

Implementação e análise de dados de uma rede IoT

RAFAEL NEVES MIRANDA

novembro de 2020

IMPLEMENTAÇÃO E ANÁLISE DE DADOS DE UMA REDE IOT

Rafael Neves Miranda

Departamento de Engenharia Electrotécnica
Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização em Telecomunicações

Relatório elaborado para satisfação parcial dos requisitos da Unidade Curricular de
Tese/Dissertação do Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: Rafael Neves Miranda, Nº 1234567, 1140512@isep.ipp.pt

Orientação científica: Dr. Paula Viana, pmv@isep.ipp.pt

Empresa:

Supervisão: Dr. Carlos Alcobia, carlos.alcobia@pt.controlar.com



Departamento de Engenharia Electrotécnica

Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização em Telecomunicações

2020

Agradecimentos

Gostaria de agradecer em primeiro lugar, à Controlar por confiar em mim e me permitir integrar e coordenar este interessante projeto europeu, numa temática nova para a empresa, com um enorme potencial para aportar valor para os produtos já desenvolvidos pela mesma.

Assim sendo, tenho de estender o agradecimento em particular para o Dr. Carlos Alcobia, meu orientador na empresa, e para a Dr. Carla Pereira, minha superior. Ambos auxiliaram-me imenso ao longo da criação e desenvolvimento do presente trabalho.

Agradecer também à minha equipa, que sempre se mostrou disponível e sempre através das suas opiniões e recomendações me permitiram direcionar o trabalho no caminho correto.

Ainda num nível técnico, gostaria de agradecer à minha orientadora a Dr. Paula Viana, por toda a ajuda e disponibilidade ao longo da realização deste trabalho. As suas sugestões sempre pertinentes permitiram aumentar o nível de qualidade do trabalho.

Num nível mais pessoal, gostaria de agradecer aos meus pais e irmã, por todo o apoio, motivação e interesse ao longo do meu percurso académico.

Por fim, gostaria de deixar um agradecimento especial à minha namorada, por todo o apoio, motivação e insistência, que foram sem dúvida cruciais para o desenvolvimento deste trabalho.

Resumo

A indústria 4.0, também denominada como IIoT, através da implementação massiva de sensores e dispositivos IoT, leva à produção de grandes quantidades de dados (*big data*). O processamento destes dados é um grande desafio à indústria atual pois obriga a grandes capacidades de armazenamento e de recursos computacionais. Assim sendo, as técnicas de *big data analytics* vêm tentar fazer face a estes desafios.

É neste contexto que surge o projeto europeu IS4SE, na medida em que o seu principal objetivo é desenvolver uma infraestrutura de IIoT numa empresa de siderurgia com as ferramentas capazes de analisar dados. A equipa de I&D da Controlar encontra-se envolvida na realização deste projeto. Assim, este trabalho pretende executar algumas das principais tarefas atribuídas à Controlar no âmbito deste projeto, incluindo a criação de uma rede IoT e uma aplicação capaz de analisar os dados provenientes desta rede.

Numa primeira fase foi desenvolvida a rede IoT, tendo por base dois protocolos principais: o LoRaWAN e o Wi-Fi via MQTT. Por um lado, utilizou-se o LoRaWAN devido ao seu baixo consumo e longo alcance. Por outro lado, recorreu-se ao Wi-Fi fazendo uso do protocolo MQTT devido à flexibilidade que este oferece. De seguida, criou-se uma *gateway* multiprotocolar capaz de gerir estes dois protocolos e desenvolveu-se um servidor para armazenamento, no mesmo local, de toda a informação proveniente dos diferentes protocolos da rede IoT.

A fase final do trabalho incidiu sobre o desenvolvimento de uma aplicação *web* capaz de tratar os dados provenientes dos diferentes nós conectados à rede IoT, de analisá-los e, conseqüentemente, criar estatísticas sobre os mesmos. Esta análise, engloba, igualmente, a aplicação de algoritmos de *machine learning* na informação adquirida.

De forma a testar e validar a robustez da rede IoT e o bom funcionamento da aplicação *web*, foram integrados nós LoRaWAN e Wi-Fi na rede IoT criada ao longo de alguns meses, o que gerou um grande conjunto de dados a serem analisados pela aplicação *web*.

Palavras-Chave

IIoT, IoT, LoRaWAN, MQTT, aplicação web, análise de dados, *machine learning*.

Abstract

Industry 4.0, also known as IIoT, works through the massive implementation of IoT sensors and devices, which lead to the production of massive quantities of data (big data). Big data processing presents a great challenge to the current industry because it requires large storage and computational resources. big data analytics is related to the approaches developed to face these challenges.

It is in this context that the European project IS4SE emerges, having as main goal to develop an IIoT infrastructure capable of analyzing data in a steel company. Controlar's I&D team is involved in carrying out this project. This thesis intends to carry out some of the main tasks assigned to Controlar in the scope of this project, in particular, the creation of an IoT network and a web application capable of analyzing the data coming from this network.

In a first phase, an IoT network, based on two main protocols, was developed: LoRaWAN and Wi-Fi via MQTT. On the one hand, LoraWAN was used due to its low consumption and long reach. On the other hand, Wi-Fi was used making use of the MQTT protocol because of the flexibility it offers. Then, a multiprotocol gateway capable of managing these two protocols was created. It was also necessary to develop a server that would store, in the same location, all the information from the different protocols of the IoT network.

In a later phase, a web app, capable of handling data from different nodes connected to the IoT network, to analyze them and, consequently, to create statistics on them, was developed. This analysis includes the use of machine learning algorithms in the acquired information.

In order to test and validate the strength of the IoT network and the good functioning of the web app, LoRaWAN and Wi-Fi nodes were integrated into the IoT network enabling the collection of a large set of data, during a few month, to be analyzed by the web app.

Keywords

IIoT, IoT, LoRaWAN, MQTT, Web Application, data analysis, machine learning.

Índice

1. INTRODUÇÃO	1
1.1.CONTEXTUALIZAÇÃO	2
1.2.OBJETIVOS	4
1.3.ORGANIZAÇÃO DO RELATÓRIO	5
2. IIOT	6
2.1.PILHA PROTOCOLAR	6
2.1.1. <i>Camadas Inferiores</i>	7
2.1.2. <i>Camada Intermédia</i>	9
2.1.3. <i>Camadas Superiores</i>	9
2.2.TECNOLOGIAS RELEVANTES PARA IMPLEMENTAÇÃO	11
2.2.1. <i>MQTT</i>	11
2.2.2. <i>LoRaWAN</i>	14
2.2.2.1. <i>Chirp Spread Spectrum</i>	15
2.2.2.2. <i>Stack LoRaWAN</i>	17
2.2.2.3. <i>Tipo de dispositivos</i>	18
2.2.2.4. <i>Métodos de Ativação dos Sensores</i>	19
2.2.2.5. <i>Parâmetros Regionais</i>	20
2.2.2.6. <i>Trama Cayenne LPP</i>	21
3. BIG DATA ANALYTICS	25
3.1.DEFINIÇÃO DE <i>BIG DATA</i>	25
3.2.DEFINIÇÃO DE <i>BIG DATA ANALYTICS</i>	27
3.3. <i>MACHINE LEARNING</i>	28
3.3.1. <i>Machine Learning Relacionado com o Consumo Energético</i>	30
4. IMPLEMENTAÇÃO DA REDE IOT	32
4.1.ARQUITETURA <i>GENÉRICA</i>	32
4.2.SERVIDOR	33
4.3.GATEWAY	36
4.3.1. <i>Arquitetura e Hardware</i>	36
4.3.2. <i>Interface Web de Configuração</i>	40
4.4. NÓS	45
4.4.1. <i>Nós LoRaWAN</i>	45
4.4.2. <i>Nós Wi-Fi/Ethernet</i>	46
4.4.3. <i>Nós de PLC</i>	48

4.5. LEITURA E VISUALIZAÇÃO DOS DADOS DOS SENSORES	50
5. ANÁLISE E TRATAMENTO DOS DADOS	52
5.1. MODELO DE DADOS	53
5.1.1. <i>Tecnologias Utilizadas</i>	53
5.1.2. <i>Principais Funções</i>	54
5.2. ESTRUTURA DO SOFTWARE DA PLATAFORMA	57
5.3. FUNCIONALIDADES DA APLICAÇÃO WEB	58
5.3.1. <i>Home</i>	58
5.3.2. <i>Databases</i>	59
5.3.3. <i>Alerts</i>	61
5.3.4. <i>Sampling Rate</i>	62
5.3.5. <i>Config User</i>	63
5.3.6. <i>Aplicação de algoritmos de machine Learning</i>	64
6. APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS	69
6.1. DESCRIÇÃO DO CENÁRIO DE TESTE	69
6.2. CONFIGURAÇÃO DOS NÓS NA REDE IOT	71
6.3. MANUAL DE UTILIZAÇÃO E RESULTADOS DO CENÁRIO DE TESTE	75
6.3.1. <i>Home</i>	75
6.3.2. <i>Databases</i>	76
6.3.3. <i>Alerts</i>	84
6.3.4. <i>Sampling Rate</i>	85
6.3.5. <i>Config User</i>	86
6.3.6. <i>Aplicação de algoritmos de Machine Learning</i>	87
6.3.6.1. <i>Energy_Meter</i>	87
6.3.6.2. <i>Smart_Composter</i>	89
7. CONCLUSÕES	91

Índice de Figuras

Figura 1	Pilha protocolar IIoT [8]	7
Figura 2	Exemplo de uma arquitetura MQTT [20]	11
Figura 3	(Esquerda) Sinal na frequência (direita) Sinal no tempo [26]	15
Figura 4	Arquitetura de rede LoRaWAN [29]	18
Figura 5	Os três Vs do <i>big data</i> [40]	26
Figura 6	Processo para extrair informação relevante da <i>big data</i> [42]	28
Figura 7	Diferentes tipos de <i>Machine Learning</i> e os correspondentes algoritmos [48]	30
Figura 8	Arquitetura típica de uma rede IoT	33
Figura 9	Arquitetura típica de uma rede IoT	33
Figura 10	Arquitetura do servidor ChirpStack [55]	34
Figura 11	<i>Arquitetura da gateway</i>	36
Figura 12	<i>Gateway desenvolvida</i>	37
Figura 13	Concentrador ic880a [60]	37
Figura 14	PoE Hat que alimenta a gateway [62]	39
Figura 15	MSC do <i>Uplink</i> das mensagens LoRa	40
Figura 16	Aplicação Web da Gateway	41
Figura 17	Painel dos serviços LoRaWAN	42
Figura 18	Painel de configuração do ChirpStack Gateway Bridge	42

Figura 19	Painel de configuração do Packet Forwarder	43
Figura 20	Painel de configuração do AP	43
Figura 21	Painel de configuração do AP	44
Figura 22	Exemplo de configuração do Telegraf tendo o MQTT como Input	46
Figura 23	Exemplo do influxDB Line Protocol [64].	47
Figura 24	Ficheiro de configuração do plug-in Modbus TCP/IP	49
Figura 25	Ficheiro de configuração do plug-in OPC-UA	50
Figura 26	Exemplo de um <i>dashboard</i> criado no Chronograf [71]	51
Figura 27	Menu lateral da aplicação <i>web</i>	52
Figura 28	Exemplo das tabelas da base de dados	56
Figura 29	Estrutura da plataforma	57
Figura 30	Ficheiro de configurações das bases de dados	59
Figura 31	Processo para a análise de um parâmetro de um nó	60
Figura 32	Interação entre o script de Alertas e a plataforma	62
Figura 33	Função <i>Sampling_Rate</i>	63
Figura 34	Atualização das credencias	64
Figura 35	Exemplo de um gráfico representativo do método do cotovelo [77]	67
Figura 36	Escolha de um algoritmo de <i>Machine Learning</i>	68
Figura 37	Nó <i>Smart Composter</i>	70
Figura 38	Nó <i>Energy Meter</i>	70

Figura 39	Microprocessador Fipy [78]	71
Figura 40	Configuração da <i>gateway</i>	71
Figura 41	Configurações do dispositivo	72
Figura 42	Configurações da aplicação	72
Figura 43	Configurações da integração com o InfluxDB	73
Figura 44	Configuração do Telegraf para o protocolo MQTT	74
Figura 45	Configuração do ficheiro <i>databases.ini</i>	75
Figura 46	<i>Login</i>	76
Figura 47	Home	76
Figura 48	Menu <i>Databases</i>	77
Figura 49	Medidas armazenadas na BD <i>Energy_Meter-1</i>	77
Figura 50	Tabela de estatísticas da potência	78
Figura 51	Gráfico comparativo do tempo que o nó <i>Energy_Meter-1</i> se encontrou em funcionamento	78
Figura 52	Média mensal da potência	79
Figura 53	Média diária da potência no último mês	80
Figura 54	Média diária da potência durante o tempo total	80
Figura 55	Tabela de estatísticas da concentração de metano	81
Figura 56	Gráfico comparativo do tempo que o nó <i>Smart_Composter</i> se encontrou em funcionamento	82
Figura 57	Média mensal da concentração de metano	82

Figura 58	Média diária da concentração de metano no último mês	83
Figura 59	Média diária da concentração de metano durante o tempo total	83
Figura 60	Menu para criar um Alerta	84
Figura 61	Menu para apagar um alerta	84
Figura 62	Menu para enviar um pedido de alteração da taxa de amostragem	85
Figura 63	Histórico dos diferentes pedidos	86
Figura 64	Formulário para atualizar as credencias	86
Figura 65	Menu de escolha da BD a aplicar os algoritmos de <i>Machine Learning</i>	87
Figura 66	Método do cotovelo aplicado aos dados do Energy_Meter	88
Figura 67	Diferentes perfis energéticos criados pelo algoritmo	88
Figura 68	Formulário para configurar a regressão	89
Figura 69	Resultado da regressão	89
Figura 70	Formulário para configurar a classificação	90
Figura 71	Resultado da classificação	90

Índice de Tabelas

Tabela 1	Cabeçalho fixo do pacote MQTT [22]	13
Tabela 2	Tipos de mensagem MQTT [23]	13
Tabela 3	Parâmetros da modulação para uma BW = 125kHz [27]	17
Tabela 4	Especificações do protocolo LoRaWAN [28]	21
Tabela 5	Estrutura do <i>payload</i> [34]	22
Tabela 6	ID e respetivo número de bytes de cada medida [34]	23
Tabela 7	Exemplo de um payload Cayenne LPP [34]	23
Tabela 8	Comparação entre <i>data</i> Tradicional e <i>big data</i> [41]	27
Tabela 9	Ligação entre os pinos do concentrador e do raspberry Pi [61]	38

Acrónimos

ABP	–	Activation by Personalization
ANN	–	Artificial Neural Network
AP	–	Access Point
API	–	Application Programming Interfaces
AppEUI	–	Application EUI
APPKey	–	Application Key
AppSKey	–	Application Session Key
ARIMA	–	Autoregressive Integrated Moving Average
BD	–	Base de Dados
BLE	–	Bluetooth Low Energy
CHIRP	–	Compressed High Intensity Radar Pulse
CIP	–	Common Industrial Protocol
CoAP	–	Constrained Application Protocol
CPS	–	Cyber-Physical Systems
CSS	–	Chirp Spread Spectrum
CSV	–	Comma-Separated Values
DevAddr	–	Device Address
DevEUI	–	Device Unique Identifier

- DevNonce – Device Number Used Once
- DHCP – Dynamic Host Configuration Protocol
- ETSI – European Telecommunications Standards Institute
- FSK – Frequency-shift keying
- GUI – Graphical User Interface
- I&D – Investigação e Desenvolvimento
- IACS – Automation and Control Systems
- IETF – Internet Engineering Task Force
- IIoT – Industrial Internet of Things
- IoT – Internet of Things
- IP – Internet Protocol
- IS4SE – Industrial Sensing for Smart Europe
- JoinEUI – Join Unique Identifier
- JSON – JavaScript Object Notation
- LPP – Low Power Payload
- LPWAN – Low-Power Wide-Area Network
- LSTM – Long Short Term Memory
- M2M – Machine-to-Machine
- MQTT – Message Queuing Telemetry Transport

MSC	–	Message Sequence chart
NwkSKey	–	Network Session Key
OPC	–	Open Platform Communications
OPC-UA	–	Open Platform Communications - Unified Architecture
OTAA	–	Over-The-Air Activation
PLC	–	Programmable Logic Controller
PME	–	Pequena e Média Empresa
QoS	–	Quality of Service
RNN	–	Recurrent Neural Network
SF	–	Spreading Factor
SPI	–	Serial Peripheral Interface
SQL	–	Structured Query Language
SSID	–	Service Set Identifier
SSL	–	Secure Socket Layer
SVM	–	Support Vector Machine
TSDB	–	Time Series Database
UDP	–	User Datagram Protocol
URL	–	Uniform Resource Locater

1. INTRODUÇÃO

O termo IoT foi introduzido no ano de 1999 para referir uma rede de objetos físicos como dispositivos diversos incorporando sensores, munidos com *software* e interligados por uma rede, com o objetivo de recolher, agir e trocar dados [1]. Outra definição dada pela *Internet Society*, define o IoT num sentido mais vasto como “a extensão da conectividade de rede e capacidade de computação para objetos, dispositivos, sensores e outros artefactos que normalmente não são considerados computadores”. Este conceito tem sido amplamente aplicado para conectar dispositivos em contexto de consumidor doméstico e nas mais variadas aplicações a nível industrial, sendo que as aplicações na vertente industrial têm vindo a crescer exponencialmente [2].

Na atualidade, o paradigma industrial tende cada vez mais para uma digitalização do mesmo, tendo como objetivo possibilitar a integração da tecnologia operacional com a tecnologia de informação.

O conceito da Indústria 4.0, que representa a quarta revolução industrial, chegou em 2011, apresentado pela Alemanha durante a feira de Hannover como uma iniciativa estratégica. Este conceito ganhou grande visibilidade a nível mundial e é universalmente adotado de forma a melhorar a eficiência da produção por meio de serviços inteligentes em fábricas inteligentes, também chamado de *smart manufacturing* [3].

O *Industrial Internet of Things* (IIoT) é então a base para este *smart manufacturing*, conectando os ativos industriais, incluindo máquinas e sistemas de controlo, com os sistemas de informação. É, portanto, uma extensão do conceito IoT em que é feito o uso desta tecnologia para fins de manufatura e dos mais diversos setores e áreas industriais.

1.1. CONTEXTUALIZAÇÃO

A CONTROLAR S.A. é uma Pequena e Média Empresa (PME) portuguesa com 25 anos de história dedicada principalmente à indústria automóvel, trabalhando para a maioria dos fabricantes e fornecedores de *Tier 1* e *Tier 2*, como Robert Bosch, Delphi, Visteon, Hanon, BorgWarner, Continental, Magneti Marelli, Preh, entre outros. A empresa está dividida em 4 linhas de negócio:

- **Sistemas de Teste:** abrangendo todas as áreas de testes eletrónicos, desde a validação de produtos até testes de fim de linha, incluindo programação no sistema, testes funcionais, inspeção ótica, confiabilidade e validação climática, depois aplicados em soluções de teste para unidades de infotainment, telemática, segurança, sistemas de condução autónomos, clusters, entre outros.
- **Sistemas de Automação:** dedicado a projetar e desenvolver todos os aspetos mecânicos e de automação dos sistemas de teste, mas também para montagem de produtos, automação de processos, robótica colaborativa e integração de sistemas de linhas completas com os mais altos níveis de precisão e fabricação.
- **Auto.ID:** de forma a melhorar o valor e qualidade dos sistemas produzidos, comercializa e distribui equipamentos relacionados a diferentes áreas de automação e identificação (computadores industriais, controladores, equipamentos de identificação automática para rastreabilidade, entre outros).

- **Aeroespacial e Defesa:** aproveitando todo o conhecimento de testes e automação aplicado ao setor aeroespacial e de defesa para desenvolver sistemas de validação e certificação para componentes eletrônicos, elétricos e eletromecânicos.

O departamento de Investigação e Desenvolvimento (I&D) da Controlar, tendo em conta as novas tendências tecnológicas, leva a cabo projetos disruptivos face ao negócio base da empresa, tentando criar produtos e diversificar a área de atuação da empresa. Um projeto que exemplifica o que foi anteriormente referido é o projeto *Industrial Sensing for Smart Europe* (IS4SE) [4].

O IS4SE é um projeto de I&D industrial à escala europeia em co-promoção com o Programa Interface [5], com o objetivo de criar uma infraestrutura capaz de integrar diferentes tipologias de sensores para a monitorização de linhas de produção de folhas de aço.

Esta infraestrutura irá ser responsável pela recolha, tratamento e análise dos dados, usando algoritmos de *machine learning*. Esta solução permitirá o aumento dos indicadores chave de desempenho, ao nível da produtividade e da segurança dos processos no sector da siderurgia.

Este projeto resulta então de uma parceria europeia liderada pela empresa turca Borçelik [6], onde a infraestrutura desenvolvida será instalada, e de um consórcio nacional constituído pela Controlar e pelo Centro de Nanotecnologia e Materiais Técnicos Funcionais e Inteligentes, Centi [7].

Das tarefas delineadas para cada um dos parceiros, a Controlar ficou responsável pelos seguintes pontos:

- Vigilância tecnológica das diferentes tecnologias que possam ser integradas na solução a desenvolver;
- Definição dos requisitos e design da infraestrutura que irá englobar todos os sensores;
- Desenvolvimento da plataforma capaz de anexar todos dados provenientes dos sensores e posteriormente, tratar esses dados e analisá-los.

- Desenvolvimento de algoritmos que permitam não só otimizar o consumo energético da siderurgia como melhorar a sua produtividade.

O trabalho documentado ao longo deste relatório, tem como objetivo endereçar algumas destas tarefas e permitir adquirir *know-how* e desenvolver o primeiro protótipo da rede que será mais tarde instalada na Borçelik.

1.2. OBJETIVOS

A partir do projeto IS4SE, foram delineados os objetivos que a seguir se apresentam.

O principal objetivo deste projeto é desenvolver um protótipo de uma rede IoT aplicável em ambiente industrial, utilizando primordialmente o protocolo Wi-Fi e o LoRaWAN. Paralelamente a este objetivo, pretende-se criar uma aplicação *web* capaz de tratar os dados recolhidos, oferecer estatísticas, assim como uma variedade de informação sobre a rede criada.

Tendo em conta os diversos aspetos que inicialmente se pretendia abordar, surgiu a necessidade de dividir o trabalho em etapas mais específicas. Estas são:

- Analisar o conceito de IIoT e as respetivas tecnologias por ele utilizadas;
- Avaliar as diferentes funcionalidades de cada tecnologia;
- Caracterizar o conceito de *big data analytics*;
- Desenvolver um protótipo de uma rede IoT;
- Implementar uma aplicação *web* capaz de analisar os dados da rede IoT dinamizada;
- Identificar um caso de estudo de forma a demonstrar as potencialidades da solução desenvolvida;
- Elaborar o presente relatório.

1.3. ORGANIZAÇÃO DO RELATÓRIO

O presente relatório está dividido em sete capítulos. O primeiro capítulo é dedicado à contextualização e descrição do problema, bem como das metas e objetivos inerentes ao desenvolvimento deste projeto.

O segundo e terceiro capítulo são dedicados à fundamentação teórica da temática em análise. Em primeiro lugar discute-se o conceito de IIoT e das tecnologias utilizadas para o desenvolvimento da rede criada. Segue-se a definição de *big data analytics*, assim como termos associados a este, tais como *big data* e *machine learning*.

No quarto capítulo é descrita a rede IoT criada, os aspetos mais relevantes da sua conceção e as tecnologias e protocolos de comunicação utilizados.

No quinto capítulo é apresentada a aplicação *web* desenvolvida, isto é, a pertinência e o objetivo da mesma na elaboração deste projeto. Este capítulo descreve ainda as funcionalidades da mesma, e são enumeradas as ferramentas utilizadas.

O sexto capítulo abre caminho para a apresentação dos resultados através de um caso de estudo em que é criada uma aplicação *web* ligada à rede IoT, de forma a ser possível visualizar a solução criada.

Por fim, o sétimo capítulo apresenta as conclusões finais do projeto desenvolvido, discutindo também as limitações encontradas no decorrer da sua realização e, conseqüentemente, são apresentadas possíveis melhorias futuras.

2. IIoT

Neste capítulo é apresentado um estado de arte das diversas tecnologias que compõem a pilha protocolar de uma rede IIoT, dando de seguida ênfase aos protocolos de comunicação utilizados na implementação da rede IoT.

2.1. PILHA PROTOCOLAR

O débito, número de dispositivos suportados, atrasos e outros aspetos de comunicação relevantes de uma rede IIoT dependem de um conjunto importante de fatores, como o uso de tecnologias com ou sem fios, as condições físicas em que a rede será implementada, podendo sofrer interferências eletromagnéticas, e a forma como os dispositivos serão alimentados, o que poderá suscitar a necessidade de recorrer ao uso de baterias [8].

Tal como em diversas outras situações, um objetivo fundamental é evitar sistemas isolados com base em soluções proprietárias e permitir a partilha de dados e interoperabilidade entre esses subsistemas fechados. A viabilidade das iniciativas IIoT requerem protocolos de comunicação capazes de oferecer suporte à agregação e disponibilização de informações eficientes [8].

Três camadas principais podem ser identificadas: rede, conectividade e informação (Figura 1).

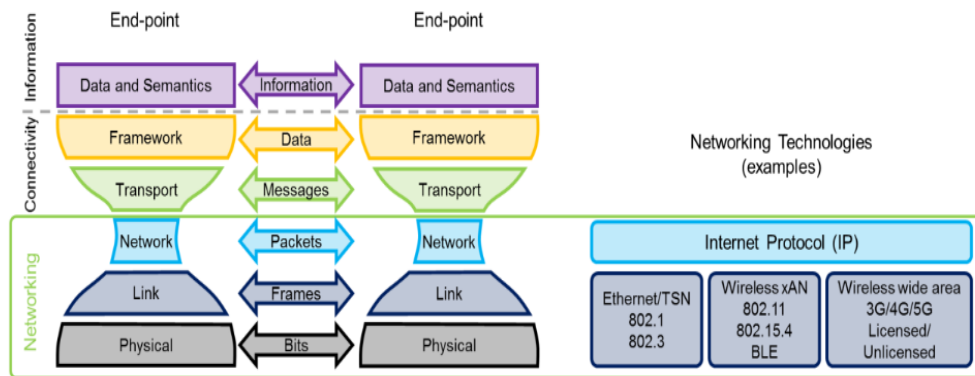


Figura 1 Pilha protocolar IIoT [8]

2.1.1. CAMADAS INFERIORES

A camada mais baixa da pilha protocolar é a camada física. Esta define as propriedades físicas e técnicas do meio de transmissão e regula a relação entre o *hardware* da rede e o meio de transmissão físico. Em cima encontra-se a camada de controlo de ligação lógica, responsável pela interação de vários componentes da rede de forma a garantir uma transmissão confiável e determinística entre os nós da rede.

No mercado já se encontram disponíveis diversas tecnologias desenhadas para o ambiente industrial que implementam estas funções. No entanto, todas elas apresentam certas limitações. Protocolos bem difundidos comercialmente, conhecidos como WirelessHART e ISA100.11a são baseadas no protocolo IEEE802.15.4, mas não foram desenhados para ligar um número elevado de dispositivos, como tipicamente as aplicações de IIoT requerem, sendo necessário para colmatar esta limitação a implementação de diversas redes independentes [9].

Contrariamente, as *Low-Power Wide-Area Networks* (LPWAN) são uma solução que tem vindo a ganhar terreno nos últimos anos, permitindo a comunicação ao longo de grandes distâncias, na ordem dos vários quilómetros com uma baixa potência de transmissão. Duas das tecnologias mais promissoras dentro destas redes é o LoRaWAN e o SigFox.

No entanto a tecnologia SigFox utiliza modulação de ultra banda estreita, sendo mais direcionada para aplicações relacionadas com *smart cities* uma vez que permitem um

número reduzido de mensagens por dia. Esta característica faz com que não seja apropriada para a maior parte das aplicações industriais que têm de ser determinísticas. Para além desta desvantagem estas tecnologias ao aumentarem a taxa de amostragem dentro da capacidade da rede, aumenta a probabilidade de colisões de pacotes e consequentemente perda de informação.

O LoRaWAN utiliza a modulação LoRa, que contrariamente às redes SigFox, permite uma taxa de amostragem maior por parte dos dispositivos [10].

O espectro utilizado pelas LPWAN levanta algumas condicionantes nomeadamente por ser um espectro não licenciado, não garantindo a disponibilidade do serviço e levantando questões associadas aos regulamentos relacionados com a ocupação do espectro. Além deste aspeto negativo, é também de realçar que estas tecnologias funcionam maioritariamente no sentido de *uplink*, podendo o LoRaWAN funcionar na vertente bidirecional, acrescentando sobrecarga à transmissão [10].

O *Bluetooth Low Energy* (BLE), foi inicialmente pensado para o ramo da domótica, mas é outra possível alternativa para o IIoT, visto que, oferece um consumo de energia muito reduzido, tendo como principal ponto negativo a limitação do alcance da transmissão, uma vez que suporta apenas a topologia em estrela e um número reduzido de dispositivos.

Tendo em vista estas limitações recentemente foi lançado um novo protocolo o BLE *mesh*. O principal desafio desta tecnologia é encontrado quando se pretende visar uma comunicação em tempo real, consequência do procedimento de estabelecimento da conexão que introduz um atraso considerável. Este senão pode ser contornado utilizando o BLE *mesh* em aplicações que possuam uma taxa de amostragem baixa [11].

Tendo em conta algumas destas desvantagens mencionadas, existem um conjunto de tecnologias, que não são tão específicas ou enquadradas ao meio industrial, mas que as suas características fazem considerar o seu uso nesta vertente.

A tecnologia 5G é vista como uma solução possivelmente viável para a implementação do paradigma de IIoT, apesar de ainda se encontrar numa fase bastante preliminar de

disponibilização. Apresenta também como ponto negativo o custo da solução a ser aplicada na camada física [12].

Ao nível das redes LPWAN existe ainda outra alternativa denominada de NB-IoT, que alia os benefícios de redes LPWAN com o facto de usarem infraestruturas móveis que não contêm restrições de uso. Outras duas tecnologias com estas características adequadas são o GSM-IoT e o LTE, sendo que, sem infraestrutura móvel estes três padrões não funcionam [13].

2.1.2. CAMADA INTERMÉDIA

A camada de rede implementa os protocolos IPv4 e IPv6 definidos pela *Internet Engineering Task Force* (IETF). Além destes, existem algumas iniciativas do *European Telecommunications Standards Institute* (ETSI) que têm como objetivo promover a inovação na camada de rede.

As tecnologias consideradas tradicionais da camada de controlo de ligação lógica relativamente ao meio industrial, exploram barramentos de campo, cada uma tendo o seu próprio ecossistema, que se traduz numa interoperabilidade fraca.

No entanto, tecnologias mais recentes que adotam nativamente os protocolos *ethernet* e *Internet Protocol* (IP), mitigam a questão da interoperabilidade. Devido à sua compatibilidade total com o protocolo IP e incorporação do *Common Industrial Protocol* (CIP), o protocolo EtherNet/IP torna-se adequado para o uso em redes IIoT.

2.1.3. CAMADAS SUPERIORES

O *Industrial Internet Consortium* propõe uma divisão desta camada em dois níveis: o nível mais baixo é ocupado pela camada de transporte, que é responsável pela troca de mensagens de comprimento variável entre os diferentes dispositivos que compõem a rede.

O nível superior corresponde à camada de estrutura que gere a transferência de dados com um maior nível de abstração [8].

Uma solução vastamente utilizada neste contexto para a troca de mensagens na camada de transporte é a utilização de protocolos de mensagem. Estes protocolos suportam a metodologia de *publish/subscribe* em que um cliente publica a informação e outro subscreve para a receber. Este facto permite que este tipo de soluções tenham uma grande escalabilidade. Das diversas tecnologias disponíveis no mercado que utilizam este protocolo de mensagens o *Message Queuing Telemetry Transport* (MQTT) normalizado pela OASIS *alliance* é a opção mais conhecida [14].

Uma abordagem alternativa é a implementação do modelo *request/response*, onde é permitida troca de informação síncrona e assíncrona entre os intervenientes. Na modalidade síncrona, o cliente que está a fazer o pedido tem de aguardar até receber a informação que requisitou, para efetuar um novo pedido. Enquanto que na modalidade assíncrona a resposta é recebida pelo cliente que efetuou o pedido num tempo desconhecido após ter efetuado o mesmo. Um dos principais exemplos no que toca a esta tecnologia de *request/response* é *Constrained Application Protocol* (CoAP) [15].

Os protocolos empregues na camada de estrutura oferecem a capacidade de descobrir e identificar objetos de dados e de compreender o significado dos dados transportados, de forma a otimizar a troca de dados com a camada aplicacional.

O *Open Platform Communications - Unified Architecture* (OPC-UA) é gerida pela fundação OPC é um exemplo de um protocolo empregue nesta camada. O *Open Platform Communications* (OPC) é o padrão de interoperabilidade responsável pela troca, segura e confiável, de dados no espaço de automação industrial. Esta solução é independente da plataforma e garante o fluxo contínuo de informações entre dispositivos de vários meios [16]. A sua mais recente iteração, isto é, o OPC-UA foi criada com o propósito de colmatar os desafios, particularmente como a segurança e modelagem dos dados, com a introdução de arquiteturas orientadas a serviços em sistemas de ambiente industrial [17].

2.2. TECNOLOGIAS RELEVANTES PARA IMPLEMENTAÇÃO

Após ter sido feito o levantamento das mais variadas tecnologias que podem ser utilizadas nos diferentes níveis da pilha protocolar de uma rede IIoT, neste presente subcapítulo, será dado maior relevo às tecnologias utilizadas na implantação da rede IoT desenvolvida.

2.2.1. MQTT

O MQTT é um protocolo de comunicação Machine-to-Machine (M2M) criado pela IBM e pela Euritech no final da década de 90, desenhado de forma a ser um protocolo leve de transporte de mensagens através do método *publish/subscribe*. Este é baseado na publicação de mensagens e subscrição de tópicos [18].

Nesta solução, múltiplos clientes ligam-se a um *broker* utilizando o protocolo TCP/IP. Após efetuada a ligação ao *broker*, os clientes podem publicar mensagens em tópicos e/ou subscrever tópicos, sem que seja necessário qualquer tipo de configuração.

O *broker* opera como um servidor, sendo uma interface comum que interliga todo o sistema (Figura 2). Ou seja, é da responsabilidade do broker receber todas as mensagens, filtrá-las, determinar que clientes subscreveram determinados tópicos e reencaminhar as mensagens correspondentes [19].

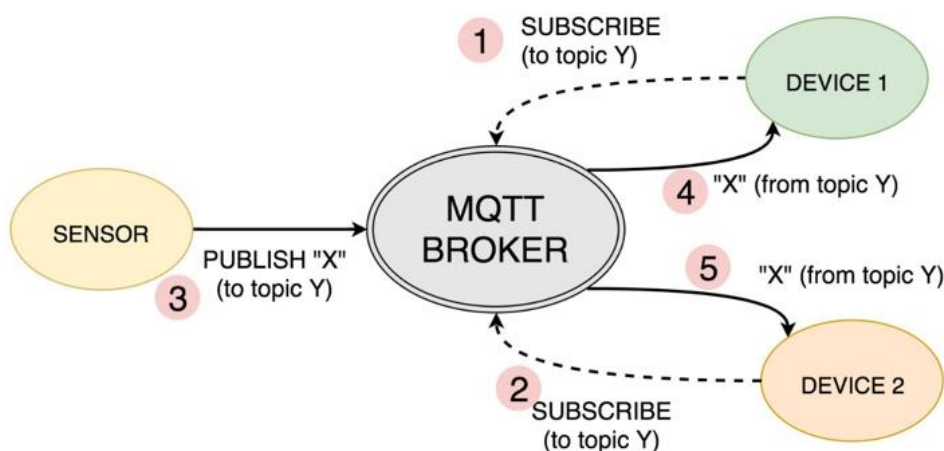


Figura 2 Exemplo de uma arquitetura MQTT [20]

Os tópicos seguem uma hierarquia usando o caracter “/” como separador. Esta característica é essencial, uma vez que possibilita criar uma organização entre todos os tópicos. Ou seja, anexa diversas mensagens do mesmo destinatário ou mensagens semelhantes que estão a ser enviadas como um sistema de arquivos [21].

Por exemplo, vários computadores podem publicar as respetivas temperaturas do disco rígido no seguinte tópico substituindo, conforme apropriado, o respetivo nome do computador e do disco rígido:

sensores/nome_computador /temperatura/Nome_DiscoRigido

A subscrição pode ser efetuada a um tópico explícito, caso em que o cliente apenas irá receber mensagens desse tópico específico, ou pode fazer uso de dois caracteres especiais “#” e “+”.

O caracter “+” permite que uma hierarquia do tópico não seja considerada, o que significa que subescrevendo o tópico em baixo, irá receber a temperatura de todos os discos rígidos de todos os computadores.

sensores/+ /temperatura/+

Por sua vez, o uso do caracter “#” permite englobar todos os níveis restantes da hierarquia. Ou seja, subescrevendo o tópico em baixo, o cliente irá receber todo o tipo de informação proveniente dos sensores, uma vez que a primeira hierarquia é `sensores`:

sensores/#

Relativamente à segurança, pode obrigar o uso de credenciais de acesso ao broker ou permitir a encriptação da própria rede usufruindo de certificados *Secure Socket Layer* (SSL).

A mensagem ou pacote MQTT possui um cabeçalho fixo constituído por dois bytes (Tabela 1):

Byte 1 - Contém o tipo de mensagem (pedido de conexão do cliente, confirmação de subscrição do broker) (Tabela 2), uma *flag* de duplicação e informação quanto ao *Quality of Service* (QoS) escolhido.

Byte 2 – Contém informação referente ao comprimento restante da mensagem, incluindo o *payload* e qualquer dado adicional.

Tabela 1 Cabeçalho fixo do pacote MQTT [22]

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type				Flags specific to each MQTT Control Packet type			
byte 2...	Remaining Length							

Tabela 2 Tipos de mensagem MQTT [23]

Message type	Value	Description	Fixed Header
Reserved	0	Reserved	Present
CONNECT	1	Client connect request to server or broker	Present
CONNACK	2	Connect request acknowledgment	Present
PUBLISH	3	Publish message	Present
PUBACK	4	Publish acknowledgment	Present
PUBREC	5	Publish receive	Present
PUBREL	6	Publish release	Present
PUBCOMP	7	Publish complete	Present
SUBSCRIBE	8	Client subscribe request	Present
SUBACK	9	Subscribe request acknowledgment	Present
UNSUBSCRIBE	10	Unsubscribe request	Present
UNSUBACK	11	Unsubscribe acknowledgment	Present
PINGREQ	12	PING request	Present
PINGRESP	13	PING response	Present
DISCONNECT	14	Client is disconnecting	Present
Reserved	15	Reserved	Present

Os três níveis de QoS definidos neste protocolo indicam o quanto o *broker* e o cliente garantem que a mensagem seja recebida. Importa realçar que quantas mais garantias de entrega o *broker* der, maior será a latência [21]:

- QoS 0 – O cliente/*broker* envia a mensagem uma vez sem confirmação de entrega;
- QoS 1 - O cliente/*broker* envia repetidamente a mensagem até receber a confirmação de entrega;
- QoS 2 - O cliente/*broker* segue um conjunto de passos durante o envio da mensagem de forma a garantir a confirmação de entrega.

O método de funcionamento deste protocolo confere-lhe uma elevada escalabilidade e versatilidade, afirmação comprovada pelo exemplo na referência [24], em que dois

laboratórios são equipados com sensores de temperatura. Cada um destes sensores irá publicar em tópicos diferentes por exemplo `sensor/temperatura/sala1` e `sensor/temperatura/sala2`. Fazendo uso da hierarquia de tópicos explicada em cima, esta configuração permite obter os dados de ambos os sensores subscrevendo o tópico `/sensor/temperatura/#`. Esta característica permite adicionar novos sensores sem ser necessário reprogramar o dispositivo que se encontra a agregar os dados.

Aufere ainda independência em relação ao sensor físico, uma vez que, em caso de avaria de um sensor este pode ser substituído por outro que publique a mesma informação no mesmo tópico.

Além disto, outras características, como o facto de o cabeçalho ser muito reduzido, o que otimiza o uso da largura de banda, e o facto de os clientes/bibliotecas MQTT serem pequenas requerendo recursos reduzidos por parte dos microprocessadores, fazem com que o IoT seja uma das principais aplicações do MQTT.

2.2.2. LORAWAN

LoRa é uma modulação de espelhamento espectral proprietária da empresa Semtech, que deriva da modulação *Chirp Spread Spectrum* (CSS). Esta modulação permite ao LoRa ser resistente a fenómenos de desvanecimento, *multipath* e Doppler.

Como foi anteriormente mencionado, sendo uma rede LPWAN permite comunicação a longa distância por dispositivos de baixo consumo energético, característica que para ser atingida à partida se traduz na necessidade de uma maior potência de transmissão para fazer face à atenuação que aumenta com o aumento da distância. Porém, sendo estes dispositivos de baixo consumo não lhes permite despende energia requerida para aumentar a potência. A solução passa por facultar aos recetores sensibilidades maiores. Fazendo uso do teorema de Shannon-Hartley, atendendo à seguinte expressão (1) [25]:

$$\frac{C}{B} = 1.433 * \frac{S}{N}. \quad (1)$$

Sendo que C representa a capacidade do canal, S a potência do sinal recebido, B a largura de banda e N a potência do Ruído.

É então possível concluir que, por exemplo, uma diminuição de metade da capacidade de sinal irá duplicar a energia por *bit* o que perfaz num aumento de probabilidade de receção dos pacotes. Isto significa que pelo teorema de Shannon-Hartley é possível aumentar a sensibilidade do recetor em detrimento da capacidade do canal. Tendo em conta a fórmula apresentada em cima, assumindo que a relação $\frac{S}{N} \ll 1$ pode ter-se a seguinte relação:

$$\frac{C}{B} \approx \frac{S}{N} \quad (2)$$

Esta relação permite reforçar que uma largura de banda maior acarreta que a relação sinal ruído seja menor para um determinado ritmo binário.

2.2.2.1. CHIRP SPREAD SPECTRUM

Esta modulação de espalhamento espectral, utiliza sinais sinusoidais denominados *Compressed High Intensity Radar Pulse* (CHIRP) que transportam a informação que será enviada. Estes possuem amplitude constante e percorrem toda a largura de banda, fazendo variar a sua frequência de uma forma linear durante intervalos de tempo definidos. Se existe uma mudança da frequência mínima para a frequência máxima é denominado *up-chirp* a mudança inversa é uma *down-chirp*. Sendo que a largura de banda BW é a diferença entre a frequência máxima e a mínima (Figura 3).

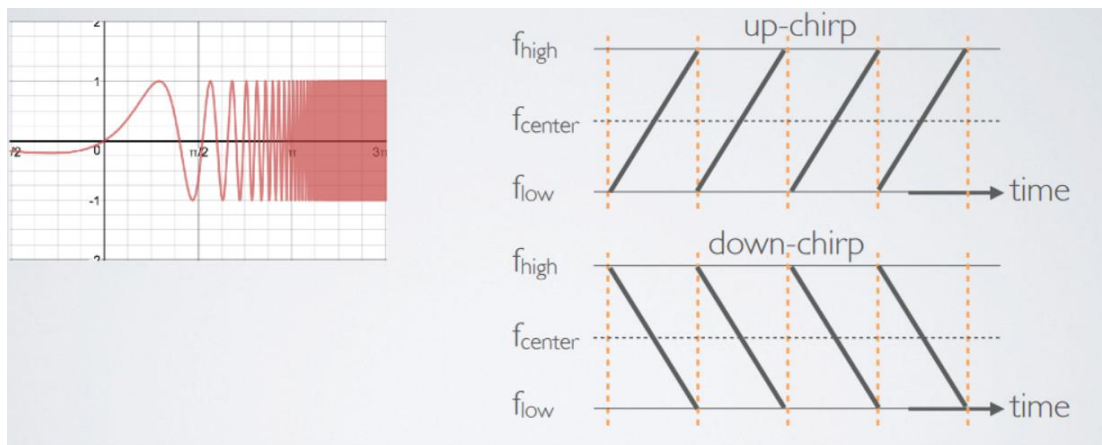


Figura 3 (Esquerda) Sinal na frequência (direita) Sinal no tempo [26]

Cada um dos *chirps* ou símbolos são compostos por um respetivo número de *chips*, que correspondem à quantidade de oscilações efetuadas durante o tempo de um símbolo. Assim sendo, o número de oscilações por segundo é denominado por *chip Rate*. O *Spreading Factor (SF)* representa o número de bits de informação que são codificados num único símbolo e assume valores entre 7 e 12. O tempo de símbolo é então dado pela seguinte equação:

$$T_s = \frac{2^{SF}}{BW} [s] \quad (3)$$

Logo, a taxa de símbolo é dada pela seguinte expressão:

$$R_s = \frac{1}{T_s} = \frac{BW}{2^{SF}} [\text{símbolo/s}] \quad (4)$$

Uma vez que em cada símbolo contém SF bits de informação codificados, o ritmo binário de uma transmissão LoRaWAN é dada pela expressão:

$$R_b = R_s * SF [\text{bits/s}] \quad (5)$$

Demonstradas estas expressões é possível concluir que a escolha do SF depende do tipo de aplicação, uma vez que faz variar o tempo de símbolo. Isto é, verifica-se que para uma largura de banda fixa, um incremento do SF, traduz-se no dobro do tempo por ciclo, que leva a uma diminuição do ritmo binário. Por outro lado, esta diminuição da capacidade do canal, tendo em conta a expressão (2), converte-se numa maior sensibilidade por parte do recetor (Tabela 3). De referir ainda que a modulação LoRa permite utilizar também uma largura de banda de 250kHz e de 500kHz

Tabela 3 Parâmetros da modulação para uma BW = 125kHz [27]

SF	Limite $\frac{S}{N}$	<i>Time-on-air</i> (pacote de 10 bytes) [ms]	Ritmo binário [bits/s]
7	-7.5	56	5469
8	-10	103	3125
9	-12.5	205	1758
10	-15	371	977
11	-17.5	741	537
12	-20	1483	293

2.2.2.2. STACK LORAWAN

Nas redes que utilizam a arquitetura de rede *mesh*, os nós encaminham pacotes entre eles de modo a aumentar o alcance da comunicação e o tamanho da rede. No entanto, esta abordagem resulta tipicamente em maiores consumos de energia. Para evitar esta situação, o LoRaWAN, protocolo de rede que usa na sua camada física o LoRa, utiliza a topologia *Long Range Star* permitindo manter o alcance da rede e preservar o ciclo da bateria dos dispositivos [28].

Nesta rede, os nós não são associados a uma *gateway* específica, pois normalmente a informação é recebida por várias *gateways*, que posteriormente reencaminham para o servidor. Com este processo, também é possível passar a gestão da complexidade da rede para o lado do servidor, filtrar os pacotes redundantes e executar análises de segurança [28]. De referir ainda que as *gateways* estão conectadas ao servidor da rede usando o protocolo IP, atuando como conversores de pacotes RF em pacotes IP e vice versa.

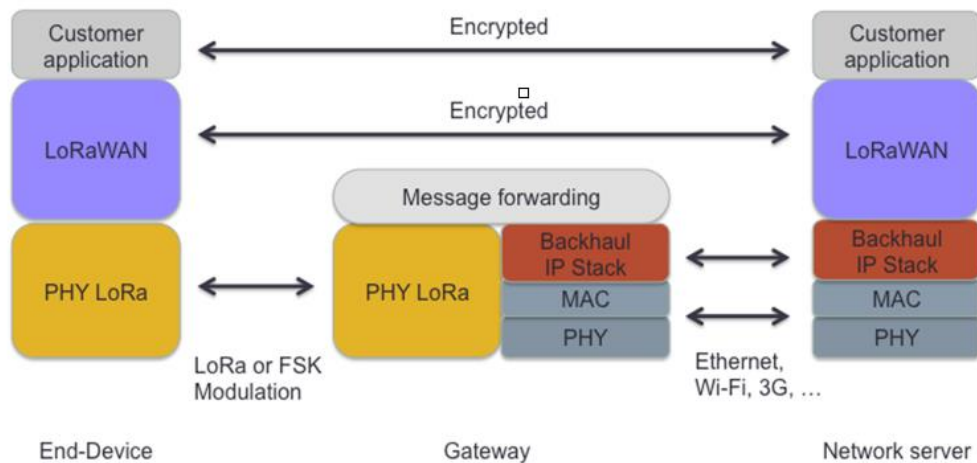


Figura 4 Arquitetura de rede LoRaWAN [29]

Na Figura 4 é possível observar um esquema da arquitetura de uma rede LoRaWAN, destacando-se os seguintes pontos:

Customer Application – Composta pelos dados dos diferentes dispositivos LoRa.

LoRaWAN – Responsável pela gestão da rede, tendo em conta o tipo de classe utilizada pelo dispositivo.

PHY LoRa – A camada física contempla o tipo de modulação utilizada que na sua grande maioria é a modulação LoRa, mas também possibilita a utilização da modulação *Frequency-shift keying* (FSK). Esta camada contempla ainda as especificações da banda de frequências de cada região que têm de ser respeitadas.

Gateway – É o elo entre o dispositivo LoRa e a rede, traduzindo as mensagens da camada PHY LoRa para mensagens do protocolo IP.

2.2.2.3. TIPO DE DISPOSITIVOS

Os dispositivos finais, como sensores e atuadores, servem um diferente número de aplicações cada uma com diferentes requisitos. De forma a otimizar uma variedade de perfis de utilização, o LoRaWAN possui diferentes classes de dispositivos, que equilibram consoante a classe escolhida a latência da comunicação *downlink* com a vida útil da bateria

do dispositivo [28]. Existem então três tipos de classificação de dispositivos numa rede LoRaWAN [30]:

- **Classe A** – Suporta a comunicação bidirecional entre o dispositivo e a *gateway*. As mensagens de *uplink* podem ser enviadas a qualquer altura. No entanto, assim que são enviadas, o dispositivo abre uma janela de tempo para receber o *downlink*. Caso isto não aconteça só terá hipótese na próxima mensagem de *uplink*.
- **Classe B** – É uma extensão da classe A: agenda intervalos de tempo para que o dispositivo possa receber o *downlink*.
- **Classe C** – Mantém a janela de receção de uma mensagem de *downlink* aberta sempre que o dispositivo não esteja a transmitir dados, diminuindo assim a baixa latência da comunicação. No entanto, em comparação com o dispositivo de classe A, aumenta drasticamente o consumo energético.

2.2.2.4. MÉTODOS DE ATIVAÇÃO DOS SENSORES

O protocolo LoRaWAN permite dois tipos de ativação o *Over-The-Air Activation* (OTAA) e o *Activation by Personalization* (ABP).

O método OTAA é o mais seguro dos dois métodos e requer a concretização de duas etapas. Previamente a iniciar a ativação, o dispositivo tem que ter configurado o identificador do dispositivo - *Device unique Identifier* (DevEUI) , o identificador de registo do dispositivo na rede – *Join Unique Identifier* (JoinEUI), e duas chaves para encriptar a comunicação: a *Application EUI* (AppEUI), que identifica a aplicação final e a *Application Key* (APPKey) que é usada para gerar chaves de sessão. É ainda necessária a configuração do campo - *Device Number used Once* (DevNonce). Este último é um valor incrementado sempre que é feito um *Join Request*, isto é, um pedido de acesso à rede, de modo a evitar ataques pelo envio de um pacote replicado [31].

Assim que o pedido de *Join Request* é recebido por uma *gateway* é reencaminhado para o servidor da sua rede, que ao receber o pedido consulta o AppEUI para validar o pedido. Se a permissão for concebida, o servidor responde com uma mensagem de *Join Accept* que

contem o NetID, identificador único da rede o *Device Address* (DevAddr), identificador do dispositivo na rede, DevNonce e outras configurações do sistema [32].

O dispositivo guarda as configurações da rede, assim como o devaddr e o NetID e fazendo uso do DevNonce gera as chaves de sessão necessárias para pertencer à rede *Application Session Key* (AppSKey) e *Network Session Key* (NwKSKey). O facto de as chaves de sessão serem apenas geradas quando necessárias, faz com não possam ser comprometidas antes da ativação, tornando este método seguro [32].

O método ABP contrariamente ao OTAA não necessita de recorrer ao procedimento de Join Request e Join Accept, uma vez que a AppSKey, o NwKSKey e o DevAddr são previamente configurados, permitindo uma ligação imediata do dispositivo. Este método executa um procedimento mais simples pois exige uma menor troca de chaves pela plataforma. É, também, menos seguro porque no caso das chaves serem descobertas comprometem a integridade do sistema [32].

2.2.2.5. PARÂMETROS REGIONAIS

O LoRaWAN opera num espectro de rádio não licenciado, o que significa que as bandas de frequências alocadas para estas tecnologias podem ser utilizadas sem a necessidade de pagar direitos de transmissão, semelhante ao Wi-Fi que usa as bandas de 2.4 GHz e a de 5GHz.

Como já foi mencionado o LoRaWAN utiliza frequências de rádio baixas com um alcance maior. Esta característica acarreta um maior número de restrições, que por norma são específicas de cada país

As diferentes restrições dos países representam um desafio para a tecnologia, que tem o intuito de ser o mais uniforme possível em todas as diferentes regiões do mundo. Como resultado, o LoRaWAN é especificado para um número de bandas de frequência para essas regiões. Bandas essas que são semelhantes o suficiente para suportar um protocolo agnóstico da região em que é implementado (Tabela 4) [33].

Tabela 4 Especificações do protocolo LoRaWAN [28]

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 + 8	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kpbs				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

Para além dos parâmetros mencionados na tabela em cima, outro parâmetro importante quanto à utilização desta tecnologia, que também é regulado em função da região, é o *duty cycle*. Este corresponde à fração de tempo que uma banda de frequência pode estar ocupada. No caso específico da Europa, este parâmetro é regulado pela norma EN300.220 do ETSI, definindo as seguintes sub-bandas com os respetivos *duty cycle*:

- g (863.0 – 868.0 MHz): 1%
- g1 (868.0 – 868.6 MHz): 1%
- g2 (868.7 – 869.2 MHz): 0.1%
- g3 (869.4 – 869.65 MHz): 10%
- g4 (869.7 – 870.0 MHz): 1%

2.2.2.6. TRAMA CAYENNE LPP

O Cayenne *Low Power Payload* (LPP) é um formato de *payload* criado pela empresa mydevices, que providencia uma forma fácil e conveniente de troca de dados em redes LPWAN como o LoRaWAN.

Este protocolo é compatível com a restrição do tamanho do payload, podendo reduzi-lo para 11 bytes. Este permite que um dispositivo/nó envie várias medidas em simultâneo. Para tal, a medida de um sensor é antecedida por dois bytes (Tabela 5). O primeiro `channel` reconhece o sensor que está a enviar e o segundo identifica o `Data` que, por sua vez, identifica o tipo de medida. Esta característica confere-lhe grande escalabilidade que outros formatos mais conhecidos como *JavaScript Object Notation* (JSON) ou o *Comma-Separated Values* (CSV) não oferecem [34].

Tabela 5 Estrutura do *payload* [34]

1 Byte	1 Byte	N Bytes	1 Byte	1 Byte	M Bytes	...
Data1 Ch.	Data1 Type	Data1	Data2 Ch.	Data2 Type	Data2	...

Os principais tipos de medidas criadas para este protocolo, bem como o seu ID e tamanho, encontram-se resumidas na (Tabela 6). De referir que a atribuição de ID a estes diferentes tipos de medidas está em conformidade com a diretriz IPSO *Alliance Smart Objects Guidelines*, diretriz que especifica os ID atribuídos às diferentes medidas provenientes de *smart objects*, como atuadores e sensores [35].

Tabela 6 ID e respetivo número de bytes de cada medida [34]

Type	IPSO	LPP	Hex	Data Size	Data Resolution per bit
Digital Input	3200	0	0	1	1
Digital Output	3201	1	1	1	1
Analog Input	3202	2	2	2	0.01 Signed
Analog Output	3203	3	3	2	0.01 Signed
Illuminance Sensor	3301	101	65	2	1 Lux Unsigned MSB
Presence Sensor	3302	102	66	1	1
Temperature Sensor	3303	103	67	2	0.1 °C Signed MSB
Humidity Sensor	3304	104	68	1	0.5 % Unsigned
Accelerometer	3313	113	71	6	0.001 G Signed MSB per axis
Barometer	3315	115	73	2	0.1 hPa Unsigned MSB
Gyrometer	3334	134	86	6	0.01 °/s Signed MSB per axis
GPS Location	3336	136	88	9	Latitude : 0.0001 ° Signed MSB
					Longitude : 0.0001 ° Signed MSB
					Altitude : 0.01 meter Signed MSB

Tendo em conta as duas tabelas anteriores, um dispositivo que contenha dois sensores de temperatura irá ter um *payload* semelhante ao demonstrado na (Tabela 7).

Tabela 7 Exemplo de um payload Cayenne LPP [34]

Payload (Hex)	03 67 01 10 05 67 00 FF	
Data Channel	Type	Value
03 ⇒ 3	67 ⇒ Temperature	0110 = 272 ⇒ 27.2°C
05 ⇒ 5	67 ⇒ Temperature	00FF = 255 ⇒ 25.5°C

3. BIG DATA ANALYTICS

Neste capítulo é apresentado o conceito de *big data analytics*, bem como outros similares ou integrantes como *big data* e *machine learning* que serão utilizados no tratamento dos dados recolhidos na rede experimental criada e que incluem parâmetros ambientais e consumo energético.

3.1. DEFINIÇÃO DE *BIG DATA*

O paradigma emergente do *big data*, transformou profundamente a sociedade moderna, consequência do seu amplo impacto, o que continuará a atrair cada vez mais a atenção dos especialistas como também do público em geral [36]. Um exemplo disto é um relatório do

IDC que prevê que, entre 2005 e 2020, o volume global de dados irá crescer segundo o fator de 300, de 130 *exabytes* a 40000 *exabytes*, o que representa um crescimento duplo a cada dois anos [37].

O termo *big data*, segundo a IBM, é aplicado a conjuntos de dados cujo tipo e tamanho excede as capacidades das bases de dados tradicionais para capturar, gerir e processar os dados com baixas latências [38].

A *TechAmerica Foundation* define o conceito de *Big Data*, como um termo que descreve grandes volumes de dados complexos e variáveis transferidos a alta velocidade que requerem técnicas avançadas e tecnologias para permitir a captura, armazenamento, distribuição, gestão e análise das informações [39].

Tendo por base estas duas definições, é então possível aferir que a velocidade, a variedade e o volume dos dados são as três características essenciais que definem um conjunto de dados como *big data* (Figura 5).

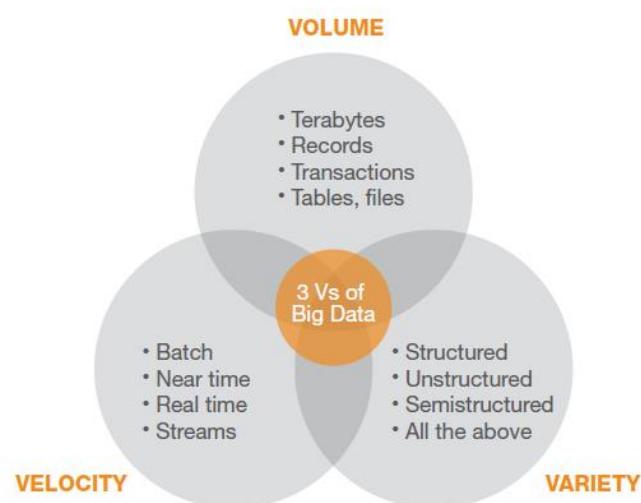


Figura 5 Os três Vs do *big data* [40]

A Tabela 8, tendo como base as características basilares do *big data*, apresenta uma comparação entre a informação tradicional e este conceito, explicitando ainda alguns desafios quando se utiliza este tipo de informação.

Tabela 8 Comparação entre *data* Tradicional e *big data* [41]

<i>Data</i> Tradicional	<i>Big Data</i>
Os dados recolhidos são estruturados	Os dados não são estruturados ou semiestruturados
O tamanho dos dados é pequeno	Tamanho consideravelmente maior que da <i>Data</i> tradicional
Os dados são centralizados	Os dados são descentralizados
Dados fáceis de serem manipuláveis	Dados difíceis de serem manipuláveis
Sistema normal é capaz de processar a informação	É necessário Sistema com grande poder de processamento
Ferramentas tradicionais das BD são suficientes	São necessárias ferramentas específicas
Funções normais são o suficiente para manipular os dados	É necessário um tipo específico para manipular as funções

3.2. DEFINIÇÃO DE BIG DATA ANALYTICS

Big data é um conceito que por si só não acrescenta valor a nenhuma indústria ou negócio. O seu potencial é desbloqueado quando a informação é utilizada para apoiar a tomada de decisões. Para tal, as organizações necessitam de processos eficientes para transformar grandes volumes de dados diversificados e rápidos em informações relevantes.

Esta transformação pode ser dividida em 5 estádios diferentes (Figura 6). Estes estádios formam dois subgrupos principais: a gestão dos dados e a análise dos mesmos.

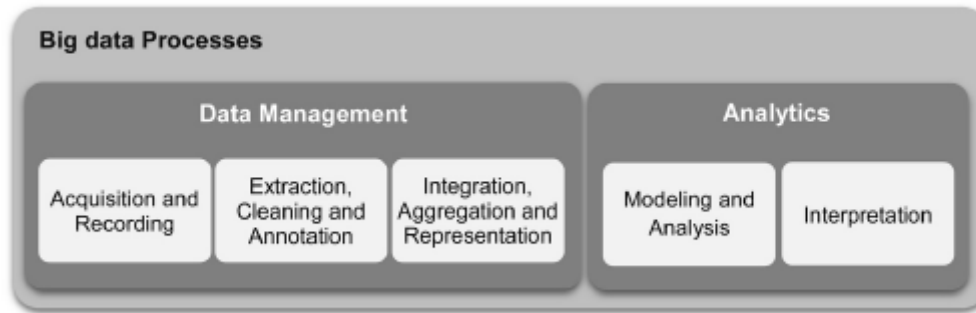


Figura 6 Processo para extrair informação relevante da *big data* [42]

A gestão dos dados envolve técnicas e tecnologias de suporte para adquiri-los e prepará-los para análise. Por outro lado, a análise engloba as técnicas utilizadas para analisar e adquirir uma nova percepção dos dados recolhidos. Por conseguinte, o termo de *big data analytics* pode ser definido como um subprocesso na extração de nova informação a partir da *big data*.

3.3. MACHINE LEARNING

Machine learning é um conceito que pretende fazer um paralelo entre a capacidade humana em aprender com a experiência passada desenvolvendo sistemas computacionais capazes de otimizar um dado processo de tomada de decisão, usando dados recolhidos no passado.

Usando como exemplo um modelo que contém um certo número de parâmetros, a aprendizagem é a execução de um programa de computador para otimizar esses mesmos parâmetros usando um volume de dados como exemplo. O modelo pode ser preditivo, fazendo previsões sobre o futuro, ou descritivo para obter conhecimento dos dados recolhidos, ou pode ser ambos.

A ciência de computação desempenha um papel duplo na criação de soluções: inicialmente, na fase de treino dos modelos, são necessários algoritmos eficientes para conseguir processar a enorme quantidade de dados e, posteriormente à aprendizagem do modelo, a representação e solução algorítmica também tem de ser eficiente de forma a ser possível retirar conclusões a partir dos resultados obtidos [43].

O conceito de *Machine Learning* não é recente, tendo, no entanto, vindo a ganhar cada vez mais importância devido às novas tecnologias de computação. Estas possibilitam a aplicação destes algoritmos à *big data*, com tempos de processamento progressivamente inferiores. Inicialmente, a sua base era o reconhecimento de padrões e o pressuposto que os computadores podem aprender sem requerer que sejam programados, posteriormente, este pressuposto foi transmutado para realizar tarefas específicas [44].

Pode afirmar-se que o conceito de *machine learning* apresenta um diverso leque de aplicações que vão desde a área da saúde, ao fornecer métodos e ferramentas que ajudam no diagnóstico e prognóstico numa variedade de domínios médicos, como por exemplo nas doenças oncológicas [45] e na radiologia [46], até ao mercado financeiro onde é usado para prever o preço de ações [47].

Para além destas aplicações mais específicas, apresenta ainda inúmeras aplicações usadas no quotidiano, como nas recomendações de produtos em websites de vendas, filtragem de *spam* no e-mail ou nos assistentes virtuais como a Alexa da Amazon.

Existem algumas variações quanto à definição dos diferentes tipos de algoritmos de *machine learning* (Figura 7), sendo que normalmente estes podem ser divididos em categorias de acordo com a sua finalidade:

- *Supervised Learning* – Algoritmos que envolvem supervisão direta da operação, visto que necessitam que os dados estejam anotados e que seja definido os limites sobre os quais o algoritmo vai operar.
- *Unsupervised Learning* – Não envolve controlo direto do algoritmo. Se, no caso anterior, os resultados são conhecidos e precisa de por exemplo os classificar, nesta situação os resultados desejados são desconhecidos por isso ainda não se encontram definidos.
- *Semi-supervised Learning* – Este algoritmo recai entre os dois anteriores é referente à situação em que nem todos os dados estão anotados, sendo então explorada a ideia que, apesar de existirem dados desconhecidos, estes contêm informações valiosas sobre o grupo de dados

- *Reinforcement Learning* – Este tipo de *machine learning* permite que o algoritmo determine automaticamente o comportamento ideal dentro de um contexto específico, a fim de maximizar o seu desempenho.
- *Deep learning* – Estes algoritmos são baseados em redes neurais, só que em vez de organizar os dados para serem executados através de equações predefinidas, o *deep learning* configura parâmetros básicos sobre os dados de forma que seja aprendido automaticamente, através do reconhecimento de padrões em várias camadas de processamento.

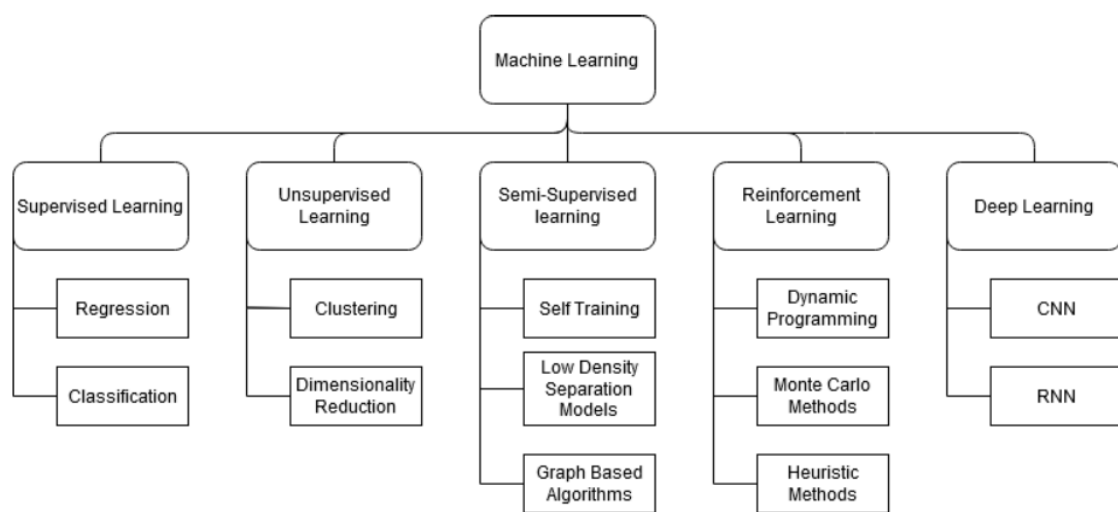


Figura 7 Diferentes tipos de *Machine Learning* e os correspondentes algoritmos [48]

3.3.1. *MACHINE LEARNING* RELACIONADO COM O CONSUMO ENERGÉTICO

O *machine learning*, tal como anteriormente mencionado, possui um enorme número de aplicações sendo uma delas na vertente do consumo energético. Com o aumento contínuo do consumo de energia, têm sido feitos cada vez mais esforços para encontrar maneiras inovadoras de usar o *machine learning* neste setor com o objetivo de tornar mais eficiente o seu consumo [49].

A utilização destes algoritmos tem tipicamente dois propósitos específicos: definir/descobrir diferentes perfis ou padrões de consumo ou estimar/prever o consumo energético.

De forma a definir perfis de consumo, recorre-se tipicamente à técnica de *clustering*, uma forma de *unsupervised learning*. A finalidade desta técnica é a de detetar subgrupos dentro de um conjunto de dados, sendo que este subgrupo contém semelhanças entre si. Um algoritmo muito utilizado no setor energético para implementar estes conjuntos é o *K-means* [50] [51] [52].

A previsão de consumo energético é uma tarefa complexa devido à presença de padrões complexos lineares e não lineares nos *datasets* temporais do consumo de energia.

Uma das opções para estimar o consumo energético é o *Autoregressive Integrated Moving Average* (ARIMA) um modelo de análise estatística que é uma forma de análise de regressão que avalia a força de uma variável dependente em relação a outras variáveis. Este método oferece um bom desempenho a identificar os aspetos lineares de uma série temporal.

Outra opção para estimar consumos tem sido o recurso ao *deep Learning* através de algoritmos baseados em redes neuronais tais como *Artificial Neural Network* (ANN), *Recurrent Neural Network* (RNN) e *Long Short Term Memory* (LSTM). Estes algoritmos têm um melhor desempenho a identificar aspetos não lineares do que lineares [53].

4. IMPLEMENTAÇÃO DA REDE IOT

O cenário de teste da rede IoT, que deverá funcionar na siderurgia da Borçelik, não se encontra definido em detalhe, sendo apenas referido no anexo técnico do projeto IS4SE que irá ter como objetivo a medição de parâmetros ambientais e de parâmetros energéticos.

Tendo como base esse requisito do projeto, a rede IoT a implementar no âmbito das tarefas atribuídas à Controlar, que deverá permitir uma primeira prova de conceito, inclui dois tipos de nós específicos: um referente à medição de parâmetros ambientais e outro sobre a medição de parâmetros energéticos.

Neste capítulo faz-se uma apresentação da arquitetura da rede desenvolvida, descrevendo cada um dos seus componentes e as suas interações.

4.1. ARQUITETURA GENÉRICA

Tendo em vista a criação de uma rede IoT versátil, um dos requisitos identificados foi a possibilidade de integração de diferentes protocolos de comunicação e a utilização de um servidor único para agregar toda a informação proveniente dos diversos elementos que a constituem.

Assim sendo, tendo em vista o capítulo relativo ao estado de arte sobre as tecnologias utilizadas em redes IIoT, optou-se pela utilização do protocolo LoRaWAN no nó de parâmetros ambientais, supondo que a taxa de amostragem seja suficiente para este nó. Para o nó de energia recorre-se ao MQTT, permitindo uma maior taxa de amostragem.

Para a implementação desta rede IoT foram utilizadas diversas tecnologias e ferramentas, sendo que cada uma desempenha uma função específica na plataforma criada. A Figura 8 apresenta a arquitetura genérica do sistema onde é possível identificar os elementos

principais que a constituem. Nas seções seguintes serão detalhadas as opções tecnológicas, aspectos de implementação e formas de interação entre os diversos sub-sistemas.

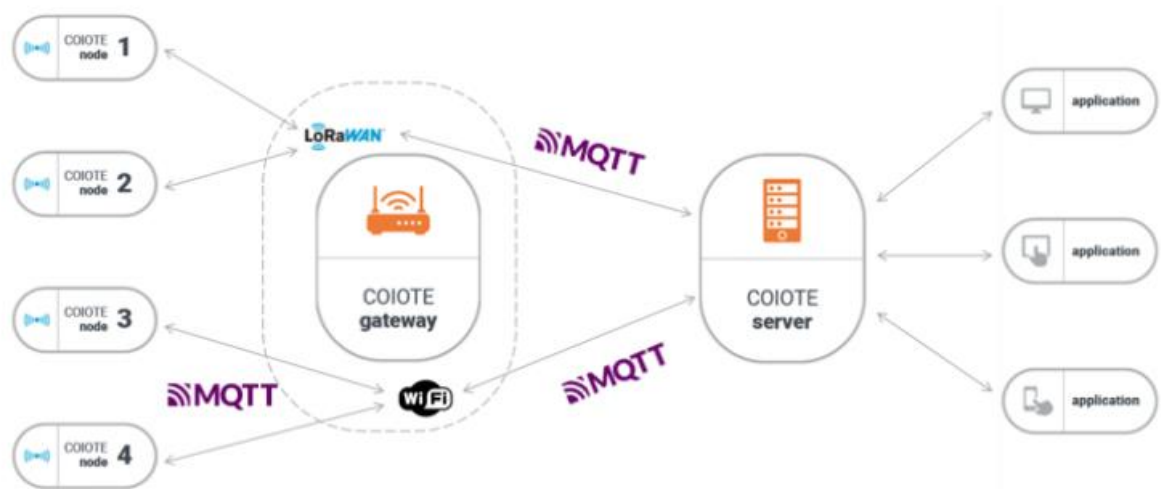


Figura 8 Arquitetura típica de uma rede IoT

4.2. SERVIDOR

A característica principal do servidor da solução de IoT desenvolvida é a capacidade de associar toda a informação dos diferentes nós. Ou seja, reunir na mesma base de dados a informação proveniente de todos os nós existentes na rede, munidos de diferentes protocolos de comunicação, permitindo assim a agregação dos diversos dados armazenados no servidor (Figura 9).

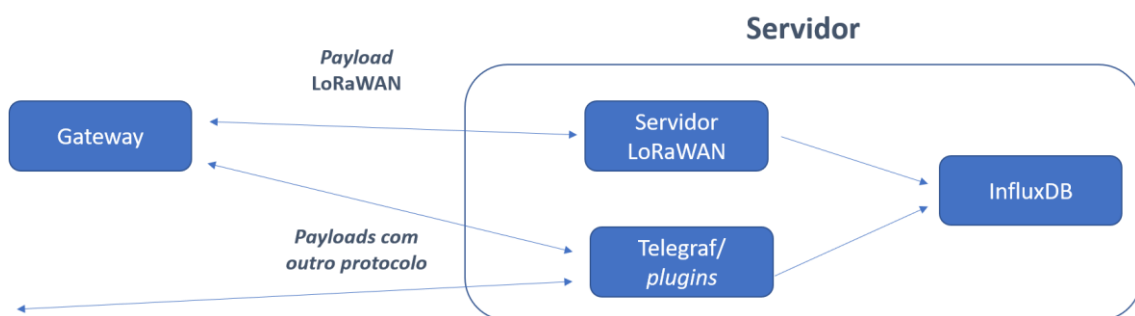


Figura 9 Arquitetura típica de uma rede IoT

Como foi referido no subcapítulo do estado de arte sobre o LoRaWAN, esta tecnologia transfere grande parte da complexidade relacionada com a gestão dos pacotes para o lado do servidor, necessitando assim de um servidor dedicado. A evidencia desta característica tem o propósito de explicar a distinção entre os *payloads* LoRaWAN e os *payloads* provenientes de outros protocolos presente na Figura 9.

O ChirpStack é um servidor LoRaWAN *open-source* capaz de manipular os pacotes de *uplink* provenientes de uma *gateway* específica e agendar os pacotes de *downlink*. Esta ferramenta suporta as principais funcionalidades de um servidor e permite a sua configuração através de uma interface *web* [54].

Na Figura 10 podemos observar a arquitetura desta solução e aferir que, para o funcionamento deste servidor, são necessárias três componentes principais: (1) ChirpStack Gateway Bridge, (2) ChirpStack Network Server e (3) ChirpStack Application Server.

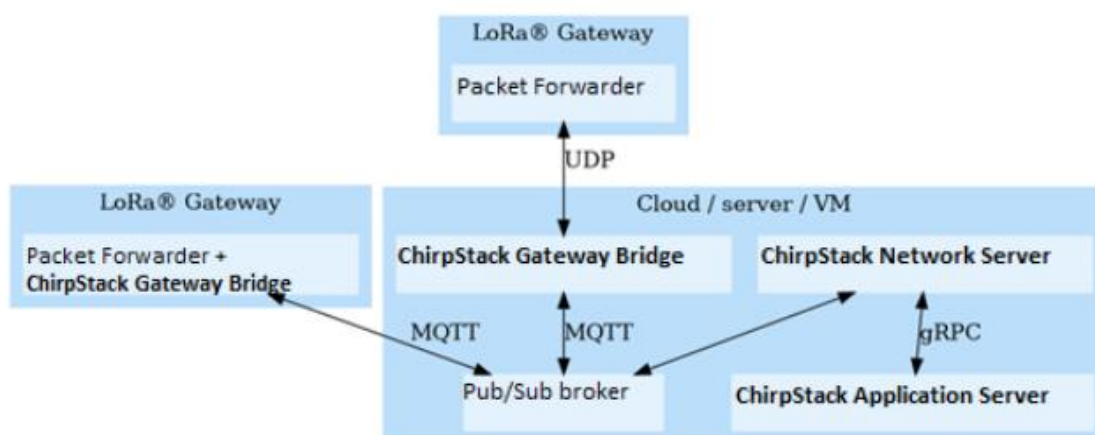


Figura 10 Arquitetura do servidor ChirpStack [55]

O ChirpStack Gateway Bridge possibilita dois tipos de comunicações com o servidor. Por um lado, caso esta ferramenta esteja instalada no servidor, esta comunicação é realizada através de *User Datagram Protocol* (UDP). Por outro lado, caso a instalação seja feita na gateway a comunicação é feita por MQTT.

A escolha recaiu pela segunda opção uma vez que o protocolo UDP, por si só, não oferece garantia de entrega de pacotes, logo é mais fiável usar MQTT. Esta ferramenta será apresentada com maior pormenor no ponto referente à *gateway*.

O ChirpStack Network Server é a peça fulcral desta solução uma vez que, é responsável pela manipulação e armazenamento da informação e, também, por comunicar com outros dispositivos. O ChirpStack Application Server é a *web Graphical User Interface* (GUI) deste servidor.

A partir da *web* GUI é possível criar, organizar e adicionar as *gateways*. Dentro das Gateways é possível configurar os diferentes nós LoRaWAN que irão transmitir os pacotes para as determinadas *gateways*. Aqui é possível configurar todos os aspetos relativos à comunicação, como por exemplo o tipo de ativação do dispositivo que se vai ligar à rede.

O ChirpStack possui uma variada gama de softwares compatíveis, sendo um deles o influxDB [56]. Este é uma *Time Series Database* (TSDB), isto é, uma base de dados otimizada para dados com registo de data e hora. Estes dados de séries temporais são medições ou eventos monitorizados, redimensionados e agregados ao longo do tempo pela base de dados.

Esta característica faz com que este tipo de base de dados seja utilizado especificamente para lidar com eventos ou medições com *timestamp* associado, o que a torna ideal para agregar todas as medições de uma rede IoT. Por esta razão, a influxDB foi escolhida como a base de dados que irá concentrar toda a informação [57].

O Telegraf é um software integrante do leque da influxData, empresa que desenvolveu a influDB. Este software age como agente de servidor que funciona à base de plug-ins para recolher e enviar medidas e eventos de outras bases de dados, sistemas e sensores de IoT.

Assim sendo, o Telegraf é um agente que recolhe informação/medidas de um espectro variado de inputs para um leque de variado de outputs e, por isso oferece uma enorme agilidade à solução IoT [58].

4.3. GATEWAY

Neste subcapítulo será apresentado todas as partes integrantes da *gateway*, desde o *hardware* e *software* que contém. Será ainda explicado o seu modo de funcionamento e de que forma se conecta ao servidor. Por fim será demonstrada a interface de configuração da mesma.

4.3.1. ARQUITETURA E *HARDWARE*

Visto que, a principal premissa da rede IoT construída é poder simultaneamente usufruir das vantagens ou pontos positivos dos protocolos Wi-Fi e LoRaWAN, a *gateway* (Figura 12) integrante nesta solução teria de ser multiprotocolo, suportando não só os dois protocolos mencionados, como também o MQTT uma vez que, é necessária a sua utilização nos dois protocolos anteriormente mencionados, originando a arquitetura da Figura 11.

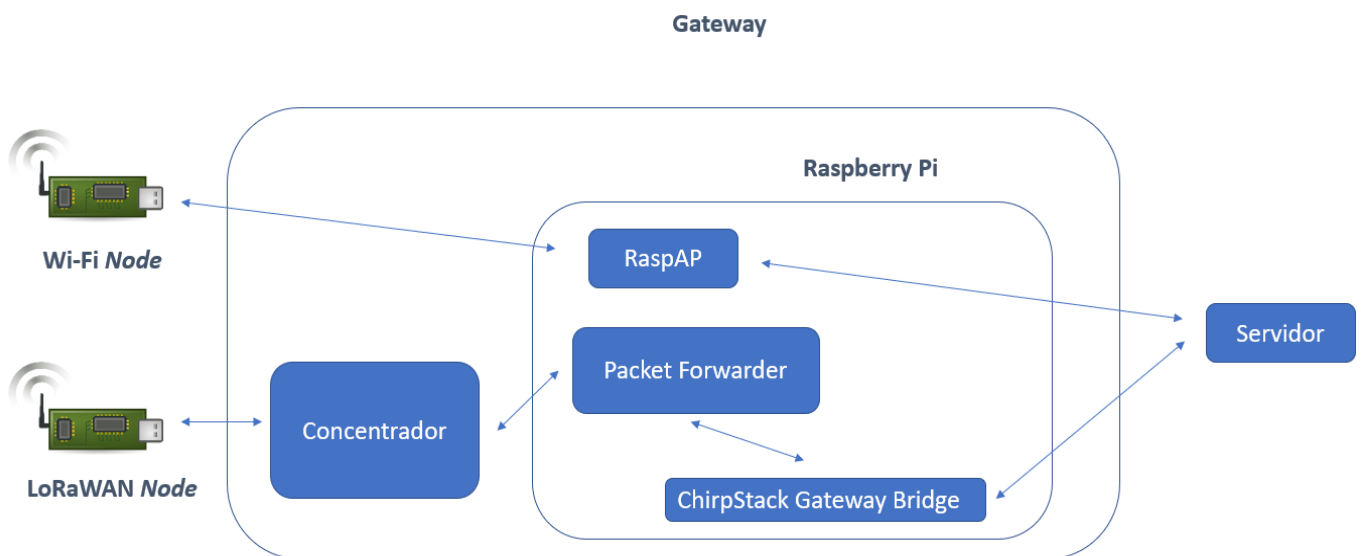


Figura 11 Arquitetura da gateway



Figura 12 Gateway desenvolvida

A escolha de hardware para a *gateway* recaiu no uso de um raspberry Pi 3B+ [59], juntamente com um concentrador de pacotes LoRaWAN ic880a fabricado pela IMST (Figura 13) , que faz uso do chip SX1301 proprietário da Semtech.



Figura 13 Concentrador ic880a [60]

Este concentrador pode receber até 8 pacotes de nós LoRaWAN enviadas em simultâneo com diferentes canais e fatores de propagação. Algumas dos seus principais recursos incluem [61]:

- Banda de frequência 868MHz
- Interface *Serial Peripheral Interface* (SPI)
- Potência de até 20 dBm
- Alcance de até 15 Km em linha de vista
- Possui um conjunto de *Application Programming Interfaces* (API)

O concentrador comunica por SPI com o raspberry Pi (Tabela 9). Segundo a empresa que o fabricou, o *reset* é feito no arranque do sistema, permitindo promover uma comunicação mais eficiente entre os dois sistemas.

Tabela 9 Ligação entre os pinos do concentrador e do raspberry Pi [61]

iC880A pino	Descrição	RPi pino
21	Fonte 5 V	2
22	GND	6
13	<i>Reset</i>	22
14	SPI CLK	23
15	MISO	21
16	MOSI	19
17	NSS	24

Ainda relacionada com o *hardware* outra questão que se teve em conta foi quanto à alimentação da solução desenvolvida. Uma vez que, a gateway irá necessitar sempre de estar ligada à rede interna, onde o servidor se encontra hospedado, por *ethernet*. Utilizou-se essa mesma ligação em simultâneo com o uso de um *Power over Internet* (PoE)

Hat,(Figura 14), para fornecer a energia necessária a todos os componentes. De referir que o cabo *ethernet* tem que também estar ligado a uma entrada PoE da rede (Figura 14).

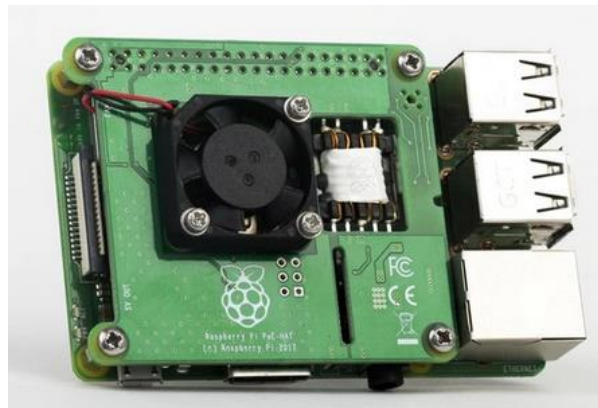


Figura 14 PoE Hat que alimenta a gateway [62]

O *LoRa Gateway Project* é o nome do diretório que contem o código fonte necessário para construir um gateway LoRa baseada num chip da Semtech [63]. Este diretório após ser complicado oferece um conjunto de APIs que habilita o concentrador de diferentes recursos, sendo que o software utilizado foi o *packet forwarder*. Este é executado no raspberry, que recebe os pacotes de rádio frequência (RF) que são capturados pelo concentrador e adiciona-lhes *metadata* e o status da gateway.

Para além desta ferramenta, para completar a restante cadeia de *uplink* é ainda utilizado *chirpStack Gateway Bridge* que ira ser responsável por fazer chegar ao servidor os pacotes mencionados em cima. No seguinte *Message Sequence Chart* (MSC) é possível observar a interligação destas duas ferramentas bem como os protocolos por elas utilizados (Figura 15).

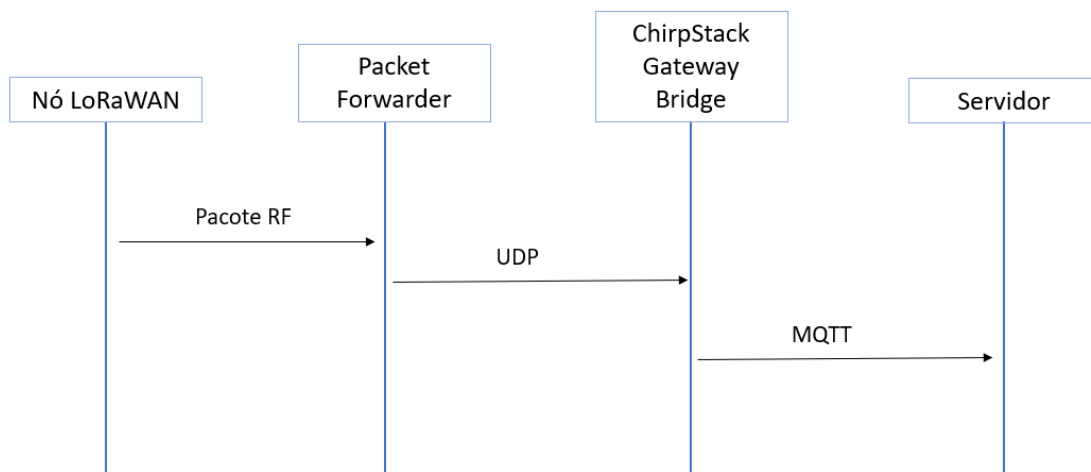


Figura 15 MSC do *Uplink* das mensagens LoRa

A import ncia do uso da gateway est  relacionada com o tipo de n os que ir o compor a rede IoT. Se forem usados n os LoRaWAN, o uso da gateway torna-se imprescind vel porque, ao n o ser utilizada, n o   poss vel receber os pacotes RF e, posteriormente, envi -los para o servidor. Assim sendo a gateway tem a fun o de desmodular os pacotes LoRa, convert -los em pacotes IP e envi -los recorrendo ao MQTT.

Num polo oposto, o uso de n os *ethernet* n o precisar o do uso de gateway, uma vez que, estes n os estar o ligados   rede do servidor e enviar o diretamente os pacotes IP. No entanto, se estiverem integrados na rede IoT n os Wi-Fi, o uso da gateway pode ser bastante vantajoso, pois ir  permitir que o tr fego proveniente desses n os n o interfira diretamente com a rede interna, uma vez que o raspberry ir  criar um *Access Point* (AP).

4.3.2. INTERFACE WEB DE CONFIGURA O

Tendo em conta a gateway desenvolvida para esta solu o IoT, a interface web foi dividida em tr s partes: LoRa Gateway, Wi-Fi gateway e Admin (Figura 16).

Importa referir que esta interface web partiu de um projeto *open-source* chamada RaspAP. Este projeto tem como objetivo munir dispositivos baseados em Debian, como por exemplo

o raspberry PI, com uma interface capaz de configurar um AP e gerir as ligações Wi-Fi desses mesmo dispositivos.

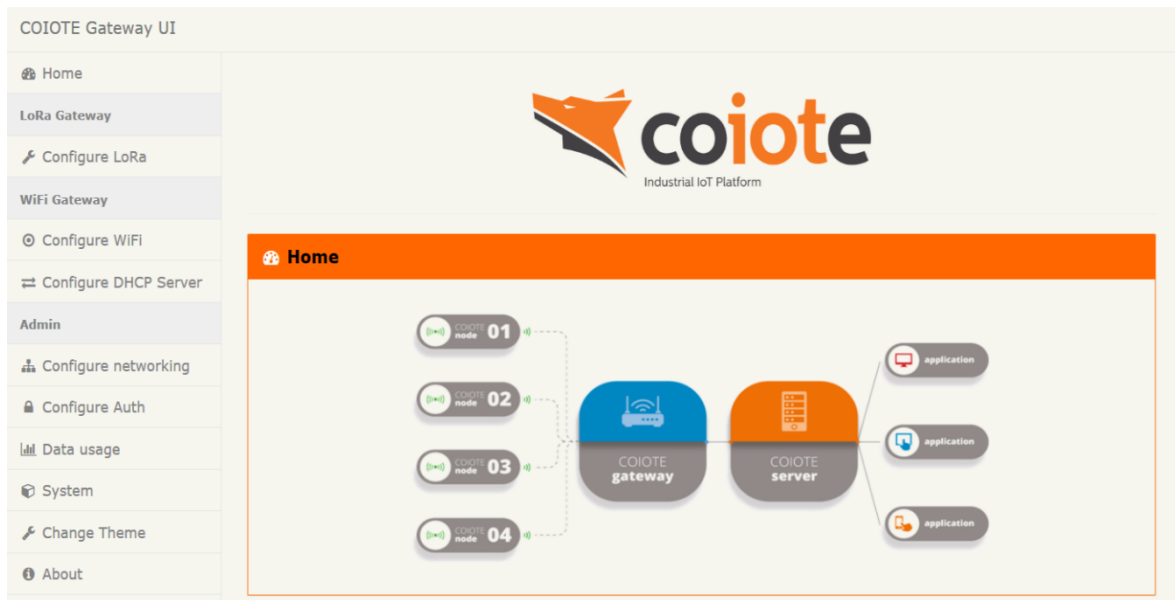


Figura 16 Aplicação Web da Gateway

A principal alteração feita ao código fonte do RaspAP foi relacionada com secção intitulada LoRa Gateway. Esta secção permite ao utilizador configurar todos os parâmetros, de forma a que seja possível receber e reencaminhar os pacotes RF, provenientes dos nós LoRaWAN.

Na Figura 17 é possível observar o funcionamento das duas aplicações previamente mencionadas, capazes de munir a gateway com o protocolo LoRaWAN. Através da imagem conclui-se que, estando ambos a verde, a gateway está habilitada a comunicações LoRaWAN. Isto significa que, apesar de ser uma gateway multiprotocolo, é possível desligar-se uma destas duas funcionalidades e esta passar apenas a permitir comunicação Wi-Fi, auferindo-lhe assim uma maior versatilidade.

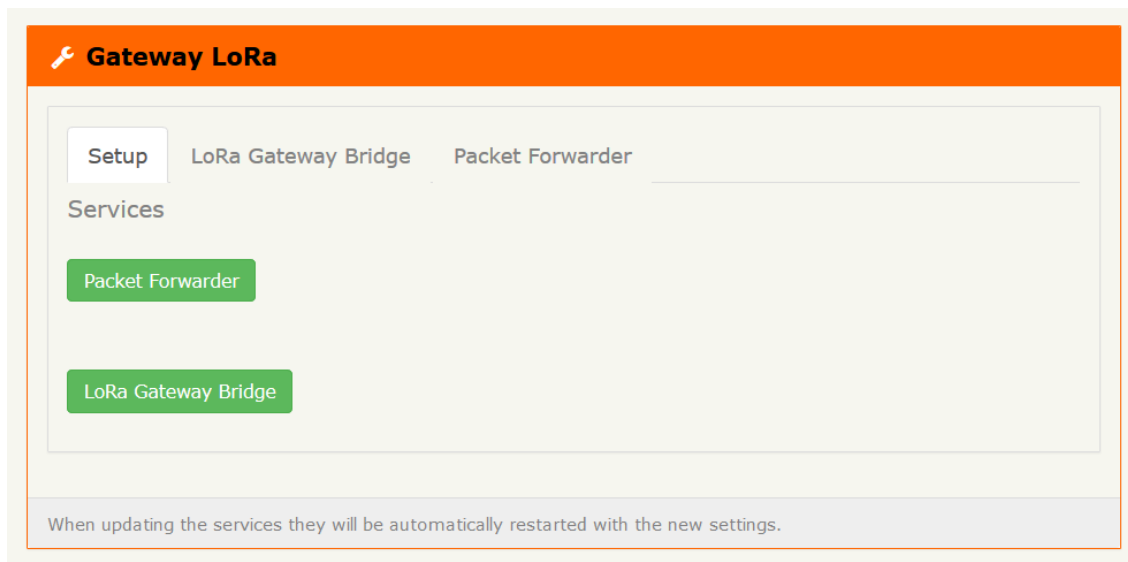


Figura 17 Painel dos serviços LoRaWAN

As duas abas expostas nas Figura 18 e Figura 19 permitem a configuração simples e rápida, dos dois serviços afetos ao funcionamento deste protocolo.

O objetivo inerente a estes desenvolvimentos, foi automatizar a configuração destas ferramentas, visto que, a sua configuração carece de uma alteração manual de ficheiros de configuração.

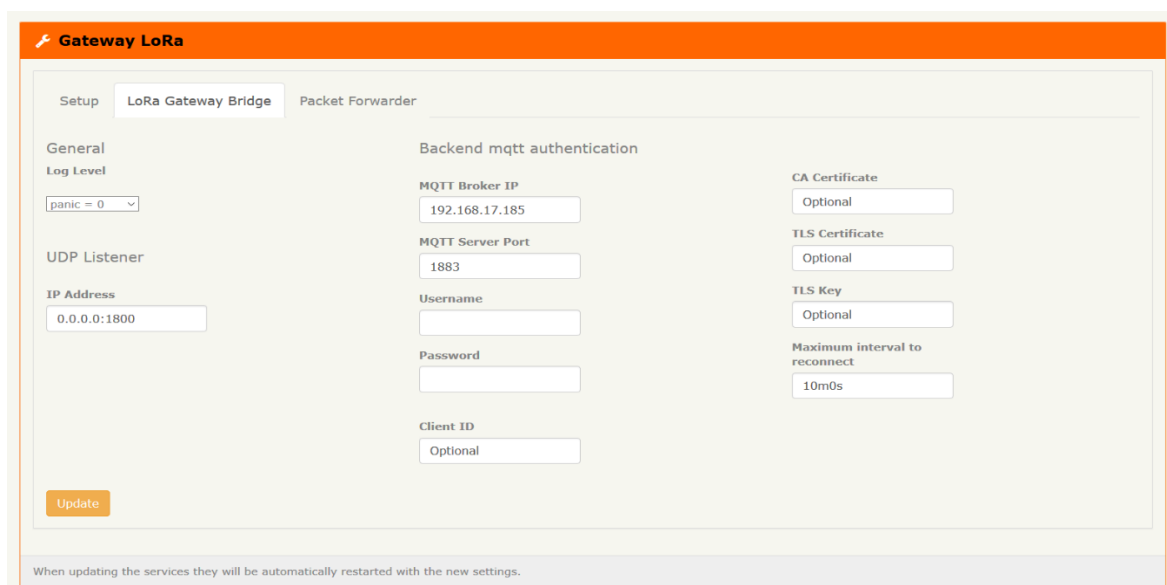


Figura 18 Painel de configuração do ChirpStack Gateway Bridge

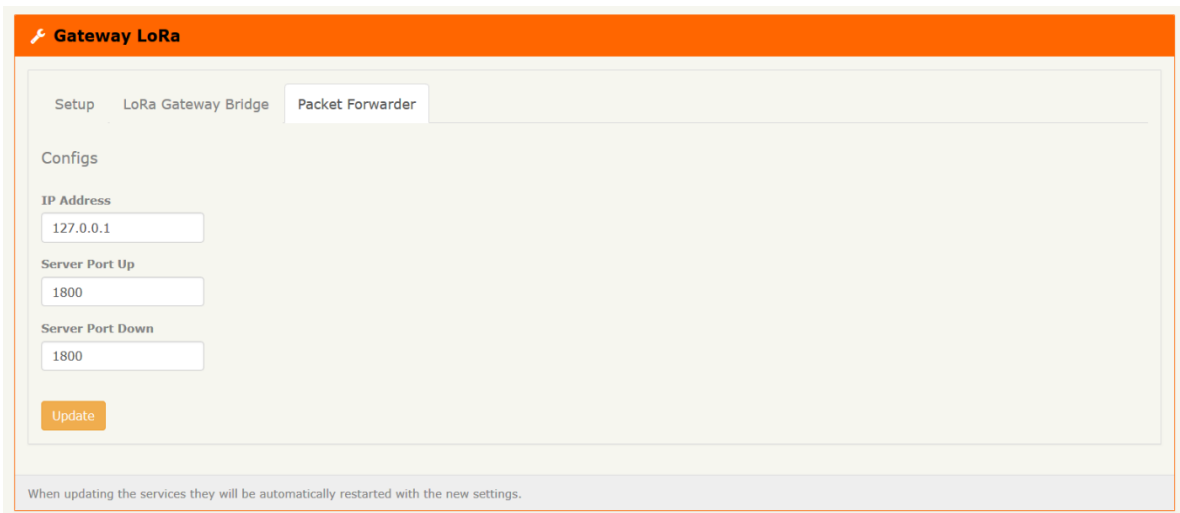


Figura 19 Painel de configuração do Packet Forwarder

A principal finalidade com a Wi-Fi Gateway foi modificar e criar um AP ou um *hotspot* com dois painéis de configuração. O primeiro (Figura 20) oferece a possibilidade de configuração dos aspetos mais simples de um AP, nomeadamente o seu *Service Set Identifier* (SSID), o tipo de segurança e a password.

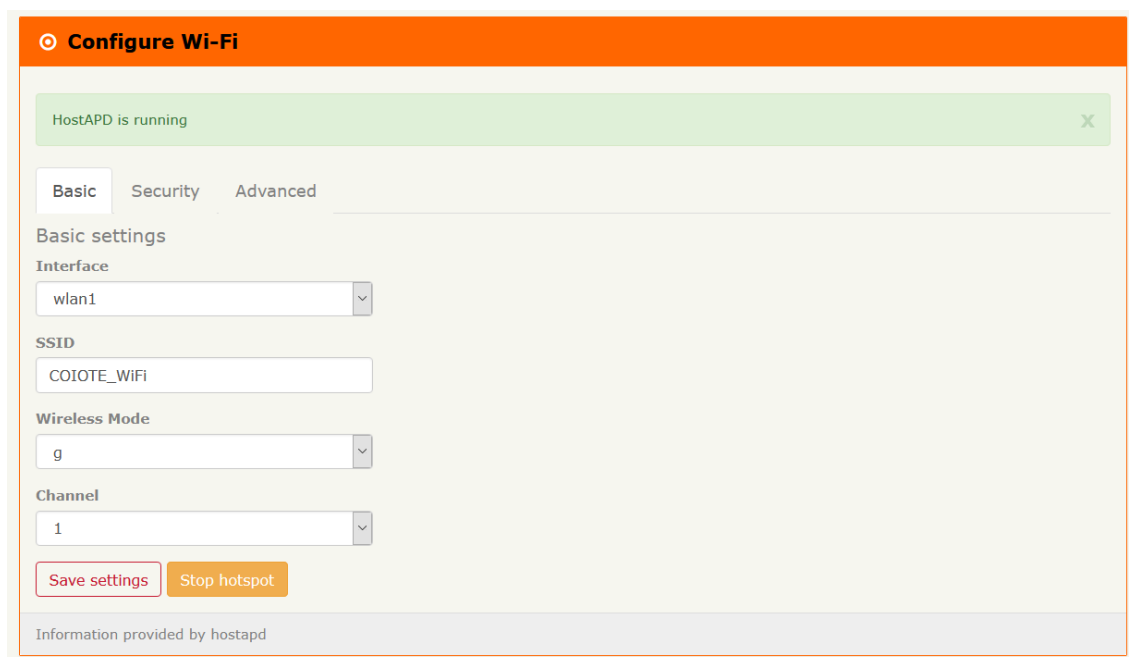


Figura 20 Painel de configuração do AP

O segundo painel (Figura 21), permite ao utilizador configurar o servidor *Dynamic Host Configuration Protocol* (DHCP) do AP, e configurar os parâmetros incluídos neste tipo de servidor. Este AP é de extrema importância na medida em que, todos os nós Wi-Fi irão

conectar-se a esse AP e, conseqüentemente, a gateway irá reencaminhar os pacotes para a rede interna e este, por sua vez, para o servidor. Caso o AP seja desligado, a gateway perde a comunicação Wi-Fi permanecendo apenas a comunicar com o LoRaWAN.

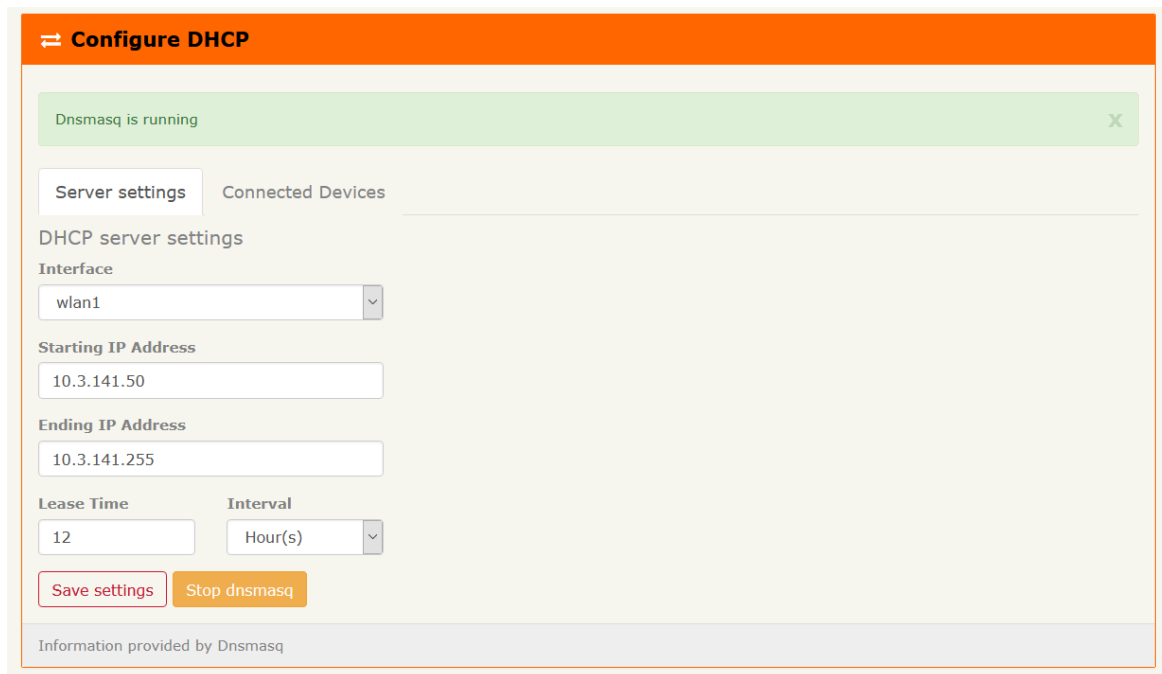


Figura 21 Painel de configuração do AP

Por fim, terceira parte da aplicação o admin, permite, alterar o IP da gateway, na rede interna. Ainda, acrescenta algumas funcionalidades ao utilizador, tal como alterar as credenciais de acesso à interface, apresentar na aba “Data Usage” um gráfico referente ao tráfego que passa pela gateway. Permite ainda, ao utilizador, na aba “System”, atuar diretamente na própria gateway e mostrar alguma informação adicional sobre a mesma.

De forma a que o *set-up* da gateway fosse o mais imediato possível. A gateway foi programada, de forma a que, a partir do momento em que seja ligada por PoE à rede onde o servidor se encontra instalado, um AP predefinido será criado e conectando-se a ele, através de umas credencias predefinidas, é possível aceder à interface demonstrada.

4.4. NÓS

Apesar de esta solução ter sido idealizada para comunicações LoRaWAN e Wi-Fi tendo em mente o projeto IS4SE, foi desenvolvida de forma a torná-la mais versátil e, conseqüentemente, ser integrada em soluções desenvolvidas pela CONTROLAR. Isto porque, a empresa usa na grande maioria *Programmable Logic Controller* (PLC). Neste sentido, foram criados os nós descritos nas próximas secções.

4.4.1. NÓS LORAWAN

Como foi demonstrado no estado de arte todos os nós LoRaWAN da usam o protocolo de *payload cayenne LPP*, que é o padrão predefinido pelo servidor ChirpStack. A adição de novos nós não é feita de forma automática, requerendo a configuração no ChirpStack Application Server dos seguintes parâmetros:

- *Gateway* – Quando a API do concentrador é instalada é configurado um Gateway ID, que é semelhante a um endereço MAC. Este identificador é necessário para registar a gateway no servidor. Outros parâmetros que podem ser configurados aquando o registo da gateway são a adição da *metadata* e a geolocalização.
- *Perfil do dispositivo* – Neste parâmetro são configurados todos os aspetos referentes ao dispositivo, desde a versão do LoRaWAN que este suporta, como os parâmetros regionais e também o tipo de ativação que irá usar para se conectar à rede. É ainda, escolhido nesta secção o tipo de classe do dispositivo.
- *Aplicação* - Por último, é feita a associação de um perfil de dispositivo com as chaves necessárias consoante o tipo de ativação escolhido e permite configurar a base de dados do InfluxDB que ira receber e armazenar a informação.

4.4.2. NÓS WI-FI/ETHERNET

O ChirpStack utiliza o protocolo MQTT para comunicação entre os seus três componentes, por isso, obrigou à instalação de um broker MQTT. Esta razão aliada ao facto de o MQTT ser um protocolo compatível com o Telegraf, tornou a utilização do MQTT por parte dos nós Wi-Fi e Ethernet mais imediata.

Além disto, as comunicações LoRaWAN têm inerente uma encriptação do algoritmo AES de 128 bits. De forma tornar o envio e a receção dos pacotes provenientes destes nós também mais segura, configurou-se o broker MQTT de modo a que todas as mensagens publicadas e subscritas exigissem um username e uma *password*.

Para o Telegraf receber como input o protocolo MQTT optou-se por seguir o exemplo da configuração demonstrada na Figura 22. Esta oferece uma configuração do IP do broker, as credenciais de acesso e os respetivos tópicos em que os nós irão publicar os *payloads*. Será preciso depois configurar o *output*, para que seja a base de dados do InfluxDB desejada.

```
[[inputs.mqtt_consumer]]
  ## Broker URLs for the MQTT server or cluster. To connect to multiple
  servers = ["tcp://127.0.0.1:1883"]

  ## Topics that will be subscribed to.
  topics = [
    "wifi_node/#",
    "ethernet_node/#",
  ]

  ## The message topic will be stored in a tag specified by this value. If set
  ## to the empty string no topic tag will be created.
  # topic_tag = "topic"

  ## QoS policy for messages
  # qos = 0

  ## Connection timeout for initial connection in seconds
  # connection_timeout = "30s"
  # max_undelivered_messages = 1000
  # persistent_session = false

  ## If unset, a random client ID will be generated.
  # client_id = ""

  ## Username and password to connect MQTT server.
  # username = "telegraf"
  # password = "metricsmetricsmetricsmetrics"

  ## Optional TLS Config
  # tls_ca = "/etc/telegraf/ca.pem"
  # tls_cert = "/etc/telegraf/cert.pem"
  # tls_key = "/etc/telegraf/key.pem"

  ## Data format to consume.
  data_format = "influx"
```

Figura 22 Exemplo de configuração do Telegraf tendo o MQTT como Input

Neste exemplo é possível observar que apenas dois tópicos são subscritos. No entanto, tendo em conta o que foi demonstrado, os tópicos do MQTT têm uma relação de hierarquia. Portanto, a partir do momento em que um nó Wi-Fi se conecta ao AP criado pela gateway, sendo a parte inicial do tópico igual a configurada este irá sempre ser adicionado automaticamente à rede.

O Telegraf irá receber o `influx`, como formato do payload. Este tem de estar de acordo com o `influxDB line protocol`, usado para gravar a informação na base de dados selecionada.

Cada linha deste protocolo representa um ponto que irá ser gravado. Na Figura 23 pode observar-se a síntese do mesmo. Este contempla 3 campos obrigatórios: um para o nome, que irá agregar um conjunto de medidas, outro para o `tag_set`, capaz de filtrar as medidas que se encontram guardadas na base de dados, sendo contudo um campo opcional; por fim, o `field_set` para as respetivas medidas [64].

Uma vez que o `timestamp` também é um campo opcional, optou-se por não o preencher e, por isso, irá obrigar o servidor a atribuir um `timestamp` assim que receba determinado `payload`.

```
weather,location=us-midwest temperature=82 1465839830100400200
|-----|
|           |           |           |
+-----+-----+-----+-----+
|measurement|,tag_set| |field_set| |timestamp|
+-----+-----+-----+-----+
```

Figura 23 Exemplo do influxDB Line Protocol [64].

4.4.3. NÓS DE PLC

Os nós PLC têm como finalidade permitirem que PLC's possam estar conectados à rede IoT, de forma a que seja possível a extração de informação dos mesmo e, conseqüentemente, o envio para o servidor.

Para atingirmos esta finalidade foram desenhadas 3 estratégias diferentes:

- **Modbus Rs485 para ethernet ou Wi-Fi** - Esta primeira alternativa visa o uso de um conversor colocado no PLC. Este irá conectar-se ao mesmo por rs485, extrair a informação do PLC e enviá-la para o servidor recorrendo ao protocolo MQTT de forma idêntica ao nós mencionados no subcapítulo anterior.
- **Modbus TCP/IP** – Foi ainda criada a hipótese de recolher dados de PLC's usando o protocolo modbus, sem o constrangimento ou necessidade de um conversor, usando o Modbus TCP/IP. Isto permite enviar a comunicação do PLC através do protocolo de comunicação TCP/IP. Desta forma, o PLC requer apenas de estar conectado à rede do servidor por ethernet. Tendo em conta que, o Telegraf não possui um plug-in para integrar este protocolo na base de dados, recorreu-se à biblioteca python PyModbus [65]. Das soluções experimentadas, esta revelou-se ser de fácil integração e robusta, disponibilizando uma vasta documentação. Seguindo o método de operação do Telegraf, este plug-in recorre ao uso de um ficheiro de configuração (Figura 24).

```

[PLC1]
Slave_ID = 1
IP = "localhost"
Port = 502
Database = "PLC1"
discrete_inputs = [
  { "name" : "Start" , "address" : "0" },
  { "name" : "Stop" , "address" : "1" },
]
coils = [
  { "name" : "Motor" , "address" : "0" },
  { "name" : "Rotor" , "address" : "1" },
]
holding_registers = [
  { "name" : "PowerFactor" , "byte_order" : "BE" , "data_type" : "FLOAT32" , "scale" : "0.01" , "address" : [0,1]},
]
input_registers = [
  { "name" : "TankLevel" , "byte_order" : "BE" , "data_type" : "INT32" , "scale" : "1.0" , "address" : [0,1]},
]

[PLC2]
Slave_ID = 2
IP = "localhost"
Port = 504
Database = "PLC2"
discrete_inputs = [
  { "name" : "Start1" , "address" : "0"},
  { "name" : "Stop1" , "address" : "1"},
]
coils = [
  { "name" : "Motor1" , "address" : "0"},
  { "name" : "Rotor1" , "address" : "1"},
]
holding_registers = [
  { "name" : "PowerFactor1" , "byte_order" : "BE" , "data_type" : "FLOAT32" , "scale" : "0.01" , "address" : [0,1]},
]
input_registers = [
  { "name" : "TankLevel1" , "byte_order" : "BE" , "data_type" : "INT32" , "scale" : "1.0" , "address" : [0,1]},
]

```

Figura 24 Ficheiro de configuração do plug-in Modbus TCP/IP

Como é possível verificar, este ficheiro de configuração possibilita de uma forma fácil e intuitiva a configuração do *Uniform Resource Locater* (URL) dos PLC's, como também dos registos dos parâmetros e da base de dados onde contará a informação a ser enviada.

- **OPC-UA** – Este protocolo não foi descrito de forma aprofundada no estado de arte, porque não foi amplamente aprofundado e integrado nesta solução. No entanto, foram feitos testes recorrendo a um plug-in em *javascript* previamente criado [66]. O ficheiro de configuração deste *software* (Figura 25), contem três componentes de configuração, uma responsável pela configuração dos parâmetros de conexão OPC-UA, outra relacionada com a conexão à base de dados do influx e um segmento final responsável pela configuração das medidas que serão inseridas nessa mesma base de dados.

```

#The OPCUA connection parameters. If you want to use anonymous auth,
# remove the username and password lines.
[opcua]
url          = "opc.tcp://localhost:53530/OPCUA/SimulationServer"
user         = "test"
pass        = "test1"

# The InfluxDB connection parameters. Use a connection url containing all
# details, ie. http(s)://user:password@host:port/database
[influx]
url          = "http://user:password@localhost:8086/opcua"
writeInterval = 1000          # optional. defaults to 1000ms
writeMaxPoints = 1000        # optional. defaults to 1000 points

# For each metrics you want to collect, add a [[metrics]] object.
[[metrics]]
measurement = "polled1"
datatype    = "number"
tags        = { simulation = "true", location = "ghent" }
nodeId      = "ns=5;s=Sinusoid1"
method      = "polled"
interval    = 1000

```

Figura 25 Ficheiro de configuração do plug-in OPC-UA

A necessidade de produzir uma prova de conceito, mostrando a possibilidade de uma integração deste protocolo na solução, advém do facto de o OPC-UA ser cada vez mais difundido e utilizado em IIoT [67] [68] [69].

4.5. LEITURA E VISUALIZAÇÃO DOS DADOS DOS SENSORES

Esta camada de rede IoT abrange as aplicações que fazem uso dos dados/informação recolhida por todos os sensores conectados à rede, cuja informação se encontra armazenada no servidor, permitindo que esta secção final da rede IoT seja a sua interface para o utilizador.

Para esse efeito, inicialmente, foi utilizada a ferramenta Chronograf da influxData [70]. Esta ferramenta oferece uma *user interface* e uma componente administrativa da base de dados influxDB. Permitindo a criação de novas bases de dados bem como utilizadores e gerir as respetivas permissões.

Sendo uma ferramenta desenvolvida para a base de dados influxDB é totalmente compatível com a mesma, possibilitando assim, de uma forma instantânea e intuitiva,

selecionar as medidas das diferentes bases de dados configuradas e criar os mais diversos *dashboards* (Figura 26).



Figura 26 Exemplo de um *dashboard* criado no Chronograf [71]

Além disto, o Chronograf permite a criação de diversos tipos de alerta que permitem informar o utilizador caso algum sensor falhe o envio de algum pacote e cria limites. Ainda, caso um alerta seja despoletado oferece um leque variado de outputs desde mensagens MQTT até e-mails.

Com o propósito de aprofundar este segmento da solução IoT e fornecer uma informação mais ampla e diversificada, foi desenvolvida a aplicação *web* que é descrita no próximo capítulo.

5. ANÁLISE E TRATAMENTO DOS DADOS

Uma parte importante da operação e gestão de uma rede IoT é poder processar e analisar os dados provenientes desta, de modo a obter informações valiosas sobre a mesma. Com esta finalidade, foi desenvolvida uma aplicação *web*, de forma a criar estatísticas e analisar os dados dos diferentes nós conectados à rede IoT.

Assim sendo, ao longo deste capítulo serão abordadas as diferentes tecnologias empregues, bem como as principais funções criadas para o funcionamento da aplicação.

Por fim, será apresentada a estrutura dos diferentes *scripts* constituintes da aplicação *web* e serão apresentadas, tendo em conta o seu menu lateral (Figura 27), as funcionalidades implementadas.

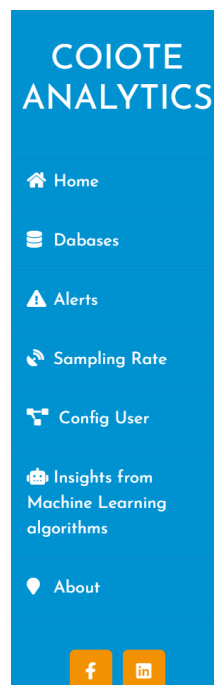


Figura 27 Menu lateral da aplicação *web*

5.1. MODELO DE DADOS

A implementação desta ferramenta de processamento e análise de dados, contemplou o desenvolvimento de funções e classes transversais a toda a plataforma, isto é, funções integradas em diversas funcionalidades. Estas funções, aliadas a bibliotecas e ferramentas da linguagem python, permitiram este processamento e visualização estruturada dos dados. Nas próximas secções serão enumeradas essas ferramentas e respectivas funções.

5.1.1. TECNOLOGIAS UTILIZADAS

O desenvolvimento da aplicação web foi suportado por um conjunto de tecnologias/ferramentas sendo as seguintes as mais importantes:

- **Pandas**

Pandas é uma biblioteca de *software*, de alto nível, para a linguagem python que permite estruturar conjuntos de dados ou *big Data*, de forma rápida e flexível. É, portanto, utilizada para manipular e analisar os dados e, por isso, uma característica necessária para tratar os dados provenientes dos diferentes sensores e destes extrair diferentes estatísticas [72].

- **Flask**

O flask é um micro *framework web* em python que oferece a flexibilidade desta linguagem de programação, que se traduz num modelo simples para o desenvolvimento *web*, permitindo assim economizar o tempo de desenvolvimento [73]. Visto que, as diversas ferramentas utilizadas para o tratamento e análise de dados fazem uso de python, a escolha para o desenvolvimento da plataforma *web* recaiu numa ferramenta que fizesse uso da mesma linguagem de programação.

- **SQLite**

O SQLite é uma biblioteca em linguagem C que implementa um mecanismo de base de dados *Structured Query Language* (SQL) pequeno, rápido e de alta confiabilidade. Neste

sentido, tornou-se uma das bases de dados mais usadas [74] e, conseqüentemente, o mecanismo escolhido para guardar os diferentes resultados da análise dos dados.

- **Scikit-learn**

O Scikit-learn é uma biblioteca gratuita de *machine learning* para python, que possui uma API de alto nível capaz de implantar os mais diversos algoritmos, como por exemplo, *random forests*, *support vector machine*, entre outros. Ainda, suporta as bibliotecas numérica e científica do python NumPy e SciPy [75].

5.1.2. PRINCIPAIS FUNÇÕES

As principais funções e classes utilizadas para o funcionamento da plataforma são as seguintes:

- **Interface com a base de dados InfluxDB**

Sendo a InfluxDB a base de dados (BD) onde são guardadas as informações, é necessário extrair e tratar os valores dos diferentes sensores. Assim sendo, a classe `InfluxDB` criada referente a esta BD contempla as seguintes funções:

- `Main_stats ()` – Faz *query* das informações a serem apresentadas na página inicial da plataforma;
- `query_points ()` – Faz *query* a todos os pontos de uma base de dados e apresenta-os na forma de um dicionário python, em que a *key* será o nome da medida e o *value* uma lista com todos os pontos correspondentes. A este dicionário é acrescentada uma *key* com o *timestamp* dos parâmetros. Importa também referir que, o formato deste é alterado do padronizado pela InfluxDB, rfc339, para o formato compatível com a biblioteca Pandas, de forma a que seja possível que esta trate os dados.
- `query_points_2()` – Esta função desempenha exatamente o mesmo processo que a enunciada em cima, com a diferença que faz *query* a BDs que tenham sido criadas para armazenar informação proveniente de nós

LoRaWAN. No capítulo seguinte é explicado o porquê de requerer duas abordagens.

- `query_point ()` – Esta função faz query de apenas uma medida específica da base de uma dada BD.
- `query_point_2 ()` – Query de uma medida específica da BD associada a um nó LoRaWAN.

- **Interface com a base de dados SQLite**

Uma vez que o objetivo principal desta plataforma é extrair informação dos nós e criar estatísticas através dos mesmos, recorreu-se à base de dados SQLite para guardar os registos dos valores obtidos.

Na Figura 28 é possível observar a estrutura base da BD. Esta é composta por uma tabela *user* que contém a informação de todas as pessoas que têm acesso à plataforma, uma tabela composta pelos alertas configurados na plataforma e uma tabela que regista sempre que um alerta é despoletado.

As restantes tabelas são referentes a cada parâmetro pertence às BD do InfluxDB que se encontram a ser analisadas pela plataforma. Estas encontram-se organizadas pelas estatísticas mensais, que incluem o número de pontos, o máximo e o mínimo valor, bem como a média e o desvio padrão. Na figura abaixo é possível identificar as medidas que compõem um nó de medição de energia:

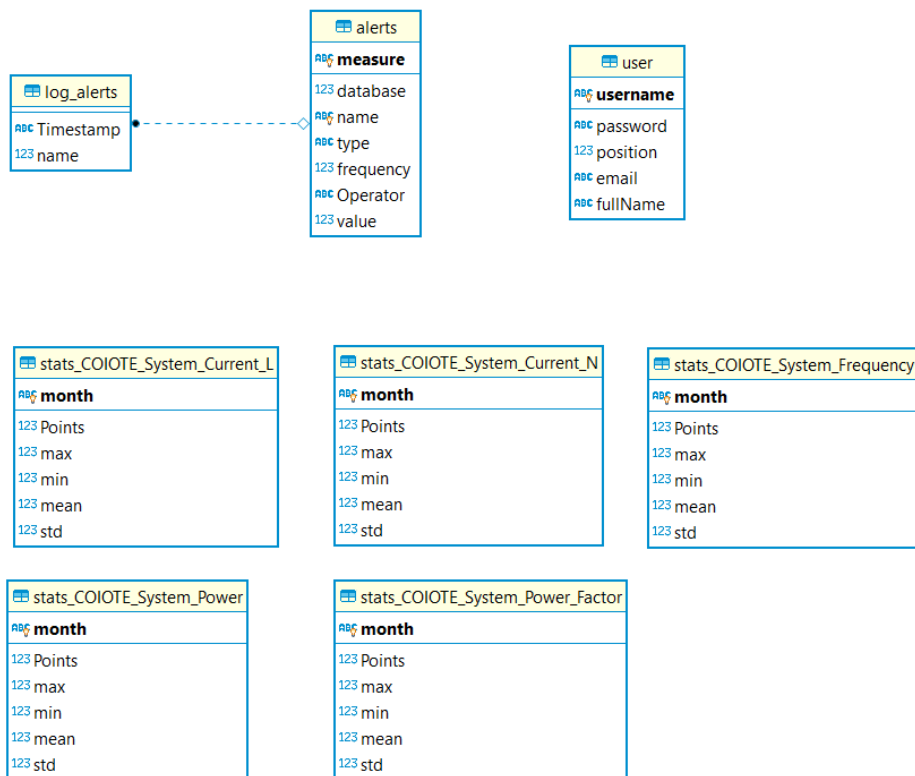


Figura 28 Exemplo das tabelas da base de dados

Tendo em conta as tabelas enumeradas, a classe responsável pela ligação e gestão da BD SQLite terá de ser capaz de adicionar e remover alertas, de registar as entradas que despoletem alertas, preencher e atualizar as tabelas com as estatísticas de cada parâmetro e, caso seja adicionada uma nova BD proveniente de um novo nó, criar as tabelas correspondentes.

- **Cálculo das estatísticas**

Esta classe é responsável por tratar os dados presentes no InfluxDB através da utilização da biblioteca Pandas. Esta função recebe como parâmetro o resultado das funções `query_point ()` ou `query_point_2 ()` e contempla funções inerentes à classe que permite calcular parâmetros que serão demonstrados nos seguintes subcapítulos.

5.2. ESTRUTURA DO SOFTWARE DA PLATAFORMA

A Figura 29 apresenta a estrutura de diretórios que compõem o servidor *web* criado pelo flask, bem como os diferentes scripts que permitem o funcionamento da plataforma.

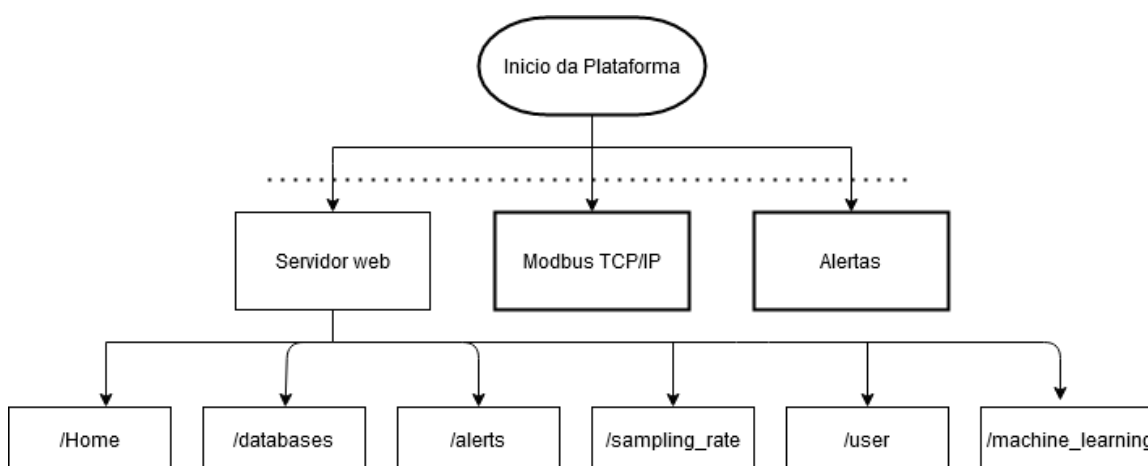


Figura 29 Estrutura da plataforma

Tal como indicado no esquema, assim que a plataforma é iniciada os scripts responsáveis pela comunicação Modbus TCP/IP e pelos alertas são arrancados em simultâneo com o servidor web.

O `modbus_tcp.py` é responsável pela receção dos pacotes que fazem uso da tecnologia Modbus TCP/IP. Este funciona em simultâneo com o servidor web e faz uso do ficheiro de configuração demonstrado na Figura 24.

De um modo sintetizado, este *software* assume cada secção do ficheiro de configuração como um PLC e utiliza o `slave_ID`, `Port` e `IP` para estabelecer conexão com o PLC. São também identificadas as medidas que o script irá ler, que serão irá enviadas para a base de dados do InfluxDB explicitada na configuração. De notar que, caso sejam configurados múltiplos PLCs, será criada uma *thread* para cada um deles. Esta, por sua vez, irá fazer a leitura e escrita na base de dados, em simultâneo com as *threads*, correspondentes aos demais PLCs.

O outro *script* é responsável pelos alertas da plataforma e funciona de maneira muito semelhante ao responsável pelo Modbus TCP/IP, com a exceção de que sempre que é despoletado um alerta é gravado na base de dados SQLite. Assim sendo, existem três tipos de alerta:

- *DeadMan*- É definido um intervalo de tempo, na medida em que, se o nó não enviar um *payload* durante esse período é despoletado o alerta.
- *Threshold* – São escolhidos um nó e um parâmetro associado e caso o valor desse parâmetro ultrapasse uma expressão determinada, como por exemplo, ≥ 20 , é despoletado o alerta.
- *Relative* – Idêntico ao de cima, no entanto, neste tipo de alerta o valor a ser definido para criar a expressão é relativo. Neste caso a expressão ≥ 20 , significaria que caso o valor seguinte fosse maior 20 unidades o alerta seria despoletado.

5.3. FUNCIONALIDADES DA APLICAÇÃO WEB

Tal como enunciado, este subcapítulo irá enumerar e explicar com detalhe a lógica e o modo de funcionamento de cada uma das funcionalidades da aplicação *web* presentes na Figura 29, com a exceção da opção *About* que apenas mostra ao utilizador um simples resumo sobre a aplicação.

5.3.1. HOME

No capítulo referente à rede IoT, foi mencionado que o Telegraf, para além de integrar nós que comuniquem através de diferentes tecnologias, permite receber e registar na base de dados influxDB outro tipo de informações.

Recorreu-se então a diversos *plugins* que permitem monitorizar o estado do computador em que o servidor está a correr e que proporcionam informações extras referentes à InfluxDB.

De notar que o Telegraf contém três principais campos de configuração, um define o comportamento do próprio software e os outros dois definem os *plugins* de entrada e saída. Com isto em mente, conclui-se que para que a informação esteja agregada num só local o *plugin* de saída, o InfluxDB, que terá de ser o mesmo para todas as entradas.

Como o próprio nome indica esta é a página inicial e, por isso, assim que a autenticação é executada aparece esta página, onde são apresentadas as informações consideradas mais pertinentes dos inúmeros parâmetros medidos pelos *plugins* anteriores.

5.3.2. DATABASES

Esta funcionalidade permite ver a análise de dados dos diferentes parâmetros aglomerados nos diferentes nós que se encontram conectados à rede. Para que tal fosse possível, recorreu-se a um ficheiro de configuração - “databases.ini”, representado na Figura 30.

```
# databases info
[_internal]
text = Internal database created automatically
image = ../static/images/cloud-server.png
type = none

[Coiote_System]
text = Database with infomation displayed in the Home page
image = ../static/images/cloud-server.png
type = none

[Smart_Composter]
text = Sensor that measures environmental parameters
image = ../static/images/Smart_Composter.PNG
type = lora

[Energy_Meter]
text = Energy meter of a workbench using LoRaWAN
image = ../static/images/cloud-server.png
type = lora
```

Figura 30 Ficheiro de configurações das bases de dados

Este ficheiro, para além de permitir customizar alguns aspetos visuais, como a imagem e uma identificação do nó a que a base de dados se encontra associada, permite distinguir as bases de dados que são de uso interno da InfluxDB e as bases de dados que guardam

informação de nós LoRaWAN e das restantes tecnologias de comunicação. Esta diferenciação prende-se com a forma como a *query* será executada. Sendo que, no caso de serem bases de dados internas não possuem qualquer tipo de ação relativa à análise de dados.

Assim sendo, todas as bases de dados que apareçam neste menu foram previamente configuradas no ficheiro de configuração (Figura 30). De seguida, cabe ao utilizador escolher qual a base de dados e medida a analisar, sendo que, é seguida a lógica representada no seguinte fluxograma (Figura 31):

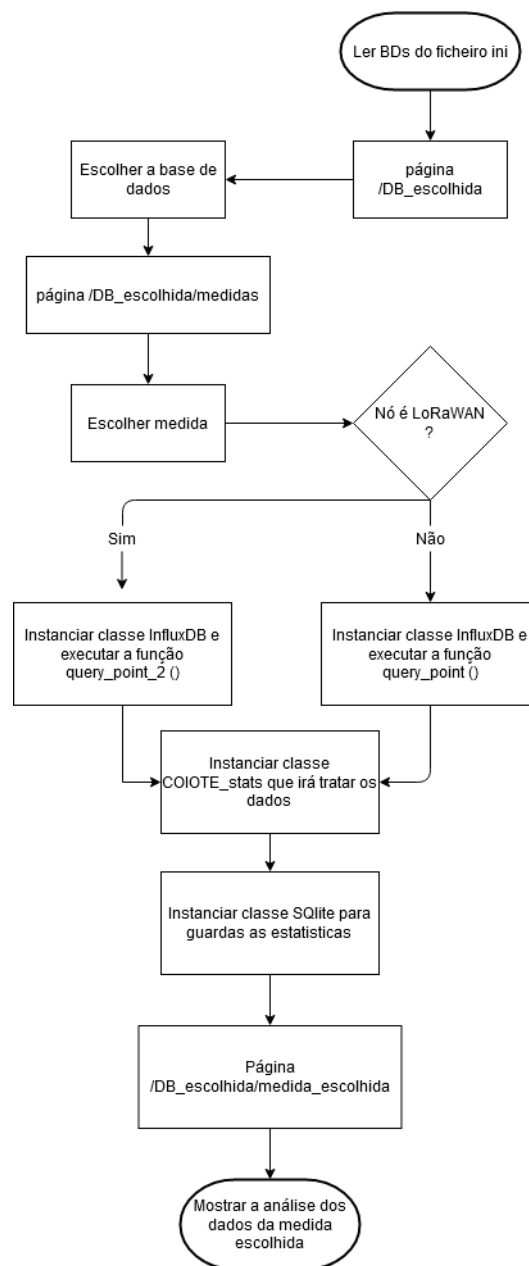


Figura 31 Processo para a análise de um parâmetro de um nó

De referir que, como se encontra demonstrado no fluxograma acima, a classe InfluxDB é instanciada e que, por sua vez, a função `query_point_2()` é executada ou a função `query_point ()`, consoante o tipo de nó. Esta função irá permitir que a plataforma analise uma medida de cada vez, portanto apenas faz query de um parâmetro para ser analisado.

Após estruturar os dados, estes serão tratados pela classe `COIOTE_stats`. Como foi anteriormente explicado, esta classe contém todas as funções responsáveis pela produção de estatísticas. Na parte de demonstração destas mesmas estatísticas criadas recorre-se à biblioteca `python pygal` [76], de forma a apresentar alguns dos resultados através de gráficos. Esta biblioteca, para além de ser de fácil integração com o `html` e o `flask`, oferece uma grande rapidez no processamento de toda a informação.

5.3.3. ALERTS

Esta secção permite gerir os alertas que estão a correr no script, demonstrado no subcapítulo de Visão Geral. Possibilitando assim adicionar e remover alertas, listar os diferentes alertas que estão configurados e as vezes que esses alertas foram despoletados. O fluxograma (Figura 32) abaixo demonstra a interação entre o servidor web e o script de alertas e de que forma é gerida a submissão de um dos dois formulários disponíveis, isto é, o formulário para adicionar alertas e o para remover.

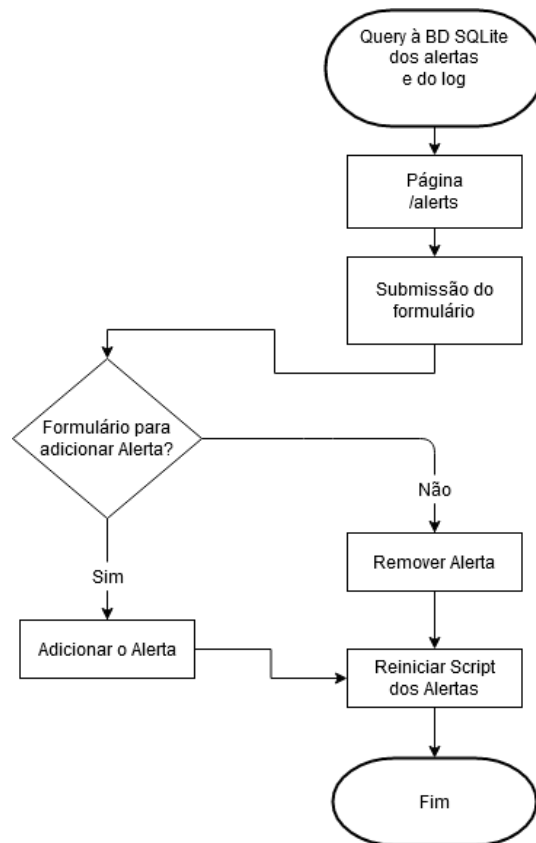


Figura 32 Interação entre o script de Alertas e a plataforma

5.3.4. SAMPLING RATE

Esta página oferece ao utilizador a possibilidade de modificar a taxa de amostragem dos nós que utilizem o protocolo MQTT, o que engloba os nós Wi-Fi e os ethernet, seguindo a logica descrita no seguinte fluxograma (Figura 33):

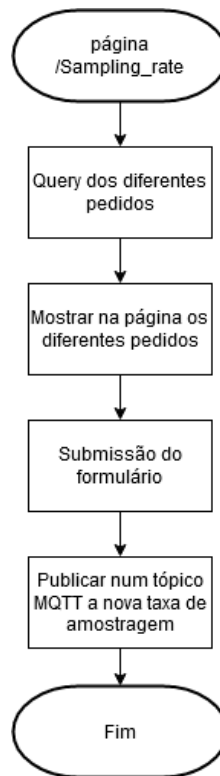


Figura 33 Função Sampling_Rate

Deste modo, de forma a que seja enviada a mensagem MQTT é necessário a utilização de um tópico predefinido que segue a seguinte nomenclatura `sampling_rate/<Nome_nó>` para além de publicarem as suas medidas é, ainda, necessário que os nós Wi-Fi e ethernet subscrevam simultaneamente este tópico.

5.3.5. CONFIG USER

Esta página oferece um simples formulário para o caso de o utilizador necessitar de modificar as suas credenciais ou algum dos seus dados. A lógica do mesmo encontra-se apresentada no fluxograma da Figura 34.

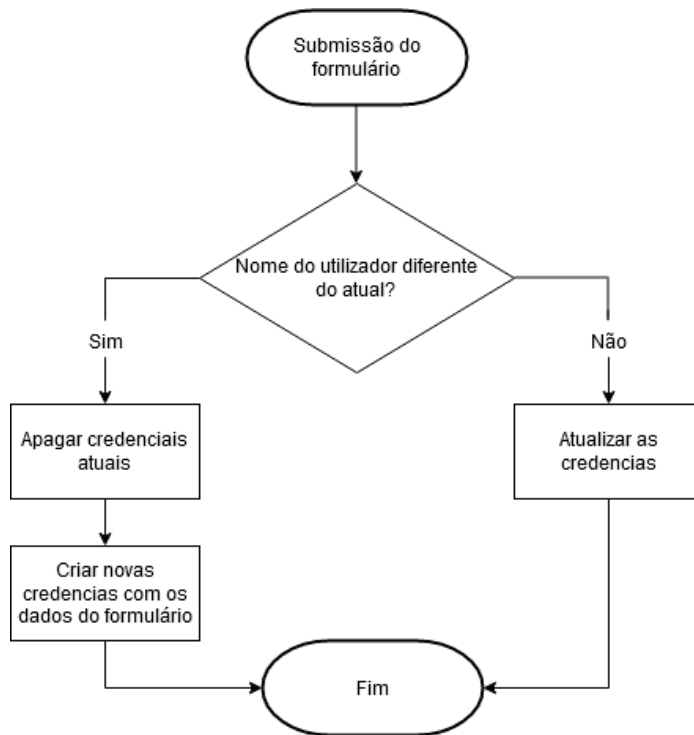


Figura 34 Atualização das credencias

5.3.6. APLICAÇÃO DE ALGORITMOS DE MACHINE LEARNING

Esta parte da aplicação *web* é referente à aplicação de diferentes algoritmos de *machine learning*. Estes algoritmos serão aplicados nos dois tipos de nós idealizados aquando a listagem de requisitos, sendo um de medição de parâmetros ambientais e outro de medição de energia.

O primeiro nó engloba parâmetros como a temperatura externa e interna do nó, a humidade externa e a concentração de metano. Neste nó considerou-se como variável dependente a concentração de metano e as restantes medidas como variáveis independentes. Tendo por base esta designação e com o objetivo de ser possível estimar o valor da concentração de metano em função dos restantes parâmetros, foram utilizados três algoritmos para efetuar a regressão: *Support Vector Machine (SVM)*, *Decision Tree* e *Random Forest*.

Antes de treinar estes algoritmos com as medidas armazenadas na DB, é necessário a organização da informação através da classe *InfluxDB*, e, posteriormente, transformar a

informação num *dataframe*, estrutura esta suportada pela biblioteca Pandas. O seguinte extrato de código representa o procedimento descrito:

```
SmartComposter= InfluxDB ("localhost",
"Smart_Composter", measures)

data = SmartComposter.query_points()
dataset = pandas.DataFrame(data)
dataset = dataset.drop("time", axis=1)
```

Importa referir que a coluna com os *timesteps* é excluída e, por isso, ficam apenas as das medidas. De seguida, tal como já foi referido, é necessário separar as variáveis independentes da depende e separar em dois *datasets*, um para treinar o algoritmo e outro para testar o algoritmo. Isto é conseguido através das seguintes linhas de código:

```
X = dataset.iloc[:, 0:2].values
y = dataset.iloc[:, -1].values

X_train,X_test,y_train,y_test = (X, y,
test_size = 0.2, random_state )
```

A variável X representa as variáveis independentes e a variável y a variável independente. Após serem separadas, estas variáveis são divididas em dois diferentes *datasets*. A variável `test_size` refere-se ao tamanho do *dataset* de teste, no caso demonstrado 0.2 significa que é usado 80% dos dados para treinar o algoritmo e 20% para o testar.

Seguidamente, é utilizado cada um dos três algoritmos para prever os valores de y , y_{pred} , para a variável X_{test} e é medido o desempenho de cada algoritmo recorrendo à métrica R^2 que irá comparar o valores previstos do X_{test} com os valores verdadeiros y_{test} . Será usado o algoritmo com maior R^2 .

Ainda alusivo a este nó, de modo a efetuar a classificação, foram utilizados os seguintes algoritmos: o modelo K-NN, o SVM, *Decision Tree Classification* e o *Random Forest Classification*.

As utilizações destes algoritmos pressupõem a divisão anterior das medidas em variáveis dependentes e independentes e da mesma organização da informação demonstrados nos extratos de código anteriores. Na aplicação destes algoritmos é crucial definir a condição relativa à concentração de metano que irá dividir o *dataset*.

De seguida é aplicado os diferentes algoritmos para prever se os valores de y pertencem ou não à condição anteriormente mencionada. Neste caso, a métrica de eficácia é a percentagem de acerto.

Por sua vez, o nó de energia mede as unidades relacionadas com o consumo energético, em que o objetivo recaiu em detetar diferentes perfis ou grupos de consumo. Tal como mencionado no estado de arte, para efetuar o *clustering* recorreu-se ao algoritmo *K-means*.

Como foi demonstrado no nó anterior, para executar este algoritmo é necessário proceder ao processamento dos dados, com a diferença de que a coluna que contém o *timestamp* das medidas passa a ser o *index* do *dataframe*. Posteriormente, o *dataframe* é organizado de modo a que o *index* represente apenas os diferentes dias e que haja colunas para cada hora do dia. Essas colunas são preenchidas pela respetiva potência média, referente a uma dada hora e a um dado dia. Este processo é consigo através do seguinte excerto de código:

```
dataset = dataset.set_index('time')
dataset.index = pd.to_datetime(dataset.index)
dataset = dataset.resample('H').sum()
dataset["hour"] = dataset.index.hour
dataset.index = dataset.index.date
dataset = dataset.pivot(columns='hour')
dataset = dataset.dropna()
dataset = dataset[~(dataset == 0).any(axis=1)]
dataset["Power"] = dataset["Power"].div(1000)
.round(3)
```

Após este processamento dos dados é necessário determinar o diferente número de *clusters* para treinar o algoritmo. Neste sentido, recorre-se ao método do cotovelo (Figura

35). O gráfico apresentado nesta figura representa a variação da distorção, tendo em conta o número de *clusters*. Neste exemplo específico, o valor ideal é 3 uma vez que é quando o declive começa a ter um decréscimo pouco significativo.

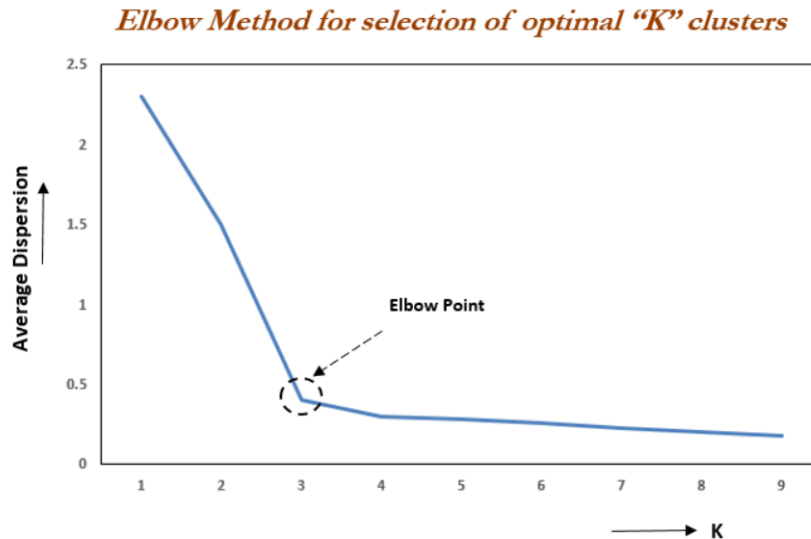


Figura 35 Exemplo de um gráfico representativo do método do cotovelo [77]

Após a atualização deste método é possível treinar o algoritmo *K-means* com o número de clusters ideal e segmentar os diferentes dias, tendo por base o consumo energético ao longo das diferentes horas do dia.

O fluxograma da Figura 36 apresenta, de uma forma sumariada, o modo de procedimento da plataforma previamente a treinar os algoritmos de *machine Learning* com a informação recolhida dos diferentes nós.

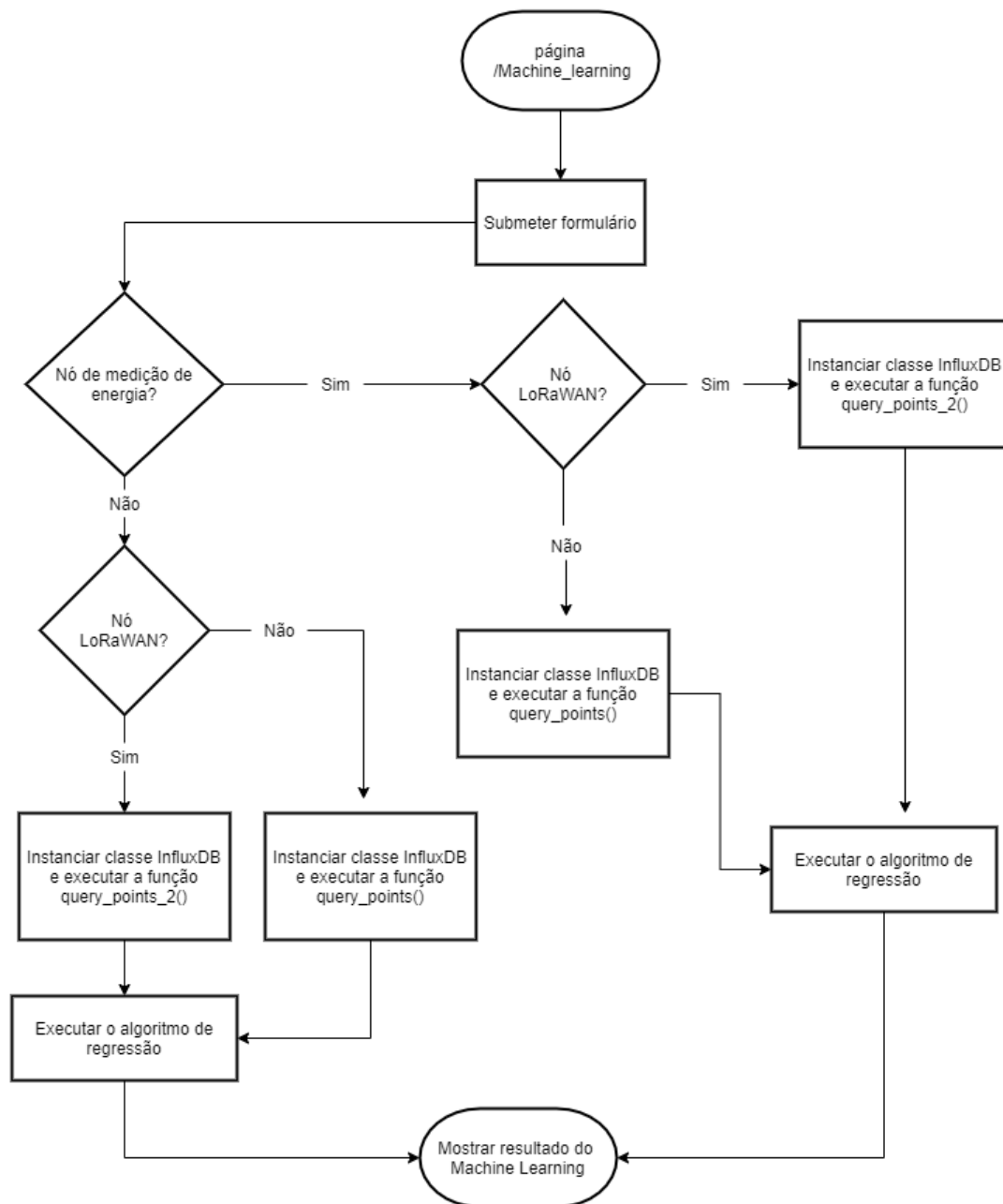


Figura 36 Escolha de um algoritmo de *Machine Learning*

6. APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS

Com o desígnio de mostrar o funcionamento da rede IoT e a aplicação web em cima dessa mesma rede, neste capítulo será criado um caso de estudo, onde serão ligados à rede nós que usem diferentes tecnologias e será explicado a sua integração e configuração do software da rede e posteriormente demonstrado os resultados.

6.1. DESCRIÇÃO DO CENÁRIO DE TESTE

Para o caso de estudo em questão foram então equacionados os seguintes nós:

- **Smart Composter** – este nó comunica exclusivamente por LoRaWAN, e unidireccionalmente, apenas no sentido de *uplink* (Figura 37). O conceito que originou este nó encontra-se relacionado com a sensorização de Compostagem, sendo que este sensor contempla quatro medidas: a temperatura interna e humidade interna do próprio esqueleto/carcaça do nó, a temperatura externa e a concentração de metano do ambiente em que o sensor seja inserido.



Figura 37 Nó *Smart Composter*

- **Energy Meter** – Este nó específico comunica simultaneamente por LoRaWAN e Wi-Fi (Figura 38), utilizando o microprocessador fipy (Figura 39) que, para além destes protocolos, permite ainda a comunicações por LTE, sigfox e *bluetooth*. Uma vez que a intenção deste nó é medir a energia, irá parametrizar as seguintes grandezas: tensão, frequência da rede, fator de potência, corrente na fase e corrente no neutro. Contrariamente ao nó descrito em cima, este permite também comunicação no sentido de *downlink*, permitindo a configuração da taxa de amostragem.

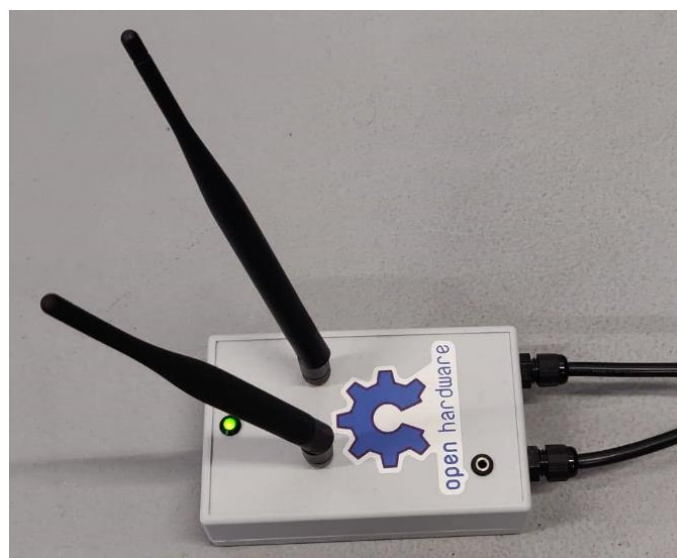


Figura 38 Nó *Energy Meter*



Figura 39 Microprocessador Fipy [78]

6.2. CONFIGURAÇÃO DOS NÓS NA REDE IOT

O nó *Smart Composter*, sendo um nó LoRaWAN, requer a configuração dos seguintes parâmetros na interface *web* do ChirpStack:

- **Gateway** – Neste passo é necessário registrar a *gateway* desenvolvida no servidor ChirpStack. Na Figura 40 é possível observar as configurações efetuadas, sendo de salientar que o parâmetro crucial da configuração é o *gateway ID*. Este é uma identificação única do concentrador de pacotes LoRaWAN usado na *gateway*.

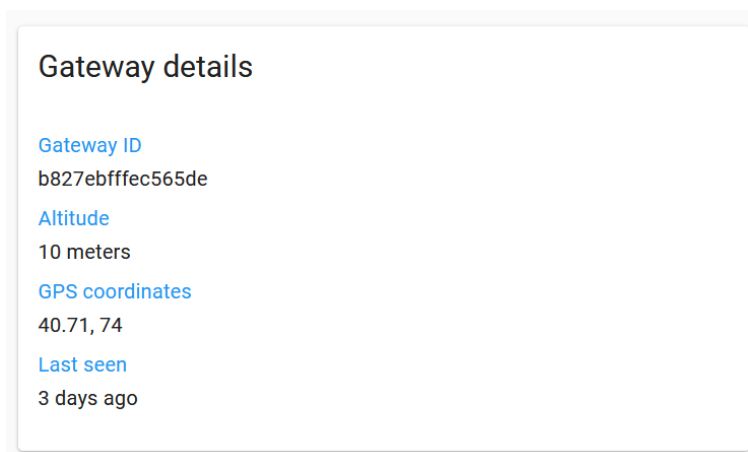


Figura 40 Configuração da *gateway*

Perfil do dispositivo – Na Figura 41 é apresentada a configuração do dispositivo LoRaWAN. Como é possível observar, são configurados os parâmetros regionais do LoRaWAN e o modo de ativação do dispositivo.

The screenshot shows the 'GENERAL' configuration tab for a LoRaWAN device. The fields and their values are as follows:

- Device-profile name ***: FIPy
- LoRaWAN MAC version ***: 1.0.2
- LoRaWAN Regional Parameters revision ***: A
- Max EIRP ***: 0
- Geolocation buffer TTL (seconds)**: 0
- Geolocation minimum buffer size**: 0

Figura 41 Configurações do dispositivo

- **Aplicação** – A adição de uma aplicação tem associada a *gateway* e perfil de dispositivo anteriormente configurados, bem como, o DevAddr, a NwkSKey e o AppSKey, responsáveis pelo método de ativação ABP.

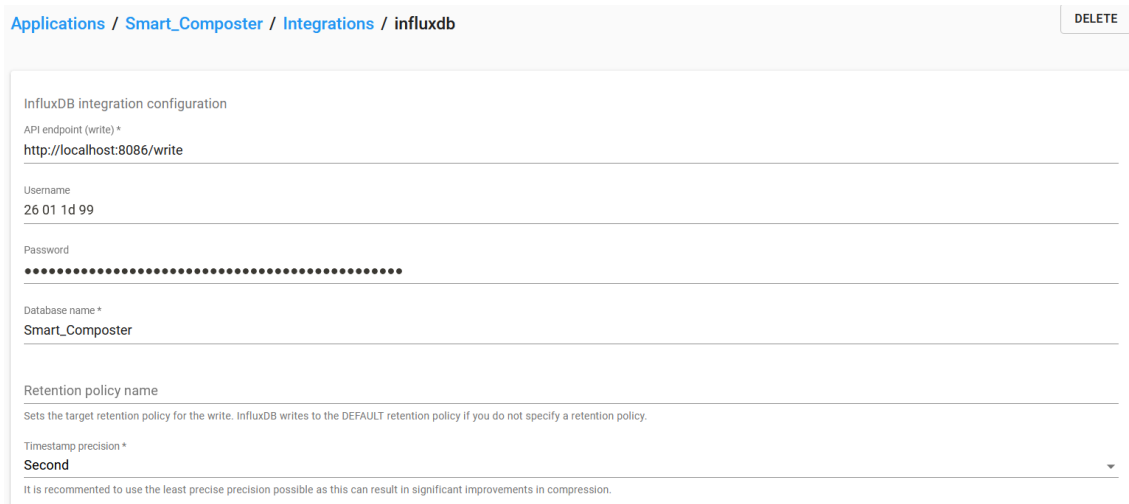
The screenshot shows the 'ACTIVATION' configuration tab for an application. The fields and their values are as follows:

- Device address ***: 26 01 1d 99
- Network session key (LoRaWAN 1.0) ***: 7f 03 c1 28 d6 3c 18 20 b0 1c 05 ad 43 13 a7 31
- Application session key (LoRaWAN 1.0) ***: 10 00 00 00 00 00 00 00 00 00 00 00 00 00 01
- Uplink frame-counter ***: 3330
- Downlink frame-counter (network) ***: 61690

At the bottom right, there is a button labeled '(RE)ACTIVATE DEVICE'.

Figura 42 Configurações da aplicação

- **Ligação ao InfluxDB** – Após a realização dos 3 passos anteriores, os dados provenientes do nó *Smart Composter* começam a ser descriptados e armazenados no servidor ChirpStack. Por fim, de modo a que a informação seja armazenada no InfluxDB, é configurada uma integração na qual se estabelece uma BD pertencente ao InfluxDB para onde os dados da atual aplicação serão enviados.



Applications / Smart_Composter / Integrations / influxdb DELETE

InfluxDB integration configuration

API endpoint (write) *

http://localhost:8086/write

Username

26 01 1d 99

Password

.....

Database name *

Smart_Composter

Retention policy name

Sets the target retention policy for the write. InfluxDB writes to the DEFAULT retention policy if you do not specify a retention policy.

Timestamp precision *

Second

It is recommended to use the least precise precision possible as this can result in significant improvements in compression.

Figura 43 Configurações da integração com o InfluxDB

Em relação ao nó *Energy Meter*, a configuração referente à comunicação LoRaWAN é idêntica à que foi anteriormente apresentada diferindo apenas na configuração da aplicação na qual serão usadas outras chaves específicas a este nó.

No que diz respeito à comunicação Wi-Fi, foi definido o tópico `Coiote_nodes/energy_meter/rx` no qual a trama, no formato *command line protocol*, é enviada.

De modo a receber e armazenar, numa BD do InfluxDB, a informação proveniente deste tópico, é necessário que seja configurado como *input plugin* o protocolo MQTT e como *output plugin* o próprio InfluxDB. Na Figura 44 é possível observar o excerto do ficheiro de configuração do Telegraf responsável por configurar estes dois *plugins*.

```

# Configuration for inserver to send metrics to
[[outputs.influxdb]]
# urls = ["udp://localhost:8089"] # UDP endpoint example
urls = ["http://localhost:8086"] # required
## The target database for metrics (telegraf will create it if not exists).
database = "COIOTE_Nodes_MQTT" # required
retention_policy = ""
write_consistency = "any"
timeout = "5s"

# Read metrics from MQTT topic(s)
[[inputs.mqtt_consumer]]
servers = ["tcp://localhost:1883"]
## MQTT QoS, must be 0, 1, or 2
qos = 0

## Topics to subscribe to
topics = [
  "coiote_nodes/#",
]

persistent_session = false
# If empty, a random client ID will be generated.
client_id = ""

## username and password to connect MQTT server.
username = "bl"
password = "Coiote_bl"
insecure_skip_verify = true

## Data format to consume.
data_format = "influx"

```

Figura 44 Configuração do Telegraf para o protocolo MQTT

Tendo em conta o que foi mencionado no subcapítulo sobre o protocolo MQTT, toda a informação enviada pelos nós que publiquem num tópico em que a primeira partição seja `coiote_nodes` será armazenada e estruturada na DB `COIOTE_Nodes_MQTT`.

No que diz respeito à aplicação web, foram configuradas 5 BDs no ficheiro `databases.ini` como é ilustrado na Figura 45, sendo que duas delas, a `_internal` e a `Coiote_System`, são bases de dados internas ao InfluxDB e por isso o seu `type` corresponde a *none*.

A BD *Smart_Composter* é relativa ao nó com o mesmo nome, enquanto que o nó *Energy_Meter* possui duas BDs uma para o protocolo LoRaWAN e outra para o Wi-Fi.

```

# databases info
[_internal]
text = Internal database created automatically
image = ../static/images/cloud-server.png
type = none

[Coiole_System]
text = Database with infomation displayed in the Home page
image = ../static/images/cloud-server.png
type = none

[Smart_Composter]
text = Sensor that measures environmental parameters
image = ../static/images/Smart_Composter.PNG
type = lora

[Energy_Meter]
text = Energy meter of a workbench using LoRaWAN
image = ../static/images/cloud-server.png
type = lora

[Energy_Meter-1]
text = Energy meter of a workbench using Wi-Fi
image = ../static/images/cloud-server.png
type = other

```

Figura 45 Configuração do ficheiro databases.ini

6.3. MANUAL DE UTILIZAÇÃO E RESULTADOS DO CENÁRIO DE TESTE

Nas secções seguintes são descritas as diferentes funcionalidades da aplicação do ponto de vista do utilizador

6.3.1. HOME

Assim que se acede ao URL da aplicação *Web*, aparece o formulário de *login* (Figura 46), onde o utilizador poderá digitar as credencias de acesso.



Figura 46 Login

Após a submissão do formulário de *login* o utilizador é redirecionado para a página *home* da plataforma. Esta, como se vê na Figura 47, apresenta diversos parâmetros do computador em que o servidor da rede IoT está a correr.

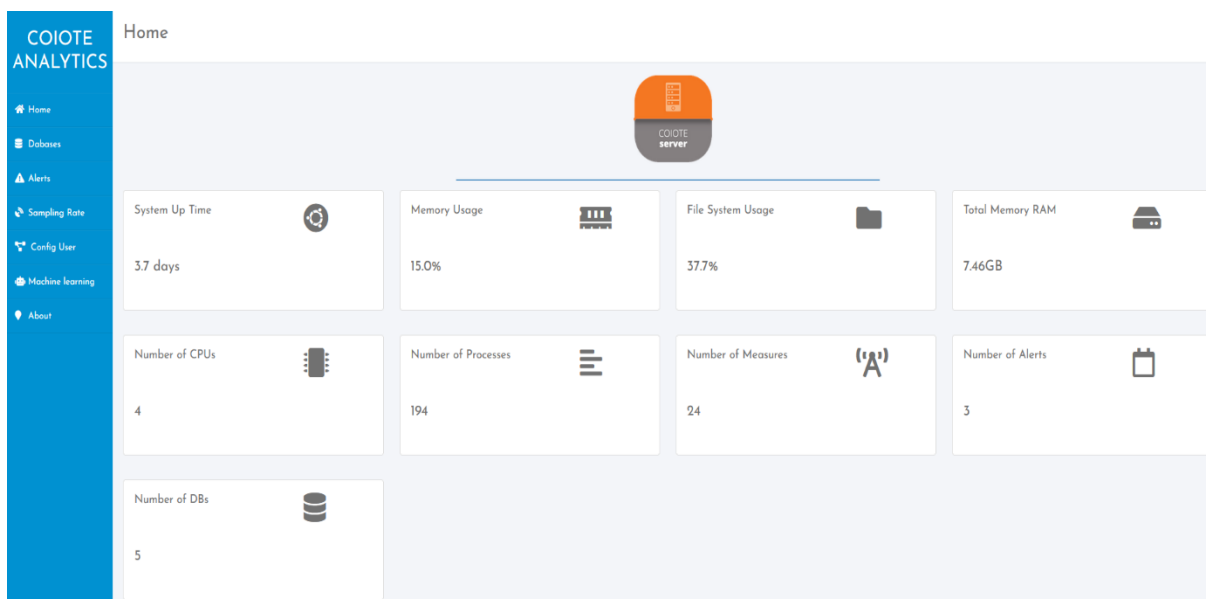


Figura 47 Home

6.3.2. DATABASES

O menu *Databases* apresenta a listagem das BDs configuradas na secção anterior deste capítulo. Apenas três delas tem a opção de “Check Stats” visto que são BDs diretamente

associadas a nós, o que permite visualizar as estatísticas dos diferentes parâmetros que constituem esse nó.

Databases

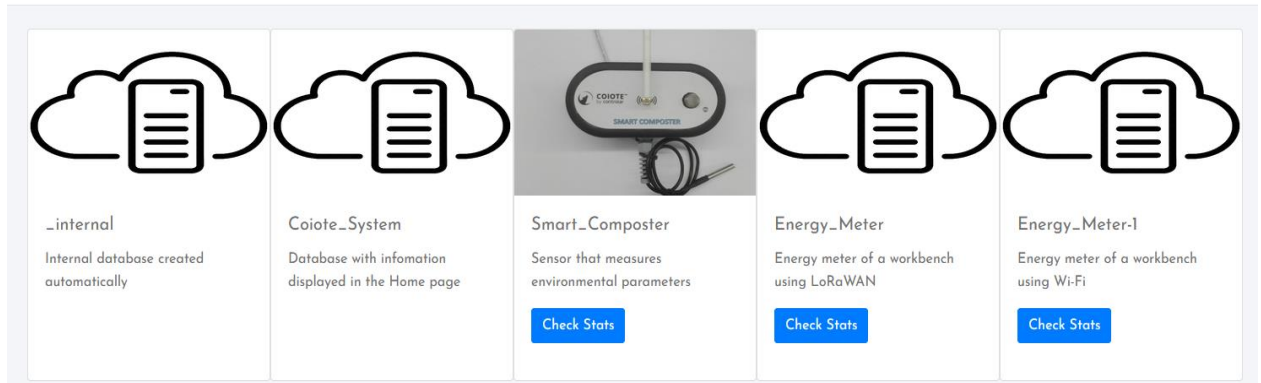


Figura 48 Menu *Databases*

Pressionando, por exemplo, no botão “Check Stats” da DB *Energy_Meter-1*, irá aparecer o menu da Figura 49 que apresenta as diferentes medidas armazenadas na BD selecionada. A plataforma irá realizar, à medida escolhida neste menu, a análise de dados e irá apresentar diferentes estatísticas sobre a mesma.

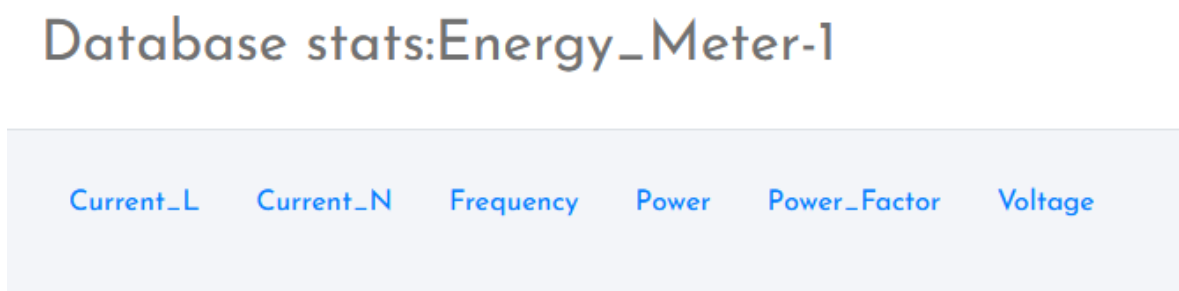


Figura 49 Medidas armazenadas na BD *Energy_Meter-1*

A opção *Power* permite aceder a uma página composta por uma tabela com estatísticas e diversos gráficos com informações sobre a medida em questão.

A tabela da Figura 50 apresenta diversas estatísticas agrupadas por três intervalos de tempo diferentes: o tempo total em que o sensor está conectado à plataforma, o mês e a semana anterior. Através desta análise é possível verificar que no momento desta análise

este nó não emitiu nenhum ponto na semana anterior. É ainda possível através da variável total_time saber há quanto tempo a BD armazena dados deste parâmetro.

Database stats:Energy_Meter-1

Power

total_time:67 days, 23:24:26

#	Total Time	Last Month	Last Week
Number of Points	403565	138305	0
offset Points	19160	104	0
Max	637.0	368.0	nan
Mean	69.411	56.732	nan
Min	0.0	0.0	nan
std	72.97	28.59	nan

Figura 50 Tabela de estatísticas da potência

O gráfico da Figura 51 apresenta uma comparação entre a percentagem de tempo em que o nó se encontrou em funcionamento e a percentagem de tempo em que não enviou informação para o servidor, desde o momento em que este foi integrado na plataforma. Ainda de referir que os gráficos mostrados por esta página de análise de dados, são todos interativos, ou seja, o utilizar passando o cursor por cima de uma determinada parcela consegue saber o seu valor.

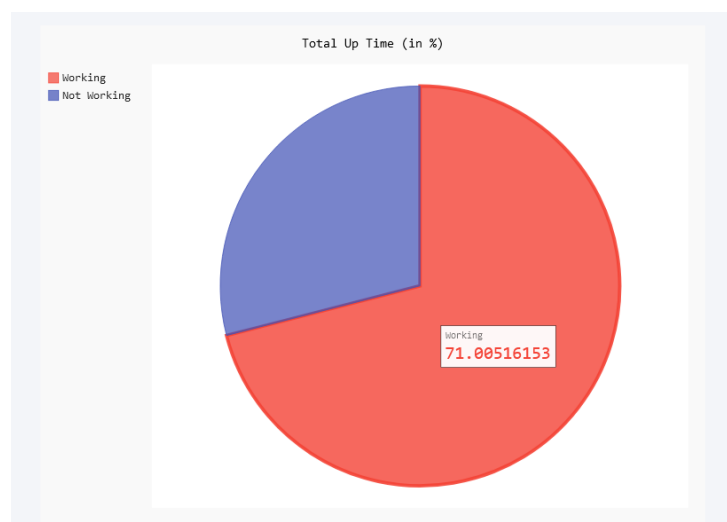


Figura 51 Gráfico comparativo do tempo que o nó Energy_Meter-1 se encontrou em funcionamento

Os restantes gráficos apresentados na nesta página são referentes à media mensal (Figura 52), à media diária do últimos mês (Figura 53) e por fim à media diária do tempo total em que este nó enviou dados (Figura 54).

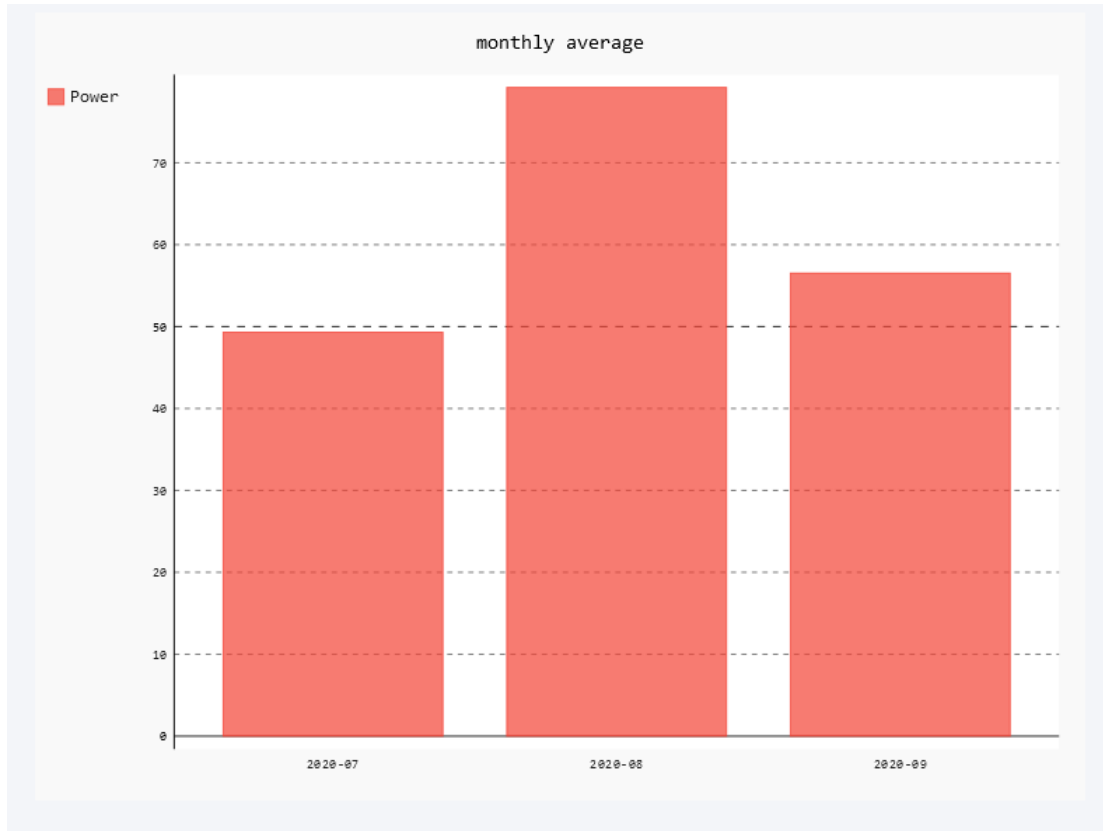


Figura 52 Média mensal da potência

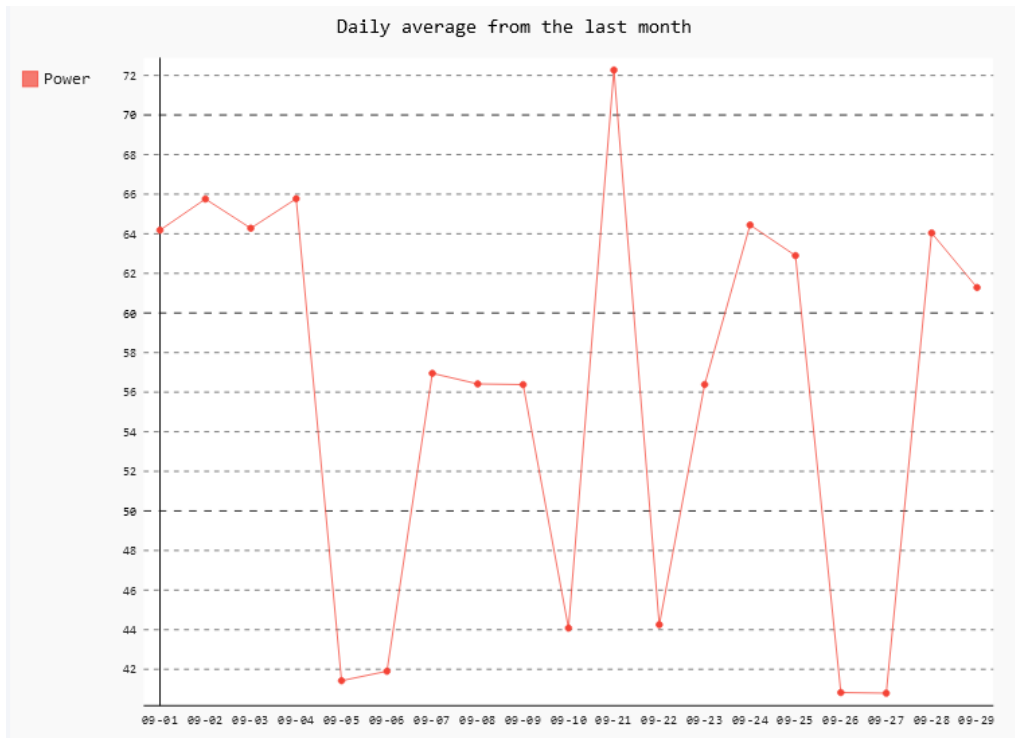


Figura 53 Média diária da potência no último mês

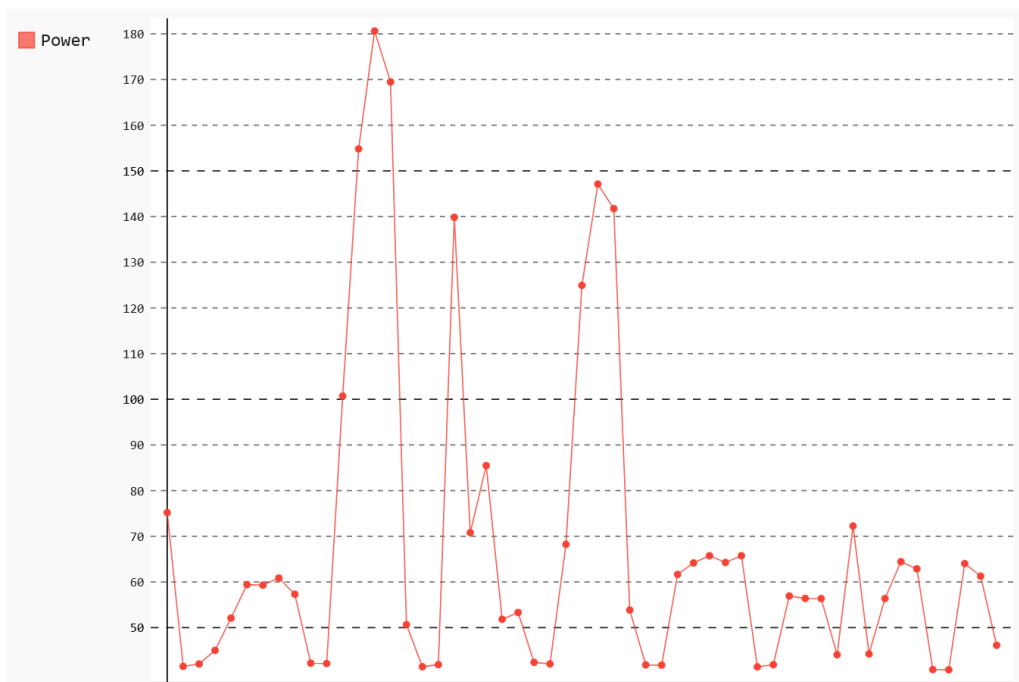


Figura 54 Média diária da potência durante o tempo total

Deste modo, é possível observar a análise anteriormente demonstrada para qualquer parâmetro presente nas BDs, que armazenem informação proveniente dos nós. Assim sendo, recuando até à lista das DBs presente na Figura 48 e escolhendo inicialmente o BD

Smart_Composter e depois o parâmetro Methane é possível observar a análise que será discutida de seguida.

Na tabela da Figura 55 é possível ver as estatísticas referentes à concentração de metano. É possível observar que este parâmetro se encontra designado como device_frmpayload_data_illumiance_sensor_3, causa do ID do protocolo Cayenne LPP escolhido para a medida e devido à nomenclatura do servidor ChirpStack.

Database stats:Smart_Composter
device_frmpayload_data_illumiance_sensor_3
total_time:295 days, 3:52:50

#	Total Time	Last Month	last Week
Number of Points	61698	4213	1624
offset Points	344	15	0
Max	4723	2367	69
Mean	30.01	39.42	23.768
Min	0	9	14
std	30.85	40.31	10.55

Figura 55 Tabela de estatísticas da concentração de metano

A página apresenta ainda, de forma semelhante à análise da potência, os seguintes gráficos:

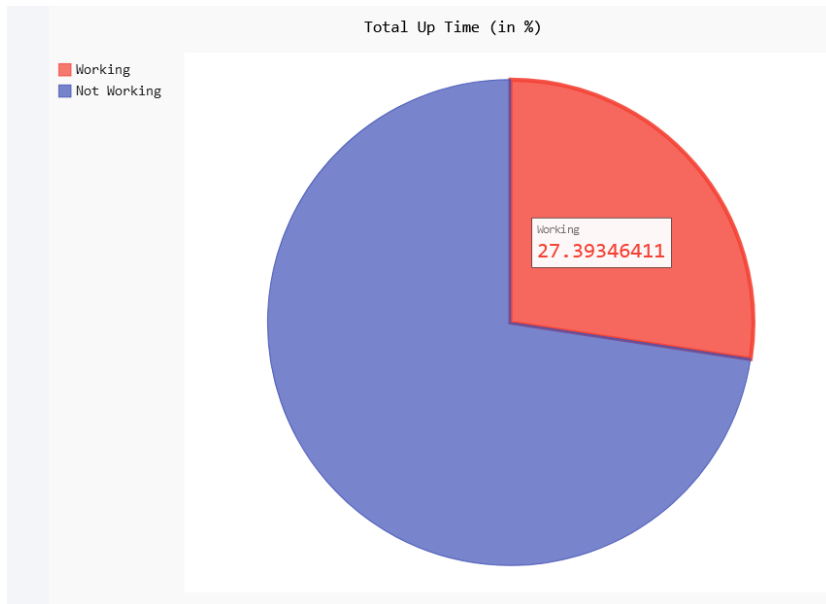


Figura 56 Gráfico comparativo do tempo que o nó Smart_Composter se encontrou em funcionamento

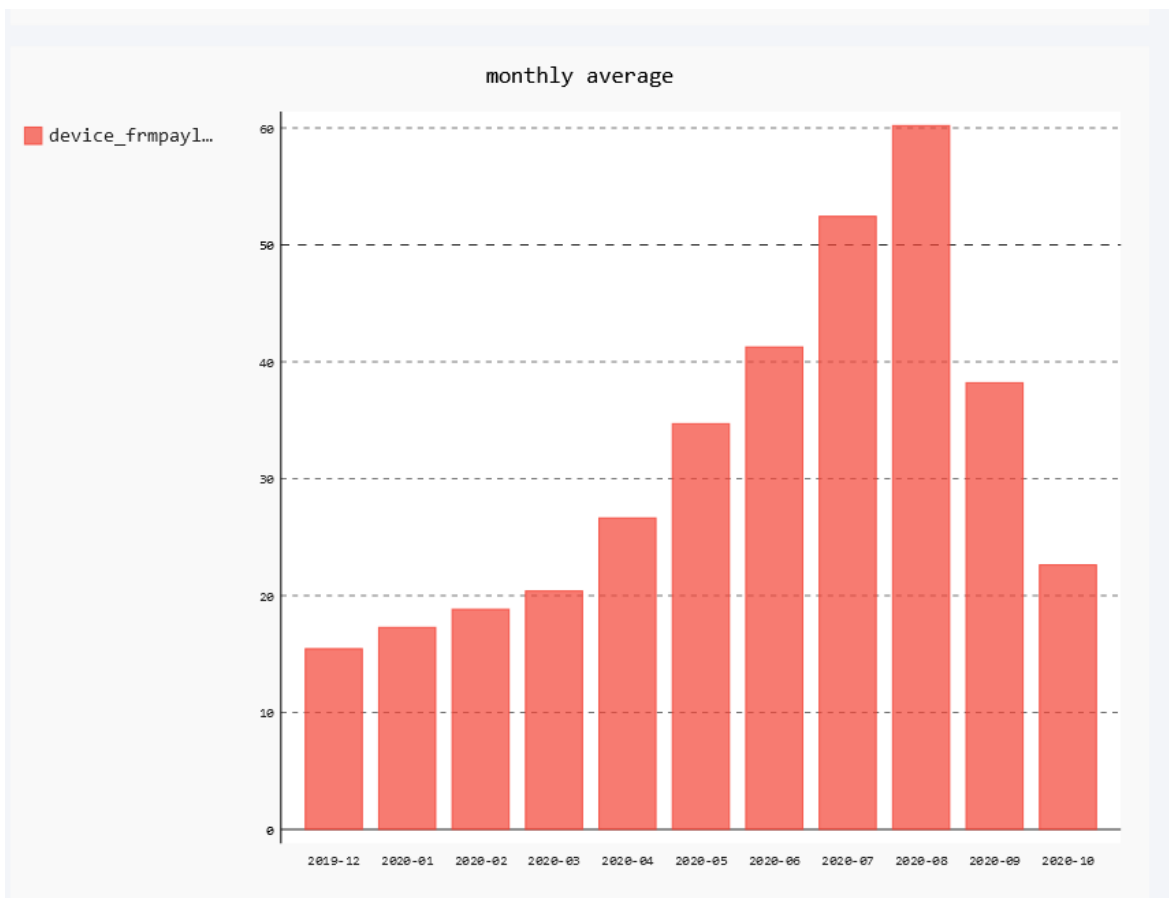


Figura 57 Média mensal da concentração de metano

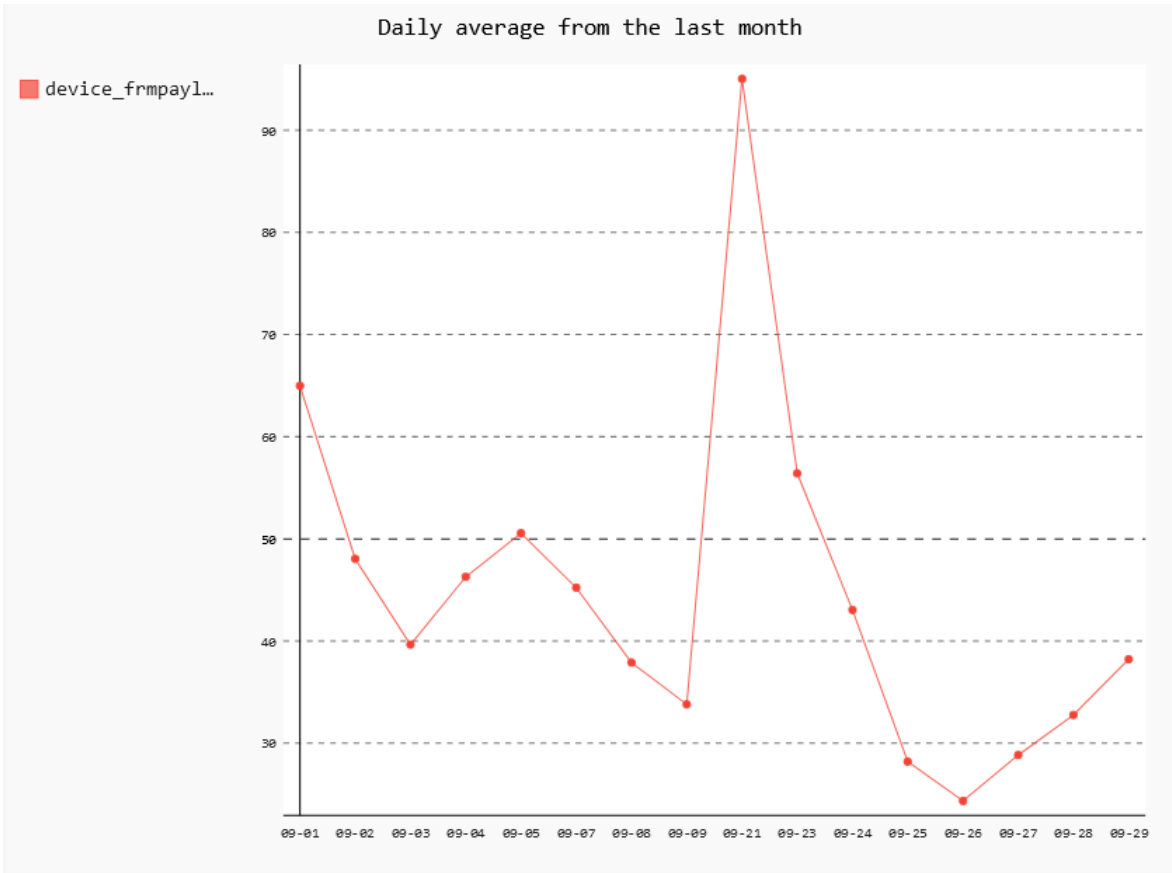


Figura 58 Média diária da concentração de metano no último mês

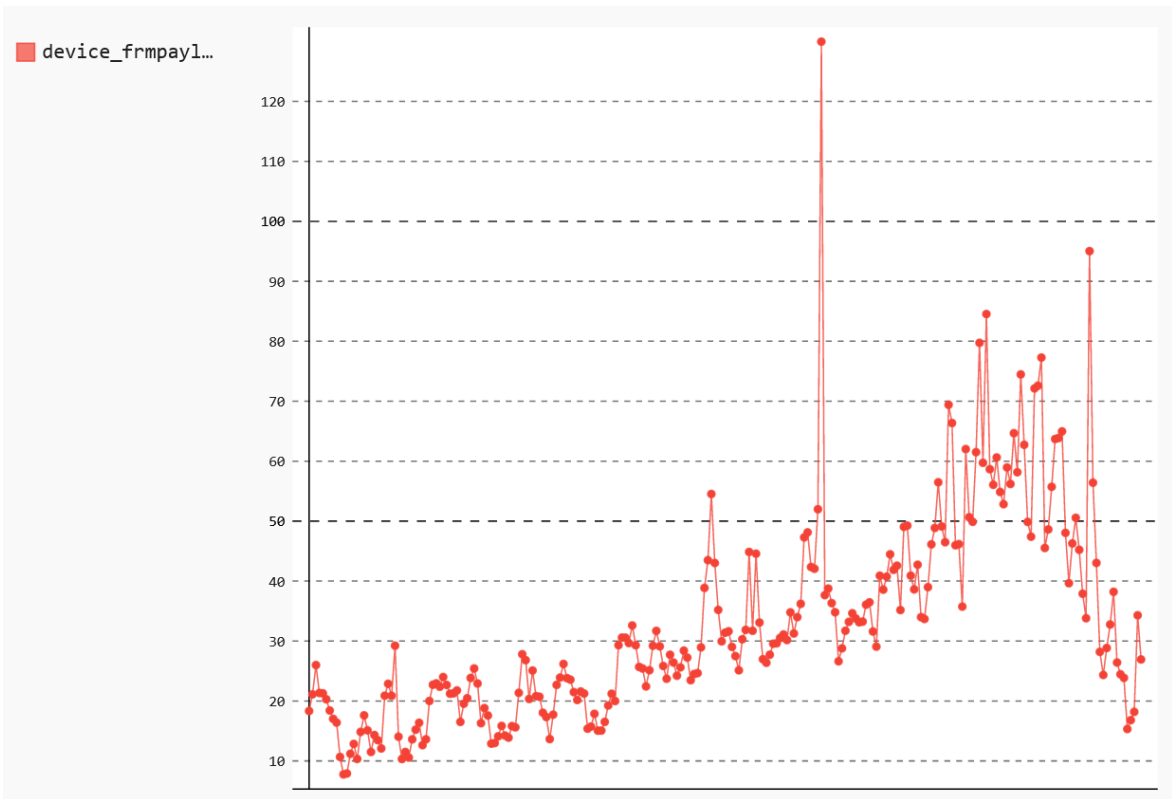
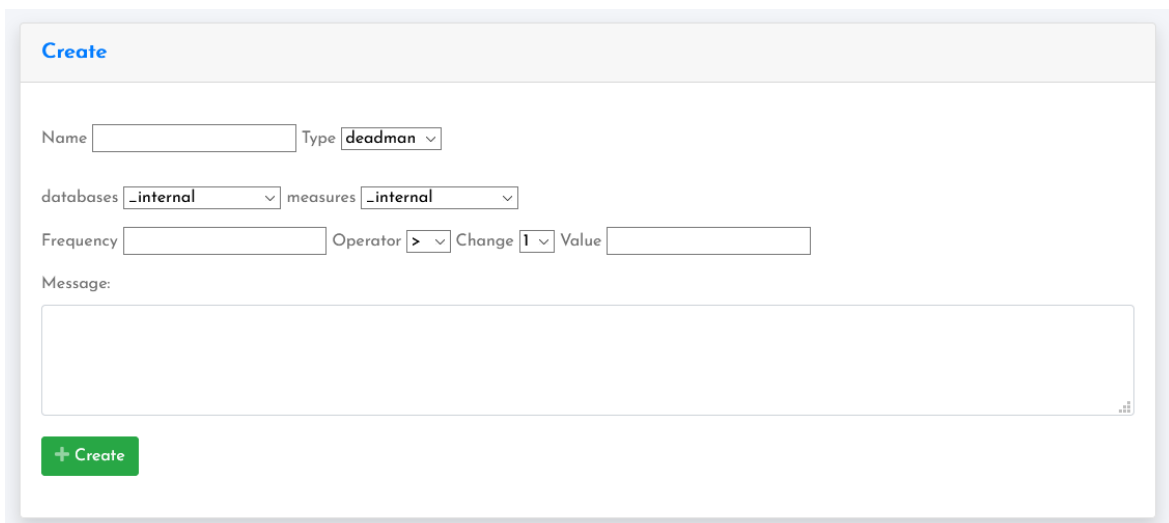


Figura 59 Média diária da concentração de metano durante o tempo total

6.3.3. ALERTS

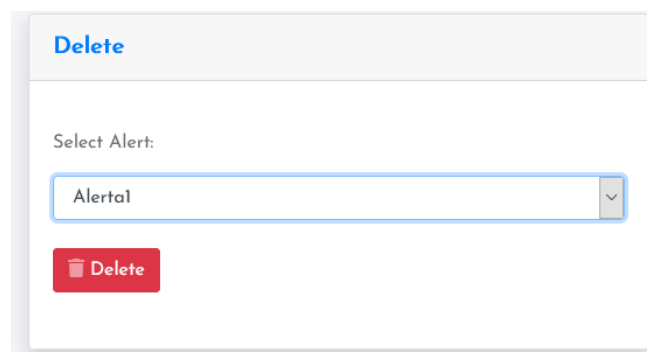
No menu de alertas, através do formulário da Figura 60, o utilizador consegue criar alertas e configurá-los. Permite ainda associar uma mensagem ao alerta que vai ser criado de forma a que sempre que este seja despoletado apareça associado a esse aviso a mensagem de forma a melhor identificar a razão do aviso. Por sua vez, o formulário da Figura 61 possibilita a eliminação de qualquer um dos alertas configurados.



The screenshot shows a web form titled "Create" for configuring an alert. It includes the following fields and controls:

- Name:** A text input field.
- Type:** A dropdown menu currently set to "deadman".
- databases:** A dropdown menu set to "_internal".
- measures:** A dropdown menu set to "_internal".
- Frequency:** A text input field.
- Operator:** A dropdown menu set to ">".
- Change:** A dropdown menu set to "1".
- Value:** A text input field.
- Message:** A large text area for entering a message.
- + Create:** A green button to submit the form.

Figura 60 Menu para criar um Alerta



The screenshot shows a web form titled "Delete" for removing an alert. It includes the following elements:

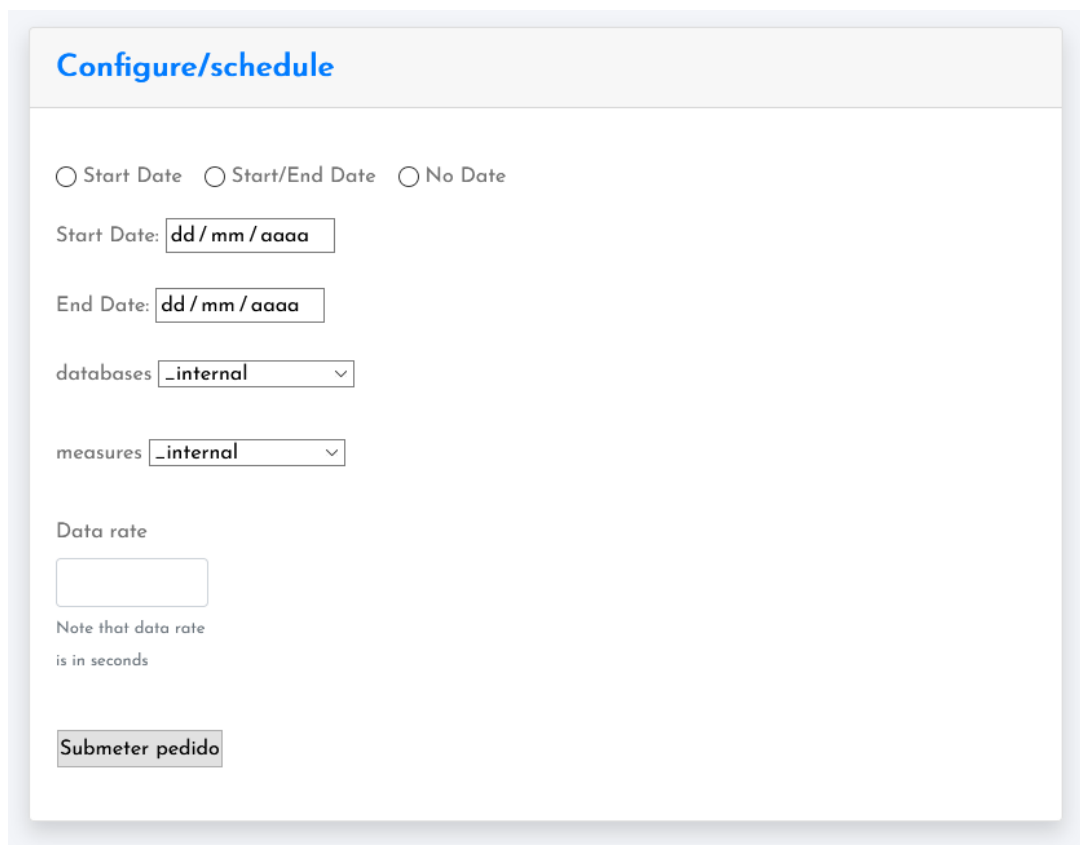
- Select Alert:** A label above a dropdown menu.
- Alerta:** The selected item in the dropdown menu.
- Delete:** A red button with a trash icon to confirm the deletion.

Figura 61 Menu para apagar um alerta

Caso algum dos alertas configurados desencadeie um aviso, é também nesta página que esses avisos surgem acompanhados da informação sobre o momento em que aconteceram e da mensagem configurada para esse determinado alerta.

6.3.4. SAMPLING RATE

Para enviar/alterar as taxas de amostragens de sensores integrados na plataforma, que façam uso do protocolo MQTT, o utilizador deve submeter um pedido de alteração através do formulário da Figura 62.



The screenshot shows a web form titled "Configure/schedule". At the top, there are three radio buttons: "Start Date", "Start/End Date", and "No Date". Below these are two date input fields: "Start Date: dd/mm/yyyy" and "End Date: dd/mm/yyyy". There are two dropdown menus: "databases" with "_internal" selected and "measures" with "_internal" selected. A "Data rate" input field is present, followed by the text "Note that data rate is in seconds". At the bottom, there is a "Submeter pedido" button.

Figura 62 Menu para enviar um pedido de alteração da taxa de amostragem

Esta página, para além do menu para configurar os pedidos, contém ainda uma listagem com o histórico dos diferentes pedidos de alteração de amostragem, agrupados pelas medidas dos diferentes nós (Figura 63).

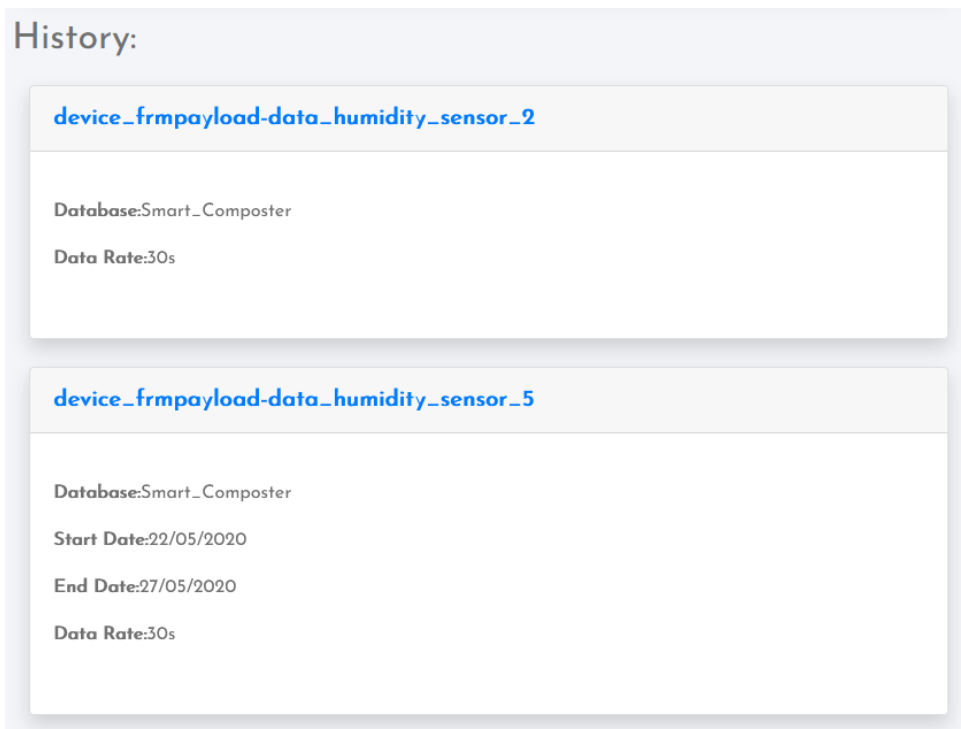


Figura 63 Histórico dos diferentes pedidos

6.3.5. CONFIG USER

Esta funcionalidade permite, através do preenchimento do formulário presente na Figura 64, atualizar as credenciais de acesso à aplicação bem como alguns dados relativos ao utilizador.

Config User

Update Credencials

username e-mail

password Full Name

Figura 64 Formulário para atualizar as credencias

6.3.6. APLICAÇÃO DE ALGORITMOS DE *MACHINE LEARNING*

No menu onde se configura a aplicação dos algoritmos de *machine learning* é primeiro dada a hipótese ao utilizar de escolher entre o nó *Smart_Composter* e o *Energy_Meter* ().

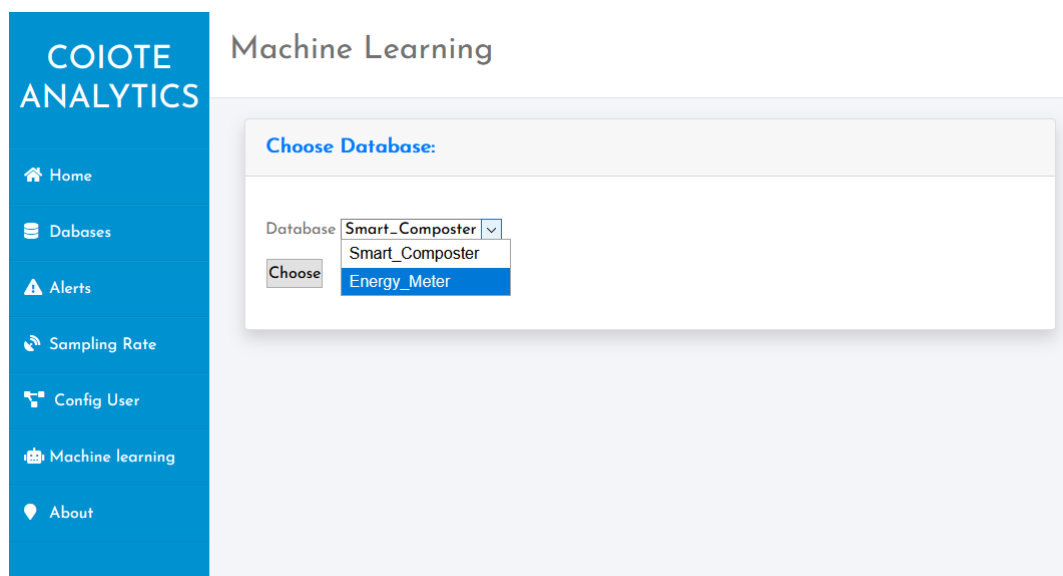


Figura 65 Menu de escolha da BD a aplicar os algoritmos de *Machine Learning*

6.3.6.1. ENERGY_METER

Escolhendo o nó *Energy_Meter* não é necessário mais nenhuma tipo configurações uma vez é automaticamente aplicado o algoritmo de K-means de modo a efetuar o *clustering* dos diferentes perfis diários de consumo energético.

Assim sendo, são apresentados dois gráficos ilustrados na Figura 66 e na Figura 67. O primeiro mostra o método do cotovelo aplicado ao nó *Energy_Meter*, através do resultado obtido é possível afirmar que 3 clusters é o número ideal uma vez que o declive a partir do número 3 começa a atenuar. O segundo gráfico representa os diferentes perfis de consumo diários descobertos pelo algoritmo, e à direita encontra-se representado os dias que correspondem a cada dos diferentes perfis.

Uma vez que este nó se encontrava a medir o consumo energético de uma bancada onde se encontram maioritariamente duas pessoas, o perfil a azul conclui-se que representa os dias em que apenas uma das pessoas se encontra a trabalhar. O perfil a vermelho está

relacionado com a situação de duas pessoas a trabalhar, sendo possível observar um pequeno aumento do consumo energético. O perfil verde corresponde aos dias em que foi também ligada a esta bancada uma máquina em desenvolvimento, responsável por um aumento substancial do consumo energético.

Clustering of energy consumption using Energy_Meter DB

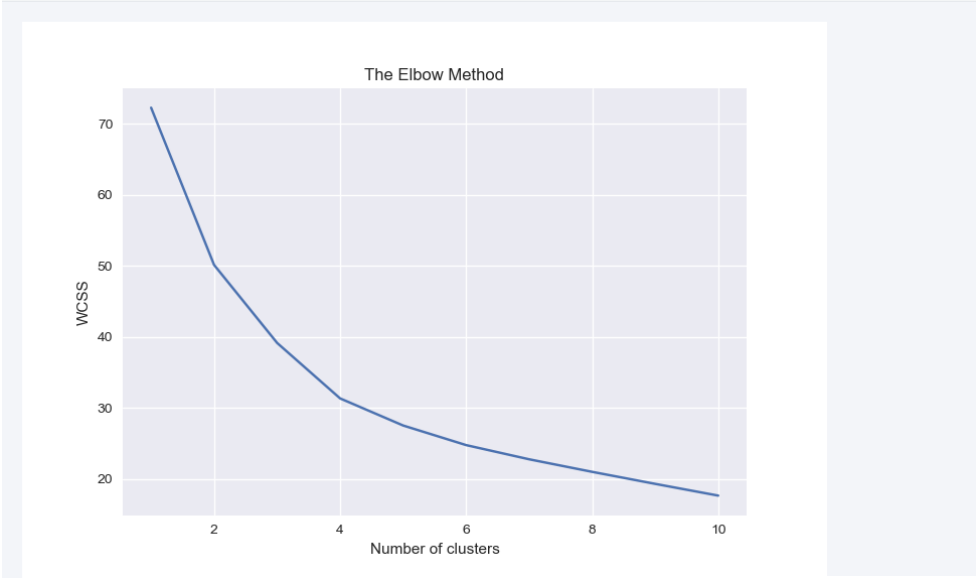


Figura 66 Método do cotovelo aplicado aos dados do Energy_Meter

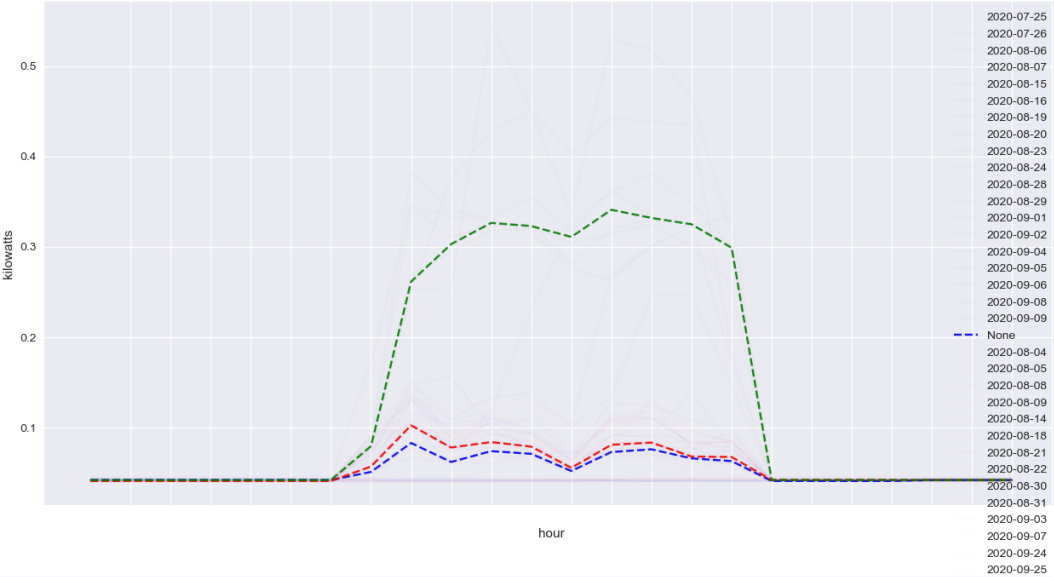


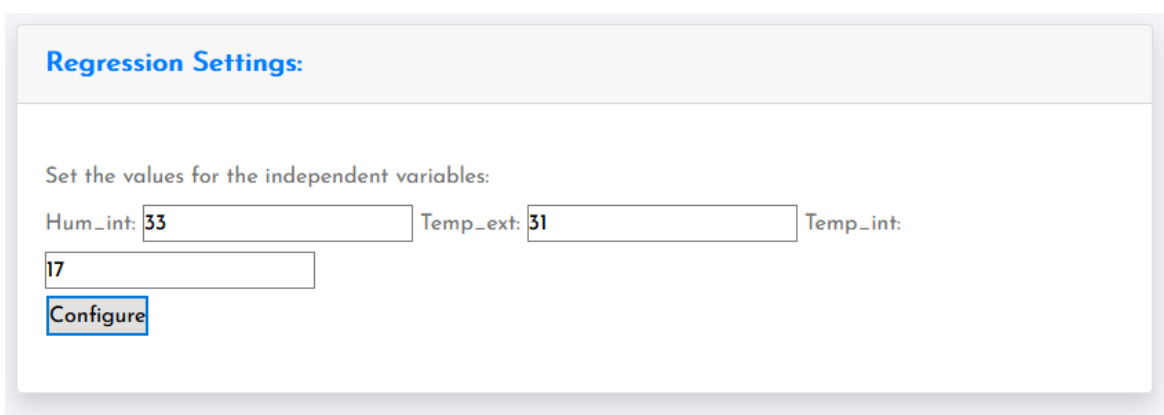
Figura 67 Diferentes perfis energéticos criados pelo algoritmo

6.3.6.2. SMART_COMPOSTER

Escolhendo o nó Smart_Composter, é dada ao utilizador a possibilidade de escolher entre aplicar algoritmos de regressão ou de classificação. De modo a apresentar o resultado destes algoritmos foi considerado a seguinte medição que foi enviada por este nó:

Temperatura interna: 17; Temperatura externa =31; Humidade Interna =33; Metano = 21

Inicialmente é configurado o formulário presente na Figura 68, com os valores acima mencionados para comparar o valor obtido da concentração de metano face ao valor real.



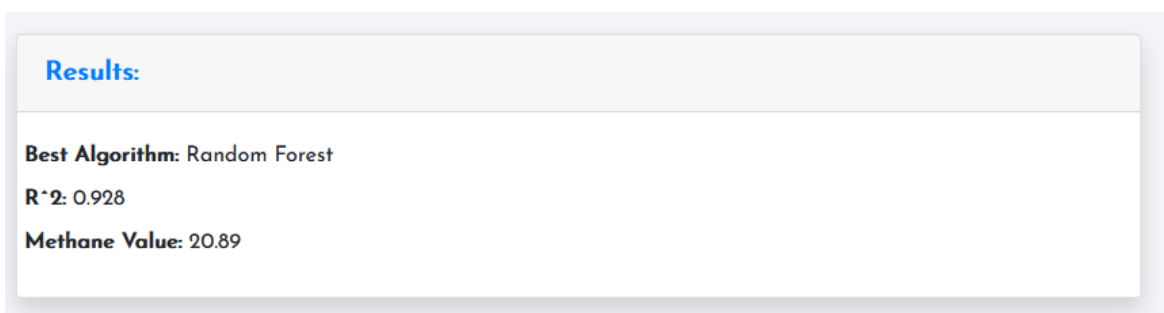
Regression Settings:

Set the values for the independent variables:

Hum_int: Temp_ext: Temp_int:

Figura 68 Formulário para configurar a regressão

A Figura 69 mostra o resultado da aplicação da regressão. É possível concluir que o algoritmo mais eficaz foi o *Random Forest* e que para os valores configurados previu uma concentração de metano com o valor 20.89, valor muito próximo do real.



Results:

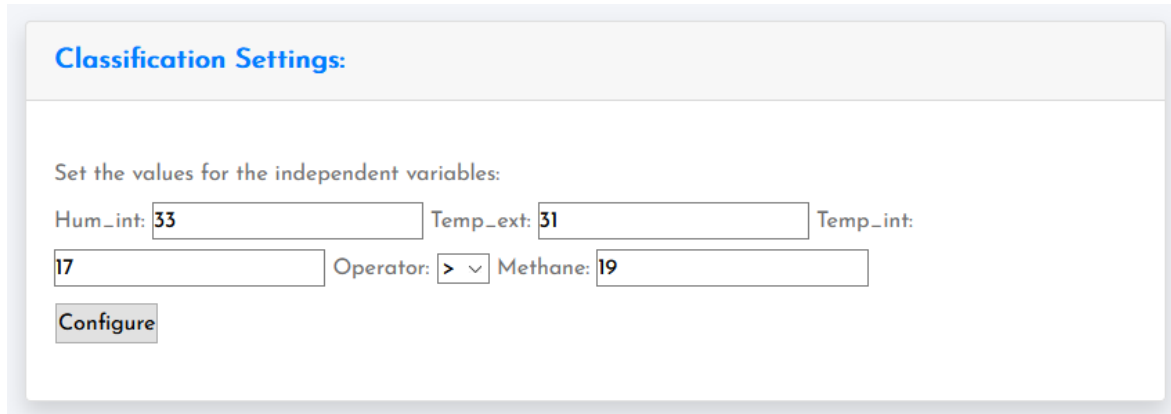
Best Algorithm: Random Forest

R²: 0.928

Methane Value: 20.89

Figura 69 Resultado da regressão

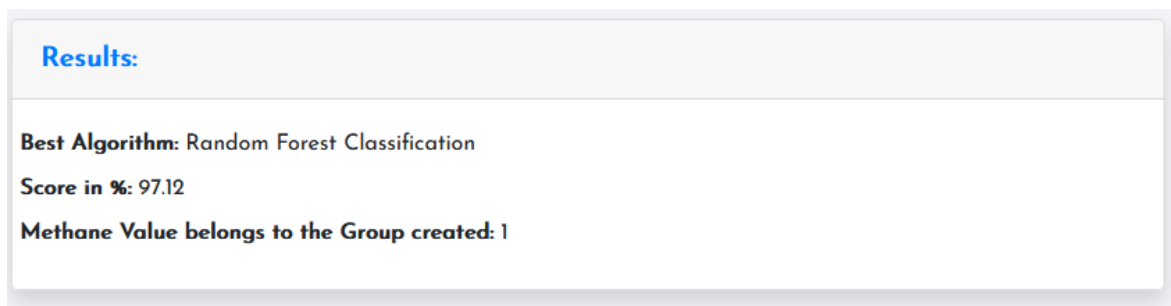
Para configurar a classificação é necessário configurar o formulário presente na Figura 70. Neste queremos avaliar a possibilidade de o valor da concentração de metano correspondente aos restantes valores é superior a 19ppm.



The screenshot shows a web form titled "Classification Settings". Below the title, it says "Set the values for the independent variables:". There are four input fields: "Hum_int:" with the value "33", "Temp_ext:" with the value "31", "Temp_int:" with the value "17", and "Methane:" with the value "19". Between the "Temp_int:" and "Methane:" fields is an "Operator:" dropdown menu showing a greater-than sign (>). A "Configure" button is located at the bottom left of the form area.

Figura 70 Formulário para configurar a classificação

A Figura 71 apresenta os resultados desta tarefa de classificação. É possível concluir que o método mais eficaz é o *Random Forest Classification*, com uma precisão de 97.12% e que o valor da concentração de metano é superior a 19, o que é verdade.



The screenshot shows a web page titled "Results:". It contains three lines of text: "Best Algorithm: Random Forest Classification", "Score in %: 97.12", and "Methane Value belongs to the Group created: 1".

Figura 71 Resultado da classificação

7. CONCLUSÕES

Atualmente, o conceito de Indústria 4.0 é amplamente conhecido e começa a ser, cada vez mais, aplicado nos diversos setores da indústria. Este facto permite a transição do paradigma industrial tradicional para o *smart manufacturing*. Na base desta mudança está o IIoT que, aliado ao termo *Big Data Analytics*, aporta um enorme leque de vantagens.

Por esta razão, o Departamento de ID da Controlar, considerou o projeto IS4SE uma excelente oportunidade para adquirir competências e conhecimento no âmbito do IIoT, de forma a diversificar a sua área de atuação. O presente trabalho é o catalisador das tarefas atribuídas à Controlar, no âmbito do Projeto IS4SE.

Após a conclusão do trabalho, verifica-se a concretização dos principais objetivos inicialmente propostos, tendo sido desenvolvida uma estrutura funcional de uma rede IoT e uma plataforma *web* capaz de atuar e analisar a informação, bem como de aplicar algoritmos de *machine learning*.

O principal objetivo delineado no início do desenvolvimento da rede IIoT foi atingido, na medida em que esta permite o uso de diferentes protocolos de comunicação e concentra toda a informação no mesmo servidor. As tecnologias LoRaWAN e Wi-Fi foram testadas ao longo de vários meses, comprovando o seu funcionamento. Tal não foi possível para os restantes protocolos (OPC-UA; ethernet; Modbus TCP/IP). No entanto, estes foram integrados como provas de conceito, uma vez que, enquanto foram testados, demonstraram o seu correto funcionamento.

Outro aspeto positivo da estrutura da IIoT apresentada foi o desenvolvimento de uma *gateway* multiprotocolar, que permite, em simultâneo, a transmissão de pacotes Wi-Fi e de LoRaWAN. Esta *gateway* mostrou-se fiável e de fácil configuração devido à interface gráfica, que consegue gerir, de uma forma bastante eficaz, os diferentes scripts que se encontram a correr na *gateway*.

A aplicação *web* desenvolvida é capaz de recolher, analisar os dados e disponibiliza algumas estatísticas provenientes dos diferentes parâmetros de cada um dos nós conectados à rede. Esta aplicação permite ainda mostrar alguns parâmetros da máquina onde o servidor da rede IoT está a correr.

As características acima mencionadas conferem à aplicação a potencialidade desta ser modulada consoante os nós conectados e o contexto em que estes se inserem. Isto também permite extrair informação específica e criar uma análise mais detalhada e pertinente dos dados.

No que diz respeito à implementação de algoritmos de *machine learning* na plataforma *web*, foi possível aplicar diversos algoritmos capazes de fazer a regressão e classificação para um determinado tipo de nós.

Por outro lado, no momento de aplicação destes algoritmos em nós de energia apenas foi possível agrupar os dados e criar perfis de uso. Sendo que, para a previsão do consumo energético os algoritmos de regressão aplicados não apresentaram resultados satisfatórios.

Ainda sobre a aplicação *web*, importa mencionar que foram adicionadas a capacidade de gerar diferentes tipos de alertas e a funcionalidade que permite configurar, a partir da aplicação, a taxa de amostragem dos nós *ethernet* e *Wi-Fi*.

Vários desafios se colocaram ao longo da concretização deste projeto, nomeadamente a falta de disponibilidade em usar PLC's durante o tempo necessário para testar a robustez da integração dos protocolos Modbus TCP/IP e OPC-UA na rede IoT. Outro obstáculo, foi a escassez de tempo na aplicação dos diversos algoritmos de *machine learning*, capazes de prever o consumo energético.

Em termos de melhorias futuras, sugere-se a alteração da estrutura dos diferentes tipos de nós no servidor da rede, de modo a que a aplicação *web* os consiga distinguir sem recorrer a um ficheiro de configuração. Sugere-se, também, a utilização de algoritmos de *deep learning*, uma vez que, é possível que estes consigam prever o consumo energético com maior eficácia.

Desta forma, em concordância com o departamento de ID da Controlar, conclui-se que a execução deste trabalho foi decisiva e bastante pertinente para o projeto IS4SE.

Referências Documentais

- [1] TREND Micro, “Internet of Things (IoT),” [Online]. Available: <https://www.trendmicro.com/vinfo/us/security/definition/internet-of-things>. [Acedido em 8 06 2020].
- [2] IoT Analytics, “Top 10 IoT applications in 2020,” 07 2020. [Online]. Available: <https://iot-analytics.com/top-10-iot-applications-in-2020/>. [Acedido em 31 08 2020].
- [3] A. Rojko, “Industry 4.0 Concept: Background and Overview,” *JIM – Vol. 11, No. 5*, 2017.
- [4] Centi, “IS4SE,” [Online]. Available: <https://www.centi.pt/projetos/outros-centi-nanotecnologia/is4se>. [Acedido em 05 10 2020].
- [5] Agencia Nacional de Inovação - ANI, “Interface,” [Online]. Available: <https://www.ani.pt/pt/valorizacao-do-conhecimento/interface/>. [Acedido em 05 10 2020].
- [6] Borçelik, “Products And Services,” [Online]. Available: <https://www.borcelik.com/en/products-and-services>. [Acedido em 05 10 2020].
- [7] Centi. [Online]. Available: <https://www.centi.pt/>. [Acedido em 05 10 2020].
- [8] Industrial Internet Consortium, “Industrial Networking Enabling IIoT Communication,” 29 08 2018.
- [9] M. Nixon, “A Comparison of WirelessHART™ and ISA100.11a,” *Emerson Process Management*, 2012.

- [10] A. Ferran, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui e T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34-40, 2017.
- [11] R. Rondon, M. Gidlund e K. Landernäs, "Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications," *International Journal of Wireless Information Networks*, 2017.
- [12] J. Cheng, W. Chen, F. Tao e L. Chung-Liang, "Industrial IoT in 5G environment towards smart manufacturing," *Journal of Industrial Information Integration*, 2013.
- [13] i-SCOOP, "NB-IoT explained: a complete guide to Narrowband-IoT," [Online]. Available: <https://www.i-scoop.eu/internet-of-things-guide/lpwan/nb-iot-narrowband-iot/>. [Acedido em 01 09 2020].
- [14] MQTT: The Standard for IoT Messaging, [Online]. Available: <https://mqtt.org/>. [Acedido em 25 03 2020].
- [15] (IETF), Internet Engineering Task Force, "The Constrained Application Protocol (CoAP)," *Internet Standards Track*, 2014.
- [16] OPC Foundation, "What is OPC?," [Online]. Available: <https://opcfoundation.org/about/what-is-opc/>. [Acedido em 31 08 2020].
- [17] OPC Foundation, "Unified Architecture," [Online]. Available: <https://opcfoundation.org/about/opc-technologies/opc-ua/>. [Acedido em 31 08 2020].
- [18] Steve's Internet Guide, "Beginners Guide to The MQTT Protocol," [Online]. Available: <http://www.steves-internet-guide.com/mqtt/>. [Acedido em 31 08 2020].
- [19] OPC Router, "What is MQTT?," [Online]. Available: <https://www.opc-router.com/what-is-mqtt/>. [Acedido em 20 07 2020].

- [20] Norwegian Creations, "MQTT – What Is It? And How Can You Use It?," 20 07 2017. [Online]. Available: <https://www.norwegiancreations.com/2017/07/mqtt-what-is-it-and-how-can-you-use-it/>. [Acedido em 20 06 2020].
- [21] Mosquitto, "MQTT man page," [Online]. Available: <https://mosquitto.org/man/mqtt-7.html>. [Acedido em 21 04 2020].
- [22] OASIS, "MQTT Version 3.1.1," *OASIS Standard*, 29 10 2014.
- [23] BytesofGigabytes, "MQTT Packet Structure," [Online]. Available: <http://bytesofgigabytes.com/mqtt/mqtt-protocol-packet-structure/>. [Acedido em 24 05 2020].
- [24] R. Jamie M, F. Jeremy G, S.-C. Andy J, R. Andrew D e B. Bharat V, "Sensor Networks and Grid Middleware for Laboratory Monitoring," *First International Conference on e-Science and Grid Computing (e-Science'05)*, 12 2005.
- [25] Semtech, "LoRa Modulation Basics," *AN1200.22*, 2015.
- [26] "LORA/LORAWAN TUTORIAL 12," [Online]. Available: https://www.mobilefish.com/download/lora/lora_part12.pdf. [Acedido em 2 09 2020].
- [27] LoRa - DEVELOPER PORTAL, "What is an Adaptive Data Rate?," [Online]. Available: <https://lora-developers.semtech.com/library/tech-papers-and-guides/understanding-adr/>. [Acedido em 2 09 2020].
- [28] LoRa Alliance , "What is it? A technical overview of LoRa and LoRaWAN," *LoRa® Alliance Technical Marketing Workgroup*, 2015.
- [29] L. Casals, B. Mir, . R. Vida e C. Gomez, "Modeling the Energy Performance of LoRaWAN".*Sensors 2017*, 17, 2364.

- [30] The Things Network, “classes,” [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/classes.html>. [Acedido em 25 08 2020].
- [31] A. F. B. Rabaça, “Aplicação de Tecnologia LoRaWAN à Monitorização de Redes de Distribuição de Energia,” *IST*, 2018.
- [32] Newie Ventures, “LoRaWAN: OTAA or ABP?,” 02 2018. [Online]. Available: <https://www.newieventures.com.au/blogtext/2018/2/26/lorawan-otaa-or-abp>. [Acedido em 15 08 2020].
- [33] The Things Network, “LoRaWAN - Regional Parameters,” [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/regional-parameters.html>. [Acedido em 31 08 2020].
- [34] mydevices, “Cayenne Docs,” [Online]. Available: <https://developers.mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload>. [Acedido em 15 04 2020].
- [35] IPSO, “IPSO Smart Object Guideline,” (*IPSO Alliance Technical Guideline*), 05 2017.
- [36] Y. T.-S. A. Hanhu, “Toward Scalable Systems for Big Data Analytics: A Technology Tutorial,” *IEEE Access*, vol. 2, pp. 652-687, 2014.
- [37] J. F. Gantz e D. Reinsel, “The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east,” *IDC iView*, 2012.
- [38] IBM, “What is big data analytics?,” [Online]. Available: <https://www.ibm.com/analytics/hadoop/big-data-analytics>. [Acedido em 2 09 2020].
- [39] Tech America Foundation, “Demystifying Big Data: A Practical Guide To Transforming The Business of Government,” *Federal Big Data Commission*, 2012.
- [40] P. Russom, “Big Data Analytics,” *TDWI Best Practices Report*, 2011.

- [41] P. K. Rajendran, A. Asbern, M. K. Kandala, R. Manoharan e A. Rajendra, "Implementation and Analysis of MapReduce on Biomedical Big Data," *Indian Journal of Science and Technology*, 2016.
- [42] A. Gandomi e M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, 2014.
- [43] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press, 2020.
- [44] sas, "Machine Learning - What it is and why it matters," [Online]. Available: https://www.sas.com/en_us/insights/analytics/machine-learning.html. [Acedido em 11 09 2020].
- [45] J. A. Cruz e D. S. Wishart, "Applications of Machine Learning in Cancer Prediction and Prognosis," *Cancer Informatics*, pp. 59–77, 2007.
- [46] Department of Radiology, Massachusetts General Hospital, "Current Applications and Future Impact of Machine Learning in Radiology," *Radiology*; 288, p. 318–328, 2018.
- [47] A. V. Narkar, "Stock price prediction using feature engineering and machine learning techniques," *B.E., University of Mumbai*, 2019.
- [48] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh e K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," *IEEE Access*, vol. 7, pp. 19143-19165, 2019.
- [49] DEXMA, "Forecasting Energy Consumption using Machine Learning and AI," 2020. [Online]. Available: <https://www.dexma.com/blog-en/forecasting-energy-consumption-using-machine-learning-and-ai/>. [Acedido em 11 09 2020].
- [50] L. G. Viola, "Segmentation of household load-profiles with K-means clustering algorithm," *University of Campinas, SP, Brazil*, 2016.

- [51] Y. Wang, Q. Chen, C. Kang e Q. Xia, "Clustering of Electricity Consumption Behavior Dynamics Toward Big Data Applications," *IEEE Transactions on Smart Grid*, vol. 7, pp. 2437-2447, 2016.
- [52] B. Nepal, M. Yamaha, Sahashi e A. Yokoe, "Analysis of Building Electricity Use Pattern Using K-Means Clustering Algorithm by Determination of Better Initial Centroids and Number of Clusters," *Energies*, 12, 2451, 2019.
- [53] V. Jadhav e V. Ligay, "Forecasting Energy Consumption using Machine Learning," *University College London*, 2016.
- [54] ChipStack. [Online]. Available: <https://www.chirpstack.io/>. [Acedido em 20 01 2020].
- [55] ChirpStack, "ChirpStack architecture," [Online]. Available: <https://www.chirpstack.io/project/architecture/>. [Acedido em 20 02 2020].
- [56] ChirpStack, "InfluxDB," [Online]. Available: <https://www.chirpstack.io/application-server/integrations/influxdb/>. [Acedido em 1 09 2020].
- [57] InfluxData, "Time series database (TSDB) explained," [Online]. Available: <https://www.influxdata.com/time-series-database/>. [Acedido em 7 06 2020].
- [58] InfluxData, "Telegraf," [Online]. Available: <https://www.influxdata.com/time-series-platform/telegraf/>. [Acedido em 23 05 2020].
- [59] Raspberrypi, "Raspberry Pi 3 Model B+," [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. [Acedido em 23 08 2020].
- [60] IMST. [Online]. Available: <https://shop.imst.de/wireless-modules/lora-products/8/ic880a-spi-lorawan-concentrator-868-mhz>. [Acedido em 24 05 2020].
- [61] IMST GmbH, "WiMOD iC880A - Datasheet".

- [62] Raspberry PI, "PoE Hat," [Online]. Available: <https://www.raspberrypi.org/products/poe-hat/>. [Acedido em 1 08 2020].
- [63] Semtech, "LoRa Gateway project," [Online]. Available: https://github.com/Lora-net/lora_gateway. [Acedido em 20 01 2020].
- [64] influxdata, "InfluxDB line protocol tutorial," [Online]. Available: https://docs.influxdata.com/influxdb/v1.8/write_protocols/line_protocol_tutorial/. [Acedido em 24 03 2020].
- [65] "pyModbus," [Online]. Available: <https://pymodbus.readthedocs.io/en/latest/>. [Acedido em 2020 03 17].
- [66] J. Coussement, "node-opcua-logger," [Online]. Available: <https://github.com/coussej/node-opcua-logger>. [Acedido em 12 02 2020].
- [67] Exor, "The Introduction of OPC UA Publish-Subscribe and its Importance to Manufacturers," 09 2019. [Online]. Available: <https://www.exorint.com/en/blog/the-introduction-of-opc-ua-pubsub-publish-subscribe-and-its-importance-to-manufacturers>. [Acedido em 31 08 2020].
- [68] O. Router, "What is OPC UA? A practical introduction," [Online]. Available: <https://www.opc-router.com/what-is-opc-ua/#OPC-Server>. [Acedido em 31 08 2020].
- [69] P. Ferrari, A. Flammini, S. Rinaldi, . E. Sisinni, . D. Maffei e M. Malara, "Impact of Quality of Service on Cloud Based Industrial IoT Applications with OPC UA," *Electronics*, 7, 109, 2018.
- [70] influxdata, "Chronograf," [Online]. Available: <https://www.influxdata.com/time-series-platform/chronograf/>. [Acedido em 29 05 2020].

- [71] influxdata, “How To: Building Flux Queries in Chronograf,” 15 11 2018. [Online]. Available: <https://www.influxdata.com/blog/how-to-building-flux-queries-in-chronograf/>. [Acedido em 10 06 2020].
- [72] Pandas, “Package Overview,” [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html. [Acedido em 12 09 2020].
- [73] Flask, “Flask web development, one drop at a time,” [Online]. Available: <https://flask.palletsprojects.com/en/1.1.x/>. [Acedido em 10 09 2020].
- [74] SQLite, “What is SQLite?,” [Online]. Available: <https://www.sqlite.org/index.html>. [Acedido em 21 09 2020].
- [75] scikit-learn. [Online]. Available: <https://scikit-learn.org/stable/>. [Acedido em 11 09 2020].
- [76] pygal. [Online]. Available: <http://www.pygal.org/en/stable/>. [Acedido em 18 08 2020].
- [77] O'Reilley, “The elbow method,” [Online]. Available: <https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml>. [Acedido em 04 10 2020].
- [78] Pycom, “Fipy,” [Online]. Available: <https://pycom.io/product/fipy/>. [Acedido em 11 09 2020].