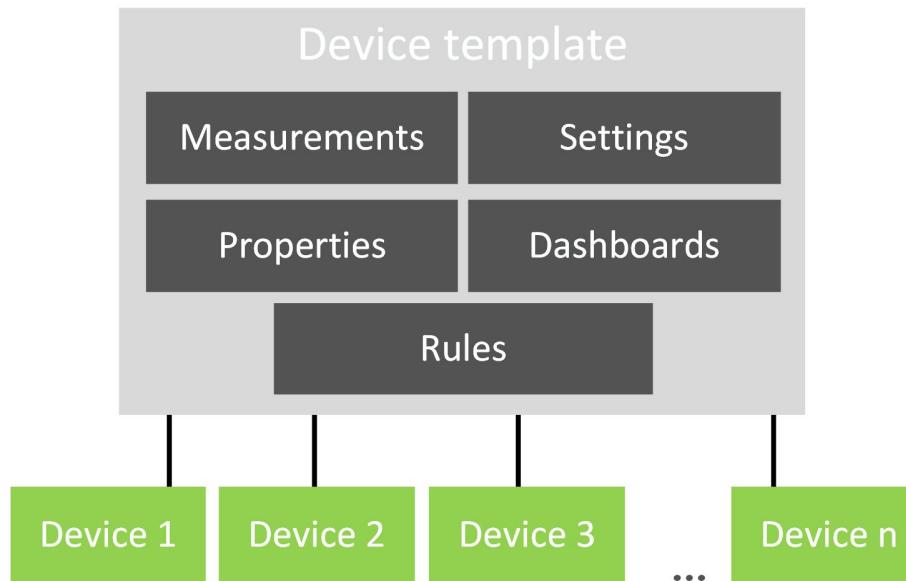
# Metadados na Plataforma IoT da AWS (Cloud)

- Envio de metadados na plataforma IoT da AWS: **IoT Core + Serviços de BD**



# Metadados na Plataforma IoT da Azure (Cloud)

- Template de um dispositivo IoT na plataforma IoT da Azure: **IoT Central**



# Metadados na IoT Central do Azure



Custom app

Search for devices

Version Manage test device Publish Rename Delete

Device templates > Sensor Controller > Model > Sensor Controller

## Sensor Controller

Application updated: yesterday    Interfaces published: yesterday

Sensor Controller Root Published

Add capabilities specific to this device model. [Learn more](#)

Save Add capability Edit identity Export Delete ...

Display name	Name *	Capability type *
Working Set	workingSet	Telemetry
Serial Number	serialNumber	Property
Reboot	reboot	Command

Identity

Default component

Display name ⓘ

Sensor Controller

Namespace ⓘ

Espressif

Name ⓘ

SensorController

Version ⓘ

2

Interface @id

dtmi:Espressif:SensorController;2

# Metadados na IoT Central do Azure

The screenshot shows the Azure IoT Central interface for a "Custom app". The left sidebar includes options like "Connect", "Devices" (which is selected), "Device groups", "Device templates", "Edge manifests", "Analyze", "Data explorer", and "Dashboards". The main area has tabs for "Manage template" and "Manage device". Below these are three rows of device data:

	Date	Type	Date
>	10/28/2022, 1:30:21 PM	Telemetry	10/28/2022, 1:30:21 PM
>	10/28/2022, 1:30:21 PM	Telemetry	10/28/2022, 1:30:21 PM
▽	10/28/2022, 1:30:21 PM	Telemetry	10/28/2022, 1:30:21 PM

When the third row is expanded, a JSON representation of the event data is displayed:

```
1 {  
2   "_eventcreationtime": "2022-10-28T12:30:21.35Z",  
3   "SensorMagnetY": {  
4     "SensorMagnetY": 39.6195458116379  
5   },  
6   "_eventtype": "Telemetry",  
7   "_timestamp": "2022-10-28T12:30:21.369Z"  
8 }
```

# Metadados na IoT Central do Azure

The screenshot shows the Azure IoT Central interface. On the left, a sidebar titled "Custom app" has "Devices" selected. The main area shows a device named "2brchwn7" connected. Below the device name are tabs for "Raw data" and "Mapped aliases", with "Raw data" being active. A timestamp at the bottom indicates the data is from 10/28/2022, 3:41:52 P. In the center, a "Data preview" section displays the following JSON data:

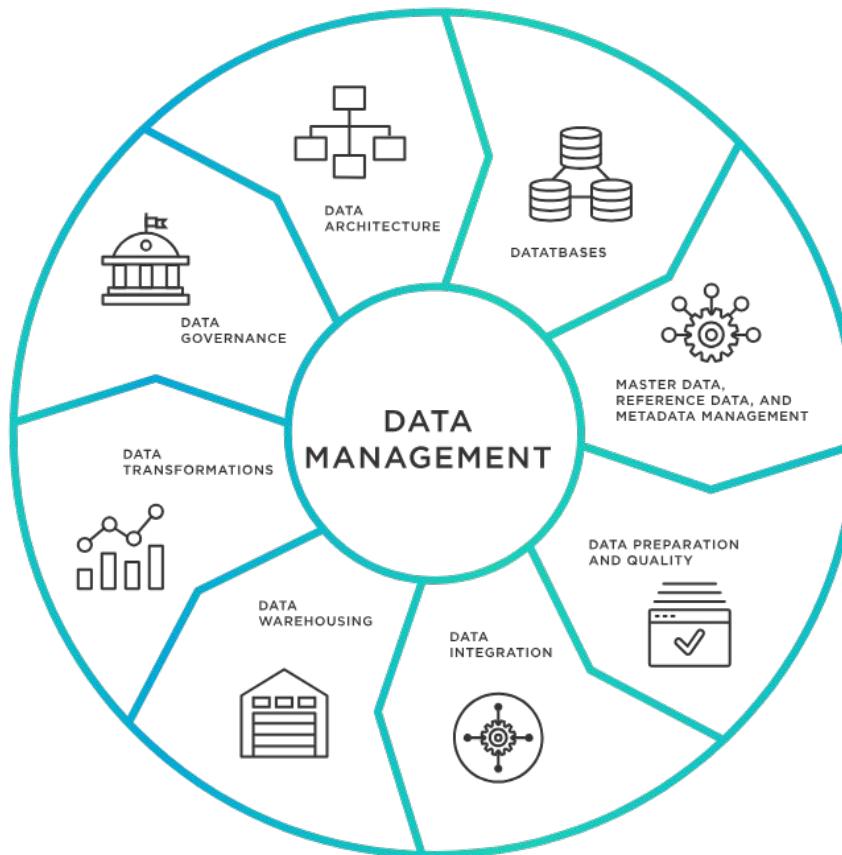
```
1  {
2    "telemetries": {
3      "temperature": 10
4    },
5    "properties": {
6      "maxTempSinceLastReboot": 10.96
7    }
8 }
```

- Os **metadados** são imutáveis ou raramente muda.
  - O ID de um dispositivo não deveria mudar.
  - A classe ou tipo também raramente muda.
- Quando **houver alteração** em algum metadado, é indicado **guardar o histórico** de alterações para recriar a situação na qual o dado original foi gerado/coletado.

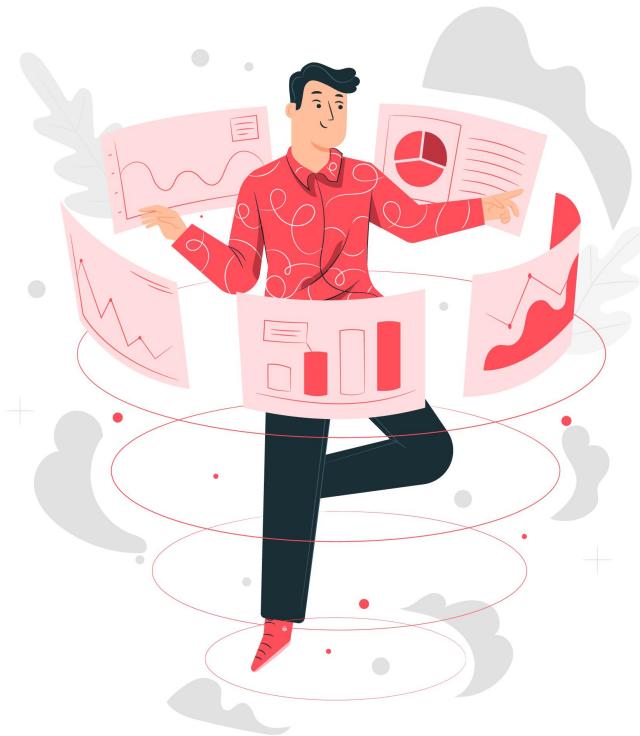
# Etapas no Gerenciamento de Dados

---

# Gerenciamento de Dados



- 1. Coleta dos dados**
- 2. Modelagem dos dados**
- 3. Pré-processamento dos dados**
- 4. Armazenamento dos dados**
- 5. Extração do conhecimento (insights)**

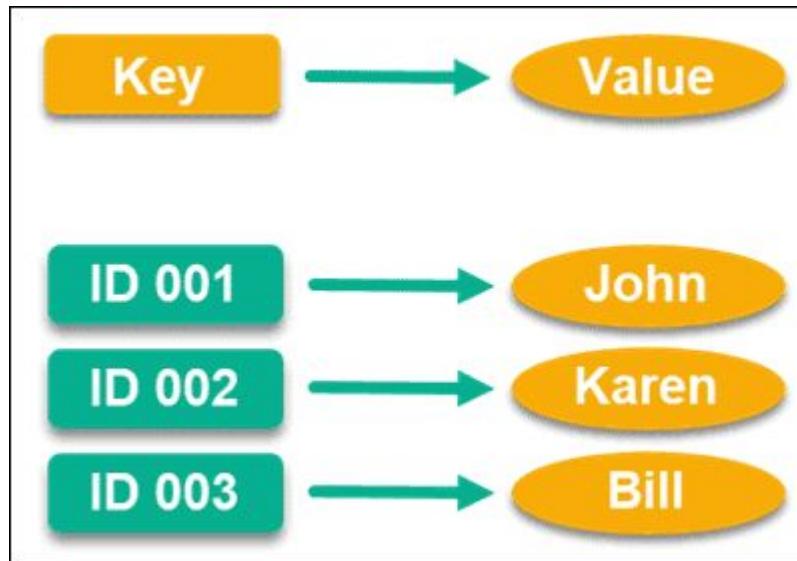


- Definir uma **representação para manipular** e trabalhar com esses dados visando uma interoperabilidade e formatos padrões interpretáveis.
- A modelagem dos dados varia de acordo com a aplicação.
- A modelagem deve considerar os tipos de dados:
  - Dados estruturados: representados em formato tabular.
  - Dados não estruturados: documentos, imagens e vídeos.
  - Dados semiestruturados: não tem um esquema único. combina os diferentes tipos de dados.



## Representação dos dados:

1. **Chave-valor (key-value):** dados são modelados como um conjunto de chaves e valores.



## Representação dos dados:

2. **Esquema de linguagem de marcação (markup scheme):** utiliza tags e estruturação hierárquica dos dados.

### XML (Extensible Markup Language)

```
<!-- Schema Components -->
<xss:complexType name="baseComponent">
    <xss:complexContent> [29 lines]
</xss:complexType>
<xss:complexType name="componentWithFacets">
    <xss:complexContent>
        <xss:extension base="baseComponent"> [3 lines]
    </xss:complexContent>
</xss:complexType>
<xss:element name="schema">
    <xss:complexType>
        <xss:complexContent>
            <xss:extension base="baseComponent">
                <xss:attribute name="type" use="required">
```

### JSON (JavaScript Object Notation)

```
{
    "book1": {
        "name": "high school mathematics",
        "price": 12
    },
    "book2": {
        "name": "advanced high school mathematics",
        "price": 14
    }
}
```

# Envio de JSON para broker MQTT

- Existem algumas bibliotecas para **formatar os dados em JSON**.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <ArduinoJson.h>

//Cria o objeto dinamico "json" com tamanho "2" para a biblioteca
DynamicJsonDocument json(JSON_OBJECT_SIZE(2));

//Atrela ao objeto "json" as leitura do sensor com os Aliases definidos
json[ALIAS1] = temperatura;
json[ALIAS2] = umidade;

//Mede o tamanho da mensagem "json" e atrela o valor somado em uma unidade ao objeto "tamanho_mensagem"
size_t tamanho_mensagem = measureJson(json) + 1;

//Cria a string "mensagem" de acordo com o tamanho do objeto "tamanho_mensagem"
char mensagem[tamanho_mensagem];

//Copia o objeto "json" para a variavel "mensagem" e com o "tamanho_mensagem"
serializeJson(json, mensagem, tamanho_mensagem);

//Publica a variavel "mensagem" no servidor utilizando a variavel "TOPICO"
Serial.println("");
Serial.print("Mensagem enviada: ");
Serial.println(mensagem);
MQTT.publish(TOPICO, mensagem);
```

- Existem algumas bibliotecas para **formatar os dados em JSON**.

```
StaticJsonBuffer<300> JSONbuffer;
JsonObject& JSONencoder = JSONbuffer.createObject();

JSONencoder["device"] = "ESP32";
JSONencoder["sensorType"] = "Temperature";
JSONArray& values = JSONencoder.createNestedArray("values");

values.add(20);
values.add(21);
values.add(23);

char JSONmessageBuffer[100];
JSONencoder.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer));
```

## Representação dos dados:

3. Baseada em objetos.
4. Baseada em lógica.
5. Baseada em ontologia (RDF, OWL).

**Situação:** Um projeto de Smart City requer monitorar dados de toda iluminação, tráfego de veículos e dados de acidentes de trânsito de uma cidade. Existem 3 tipos de coleta de dados:

- PosteConectado: um dispositivo instalado em um poste de iluminação que envia a cada 1s dados de estado do poste (ligado, desligado, em manutenção).
- RuaMonitorada: um dispositivo com uma câmera embutida, instalado em placas de avenidas, que envia a cada 1min informações dos carros que passaram (placa, modelo, motorista, velocidade), bloco de frames (imagens) do veículo e eventos (imagens de possíveis situações de colisão).
- OuvidoriaColisão: um dispositivo que envia um relato (voz, texto) dos motoristas quando ocorre uma colisão. O agente de trânsito usa o dispositivo sempre que chegar no local do acidente e, em tempo real, dados dos motoristas/veículos e os relatos (voz, texto) são enviados.

**Como modelar o envio e armazenamento desses dados para uma plataforma IoT?**

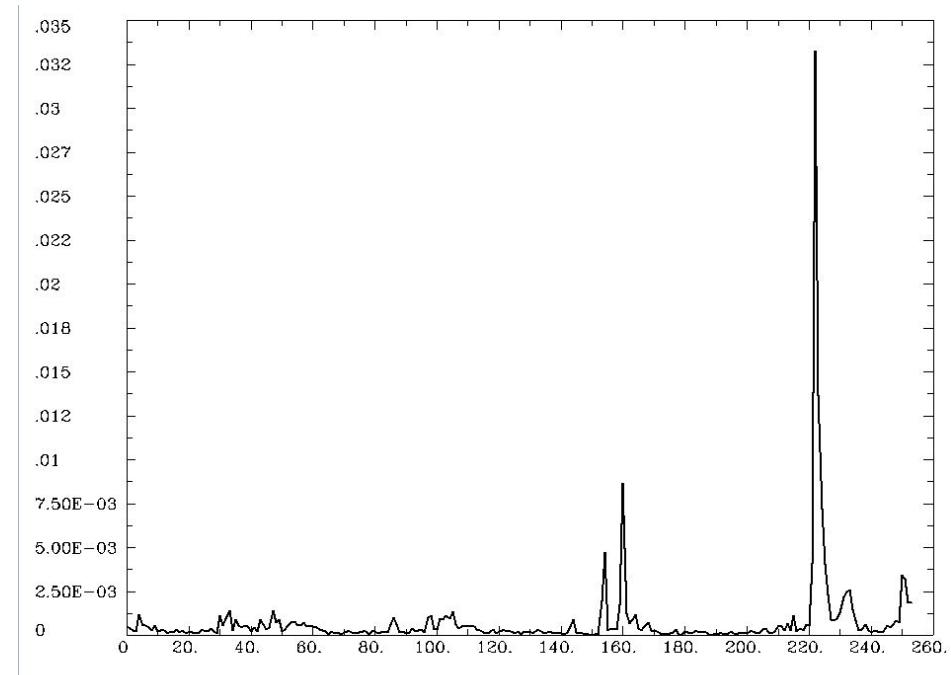
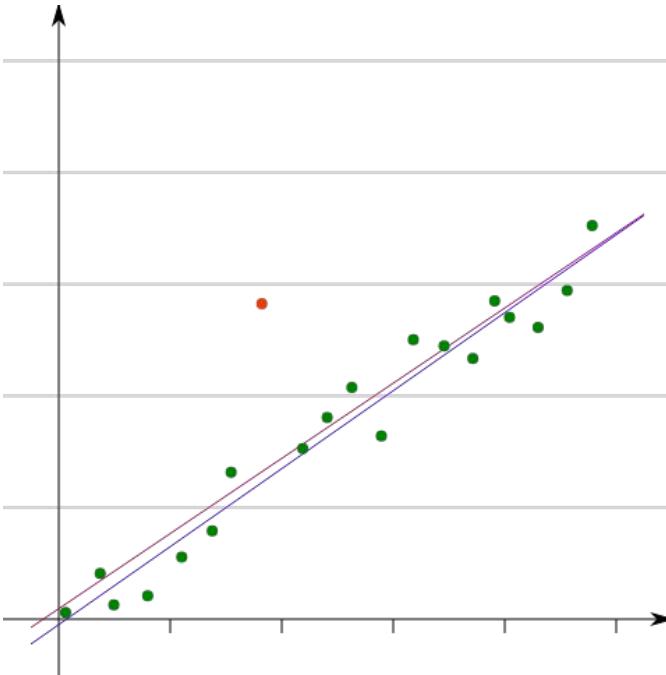
Processamento para melhoria da qualidade dos dados.

## Problemas:

- ★ Informações-chave estão **faltando** ou os dados estão **incompletos**?
- ★ Os dados atendem às regras básicas de **verificação de dados**?
  - Por exemplo, os valores de umidade devem ser de 0% a 100%.
- ★ Com que frequência **os mesmos valores/dados** aparecem no sistema?
  - Por exemplo, entradas de dados duplicadas do mesmo dispositivo.
- ★ A qualidade dos dados é **consistente** em todo o sistema?
  - Por exemplo, a data de nascimento está no formato dd/mm/aaaa em um conjunto de dados, mas no formato mm/dd/aaaa em outro conjunto de dados.

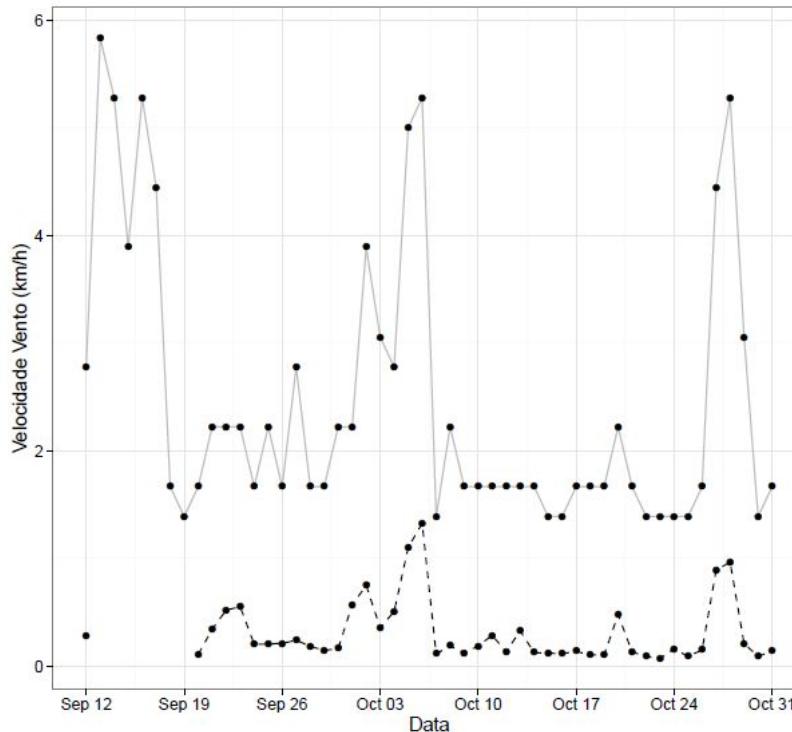
Existem diversas técnicas de pré-processamento de dados para resolver os problemas:

## 1. Imprecisões e outliers:



Existem diversas técnicas de pré-processamento de dados para resolver os problemas:

## 2. Lacunas nos dados:



Existem diversas técnicas de pré-processamento de dados para resolver os problemas:

### 3. Diferença de granularidade:

**Situação:** Um projeto de Smart City requer monitorar dados de toda iluminação, tráfego de veículos e dados de acidentes de trânsito de uma cidade. Existem 3 tipos de coleta de dados:

- PosteConectado: a cada 1seg
- RuaMonitorada: a cada 1min
- OvidoriaColisão: envio por evento

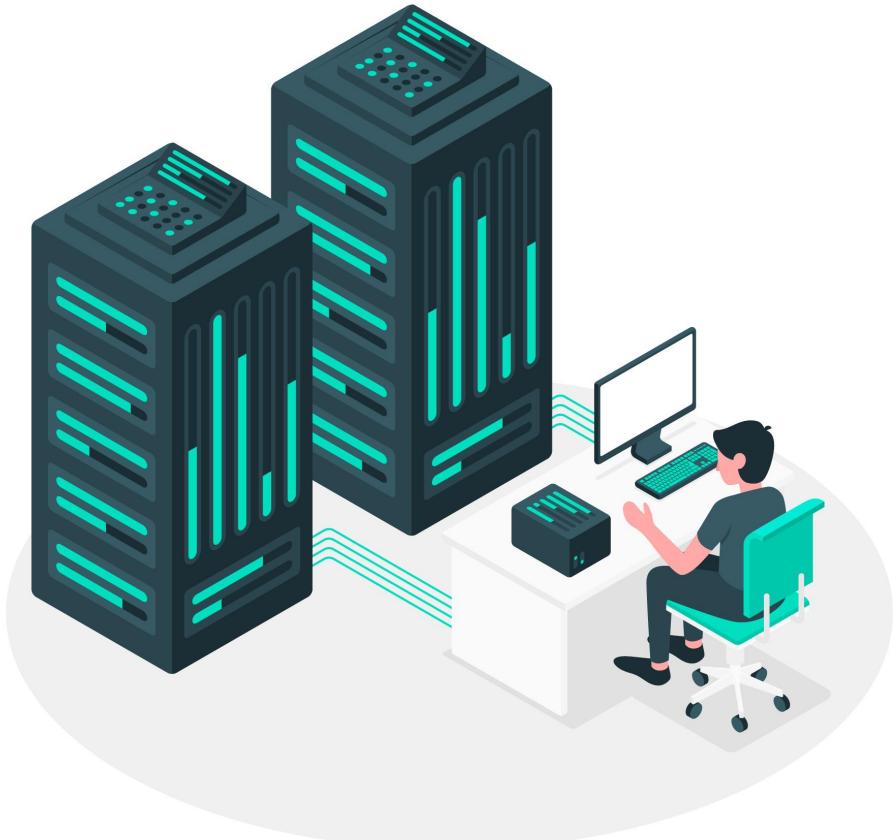
O tratamento dos dados  
pode ser realizado no  
dispositivo IoT, gateway ou  
na plataforma IoT



**Quais tratamentos de dados  
(pré-processamento) podem  
ser realizados no seu projeto?**

# Armazenamento dos dados: Banco de Dados

---



O que são  
Banco de  
Dados?

- Um conjunto estruturado de dados mantidos em um computador ou um conjunto de computadores.
- Um banco de dados é uma coleção organizada de informações ou dados. Os bancos de dados são projetados para armazenar, recuperar, modificar e deletar dados em sistemas de computador.
- **SGBD (Sistema de Gerenciamento de Banco de Dados):** é o software que gerencia e interage com o banco de dados.
  - Exemplos: MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, SQLite;
  - Exemplos: MongoDB, Redis, Amazon DynamoDB, Apache Cassandra, Neo4j;
  - Exemplos: InfluxDB, TimescaleDB;

# Por que precisamos de banco de dados?

## Tamanho

- Você pode precisar armazenar centenas, milhares, trilhares de dados em um só lugar

## Corretude

- Você pode precisar obrigar certos dados a terem determinada forma

## Segurança

- Restrição de acesso a alguns ou todos os dados

## Redundância

- Evitar redundância de dados repetidos

## Importância

- Evitar perda de dados durante problemas operacionais

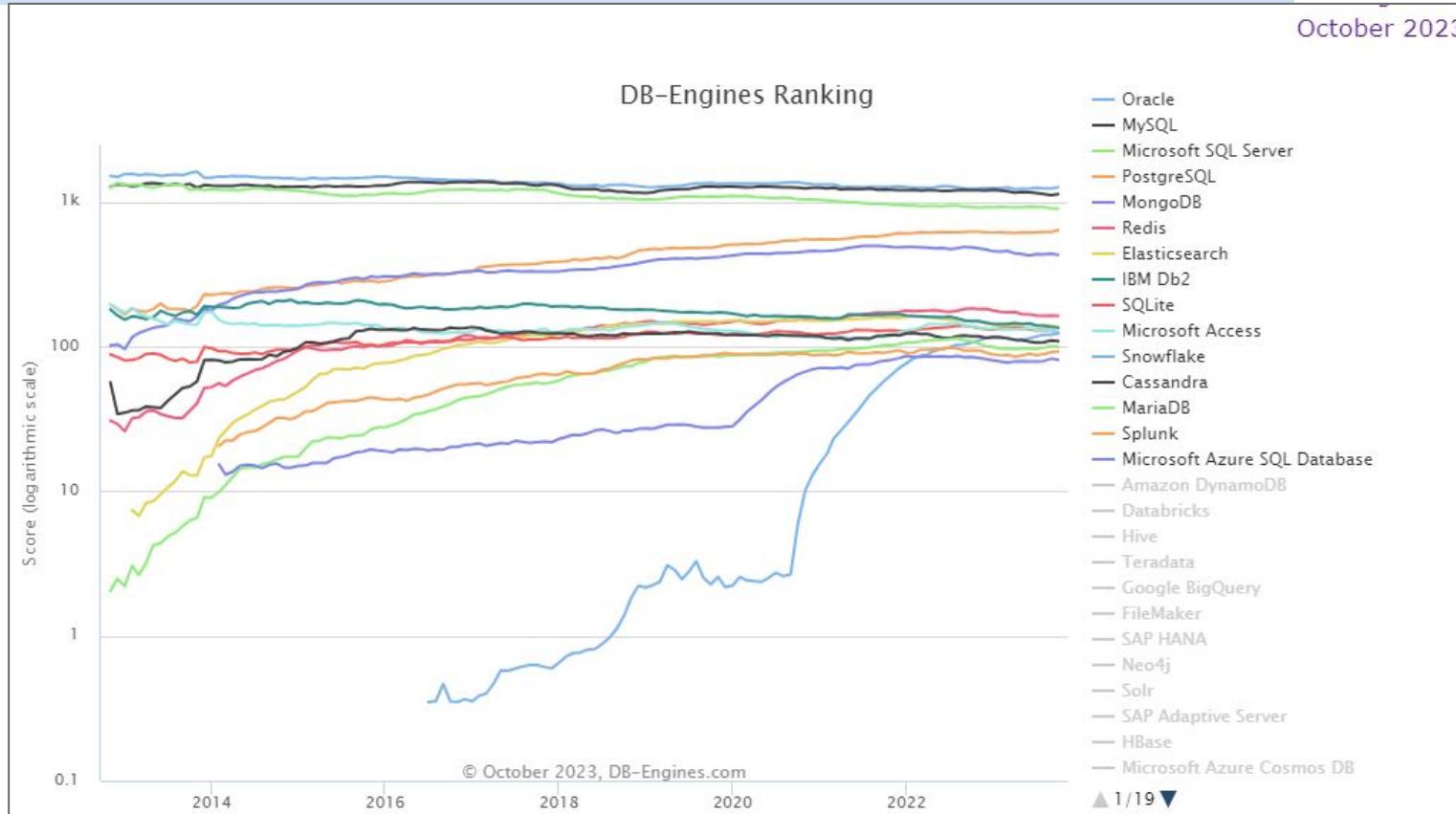
## Acesso concorrente

- Possibilitar múltiplos acessos de leitura e escrita de dados ao mesmo tempo

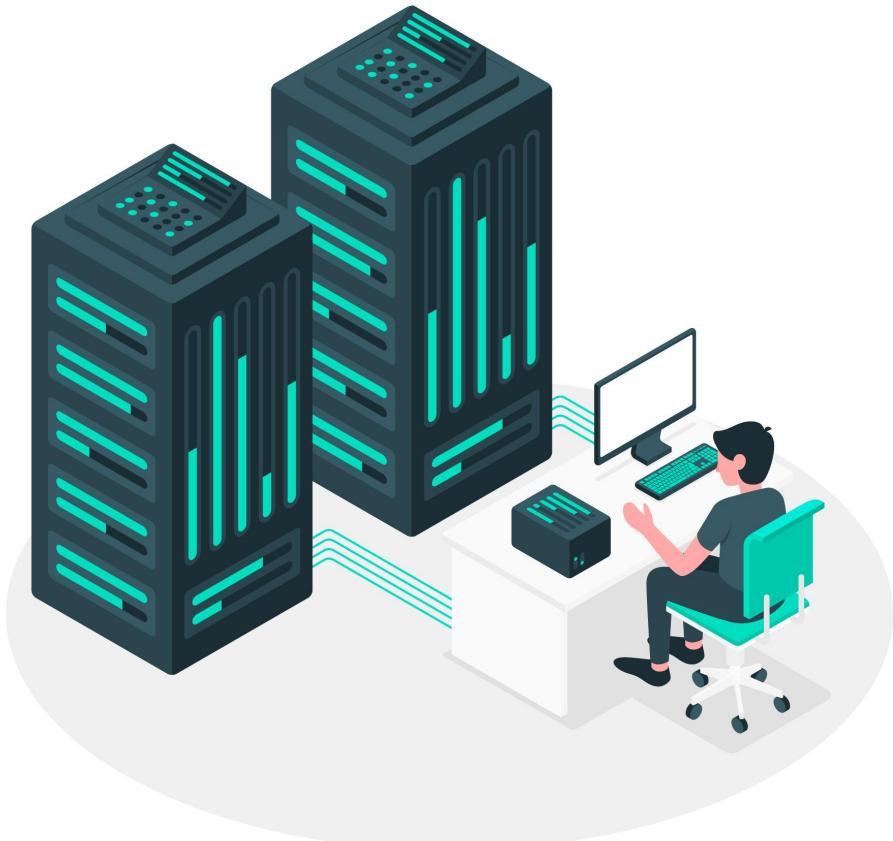
# Ranking de Bancos

415 systems in ranking, October 2023										
Rank			DBMS	Database Model	Score			Oct 2023	Sep 2023	Oct 2022
Oct 2023	Sep 2023	Oct 2022			Oct 2023	Sep 2023	Oct 2022			
1.	1.	1.	Oracle	Relational, Multi-model	1261.42	+20.54	+25.05			
2.	2.	2.	MySQL	Relational, Multi-model	1133.32	+21.83	-72.06			
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	896.88	-5.34	-27.80			
4.	4.	4.	PostgreSQL	Relational, Multi-model	638.82	+18.06	+16.10			
5.	5.	5.	MongoDB	Document, Multi-model	431.42	-8.00	-54.81			
6.	6.	6.	Redis	Key-value, Multi-model	162.96	-0.72	-20.41			
7.	7.	7.	Elasticsearch	Search engine, Multi-model	137.15	-1.84	-13.92			
8.	8.	8.	IBM Db2	Relational, Multi-model	134.87	-1.85	-14.79			
9.	9.	↑ 10.	SQLite	Relational	125.14	-4.06	-12.66			
10.	10.	↓ 9.	Microsoft Access	Relational	124.31	-4.25	-13.85			
11.	11.	↑ 13.	Snowflake	Relational	123.24	+2.35	+16.51			
12.	12.	↓ 11.	Cassandra	Wide column, Multi-model	108.82	-1.24	-9.12			
13.	13.	↓ 12.	MariaDB	Relational, Multi-model	99.66	-0.79	-9.65			
14.	14.	14.	Splunk	Search engine	92.37	+0.98	-2.28			
15.	15.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-model	80.93	-1.80	-4.03			
16.	16.	↓ 15.	Amazon DynamoDB	Multi-model	80.91	+0.00	-7.44			
17.	17.	↑ 20.	Databricks	Multi-model	75.82	+0.64	+18.21			

# Popularidade dos Bancos



Fonte: [https://db-engines.com/en/ranking\\_trend](https://db-engines.com/en/ranking_trend)

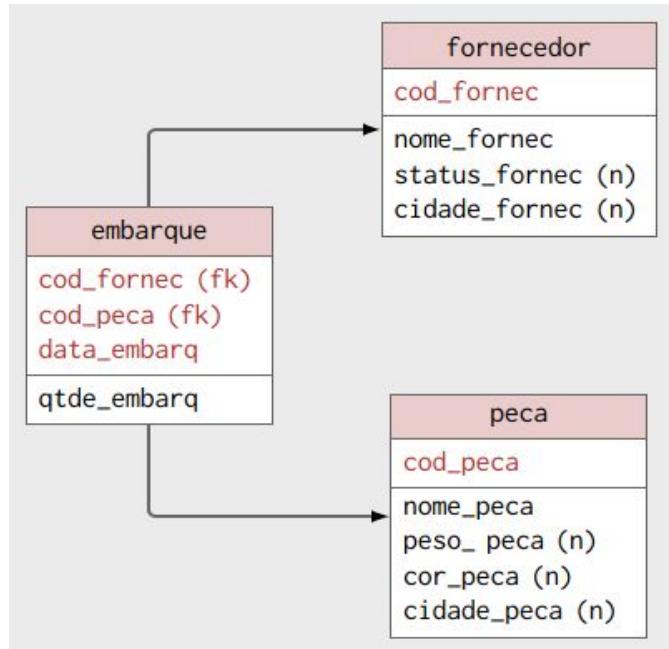


Como  
escolher  
o BD?

# Exemplo de estrutura de Banco de Dados

A escolha depende da aplicação e da estrutura dos dados.

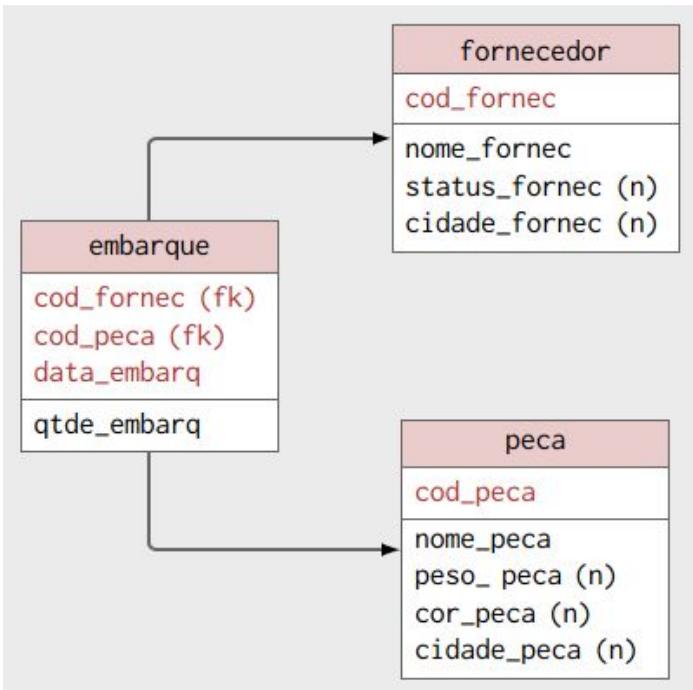
- **Exemplo:** banco de dados de um sistema de embarques.



embarque			
cod_forneç	cod_peça	data_embarq	qtde_embarq
F1	P1	2000-01-12	300
F1	P1	2000-01-15	200
F1	P2	2000-01-12	350
F1	P3	2000-07-22	250
F1	P4	2000-01-12	150
F1	P5	2000-05-14	200
F2	P1	2000-01-12	300
F2	P1	2000-12-04	300
F2	P2	2000-12-04	350
F2	P3	2000-12-04	250
F2	P4	2000-09-24	150
F3	P2	2000-04-04	200
F3	P3	2000-10-30	350

# Exemplo de estrutura de Banco de Dados

- **Exemplo:** banco de dados de um sistema de embarques.



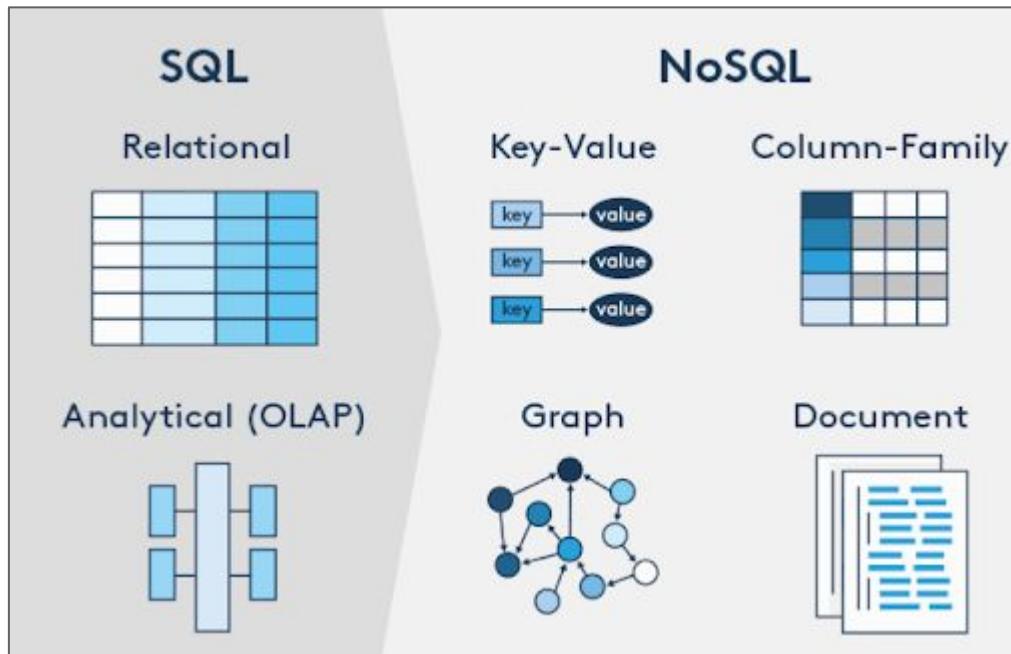
fornecedor

cod_fornecc	nome_fornecc	status_fornecc	cidade_fornecc
F1	Antunes	5	Porto Alegre
F2	Silva	10	Porto Alegre
F3	Souza	15	Curitiba
F6	Antunes	10	Rio
F4	Machado	10	<N>
F5	Barcelos	12	Rio

peça

cod_peca	nome_peca	peso_peca	cor_peca	cidade_peca
P1	Parafuso	5	Cinza	Porto Alegre
P2	Arruela	5	Cinza	Porto Alegre
P3	Mancal	25	Vermelho	Rio
P4	Eixo	15	Verde	Rio
P5	Motor	65	Vermelho	<N>

# Tipos de Banco de Dados



# Tipos de Banco de Dados



- MySQL
- Oracle
- Microsoft SQL Server
- PostgreSQL

- InfluxDB
- Timescale DB
- Graphite
- Prometheus

- MongoDB
- Cassandra
- Redis
- Couchbase

Document-based

Column-based

Key-value

Graph-database



ORACLE



APACHE  
**HBASE**



amazon  
DynamoDB

Cassandra

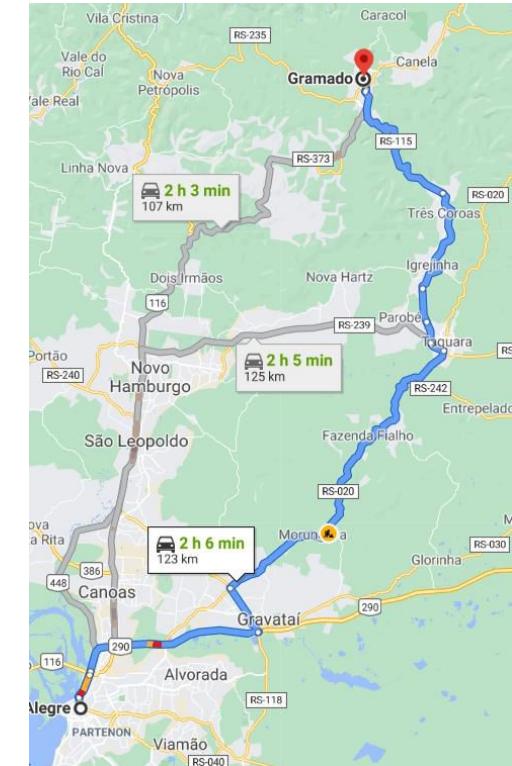
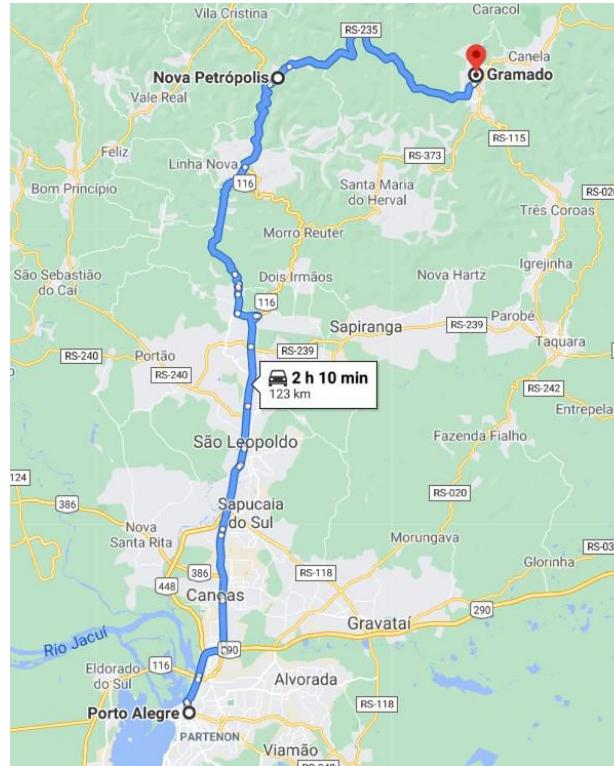
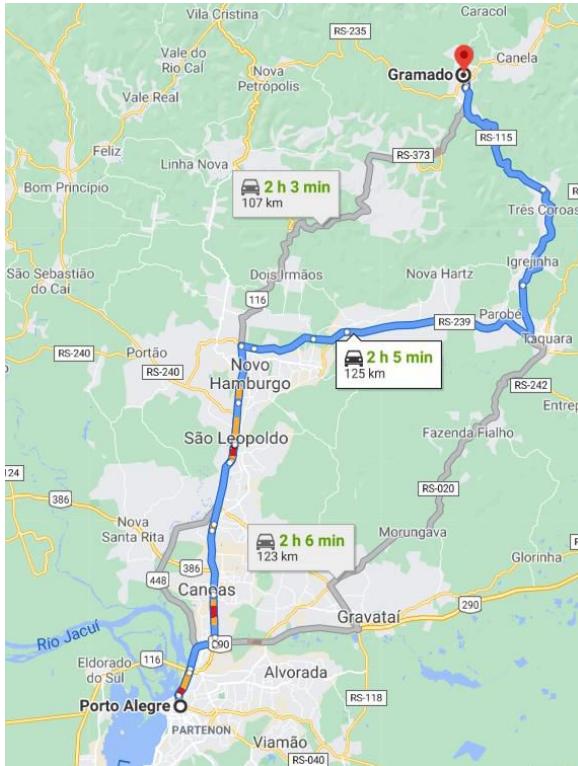
riak

mongoDB

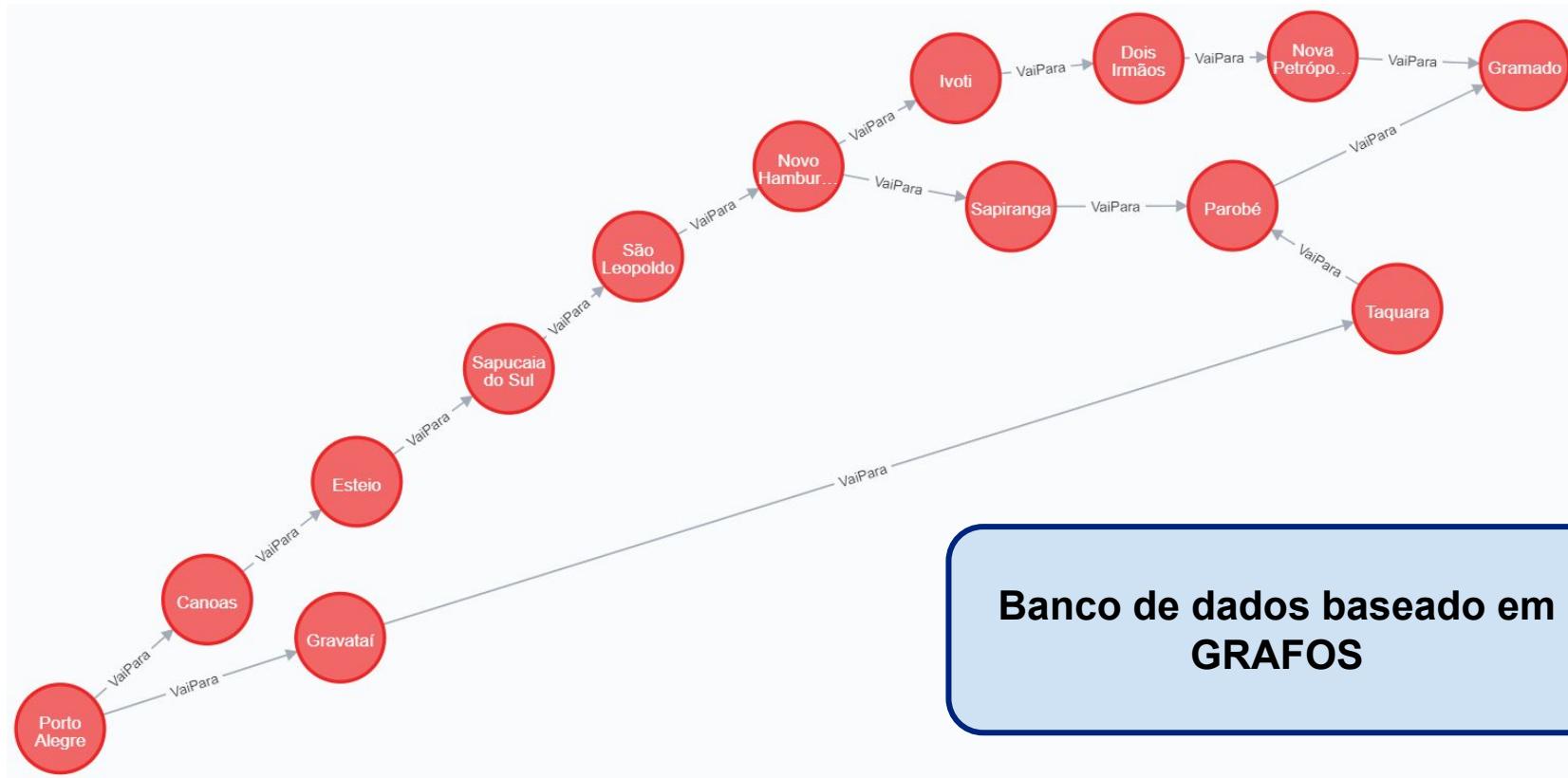


redis

# Exemplo de BD para monitoramento de rotas

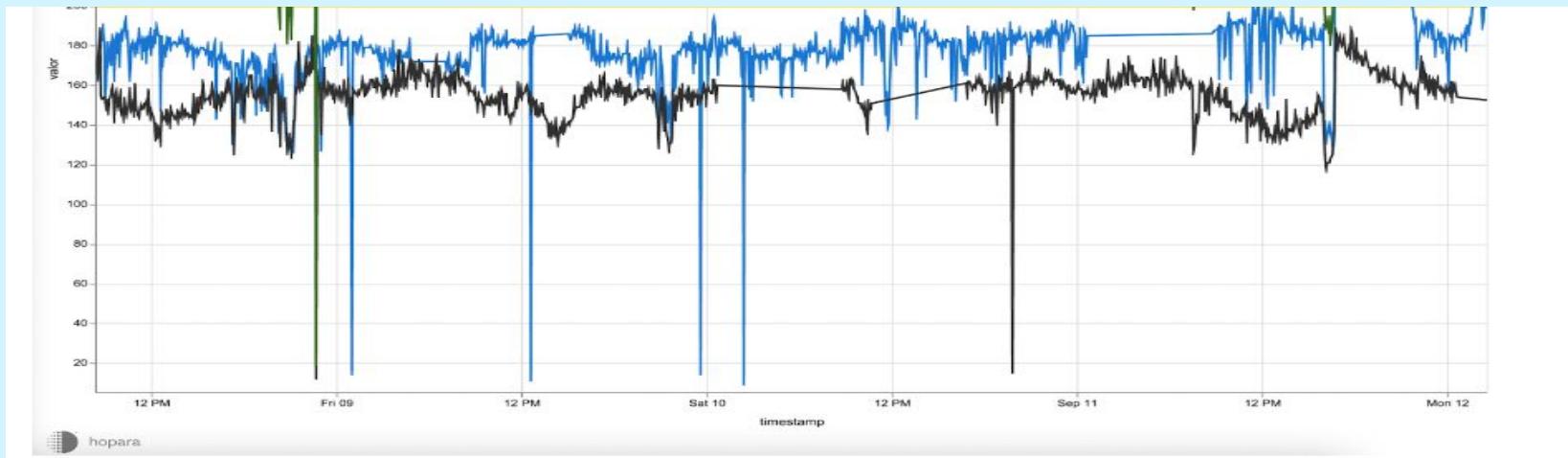


# Exemplo de BD para monitoramento de rotas

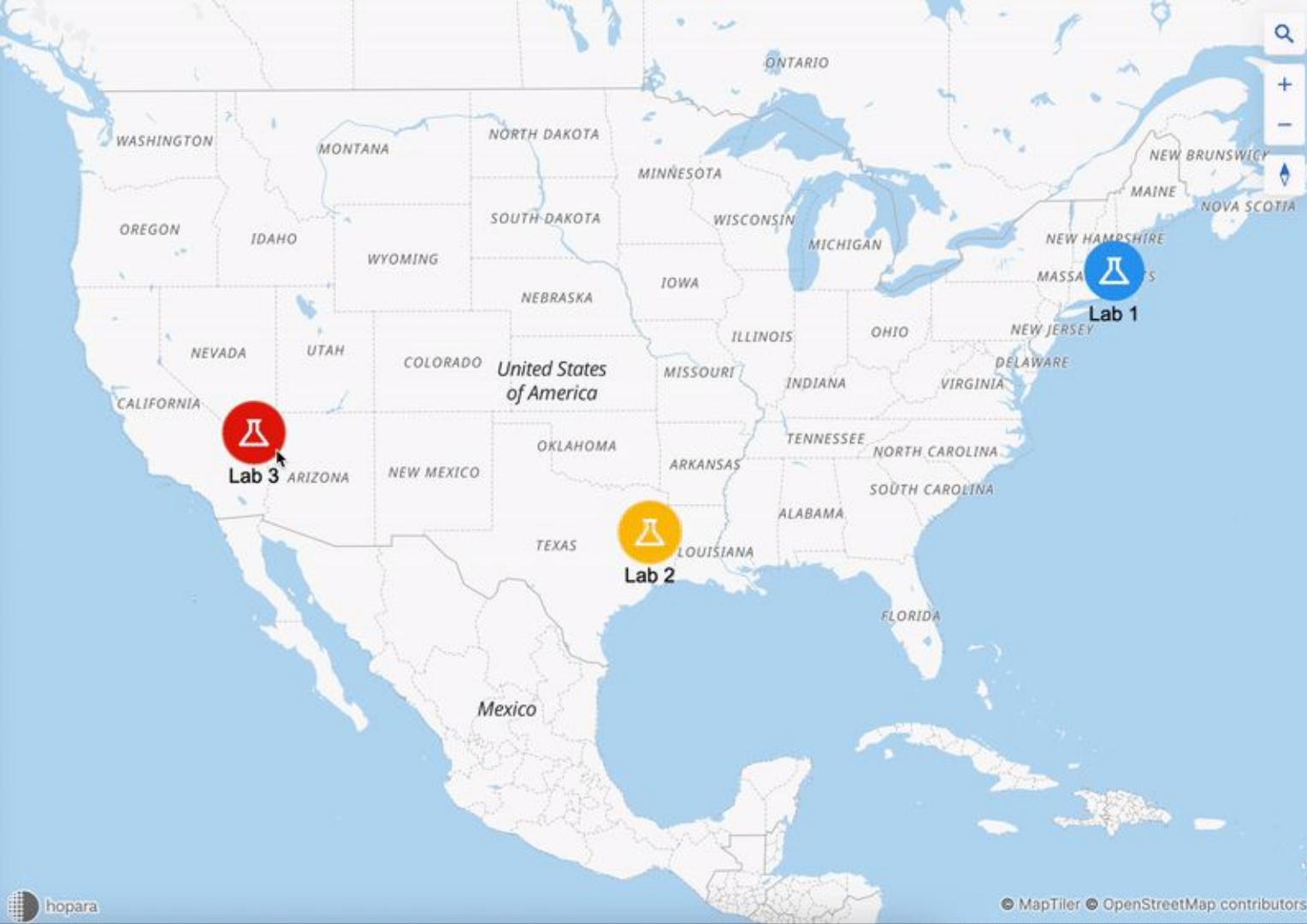


# Exemplo de escolha de Banco de Dados

**Situação:** Uma empresa monitora 50 mil sensores instalados em 58 fábricas em um país. Cada sensor envia dado de vibração de uma máquina em intervalos de 100 ms. O dispositivo IoT envia dados para o banco 24h por dia na taxa de 100ms. A variável altera de acordo com o tempo e uma vibração excessiva contínua pode ser um indício de falha na máquina.



Fonte: <https://www.timescale.com/blog/visualizing-iot-data-at-scale-with-hopara-and-timescaledb/>



# Exemplo de escolha de Banco de Dados

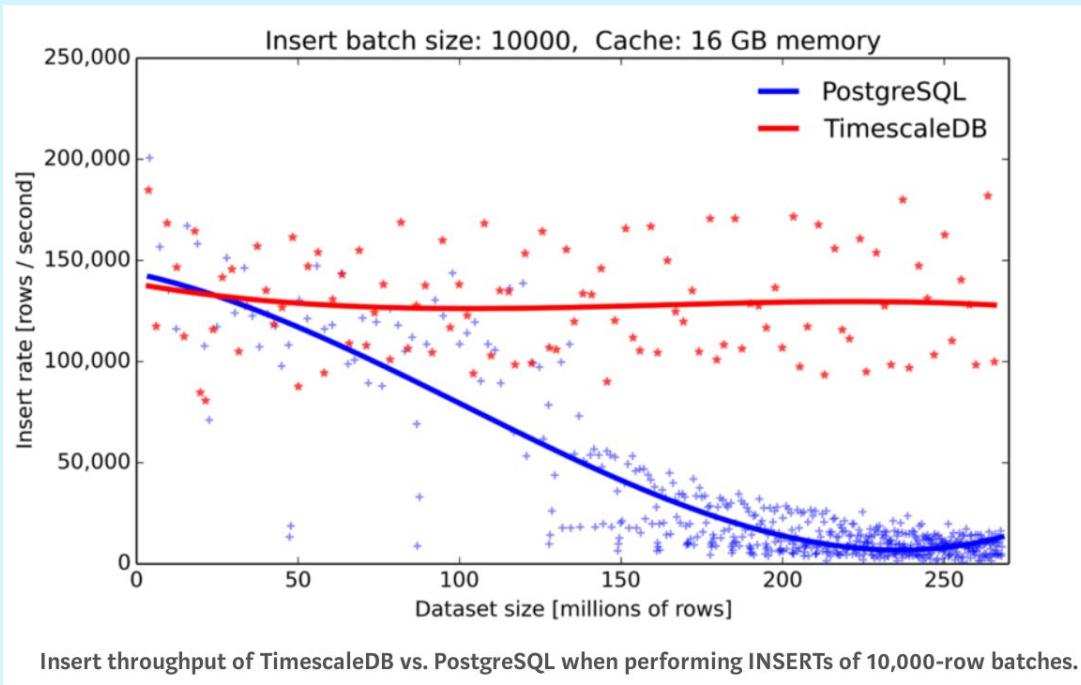
**Situação:** Uma empresa monitora 50 mil sensores instalados em 58 fábricas em um país. Cada sensor envia dado de vibração de uma máquina em intervalos de 100 ms. O dispositivo IoT envia dados para o banco 24h por dia na taxa de 100ms. A variável altera de acordo com o tempo e uma vibração excessiva contínua pode ser um indício de falha na máquina.

**Solução:** uso de banco de dados de séries temporais: TimescaleDB



# Exemplo de escolha de Banco de Dados

Solução: uso de banco de dados de séries temporais: TimescaleDB



# Banco de Dados em IoT

---

- **Múltiplas escritas** ao mesmo tempo (coisas > backend)
- **Poucas** requisições de **leitura** (frontend ou coisas)
- Dados são, normalmente, **séries temporais** (alto volume de dados)
- Aplicações podem requerer algum nível de segurança nos bancos de dados (criptografia do banco de dados e das requisições de escrita)
- Aplicações podem requerer **acesso** aos dados **em tempo real** (leitura e escrita com limitação de delay)
- Aplicações podem requerer **armazenamento interno** dos dados nas coisas (redundância interna)

## Why time-series databases for IoT?

Time-series databases are ideal for IoT applications that collect and analyze large amounts of timestamped data. Here are some scenarios where a time-series database would be a good choice as an IoT database:

1. Dispositivos IoT geram dados temporais (com timestamp) em alta velocidade
2. Aplicações e serviços exigem análise de dados em tempo real
  - Ex.: detecção de anomalias.
3. Aplicações e serviços exigem análise de histórico de dados

- Uso do termo “**banco de dados em tempo real**” para IoT
- Exemplo: serviço de BD do Firebase



<https://firebase.google.com/docs/database?hl=pt-br>

- Exemplo: serviço de BD do Firebase



## Principais recursos

### Em tempo real

Em vez de solicitações HTTP típicas, o Firebase Realtime Database usa a sincronização de dados. Sempre que os dados são alterados, todos os dispositivos conectados recebem essa atualização em milissegundos. Crie experiências colaborativas e imersivas sem se preocupar com códigos de rede.

O Realtime Database é um banco de dados NoSQL e, por isso, tem otimizações e funcionalidades diferentes de um banco de dados relacional. A API do Realtime Database foi desenvolvida para autorizar apenas operações que possam ser executadas com rapidez. Isso possibilita uma ótima experiência em tempo real que atende a milhões de usuários sem comprometer a capacidade de resposta. Por isso, é importante analisar como os usuários precisam acessar os dados e [estruturá-los adequadamente](#).

IoT Data Management Market size to grow by USD 69.28 billion from 2022 to 2027, The growing demand for smart homes to drive the market growth - Technavio



### GLOBAL IOT DATA MANAGEMENT MARKET 2023-2027



Market growth will **ACCELERATE** at a **CAGR** of

**14.1%**



Incremental growth (\$B)

**69.28**



The market is **FRAGMENTED** with several players occupying the market



Growth Contributed by  
**NORTH AMERICA**

**39%**



Growth for **2023**

**13.19%**

17000+ Reports covering niche topics. Read them at



Fonte:

<https://www.prnewswire.com/news-releases/iot-data-management-market-size-to-grow-by-usd-69-28-billion-from-2022-to-2027--the-growing-demand-for-smart-homes-to-drive-the-market-growth--technavio-301854630.html>

# O que discutimos hoje?

## Tópicos

- Importância do Gerenciamento de Dados em IoT
- Características dos Dados das Coisas e Metadados
- Etapas do Gerenciamento de Dados
- Conceitos de Banco de Dados
- Banco de Dados para IoT

# Dúvidas?

**Prof. Heitor Florencio**  
IMD/UFRN  
Sala 103 - nPITI/IMD  
[heitorm@imd.ufrn.br](mailto:heitorm@imd.ufrn.br)

# Referências

- **Minicursos / XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.** Organizado por Frank Augusto Siqueira, Lau Cheuk Lung, Fabíola Goncalves Pereira Greve, Allan Edgard Silva Freitas - Porto Alegre: SBC, 2016. Disponível em: <https://sbsrc2016.ufba.br/downloads/anais/MinicursosSBRC2016.pdf> .
- Microsoft, 2022. **What is Azure Time Series Insights Gen2.** Disponível em:  
<https://learn.microsoft.com/en-us/azure/time-series-insights/overview-what-is-tsi> . Acesso em: 09 de out de 2023.
- LMtx. **Metadata and telemetry data in IoT systems.** YouTube, 10 de set. de 2021. Disponível em:  
<https://www.youtube.com/watch?v=xIStoS87NKw> . Acesso em: 13 de out de 2023.
- Suraj S. **What is MetaData Management in IoT and why is it important?.** LinkedIn, 30 de maio de 2019. Disponível em:  
<https://www.linkedin.com/pulse/what-metadata-management-iot-why-important-suraj-s/> . Acesso em: 13 de out de 2023.
- HERNÁNDEZ-ROJAS, Dixys L. et al. **A plug-and-play human-centered virtual TEDS architecture for the web of things.** Sensors, v. 18, n. 7, p. 2052, 2018.
- Microsoft, 2023. **What are device templates?.** Disponível em:  
<https://learn.microsoft.com/en-us/azure/iot-central/core/concepts-device-templates> .
- XU, Alex. **Understanding Database Types.** ByteByteGo Newsletter, 2023. Disponível em:  
<https://blog.bytebytogo.com/p/understanding-database-types> . Acesso em: 13 de out de 2023.
- Stonebraker, Michael. **Visualizing IoT Data at Scale With Hopara and TimescaleDB.** Timescale, 18 de maio de 2023. Disponível em:  
<https://www.timescale.com/blog/visualizing-iot-data-at-scale-with-hopara-and-timescaledb/> .
- QuasarDB, 2023. **IoT Database 101: what, why, and how?.** Disponível em:  
<https://quasar.ai/2023/04/17/iot-database-101-what-why-and-how/> .
- Firebase, 2023. Disponível em: <https://firebase.google.com/docs/database?hl=pt-br> .