



certi®

## XIV Escola de Sistemas Embarcados - Embedded System School (ESSE)



# Tutorial:

---


Como gerar valor na indústria com dados de IoT utilizando uma stack de serviços simples?

## Introdução

No contexto da transformação digital das indústrias há diversos desafios para o desenvolvimento de software e transmissão de dados.

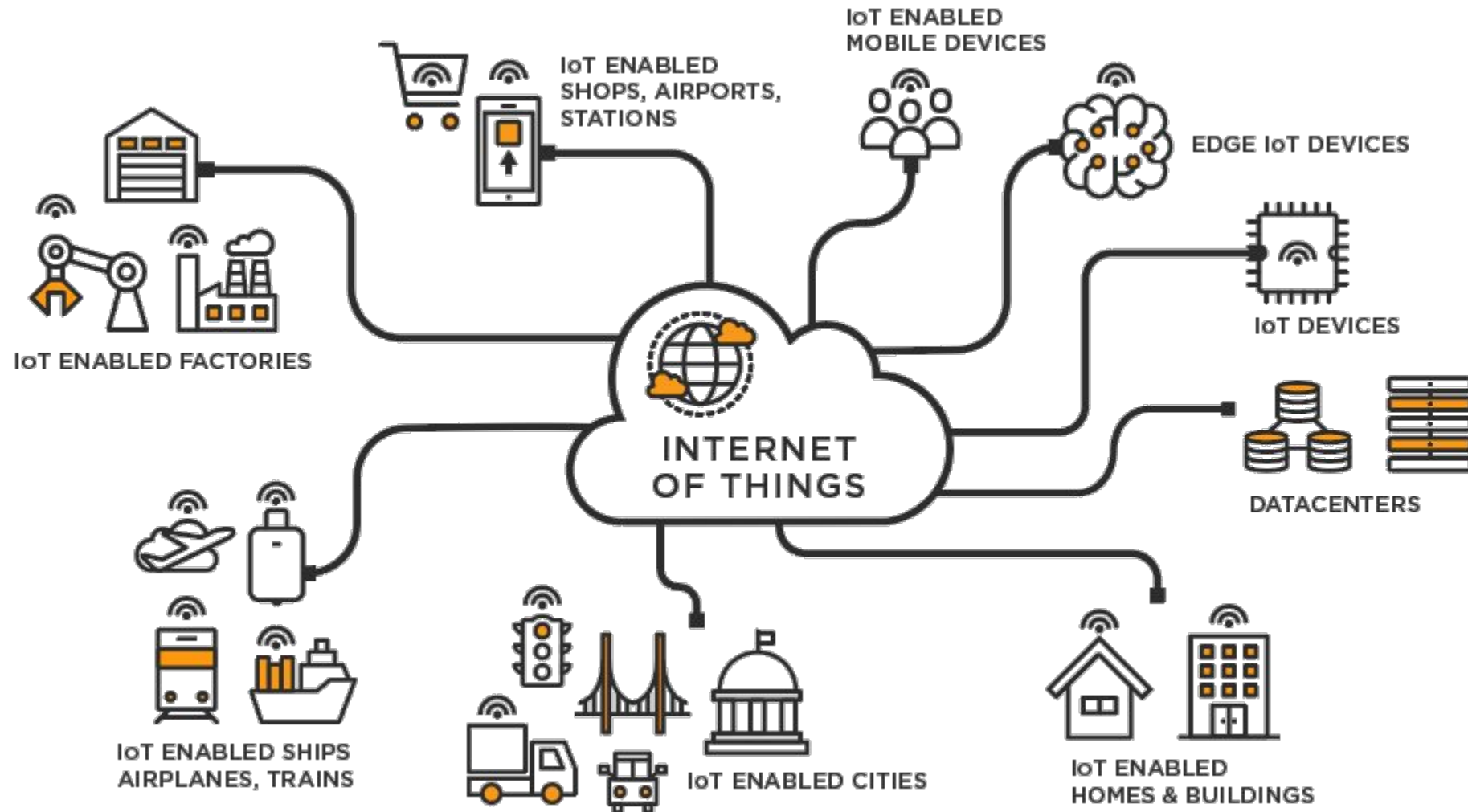
Neste tutorial será apresentada uma stack de dados simples para atacar os desafios de integração e agregação de valor em dados industriais.

Ao final os presentes serão convidados a interagir com a stack de dados desenvolvendo a coleta de dados com OPC-UA e MQTT, a integração de dados com o Node-red, o armazenamento com o InfluxDB e a visualização de dados com o Grafana.



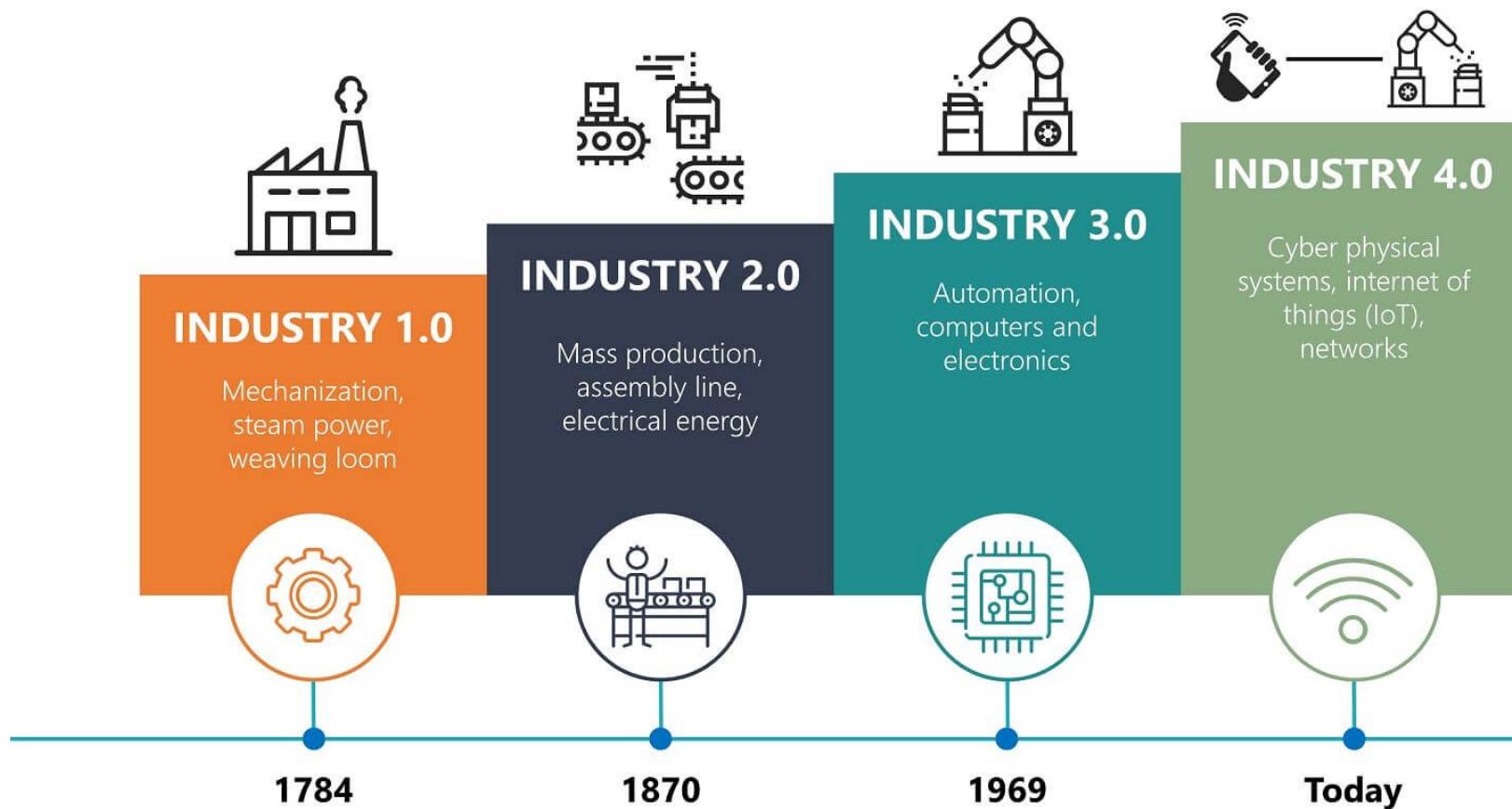
# IoT e IIoT

# ●● Internet of the Things



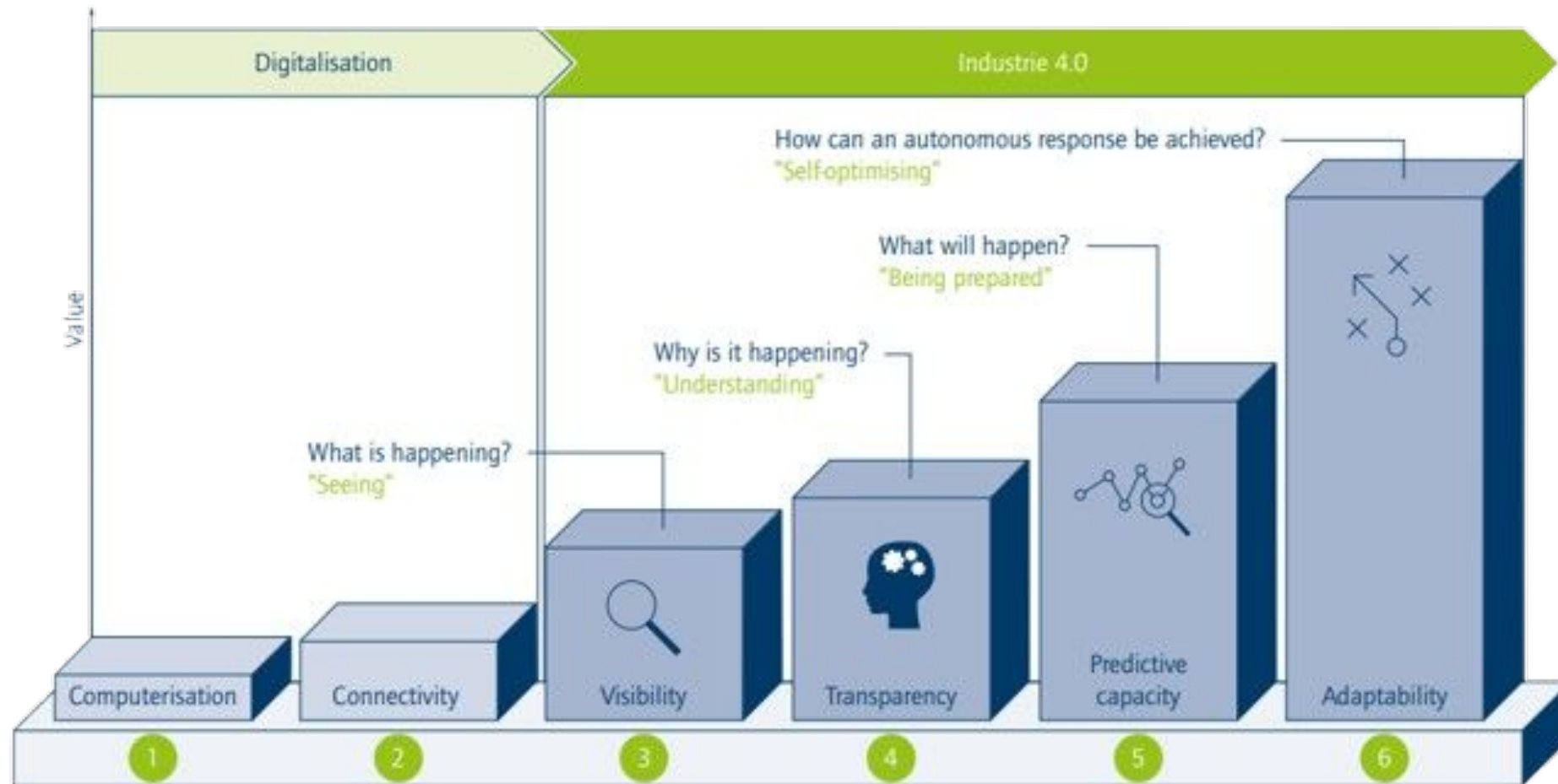
# Indústria 4.0

# ●● Indústria 4.0





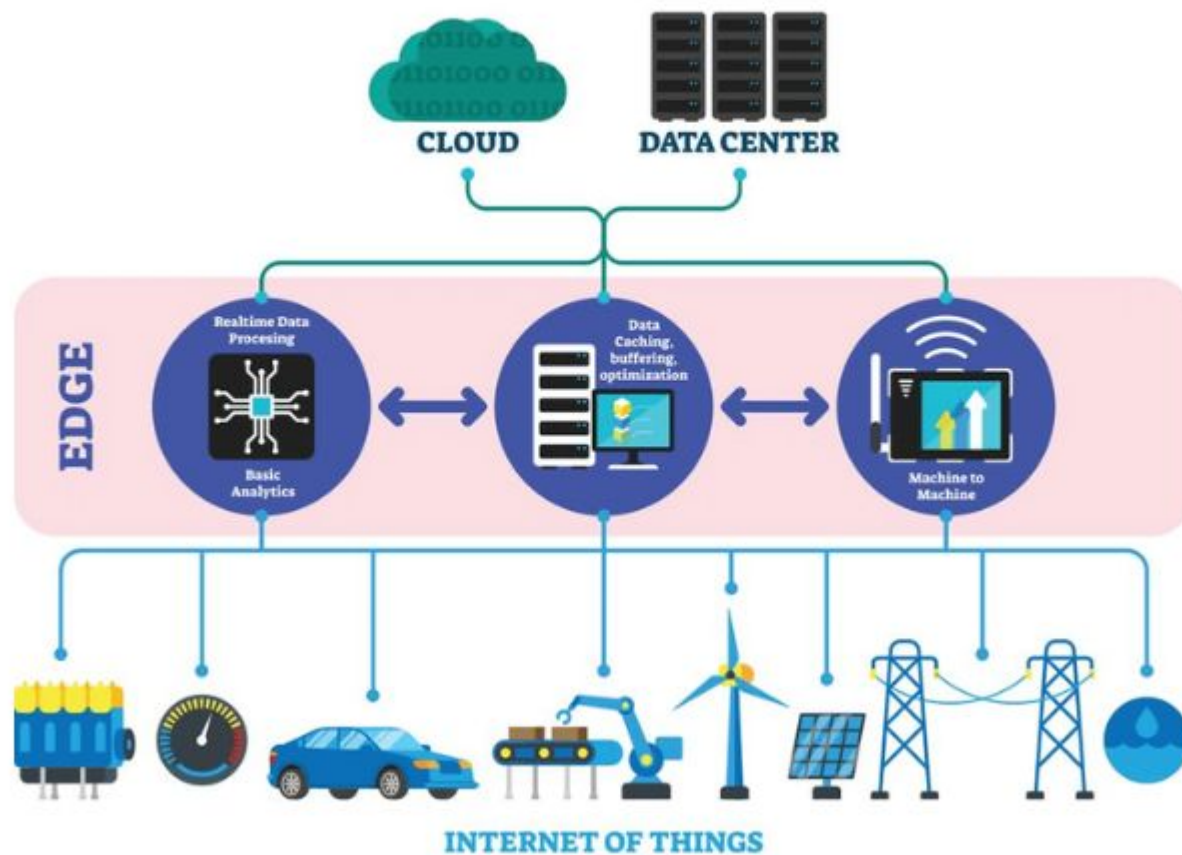
# ● ● Indústria 4.0





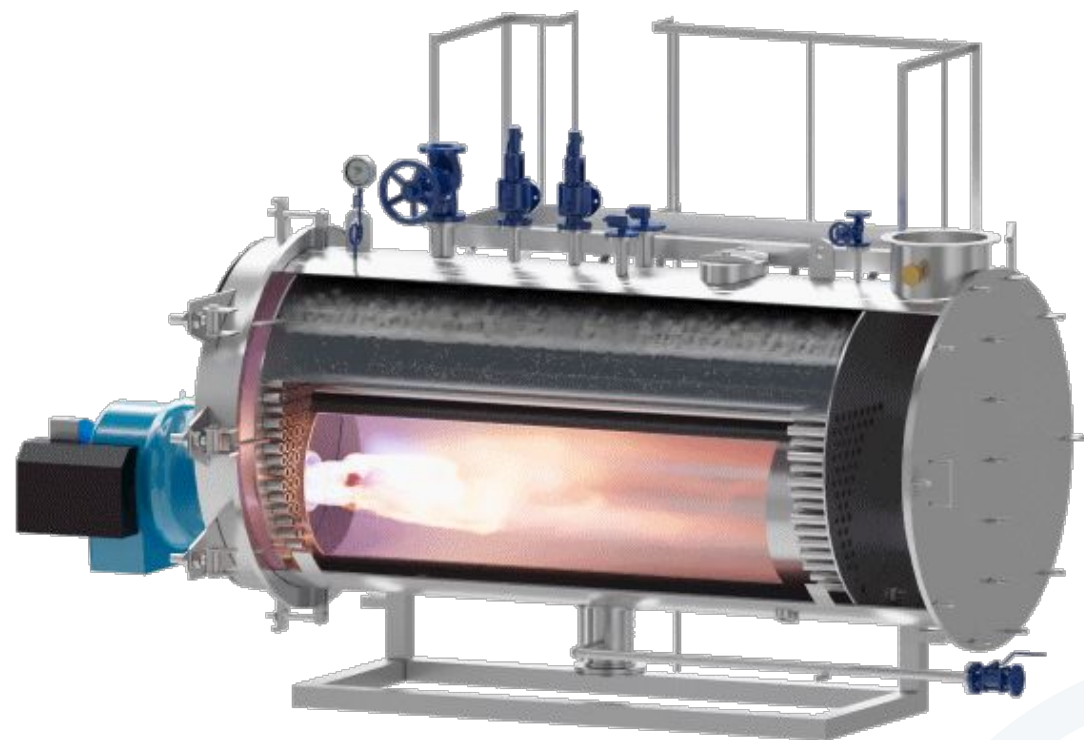
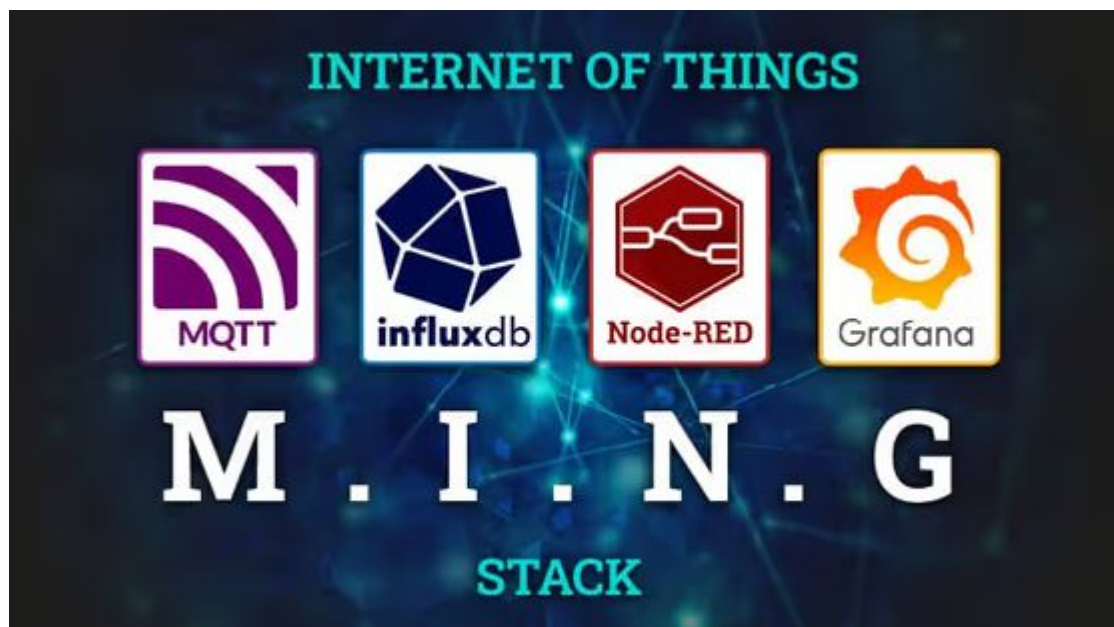
# Edge Computing

## ● ● EDGE Computing

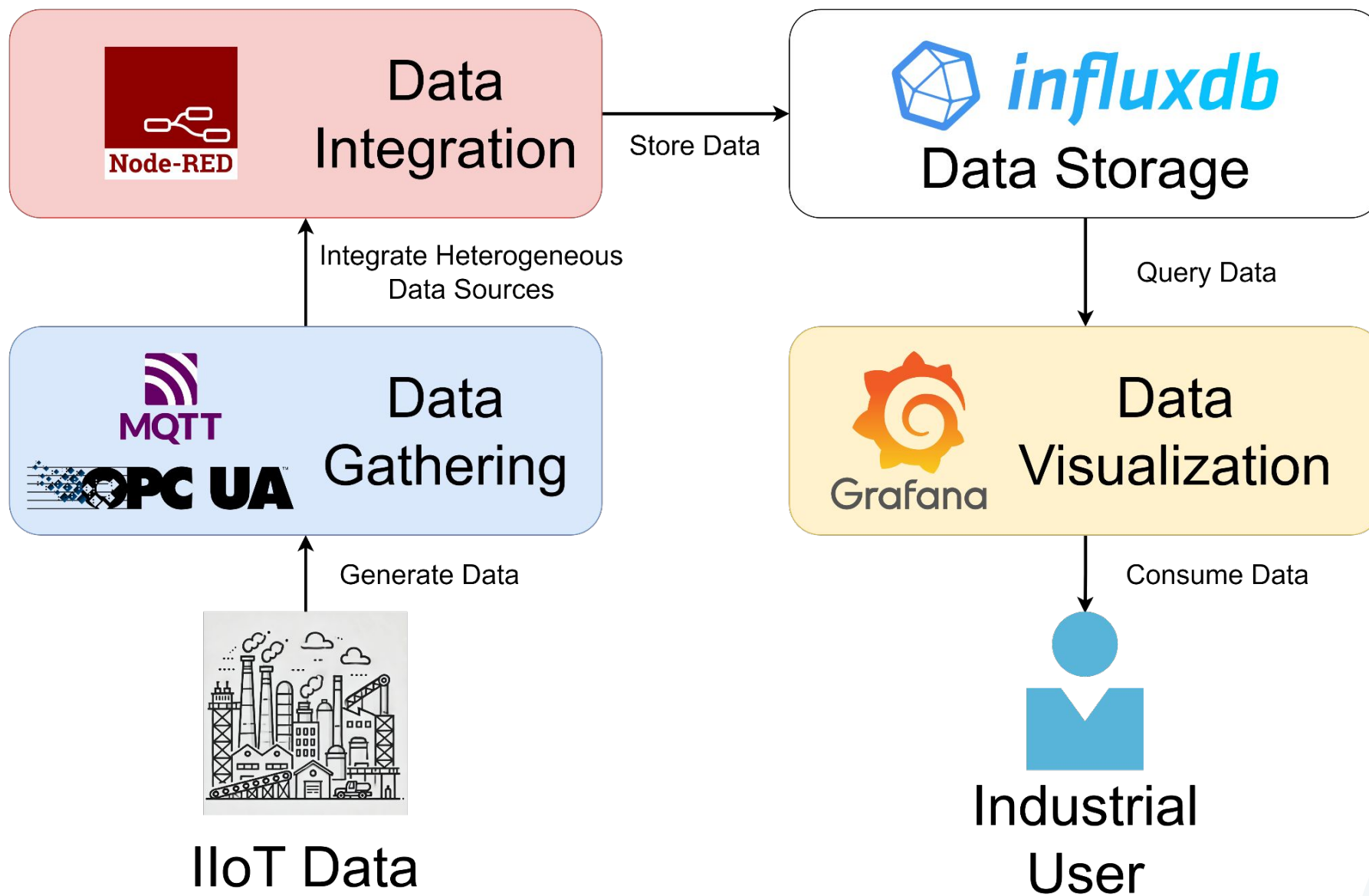


# Aplicação

## ●● Aplicação



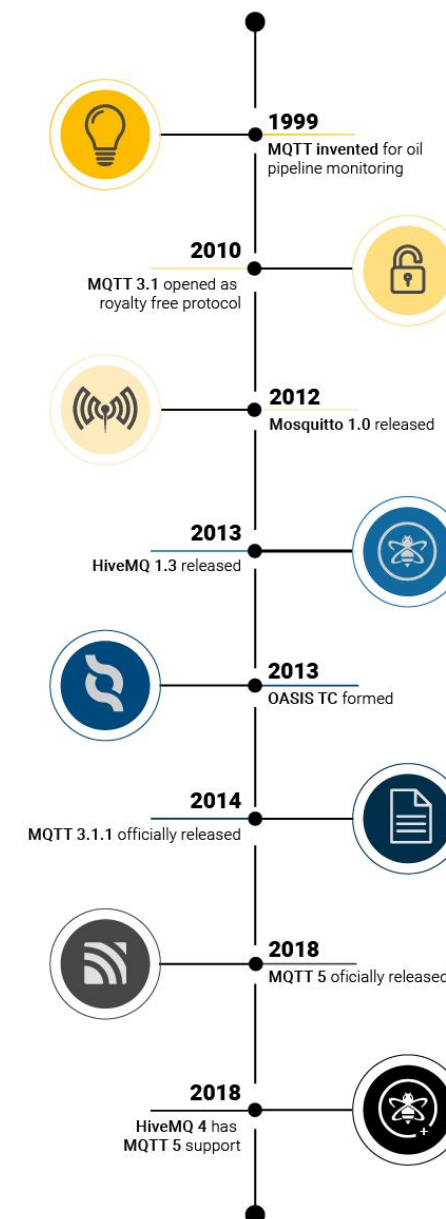


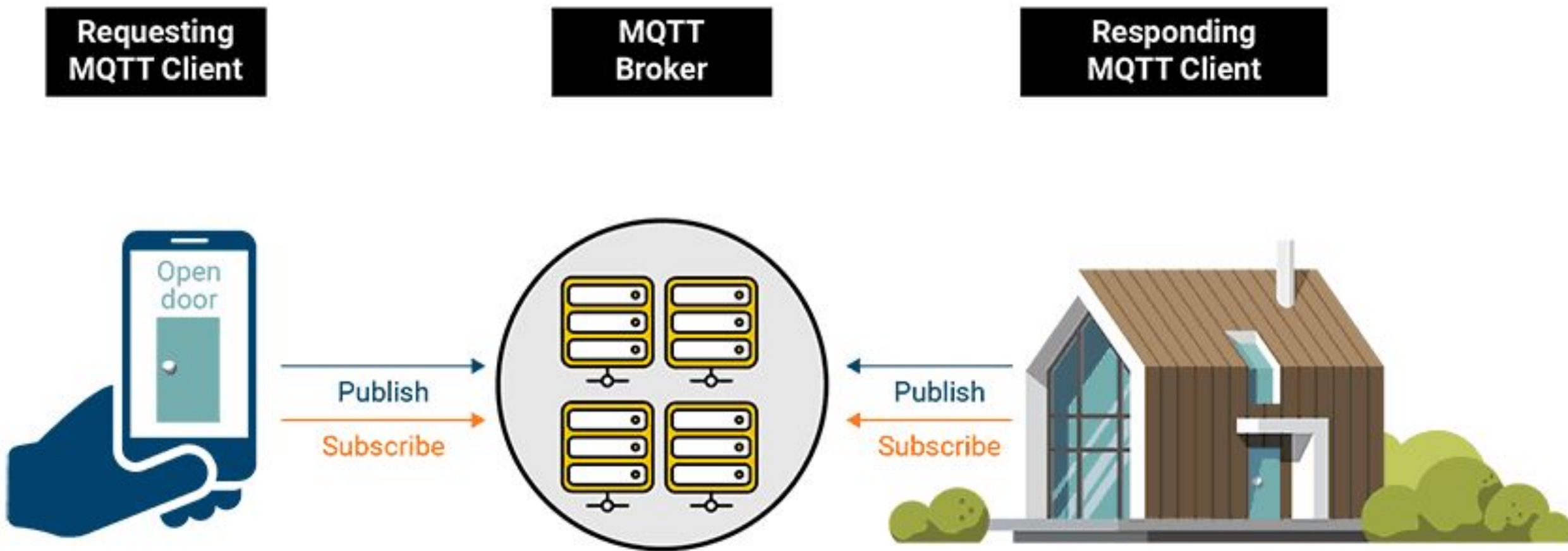


# Protocolo MQTT

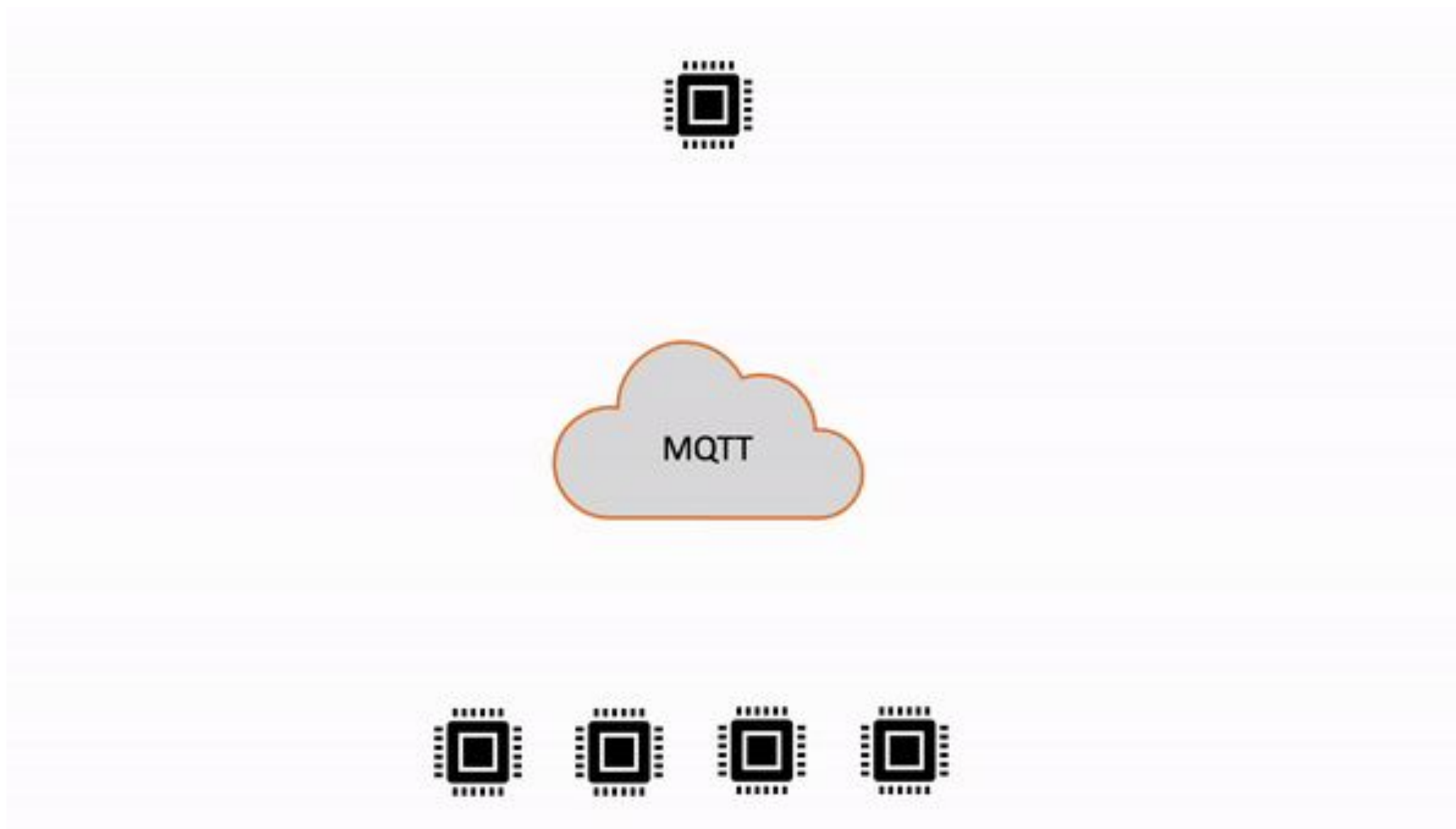
## ● ● Protocolo MQTT

- **MQTT (Message Queuing Telemetry Transport):** protocolo de mensagens leve, ideal para IoT.
- **Modelo Publish/Subscribe:** Dispositivos publicam em tópicos e assinam para receber mensagens de interesse.
- Baixo consumo de energia e largura de banda, ideal para dispositivos com recursos limitados.







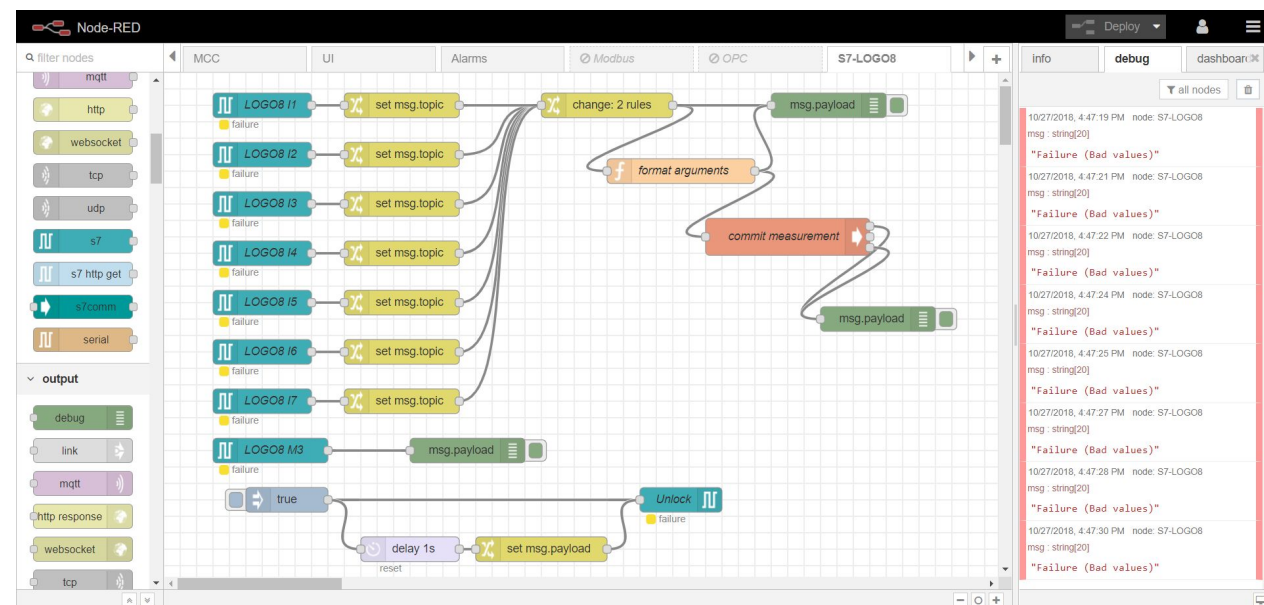
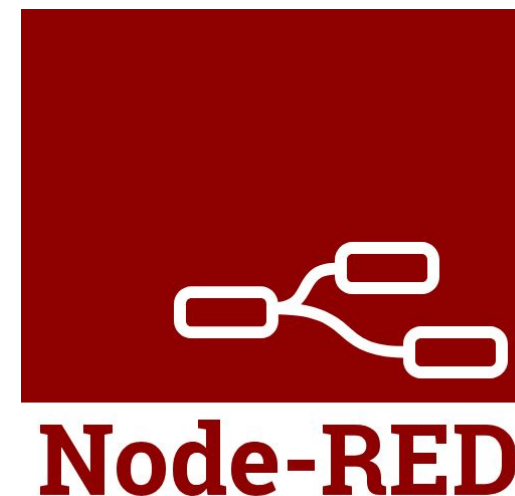


A large, solid blue circle that serves as a background for the text "Node-red".

# Node-red

# ●● Node-red

- Open Source
- Disponibilidade ampla de bibliotecas (pellets)
- Curva rápida de aprendizado
- Facilidade em criar uma interface gráfica (front-end)
- Simples de executar em qualquer máquina (PC, linux, containers)
- Simples debug
- Fácil de ler o código

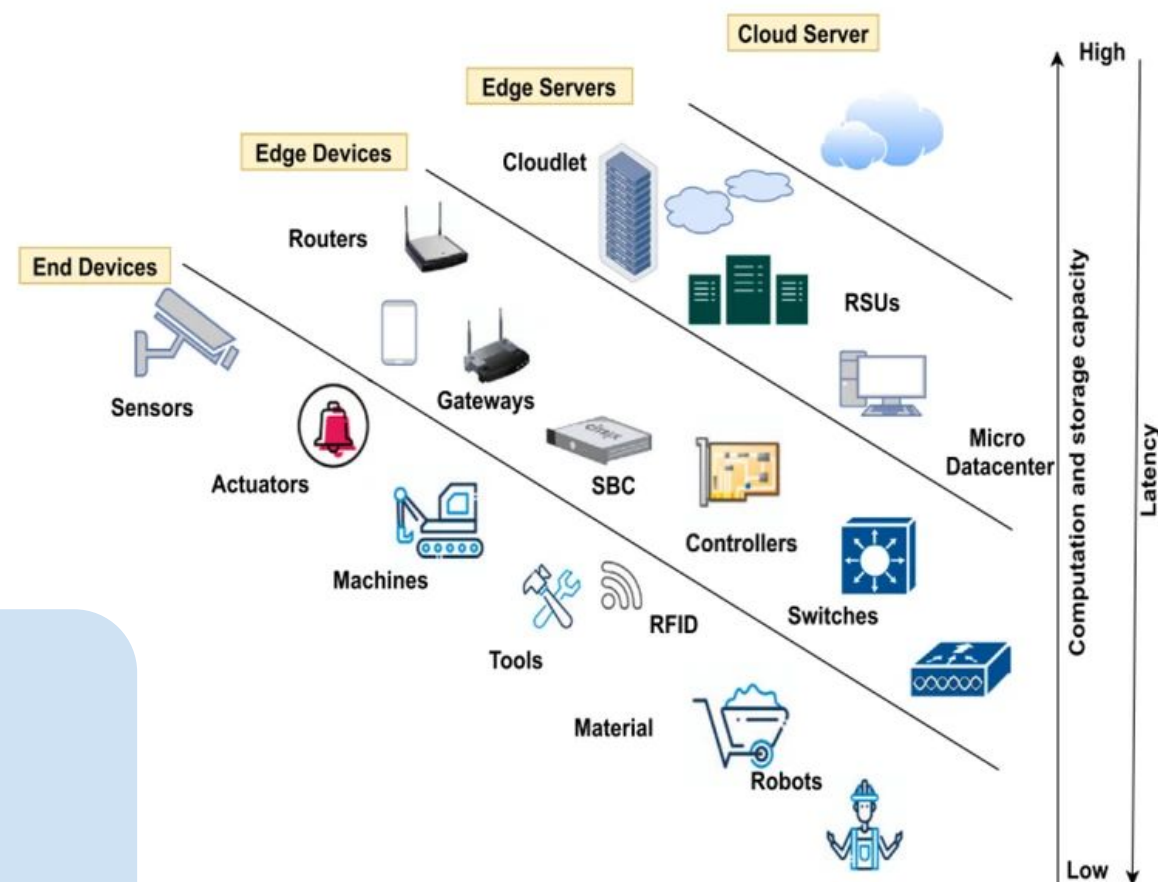


## ●● Node-red e a indústria 4.0

### Desafios

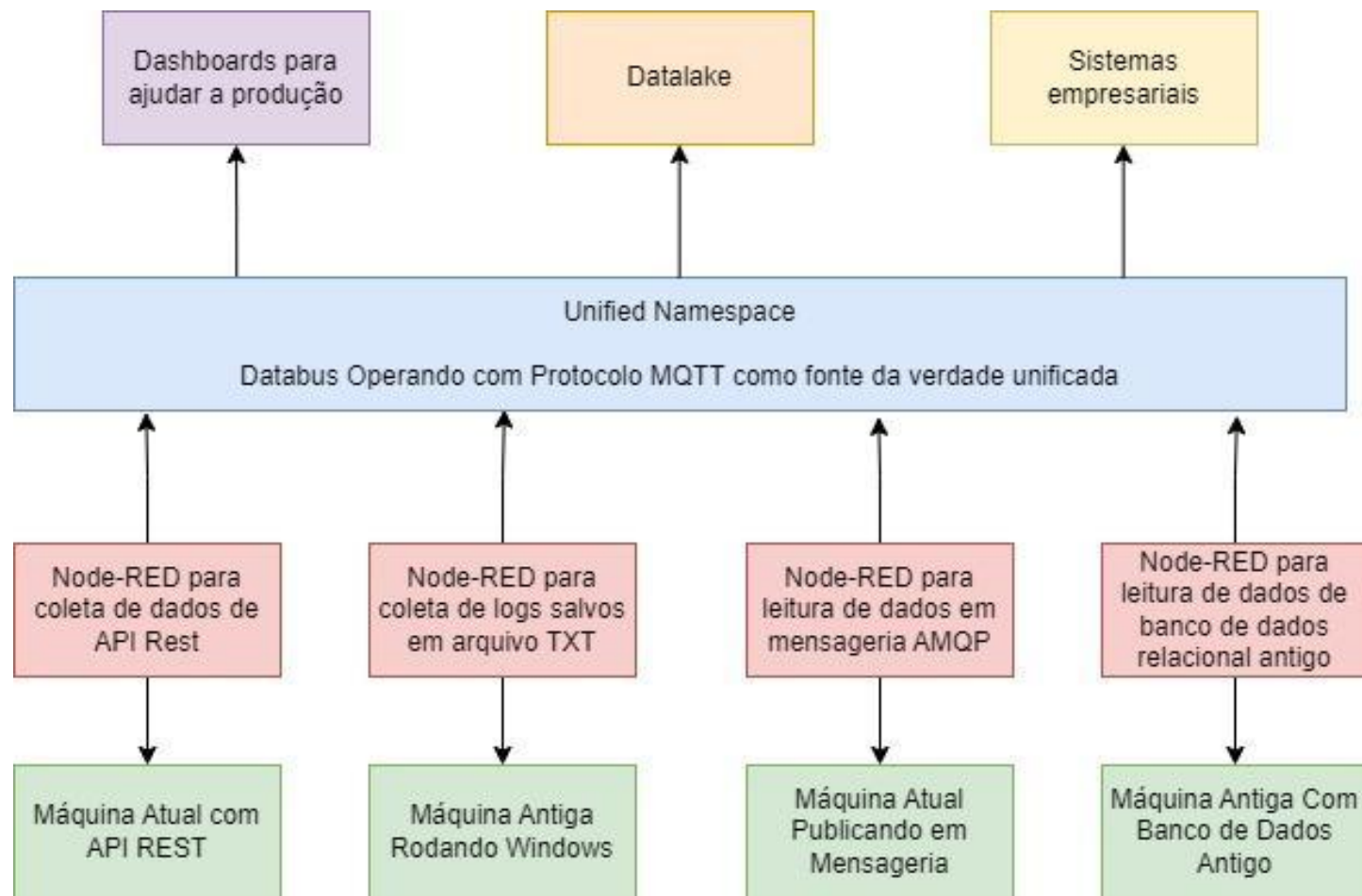
- Conectividade das Máquinas
- Interoperabilidade dos sistemas
- Ausência de normas e padrões da indústria 4.0
- Máquinas antigas
- Custos de equipamentos

O CPC propõe soluções para esses e outros desafios com Smart Smema, Unified Namespace, Arquitetura de Sistemas e também com o **Node-RED**, para citar alguns





## ●● Node-red e a indústria 4.0



# ●● Instalação

<https://nodered.org>

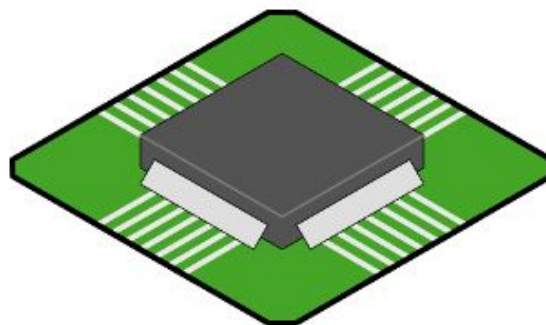
## Get Started

Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.



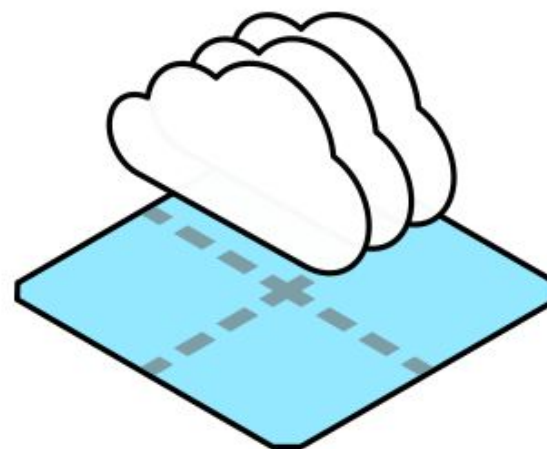
Run locally

- Getting started
- Docker



On a device

- Raspberry Pi
- BeagleBone Black
- Interacting with Arduino
- Android



In the cloud

- FlowFuse
- Amazon Web Services
- Microsoft Azure



# ●● Node-RED

A programação em node-red se baseia em **Fluxos**

Todo fluxo é um grafo direcionado composto por **nós (nodes)** por onde passam um objeto Javascript chamado **msg**

O objeto **msg** por padrão possui um atributo **payload** onde está os dados principais para o próximo nó no fluxo.

Porém um nó pode **acrescentar ou exigir** novos atributos ao objeto. Por exemplo: bloco MQTT

O bloco mqtt tem atributos opcionais, como **topic**, **qos** e **retain**.

## Inputs

<b>payload</b>	string   buffer
the payload to publish. If this property is not set, no message will be sent. To send a blank message, set this property to an empty String.	
<b>topic</b>	string
the MQTT topic to publish to.	
<b>qos</b>	number
0, fire and forget - 1, at least once - 2, once and once only. Default 0.	
<b>retain</b>	boolean
set to true to retain the message on the broker. Default false.	
<b>responseTopic</b>	string
MQTTv5: the MQTT response topic for the message	
<b>correlationData</b>	Buffer
MQTTv5: the correlation data for the message	
<b>contentType</b>	string
MQTTv5: the content-type of the payload	
<b>userProperties</b>	object
MQTTv5: any user properties of the message	
<b>messageExpiryInterval</b>	number
MQTTv5: the expiry time, in seconds, of the message	
<b>topicAlias</b>	number
MQTTv5: the MQTT topic alias to use	

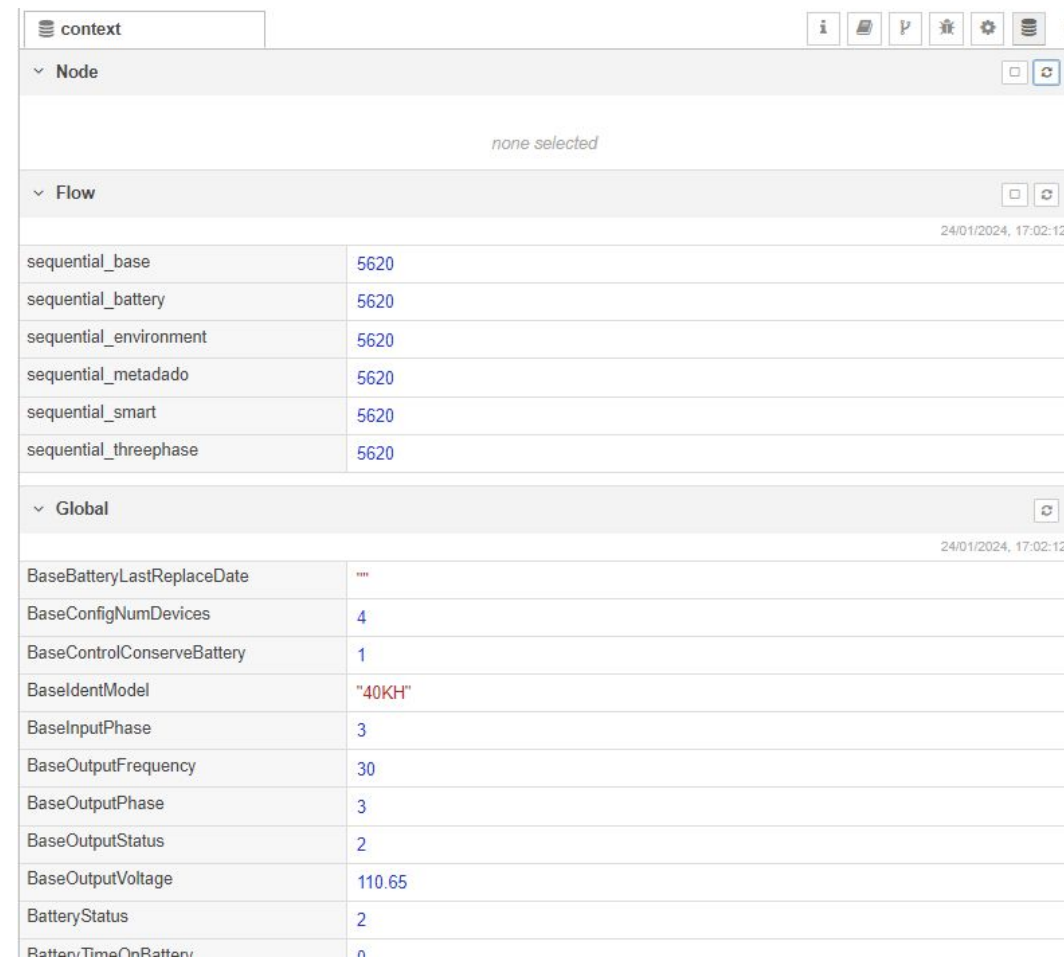
# ●● Escopos de Variáveis no Node-RED

No node-red as variáveis por padrão só são acessadas de dentro do nó, porém podem-se declarar variáveis fora do escopo do nó.

São os casos das variáveis de **flow** e **globais**

As variáveis de **flow** são enxergadas por todos os nós em um flow

As variáveis **globais** são enxergadas por todos os nós em qualquer flow



context	
Node	
none selected	
Flow	
sequential_base	5620
sequential_battery	5620
sequential_environment	5620
sequential_metadado	5620
sequential_smart	5620
sequential_threephase	5620
Global	
BaseBatteryLastReplaceDate	---
BaseConfigNumDevices	4
BaseControlConserveBattery	1
BaseIdentModel	"40KH"
BaseInputPhase	3
BaseOutputFrequency	30
BaseOutputPhase	3
BaseOutputStatus	2
BaseOutputVoltage	110.65
BatteryStatus	2
BatteryTimeOnBattery	0



## ●● Qual a linguagem de programação?

Apesar da programação ser em blocos, o Node-RED é implementado em Javascript utilizando o framework Node.js.

O Javascript é uma linguagem de programação interpretada, assim como o Python, e tem tipagem fraca. Ela tem suporte para programação funcional e orientada a objetos.

Surgiu em 1995 e é uma das tecnologias habilitadoras da internet. Apesar de antiga, ela sofre atualizações de tempos em tempos. A última foi em Junho de 2023.

JS

```
let a = 1;  
a = "foo";
```

# ●● Javascript para programadores

Todo o resto são objetos:

**Function**

**Array**

**Date**

**RegExp**

**Error**

Funções não são estruturas de dados especiais em Javascript, ela só são objetos que podem ser chamados.



## ●● Javascript para programadores

JS



```
console.log("Hello"[1] === "e"); // true
```

JS



```
const age = 25;  
console.log("I am " + age + " years old."); // String concatenation  
console.log(`I am ${age} years old.`); // Template literal
```

# ●● Javascript para programadores

```
JS

123 == "123"; // true
1 == true; // true

123 === "123"; // false
1 === true; // false
```

## let

variável disponível somente no escopo em que ela está inserida

## const

variável constante que não será alterada

## var

variáveis que fogem do escopo onde estão declaradas, causando comportamentos imprevisíveis. **Seu uso é desencorajado atualmente pela documentação.**

# ●● Javascript para programadores

if, else, for, switch case, while são similares ao C

try/catch e throw

JS

```
try {  
  buildMySite("./website");  
} catch (e) {  
  console.error("Building site failed:", e);  
}
```

JS

```
function buildMySite(siteDirectory) {  
  if (!pathExists(siteDirectory)) {  
    throw new Error("Site directory does not exist");  
  }  
}
```

# ●● Javascript para programadores

## Javascript objects

JS

```
const obj = {  
  name: "Carrot",  
  for: "Max",  
  details: {  
    color: "orange",  
    size: 12,  
  },  
};
```

JS

```
obj.details.color; // orange  
obj["details"]["size"]; // 12
```

JS

```
// Dot notation  
obj.name = "Simon";  
const name = obj.name;  
  
// Bracket notation  
obj["name"] = "Simon";  
const name = obj["name"];  
  
// Can use a variable to define a key  
const userName = prompt("what is your key?");  
obj[userName] = prompt("what is its value?");
```



# ●● Javascript para programadores

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language\\_overview](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_overview)

**Number:** usado para todo o tipo de números, ou seja, inteiros e floats.

**BigInt:** usado para inteiros muito grandes.

**String:** usado para texto

**Boolean:** verdadeiro e falso

**Symbol:** usado para identificadores únicos

**Undefined:** identifica que a variável foi declarada, mas não tem valor atribuído

**Null:** indica que é um não-valor.

```
var a = 10, b = null, c;  
console.log(a); // 10  
console.log(b); // null  
console.log(c); // undefined  
console.log(d); // Uncaught ReferenceError: d is not defined
```



The InfluxDB logo, which consists of a large blue circle with the text "InfluxDB" in white, bold, sans-serif font in the center. To the right of the circle are three light gray, curved, overlapping shapes that resemble stylized data waves or a signal.

## ● ● InfluxDB *influxdb*

- Base de dados de séries temporais preparada para Big Data com altas taxas de escrita e armazenamento eficiente
- Utiliza um dialeto SQL e a Flux Query Language com agregações avançadas de dados
- Possui uma versão totalmente open source e uma versão gerenciada na cloud com mais recursos.
- UI com capacidade de geração de dashboards e análise de dados simples.
- Integração com Grafana e Node-red
- Esquema flexível para facilitar integração de fontes de dados
- Possui tagueamento para indexação





A large, solid blue circle containing the word "Grafana" in a white, bold, sans-serif font, centered within the circle.

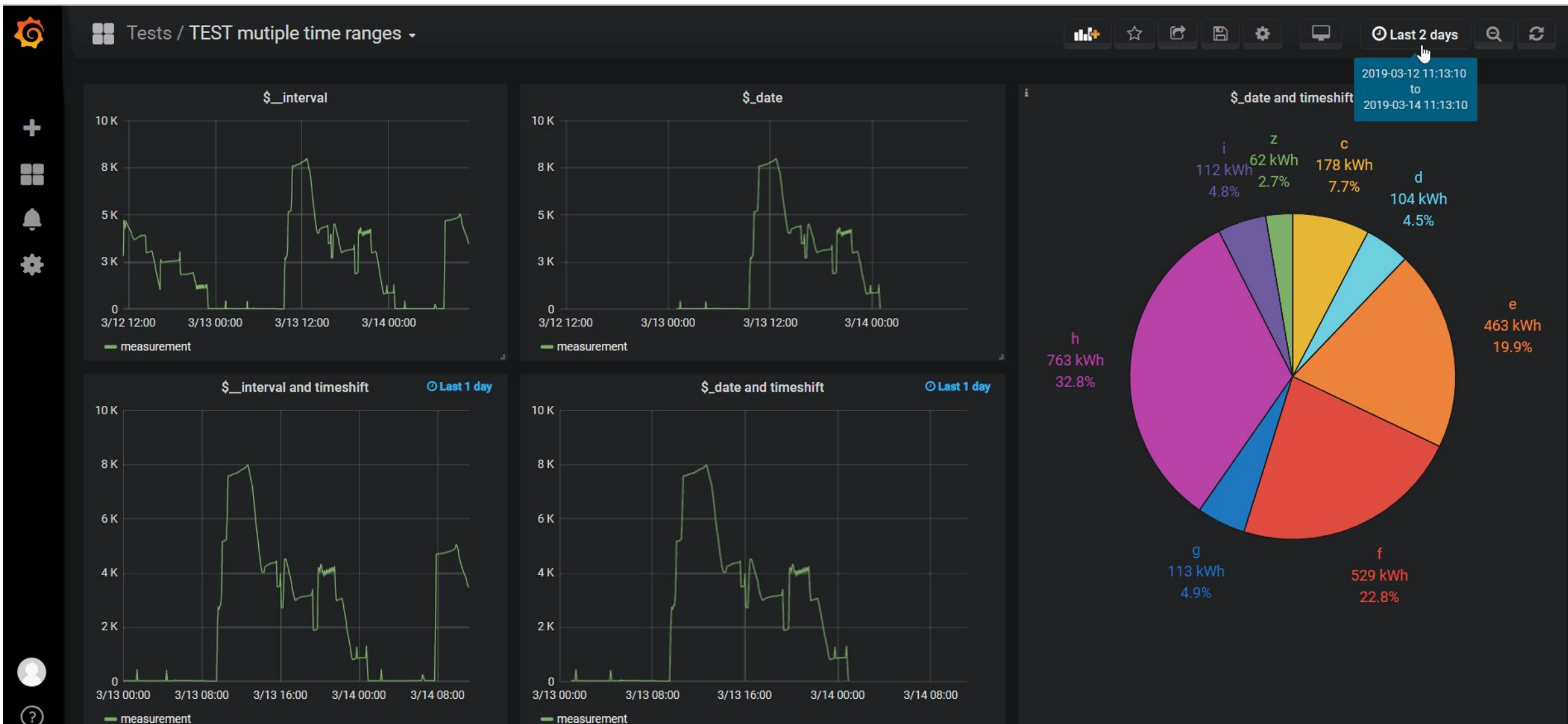
# Grafana

## ● ● Grafana

- Ferramenta de visualização de dados em tempo real open source
- Suporte a múltiplas bases de dados, como relacionais, não relacionais e object storages (S3, Minio)
- Dashboards personalizados
- Ampla comunidade disponibiliza dashboards prontos para aplicações de monitoramento de infraestrutura e serviços



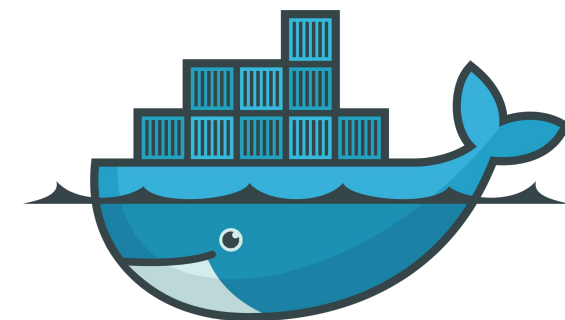




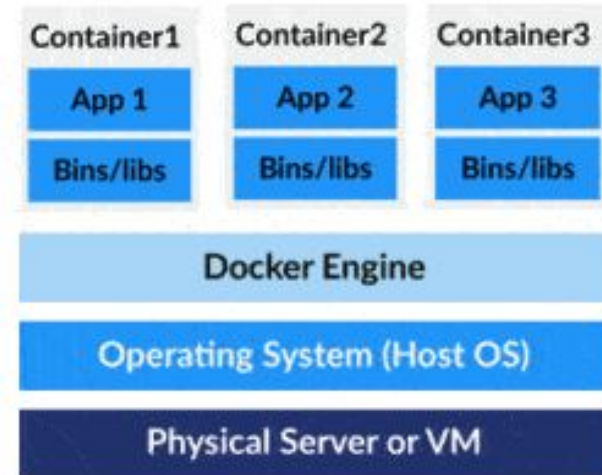
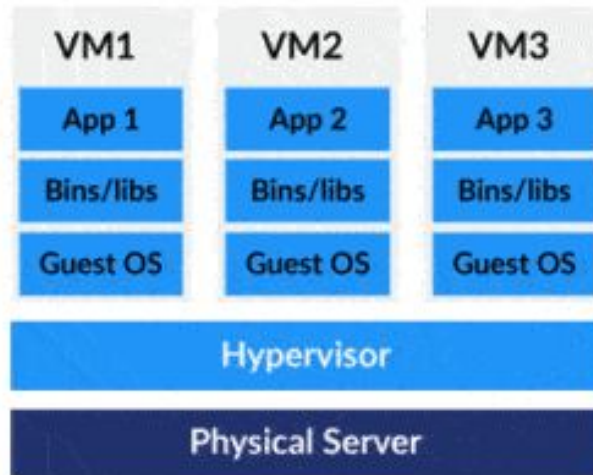
# Docker e Docker Compose

## ●● Docker e Docker Compose

- Plataforma que facilita a criação, empacotamento e execução de aplicações de forma portátil e consistente em qualquer ambiente.
- Revolucionou a distribuição de software por facilitar o desenvolvimento e deploy de aplicações.
- Pode-se dizer que é uma evolução das máquinas virtuais, porém mais leve e simples de provisionar.
- Os containers são pacotes de bibliotecas e softwares que estão salvos em uma imagem extremamente leve.
- Utilizam um sistema operacional como base (ex: VM linux) para sua execução, por isso são mais leves.

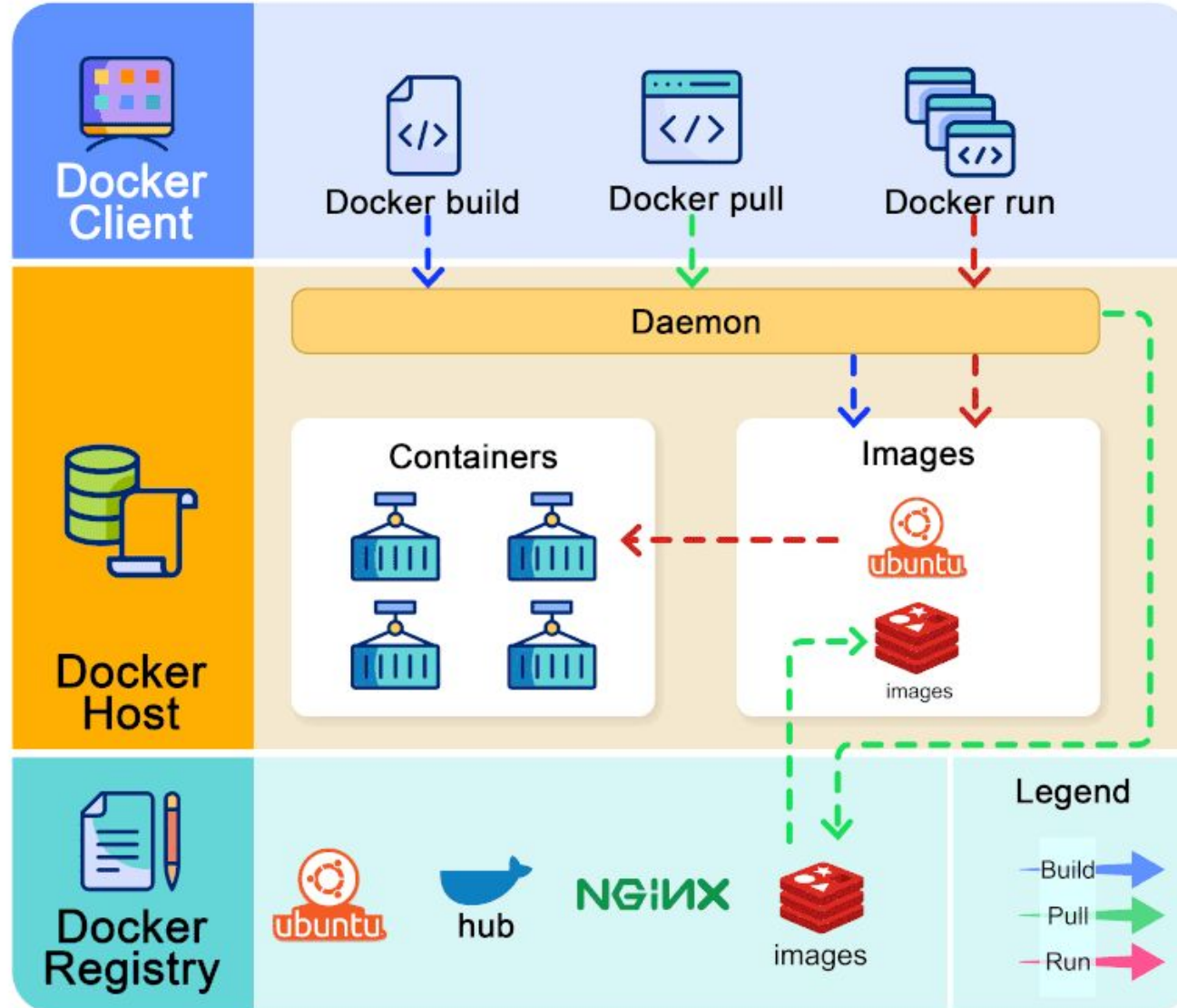


docker



# How does Docker Work ?

 [blog.bytebytego.com](https://blog.bytebytego.com)



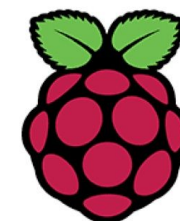
A large, solid orange circle is centered on the slide. Inside the circle, the word "Recursos" is written in a white, bold, sans-serif font.

# Recursos



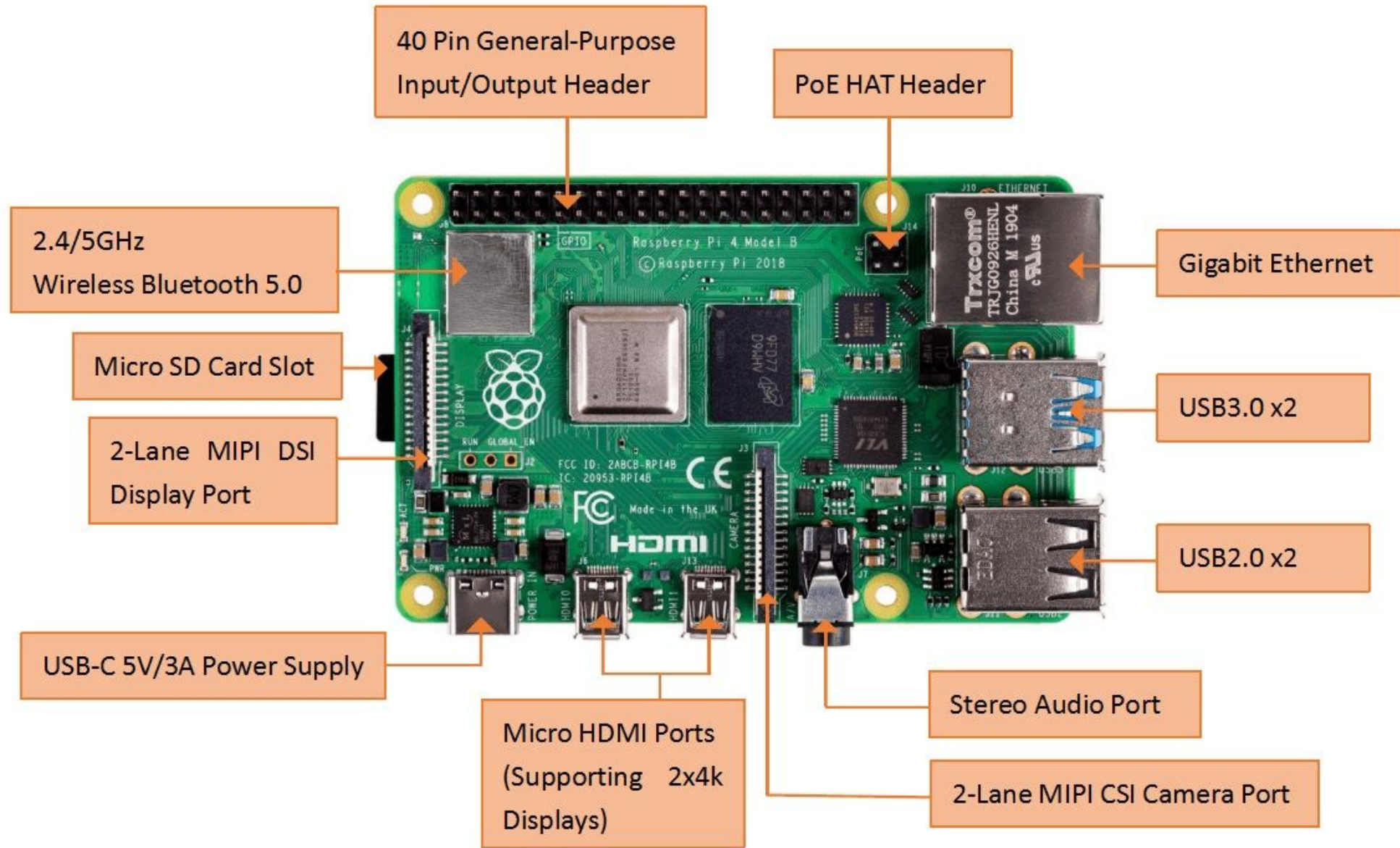
## ●● Raspberry Pi

- A Raspberry Pi é um computador de placa única (SBC - Single Board Computer) compacto, acessível e versátil, amplamente utilizado em projetos educacionais, pessoais e profissionais. Lançada inicialmente pela Raspberry Pi Foundation, foi projetada para ensinar programação e eletrônica, mas rapidamente se tornou popular em diversas aplicações.
- Possui diversos modelos que podem incluir WiFi, Bluetooth, Ethernet, conexões para periféricos como USB, HDMI, cartão SD.
- **Disponibilidade de GPIO para interação com sensores e atuadores.**



# Raspberry Pi





# Obrigado!



Arthur Raulino Kretzer  
Pesquisador de Sistemas Fabris Inteligentes  
CPC  
ark@certi.org.br



João Victor Melquiades Sátiro  
Coordenador De Área  
CPC  
jvq@certi.org.br