



Mestrado em Informática e de Sistemas

---

# **Sistema de integração e aferição da qualidade de dados sensoriais**

Trabalho de Projeto apresentado para a obtenção do grau de  
Mestre em Informática e Sistemas  
Especialização em Tecnologias da Informação e do Conhecimento

**Autor**

**Lúcia Margarida Baptista das Neves**

**Orientador**

**Prof. Doutor João Pedro Costa**

Departamento de Engenharia Informática e de Sistemas

Instituto Superior de Engenharia de Coimbra

**Coimbra, Maio, 2018**



# Agradecimentos

Agradeço em primeiro lugar ao Professor Doutor João Pedro Costa pelo apoio, orientação e incentivo ao longo deste período. Estou muito grata por não ter deixado de acreditar que era possível tendo eu duvidado tantas vezes se o fim acabaria por chegar.

Agradeço a todos os professores e colegas do Mestrado pelos ensinamentos e companheirismo.

Agradeço também aos colegas do Laboratório de Automática e Sistemas do Instituto Pedro Nunes, donde surgiu a ideia para esta dissertação, pelo apoio e incentivo, mas também pelos dados fornecidos para o projeto.

Agradeço aos meus amigos pelo ânimo e incentivo em todas as horas.

Agradeço à minha família, especialmente aos meus pais, sobrinhos, cunhado, mas sobretudo às minhas irmãs pelo apoio durante esta longa jornada que agora termina. Definitivamente não teria chegado até aqui sem o vosso apoio a todos os níveis, são as melhores irmãs do mundo.

Agradeço ao meu namorado pelo apoio incondicional, mesmo quando pensei em desistir. Obrigada pelo incentivo, paciência, carinho e amor em todas as horas.

Muito Obrigada!



# Resumo

Atualmente o termo *Internet of Things* já faz parte do cotidiano de muitas pessoas que procuram recolher e monitorizar informação sobre equipamentos e sistemas que usam diariamente, com o objetivo de otimizar recursos e facilitar tarefas. Os sistemas de *IoT* permitem a tomada de decisões e a execução de ações sem a necessidade de uma intervenção humana contínua. No entanto, para que o resultado seja o esperado é necessário que os dados usados na tomada de decisão sejam confiáveis, algo que nem sempre acontece.

Esta dissertação tem como objetivo propor um sistema que permita, através da especificação e aplicação de regras, conjuntamente com a utilização de fontes de referência (consideradas fidedignas), validar de forma contínua os dados obtidos pelos sensores. Deste modo, aumenta-se a qualidade e confiabilidade dos dados obtidos pelos sensores e nas ações programadas com base nesses dados. Além da especificação e implementação das regras, o sistema proposto possibilita ainda a aplicação de alguns mecanismos de calibração dos dados dos sensores, que inicialmente foram classificados como inválidos, permitindo assim a recuperação e utilização de alguns desses dados.

O sistema proposto é validado, na forma de protótipo, num cenário real onde é recolhida informação meteorológica, no âmbito dum projeto do IPN-LAS.

Palavras-chave: *Internet of Things*, Validação de Dados, Regras de Validação .



# Abstract

Nowadays, the Internet of Things already make part of the daily life of many people seeking to collect and monitor information from equipment and systems they use, in order to optimize resources and facilitate tasks. IoT systems enable decision-making and the execution of actions without continuous human intervention. However, the data used in the decision-making process the must be as reliable as possible, which is not always the case.

This dissertation proposes a system that allows the continuous validation of sensor data, through the application of a set of specified rules, together with the use of reliable reference sources. This approach increases the quality and reliability of the acquired sensor data and consequently the effectiveness of the actions programmed based on this data.

In addition to the specification and implementation of rules, the proposed system implements a set of different calibration mechanisms for data recovering and usage of sensorial information, which were initially classified as invalid.

A prototype of the proposed system was validated, in a real scenario, within the scope of an IPN-LAS project, where meteorological information is collected.

Keywords: Internet of Things, Validation Rules, Data Validation.





# Definições e Acrónimos

API	Application Programming Interface
DB	Database
ECA	Event Condition Action
Email	Electronic Mail
HTTP	Hipertext Transfer Protocol
HTTPS	Hipertext Transfer Protocol Secure
IoT	Internet of Things
IP	Internet Protocol
IPMA	Instituto Português do Mar e da Atmosfera
IPN	Instituto Pedro Nunes
JSON	Java Script Object Notation
KNN	K-Nearest Neighbors Algorithm
LAS	Laboratório de Automática e Sistemas
MQTT	Message Queuing Telemetry Transport
NFC	Near Field Communication
OCG	Open Geospatial Consortium
ODATA	Open Data Protocol
PHP	HyperText Preprocessor
REST	Representational State Transfer
RFID	Radio-Frequence Identification
SDK	Software Development Kit
SMS	Short Message Service
SOS	Sensor Observation Service
SQL	Structured Query Language
SWE	Sensor Web Enablement
URL	Uniform Resource Locator
WSN	Wireless Sensor Network
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol



# Índice

Agradecimentos .....	i
Resumo .....	iii
Abstract .....	v
Definições e Acrónimos.....	vii
Índice.....	ix
Índice de Figuras.....	xi
Índice de Tabelas .....	xiii
1. Introdução .....	1
1.1. Motivação .....	1
1.2. Problema .....	4
1.3. Objetivo .....	4
1.4. Enquadramento .....	7
1.5. Estrutura do relatório .....	8
2. Estado da arte.....	9
2.1. Internet of Things.....	10
2.2. <i>Sensor Observation Service</i> .....	14
2.3. <i>Sensor Things API</i> .....	16
2.4. Sistemas Existentes.....	17
2.5. Avaliação .....	19
3. Arquitetura do Sistema .....	21
3.1. Integração dos Dados de Entrada no Sistema.....	25
3.1.1. Fonte em Análise.....	26
3.1.2. Fonte de Referência.....	27
3.1.3. <i>Plugin</i> de Mapeamento dos Dados.....	28
3.1.4. Fonte Histórica .....	30
3.2. Estrutura e Formatos dos Dados .....	30
3.2.1. Tipos de Métricas dos Sensores .....	32
3.2.2. Armazenamento de Dados .....	33
3.2.3. Classificação dos Dados dos Sensores .....	34
3.3. Validação dos Dados Integrados .....	35
3.3.1. Definição de Zonas .....	36
3.3.2. Determinação Automática de Zonas .....	38
3.4. Calibração dos Dados .....	39
3.4.1. Método dos Mínimos Quadrados .....	40
3.4.2. Calibração Parcelar .....	40
3.4.3. Calibração por Extrapolação dos Valores Conhecidos .....	41

3.5. Ações a Tomar .....	41
3.5.1. Rejeição da Leitura em Análise .....	42
3.5.2. Rejeição da Métrica em Análise .....	42
3.5.3. Considerar a Leitura Anterior .....	42
3.5.4. Remover um Sensor de um Nó .....	42
3.5.5. Remover o Nó .....	42
3.5.6. Redefinir Zonas Automáticas.....	43
3.5.7. Retirar Nó da Zona.....	43
3.5.8. Calibrar o Sensor.....	43
3.5.9. Notificar Utilizadores.....	43
3.5.10. Retirar o Nó ou um Sensor de um Nó.....	44
4. Mecanismos de Validação de Dados .....	45
4.1. Regras de Validação .....	45
4.2. Validação do Domínio .....	49
4.3. Validação com o Histórico do Sensor .....	50
4.3.1. Validação por Comparação com a Previsão.....	50
4.3.2. Validação por Média Móvel.....	51
4.3.3. Variação entre Leituras Consecutivas.....	51
4.3.4. Amplitude de Valores das Últimas N Leituras .....	51
4.4. Validação com os Dados dos Nós Vizinhos (por zona) .....	52
4.5. Validação por Comparação com Fonte Certificada.....	52
4.6. Validação dos Dados de Sensores Correlacionados .....	53
5. Avaliação Experimental.....	55
5.1. Nós e Sensores da WSN .....	56
5.2. Especificação da Fonte .....	57
5.3. Regras de Validação Definidas.....	58
5.4. Protótipo Implementado para o Caso de Estudo.....	59
5.4.1. Módulo de Análise dos Dados .....	60
5.5. Exemplo de Avaliação dos Dados de um Período.....	63
5.6. Calibração do Valor Obtido pelo Sensor .....	64
6. Conclusões .....	65
6.1. Contribuição da Tese .....	65
6.2. Principais Dificuldades .....	65
6.3. Trabalho Futuro .....	66
Referências.....	67
Bibliografia .....	68
Anexo A .....	69
Anexo B .....	75

# Índice de Figuras

Figura 1 - Gráfico de temperatura	2
Figura 2 - Gráfico de temperatura de dois nós vizinhos	3
Figura 3 - Gráfico de temperatura de dois nós vizinhos	3
Figura 4 - Arquitetura da <i>IoT</i> [2]	11
Figura 5 - Taxonomia da <i>IoT</i> [2]	13
Figura 6 - Diagrama de sequência do funcionamento do <i>SOS</i> [3]	15
Figura 7 - Arquitetura do sistema da <i>OpenSensors.io</i> [7]	18
Figura 8 – Instalação típica de uma <i>WSN</i>	22
Figura 9 - Arquitetura do Sistema	24
Figura 10 - Processamento das Fontes em Análise	27
Figura 11 - Processo de aquisição de dados de referência	28
Figura 12 - Mensagem <i>JSON</i> do <i>Accuweather</i>	29
Figura 13 - Processo de aquisição de dados históricos	30
Figura 14 - Tipos de validação do sistema	35
Figura 15 - Exemplo de Zonas	37
Figura 16 - Formato genérico de validação	45
Figura 17 - Representação de métrica pertencente a um nó e a uma fonte	46
Figura 18 - Exemplos de operadores e condições das regras	48
Figura 19 - Fonte de dados proveniente da Fonte Foz Côa	55
Figura 20 - Fluxo de integração dos dados	59
Figura 21 - Definição da Regra R5 segundo a gramática	61
Figura 22 - Implementação da Regra R5	61
Figura 23 - Fluxo de validação dos dados.	62



# Índice de Tabelas

Tabela 1 - Distribuição de sensores na rede	56
Tabela 2 - Tabela com limites especificados para cada sensor	57
Tabela 3 - Lista de regras para os sensores em análise	58
Tabela 4 - Resultados da percentagem de acertos de cada regra	63
Tabela 5 - Validação dos valores calibrados	64





# 1. Introdução

Atualmente estamos na era da *Internet of Things (IoT)*, em que variadíssimos dispositivos eletrônicos são utilizados diariamente para a recolha de informação, de modo a atingir um objetivo, por exemplo a recolha de informação ambiental para adequar o controlador da temperatura de uma sala.

A informação recolhida, ou gerada, por estes dispositivos pode ser processada localmente no dispositivo ou enviada para um sistema de controlo onde a informação é processada e analisada e a tomada de decisão é realizada despoletando ações, como por exemplo baixar o regulador de temperatura de um sistema de climatização . Estes sistemas de controlo, são normalmente são desenvolvidos, especificamente, para um determinado objetivo e/ou para um conjunto de equipamentos específicos. Atualmente, existem diversos sistemas de monitorização de parâmetros, sejam eles ambientais, meteorológicos, qualidade do ar, pulsação, tensiómetros, etc.

Contudo, existe uma crescente necessidade de integrar e correlacionar a informação de diferentes sistemas, ou de fabricantes distintos, de modo a possibilitar uma melhor compreensão das causas associadas à variação dos dados recolhidos, e assim melhorar o controlo e a qualidade das ações tomadas. A título de exemplo, relacionar a variação da temperatura numa sala, com o número de pessoas que nela se encontram bem como com a informação meteorológica do exterior e o seu impacto no edifício, etc.

## 1.1. Motivação

Durante a análise da informação sensorial por vezes são detetadas anomalias provocadas por falhas ou simplesmente valores que não correspondem com o que seria espectável considerando o ambiente de onde provêm. O tratamento correto destas anomalias permite não só minimizá-las como também ter uma noção real das capacidades do sistema.

Durante a fase de testes de cenário de um projeto composto por sensores meteorológicos, surgiu a necessidade de analisar detalhadamente os dados obtidos, essa análise decorreu de forma manual. Da análise, três tipos diferentes de anomalias foram encontradas, nomeadamente:

- Valor completamente díspar do seu antecessor – como se observa no gráfico de temperatura que se segue, duas aquisições de dados consecutivas diferem em cinco graus centígrados tendo sido obtidas num espaço temporal de quinze minutos, algo que não é expectável.

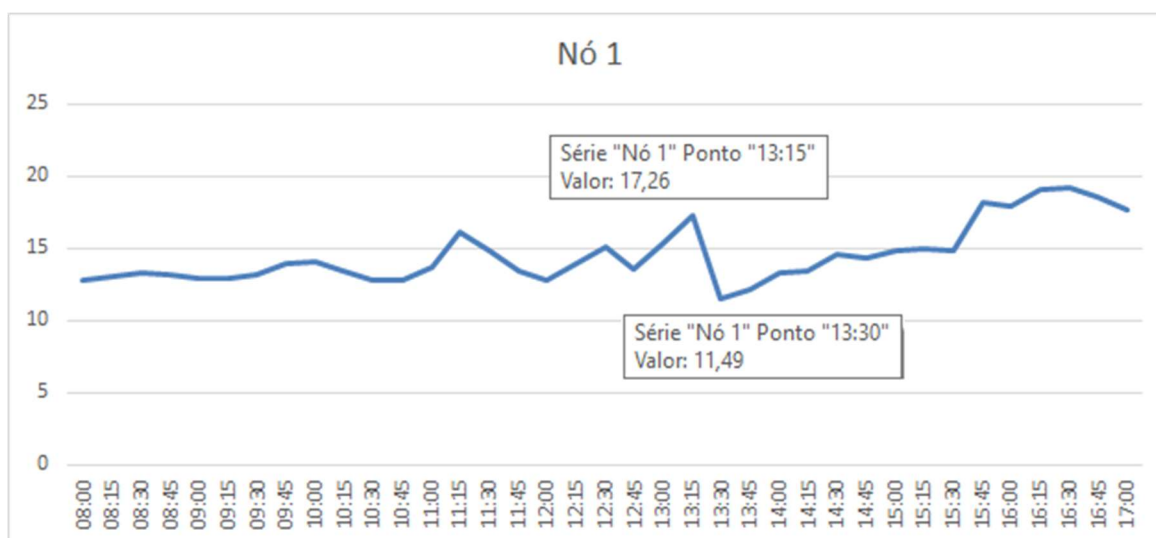
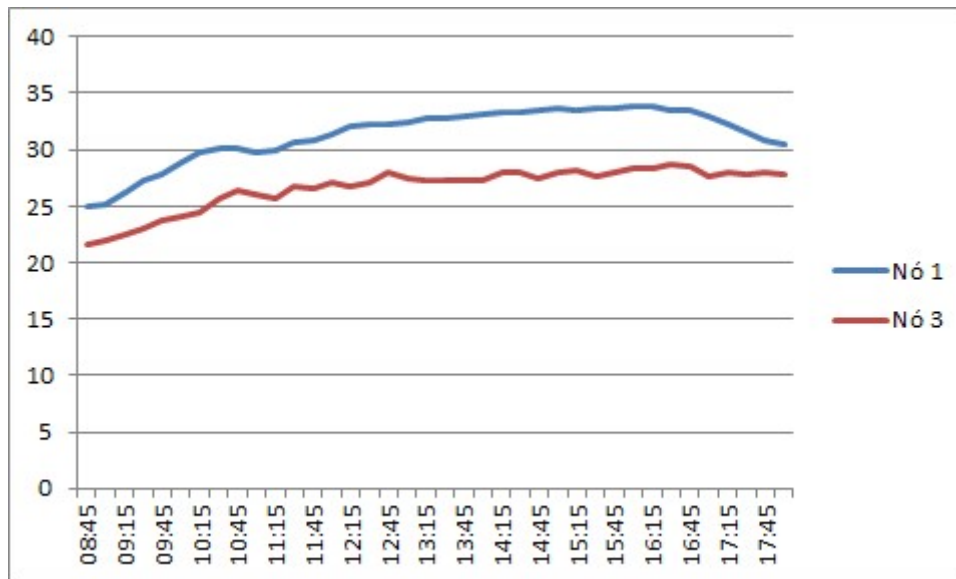


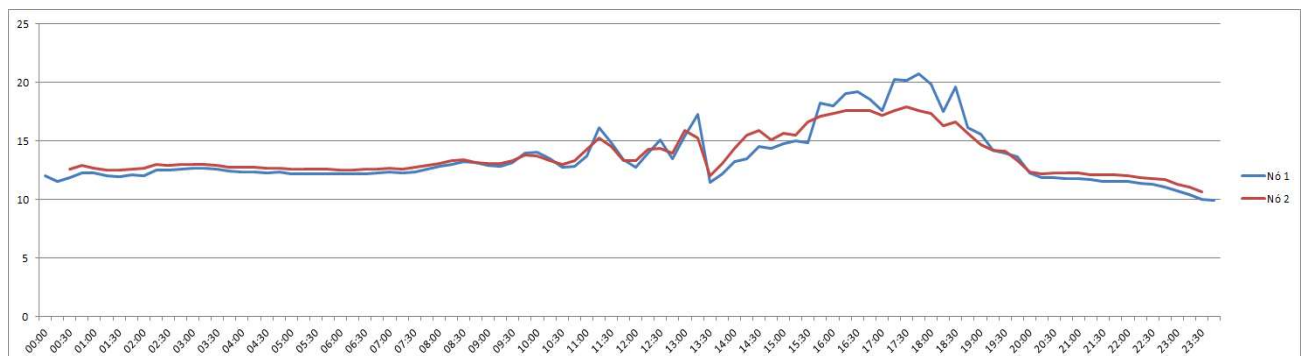
Figura 1 - Gráfico de temperatura

- Dados Sensoriais cuja distribuição apresenta uma diferença praticamente constante entre sensores instalados no mesmo local – no gráfico de temperatura da figura 2 observamos o comportamento de dois sensores que apresentam uma diferença constante entre eles ao longo de um período de tempo.



**Figura 2 - Gráfico de temperatura de dois nós vizinhos**

- Valores que diferem quando determinadas condições se verificam - por exemplo no gráfico seguinte podemos verificar que durante o período da tarde os sensores apresentam resultados díspares, contrariamente aos períodos de não exposição solar.



**Figura 3 - Gráfico de temperatura de dois nós vizinhos**

Para minimizar estes problemas pretende-se encontrar soluções ao nível do software, sempre que isso resolver o problema. Para isso pretende-se implementar algoritmos matemáticos que possam calibrar os sensores, neste contexto a calibração é feita através de software. Em alguns casos este método pode não ser suficiente para resolver o problema, sendo necessária intervenção ao nível do hardware.

## 1.2. Problema

Para possibilitar a integração e a correlação da informação é necessário a existência de um conjunto de mecanismos de integração, definição e recolha dos dados, regras de mapeamento e correspondência dos dados entre as diferentes fontes de informação. Além disso, de igual ou maior importância, é necessário validar a qualidade da informação recolhida. A informação está atualizada? Os valores recolhidos estão dentro do intervalo espectável? Apresentam desvios mínimos relativamente aos dados de fontes de referência?

As variações dos dados recolhidos podem ser derivadas de fatores externos, tais como interferências que podem alterar o funcionamento do sensor, dispositivo responsável pela aquisição de dados respondendo a um estímulo físico ou químico, dependendo do seu objetivo, de maneira específica e medível. Alguns sensores são mais suscetíveis do que outros dependendo do ambiente onde estão inseridos. Para que a influência desses fatores nos dados recolhidos, tenha um mínimo impacto nas ações de controlo, é necessário avaliar a qualidade dos dados relacionando-os, quando possível, com os dados de múltiplas fontes de dados. A avaliação também pode ser feita através da comparação com outras fontes de dados e eventualmente com alguns ajustes de calibração nos sensores, processo que dará confiança aos dados durante um determinado período temporal. Outra forma de avaliação pode ser através da análise dos dados constantemente, a cada aquisição.

## 1.3. Objetivo

Esta tese tem os seguintes objetivos genéricos:

- propor mecanismos de integração e recolha de dados de fontes heterogéneas;
- propor métodos de definição e descrição dos dados recolhidos, e normalização das suas unidades de medida;
- definir mecanismos de validação com base nos dados históricos de um sensor e na correlação de informação proveniente de sensores vizinhos;
- definir mecanismos de validação e qualificação da informação, através da especificação de um conjunto de regras que recorram a fontes de referência e ao histórico de dados do próprio sensor;
- validação das propostas apresentadas através de um protótipo dum caso de teste.

O sistema que se pretende desenvolver deve ser capaz de comparar os dados recolhidos pelos sensores e uma fonte credível à qual são imputadas as mesmas condições. As condições devem ser as mais idênticas possíveis para que o caso de teste possa ser representativo. Depois de validar a fonte e o cenário de teste, o sistema deve ser capaz de qualificar os dados recebidos, não só através da fonte definida como referência, mas também por análise do histórico dos dados.

O sistema deve ser capaz de interpretar os dados recebidos ao longo do tempo e detetar potenciais anomalias. Por exemplo, um sensor de temperatura a adquirir informação a cada dez minutos terá uma variação máxima admissível. Caso esse “*threshold*” seja ultrapassado, é necessário analisar e ajustar o sistema.

O sistema receberá dados da fonte a ser analisada, “num determinado formato e unidade de medida”, estes dados serão analisados do ponto de vista histórico e, sempre que possível, comparando-os com dados que existam de fontes externas credenciadas que permitam aferir com uma pequena margem a aceitabilidade dos mesmos.

Em casos mais específicos, a validação pode ser feita através da relação sensorial, onde se pode tomar por exemplo, a relação entre a temperatura e a pressão atmosférica e verificar se a evolução dos valores é coincidente com o expectável.

Este sistema irá validar os dados adquiridos pelos sensores, monitorizar o seu funcionamento e “gerar alertas” para os responsáveis pela manutenção dos mesmos. Desta maneira o grau de confiança nos dados é maior, melhorando qualitativamente os dados recolhidos pelos sensores.

Como caso de teste irão ser usados dados recolhidos por estações meteorológicas e será feita a validação dos mesmos recorrendo a informação oficial disponível para o mesmo local e hora. Para além disso serão feitas validações usando não só a relação entre sensores mas também analisando o histórico de cada sensor.

Das funcionalidades do sistema, destacam-se as seguintes:

- Ser genérico e permitir a integração de diversos sistemas de sensores em diversos contextos e cenários;
- Permitir a integração e comunicação de sensores com o sistema. Cada sensor deve ser registado no sistema definindo as suas características, domínio,

localização, se os seus dados são partilhados ou não com todos os utilizadores da plataforma, bem como definir se o sensor é para análise ou considerado uma referência;

- Os sensores registados devem pertencer a uma fonte de dados composta por vários sensores que podem ser do mesmo tipo ou de tipos diferentes;
- Validação dos dados recolhidos pelos sensores, através da análise de domínio, e também por comparação com sensores semelhantes;
- Determinação e criação automática de zonas de sensores que apresentam comportamentos semelhantes;
- Definição e aquisição de dados de referências oficiais, extrapolando quando necessário para as localizações em análise tendo em consideração distâncias máximas e as diferenças máximas entre altitudes;
- Criação manual de zonas, pelos utilizadores do sistema, através da disponibilização dos dados sensoriais;
- Disponibilização de mecanismos, personalizáveis, de alarme de mau funcionamento de sensor(es) quando uma determinada taxa de erro é atingida;
- Utilização dos dados históricos para a validação de novos dados, através de comparação da evolução dos mesmos;
- Permitir aos utilizadores consultar os dados relativos às suas zonas através de gráficos configuráveis, bem como consultar os dados de sensores que sejam públicos dentro do sistema;
- Aplicação de algoritmos, customizáveis, aos dados de forma a permitir corrigir alguns desvios relativamente ao previsto, bem como permitir a calibração com base em fontes de referência;
- Permitir o processamento de mensagens provenientes de fontes “*non-compliant*” com os protocolos de comunicação existente, através da customização de esquemas de dados;
- Capacidade para avaliar os dados através da verificação do domínio, comparação com sensores próximos, correlação com outros tipos de sensores, nos casos em que se aplica, e ainda com a evolução histórica que caracteriza o sensor;
- Classificação de dados considerados erróneos como inválidos, por não satisfazerem as regras de validação definidas.

## 1.4. Enquadramento

No decurso das minhas atividades profissionais no Laboratório de Automática e Sistemas do Instituto Pedro Nunes, IPN-LAS, como programadora de software e também responsável pelos testes de sistema dos produtos desenvolvidos pelo laboratório, mais concretamente no âmbito do projeto *InovWine 2* surgiu a necessidade de aferir e validar os dados recolhidos pelos diversos sensores espalhados por algumas vinhas, pertencentes a parceiros do projeto. Tipicamente os sensores estão instalados num Nó que contém uma placa de processamento responsável pelo envio dos dados para o servidor, ou no caso de redes com vários Nós para o Nó Coordenador, sendo este último responsável pelo envio periódico para o servidor. Na fase de testes de sistema os Nós eram submetidos a intervenções para verificar a resposta dos sensores aos diversos estímulos e sendo também utilizados os dados de uma estação vizinha para avaliar o desempenho dos sensores. Após a instalação nas vinhas, localizadas normalmente em locais remotos, o acompanhamento constante dos dados recebidos, permite detetar falhas nas mensagens ou problemas nos sensores, recorrendo-se para isso a estações meteorológicas nas proximidades para aferir a qualidade dos dados obtidos pelos nossos sensores. A necessidade de tornar este trabalho mais rápido e eficiente deu origem à proposta de Tese, uma vez que não foi encontrado nenhum sistema que respondesse às necessidades identificadas.

Como caso de estudo vamos colocar o enfoque numa *WSN*, *Wireless Sensor Network*, constituída por oito nós sensoriais, instalada em Vila Nova de Foz Côa, numa vinha com socalcos à beira do rio Douro. Os nós sensoriais, encontram-se espalhados pelos vários socalcos, apresentando configurações sensoriais diferentes, isto é, são constituídos por diferentes conjuntos de sensores, uns repetidos outros não. A *WSN* cobre uma área de cerca de 43ha, onde todos os sensores comunicam com um Nó denominado Coordenador que gere a comunicação com o servidor remoto. Cada sensor adquire valores, relativos ao seu tipo (temperatura, humidade, pluviosidade, etc.), a cada quinze minutos sendo o Coordenador responsável pelo envio dos mesmos a cada quatro para um servidor que os processa e disponibiliza.

## 1.5. Estrutura do relatório

O presente documento é composto por 6 capítulos:

- No capítulo introdutório é feita uma exposição dos objetivos da Tese e do problema que se pretende resolver;
- O segundo capítulo apresenta uma análise do estado de arte particularmente do *Internet of Things* e de sistemas que comunicam com o *IoT*;
- No terceiro capítulo é discutida a arquitetura do sistema e de que maneira este vai ser construído para responder aos objetivos definidos no primeiro capítulo;
- O quarto capítulo apresenta os mecanismos de validação propostos para os dados;
- No quinto capítulo é feita uma análise da aplicação do protótipo ao caso de estudo;
- Por fim, no sexto capítulo são apresentadas as conclusões do trabalho desenvolvido e propostas de trabalho futuro.



## 2. Estado da arte

Para melhor perceber o problema encontrado e qual o caminho para o resolver, realizou-se uma pesquisa sobre os assuntos envolvidos, nomeadamente sensores e os seus dados, bem como os problemas encontrados na recolha de dados meteorológicos e sistemas de processamento de dados sensoriais.

A calibração de sensores e o seu ciclo de “boa” vida é algo muito importante para diversos sistemas, nomeadamente as estações meteorológicas. Neste tipo de sensores a qualidade dos dados adquiridos é o maior requisito onde apenas dados assertivos podem ser usados para tomar decisões. A qualidade da informação sensorial é definida não só pelo seu intervalo de funcionamento, mas também pela assertividade dos valores recolhidos face a diferentes cenários de funcionamento. Neste contexto de avaliação do funcionamento, foram elaborados alguns estudos. Em *Guidelines on validation procedures for meteorological data from automatic weather stations* [1], são propostas formas de analisar os dados, verificar se estes correspondem ao estímulo imputado ao sensor, permitindo assim concluir se estes estão corretos ou não e detetar quais os erros encontrados. No caso concreto queremos sobretudo detetar as aquisições que não correspondam à realidade do ambiente onde estão inseridos, de forma a aumentar o grau de confiança dos dados obtidos.

Quando são detetados erros devem ser tomadas medidas corretivas, que sob a forma de regras, implementadas por software, permitem minimizar, e até remover, alguns dos erros detetados.

Os procedimentos mais comuns indicados pelo estudo resumem-se na:

- definição de máximos e mínimos para cada tipo de sensor de acordo com a estação do ano, filtrando os dados pelo domínio;
- comparação com dados obtidos em igual período do ano transato;
- definição de limites da variação entre aquisições consecutivas com um intervalo definido;

- identificação de dados que precisam de análise manual. Esta análise é importante quando ocorrem fenómenos extremos que provocam um desvio maior do que o esperado. Nestes casos, sempre que se verifique que o valor corresponde aos eventos ocorridos, este é considerado válido e os valores do domínio são atualizados para que de futuro este seja considerado.

Outra forma de validação é feita através da previsão dos valores que se esperam receber, onde através da análise dos dados já conhecidos é possível prever qual o valor subsequente sendo que se este for significativamente diferente do previsto, com um erro superior a 25%, este deverá ser validado com recurso à validação por comparação permitindo assim, aferir a assertividade do sensor.

De seguida analisaremos alguns softwares existentes que tentam responder às necessidades de integração e análise das informações sensoriais recolhidas pelos mais diversos dispositivos.

## 2.1. Internet of Things

A designação *Internet of Things (IoT)* refere-se sobretudo a dispositivos menos óbvios, por exemplo máquinas de lavar roupa, relógios, chaves, portões automáticos cada um com múltiplos sensores, que ligados à internet podem ser considerados fornecedores de informação. Atualmente estes dispositivos fazem parte de sistemas simples e diretos que despoletam alarmes ou que enviam mensagens, mas também de sistemas complexos de análise de ambientes em estufas, onde as variações de temperatura e humidade tem de ser compensadas com rega automática ou com abertura de condutas usando para isso, séries de sensores, que colocados estrategicamente, são analisados por um sistema inteligente que indica atuações a serem desempenhadas por atuadores responsáveis pela ativação da rega ou abertura de condutas. Quando o sistema é ativo, como descrito no exemplo anterior, após a aquisição de informação pelos sensores e, mediante algum processamento, implica uma ação que é executada pelos atuadores, dispositivos conhecidos por produzir alterações no “seu” ambiente como resposta aos comandos recebidos, estes podem ser manuais ou mecânicos, mas também elétricos.

A arquitetura da *IoT* é comumente descrita em três camadas, Aplicação, Rede e Percepção. Recorrentemente a camada de Aplicação engloba também a camada de negócio e a de Rede a camada de Transporte e de Processamento, como podemos ver na figura 4 do artigo “*Internet of Things: Architectures, Protocols, and Applications*” [2].

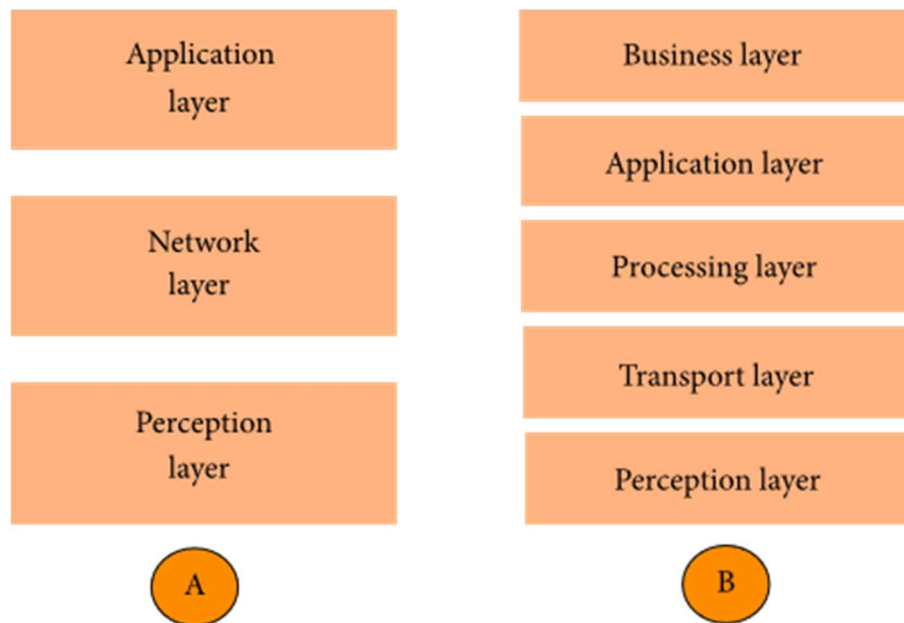


Figura 4 - Arquitetura da *IoT* [2]

Na abordagem de três camadas são descritas a “*Perception layer*”, camada física onde o sensor adquire a informação do ambiente envolvente ou interage com outros sensores ou sistemas vizinhos. A “*Network layer*” é responsável pela comunicação com servidores, outros sensores e outros dispositivos de comunicação, é também responsável pelo envio dos dados. A “*Application layer*” é responsável por fornecer serviços específicos de aplicação, define a utilização dos dados e qual a sua aplicabilidade, por exemplo o uso de *IoT* em casas inteligentes.

Por outro lado, temos a versão de cinco camadas onde a camada de ligação é dividida em “*Transport layer*”, a qual tem uma função mais linear sendo descrita na literatura como responsável por enviar os dados pela internet ou outro meio de comunicação como NFC ou RFID, da camada de percepção para a camada de processamento, e a “*Processing layer*”, camada de *middleware* que armazena, processa e analisa os dados, tem também a função de gerir as camadas inferiores, por exemplo configurar a aquisição de dados. Finalmente a “*Bussiness Layer*”, em português, camada de negócio, gere a utilização que é feita dos dados, por exemplo uma pulseira com um sensor para medir a pulsação deve enviar um alarme sonoro quando um limite é ultrapassado, mas esses dados são enviados

para um servidor que analisa e estuda o perfil da pessoa que usou esse sensor, fornecendo-lhe depois outros tipos de serviços.

Existem variadíssimas aplicações para o *IoT*, desde os sensores ligados à meteorologia que permitem a agricultores profissionais ou mesmos amadores, existem algumas soluções de baixo custo, gerir as suas regas, adubações e combate a pragas de acordo com informações precisas sobre a exposição das culturas aos fenómenos meteorológicos. Passando pelo médico que pode consultar o histórico cardíaco completo de um paciente recorrendo a um pequeno dispositivo que este usa no seu dia-a-dia como um relógio, ligado ao sistema de saúde. Ou até no combate ao desperdício de energia e tempo, tendo uma casa instrumentada por sensores e atuadores que ligados a sistemas inteligentes, permitem automatizar o aquecimento de divisões com base nos hábitos dos seus utilizadores.

As principais vantagens em usar *IoT* são:

- A automação de algumas tarefas liberta as pessoas para outras tarefas mais relevantes;
- Eficiência e eficácia, por exemplo o frigorífico regular a sua temperatura de acordo com a temperatura exterior;
- No limite, melhor qualidade de vida devido à diminuição de tarefas repetitivas.

No entanto também existem desvantagens:

- Grandes volumes de dados que necessitam de tratamento;
- Possíveis falhas de segurança podem pôr em causa a privacidade dos dados;
- Complexidade, quantas mais tarefas dependerem da *IoT* maior a necessidade de proteções e robustez;
- Substituição de pessoas por máquinas.

O sistema que pretendemos desenvolver alimenta-se dos dados recolhidos na *IoT*. A Figura 5 ilustra a taxonomia da *Internet of Things* também do artigo “*Internet of Things: Architectures, Protocols, and Applications*”,

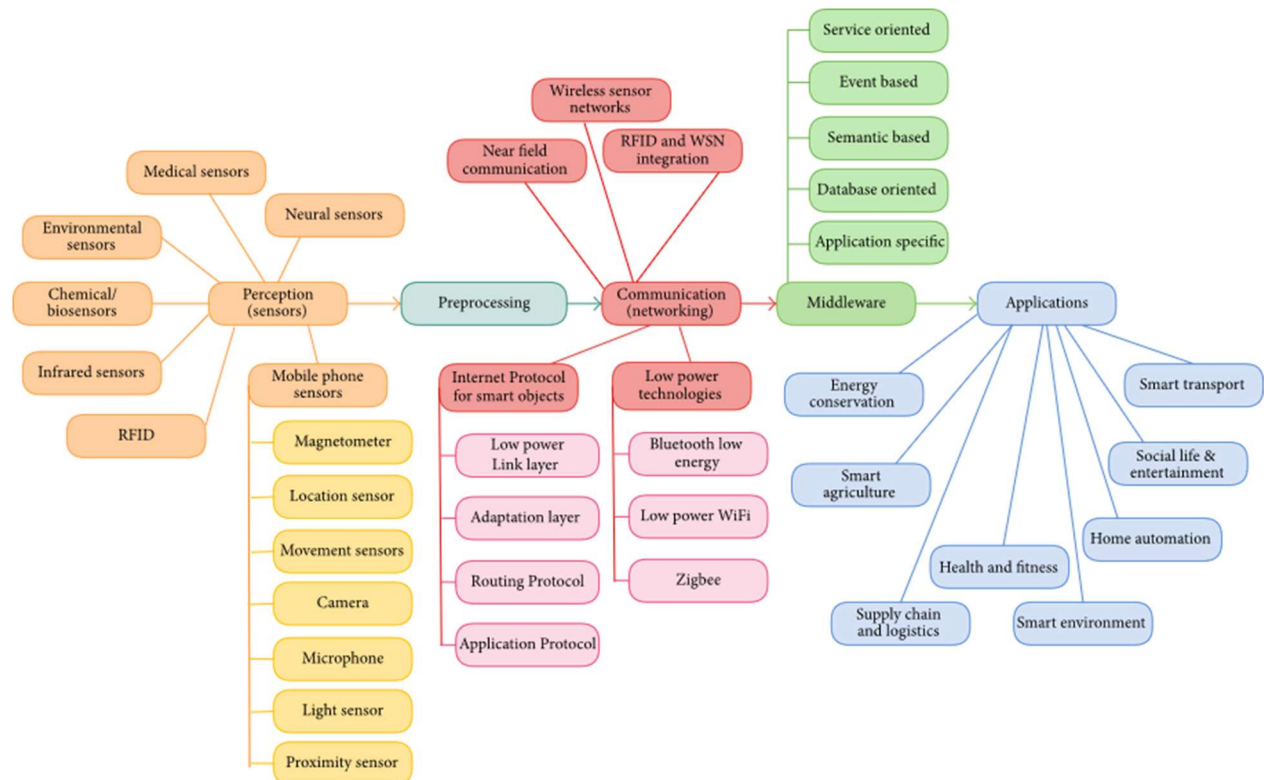


Figura 5 - Taxonomia da *IoT* [2]

Como ilustrado na figura, os dados recolhidos num sistema *IoT*, genericamente atravessam várias camadas desde que são recolhidos por sensores até serem disponibilizados a aplicações. Os sensores operam na camada de Percepção, onde recolhem a informação para a qual foram criados sobre o ambiente que os rodeiam. Esta informação, depois de um pré-processamento inicial é enviada, através da camada de comunicação, para os sistemas de interpretação, situados na camada de *Middleware*, antes de serem disponibilizados à camada de Aplicação onde os dados são usados para os mais diversos fins. A título de exemplo, um sistema que monitoriza a concentração de CO<sub>2</sub> numa garagem pública, recolhe a informação a cada 5 minutos sobre o ar no local, encapsula na camada de pré-processamento a informação sensorial filtrada, tipicamente e na sua forma mais simples, através da aplicação da média a aquisições sucessivas, envia através da camada de comunicação para o *Middleware* onde, aos dados, são aplicadas as fórmulas de conversão, sendo posteriormente disponibilizados à camada de Aplicação onde é verificado os limites de nocividade foram ultrapassados e com isso sejam ativados os extratores de fumo existentes no local.

O foco desta tese centra-se nas camadas de *Middleware* e Aplicação. Na camada de *Middleware* pretende-se analisar os dados, a sua assertividade e precisão, bem como,

detetar eventuais erros ou falhas no funcionamento dos sensores. Na camada de Aplicação, pretende-se processar os dados de forma a que permitam, não só, a tomada de decisões ao nível da manutenção e da intervenção, identificando possíveis problemas sensoriais, mas também permitir o cruzamento de dados provenientes de múltiplas fontes e a criação de “clusters” sensoriais. Por “clusters” sensoriais entendam-se sensores do mesmo tipo e geograficamente vizinhos que apresentam uma distribuição de dados similar nos mesmos instantes temporais.

## 2.2. *Sensor Observation Service*

O *standard SOS* [3] foi especificado pelo *Open Geospatial Consortium*, através de um grupo de trabalho denominado *Sensor Web Enablement*, *SWE*, que harmoniza a comunicação entre os dispositivos de recolha, a comunicação para armazenamento e/ou disponibilização dos mesmos na web.

Existem sete especificações chave que permitem efetuar várias ações sobre os dados:

- *O&M (Observation and Measurements)* - standard que define o formato do *XML* para envio de medidas e observações e tudo o que está relacionado com as amostras recolhidas pelos sensores;
- *PUCK Protocol Standard* - protocolo para obter a descrição detalhada do *SensorML*, o código de comunicação com o sensor e também outras informações relevantes para a instalação, configuração e outras operações. Facilita a instalação dos sensores;
- *SensorML (Sensor Model Language)* - standard que define modelo *XML* para definir semanticamente processos e componentes de processamento;
- *SOS (Sensor Observation Service)* - standard que define uma interface através de “*webservice*” possibilitando diversas operações que vamos detalhar de seguida;
- *SPS (Sensor Planning Service)* - define a interface de consultas que fornecem informações sobre o sensor e as suas tarefas;
- *SWE Common Data Model* – define modelos de dados para troca de dados de baixo nível entre nós da *OGC Sensor Web Enablement (SWE)*;

- *SWE Common* - define os tipos de dados a usar através dos serviços *OCG SWE*, define também operações, tipo de pedidos e respostas.

De todas as especificações a mais relevante para o nosso estudo é o *SOS*, onde se encontram definidas um conjunto de “regras” que possibilitam a disponibilização de dados recolhidos por sensores (denominados *data producers*) e, como estes podem ser utilizados por sistemas de análise ou apresentação de dados (denominados por *data consumers*). Para ambos são definidas interfaces, através de *web services*, que possibilitam a comunicação como ilustrado na figura 6.

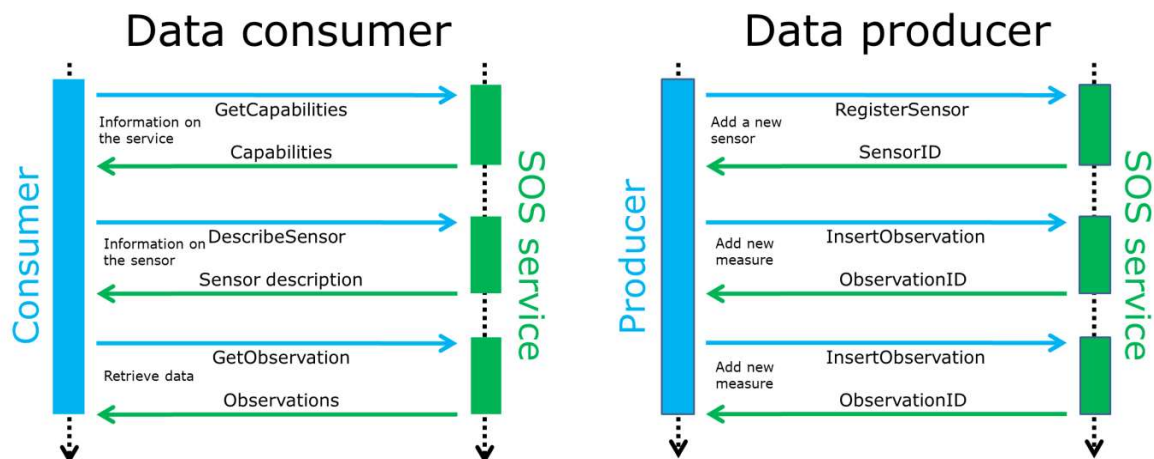


Figura 6 - Diagrama de sequência do funcionamento do *SOS* [3]

O “consumidor” pede ao serviço *SOS* as informações sobre os sensores que este disponibiliza, estes pedidos circulam sobre *HTTP POST*, o serviço responde sobre as características do sistema que está a consultar, quais as suas especificidades e, como é possível interagir com ele. Outro pedido típico é a descrição dos sensores, este fornece informação detalhada sobre os sensores e como são geradas as suas medidas. Por último o “*GetObservation*” permite o acesso aos dados observados pelo sensor através de uma *query* indexada no tempo e que pode ser filtrada por tipo de dados.

A comunicação com o “produtor” é feita através das operações de registo de sensores, ao qual o serviço atribui um identificador, depois de registado é possível adicionar observações que descrevam o funcionamento do sensor, seguindo as indicações da *O&M* e a versão previamente combinada para o sensor em causa. Por fim “*InsertObservation*”

adiciona observações ao sensor mediante a indicação do seu indicador. No âmbito do *SOS* existem cinco conceitos chave:

- Observações – representam os valores lidos num determinado instante;
- Procedimento – indica quem fornece a leitura, pode ser um sensor ou um processo que interaja com o sensor;
- Propriedades Observadas – representa o fenómeno que se quer medir, representada no *O&M standard*;
- Característica de Interesse – representa as características relevantes que determinado sensor pode ter, por exemplo a sua localização;
- Oferta – representa um conjunto de sensores que podem aparecer em conjunto, por exemplo uma estação meteorológica.

Alguns destes conceitos serão importantes para o nosso estudo, no entanto o nosso sistema pretende ser mais heterogéneo, incluir comunicação em *JSON*, *JavaScript Object Notation*, e disponibilizar alguns serviços que não estão previstos no *SOS* ou que foram abandonados na versão 2.0 como é o caso dos alertas sobre o funcionamento dos sensores que fazem parte do sistema.

### 2.3. *Sensor Things API*

O *Standard Sensor Things API* [4], também definido pelo *OCG*, pretende definir uma forma unificada de ligar dispositivos *IoT* e aplicações. Este também faz parte do *SWE*, sendo dividido em duas partes essenciais, a parte sensorial e parte das tarefas. A parte sensorial define a forma como são geridos e tratados os dados recolhidos pelos diversos tipos de sensores, a qual irá usar *O&M* definida anteriormente. Este distingue-se dos standards definidos no *SWE* pela utilização de *REST*, o uso da codificação *JSON*, o protocolo *ODATA* (*Open Data Protocol*), que permite o acesso a dados por aplicações distintas e também pelo uso do protocolo *MQTT* (*Message Queuing Telemetry Transport*). O *MQTT* é um protocolo frequentemente utilizado em sistemas de sensores e dispositivos móveis, pois é baseado no modelo *Publish/Subscribe*.



## 2.4. Sistemas Existentes

A plataforma *SensorUp* [5] implementa o *standard Sensor Things API* para comunicar com os sensores, disponibilizando uma biblioteca de *firmware*, a *SensorUp's Embedded Firmware for Gateways*, a qual permite aos dispositivos enviar os dados diretamente para a *cloud* onde é feito o armazenamento e também algum processamento. É disponibilizado um *SDK*, *SensorUp SDK*, que permite reutilizar as funções disponibilizadas para melhor se adaptarem aos diferentes tipos de dados, disponibilizando também uma *API* para comunicação com outros sistemas que utilizam os dados numa perspectiva de alto nível. Enquadrando o *SensorUp* com o sistema em estudo, este pode ser entendido como uma fonte de dados.

A *istSOS* [6] é uma plataforma de agregação de dados sensoriais que fornece um serviço de recepção de dados, desenvolvida de acordo com o *standard SOS*. Cada sensor é registado na plataforma usando as definições do *SOS* e são definidos os métodos de interpretação dos dados e as especificidades de cada sensor. Permite ainda configurar a rede sensorial, intervalos de aquisição, tipos de dados, etc. Podem também ser definidos procedimentos de manipulação de dados. A comunicação entre os sensores e o sistema é feita através do protocolo *MQTT*. Apesar de ser um sistema muito abrangente, permitindo a adição de vários métodos, não disponibiliza nativamente mecanismos de cruzamento de dados com os seus vizinhos, a análise de dados históricos ou mesmo a criação de zonas semelhantes propostas na solução em estudo.

A *OpenSensorsIO* [7] é uma plataforma que agrega dados sensoriais, permitindo integração de diversos tipos de sensores, desde que tenham uma ligação à internet para se ligarem ao “*message broker*”. Os dados podem ser visualizados numa plataforma *Web*, estes podem ser de acesso público ou restrito. Este sistema permite também a configuração de alertas de mau funcionamento dos sensores, quando os valores estão fora dos “*thresholds*” definidos. A abordagem proposta nesta tese é mais abrangente, pois permite a análise comparativa com sensores vizinhos, assim como a análise da variação e coerência com o histórico dos valores recolhidos.

A comunicação com os sensores é feita através de MQTT, estando a arquitetura da plataforma ilustrada na figura seguinte:

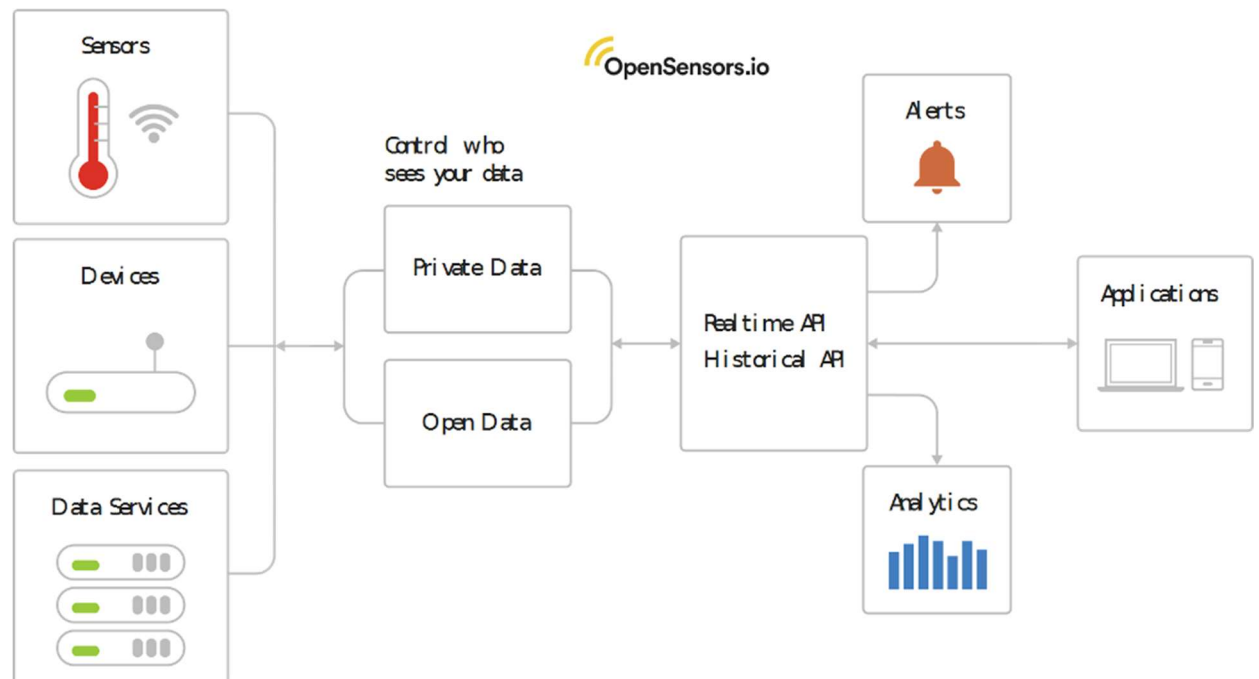


Figura 7 - Arquitetura do sistema da *OpenSensors.io* [7]

Os sensores podem comunicar diretamente com o sistema ou, caso não possuam placa de rede acoplados a um dispositivo de comunicação. Os dados lidos passam por um módulo de controlo, onde é feito o controlo do acesso aos dados, sendo posteriormente disponibilizados numa API que permite a comunicação com outros módulos e aplicações, tais como o módulo de análise e o módulo de alertas.

O *OpenIoT* [8] é uma plataforma *Open Source* que disponibiliza um interface de integração de sensores, direcionado para *IoT*, apresentando os diferentes sensores ligados através do uso de ontologias. Esta solução aborda uma vertente distinta do processamento de dados de redes *IoT*, mais focada nos tipos de ligações existentes entre os vários sensores que compõe a solução, e não nos dados propriamente ditos.

O sistema *Sigfox*, [9] possui uma rede de comunicação proprietária, sobre a qual circulam os dados recolhidos pelos sensores e é composto por módulos que permitem o acoplamento de sensores de vários tipos que comunicam com o posto de receção mais

próximo. Esse posto comunica com o sistema central onde os dados são armazenados. Estes dados podem ser consultados na área pessoal da página *web* de cada utilizador. Não é efetuada qualquer análise aos dados, no entanto é disponibilizada uma *API* baseada em *REST* através da qual é possível fazer integração com outros sistemas. É possível efetuar a análise dos dados e aplicação de medidas corretivas quando necessário, através da criação de um *plugin* que permite efetuar o mapeamento dos dados disponibilizados pela *API Sigfox*. Este sistema é complementar ao proposto, podendo disponibilizar fontes de dados, pois está mais focado na recolha e processamento dos dados, mas não tanto na análise dos mesmos.

## 2.5. Avaliação

De entre os sistemas analisados, nenhum responde às necessidades identificadas, não permitindo nenhum a criação de zonas de semelhança, nem tão pouco uma validação tão abrangente como a que se propõe. Esta é uma área em constante desenvolvimento e evolução, com propostas de sistemas e standards. O *istSOS* é um dos sistemas que implementa o *standard SOS*, não contempla a validação por zona nem a possibilidade de calibração sensorial. Outros sistemas, como o *OpenSensorIO*, usam tecnologias de comunicação diferentes e embora cubram alguns dos requisitos identificados, tais como a validação de dados e o envio de alertas de mau funcionamento, não permitem a aplicação de algoritmos de calibração. Os sistemas *Sigfox* e o *OpenIoT* podem ser integrados como fontes de dados no sistema, sendo estes posteriormente processados e calibrados.



### 3. Arquitetura do Sistema

Por forma a atingir os objetivos propostos, nomeadamente a análise e calibração de dados sensoriais, é necessário que o sistema permita não só, a integração de dados provenientes de redes de sensores, que carecem de calibração, mas também de dados de referência provenientes de Fontes Oficiais, tais como, o IPMA, Instituto Português do Mar e da Atmosfera. A proveniência de todos os dados a integrar no sistema, quer sejam de sensores descalibrados ou de referência, é registada ou denominada como uma Fonte de dados (ou simplesmente de Fonte). Além da sua designação é definida a sua origem, tipo e localização do local onde estão instalados. Uma Fonte pode ser definida como uma fonte de dados a analisar (Fonte em Análise), ou como uma Fonte Oficial, que por ser de referência, é considerada confiável. Uma Fonte pode representar, e referenciar, os dados recolhidos por um ou por vários Nós. Um Nó é um dispositivo composto por um ou mais sensores e um módulo de processamento que lhe permite indexar as leituras recolhidas pelos sensores.

Um Nó tipicamente também possui um módulo de comunicação para envio dos dados quer para um concentrador, ou mesmo diretamente pela *internet* até ao sistema de integração e processamento de dados. Internamente, cada Nó indexa as mensagens de acordo com um formato de troca de mensagens previamente estabelecido, de modo a que os recetores os possam interpretar, analisar e guardar. A figura 8 ilustra um exemplo de uma rede *WSN*, constituída por vários Nós que recolhem informação ambiental de uma vinha. Cada Nó recolhe e envia informação sensorial, proveniente dos sensores que o compõem, sejam eles de temperatura, humidade do ar, velocidade do vento, etc.



**Figura 8 – Instalação típica de uma WSN**

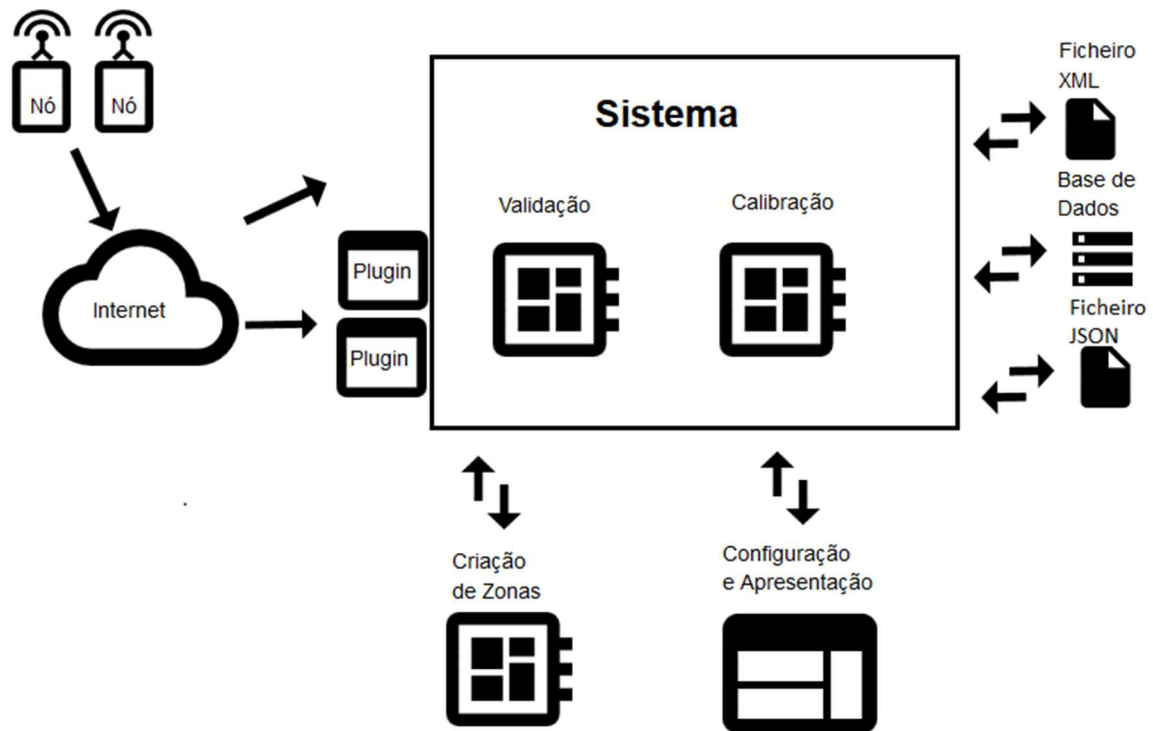
Na perspetiva do sistema, as Fontes (fontes de análise) e as Fontes Oficiais (fontes de referência) recolhem e disponibilizam um conjunto de informação, e medidas, que necessitam de ser normalizadas de modo a poderem ser correlacionadas e analisadas em conjunto. Como os dados têm origem em sistemas diferenciados, pode ser necessário algum processamento prévio, onde através da utilização de adaptadores (transformadores de dados) permite a sua normalização e posterior compatibilização. A título de exemplo, tomemos duas fontes sensoriais que recolhem informação relativa a temperatura em unidades diferentes (°C Celsius e °F Fahrenheit) que precisam de ser convertidas antes de serem usadas. Além da integração de diversas fontes de dados, no processo de análise também serão utilizados os dados históricos provenientes de aquisições anteriores, para determinar, por exemplo, variações ao comportamento espectável. Estes dados históricos serão considerados e catalogados como sendo provenientes de uma Fonte Histórica.

Como referido anteriormente, a proposta do sistema vem colmatar uma necessidade sentida aquando do desenvolvimento de um projeto do IPN-LAS, em que os dados recolhidos eram armazenados em formato *JSON*, e posteriormente analisados de forma semi-manual afim de se identificar potenciais erros. Como o *JSON* é uma tecnologia amplamente utilizada, independente da linguagem de programação e de fácil compreensão

e implementação, decidiu-se utilizar *JSON* como o formato base do sistema. Posteriormente, o *standard SOS* foi analisado e considerado como sendo uma mais-valia para tornar o sistema mais heterogéneo. No entanto, e de forma a minimizar o impacto no sistema em desenvolvimento, decidiu-se não alterar o modelo de representação de dados do sistema definido inicialmente, e optou-se por oferecer um *plugin* que permita a interpretação da informação no *standard SOS*, e a converta para o esquema de dados do sistema.

Genericamente, após a especificação das fontes de dados e a sua proveniência, a definição do esquema das mensagens enviadas pelos sistemas de sensores e quais as suas métricas, as mensagens entram no sistema para serem processadas. Estas mensagens podem vir num formato *JSON* pronto a ser interpretado, ou em formato *raw*, sendo nestes casos necessário proceder à sua conversão. Este tipo de conversão, assente em modelos de *plugin*, tal como ilustrado na figura 9, permite que sistemas com protocolos de comunicação diferentes consigam comunicar de forma transparente com o sistema em desenvolvimento. Pode ser necessário implementar um *plugin* específico para cada fonte de dados. Após os dados terem sido convertidos, as mensagens entram no sistema de forma transparente ,sendo posteriormente validadas e analisadas.

Todas as mensagens são verificadas, validadas e são processados os dados que transportam. Os dados são combinados através de algoritmos que permitem aferir a validade dos dados recolhidos pela Fonte em Análise. Posteriormente e em caso de necessidade, são aplicados algoritmos de calibração de forma a ajustar os valores recolhidos, pela Fonte em Análise, com os dados provenientes de Fontes de referência. Quando os valores recolhidos não cumprem os critérios de validação, estes são descartados e é gerado um alerta de mau funcionamento. A figura 9 ilustra a visão geral do sistema.



**Figura 9 - Arquitetura do Sistema**

Os Nós enviam as mensagens, com os valores recolhidos de um ou mais sensores, através da internet, e são integrados no sistema através do registo de uma, ou várias, fontes de dados. Estas fontes são continuamente alimentadas de acordo com a frequência de comunicação específica de cada Nó. As mensagens enviadas pelos nós podem ser diretamente integradas no sistema, quando o formato dos dados é compatível ou, caso contrário, serão integradas de forma indireta por intermédio de um plugin que possibilita que estas sejam recolhidas e convertidas de acordo com o formato especificado na definição da fonte de dados.

Após a sua integração no sistema, os dados provenientes das Fontes em Análise irão passar por duas fases distintas:

- Fase de validação – em que os dados são avaliados e validados de acordo com um conjunto de regras que permitem determinar a sua exatidão. Estas incluem a validação do domínio absoluto de cada um dos valores recolhidos pelos sensores, dos valores espectáveis para o ambiente em que os sensores estão inseridos e também se estes respeitam a sua própria arquitetura. Os dados que nesta fase sejam considerados erróneos são encaminhados para a fase de calibração;



- Fase de calibração – nesta fase, fórmulas de calibração são aplicadas aos dados, que anteriormente foram considerados erróneos. Estas fórmulas usam o valor considerado errado e os seus antecedentes para tentar aproximar esse valor do valor expectável. Uma outra forma de calibração consiste na utilização de uma fonte de referência, quando existe, a partir da qual se possa obter uma estimativa para colmatar as falhas encontradas. Os novos valores gerados a partir da calibração são novamente avaliados pelo sistema.

Os dados de um sensor são analisados, não só com os seus dados históricos, mas também com os dados obtidos por sensores que pertençam à mesma Zona. Uma Zona é composta por sensores do mesmo tipo, colocados em locais diferentes, podendo estes ser próximos ou não, os quais historicamente apresentam variações semelhantes podendo por isso serem considerados referências para as Fontes em Análise.

Os dados depois de analisados e quando necessário, calibrados, são armazenados num repositório de dados, conjuntamente com a indicação da sua validade e da sua proveniência, se real ou estimado, assim como o resultado da fase de validação. Apenas os dados considerados válidos serão considerados para a análise histórica.

O armazenamento pode ser feito em Base de dados, ficheiros *JSON* ou *XML*. O *frontend* do sistema permitirá a consulta de toda a informação associada aos dados. Nessa interface será possível consultar os dados recebidos das Fontes, assim como a classificação atribuída. No que concerne às regras de aceitabilidade dos dados, será também possível, visualizar os ajustes efetuados (calibração) e qual a taxa de erro. Ao nível da configuração de alertas e ajuste nas regras de classificação também podem ser feitos nesta interface.

Na secção seguinte é descrito de que forma são integrados os dados no sistema.

### 3.1. Integração dos Dados de Entrada no Sistema

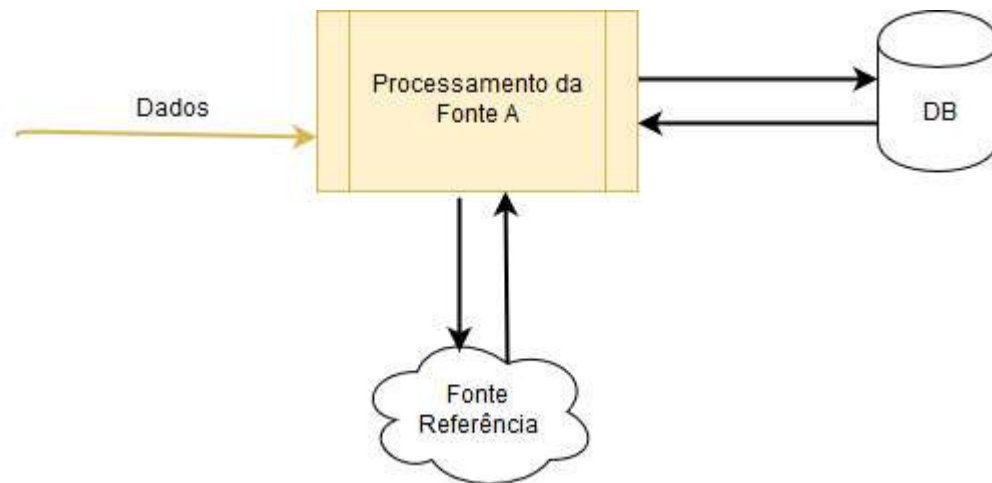
O sistema integra três tipos de dados distintos: dados provenientes de sensores das Fontes em Análise, vamos designar por **Dados em Análise** recolhidos de sensores; dados de referência obtidos de Fontes Oficiais, designados como **Dados Oficiais** e **Dados Históricos** provenientes de dados históricos armazenados pelo sistema e que já foram previamente analisados e considerados como válidos, e como tal, passíveis de serem utilizados como fonte de referência histórica.

- **Dados em Análise** - dados recolhidos por sensores provenientes de diversos sistemas, e que se pretendem analisar. Sensores estão acoplados a um módulo de processamento, designado por nó. Um nó pode conter um ou vários tipos de sensores, e tipicamente envia periodicamente uma mensagem com todas as aquisições de cada um dos sensores num determinado instante de tempo. Por forma a facilitar a análise individual de cada sensor, os dados das mensagens são divididos e agregados por tipo de sensor;
- **Dados Oficiais** – dados provenientes de fontes fidedignas, de instituições publicas e/ou privadas, mas cujos dados foram previamente validados, são considerados assertivos e confiáveis;
- **Dados Históricos** – dados recolhidos ao longo do tempo que estão guardados no repositório (ex. base de dados, sistema de ficheiros, etc.), estes dados já se encontram classificados como confiáveis ou não, podem assim filtrar apenas os dados corretos para a análise. Todos os dados são armazenados, mesmo aqueles que são considerados erróneos distinguindo-se pela sua classificação.

As subsecções seguintes descrevem o processamento de cada um destes tipos.

### 3.1.1. Fonte em Análise

Uma fonte em análise que integra os dados dum sistema de sensores, depois de registada no sistema, fica disponível para processar os dados provenientes desse sistema. Isto é, todas as mensagens que esse sistema de sensores envia, ficam disponíveis na fonte para serem posteriormente validadas. Para isso, o processo responsável pelas mensagens dessa fonte, recolhe da base de dados, as definições da fonte, nomeadamente a informação sobre os nós do sistema, os sensores e respetivos limites, e a fonte de referência a utilizar na validação dos dados. O sistema de receção de dados funciona em *push*, à espera das mensagens. A figura seguinte ilustra as interações do processo de processamento da fonte.



**Figura 10 - Processamento das Fontes em Análise**

Para cada fonte existe um processo que valida os dados da fonte, de acordo com as regras definidas para essa fonte. Cada uma das leituras obtidas inclui informação relativamente ao tipo de sensor, o valor lido pelo sensor e o *timestamp* da aquisição. Cada uma dessas mensagens é validada de acordo com as características definidas para o sensor e com as validações definidas para a fonte em concreto. Sempre que os dados cumpram as condições definidas numa regra, são aplicadas as ações previstas na definição da regra.

### 3.1.2. Fonte de Referência

Uma fonte de referência do sistema tem um funcionamento diferente. O pedido de dados é efetuado de acordo com a frequência especificada ou quando houver necessidade de obter um valor para um determinado instante, funciona em sistema *push*. Tipicamente as fontes de dados oficiais disponibilizam *APIs* que permitem recolher esses dados. Por exemplo, o *Accuweather*, sistema de informação meteorológica de previsão e aquisição de dados por todo o mundo, disponibiliza uma lista com as estações disponíveis. Após a escolha da estação, a *API* disponibiliza funções que permitem obter os dados recolhidos por essa estação. Existe um número máximo de pedidos que podem ser feitos na modalidade gratuita, sendo ilimitado na modalidade paga. Outras fontes oficiais podem ser consideradas dependendo das áreas geográficas em estudo, tais como IPMA, que fornece informação sobre os valores recolhidos nas capitais de distrito do país a cada hora ou, em Coimbra, a estação meteorológica do Laboratório de Aerodinâmica Industrial da Associação para o Desenvolvimento da Aerodinâmica Industrial, que é reconhecida pela

qualidade dos dados no site *Weather Underground* que publica os seus dados. O *Weather Underground* publica dados de várias estações, nem todas marcadas como fidedignas, podendo ser fornecedora de fontes referências.

Idealmente para cada fonte em estudo deve ser indicada uma fonte de referência que possa dar informação fidedigna da zona em causa. Como o formato dos dados disponibilizados pelas fontes de referência, normalmente é distinto, de modo a compatibilizar e a serem integrados pelo sistema pode ser necessário implementar módulos, designados de *plugins*, que permitam a transformação e conversão num formato compatível, como ilustrado na figura seguinte.

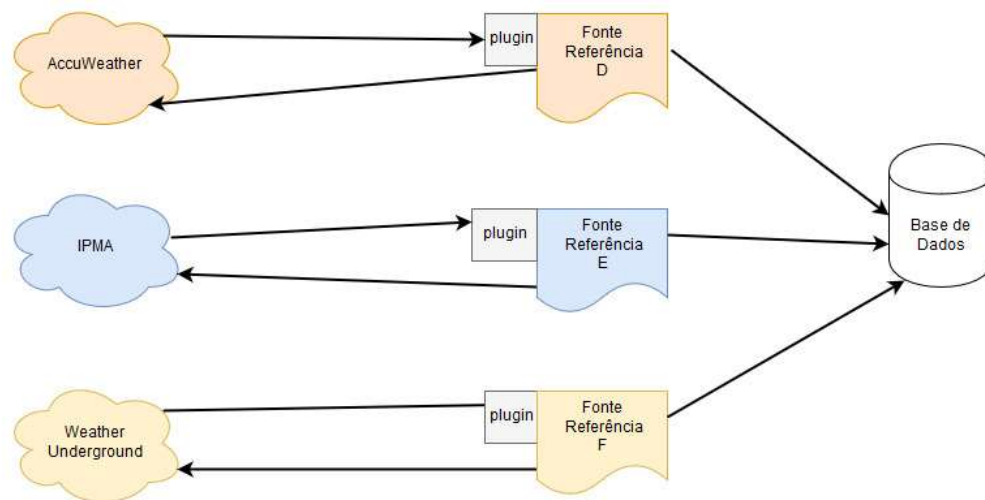


Figura 11 - Processo de aquisição de dados de referência

### 3.1.3. *Plugin* de Mapeamento dos Dados

O sistema pretende ser o mais heterogéneo possível e para que isso seja viável, será definida uma estrutura genérica, um *plugin*, que permitirá qualquer formato de dados ser adaptado para entrar no sistema.

Estes *plugins* permitem que os dados de qualquer fornecedor, ou sistema, possam ser interpretados e convertidos num formato que o sistema possa compreender e processar. O *plugin* faz a conversão dos dados das mensagens num formato em que as métricas são interpretáveis (conversão no esquema *JSON* do sistema), mas também implementa um conjunto de métodos que permitam converter a unidade de medida dos dados originais nos valores correspondente na unidade de medida definida para o sistema. Não é relevante qual

a linguagem de programação em que é implementado o *plugin* desde que o seu output seja um *JSON* com o formato e os parâmetros definidos.

Qualquer fonte de dados pode, após o registo no sistema, configurar que os dados a integrar no sistema são provenientes de um *plugin*. No registo da fonte é necessário indicar também quais os diferentes tipos de sensores, os seus limites do domínio e a variação máxima aceitável relativa ao domínio.

*Plugins* podem ser utilizados para a integração dos dados dos sistemas a analisar, mas também para a integração de dados provenientes de fontes de referências. As fontes de referência disponibilizadas, normalmente são obtidas a pedido, quando são necessárias, e requerem a implementação de um *plugin* específico para a recolha, interpretação e conversão dos dados de acordo com o utilizado pelo sistema.

Por exemplo, quando se utiliza como referência os dados disponibilizados pelo *Accuweather*, é necessário um *plugin* que interprete a mensagem *JSON* fornecida, e que traz um conjunto de parâmetros, individualize cada um dos valores e os transforme na unidade de medida usada pelo sistema. Note que nem todos os valores retornados pelo *Accuweather*, são relevantes para a análise e interpretáveis pelo sistema. A figura seguinte ilustra o exemplo de uma mensagem *JSON* disponibilizada pelo *Accuweather*.

```
[{
  "LocalObservationDateTime": "2017-02-26T22:55:00+00:00",
  "EpochTime": 1488149700,
  "WeatherText": "Mostly cloudy",
  "WeatherIcon": 38,
  "IsDayTime": false,
  "Temperature": {
    "Metric": {
      "Value": 8.8,
      "Unit": "C",
      "UnitType": 17
    },
    "Imperial": {
      "Value": 48.0,
      "Unit": "F",
      "UnitType": 18
    }
  },
  "MobileLink": "http://m.accuweather.com/en/pt/vila-nova-de-foz-coa/273408/current-weather/273408?lang=en-us",
  "Link": "http://www.accuweather.com/en/pt/vila-nova-de-foz-coa/273408/current-weather/273408?lang=en-us"
}]
```

Figura 12 - Mensagem *JSON* do *Accuweather*

Alguns dos dados, da mensagem da figura anterior, não são relevantes e por isso não são utilizados, como por exemplo, a medida da temperatura em *Fahrenheit*, pois graus centígrados é a unidade de medida de referência, assim como a informação sobre o

*EpochTime* ou o *WeatherIcon*. Além dos *plugins* específicos para as fontes de referência o objetivo é que a plataforma seja extensível e que permita a cada fornecedor de dados facilmente implementar um *plugin* para transformar os seus dados no *JSON schema* do sistema para tal, será disponibilizado a *API* com as informações genéricas do *plugin*.

### 3.1.4. Fonte Histórica

O histórico dos dados representa também uma fonte pois estes são usados, muitas vezes, como referência para validar e melhorar os dados em análise. Durante o processo de validação, também em sistema de *pull* há um processo que carrega os dados necessários para a efetuar as validações e calibrações definidas. De maneira semelhante à recolha dos dados de referência têm-se o seguinte esquema de funcionamento,



Figura 13 - Processo de aquisição de dados históricos

## 3.2. Estrutura e Formatos dos Dados

Como um dos objetivos do sistema é que seja genérico, e que possa ser utilizado por diversos sistemas, contextos e cenários, os dados são processados utilizando o formato *JSON*. Contudo, com o objetivo de clarificar e exemplificar a sua integração e utilização, será frequentemente utilizado o contexto de análise de dados meteorológicos que serviu de mote e de inspiração para o projeto. Para adicionar novas fontes de dados ao sistema deve usar-se o seguinte *JSON schema*

```

{
  "version": "http://system.isec.pt/Date_Schema",
  "title": "Add Source Schema",
  "type": "object",
  "properties": {
    "wsn": {
      "id": "number",
      "description": "string",
      "location": {
        "description": "coordenadas gps ",
        "properties": {
          "latitude": { "type": "number" },
          "longitude": { "type": "number" }
        }
      }
    }
  }
},

```

```

"nodes": {
  "uid": "text",
  "location": {
    "description": "coordenadas gps ",
    "properties": {
      "latitude" : { "type": "number" },
      "longitude": { "type": "number" }
    }
  },
  "measures": {
    "type": "array",
    "items": {
      "data": {
        "metric": { "type": "string" },
        "unit" : { "type": "string" }
      },
      "required": ["metric", "unit"]
    }
  },
  "minItems": 1,
  "uniqueItems": true
},
"required": ["id", "location", "description", "nodes"]
} } }

```

Depois do registo da fonte deve ser usado o esquema seguinte para enviar os dados para o sistema,

```

{ "version": "http://system.isec.pt/Data_Schema",
  "title": "Data Schema",
  "type": "object",
  "properties": {
    "wsn": {
      "id_system": "number",
      "nodes": {
        "uid": "text",
        "measures": {
          "type": "array",
          "items": {
            "data_acq": {
              "description": "Data de Aquisição",
              "type": "object",
              "data": {
                "metric": {
                  "type": "string"
                },
                "value": {
                  "type": "number"
                },
                "unit": {
                  "type": "string"
                }
              }
            },
            "required": ["metric", "value", "unit"]
          }
        },
        "minItems": 1,
        "uniqueItems": true,
        "required": ["id", "sensor_name"]
      }
    }
  }
}

```

Todos os ficheiros *JSON* devem incluir os seguintes parâmetros:

- *version* – a versão do *plugin* para que o sistema saiba qual o formato que vai tratar, com a evolução podem surgir novos formatos e desta maneira é possível indicar o correto;
- *Id\_system* – o identificador da fonte de dados no sistema;
- *Name* – o nome da fonte de dados;
- *Local* – o local onde os dados são recolhidos;
- *url* – o endereço do servidor de dados da fonte;
- *Nodes* – definição dos vários nós que constituem a fonte;
- *Measures* – conjunto de medidas obtidas num determinado instante de tempo;
- *Time\_acq* – instante temporal no qual a leitura do valor foi feita;
- *Data* – agrega informação sobre a leitura;
- *Type* – tipo de dados lidos;
- *Value* – valor lido pelo sensor;
- *Unit* – unidade de medida do valor lido.

### 3.2.1. Tipos de Métricas dos Sensores

Tipicamente cada estação, ou nó, envia periodicamente os dados dum determinado instante temporal, obtidos por ela e/ou por estações vizinhas que pertencem à mesma rede de sensores. Essa rede, as suas configurações e o esquema de dados das mensagens são inseridas e registadas no sistema como uma fonte de dados, para que, quando for recebido um novo conjunto de dados, estes possam ser corretamente catalogados e interpretados. Uma rede de sensores é constituída por um ou mais nós, instalados numa área geográfica próxima e que, frequentemente, concentram os seus dados recolhidos num nó coordenador, responsável por, com uma periodicidade pré-definida, enviar todos os dados para um servidor. Cada mensagem de dados está indexada temporalmente e contém os dados de vários tipos de sensores. No exemplo do cenário de recolha de informação meteorológica, a mensagem inclui dados tais como:

- Temperatura do Ar – a unidade de medida de referência no sistema, por defeito, é graus celsius (°C);
- Humidade do Ar – a unidade de medida de referência é a percentagem (%);
- Pressão Atmosférica – a unidade de medida de referência é Hectopascals (hPa);



- Radiação Solar – a unidade de medida de referência é watt por metro quadrado ( $Wm^2$ );
- Humidade do Solo – a unidade de medida de referência é a percentagem (%);
- Velocidade do Vento – a unidade de medida de referência é quilómetros por hora (km/h);
- Direção do Vento – a unidade de medida de referência são os Pontos Cardeais;
- Pluviosidade – a unidade de medida de referência é milímetros cúbicos;
- Humidade da Folha – a unidade de medida de referência é a percentagem (%).

Conjuntamente com os dados recolhidos, cada mensagem também inclui informação sobre o instante, ou *timestamp* (que inclui *AAAA-MM-DD hh:mm:ss* ) em que estes foram recolhidos, possibilitando assim o cruzamento com outros dados obtidos em instantes próximos, por este ou outros sensores.

### 3.2.2. Armazenamento de Dados

Os dados podem ser armazenados em ficheiros, *JSON* ou *XML*, ou em Base de Dados (BD). Devido ao volume, a necessidade acesso rápido e à importância dos dados nos processos do sistema nomeadamente validação, calibração e criação de zonas, recomenda-se a utilização de sistemas de bases de dados. A BD armazena a informação relativa à estrutura das fontes e também aos dados que estas recolhem, adicionalmente ainda necessita de armazenar a informação sobre as zonas e sobre os erros resultantes das validações, bem como quais os alarmes gerados e quais os utilizadores que os recebem.

As fontes de dados têm origens distintas e conseqüentemente informações também distintas, na perspectiva do sistema, todas possuem um conjunto de características e atributos comuns, pelo que é proposto um modelo de dados genérico que permita armazenar de forma transversal todos os atributos necessários ao funcionamento do sistema. Este modelo permite guardar fontes distintas de uma forma simples, evitando operações complexas e demoradas, nomeadamente derivadas das operações de *join* entre tabelas se, fosse utilizado um modelo com uma tabela específica para cada tipo de fonte.

Da mesma forma como os dados provenientes dos sensores, independentemente do seu tipo, apresenta uma estrutura de dados similares. Assim é proposta uma estrutura que

permita guardar o tipo de dados a armazenar, o valor lido, qual o instante em que foi recolhido, a origem desses dados (fonte de dados ou calibrados), e a sua avaliação qualitativa (válidos ou inválidos). De igual modo, a vantagem pela utilização de uma única estrutura genérica é a facilidade de acesso aos dados com *queries* simples e rápidas.

Os erros detetados a partir da análise do sistema estão definidos numa tabela para mais facilmente os catalogar, estes têm uma descrição associada à validação que deu origem ao erro, informação importante para enviar aos responsáveis pela manutenção dos sensores. Desta forma têm-se uma tabela onde são definidos os tipos de erros e outra onde são armazenados os erros dos sensores. É gerada uma entrada quando é detetado pela primeira vez um erro, contendo a taxa de erro, a descrição do erro, a data da primeira ocorrência e o estado. O estado varia entre aberto e fechado, aberto desde a primeira ocorrência até o sensor ser intervencionado, quando ocorre uma intervenção e o problema é resolvido o estado passa a “fechado” e nenhum dos restantes valores é mais alterado. Até fechar o erro, são atualizados de cada vez que uma nova falha é detetada nesse sensor, a taxa de erro e a descrição, se ainda não constar da descrição o erro ocorrido.

Os utilizadores que recebem os alarmes também devem ter os seus dados registados no sistema de modo a que sejam reconhecidos como responsáveis pelas fontes e sejam notificados com *emails* e *SMS* dos erros encontrados nos dados.

### 3.2.3. Classificação dos Dados dos Sensores

Os dados recebidos de Fontes em Análise são validados através da aplicação das diversas regras definidas para o sistema. Os mecanismos de especificação de regras estão definidos no capítulo 4. Após a validação e aplicação de ações aos dados, conjuntamente com cada valor armazenado na base dados, é atribuída uma classificação qualitativa e da origem desse dado.

Como os dados podem ser calibrados, isto é, os valores reais obtidos pelos sensores, podem ser ajustados de acordo com fontes de referência, é necessário, quando se guarda, saber se esse dado é o obtido pelo sensor (real), é o estimado (quando existe falha de transmissão das leituras) ou é um valor calibrado (após a aplicação de fórmulas de calibração).

Além disso, após os dados serem validados, é-lhes atribuída uma classificação qualitativa:

- Válido – valor considerado válido, após verificação das regras, e que pode ser usado para análise;
- Inválido – valor considerado inválido, após verificação das regras.

### 3.3. Validação dos Dados Integrados

Os dados de uma fonte, após serem integrados, através de *plugins* ou diretamente no sistema, passam por diversas fases, ou tipos, de validação antes de serem utilizados para a análise e, guardados no sistema para análise futura. A figura 14 ilustra essas fases desde que os dados são integrados no sistema.

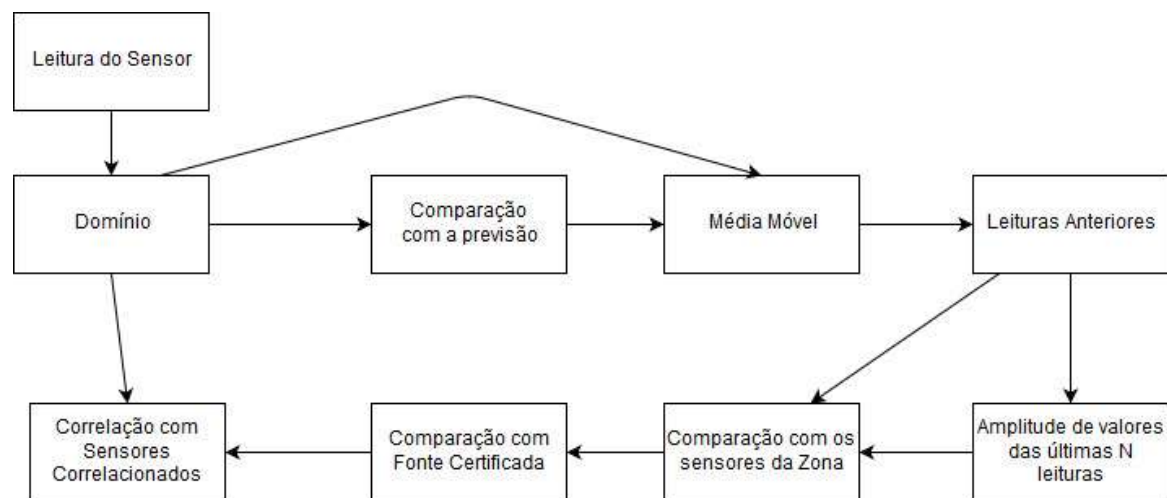


Figura 14 - Tipos de validação do sistema

De acordo com o seu tipo e objetivo, as validações podem ser simples, por exemplo a Validação de Domínio onde é verificado se o valor recebido está dentro do intervalo de valores espectáveis, ou mais complexas, recorrendo à Validação por Comparação e Correlação, comparando os valores recolhidos com os dados obtidos pelos nós vizinhos, do mesmo tipo ou de tipos diferentes, mas com uma relação conhecida entre eles.

Conjuntamente com a validação dos valores dos sensores, também é validado se o *timestamp* associado à informação recolhida está no formato interpretado pelo sistema e se

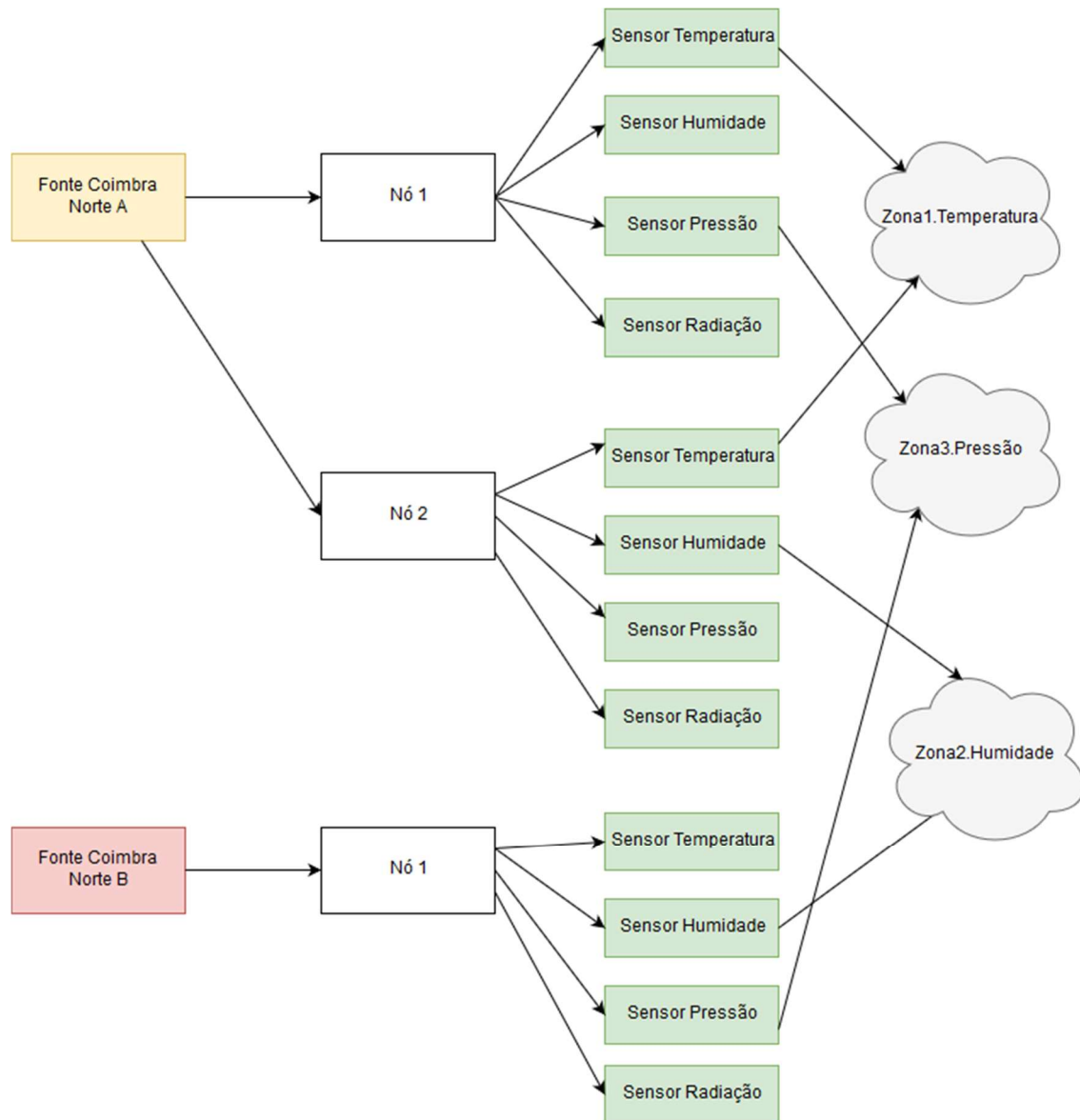
a informação temporal é coerente, por exemplo, se a data de recolha não é mais antiga do que quarenta e oito horas e se a data não é posterior à data atual do sistema.

No capítulo 4, são apresentados alguns dos tipos de validação considerados. No entanto, o sistema é aberto e extensível, permitindo a inclusão futura de novas formas de validação.

Como o tipo de validação a efetuar aos dados é variável de sensor para sensor, e de sistema para sistema, e como a complexidade da validação também pode variar, é proposto um mecanismo de validação dinâmico expresso através da especificação de regras de validação.

### 3.3.1. Definição de Zonas

Uma Zona é uma área geográfica, povoada por sensores que devido à sua proximidade podem ser referência uns dos outros. As Zonas são criadas com base nos tipos de sensores e na medida que estes adquirem. Os sensores podem não ser vizinhos, geograficamente falando mas, apesar de distantes, apresentarem uma distribuição de dados semelhante. A análise pretende ser abrangente, podendo ser definida a distância máxima entre os elementos da Zona. A determinação das Zonas é em função dos sensores e não dos nós em que eles se encontram. Isto é, um nó pode aparentemente pertencer a vários Zonas distintas, no entanto, cada um dos seus sensores apenas pertence a uma única Zona. A figura 15 ilustra como se podem aglomerar os sensores dos mesmos nós em Zonas distintas.



**Figura 15 - Exemplo de Zonas**

Manualmente podem ser definidos quais os sensores que se pretendem adicionar a cada Zona. Além disso, o sistema pode determinar, através da aplicação de algoritmos de *clustering* aos dados recolhidos, quais as zonas e os sensores que pertencem a cada uma.

A validação por Zona assenta em dois tipos, Zonas criadas automaticamente por algoritmos que analisam a distribuição dos dados ao longo do tempo e Zonas criadas manualmente.

Uma Zona pode ser definida manualmente, no *front-end* do sistema pelos utilizadores que consideram que estes podem ser referência uns dos outros, por exemplo por serem próximos geograficamente ou por apesar de não estar próximos pertencerem ao mesmo utilizador e este achar a análise relevante. Nem todos os sensores estão disponíveis

para serem incluídos nas Zonas, a sua visibilidade por outros utilizadores do sistema é determinada pelos seus administradores.

A interface do sistema permitirá aos utilizadores a criação de Zonas para cada tipo de sensor. O utilizador depois de registar a sua Fonte de Dados deve aceder às configurações da mesma para escolher a análise que será feita aos dados, ao escolher a validação por comparação com os “vizinhos” deve criar uma Zona com a lista de vizinhos que lhe parecem relevantes para cada tipo de sensor. Daí em diante estas Zonas vão ser usadas para efetuar a validação dos dados recebidos pela Fonte. Estas Zonas podem ser alteradas pelo sistema, quando um sensor fica indisponível este é removido das Zonas às quais pertencia e os utilizadores são notificados da alteração.

### 3.3.2. Determinação Automática de Zonas

A criação automática de Zonas será feita através da análise do histórico de dados de cada tipo de sensores. Para esta análise são usados algoritmos “inteligentes” que permitam identificar os sensores que ao longo do período de tempo da amostra mantiveram o mesmo comportamento.

A criação automática de Zonas também terá alguns parâmetros configuráveis como o intervalo em análise, a frequência com que essa análise deve ser repetida, se a Zona deve ser considerada automaticamente ou se é necessário a aprovação pelo administrador do sistema.

#### 3.3.2.1. Intervalo dos Dados

O intervalo de dados deve ser contínuo admitindo uma percentagem de falhas não superior a 10% e deve ter pelo menos 7 dias de dados para que se possa estabelecer com alguma certeza a similaridade. Este *set* de dados deverá ser analisado pelos algoritmos “inteligentes”.

#### 3.3.2.2. Frequência de Análise

Os dados devem ser analisados periodicamente para identificar possíveis alterações nos dados que levem a alterações nas Zonas criadas. Para que seja mais eficaz deve ser possível armazenar uma taxa de erro acumulado para que, ao atingir o limite a zona seja reavaliada. No limite a cada novo dado recebido pode realizar-se uma nova análise, no entanto não é expectável que hajam grandes alterações entre períodos muito curtos.

### 3.3.2.3. Algoritmos

Para criar as Zonas serão utilizados algoritmos que permitem analisar a semelhança entre diversos *sets* de dados e agrupá-los quando estes apresentarem uma alta similaridade. São utilizados algoritmos de *clustering*, que tomando em consideração a similaridade dos dados recolhidos pelos vários sensores, para determinar os *clusters* de nós com comportamento semelhante. Dos algoritmos de *clustering* analisados, destacam-se o *Kmeans* e o *k-NN* (*k-Nearest Neighbors Algorithm*).

O algoritmo *Kmeans* tem como objetivo criar *clusters* agrupando os dados que apresentam maior semelhança, tomando em consideração o número de *clusters* que é definido *à priori*, e que para esse número de *clusters* calcula os seus centroides que devem estar distantes uns dos outros. Cada ponto deve ser associado ao centroide mais próximo, depois de todos estarem associados a um centroide este deve ser recalculado tendo em consideração todos os elementos do *cluster*. Os dois últimos passos devem ser repetidos até que não hajam alterações nos *clusters*, considerando-se assim esses os clusters finais. Este algoritmo é frequentemente usado para prever fenómenos meteorológicos.

O *k-Nearest Neighbors Algorithm Classification (k-NN)* é um algoritmo de aprendizagem que usa um exemplo de treino para classificar o *set* de dados. Cada elemento do *set* é comparado com os elementos do exemplo de treino e, é-lhe atribuída a classe da maioria dos seus vizinhos. Essa comparação pode ser feita recorrendo a medidas de distância, por exemplo a Euclidiana ou de *Hamming* com os seus *k* vizinhos, sendo *k* o número de vizinhos com o qual é comparada a distância. O elemento herda a classe da maioria dos seus vizinhos.

Após a análise dos algoritmos, e tomando em consideração os dados que são analisados, por defeito o sistema utiliza o algoritmo *k-NN* pois é de fácil implementação, apresenta uma complexidade computacional aceitável para o sistema em causa, permite analisar a distribuição de dados numéricos, sendo o mais indicado para *datasets* pequenos.

## 3.4. Calibração dos Dados

Os dados após a validação são catalogados como válidos ou inválidos e para minimizar as perdas estes são comparados com as fontes de referência e, no caso de apresentarem desvios da fonte de referência estes devem ser analisados, não só

pontualmente com os valores do mesmo espaço temporal, mas de uma perspectiva global com o intervalo de tempo superior, ajustando a cada métrica o intervalo relevante para a análise. Existem métodos matemáticos que permitem minimizar a distância entre conjuntos de dados, um desses métodos é o dos Mínimos Quadrados detalhado de seguida.

### 3.4.1. Método dos Mínimos Quadrados

O Método dos Mínimos Quadrados [10] permite aproximar um conjunto de outro minimizando a soma dos quadrados das diferenças entre o valor em análise e o valor observado. Ou seja, através de duas amostras de dados com o mesmo número de elementos e a mesma janela temporal, uma da Fonte de Referência e outra da Fonte em Análise podem dar origem a uma terceira que aproxima a Fonte em Análise da de Referência. A função que permite esta é:

$$g(x) = a_0 + a_1x$$

Sendo  $a_0$  e  $a_1$  coeficientes reais que permitem minimizar as distâncias entre os dois grupos. Para obtermos esses valores teremos de resolver o seguinte conjunto de equações:

$$g(x) = \begin{cases} a_0 \sum n + a_1 \sum x = \sum y \\ a_0 \sum x + a_1 \sum x^2 = \sum xy \end{cases}$$

Sendo,  $n$  o número de pontos do conjunto dos valores de Referência,  $x$  o conjunto dos valores de referência e  $y$  o conjunto dos valores em análise. Após a resolução do sistema de equações temos o valor de  $a_0$  e de  $a_1$  que nos permitem calcular os pontos do conjunto aproximado, os valores obtidos passam a ser a referência para uso futuro, constituindo uma nova Fonte obtida das duas anteriores.

### 3.4.2. Calibração Parcelar

Uma outra forma de calibração é a Calibração Parcelar, apenas alguns períodos apresentam falhas, no caso dos sensores meteorológicos por vezes surgem erros apenas em determinadas condições, como por exemplo a incidência do sol sobre os sensores. Um determinado sensor de temperatura exposto diretamente ao sol aquece e quando a leitura é feita o valor é demasiado elevado comparativamente à Fonte de Referência, por outro lado



nos períodos em que o sol não incide no sensor a temperatura é similar à Fonte de Referência. Daí advém a necessidade de apenas ajustar os valores durante o período de maior exposição, ou seja, o intervalo de dados a analisar deve ser coincidente com o período em que a exposição altera significativamente os valores recolhidos, devendo ser usados os períodos semelhantes dos dias anteriores para mais facilmente obter a melhor aproximação que minimiza o erro causado pelo fator identificado.

Por exemplo, pode ser especificada uma regra, em que, no intervalo de tempo definido (por exemplo entre as 10 e as 16 horas), quando a validação com a fonte de referência for superior a um determinado valor (definido pelo utilizador), os valores obtidos nesse intervalo de tempo devem ser calibrados (ajustados) tomando em consideração a sua fonte de referência.

### 3.4.3. Calibração por Extrapolação dos Valores Conhecidos

A Calibração através da extrapolação dos valores conhecidos pretende colmatar pequenas falhas de aquisição, cada sensor tem uma frequência de aquisição definida, mas por diversos motivos algumas podem não chegar ao sistema. Quando ocorrem estas falhas pontuais num sensor, estas podem ser minimizadas se existirem dados nas aquisições anteriores e posteriores que permitam estimar o valor em falta usando a média entre o antecessor e sucessor. Para tal deve ser feita uma análise diária, das últimas 24 horas e “preencher” as falhas sempre que hajam dados que permitam essa estimativa. Este novo valor será classificado como uma estimativa e será usado como histórico para uso em análises futuras.

Após a deteção de mensagens em falta, gera essa mensagem, como uma nova, através da média dos valores (antes e depois) e volta a ser “injetado” na fonte de dados. Esta nova mensagem, posteriormente, segue o fluxo normal de validação.

## 3.5. Ações a Tomar

Após a análise dos dados, dependendo do tipo de resposta que o sistema obtém dos dados, várias ações podem ser tomadas tais como rejeição dos dados ou envio de alertas. Seguem-se os detalhes dessas mesmas ações.

### 3.5.1. Rejeição da Leitura em Análise

Quando uma leitura tem uma data de aquisição posterior à data atual esta leitura é totalmente descartada pois indica que a comunicação sofreu algum problema que alterou os dados indexando no futuro as leituras realizadas.

### 3.5.2. Rejeição da Métrica em Análise

Quando um valor não se encontra dentro do domínio da métrica este é descartado por se considerar que não é possível aplicar qualquer correção a este valor para se aproximar do seu ambiente de operação. O valor em causa é armazenado com a classificação de inválido.

### 3.5.3. Considerar a Leitura Anterior

Quando há uma falha detetada na leitura atual, de domínio ou outra, pode ser considerada a imediatamente anterior uma vez que estas não deverão divergir muito uma da outra, historicamente é possível definir que a probabilidade de duas leituras consecutivas terem o mesmo valor é elevada.

### 3.5.4. Remover um Sensor de um Nó

Ignorar todas as leituras de um ou mais sensores de um Nó por estes apresentarem uma taxa de erro demasiado elevada. Até que haja uma intervenção técnica, as suas leituras são descartadas.

### 3.5.5. Remover o Nó

Todos os valores lidos por um Nó serão descartados, isto pode acontecer quando os erros são detetados em todas as métricas ou a indexação temporal das mensagens é incorreta indicando que o sistema de processamento das leituras desse nó pode estar danificado e por isso mesmo, até que seja feita uma intervenção técnica, os seus dados serão desconsiderados.

### 3.5.6. Redefinir Zonas Automáticas

As zonas são definidas periodicamente de forma automática, no entanto quando um sensor que compõe a zona apresenta problemas, este deve ser retirado da Zona para não “contaminar” o estudo dos restantes dados. Até que o sensor em causa seja intervencionado por uma equipa técnica este deve ser retirado de todas as zonas às quais pertença.

### 3.5.7. Retirar Nó da Zona

As zonas definidas pelos utilizadores também devem ser redefinidas, os sensores que apresentem problemas devem ser removidos das zonas e o utilizador deve ser notificado da indisponibilidade temporária daquele sensor.

### 3.5.8. Calibrar o Sensor

Quando um sensor apresenta pequenos desvios das fontes de referência ou dos restantes vizinhos ou, quando este tem pequenas falhas nos dados, deve ser feita uma análise baseada no histórico dos sensores para que se obtenha uma forma de aproximar os dados do Sensor em Análise dos Dados de Referência aumentando a eficiência do Sensor. A calibração será detalhada mais à frente no ponto 3.6.

### 3.5.9. Notificar Utilizadores

Os utilizadores devem ser notificados pelo sistema quando os seus sensores apresentam problemas ou quando um sensor que eles tenham escolhido como referência apresente problemas. O sistema deverá, mediante os dados recebidos ao longo do tempo, detetar anomalias e reportá-las. Estas anomalias estão relacionadas com as validações definidas e darão origem a alguns alertas tais como:

- Valores fora do domínio – quando um sensor tem uma taxa de erro referente a valores fora do domínio superior a 25% os utilizadores responsáveis pela Fonte à qual o sensor pertence recebem um email a informar da taxa de erro do sensor. Consoante o crescimento da taxa de erro do sensor o alarme vai sendo mais frequente até ao Sensor ser colocado como indisponível.
- Valores díspares da referência – quando um sensor repetidamente apresenta valores muito distantes da Fonte de Referência atingindo uma taxa de erro de 25% os

utilizadores responsáveis pela Fonte à qual o sensor pertence recebem um email a informar da taxa de erro do sensor e a descrição do(s) erro(s) associado(s).

- Número de aquisições inferior ao mínimo admissível – quando o sensor apresenta falhas na aquisição de valores superiores a 25% os utilizadores responsáveis são notificados da falta de dados por parte do sensor.
- Sensor indisponível na própria fonte – quando um Sensor apresenta uma taxa de erro superior a 50% é colocado como indisponível e notifica os seus responsáveis da ocorrência, informando qual o erro que levou à desconsideração do Sensor.
- Sensor removido de Zona criada pelo utilizador – quando um sensor fica indisponível, se este pertencer a Zonas criadas manualmente, os utilizadores que definiram estas Zonas devem ser notificados da remoção do sensor da sua lista de validação.

#### 3.5.10. Retirar o Nó ou um Sensor de um Nó

Quando um Sensor de um Nó ou todo Nó apresenta problemas consecutivamente este deve ser desconsiderado até que uma intervenção de reparação seja efetuada. Podem acontecer vários cenários que levem à rejeição dos dados vamos dar dois exemplos, um Sensor de um Nó ter uma taxa de erro superior a 50%, independentemente da causa, ou todos os dados desse Nó estarem indexados numa data posterior à atual indicando que aquele Nó tem problemas no processamento dos dados recolhidos pelos sensores.

Para que os dados sejam os mais assertivos possíveis, as fontes que apresentam erro e que podem influenciar a validação de outros, devem ser removidos da(s) zona(s) a que pertencem. As zonas devem ser redefinidas de modo a excluir o Nó que apresenta problemas. Este Nó fica marcado como “fora de serviço” e, os seus dados, voltarão a ser considerados pelo algoritmo de criação automática de zonas depois do Nó ter sofrido uma intervenção técnica, ter enviado dados e estes terem passado pelos processos de validação definidos. Cumpridas estas metas o Nó passa novamente a estar disponível para integrar as Zonas existentes.

Neste processo são retiradas das Zonas, criadas pelo sistema ou manualmente definidas pelo utilizador, recebendo estes uma mensagem a informar que temporariamente aquela Fonte não pode servir de referência por estar fora de serviço.

## 4. Mecanismos de Validação de Dados

Neste capítulo são descritos alguns dos tipos de validação considerados, e é proposto um mecanismo que permite especificar essas validações através da definição de regras. Para facilitar o processo de descrição das regras, em cada tipo de validação são apresentados exemplos de regras que permitem a sua especificação.

### 4.1. Regras de Validação

As regras de validação permitem aferir a qualidade da informação recebida dos sensores. Estas são expressas numa pseudo-linguagem que permite especificar, as condições e validações a efetuar aos dados de cada fonte de dados. As regras seguem o seguinte formato:



Figura 16 - Formato genérico de validação

Ou seja, por fonte(s) de dados, especifica-se as condições e as consequentes ações a aplicar aquando da chegada de uma mensagem de dados. Quando os dados chegam, estes são analisados de acordo com as regras e, são tomadas as ações especificadas quando estes satisfazem as condições expressas nas regras.

Tomando como exemplo a temperatura, uma regra para especificar que é inválido o valor obtido pelo sensor de temperatura de determinado nó de uma fonte de dados, quando o valor está fora do intervalo [10;20], pode ser especificada das seguintes formas:

```
ON Fonte.No WHEN Fonte.No.temperatura NOT BETWEEN [10; 20] THEN invalid;  
ON Fonte.No WHEN temperatura > 10 OR Fonte.No.temperatura < -20 THEN invalid;  
ON Fonte.No WHEN temperatura NOT WITHIN (10,20) THEN invalid; stop process;  
ON Fonte.No WHEN temperatura WITHIN (10,20) THEN valid; next validation;
```

Sendo *invalid* a classificação com a qual o valor será armazenado. Os valores inválidos serão armazenados porque estes podem ajudar no processo de recuperação do sensor quando é realizada uma intervenção técnica, mas também, porque a ocorrência de fenómenos raros pode fazer com que um valor fora do domínio ocorra e desta forma é possível, manualmente o responsável pela Fonte de Dados alterar a classificação do valor.

Como será explicado mais à frente serão criadas Zonas que serão constituídas por Sensores que pertencem a Nós que por sua vez pertence a uma Fonte. Todos os sensores pertencem a pelo menos uma zona, mas poderá pertencer a várias, criadas automaticamente através da similaridade das leituras, ou manualmente especificadas pelo utilizador. Esta definição de zonas é útil no processo de validação dos dados, pois permite aferir se os dados provenientes de um sensor divergem significativamente dos dados recolhidos pelos restantes sensores da zona a que pertence. A referenciação dos vários elementos a ser considerados toma o formato de:

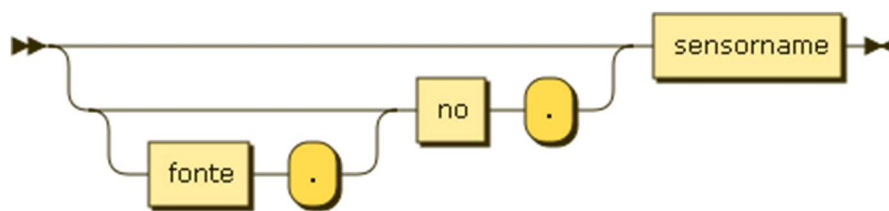


Figura 17 - Representação de métrica pertencente a um nó e a uma fonte

Sendo que a zona é opcional, e que permite apenas referenciar e agregar vários nós que podem pertencer a fontes diferentes da mesma zona geográfica. Isto segue um modelo hierárquico, em que cada nível (ex. nó) referências as características do nó em concreto.

Algumas das regras serão um pouco mais complexas e para isso é conveniente definir algumas nomenclaturas:

Fonte - Representa a fonte de dados da leitura recebida.

Fonte[n] - Representa um subconjunto das fontes de dados. Que podem ser enumeradas, por exemplo, Fonte[4,7,9] para especificar essas fontes

Fonte[\*] - Representa todas as fontes de dados.

Todas as fontes são compostas por Nós:

No - Representa um Nó que pertence a uma fonte de dados.

No[n] - Representa um conjunto de Nós, podem pertencer à mesma fonte de dados ou a fontes diferentes.

No[\*] - Representa todos os Nós do sistema.

Cada nó pode ter um ou mais sensores:

Sensor - Representa o Sensor que pertence a um Nó.

Sensor [n] - Representa n-últimas leituras do Sensor.

Sensor [intervalo] - Representa o conjunto das leituras recebidas naquele intervalo.

Um nó pode conter um ou vários sensores que recolhem informação do ambiente, mas apenas um de cada tipo. No entanto cada sensor apenas recolhe uma única métrica (ex. temperatura, humidade, ...). Para facilitar o processo de especificação das regras, não é expresso qual o sensor em si, isto é, as suas características físicas (ex. modelo, fabricante, ...) mas apenas referenciado a métrica, ou tipo de medida, que recolhe. Por exemplo, `No[1].temperatura` referencia o sensor de temperatura que está acoplado ao `No[1]`.

Como os sensores recolhem informação ao longo do tempo, para referenciar as várias leituras recolhidas, foi definida uma nomenclatura similar à usada para fontes e nós. Assim, para um determinado nó, e por exemplo, para o sensor `temperatura`:

`temperatura` - Representa a leitura mais recente do Sensor de temperatura desse nó.  
`temperatura [n]` - Representa um conjunto das últimas leituras do sensor de temperatura.  
`temperatura [*]` - Representa todas as leituras do Sensor `temperatura` do nó.

Os sensores podem ser agrupados em Zonas por isso o sistema também inclui a sua representação:

`Zona` - Representa uma zona de uma determinada métrica.  
`Zona[n]` - Representa um conjunto de Zonas.  
`Zona[*]` - Representa todas as Zonas do sistema.

Para além dessas designações é possível definir variáveis que ajudem a referenciar valores que são usados com mais frequência, vamos usar uma nomenclatura semelhante ao usado na programação. Por exemplo definimos *Early Morning* como sendo um período inicial da manhã, para analisar de forma mais concreta necessitamos de definir o intervalo de tempo concreto que este representa, intervalo este que varia consoante a estação do ano, no verão têm-se,

```
DEFINE EARLY_MORNING = [time 6 .. 9 hour]
```

Podendo ser usado para representar que a temperatura naquele intervalo é inferior a 20 graus,

```
WHEN temperatura[early_morning] < 20 THEN valid;
```

Ou ainda quando é necessário definir um conjunto de vinhas podemos ter,

```
DEFINE vinh1 = FONTE[4,7,9];
```

```
DEFINE zonal_temp = FONTE['vinh1'].NO[1,2,3].temperatura [1-4]
```

Se quisermos representar a validação de uma métrica, por exemplo temperatura pode representar-se da seguinte maneira:

```
FOR VINHA WHEN temperature NOT BETWEEN [10;20] invalid AND stop process;
```

Combinando estas nomenclaturas com operações conhecidas como “WHEN”, “OR”, “AND”, “THEN” ou com funções como o BETWEEN ou o NOT BETWEEN é possível representar de forma mais esquematizada as operações que são efetuadas sobre os dados.

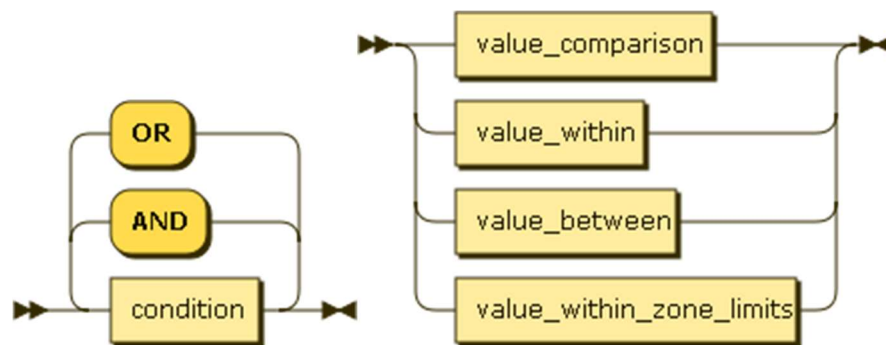


Figura 18 - Exemplos de operadores e condições das regras

Quando se pretende concretizar uma determinada Fonte de Dados pode usar-se o nome dessa fonte ou o seu identificador único, por exemplo queremos referir-nos a uma Fonte cujo nome é Vinha\_Cantanhede e o seu identificador único é o número 12 podemos fazê-lo de duas maneiras, usando a nomenclatura FONTE[‘Vinha\_Cantanhede’] ou FONTE[12], de forma análoga para os Nós podemos ter FONTE[‘Vinha\_Cantanhede’].NO[‘Norte’] ou FONTE[‘Vinha\_Cantanhede’].NO[1] seguindo-se o tipo de sensor que se pretende selecionar, tendo assim FONTE[‘Vinha\_Cantanhede’].NO[‘Norte’].Temperatura.

Para enquadrar os dados temporalmente teremos também algumas palavras-chave que poderão ajudar na definição das regras:

```
Early_Morning - período inicial da manhã, [time 10 .. 12 hour]
Late_Morning - período final da manhã
Early_Afternoon - período inicial da tarde
Late_Afternoon - período final da tarde
Early_Night - período do início da noite
Late_Night - período do fim da noite
```

Mais à frente serão adicionados exemplos específicos de cada validação.



Pretende-se que as regras sejam genéricas para que se possa adicionar novas regras de acordo com as necessidades. As regras classificam o valor de válido ou inválido consoante o teste realizado, como as regras por vezes apresentam diferentes resultados é necessário ponderar os resultados e obter a classificação final. Dependendo da classificação algumas ações serão tomadas, estas foram discutidas na secção 3.5, interessa para as regras de forma genérica definir *stop process*, como fim do processamento de regras, e o *next validation*, como indicador que a avaliação prossegue com a regra seguinte.

O *stop process* pode aparecer apenas no final de todas as regras terem sido verificadas ou quando se achar suficiente as validações já executadas. O *stop process* é uma função que avalia os resultados obtidos nos testes realizados e encaminha para as ações a realizar mediante a validação. Por exemplo existem algumas validações mais relevantes como a comparação com a Fonte de Referência, se o valor em análise passar nessa validação então pode terminar logo a validação. Nos casos em que o Valor em Análise foi verificado por várias validações e nem todas tiveram o mesmo resultado é necessário verificar os resultados e combiná-los entre si dando prioridade aos mais credíveis. Por exemplo um valor que falhou a validação por média móvel, mas passou na validação por diferença com o seu antecessor e ainda na validação por Fonte de Referência deve ser considerado válido e armazenado com essa classificação. Vamos então detalhar as diversas regras de validação já identificadas, sendo que, outras podem ser acrescentadas ao sistema.

## 4.2. Validação do Domínio

Para além desta validação transversal a todos os tipos de sensor temos ainda as validações específicas de cada tipo, nomeadamente o seu domínio, valores máximos e mínimos aceitáveis para cada tipo de sensor tendo em consideração o seu local de instalação. Dependendo do tipo de sensor e do ambiente onde está inserido os valores de domínios são fixos ou variáveis. Por exemplo a Humidade relativa varia 0% e 100% qualquer que seja o ambiente em que esteja inserido o sensor, qualquer valor que esteja fora desse domínio não é válido e requer uma análise posterior para detetar a origem do problema. Noutros tipos de métricas como a Temperatura a gama de valores é bastante dependente do lugar onde o sensor está instalado, em Portugal ou no Equador as gamas admissíveis são bastante diferentes. Em Portugal Continental este oscila entre -10°C e 45°C, portanto todas as recolhas feitas deverão estar inseridas nesse intervalo, sendo mais

específico os valores de referência no Algarve são diferentes da Serra da Estrela. Os dados se encontrem fora do domínio, mas dentro do funcionamento espetável do sensor devem ser analisados manualmente, caso tenha ocorrido algum fenómeno extremo, os valores do domínio deverão ser atualizados. Traduzindo para regras teremos algo como:

```
WHEN fonte['portugal'].temperatura NOT BETWEEN [10,20] THEN invalid;
WHEN fonte['equador'].temperatura NOT BETWEEN [15;45] THEN invalid;
```

Após a verificação do domínio é feita a validação seguinte apenas, para os dados que passarem este primeiro teste, os restantes como já referimos são armazenados e o seu estado apenas é alterado manualmente. As validações são processadas sequencialmente, seguindo a ordem em que foram introduzidas. Exceto, quando a ação definida é *stop process* caso contrário as regras são consecutivamente avaliadas.

### 4.3. Validação com o Histórico do Sensor

Além das validações de domínio, que são muito dependentes das características dos sensores utilizados, também outras podem ser muito úteis para verificar a validade dos dados nas condições em que estas foram obtidas e que têm em consideração o ambiente de operação e outros tipos de medição com os quais este se pode relacionar. Os dados históricos já recolhidos também são utilizados para validar os atuais.

#### 4.3.1. Validação por Comparação com a Previsão

Através da análise dos dados históricos de um determinado sensor é possível prever um valor aproximado da próxima leitura usando algoritmos de previsão de dados meteorológicos, como por exemplo, *Sliding window algorithm* estudado em alguns artigos científicos [11,12]. Mediante a determinação de padrões dos dados históricos é possível prever com algum grau de confiança os valores estimados que se irão obter nas leituras seguintes. Em termos de regras têm-se,

```
WHEN temperatura NOT WITHIN (PREDICTION(temperatura, 10),2) THEN invalid;
```

onde *PREDICTION*, considera as últimas 10 leituras (tamanho da janela) para determinar a estimação da próxima leitura. Usando o *WITHIN* valida se a leitura apresenta um desvio inferior a 2°C relativamente ao estimado.

### 4.3.2. Validação por Média Móvel

Um indicador importante para a validade dos dados é a diferença ao seu antecessor, a média móvel permite estimar a tendência dos valores para tal deve estudar-se as últimas N leituras. O valor de N deve ser ajustado consoante o tipo de métrica em análise e a amplitude dos dados. Pode representar-se da seguinte forma:

```
WHEN temperatura NOT BETWEEN [ 0; 20] THEN invalid;  
  
WHEN temperatura NOT WITHIN (AVG(temperatura [10]), 2) THEN invalid;
```

Onde *AVG* representa uma função que calcula a média dos últimos 10 elementos e o *WITHIN* verifica se o valor em análise é próximo da média com um desvio máximo de 2.

### 4.3.3. Variação entre Leituras Consecutivas

A aquisição de dados realiza-se em períodos regulares, tipicamente de 15 em 15 minutos, que pode variar de sistema para sistema. Através da análise dos dados, e para um determinado desvio admissível entre duas leituras de dados consecutivas, pode-se determinar se a leitura é um valor espectável ou não, não podendo este ter uma variação superior ao especificado. Por exemplo, para um desvio máximo admissível de 2° a regra seria a seguinte

```
WHEN ABS(temperatura - temperatura [1]) > 2 THEN invalid;
```

Sendo *ABS* o módulo da diferença entre as duas temperaturas.

A variação também pode ser estudada em períodos mais ampliados, comparar com dias anteriores ou mesmo semanas anteriores, a maioria dos eventos que se querem medir tem comportamentos cíclicos. Quando o histórico de dados já tem um volume considerável é possível encontrar vários padrões que ajudem na validação dos dados.

### 4.3.4. Amplitude de Valores das Últimas N Leituras

A comparação com os limites máximos e mínimos das últimas N leituras permite estreitar a janela de valores admissíveis para um dado intervalo, dependendo da fase do dia, ou seja, definindo os perfis dos dados de acordo com a evolução de cada tipo de métrica ao longo do dia. Por exemplo por volta do início da tarde a Radiação Solar deve atingir o seu

valor mais elevado sendo o período do início da manhã e do fim da tarde os valores mais baixos até atingirem 0, se surgir o valor mais elevado do dia às 20h00 seguramente é um valor que não corresponde à realidade. Temos então a seguinte regra:

```
WHEN radiacao_solar WITHIN (DIFF(radiacao_solar , radiacao_solar [1..10]),
<0)
AND (hour BETWEEN [10 ; 14] THEN invalid AND next process;
```

Sendo *DIFF* a diferença entre o valor atual e o seu antecessor. *Hour* representa a hora da leitura.

#### 4.4. Validação com os Dados dos Nós Vizinhos (por zona)

A validação por comparação com os valores obtidos pelos sensores dos nós vizinhos consiste na definição de um novo intervalo de valores expectável para o sensor em estudo. A partir dos valores lidos pelos sensores vizinhos durante o intervalo de 10 minutos que antecederam a leitura atual é possível comparar o valor recebido com estes, através do valor mínimo e do máximo obtido por eles e da variação das suas leituras. Como os nós podem conter vários sensores, este desvio é definido por sensor e não por nó. A definição dos nós vizinhos é feita através da criação de zonas automáticas ou zonas criadas pelos utilizadores definido na secção 3.5.3. Quando uma nova leitura de dados chega ao sistema, esta será avaliada com uma regra semelhante à seguinte:

```
ON fonte[1].no[*].temperatura WHEN IS_WITHIN_ZONE_LIMITS(temperature) THEN ...
```

Onde `IS_WITHIN_ZONE_LIMITS(temperatura)` verifica para o sensor em concreto (ex. Sensor temperatura do nó da leitura a ser validada) se está dentro dos limites das zonas a que este sensor pertence.

#### 4.5. Validação por Comparação com Fonte Certificada

A Validação por comparação a uma fonte certificada ou de referência permite dar mais confiança aos dados, ao contrário das validações anteriores, que podem não ser fidedignas, pois, por exemplo, se vários nós vizinhos possuírem um sensor da mesma métrica com o mesmo defeito, não é possível detetar variações anormais, pois todos irão apresentar a mesma gama de valores. Este potencial efeito, pode ser minimizado através da validação com uma fonte de dados certificados, ou oficial. Para tal é necessário ter uma

Fonte de Dados Oficial próxima geograficamente e que tenha uma frequência de aquisição semelhante, ou superior. Sempre que os tempos de aquisição não sejam coincidentes, e para tornar a aproximação mais fiável, pode ser calculado o valor expectável para aquele ponto temporal calculando a média ponderada entre as duas aquisições mais próximas do ponto que se pretende obter. Alguns parâmetros que devem ser tidos em consideração, dependendo do tipo de sensor, são a distância entre os sensores, a altitude a que o sensor está instalado, nos sensores mais suscetíveis à exposição solar a presença de obstáculos que possam interferir na aquisição. Estes fatores determinam se a referência pode ser usada para o sensor em análise e, quando este for referência qual o offset máximo entre eles. Caso a Fonte Certificada cumpra os requisitos para ser referência dos dados recebidos deve ser aplicada uma regra semelhante à seguinte:

Definir a referência segundo o formato

```
SET REFERENCE (FONTE, FONTE_REFERENCIA, SENSOR);
```

por exemplo

```
SET REFERENCE (FONTE['VINHA1'], FONTE['IPMA_COIMBRA'], 'TEMPERATURA');
```

Que será usado na regra de verificação

```
WHEN temperatura WITHIN(GET_REFERENCE_VALUE(temperatura),2) THEN valid;
```

Onde a função *GET\_REFERENCE\_VALUE* obtém o valor da referência para o mesmo instante temporal. A função *WITHIN* verifica se o valor está dentro do intervalo definido pelo valor de referência com o offset definido.

## 4.6. Validação dos Dados de Sensores Correlacionados

As validações anteriores focam-se na validação dos valores obtidos para uma determinada unidade de medida, por exemplo temperatura. No entanto, utilizando como exemplo a temperatura, os valores obtidos estão diretamente correlacionados com outros sensores, tais como sensores de humidade e pressão atmosférica. Através desta correlação é possível determinar se um determinado valor da temperatura é coerente com os valores obtidos pelos outros sensores no mesmo instante temporal. Desta maneira analisamos a coerência do sistema relacionando todos os sensores correlacionados. Para criar as funções de correlação será feita uma análise histórica do sistema e ter-se-á em consideração estudos

que indicam as correlações existentes entre medidas. Para verificar a coerência destas relações serão definidas regras tais como:

```
WHEN temperatura < temperatura[1] AND pressao > pressao[1] THEN invalid;  
WHEN temperatura > temperatura[1] AND pressao < pressao[1] THEN valid;
```

No Anexo A é especificada a gramática das regras definidas e de que maneira se pode combinar os operadores e condições para formar as várias regras.

## 5. Avaliação Experimental

Este capítulo tem como objetivo descrever o funcionamento do sistema, vamos usar como exemplo uma Fonte em Análise situada em Vila Nova de Foz Côa composta por oito nós com um número variado de sensores. Estes nós estão espalhados num terreno com 43 hectares, onde dominam os socacos. Por esse motivo, os sensores não estão uniformemente distribuídos pela área coberta, devido características particulares do terreno. O facto de ser “em escada”, com diferentes altitudes, e a proximidade ao Rio Douro que origina variações nos valores recolhidos pelos vários sensores. Assim e de modo a obter-se um conhecimento da variação dos valores das métricas, a área foi dividida em duas zonas distintas. A figura 19 ilustra a distribuição dos sensores nas duas zonas.



Figura 19 - Fonte de dados proveniente da Fonte Foz Côa

Em cada zona, os sensores foram colocados em sítios distintos por forma a ter uma cobertura mais significativa, e mais precisa dos valores recolhidos por cada métrica. Cada um dos nós representados na imagem anterior possui um conjunto de sensores. No entanto, nem todos sensores são relevantes para este estudo, por isso vamos ignorar o sensor que mede o nível de bateria e o sensor que mede a potência do sinal entre nós, analisando os restantes dados

## 5.1. Nós e Sensores da WSN

Os nós possuem um conjunto de sensores distintos. A tabela abaixo resume a distribuição dos sensores pelos vários nós,

Sensor	Nó 1	Nó 2	Nó 3	Nó 4	Nó 5	Nó 6	Nó 7	Nó 8
Controlador	✓							
Temperatura do ar (°C)	✓		✓	✓	✓	✓		
Humidade do ar (%)	✓		✓	✓	✓	✓		
Humidade do solo (%)	✓	✓	✓		✓	✓		
Humidade da Folha	✓			✓				
UV		✓					✓	
Radiação Solar		✓					✓	
Pressão Atmosférica							✓	✓
Direção do Vento							✓	✓
Velocidade do Vento (km/h)							✓	✓
Precipitação							✓	✓

**Tabela 1 - Distribuição de sensores na rede**

Como se pode observar na tabela anterior, existe um nó (Nó 1) que além de recolha de dados meteorológicos é o controlador da rede do projeto InovWine2. O controlador recebe e agrega os valores recolhidos pelos sensores dos outros nós, encapsula os dados e envia-os para o sistema *WSN Server*. Estes dados são posteriormente validados e processados de acordo com as indicações dos fabricantes dos sensores antes de os enviar para o protótipo.



## 5.2. Especificação da Fonte

No protótipo, foi registado uma fonte com a designação de “*IW2Coa*”, que referencia e integra no sistema os dados recebidos do *WSN Server*. Na sua especificação, além das características da fonte, nomeadamente a localização, os nós que compõem a rede e quais os sensores que recolhem informação, é também especificado para cada sensor quais os limites mínimos e máximos admissíveis. Na tabela seguinte estão indicados os limites definidos para alguns dos sensores utilizados no protótipo .

Sensor	Unidade de Medida	Mínimo	Máximo
Temperatura	°C	-10	45
Humidade do Ar	%	0	100
Humidade do Solo	%	0	100

**Tabela 2 - Tabela com limites especificados para cada sensor**

Abaixo encontra-se um exemplo de parte de uma mensagem disponibilizada à Fonte *IW2Coa*, apenas com a informação dos sensores definidos acima. Esta mensagem inclui os dados de 3 instantes de aquisição de dados (das 9h27m às 9h57m), e cada um inclui os dados dos 3 sensores indicados acima (temperatura, humidade do ar e humidade do solo). Para questões de demonstração, iremos denominar estes instantes como D1, D2 e D3, que serão utilizados posteriormente para demonstrar o fluxo de processamento.

```
"data_acq": [
  {"description": "Data de Aquisição",
   "date": "2017-09-14 09:27:03",
   "data": [{"metric": "temp-ar", "value": "28.144", "unit": "°C"},
            {"metric": "hum-ar", "value": "42.912", "unit": "%"},
            {"metric": "hum-solo", "value": "4.85347", "unit": "%"}
          ],
   "required": ["metric", "value", "unit"]
  },
  {"description": "Data de Aquisição",
   "date": "2017-09-14 09:42:03",
   "data": [{"metric": "temp-ar", "value": "58.0938", "unit": "°C"},
            {"metric": "hum-ar", "value": "41.2031", "unit": "%"},
            {"metric": "hum-solo", "value": "3.96703", "unit": "%"}
          ],
   "required": ["metric", "value", "unit"]
  },
  {"description": "Data de Aquisição",
   "date": "2017-09-14 09:57:03",
   "data": [{"metric": "temp-ar", "value": "29.9938", "unit": "°C"},
            {"metric": "hum-ar", "value": "38.6243", "unit": "%"},
            {"metric": "hum-solo", "value": "3.96703", "unit": "%"}
          ],
   "required": ["metric", "value", "unit"]
  }
]
```

Este excerto é parte da mensagem que se encontra no anexo B.

De modo a validar e correlacionar os dados da fonte *IW2Coa* foi definida uma fonte de referência, designada de Fonte Foz Coa *Accuweather*, que recolhe os dados do sistema *Accuweather*, através da API e com uma frequência de recolha de 20 em 20 minutos. Como esta fonte não disponibiliza os dados diretamente utilizáveis pelo sistema, foi necessário implementar um plugin, em Java, que além de recolher os dados com a frequência especificada, transforma os dados num formato compatível.

### 5.3. Regras de Validação Definidas

Cada um dos valores recolhidos por cada um dos sensores, é analisado individualmente. Para o efeito foi definido um conjunto de regras para cada uma das métricas dos sensores. Na tabela 3 são indicadas algumas das regras definidas, algumas das quais são aplicáveis para algumas das métricas (ex. R4 não é aplicada à métrica Humidade do Solo, pois não existe uma fonte de referência).

Regra	Temperatura	Humidade do Ar	Humidade do Solo
R1 – Validação por domínio	✓	✓	✓
R2 – Validação por variação entre leituras consecutivas	✓	✓	✓
R3 – Amplitude de valores das últimas 10 leituras	✓	✓	✓
R4 – Validação com a fonte de referência	✓	✓	✗
R5 – Validação por comparação com os nós vizinhos	✓	✓	✓

**Tabela 3 - Lista de regras para os sensores em análise**

A humidade do solo é uma métrica para a qual não existe uma fonte de referência e, mesmo se esta fosse recolhida as distâncias entre os pontos de recolha teria de ser mínima. A humidade do solo é muito dependente do tipo de solo e da proximidade a fontes de água. Desta forma temos de recorrer ao histórico recente do próprio sensor para o validar bem como, aos seus vizinhos. Das regras apresentadas na tabela 3, para questões de demonstração do protótipo, apenas são utilizadas as regras R1, R2 e R5 (assinaladas a verde) unicamente para a métrica de temperatura. A aplicação das regras para as outras métricas será processada de um modo similar.

## 5.4. Protótipo Implementado para o Caso de Estudo

Para o correto funcionamento do protótipo, e considerando que este necessita de adequar a estrutura de dados a processar, é necessário a implementação dos plugins específicos para cada uma das fontes de dados: a fonte de dados a analisar (dos sensores) e uma fonte de dados oficial (*Accuweather*).

Cada um desses plugins é executado numa *Thread* distinta, com comportamentos distintos. Enquanto que a fonte de dados (neste caso denominada *IW2Coa*) está constantemente à escuta de novas mensagens, a fonte do *Accuweather*, “acorda” periodicamente, por defeito de 20 em 20 minutos, e envia um pedido para obter as informações atualizadas. Em ambas as fontes, a informação recolhida é depois processada por um objeto que faz o *parsing* dos dados e os disponibiliza para o módulo de análise. Os dados da fonte *IW2Coa* são posteriormente analisados e avaliados segundo as regras previamente definidas.

O diagrama seguinte representa o modo de integração dos dados das fontes.

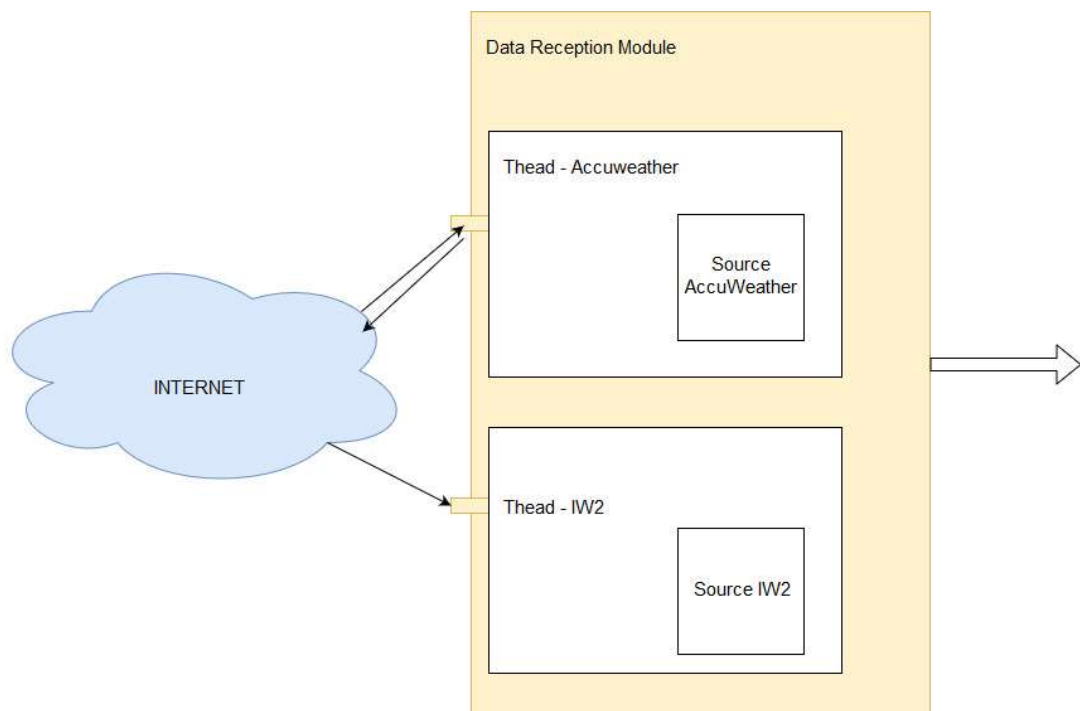


Figura 20 - Fluxo de integração dos dados

Genericamente, existem estes dois tipos distintos de *plugins*, sendo o seu comportamento dependente do tipo de Fonte de dados (implementado com sendo um objeto que implementa o interface *Source*). Na figura anterior, estão ilustrados os objetos *SourceAccuweather* e *SourceIW2* que, após o *parsing*, validam se os dados recebidos estão de acordo com o formato definido, e disponibilizando-os para o módulo de análise de dados.

Além dos plugins, também é necessário traduzir as regras especificadas pelo utilizador em ações de processamento dos dados.

#### 5.4.1. Módulo de Análise dos Dados

O módulo de análise de dados processa e valida os dados provenientes das fontes de acordo com as regras definidas. Para o efeito, primeiro é necessário traduzir as regras definidas pelo utilizador, definidas de acordo com a gramática proposta no capítulo 4, em ações de processamento.

Para o processamento das regras optou-se por utilizar a biblioteca *J-Easy* [13], um motor de processamento de regras. Este motor de processamento é implementado na linguagem *Java* e permite criar um mecanismo de aplicação de regras, segundo uma especificação definida no *J-Easy*. Estas regras podem ser definidas da seguinte forma:

- Através da implementação de classes que implementam os interfaces “*Condition*” e “*Action*”;
- Através da implementação de uma classe do tipo *Rule* que implementa um conjunto de métodos, em que especificadas as notações (*Java Notation*): *@Rule*, *@Condition* e *@Action*;
- Através de uma do tipo “*MVELRule*” mais adequada a regras baseadas em linguagem humana que pode ser definida através da criação do objeto ou da leitura de um ficheiro de configuração do tipo “*yml*”.

Qualquer uma destas formas de implementação avalia objetos da classe *Facts*. Um objeto *Facts* representa um acontecimento que se pretende analisar, no caso concreto, um valor lido por um sensor correspondente a uma medida é um *Facts* a analisar.

No exemplo anterior da mensagem *JSON*, a linha:

```
[{"metric": "temp-ar", "value": "58.0938", "unit": "°C"}]
```

representa um *Facts* a avaliar pelo sistema de regras.

Dos três tipos de implementação mencionados foi escolhida a estratégia de implementação de classes do tipo “*Rule*”. Em cada classe é definida a condição que se pretende avaliar e ação a tomar caso a condição se verifique, e também a prioridade de execução das regras entre si. Por exemplo, para a regra R5 definida pela gramática como,



Figura 21 - Definição da Regra R5 segundo a gramática

Para traduzir a regra R5 numa *Rule* do *J-Easy*, foi implementada a classe “*RuleImplValidationZone*”, apresentada abaixo, onde definimos a condição, a ação e a prioridade de execução da regra.

```

@Rule
public class RuleImplValidationZone {

    DataMetric dataMetric;
    Zone zona;
    DBConnector db;

    RuleImplValidationZone(Zone z, DBConnector dbconn) {
        zona = z;
        db = dbconn;
    }
    @Condition
    public boolean isWithinAvgZone(Facts f) {
        dataMetric = f.get("value");
        return (dataMetric.getValueOriginal() >= zona.getMin()
            || dataMetric.getValueOriginal() <= zona.getMax());
    }
    @Action
    public void saveData() {
        dataMetric.setAcceptableMetricData(true);
        db.updateDataMetricStatus(dataMetric);
    }
    @Priority
    public int getPriority() {
        return 1;
    }
}

```

Figura 22 - Implementação da Regra R5

No método *isWithinAvgZone()* verifica-se se o valor recolhido pelo sensor está dentro do intervalo mínimo e máximo da sua zona de validação. O valor recolhido é carregado previamente para um objeto *DataMetric* com os atributos dessa leitura e, adicionado à lista de *Facts* a analisar. A zona de validação é previamente definida e permite comparar os valores em análise com valores já recolhidos pelos seus vizinhos.

No método *saveData()* é atualizada a informação relativa à validação do valor no objeto *DataMetric*, sendo este depois atualizado na base de dados.

O método *getPriority()* define a prioridade desta regra em relação às restantes.

Como as regras do *J-Easy* são aplicados a objetos *Facts*, é necessário que os dados recolhidos sejam traduzidos, “injetados” e interpretados pelo motor de regras do *J-Easy* como *Facts*. O sistema carrega os dados por avaliar para a lista de *Facts* através dos objetos *DataMetric*,

```
Facts facts = new Facts ();
Facts.put ("value", dataMetric);
```

E de seguida estes são “injetados” nas regras como se exemplifica de seguida,

```
Rules rulesFive = new Rules();
rulesFive.register(new RuleImplValidationZone(zona, db));
RulesEngine rulesEngine = new DefaultRulesEngine();
rulesEngine.fire(rulesFive, facts);
```

onde é ativado o *RuleEngine* e são efetuadas as validações.

Os *Facts* F1, F2 e F3 referem-se respetivamente aos dados de exemplo D1, D2 e D3. Estes *Facts* são avaliados de forma sequencial sendo que, o fluxo que cada *Facts* pode percorrer pode ser terminado em qualquer uma das regras, como ilustrado na figura seguinte.

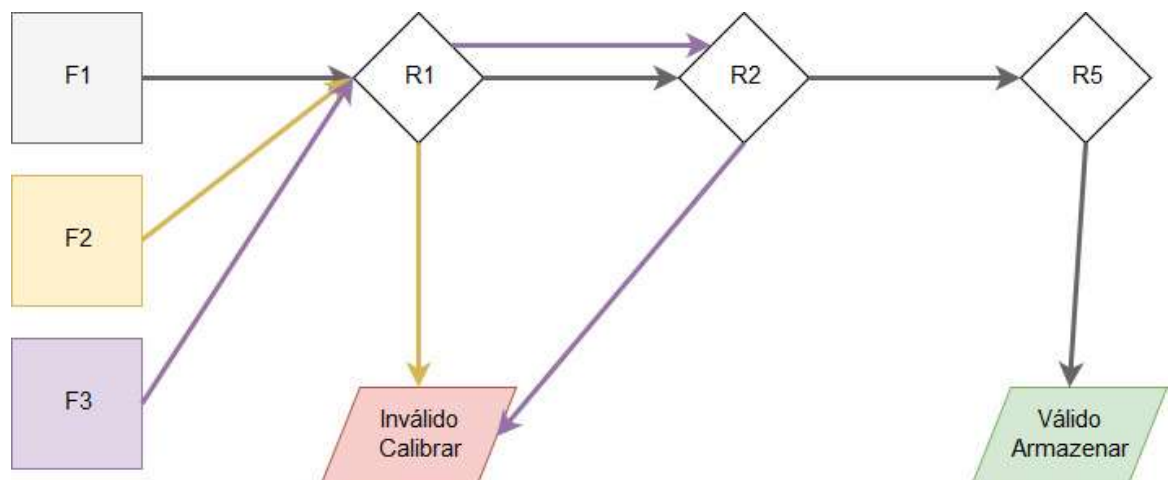


Figura 23 - Fluxo de validação dos dados.

Na figura, F1, F2 e F3 representam os dados de exemplo (secção 5.2), que são avaliados pelas regras R1, R2 e R5 que foram definidas para estes dados (definidas na secção 5.3), e as “setas” ilustram o fluxo percorrido por cada um dos dados. O fluxo é determinado pelo resultado da avaliação e da ação definida para cada uma das regras. Por exemplo, da aplicação regra R1, pode resultar o encaminhamento para a regra R2 (ex. para os dados F1 e F3) ou resultar no armazenamento do dado com a catalogação de que este é inválido e é necessário ser calibrado (ex. o dado F2). Quando a catalogação é inválida deve iniciar-se o processo de calibração, dessa forma é feita uma tentativa de recuperação dos dados não válidos. Os dados calibrados são reavaliados pelo sistema de modo a validar a calibração.

## 5.5. Exemplo de Avaliação dos Dados de um Período

O período em análise é composto por 48 tempos de aquisição, correspondentes ao intervalo das 9h00 às 20h00, obtendo assim 1392 aquisições de dados por parte dos 29 sensores que compõem a fonte. Foi escolhido este intervalo de tempo por haver dados da fonte de referência neste intervalo permitindo assim realizar as comparações com a fonte de referência nos casos em que se aplique.

Particularmente vamos analisar as métricas presentes no exemplo para as quais foram definidas as regras, apresentando na tabela a taxa de sucesso de cada uma das regras,

<b>Regra</b>	<b>Temperatura</b>	<b>Humidade do Ar</b>	<b>Humidade do Solo</b>
R1	97.91%	100%	100%
R2	97.3%	90.5%	100%
R3	55.55%	70.73%	100%
R4	32.14%	35.29%	---
R5	84%	72.72%	34.09%

Tabela 4 - Resultados da percentagem de acertos de cada regra

Analisando a tabela anterior podemos concluir que embora em termos de sequência de valores estes apresentem uma percentagem elevada de acertos, no que toca à comparação com a fonte de referência esse valor é demasiado baixo. De maneira a minimizar essas falhas aplicou-se a calibração pelo método dos mínimos quadrados nos sensores de temperatura do ar e humidade do ar para os quais temos dados da fonte de referência.

## 5.6. Calibração do Valor Obtido pelo Sensor

Como definido na arquitetura do sistema, quando um valor é considerado inválido, este deve ser calibrado. Neste protótipo optou-se pela calibração pelo método dos mínimos quadrados. Desta forma foi selecionada uma amostra de 28 elementos da fonte de referência e da fonte em análise donde surgiu a função de aproximação para a temperatura do ar,

$$y = 0.44551x + 12,958$$

E para a humidade do ar a seguinte função

$$y = 1.0676x + 8,2649$$

Aplicando esta função ao valor inválido temos um novo valor que terá de passar pelas fases de validação definidas anteriormente, à exceção da comparação com os vizinhos visto que estes não foram calibrados, tendo então os seguintes resultados para as métricas calibradas

<b>Regra</b>	<b>Temperatura</b>	<b>Humidade do Ar</b>
R1	100%	100%
R2	100%	100%
R3	57.14%	60.71%
R4	67.85%	64.28%

**Tabela 5 - Validação dos valores calibrados**

Comparando a tabela 3 e 4 temos uma maior percentagem de acertos em quase todas as regras, podendo concluir que o processo de calibração obteve sucesso, minimizando as perdas de informação que resultaram da primeira validação.



## 6. Conclusões

Neste capítulo apresentam-se as contribuições desta dissertação para o panorama tecnológico na área de *Internet of Things*, as principais dificuldades encontradas ao longo da dissertação e, por fim, algumas sugestões de trabalho futuro para enriquecer este trabalho.

### 6.1. Contribuição da Tese

O sistema proposto permite oferecer a sistemas de *IoT* maior credibilidade e consistência dos dados, através da análise e validação constantemente dos dados, recorrendo a diversos métodos e ao cruzamento de diversas fontes de informação. Também é proposto um mecanismo que possibilita a calibração e estimação de valores a partir dos dados que são conhecidos, obtendo assim um fluxo de dados mais completo.

É um sistema heterogéneo que permite o registo de qualquer fonte de dados, sem a necessidade de alterações no seu funcionamento base, requerendo apenas a implementação de um *plugin*, que possibilita a comunicação com o sistema de análise. O sistema é aberto, permitindo a adição de novas regras de validação e ações a tomar, adequadas aos dados a analisar.

### 6.2. Principais Dificuldades

A maior dificuldade foi a definição da estrutura de um sistema heterogéneo, que permita a inclusão de vários tipos de sensores, tão distintos como um sensor de radiação solar e um sensor de deteção de intrusão. Mais genericamente a definição das fontes de dados, como registar fontes com vários nós e sensores e fontes com apenas um sensor, selecionar apenas as características fundamentais para o estudo. As fontes de referência são um elemento fundamental para o sistema e por isso a sua angariação e o desenvolvimento de métodos para obter os dados num formato que pudesse satisfazer as necessidades.

Outra dificuldade prendeu-se com a definição das regras de validação, com tanta diversidade de dados era necessário encontrar uma forma de definir genericamente as regras de modo a que estas pudessem ser transversais nos casos comuns, como a validação por domínio, mas também fosse fácil definir algumas mais específicas de cada tipo de sensor.

### 6.3. Trabalho Futuro

De futuro pretende-se desenvolver o protótipo para que este possa integrar novas funções de calibração e novas regras de validação, com especial foco para os sensores que, não tendo uma fonte de referência, possam através da correlação com outros sensores ser validados. Com o uso continuado do sistema mais tipos de dados surgirão o que fomentará o seu crescimento e versatilidade. É também fundamental encontrar mais fontes de referência de modo a enriquecer o sistema nem que para isso seja necessário ter uma modalidade com recursos pagos.

Pretende-se também implementar o *front-end* do sistema para tornar mais fácil a configuração de fontes de dados, bem como consultar os resultados obtidos pela validação e pelas ações realizadas.

# Referências

- [1] Guidelines on validation procedures for meteorological data from automatic weather stations in Journal of Hydrology Volume 402 (2011)
- [2] Internet of Things: Architectures, Protocols, and Applications in Journal of Electrical and Computer Engineering Volume 2017 (2017)
- [3] <http://www.opengeospatial.org/standards/sos> (acedido em Janeiro de 2017)
- [4] <http://www.opengeospatial.org/standards/sensorthings> (acedido em Janeiro de 2017)
- [5] <http://www.sensorup.com/iot-platform/> (acedido em Janeiro de 2017)
- [6] <http://istsos.org/en/trunk/doc/intro.html> (acedido em Janeiro de 2017)
- [7] <https://opensensors.com/> (acedido em Janeiro de 2017)
- [8] <http://www.openiot.eu/> (acedido em Janeiro de 2017)
- [9] <https://www.sigfox.com/en/solutions/sigfox-services> (acedido em Janeiro de 2016)
- [10] Santos, F. Correia (2002) Fundamentos de Análise Numérica, Lisboa: Edições Sílabo
- [11] Weather Forecasting Using Sliding Window Algorithm, Piyush Kapoor and Sarabjeet Singh Bedi, in ISRN Signal Processing Volume 2013 (2013)
- [12] Predicting the Temperature using Sliding Window Algorithm, Er. Garima Jain et al, International Journal of Research in computer and communication Technology, Vol 6, Issue- 4, April- 2017
- [13] <https://github.com/j-easy> (acedido em Janeiro de 2018)

---

# Bibliografia

<http://www.rmets.org/sites/default/files/pdf/guidelines/aws-guide.pdf> (acedido em Junho de 2016)

<http://www.acurite.com/5in1#calibrate> (acedido em Junho de 2016)

<http://www.rmets.org/weather-and-climate/observing/guidelines-observing-0> (acedido em Junho de 2016)

<http://www.vaisala.com/en/events/webinarcentral/Pages/Vaisala-Sensor-Calibration-for-Automatic-Weather-Stations.aspx> (acedido em Junho de 2016)

<https://learn.adafruit.com/downloads/pdf/calibrating-sensors.pdf> (acedido em Junho de 2016)

<http://www.vectornav.com/support/library/calibration> (acedido em Junho 2016)

<https://www.endevco.com/support/calibration-services/> (acedido em Junho 2016)

<https://www.xively.com/navigating-iot/iot-basics> (acedido em Outubro 2016)

[http://www.engineeringtoolbox.com/air-altitude-pressure-d\\_462.html](http://www.engineeringtoolbox.com/air-altitude-pressure-d_462.html) (acedido em Janeiro 2017)

[https://en.wikipedia.org/wiki/Atmospheric\\_pressure](https://en.wikipedia.org/wiki/Atmospheric_pressure) (acedido em Janeiro 2017)

[https://en.wikipedia.org/wiki/Event\\_condition\\_action](https://en.wikipedia.org/wiki/Event_condition_action) (acedido em Outubro 2017)

<https://flink.apache.org/news/2016/04/06/cep-monitoring.html> (acedido em Outubro 2017)

[https://www.tutorialspoint.com/drools/drools\\_sample\\_drools\\_program.htm](https://www.tutorialspoint.com/drools/drools_sample_drools_program.htm) (acedido em Outubro 2017)

<https://github.com/j-easy/easy-rules> (acedido em Outubro 2017)

<https://www.datascience.com/blog/k-means-clustering> (acedido em Novembro de 2017)

---

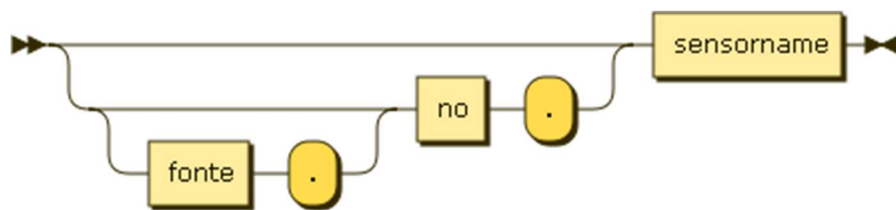
# Anexo A

Especificação da gramática das regras validação, expressa na notação EBNF (*extended Backus-Naur form*).

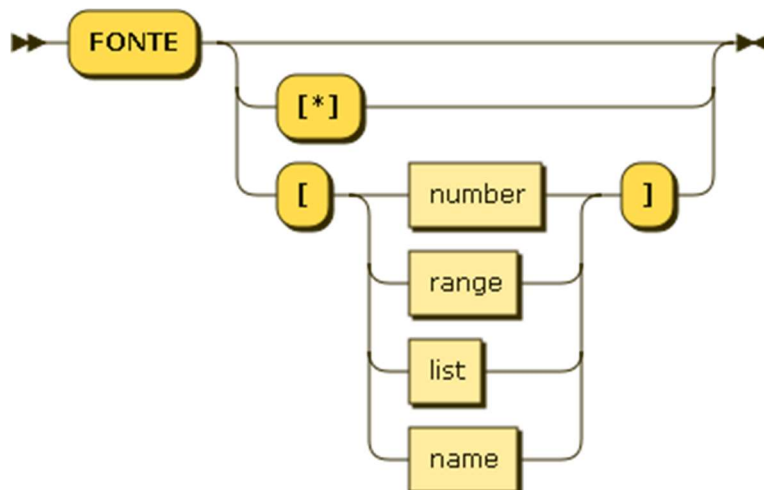
**Rule:**



**sensor:**



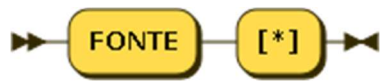
**fonte:**



**fonte\_da\_leitura:**



**fonte\_qualquer:**



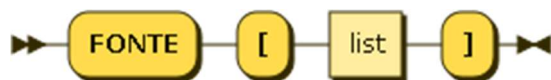
**fonte\_index\_i:**



**fonte\_range:**



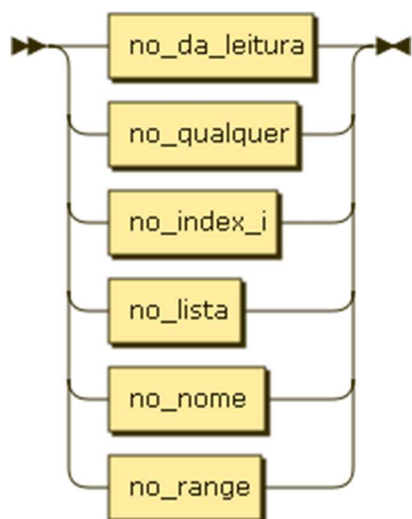
**fonte\_lista:**



**fonte\_nome:**



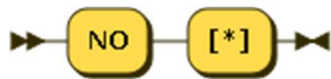
**no:**



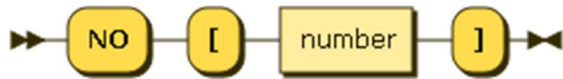
**no\_da\_leitura:**



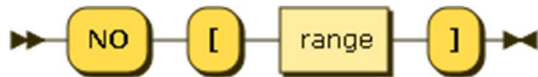
**no\_qualquer:**



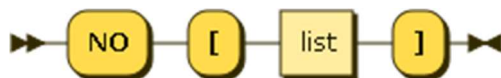
**no\_index\_i:**



**no\_range:**



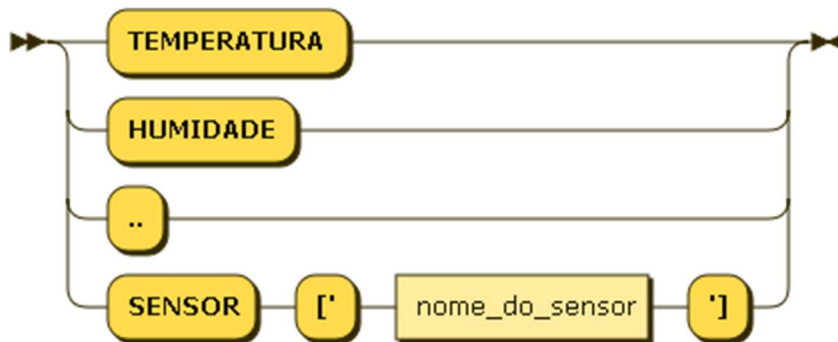
**no\_lista:**



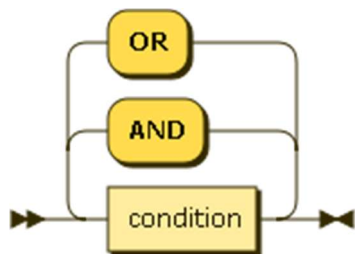
**no\_nome:**



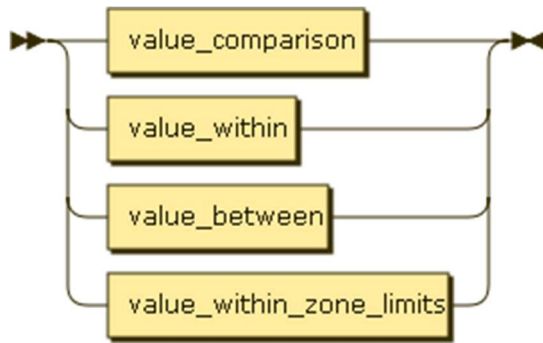
**sensorname:**



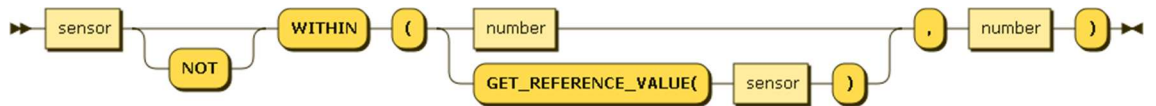
**conditions:**



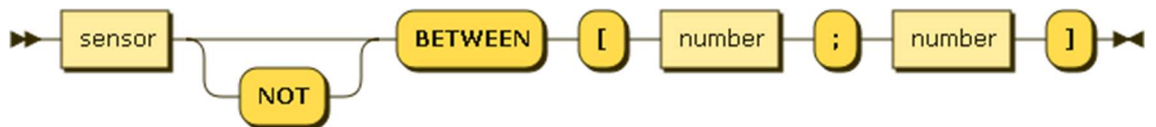
**condition:**



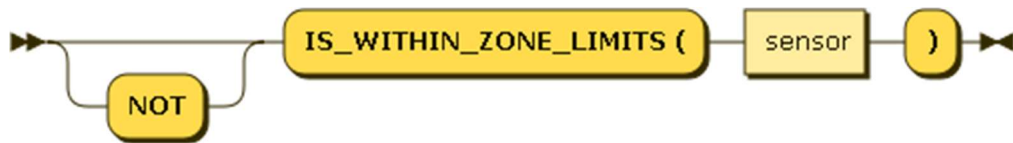
**value\_within:**



**value\_between:**



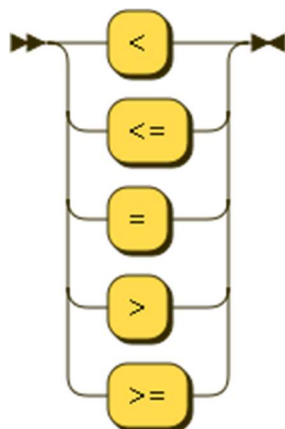
**value\_within\_zone\_limits:**



**value\_comparison:**

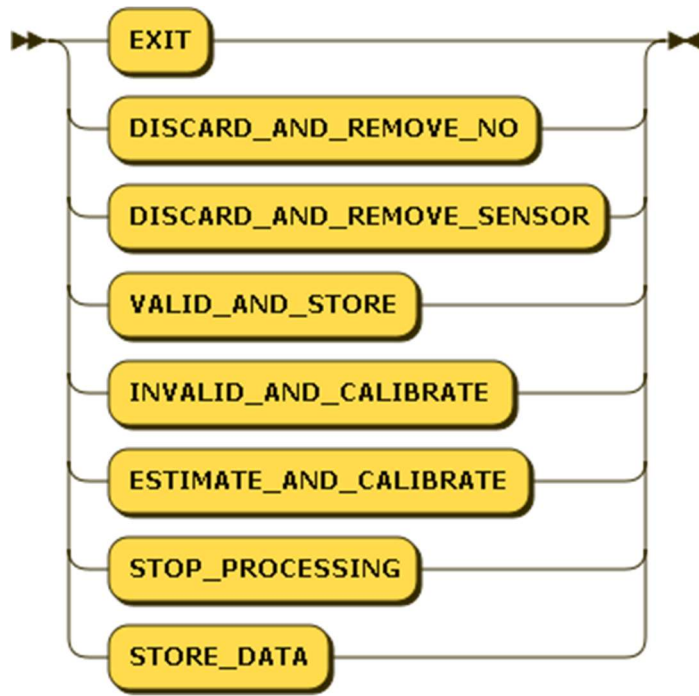


**comparator:**

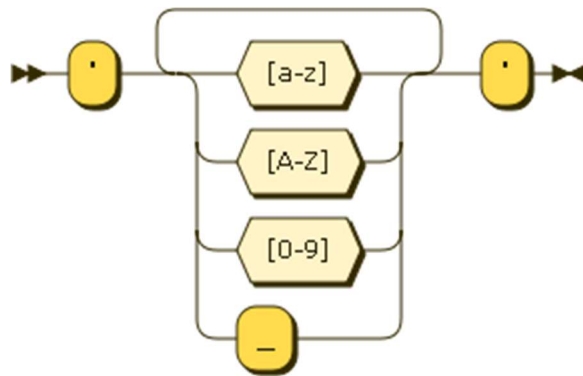


**action:**





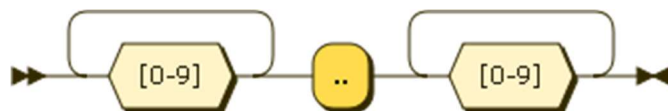
**name:**



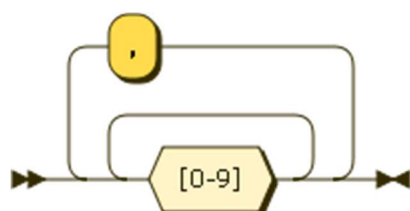
**number:**



**range:**



**list:**





## Anexo B

```
{
  "version": "http://wsnserver.las.ipn.pt/Date_Schema_v1",
  "title": "IW2Coa",
  "type": "wsn",
  "properties": {
    "wsn": {
      "uid": "6f910ed9000b0012",
      "nodes": [{
        "uid": "4714af4200190016",
        "data": {
          "measures": [{
            "data_acq": "2017-09-14 09:27:03",
            "metric": "temp-ar",
            "value": "28.144",
            "unit": "°C"
          }, {
            "data_acq": "2017-09-14 09:27:03",
            "metric": "hum-ar",
            "value": "42.912",
            "unit": "%"
          },
          {
            "data_acq": "2017-09-14 09:27:03",
            "metric": "hum-solo",
            "value": "4.3406539",
            "unit": "%"
          },
          {
            "data_acq": "2017-09-14 09:42:03",
            "metric": "temp-ar",
            "value": "58.0938",
            "unit": "°C"
          }, {
            "data_acq": "2017-09-14 09:42:03",
            "metric": "hum-ar",
            "value": "41.2031",
            "unit": "%"
          },
          {
            "data_acq": "2017-09-14 09:42:03",
            "metric": "hum-solo",
            "value": "3.96703",
            "unit": "%"
          },
          {
            "data_acq": "2017-09-14 09:57:03",
            "metric": "temp-ar",
            "value": "29.9938",
            "unit": "°C"
          },
          {
            "data_acq": "2017-09-14 09:57:03",
            "metric": "hum-ar",
            "value": "38.6243",
            "unit": "%"
          },
          {
            "data_acq": "2017-09-14 09:57:03",
            "metric": "hum-solo",
            "value": "3.96703",
            "unit": "%"
          }
        ]
      }
    ]
  }
}
```

```

        "metric": "hum-solo",
        "value": "3.96703",
        "unit": "%"
    }
}

]
}
}, {
  "uid": "47112683000e0021",
  "data": {
    "measures": [{
      "data_acq": "2017-09-14 09:27:03",
      "metric": "radiacao",
      "value": "64.702255",
      "unit": "W/m2"
    }, {
      "data_acq": "2017-09-14 09:27:03",
      "metric": "hum-solo",
      "value": "4.3406539",
      "unit": "%"
    },
    {
      "data_acq": "2017-09-14 09:42:03",
      "metric": "radiacao",
      "value": "80.278725",
      "unit": "W/m2"
    }, {
      "data_acq": "2017-09-14 09:42:03",
      "metric": "hum-solo",
      "value": "4.2239",
      "unit": "%"
    },
    {
      "data_acq": "2017-09-14 09:57:03",
      "metric": "radiacao",
      "value": "80.278725",
      "unit": "W/m2"
    }, {
      "data_acq": "2017-09-14 09:57:03",
      "metric": "hum-solo",
      "value": "4.0406539",
      "unit": "%"
    }
  ]
}
}, {
  "uid": "4714ae42000e0015",
  "data": {
    "measures": [{
      "data_acq": "2017-09-14 09:27:03",
      "metric": "radiacao",
      "value": "64.702255",
      "unit": "W/m2"
    }, {
      "data_acq": "2017-09-14 09:27:03",
      "metric": "hum-solo",
      "value": "4.3406539",
      "unit": "%"
    }, {
      "data_acq": "2017-09-14 09:27:03",
      "metric": "temp-ar",
      "value": "20.761017",
      "unit": "°C"
    }, {
      "data_acq": "2017-09-14 09:27:03",
      "metric": "hum-ar",

```

```
    "value": "43.514771",
    "unit": "%"
  }, {
    "data_acq": "2017-09-14 09:27:03",
    "metric": "hum-folha",
    "value": "0.0",
    "unit": "%"
  },
  {
    "data_acq": "2017-09-14 09:42:03",
    "metric": "radiacao",
    "value": "64.702255",
    "unit": "W/m2"
  }, {
    "data_acq": "2017-09-14 09:42:03",
    "metric": "hum-solo",
    "value": "1.69596",
    "unit": "%"
  }, {
    "data_acq": "2017-09-14 09:42:03",
    "metric": "temp-ar",
    "value": "21.329447",
    "unit": "°C"
  }, {
    "data_acq": "2017-09-14 09:42:03",
    "metric": "hum-ar",
    "value": "43.12561",
    "unit": "%"
  }, {
    "data_acq": "2017-09-14 09:42:03",
    "metric": "hum-folha",
    "value": "0.0",
    "unit": "%"
  }, {
    "data_acq": "2017-09-14 09:57:03",
    "metric": "radiacao",
    "value": "68.3585",
    "unit": "W/m2"
  }, {
    "data_acq": "2017-09-14 09:57:03",
    "metric": "hum-solo",
    "value": "1.5860779",
    "unit": "%"
  }, {
    "data_acq": "2017-09-14 09:57:03",
    "metric": "temp-ar",
    "value": "21.801352",
    "unit": "°C"
  }, {
    "data_acq": "2017-09-14 09:57:03",
    "metric": "hum-ar",
    "value": "42.8281",
    "unit": "%"
  }, {
    "data_acq": "2017-09-14 09:57:03",
    "metric": "hum-folha",
    "value": "0.0",
    "unit": "%"
  }
}
]
}
}
}
}
```