

Relatório Final de Projeto
“P7 - Estação meteorológica”

Departamento de Engenharia de Eletrónica e Telecomunicações e de Computadores.

Licenciatura em Engenharia Informática, Redes e Telecomunicações (LEIRT)

Júri

Presidente.: Prof. Dr. Pedro Vieira.

. Orientador: Prof. Dr. Paulo Lourenço.

Arguente: Prof. Dr. Luís Miguel Fernandes,

2024/2025

Duarte Silva, 48711
Rafael Miranda, 49831

I. Declaração de Integridade

Declaro que este trabalho de projeto é o resultado da nossa investigação pessoal e independente. O seu conteúdo é original e todas as fontes listadas nas referências bibliográficas foram consultadas e estão devidamente mencionadas no texto. Mais declaro que todas as referências científicas e técnicas relevantes para o desenvolvimento do trabalho estão devidamente citadas e constam das referências bibliográficas.

Os autores

Duarte Silva e Rafael Miranda

Lisboa, 02 de maio de 2025

II. Índice

I.	Declaração de Integridade	3
II.	Índice	5
III.	Lista de Símbolos e Acrónimos.....	7
IV.	Lista de Figuras	11
IV.	Resumo	13
1.	Introdução.....	15
2.	Tecnologia <i>LoRa</i> e <i>LoRaWAN</i>	17
	Modulação <i>LoRa</i>	17
	Interferência.....	18
	<i>LoRaWAN</i>	19
	<i>OTAA</i> vs <i>ABP</i>	23
3.	Arduino IDE.....	25
4.	Detalhes da Arquitetura do Sistema.....	27
	Placa Seeeduino <i>LoRaWAN</i>	27
	Sensores para Monitorização Meteorológica	29
	Protocolos de Ligação dos Sensores.....	36
	Análise da Leitura de dados e Interação Placa/PC	39
5.	Plataforma The Things Network	43
6.	DataCake	45
7.	Caixa de Suporte e proteção do sistema	51
8.	Conclusão	53
9.	Bibliografia.....	55
	Bibliografia de Imagens	59

III. Lista de Símbolos e Acrónimos

A

- ABP - Activation By Personalization (Ativação Por Personalização)
- ACK - Acknowledge (Confirmação)
- ADC- Analog-to-Digital Converter (Conversor Analógico-Digital)
- API- Application Programming Interface (Interface de Programação de Aplicações)
- AppEUI - Application Extended Unique Identifier(Identificador Único Extendido da Aplicação)
- AppKey - Application Key (Chave da Aplicação)
- AppSKey - Application Session Key (Chave de Sessão da Aplicação)
- ARM - Advanced RISC Machines (Máquinas RISC avançadas)
- ASIC- Application-Specific Integrated Circuit (Circuito integrado de aplicação específica)

B

- Back-end

C

- CSS - Chirp Spread Spectrum (Espectro de Espalhamento Chirp)
- CPU – Central Processing Unit (Unidade Central de Processamento)

D

- DataCake - Plataforma de visualização de dados IoT
- DevAddr - Device Address (Endereço do Dispositivo)
- DevEUI - Device Extended Unique Identifier (Identificador Único Estendido do Dispositivo)
- DFU - Device Firmware Update (Atualização de Firmware do Dispositivo)
- DHT20 - Sensor de Temperatura e Humidade
- DPS310 - Modelo de sensor de pressão barométrica
- DR - Data Rate (Taxa de Dados)
- Duty-Cycle -

E

- ETSI- European Telecommunications Standards Institute (Instituto Europeu de Normas de Telecomunicações)

F

- FIFO – First in, First Out (Primeiro a Entrar, Primeiro a Sair)

G

- GPS - Global Positioning System (Sistema de Posicionamento Global)

H

- Hz-Hertz
- HTTP- Hyper Text Transfer Protocol (Protocolo de Transferência de Hipertexto)
- hPa- HectoPascal

I

- I2C - Inter-Integrated Circuit (Circuito Interintegrado)
- ICSP - In-Circuit Serial Programming (Programação Série no Circuito)
- IDE - Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
- IoT - Internet of Things (Internet das Coisas)
- ISM - Industrial, Scientific and Medical (Industrial, Científico e Médico)

J

- JoinEUI - Join Extended Unique Identifier (Identificador Único Estendido de Associação)

L

- LED - Light Emitting Diode (Díodo Emissor de Luz)
- LiPo - Lithium Polymer Battery (Bateria de Polímero de Litíio)
- LoRa - Long Range (Tecnologia de comunicação de longo alcance)
- LoRaWAN - Long Range Wide Area Network (Rede de Área Ampla de Longo Alcance)
- LPWAN - Low Power Wide Area Network (Rede de Área Ampla de Baixo Consumo)
- LPO Time- Low Pulse Occupancy Time (Tempo de Ocupação de Baixo Pulso)

M

- M445C - Modelo de sensor de chuva
- MAC- Medium Access Control (Controle de Acesso ao Meio)
- MCU- Microcontroller Unit (Unidade Microcontroladora)
- MEMS- Micro-Electro-Mechanical Systems (Sistemas Microeletromecânicos)
- MHz- Megahertz

- MQTT- Message Queuing Telemetry Transport (Transporte de Telemetria com Fila de Mensagens)

N

- NACK - Negative Acknowledge (Confirmação Negativa)
- NwkSKey - Network Session Key (Chave de Sessão de Rede)

O

- OSI - Open Systems Interconnection (Interconexão de Sistemas Abertos)
- OTA - Over-The-Air (Através do Ar)
- OTAA - Over-The-Air Activation (Ativação através do ar)

P

- PWM - Pulse Width Modulation (Modulação por Largura de Pulso)
- PC – Personal Computer (Computador Pessoal)
- PM- Particulate Matter (Matéria Particulada)

R

- R/W - Read/Write (Leitura/Escrita)
- RHF76-052AM - Módulo de comunicação LoRa utilizado na placa Seeeduino
- RISC- Reduced instruction set computer (Computador com Conjunto Reduzido de Instruções)

S

- SCL - Serial Clock Line (Linha de Clock Série)
- SDA - Serial Data Line (Linha de Dados Série)
- SF - Spreading Factor (Fator de Espalhamento)
- SHINYEI PPD42 - Modelo de sensor de qualidade do ar
- SPI - Serial Peripheral Interface (Interface Série Periférica)

T

- TSL2561 - Modelo de sensor de luz digital
- TTN - The Things Network
- TTS - The Things Stack

U

- USB - Universal Serial Bus (Porta Série Universal)
- UV - Ultravioleta

IV. Lista de Figuras

Figura 1 - Up-chirp e Down-chirp ao longo do tempo	18
Figura 2 - Mensagem Lora no domínio da frequência	18
Figura 3 - Mapa de antenas TTN disponíveis na área do ISEL	19
Figura 4 - Distribuição de Comunidades em Portugal e Gateways de Lisboa	20
Figura 5 - Ilustração do processo "Join"	23
Figura 6 - Principais Componentes Arduino IDE	25
Figura 7 - Menu Sketch e Tools Arduino IDE	26
Figura 8 - Componentes da Estação meteorológica	27
Figura 9 - Placa Seeeduino LoRaWAN	28
Figura 10 - Sensor Grove DHT20	30
Figura 11 - Modulo DHT20	30
Figura 12 - Sensor de pressão atmosférica Grove DPS310	30
Figura 13 - SHINYEI PPD42NS	31
Figura 14 - Ilustração do mecanismo de leitura de partículas	31
Figura 15 - Sensor Grove Uv sensor e cabo grove	33
Figura 16 - Sensor Grove Digital Light v1.1	34
Figura 17 - Ilustração da variação de resistência	35
Figura 18 - Paine Solar 1W 5V 200mA - 110x80mm	36
Figura 19 - Variação do sinal SDA e SCL	38
Figura 20 - Código Arduino	39
Figura 21 - Resultados Índice qualidade do Ar em experiencia1	40
Figura 22 - Leitura 1	42
Figura 23 - Localização Leitura 1	42
Figura 24 - Dashboards de leitura 1	42
Figura 25 - Weather Prediction leitura 1	42
Figura 26 - Payload Decoder TTN	43
Figura 27 - Troca de mensagens bem sucedida	44
Figura 28 - Integração TTN e DataCake	45
Figura 29 - Continuação do processo de integração	46
Figura 30 - Payload Decoder DataCake	47
Figura 31 - Separadores Webpage	48
Figura 32 - HomePage	49
Figura 33 - Dashboards	49
Figura 34 - History	49
Figura 35 - Location	50
Figura 36 - Weather Predication	50
Figura 37 - Render	52
Figura 38 - Render 2	53

IV. Resumo

Pretende-se, com este projeto, desenvolver uma estação meteorológica baseada em Arduino para a monitorização da temperatura, humidade, pressão atmosférica, luz solar, níveis UV, qualidade do ar, deteção de chuva e coordenadas geográficas. Este sistema de sensores deverá operar na gama de frequências atribuída às redes *Long Range (LoRa)* para envio e registo dos parâmetros monitorizados, sendo valorizada a integração do dispositivo na infraestrutura *Long Range Wide Area Network (LoRaWAN)*. Para complementar o desenvolvimento do projeto, poderemos usar um simulador *Internet of Things (IoT)* que possui uma variedade de funcionalidades úteis, entre as quais a implementação virtual da rede e a sua simulação, a interação entre os nós virtuais e físicos da rede implementada no simulador, e o envio para o Arduino do código desenvolvido para a troca de dados entre dispositivos.

Neste relatório, iremos abordar mais densamente as características técnicas do hardware, bem como o desenvolvimento técnico do sistema.

The aim of this project is to develop a weather station based on Arduino for monitoring temperature, humidity, sun light, UV intensity, air quality, rain detection and geographical coordinates. This sensor system will operate within the frequency range allocated to LoRa networks for the transmission and logging of monitored parameters, with emphasis on integrating the device into the LoRaWAN infrastructure.

To complement the project's development, we can use an IoT simulator that offers a variety of useful functionalities, including virtual network implementation and simulation, interaction between virtual and physical nodes within the network, and the ability to send the developed code to the Arduino for data exchange between devices.

In this report, we will focus more extensively on the technical characteristics of the hardware as well as the technical development of the system.

1. Introdução

Este relatório académico constitui a terceira e última fase de entrega do projeto final do curso de Licenciatura em Engenharia Informática, Redes e Telecomunicações (LEIRT). O seu propósito principal é demonstrar a aplicação e consolidação dos conhecimentos adquiridos ao longo da formação, centrando-se no desenvolvimento integral de um sistema de monitorização ambiental autónomo, robusto, de fácil integração, portabilidade e sustentável.

O projeto abrangeu a conceção e construção de uma caixa de proteção e suporte para os diversos componentes do sistema, bem como a implementação de uma solução de comunicação sem fios baseada em *LoRaWAN*. Esta solução permite a recolha de dados por via da placa microcontroladora *Seeeduino LoRaWAN* com *Global Positioning System (GPS)* e sensores conectados, integrando-os posteriormente na plataforma *The Things Network (TTN)*. Os dados recolhidos são então processados e apresentados visualmente numa página Web através da plataforma *DataCake*, concretizando assim os princípios da *Internet of Things*.

O documento detalha as especificações técnicas do hardware utilizado, nomeadamente a placa *Seeeduino* e os sensores selecionados. Apresenta e explica o processo completo da cadeia de comunicação de dados, desde a sua origem nos sensores e transmissão via *LoRaWAN* até à sua disponibilização na *TTN*, abordando os fundamentos tecnológicos envolvidos. Inclui igualmente um capítulo dedicado à análise e verificação dos resultados obtidos durante a operação do sistema e outro que descreve o trabalho desenvolvido na conceção e construção da caixa de proteção.

Este trabalho culmina no desenvolvimento de um protótipo funcional que demonstra a integração de hardware, comunicação sem fios e plataformas *IoT* para monitorização de dados, cumprindo o planeamento temporal estabelecido para esta fase do projeto.

2. Tecnologia LoRa e LoRaWAN

A tecnologia *LoRa* é uma tecnologia de comunicação wireless que se insere nas tecnologias *Low Power, Wide Area Networking (LPWAN)*. A *LoRa* tem diversas características interessantes, nomeadamente baixo consumo, longo alcance e resistência a interferências dado que a tecnologia *LoRa* possui uma modulação própria o que a torna perfeita para dispositivos *IoT*.

Modulação LoRa

A modulação utilizada na tecnologia *LoRa* baseia-se na técnica *Chirp Spread Spectrum (CSS)*. Esta técnica recorre ao espalhamento espectral (*Spread Spectrum*), onde o sinal é distribuído por uma ampla gama de frequências, utilizando impulsos do tipo *Chirp* para codificar a informação.

Um *Chirp* é um sinal cuja frequência varia linearmente ao longo do tempo — podendo ser ascendente (*Up-chirp*) para a codificação da informação ou descendente (*Down-chirp*), utilizados para sincronização.

Durante a transmissão de um símbolo, o *Chirp* começa numa determinada frequência dentro da largura de banda disponível e aumenta linearmente até atingir o limite superior. Caso o tempo de símbolo ainda não tenha terminado, a frequência é reiniciada no limite inferior da banda, continuando a subida até completar o tempo de símbolo.

Cada símbolo é identificado pela frequência inicial do seu *Chirp*, o que permite representar múltiplos valores por símbolo. O número de símbolos distintos possíveis depende do *Spreading Factor (SF)*, que pode variar entre 7 e 12, resultando num total de 2^{SF} símbolos únicos.

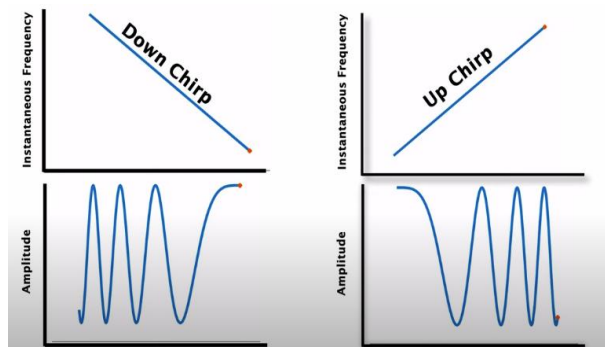


Figura 1 - Up-chirp e Down-chirp ao longo do tempo

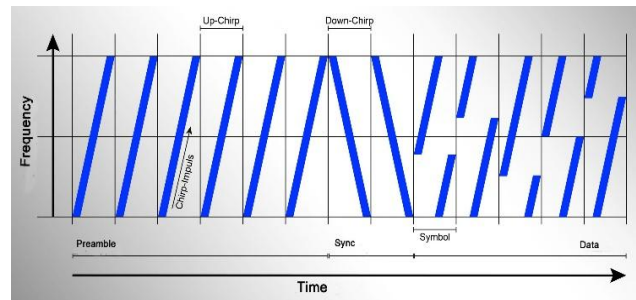


Figura 2 - Mensagem LoRa no domínio da frequência

Para ajudar a compreender estes conceitos, é possível observar, na figura 1, a representação de um *Up-chirp* e de um *Down-chirp* e, na figura 2, uma mensagem *LoRa* no domínio da frequência.

Interferência

A modulação utilizada é o principal fator que garante a resistência às interferências da tecnologia *LoRa*. É importante sublinhar que, devido a esta modulação, sinais *LoRa* com diferentes SFs são ortogonais, podendo ocupar a mesma largura de banda e o mesmo canal, sendo transmitidos, simultaneamente, sem interferirem entre si.

Para diferentes regiões do mundo, estão alocadas diferentes faixas de frequência para a comunicação *LoRa*, como mostramos de seguida.

- **Europa:** EU868 (863–870/873 MHz).
- **América do Norte:** US915 (902–928 MHz).
- **América do Sul:** AU915/AS923-1 (915–928 MHz).
- **Índia:** IN865 (865–867 MHz).
- **Ásia:** AS923 (915–928 MHz).

A comunicação nestas frequências refere-se à comunicação que os *End Devices* estabelecem com as *gateways LoRa*. De um modo geral, a tecnologia *LoRa* é composta por três tipos de dispositivos:

End Devices (Nós LoRa): Sensores/atuadores com chips *LoRa*, que recolhem/enviam dados, podendo operar ponto-a-ponto ou via *LoRaWAN*, como é exemplo o nosso sistema de monitorização, composto pela placa mais os sensores.

Gateways: Intermediários que recebem sinais *LoRa* e retransmitem aos servidores via conexões de alta largura de banda (Ethernet, Wi-Fi, celular).

Servidores de rede: Processam dados recebidos, aplicam a gestão de *routing* e garantem segurança da comunicação.

LoRaWAN

LoRaWAN é o protocolo de comunicação que funciona sobre a tecnologia LoRa. Este define a estrutura da rede, incluindo regras de transmissão, segurança e gerenciamento de dispositivos. O LoRaWAN possibilita a conexão de vários dispositivos IoT a uma infraestrutura centralizada, como gateways e servidores. A fig.3 representa um mapa com as gateways The Things Network disponíveis na área do ISEL.

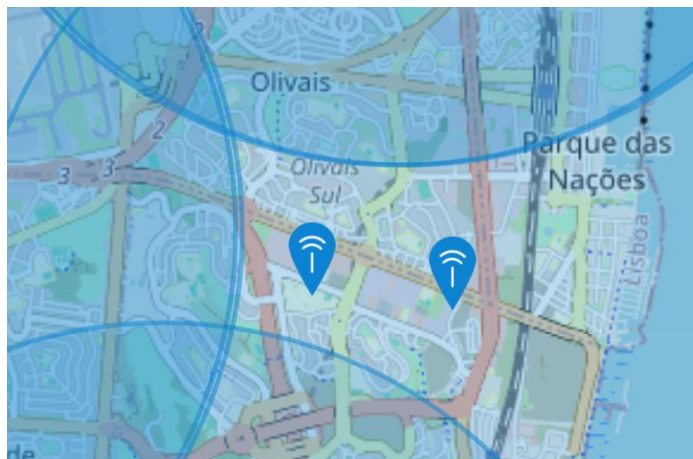


Figura 3 - Mapa de antenas TTN disponíveis na área do ISEL

O protocolo LoRaWan segue regulamentos de espectro de rádio impostos por entidades reguladoras, como o *European Telecommunications Standards Institute (ETSI)*. A principal limitação está no uso de bandas de frequência *Industrial, Scientific and Medical (ISM)*, que operam sem licenciamento e possuem restrições de tempo de ocupação. Não existe um limite estipulado para o número de dispositivos na rede, mas fatores como a densidade local de dispositivos, configurações dos gateways e parâmetros como o SF influenciam diretamente a capacidade do canal e o desempenho da rede. Existe uma restrição de 30 minutos de ocupação do espectro a cada 24 horas, relacionada ao *Duty Cycle*¹, que limita a transmissão contínua para evitar sobrecargas e interferências

¹ Duty Cycle (Ciclo de Trabalho): Proporção de tempo em que um sinal ou dispositivo permanece ativo (ligado) durante um ciclo completo de operação.

Rede LoRaWAN

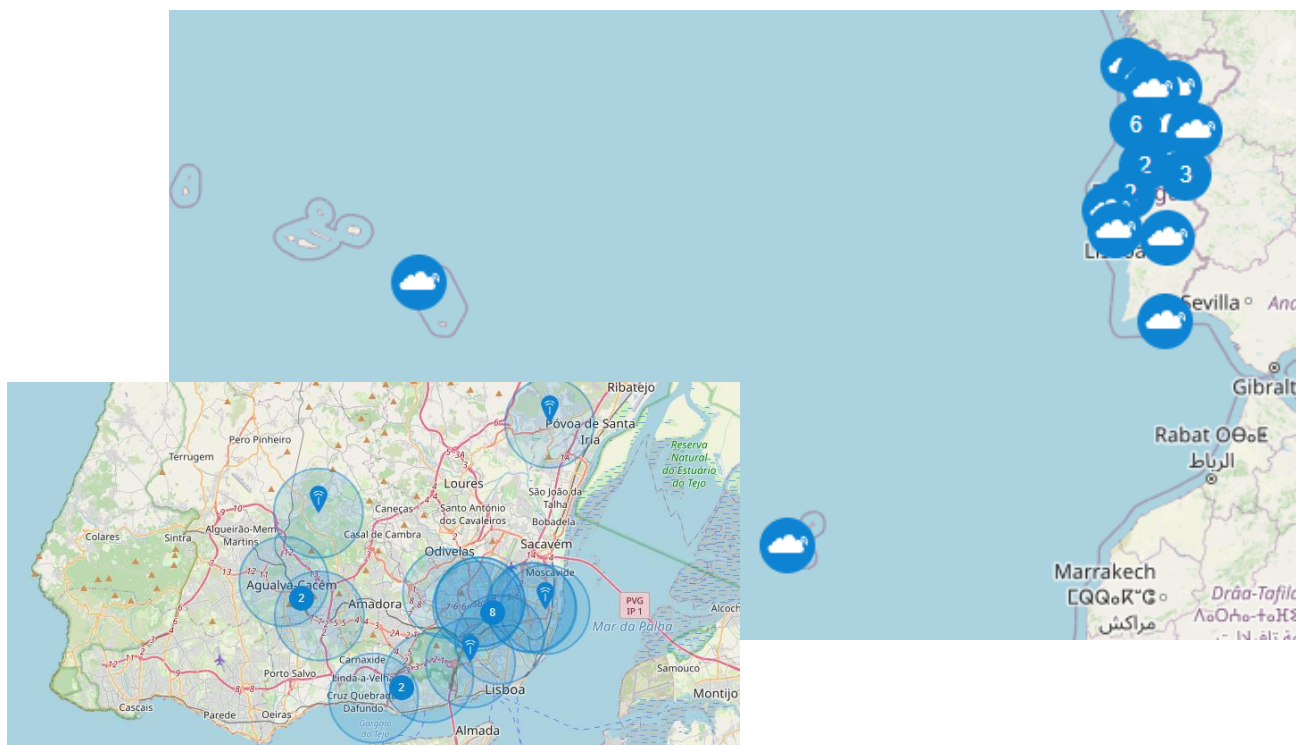


Figura 4 - Distribuição de Comunidades em Portugal e Gateways de Lisboa

Uma rede *LoRaWAN* é uma rede de comunicação de baixa potência e longo alcance, concebida para ligar dispositivos *IoT* a grandes distâncias, consumindo muito pouca energia, onde os dispositivos comunicam entre si utilizando a tecnologia LoRa. Estas redes são compostas por *End Devices*, *Gateways* e Servidores de rede.

A arquitetura da *LoRaWAN* é baseada numa topologia de rede de estrela, onde os dispositivos finais (*End Devices*) comunicam com *gateways* locais, que por sua vez encaminham os dados para uma rede de *Back-End*². Estas *gateways* atuam como pontes entre os dispositivos e a infraestrutura de rede, permitindo que os dados sejam transmitidos de forma eficiente e confiável. Além disso, o *LoRaWAN* oferece uma camada de segurança robusta para proteger a integridade e a confidencialidade dos dados transmitidos. Isso inclui criptografia de ponta a ponta, autenticação de dispositivos e mecanismos de gestão de chaves

² Rede de Back-End: Refere-se à parte da infraestrutura tecnológica que opera nos bastidores de um sistema. No contexto de redes *LoRaWAN*, o Back-End inclui servidores, bases de dados, APIs e lógica de aplicação que processam, armazenam e disponibilizam os dados recebidos dos dispositivos finais.

No nosso projeto, a estação meteorológica está ligada à rede *LoRaWAN TTN*. Esta rede é composta por uma série de comunidades que consistem em grupos locais de utilizadores, entusiastas, investigadores ou empresas que colaboram para expandir e manter a infraestrutura da rede *LoRaWAN* aberta na sua região.

Na figura 4, estão identificadas as *gateways* TTN na zona de Lisboa (imagem menor) e as comunidades *TTN* à escala nacional (imagem maior).

OTAA vs ABP

De modo a conectar um dispositivo LoRa a uma rede *LoRaWAN* é necessário escolher um de dois modos de ativação/ligação possíveis: o modo *Over-The-Air Activation (OTTA)* e modo *Activation By Personalization (ABP)*. Em seguida, será especificado o funcionamento de cada uma das formas de conexão do dispositivo *LoRa* à rede *LoRaWAN*, de modo a justificar mais claramente a nossa escolha.

OTAA

Os dispositivos que implementam OTAA para se conectarem à rede devem passar pelo processo *Join* com um servidor da rede *LoRaWAN* a que se querem conectar. Este procedimento ocorre, obrigatoriamente, sempre que o dispositivo é ligado. Durante o mesmo são negociadas, dinamicamente, as chaves da sessão (*Network Session Key- NwkSKey* e *Application Session key- AppSKey*) que será estabelecida entre o dispositivo e o servidor, tendo por base as chaves raiz pré-compartilhadas *Application Key (AppKey)* e *Application Extended Unique Identifier (AppEUI)/ Join Extended Unique Identifier (JoinEUI)*. No decorrer desta operação, é atribuído um endereço dinâmico e único dentro da rede ao dispositivo, *Device Address (DevAddr)*, que pode ser alterado a cada nova sessão.

Para ajudar a compreender melhor este processo, é apresentada a sua ilustração na figura 5

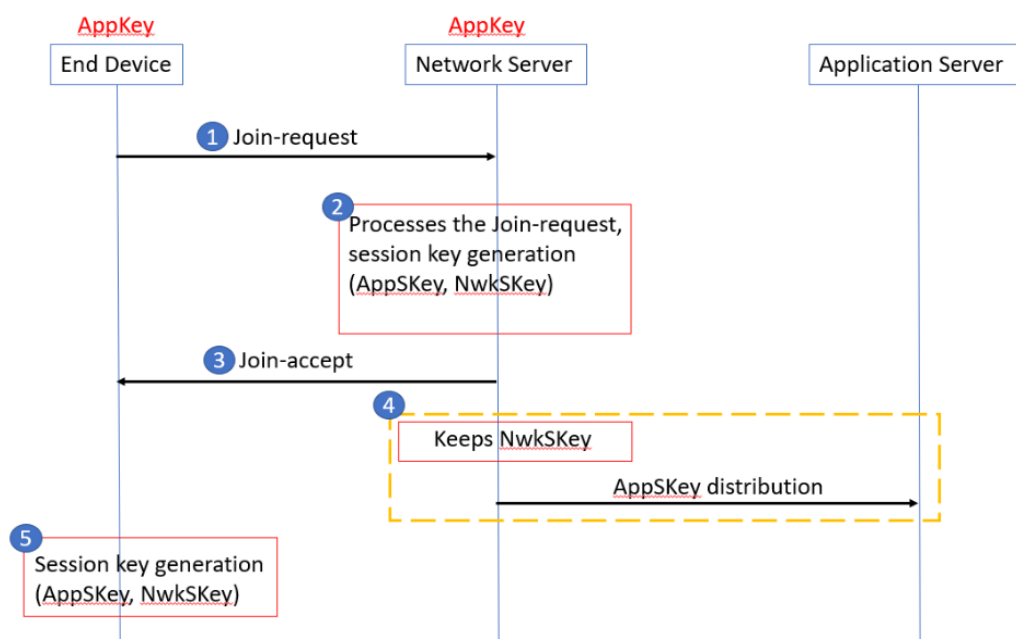


Figura 5 - Ilustração do processo "Join"

ABP

No caso dos dispositivos que implementam o *ABP*, não há processo *Join*. As chaves de sessão *NwkSKey* e *AppSKey* são pré-configuradas (*hardcoded*), no dispositivo, durante a sua fabricação, assim como o seu endereço *DevAddr*, que permanece o mesmo durante toda a sua vida útil.

Comparação entre o OTAA e o ABP

Comparando os dois métodos podemos observar o porquê de o *OTAA* ser o mais recomendado. O *OTAA* é mais seguro, dado que as chaves utilizadas na comunicação entre o dispositivo e o servidor são alteradas a cada nova sessão estabelecida, ao passo que, no *ABP*, estas mantêm-se imutáveis, a não ser que sejam manualmente configuradas (*hardcoded*). Por outro lado, dado que, no *ABP*, o *DevAddr* é configurado durante o processo de fabricação e mantém-se constante durante toda a vida do dispositivo, pode ocorrer que, quando o dispositivo se tente conectar a uma rede *LoRaWAN*, o seu *DevAddr* já esteja a ser utilizado por outro dispositivo, levando isto a colisão de pacotes, comportamento indefinido da rede e à impossibilidade de comunicação confiável. Para finalizar, no *ABP* temos o problema de escalabilidade, uma vez que o controlo da unicidade de cada uma das chaves utilizadas nas comunicações entre os diversos dispositivos e os servidores é responsabilidade de quem está a configurar o *ABP*, enquanto toda esta gestão é feita automaticamente no *OTAA*.

Decidiu-se utilizar, no projeto, o *OTAA*, devido à sua flexibilidade e segurança superiores. Este método simplifica o processo de ativação ao permitir que dispositivos *LoRa* gerem chaves dinâmicas e se autenticuem automaticamente na rede, garantindo maior proteção contra acessos não autorizados e facilitando a gestão de dispositivos em larga escala.

3. Arduino IDE

A plataforma *Arduino Integrated Development Environment (IDE)*, foi a plataforma que utilizamos para desenvolver o código da nossa placa. Esta é uma ferramenta intuitiva e essencial para programadores e entusiastas de sistemas como o Arduino. Tem como vantagem a facilidade de uso, mesmo para iniciantes, permitindo interação rápida e eficiente com diferentes placas microcontroladoras.

A plataforma permite a detecção automática de portas série ou outras interfaces conectadas, que de outro modo teriam de ser realizadas manualmente através da instalação de *drivers*(controladores). Isso simplifica o processo de configuração e garante uma experiência fluida, ao programar e testar novos projetos com diferentes tipos de placas e sensores.

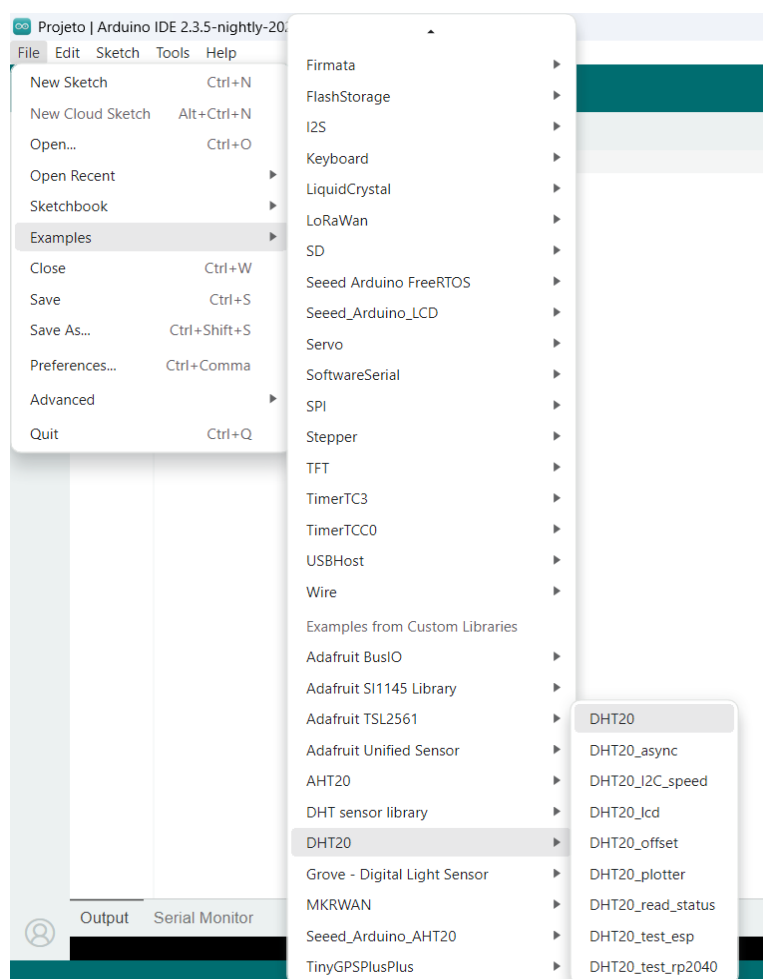


Figura 6 - Principais Componentes Arduino IDE

A existência de exemplos genéricos pré-instalados é outro fator positivo (Fig. 6), fornecendo, aos utilizadores, testes para componentes como sensores, *Light*

Emitting Diodes (LEDs), e outros elementos. Isso ajuda a acelerar o desenvolvimento, validação e a familiarização com o hardware.

Para projetos mais avançados, a Arduino IDE inclui ferramentas robustas para *debugging* (Fig. 7) e interação com a placa, como leitura de saídas no Monitor Serial, atualizações de *firmware* e funções de diagnóstico. Essas capacidades são cruciais, tanto para corrigir erros, quanto para monitorizar a execução de programas em tempo real.

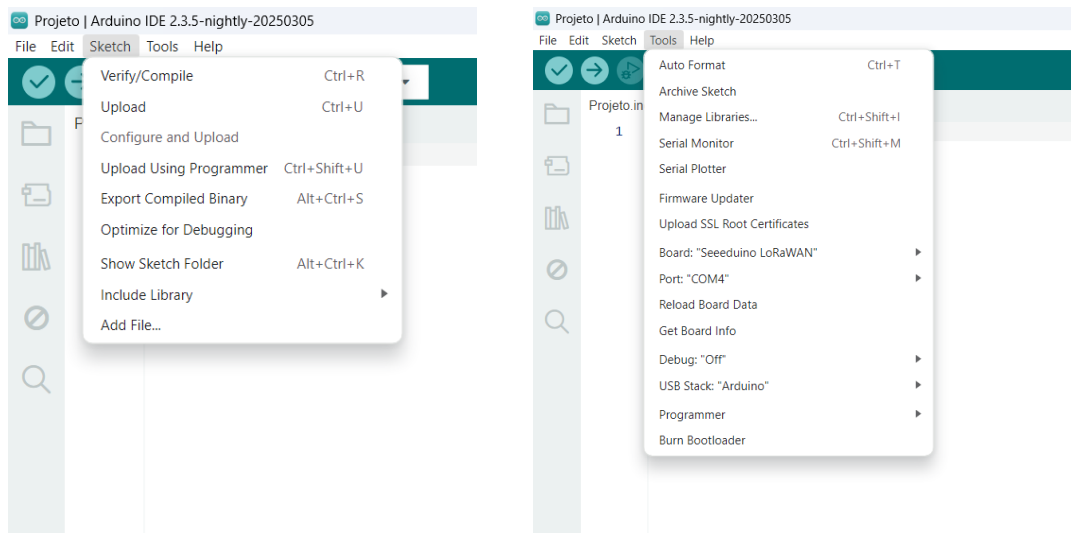


Figura 7 - Menu Sketch e Tools Arduino IDE

Outra diferença significativa é a gestão simplificada de bibliotecas. O IDE oferece um gestor de bibliotecas integrado, a partir do qual podemos transferir a maioria das bibliotecas de que necessitamos para os nossos projetos, permitindo eliminar a necessidade de pesquisar e descarregar arquivos de páginas web de terceiros, otimizando mais uma vez o fluxo de trabalho.

4. Detalhes da Arquitetura do Sistema

A nossa estação meteorológica é composta por diversos elementos (Fig. 8), que trabalham de forma integrada para fornecer as medições climatéricas necessárias. De seguida, estes serão apresentados e descritos em detalhe, destacando o papel de cada um no funcionamento do sistema.

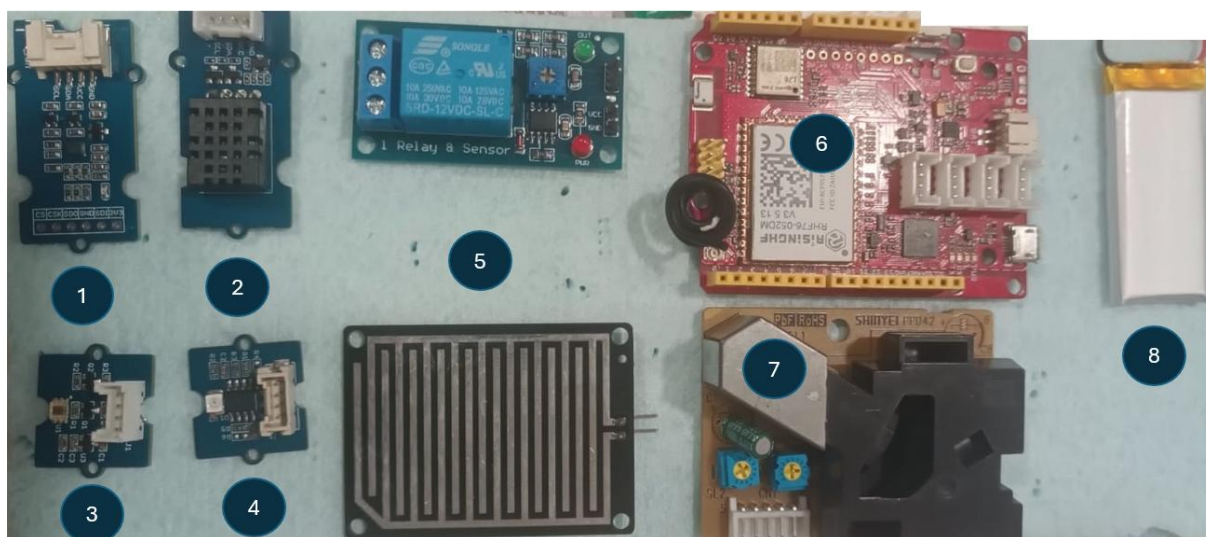


Figura 8 - Componentes da Estação meteorológica

Na fig. 8 para facilitar a visualização dos componentes, os cabos do sistema foram retirados.

Fig. 8 – (1) Sensor de pressão atmosférica (2) Sensor de Temperatura e humidade (3) Sensor de Luz solar (4) Sensor de radiação Ultravioleta (5) Sensor de precipitação + Relé de controlo (6) Placa Seeeduino LoRaWAN c/ GPS (7) Sensor de Qualidade do Ar (8) bateria de Polímero de Litio

Placa Seeeduino LoRaWAN

A *Seeeduino LoRaWAN* é uma placa de desenvolvimento Arduino que integra de forma nativa o protocolo *LoRaWAN* e pode ser observada na figura 9. O seu módulo de comunicação RHF76-052AM é um componente fundamental que simplifica o desenvolvimento de aplicações *LoRa* e permite aos utilizadores preocupar-se com a lógica da aplicação em si. Esta placa versátil suporta as classes A e C do protocolo *LoRaWAN*, proporcionando flexibilidade para se adaptar a diferentes tipos de aplicações

e cenários de uso. Além disso, opera em uma variedade de frequências de comunicação, garantindo a sua adaptabilidade a diversas regiões e regulamentações. Um dispositivo pertencente à classe A pode enviar uma mensagem *uplink* (sentido ascendente) a qualquer momento. Após a transmissão ser efetuada, o dispositivo abre duas janelas temporais pequenas, para receber mensagens *downlink* (sentido descendente) da rede. Se não forem recebidas quaisquer mensagens *downlink*, o próximo período para receber mensagens *downlink* será agendado a seguir ao envio da próxima mensagem *uplink*. Um dispositivo da classe C, mantém a janela temporal para a recepção de mensagens *downlink* sempre aberta, com a exceção de quando quer transmitir uma mensagem *uplink*.

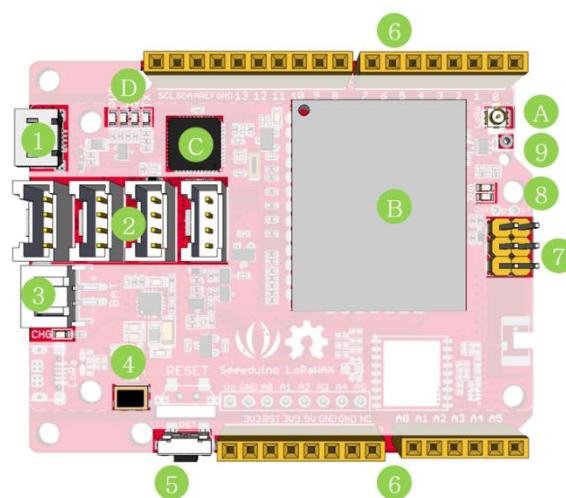


Figura 9 - Placa Seeeduino LoRaWAN

O núcleo da *Seeeduino LoRaWAN* é impulsionado pelo processador *ARM Cortex M0* (C), um microcontrolador de 32 bits que opera a uma frequência de 48MHz. O módulo *LoRaWAN* (B) bem como a antena (A) são responsáveis pela comunicação LoRa, proporcionando alta sensibilidade e potência de saída ajustável.

A placa oferece uma variedade de interfaces de conexão, incluindo *Micro Universal Serial Bus (USB)* (1) para programação e alimentação, conectores *Grove* para sensores e atuadores (2), uma entrada para uma bateria *Lithium Polymer Battery (LiPo)* para alimentação autônoma (3), e pinos *Arduino* e *In-Circuit Serial Programming (ICSP)* para expansão e depuração (6). *LEDs* de status (D), um botão de *reset* (5) e um botão *Device Firmware Update (DFU)* (4) complementam as funcionalidades da placa. A fig. 9, presente na ficha técnica online da placa, mostra a distribuição destes componentes na placa.

Sensores para Monitorização Meteorológica

Durante a escolha dos componentes, demos prioridade a sensores mais gerais, que nos permitissem realizar a medição de valores mais cruciais para o bem-estar e planeamento da vida quotidiana. As medições apresentadas diferem das nossas escolhas iniciais apenas na componente de pressão atmosférica que inicialmente considerámos menos essencial do que, por exemplo, a medição da intensidade e direção do vento; mas, pela dificuldade de aquisição, pelo preço elevado e dificuldade de incorporação no projeto, dado o seu tamanho e tipo de conectores utilizados, foi substituído por um barómetro.

Sensor de Temperatura e humidade (DHT20 Grove)

O Grove DHT20 (fig. 10) é um sensor digital de temperatura e humidade, reconhecido pela sua alta performance e estabilidade em ambientes de alta temperatura e humidade. Este dispositivo integra um chip *Application-Specific Integrated Circuit (ASIC)* dedicado e um sensor capacitivo de humidade, juntamente com um sensor de temperatura integrado, oferecendo uma saída digital calibrada.

O sensor (fig.11) mede a humidade relativa através do sensor capacitivo e mede a temperatura utilizando um termistor³. O sensor capacitivo⁴ de humidade consiste num material dielétrico higroscópico⁵ entre dois elétrodos. A absorção de vapor de água por este material altera a sua constante dielétrica⁶, o que, por sua vez, modifica a capacitância do sensor. Esta variação é proporcional à humidade relativa.

Para a temperatura, o termistor, um dispositivo semicondutor, apresenta uma alteração na sua resistência elétrica em função da temperatura ambiente, permitindo uma medição precisa da temperatura do meio em que está inserido.

³ Termistor: Resistor de materiais semicondutores cuja resistência elétrica varia com a temperatura.

⁴ Capacitivo/Capacitância: Capacidade de armazenar carga elétrica entre dois condutores separados por um material isolante (dielétrico). É medida em farads (F) e usada em sensores, circuitos e eletrónica para detetar variações de proximidade, toque ou campo elétrico.

⁵ Higroscópico: Propriedade de uma substância de atrair e reter moléculas de água do ambiente por absorção ou adsorção.

⁶ Dielétrico: Material isolante elétrico que pode ser polarizado por um campo elétrico. Em sensores capacitivos, o material dielétrico é sensível à humidade, alterando as suas propriedades elétricas em função da quantidade de vapor de água absorvida.

A escolha do Grove DHT20, sensor presente na fig.10, é vantajosa devido à sua alta precisão, sendo as margens de erro reduzidas ($\pm 3\%$ para humidade, $\pm 0.5^{\circ}\text{C}$ para temperatura). Apesar das suas vantagens, o DHT20 apresenta algumas limitações. A exposição prolongada a humidade acima de 80% pode causar um desvio temporário no sinal de saída, embora o sensor se recupere lentamente em condições normais. A sua precisão pode ser afetada por vapores químicos voláteis, exigindo um ambiente de armazenamento e operação bem ventilado. Adicionalmente, a frequência de medição deve ser controlada para evitar o auto aquecimento do sensor, que pode comprometer a precisão da temperatura.

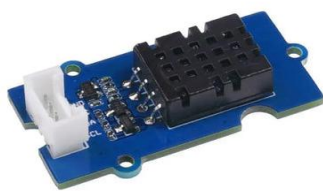


Figura 10 - Sensor Grove DHT20

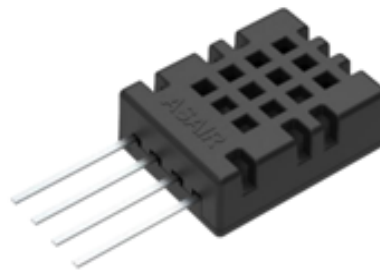


Figura 11 - Modulo DHT20

Sensor de pressão atmosférica (DPS310 Grove)

O DPS310 é um sensor barométrico digital de pressão atmosférica e temperatura, miniaturizado e de alta precisão. O elemento sensor de pressão do DPS310 (fig.12) baseia-se num princípio de deteção capacitivo, utilizando um diafragma de silício flexível sobre uma cavidade de ar. A variação da pressão atmosférica provoca uma deformação neste diafragma, alterando a capacitância do sensor. Esta alteração é convertida num sinal elétrico e, posteriormente, digitalizada por um processador de sinal interno. O sensor também inclui um elemento de temperatura para compensar a sensibilidade da pressão à temperatura, garantindo maior precisão.

O sensor pode operar em três modos:

- *Standby*,
- *Command* (medição única)
- *Background* (medição contínua).

Cada medição é calibrada individualmente, e os coeficientes de calibração são armazenados no sensor para correção das medições.

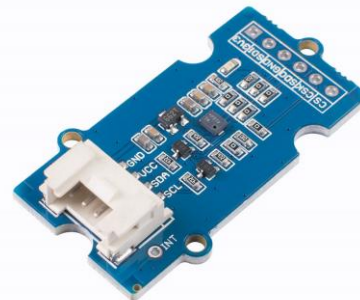


Figura 12 . Sensor de pressão atmosférica Grove DPS310

A alta precisão do DPS310, conferida pelas suas pequenas margens de erro, (± 0.002 Hectopascal (hPa) (medida de pressão equivalente a 0.001 BAR) para pressão, $\pm 0.5^\circ\text{C}$ para temperatura) é fundamental para uma estação meteorológica, permitindo medições detalhadas de pressão atmosférica e altitude. O seu baixo consumo de corrente ($1.7\ \mu\text{A}$ para pressão a 1Hz) é crucial para aplicações de baixa potência, como a estação meteorológica alimentada por bateria. A flexibilidade das interfaces *Inter-Integrated Circuit (I²C)* e *Serial Peripheral Interface (SPI)*, juntamente com a capacidade de armazenamento *First in First Out (FIFO)*⁷, otimiza a eficiência da comunicação e o consumo de energia do sistema.

A precisão dos sensores barométricos pode ser afetada por flutuações de temperatura e humidade, embora o DPS310 inclua compensação de temperatura. A estabilidade a longo prazo pode ser comprometida por ciclos contínuos de pressão e temperatura, que podem causar um desvio nos materiais do sensor ao longo do tempo. É essencial que o orifício de ventilação do sensor não seja exposto a vácuo ou sopro de ar, pois isso pode danificar a estrutura *Micro-Electro-Mechanical Systems (MEMS)*⁸.

Sensor da Qualidade do Ar (SHINYEI PPD42NS)

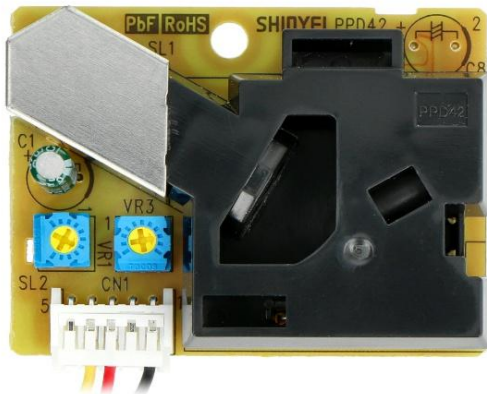


Figura 13 - SHINYEI PPD42NS

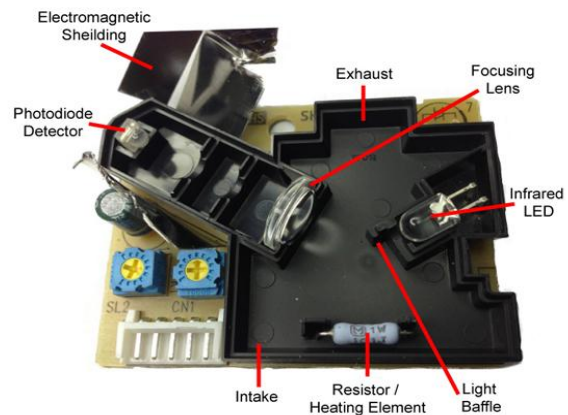


Figura 14 - Ilustração do mecanismo de leitura de partículas

O SHINYEI PPD42NS (fig. 13) é um sensor de qualidade do ar projetado para detetar partículas em suspensão, baseando-se no método de dispersão de luz.

⁷ FIFO: First In, First Out / Primeiro a Entrar, Primeiro a Sair- É uma técnica de gestão de dados ou recursos em que o primeiro elemento inserido é o primeiro a ser processado ou retirado

⁸ MEMS: Tecnologia que integra elementos mecânicos, sensores e eletrônicos em um pequeno chip, que possui uma informação gravada que determina seu funcionamento.

Internamente, o PPD42NS contém um emissor de luz (LED infravermelho) e um recetor ótico (fototransistor) com lentes que focam o feixe de luz num ponto de deteção. Um resistor aquecedor gera uma corrente de ar ascendente, que aspira as partículas para dentro da caixa do sensor. Quando as partículas atravessam o feixe de luz, parte da luz é dispersa e captada pelo recetor. A intensidade da luz dispersa é diretamente proporcional à concentração de partículas no ar, permitindo ao sensor determinar a sua quantidade todo este processo pode ser mais fácil de compreender ao analisar a figura 14 que identifica os componentes.

O sensor fornece uma saída digital sob a forma de um sinal de modulação por largura de pulso *Pulse Width Modulation (PWM)*⁹ onde o tempo de ocupação do pulso baixo *Low Pulse Occupancy Time (LPO Time)*¹⁰ é proporcional à concentração de partículas. O sensor tem dois pinos de saída ('P1' para partículas maiores que 1 µm e 'P2' para partículas maiores que 2.5), permitindo aqui a distinção entre diferentes tamanhos de partículas. A interação com o microcontrolador envolve sucessivas leituras para inferir a concentração de partículas, sendo que o sensor requer uma alimentação de 5V DC alimentados no barramento da placa.

A capacidade do sensor de detetar partículas a partir de 1 µm, incluindo fumo de cigarro e pó doméstico, torna-o valioso para a avaliação da qualidade do ar. A sua robustez e facilidade de manutenção são aspetos práticos importantes para um projeto de estação meteorológica a longo prazo. A saída *PWM*, embora exija processamento para converter o ciclo de trabalho em concentração, permite uma representação eficaz da densidade de partículas.

O desempenho de sensores de partículas baseados em dispersão de luz pode ser influenciado por fatores como a humidade relativa e as propriedades das partículas (forma, cor, índice de refração). O equipamento pode apresentar correlações fracas com instrumentos de referência em baixas concentrações de Particulate Matter (PM)_{2.5} (< 20 µg m⁻³) e é suscetível a luz difusa, exigindo proteção adequada. A limpeza periódica da lente é essencial, pois o acúmulo de sujidade pode afetar a sensibilidade do sensor.

Sensor de radiação UV (Grove- UV Sensor)

O Grove - UV Sensor (fig. 15) consiste num sensor analógico concebido para a medição da intensidade da radiação ultravioleta (UV). Este sensor apresenta uma sensibilidade de comprimentos de onda na gama compreendida entre aproximadamente 240 nm e

⁹ PWM: modulação por largura de pulso (PWM na sigla em inglês) é uma técnica usada para controlar a quantidade de energia fornecida a um dispositivo eletrónico

¹⁰ LPO Time: Low Pulse Occupancy time (LPO time) é proporcional à concentração de partículas.

370 nm. Esta banda de sensibilidade abrange parcialmente as faixas de radiação UVC, UVB e UVA do espectro eletromagnético UV.

Para efeitos de contextualização e classificação, a radiação ultravioleta é convencionalmente subdividida em três bandas principais, como foi referido anteriormente, diferenciadas pelos seus comprimentos de onda:

- UVC: 100–280 nm
- UVB: 280–315 nm
- UVA: 315–400 nm

O funcionamento do sensor baseia-se no princípio fotoelétrico, utilizando um fotodíodo sensível à radiação ultravioleta. Este componente tem a capacidade de converter a radiação UV incidente num sinal elétrico. O sinal de saída do sensor manifesta-se como uma tensão analógica, que é diretamente proporcional à intensidade da radiação UV que incide sobre o fotodíodo¹¹.

Esta característica de saída permite a monitorização contínua e em tempo real da exposição à radiação UV. A tensão analógica variável gerada pelo sensor pode ser adquirida e processada por um microcontrolador ou por um sistema de aquisição de dados. Através da calibração adequada, a leitura desta tensão possibilita a estimação do índice UV ou a avaliação dos níveis de exposição à radiação ultravioleta num determinado ambiente.

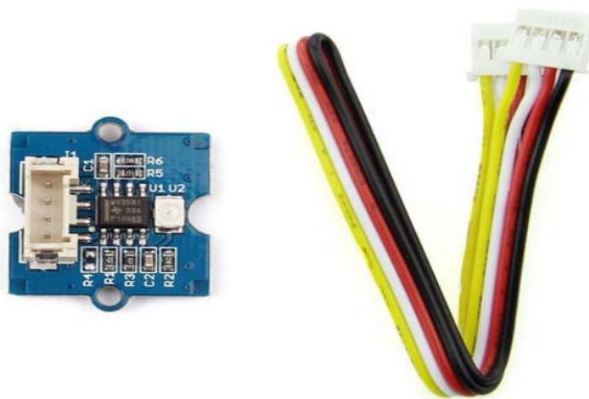


Figura 15 - Sensor Grove UV sensor e cabo Grove

¹¹ Fotodíodo: Dispositivo semiconductor que converte luz em corrente elétrica. Quando é atingido por um fóton com energia suficiente, gera um sinal elétrico proporcional à intensidade da luz incidente. Nos sensores de UV, este sinal serve de base para medir a radiação ultravioleta.

Sensor de Luz solar (Grove – Digital Light Sensor)

O Grove – Digital Light Sensor(fig.16) é um sensor digital de luminosidade que mede a intensidade da luz ambiente e a converte num valor de lux, unidade padrão de iluminância. Ao contrário dos sensores analógicos, este transmite os dados sob a forma de um sinal digital.

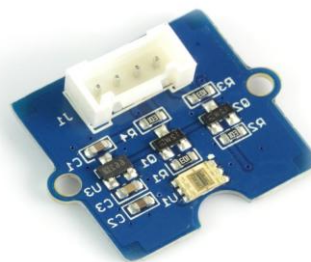


Figura 16 - Sensor Grove Digital Light v1.1

A leitura de dados do sensor é feita à medida que os fótons incidem sobre os dois fotodíodos incorporados no sensor, um sensível apenas à luz

infravermelha e outro sensível à luz visível e infravermelha. A energia luminosa é assim convertida em corrente elétrica pelos fotodíodos. Posteriormente, essas duas correntes são convertidas em sinais digitais por dois conversores ADC (análogo–digital), e transmitidas via protocolo I²C para um microcontrolador. Este, por sua vez, executa um algoritmo que combina os dados de ambos os canais para gerar um valor de iluminância em lux.

Este sensor pode operar em três modos distintos:

- Medição da intensidade da radiação infravermelha (IV)
- Medição da intensidade da luz visível
- Medição da intensidade do espectro combinado de luz visível e infravermelha

Optámos pela medição da luz visível por ser aquela que melhor representa a claridade tal como é percebida pelo olho humano, evitando distorções provocadas pela radiação infravermelha e garantindo maior precisão na análise do ambiente.

Sensor de Chuva (M445C)

O M445C é um sensor analógico utilizado na deteção da presença de precipitação (chuva) e na estimativa da sua intensidade no ambiente onde está instalado. Estruturalmente, é composto por uma superfície sensível dotada de trilhas condutoras paralelas adjacentes que, em condições de ausência de humidade, permanecem eletricamente isoladas.

Na ausência de precipitação, a tensão de saída é nominalmente equivalente à tensão de alimentação aplicada ao sensor. Com o aumento da quantidade de água acumulada na superfície sensível, a resistência elétrica entre as trilhas decresce progressivamente, permitindo um maior fluxo de corrente e, conseqüentemente,

resultando numa diminuição da tensão de saída. Esta relação inversa entre a quantidade de água e a tensão de saída permite inferir a intensidade da precipitação com base na leitura do sinal analógico.

A conexão elétrica do sensor envolve um ponto de alimentação (3,3 volts do barramento da placa) e um terminal de saída analógica, o qual deve ser conectado a um pino de conversão Analógico-Digital (ADC) do microcontrolador para leitura da tensão gerada. Sem a presença de água na superfície sensível, a resistência é elevada e a tensão medida no terminal de saída aproxima-se do valor da tensão de alimentação (3,3 V). Quando a água está presente, a diminuição da resistência facilita a passagem de corrente pelo sensor, resultando numa redução proporcional da tensão lida (por exemplo, 1,1 V) como ilustrado na fig.17 abaixo.

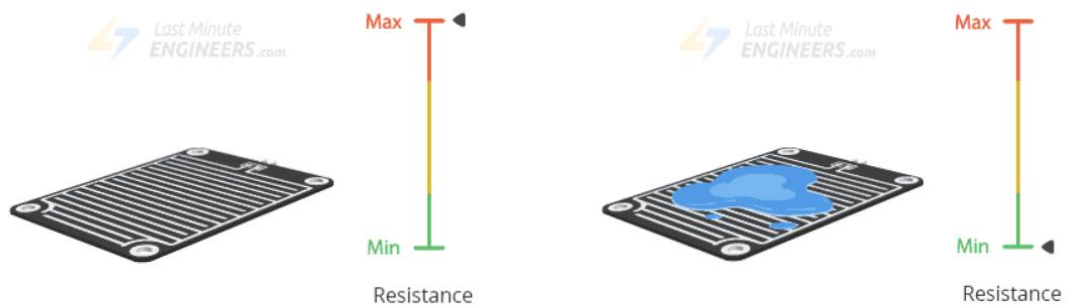


Figura 17 - Ilustração da variação de resistência

Bateria Lipo e Painel Solar

Para dotar a estação meteorológica de autonomia energética e permitir a sua instalação em locais remotos sem acesso direto à rede elétrica, o sistema foi concebido para operar com uma combinação de bateria recarregável (Figura 8, componente 8) e um painel solar (Figura 18) para colheita de energia.

A componente de armazenamento de energia é assegurada por uma bateria de polímero de lítio (*LiPo*). Este tipo de bateria foi escolhido pela sua densidade de energia e capacidade de fornecer a energia necessária para o funcionamento contínuo dos sensores e do módulo de comunicação *LoRaWAN*. A bateria é conectada diretamente à placa *Seeedduino LoRaWAN* através do conector JST 2.0 dedicado, conforme indicado na Figura 9 (elemento 3), que é a entrada específica para baterias de lítio nesta placa.

A recarga da bateria *LiPo* é realizada através de um painel solar. Este painel converte a energia luminosa do sol em energia elétrica, que é então utilizada para alimentar o sistema e, crucialmente, para carregar a bateria. O painel solar é conectado à placa *Seeedduino LoRaWAN* pela porta *micro-USB* (Figura 9, elemento 1). Esta porta serve como a entrada principal de energia para a placa.



Figura 18 - Painel Solar 1W 5V 200mA - 110x80mm

A placa *Seeedduino LoRaWAN* possui um circuito de gestão de carregamento integrado. Este circuito é responsável por regular o processo de carga da bateria *LiPo* conectada ao conector JST 2.0. Ele utiliza a corrente fornecida através da entrada *micro-USB* (proveniente do painel solar) para carregar a bateria de forma eficiente e segura, monitorizando a tensão e a corrente para proteger a bateria contra sobrecarga ou descarga excessiva. Desta forma, a placa gere automaticamente a transição entre a alimentação direta pelo painel solar (quando há luz suficiente) e o uso da energia armazenada na bateria (em períodos sem sol), garantindo a operação contínua da estação meteorológica.

Protocolos de Ligação dos Sensores

No nosso projeto foram integrados vários sensores, tanto analógicos como digitais. A placa utilizada, conforme descrito anteriormente, permite a ligação direta de até dois

sensores digitais compatíveis com o protocolo I^2C (via cabos *Grove*), bem como dois sensores analógicos (ver Fig. 9). Adicionalmente, é possível expandir o número de sensores através do barramento, mediante a correta gestão dos endereços e recursos disponíveis.

Os sensores analógicos, consoante o valor que medem, emitem uma determinada tensão para a placa *Seeeduino LoRaWAN*. A placa interpreta este valor, através do código Arduino que desenvolvemos, e sabe que o valor de voltagem que recebeu do sensor corresponde diretamente a um valor medido.

No caso dos sensores digitais, o sensor mede uma grandeza física e converte essa grandeza num sinal digital. Este sinal digital tem dois estados, cada um com uma voltagem própria, o estado *high* e o estado *low*, representando um destes estados o valor binário 1 e o outro estado representando o valor binário 0. Estes *bits* no seu conjunto são a informação que é transmitida para a nossa placa (como números, ou caracteres).

Para a comunicação entre os sensores e a placa ser assegurada ambos têm de ser compatíveis com um protocolo de comunicação, neste caso o I^2C . Em seguida, detalharemos o seu funcionamento.

Protocolo I^2C :

O protocolo I^2C utiliza dois fios para a comunicação entre um dispositivo mestre e múltiplos dispositivos escravos (como sensores). A linha *Serial Clock (SCL)* é unidirecional, gerada pelo mestre para sincronizar a comunicação com os escravos. A linha *Serial Data (SDA)* é bidirecional, utilizada para a transferência de dados entre o mestre e os escravos.

A comunicação é sempre iniciada pelo mestre, que gera uma condição *START* ao forçar a linha *SDA* para o estado *Low* enquanto o sinal *SCL* está no estado *High*, alertando os escravos para o início de uma nova comunicação. Em seguida, enquanto o sinal de *clock (SCL)* está em *High*, o mestre transmite um bit de informação por vez através da linha *SDA*. Os primeiros *bits* transmitidos formam o endereço do escravo (7 ou 10 bits).

Após o envio do endereço, o mestre envia um *bit* de leitura/escrita (*R/W*) para indicar a direção da transferência de dados. O escravo cujo endereço corresponde ao enviado pelo mestre responde com um *bit* de *Acknowledge (ACK)*, puxando a linha *SDA* para *Low* durante o próximo pulso de *clock*.

A transferência de dados (*bytes*) prossegue. Após cada *byte* transmitido (pelo mestre na escrita, ou pelo escravo na leitura), o recetor (escravo na escrita, ou mestre na leitura) geralmente envia um *ACK*. No caso da leitura, o mestre envia um *Negative Acknowledge (NACK)* após receber o último *byte* desejado.

Para finalizar a comunicação, o mestre gera uma condição *STOP*, permitindo que a linha *SDA* transacione para o estado *High* enquanto o *SCL* está no estado *High*, sendo que o estado *High* no fio *SDA* significa que nenhuma comunicação está a ocorrer.

Na figura 19, podemos observar a representação da troca de mensagem entre o mestre e o escravo através da variação do sinal do *SDA* e do *SCL*. Na imagem, o mestre começa por enviar uma condição *START*, seguido do endereço do escravo com o qual quer comunicar. Posteriormente, envia o *bit* que indica se realizará uma escrita ou uma leitura. O escravo responde com um *ACK*. No momento seguinte, no caso de ser uma leitura, o escravo envia informação para o mestre; no caso de se tratar de uma escrita, o mestre envia a informação para o escravo. Em ambos os casos, quem está a receber informação envia um *ACK* a cada *byte* que recebe. Por fim, o mestre envia uma condição *STOP* para encerrar a ligação.

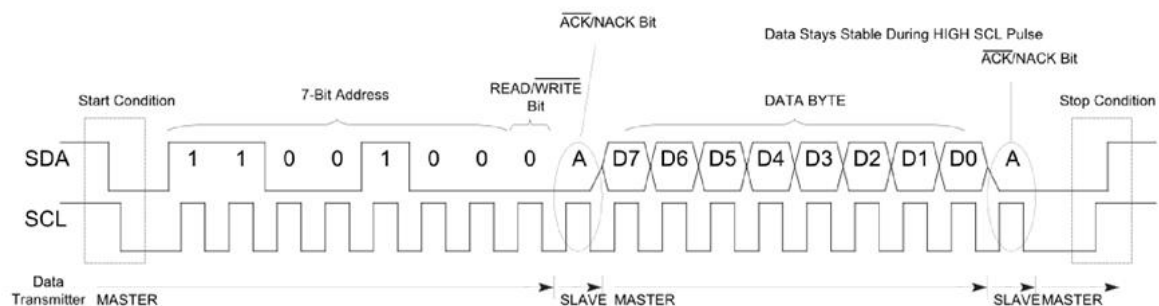


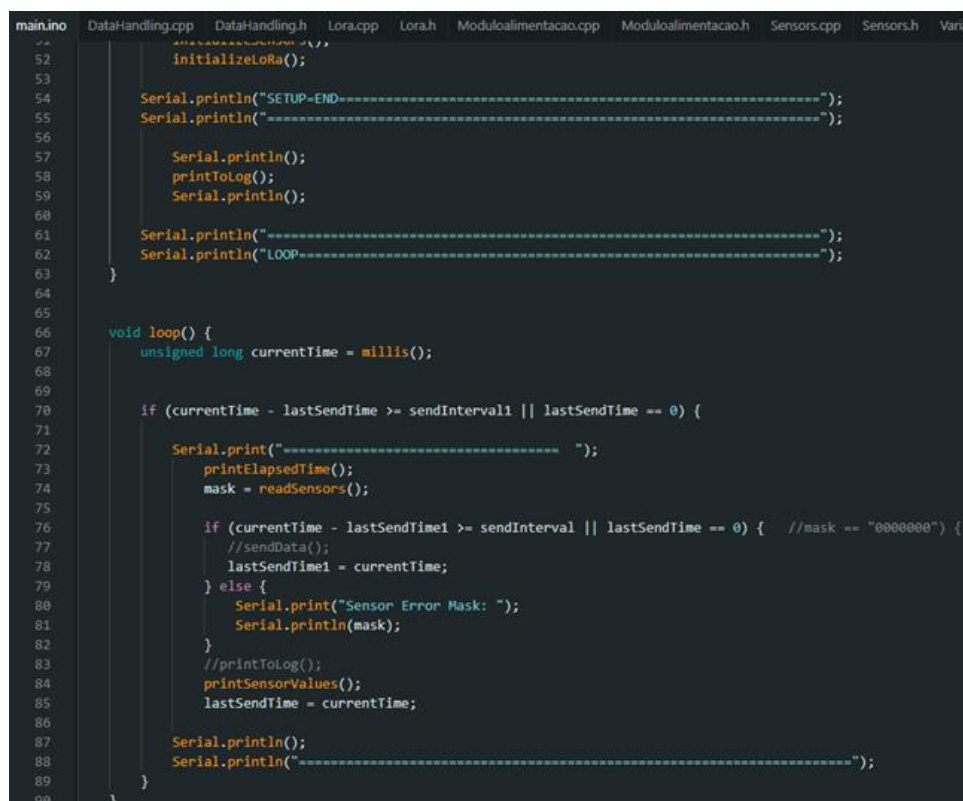
Figura 19 - Variação do sinal SDA e SCL

Análise da Leitura de dados e Interação Placa/PC

Este capítulo detalha o processo de aquisição e processamento dos dados dos sensores pela placa *Seeeduino LoraWan*, bem como a forma como estes dados são apresentados localmente para monitorização e validação através do *Serial Monitor* do ambiente *Arduino IDE*.

A lógica central para a leitura e processamento dos dados dos sensores reside no *sketch* (programa) desenvolvido para a placa Arduino anexado em conjunto com a entrega do projeto. Este script é responsável por inicializar os diversos componentes, configurar as interfaces de comunicação e, ciclicamente, ler os valores fornecidos pelos sensores conectados. O código utiliza bibliotecas específicas para cada tipo de sensor (como as do DHT20 para temperatura/humidade, DPS310 para pressão, sensores de luz/UV e qualidade do ar, e o *tinygpsplusplus* para *GPS*), abstraindo a complexidade da comunicação de baixo nível.

No *loop* principal do programa, é implementado um temporizador (*sendinterval1* = 10 segundos) para garantir que as leituras dos sensores e a atualização da interface de monitorização local ocorram a intervalos regulares.



```
main.ino DataHandling.cpp DataHandling.h Lora.cpp Lora.h Moduloolimentacao.cpp Moduloolimentacao.h Sensors.cpp Sensors.h Variab
52 initializeLora();
53
54 Serial.println("SETUP-END=====");
55 Serial.println("=====");
56
57 Serial.println();
58 printToLog();
59 Serial.println();
60
61 Serial.println("=====");
62 Serial.println("LOOP=====");
63 }
64
65
66 void loop() {
67     unsigned long currentTime = millis();
68
69
70     if (currentTime - lastSendTime >= sendInterval1 || lastSendTime == 0) {
71
72         Serial.print("===== ");
73         printElapsedTime();
74         mask = readSensors();
75
76         if (currentTime - lastSendTime1 >= sendInterval || lastSendTime == 0) { //mask == "0000000" {
77             //sendData();
78             lastSendTime1 = currentTime;
79         } else {
80             Serial.print("Sensor Error Mask: ");
81             Serial.println(mask);
82         }
83         //printToLog();
84         printSensorValues();
85         lastSendTime = currentTime;
86
87         Serial.println();
88         Serial.println("=====");
89     }
90 }
```

Figura 20 – Código Arduino

A função `readsensors()` presente na *fig.20*, chamada dentro deste ciclo, interage com cada sensor ativo, obtendo as medições mais recentes (temperatura, humidade, pressão atmosférica, estado de chuva, índice de qualidade do ar, intensidade de luz, intensidade UV, coordenadas *GPS*) e armazenando-as nas respetivas variáveis globais. A interação direta com o *Personal Computer (PC)* é estabelecida através da conexão *USB*, permitindo a comunicação via porta série. O ambiente *Arduino IDE* oferece o *Serial Monitor*, uma ferramenta indispensável para depuração e acompanhamento em tempo real do funcionamento do sistema. Através de chamadas a funções como `printsensorvalues()`, o *sketch* envia para o *Serial Monitor* os valores lidos dos sensores, bem como informações adicionais como o tempo decorrido (`printelapsedtime()`) e uma máscara de erros (*mask*) que indica potenciais problemas na leitura de algum sensor. Um exemplo desta saída pode ser observado na Figura 22, que mostra os valores dos sensores formatados e a máscara de erros no ecrã do *Serial Monitor*.

Experiencia 1 – Teste qualidade de ar

Para validar o funcionamento do sensor de qualidade do ar (SHINYEI PPD42NS) e demonstrar a capacidade de monitorização do sistema, realizou-se uma experiência numa divisão fechada. Inicialmente, a divisão apresentava um índice de qualidade do ar relativamente baixo, como pode ser observado na Figura 21, entre o tempo 0 e aproximadamente 1000 segundos. Por volta dos 90 segundos, foi introduzido fumo na divisão através da queima de um cigarro. O gráfico mostra claramente a reação do sensor a este evento, com o índice de qualidade do ar a aumentar acentuadamente, refletindo a presença de partículas no ar. A qualidade do ar manteve-se num nível elevado (índice alto) durante algum tempo.

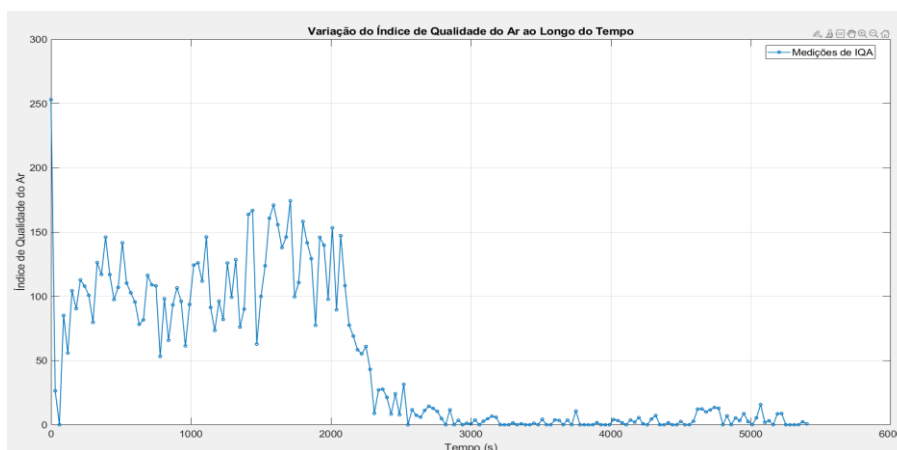


Figura 21 - Resultados Índice qualidade do Ar em experiencia1

Aproximadamente aos 2000 segundos, a janela da divisão foi aberta para promover a ventilação. Conforme esperado, o gráfico regista uma descida rápida e significativa do índice de qualidade do ar a partir deste ponto, indicando que a ventilação removeu eficientemente as partículas de fumo, e a qualidade do ar na divisão melhorou, regressando a níveis mais baixos (índice baixo). Esta experiência, visualizada na Figura 21, valida a capacidade do sistema em detetar e monitorizar alterações na qualidade do ar em tempo real.

Experiência 2 – Analise Níveis do dia a dia

Com o sistema em funcionamento, procedemos à monitorização contínua dos diversos parâmetros ambientais no dia a dia. A estação meteorológica, instalada provisoriamente na área dos Olivais, Encarnação, em Lisboa, realizou leituras regulares dos sensores conectados à placa *Seeeduino LoRaWAN*. Estas leituras eram visíveis localmente através do *Serial Monitor* do ambiente *Arduino IDE*, conforme o formato exemplificado na Figura 22, que apresenta os valores de Temperatura, Humidade, Pressão Atmosférica, estado de Chuva, Qualidade do Ar, Intensidade de Luz e radiação ultravioleta.

Ao comparar os valores registados no *Serial Monitor* (Fig. 22), obtidos diretamente da placa na localização dos Olivais/Encarnação, e apresentados na página web do *DataCake*, com os dados apresentados na página da internet de meteorologia observámos variações nos dados reportados. Estas diferenças nos valores das leituras justificam-se principalmente por dois fatores:

Primeiro, a diferença de localização geográfica entre a placa e a leitura de dados da página. As condições ambientais como temperatura, humidade, pressão atmosférica e intensidade de luz variam naturalmente entre locais distintos aumentando consideravelmente consoante distância. Portanto, era expectável que as medições instantâneas ou médias apresentadas em cada ponto de monitorização fossem diferentes.

Segundo, as margens de erro intrínsecas aos próprios sensores. Como detalhado no Capítulo 4, cada sensor possui uma precisão especificada e uma margem de erro inerente à sua tecnologia e calibração. Estas pequenas imprecisões acumuladas ou pontuais podem contribuir para variações observáveis entre as leituras brutas no dispositivo e os valores apresentados na plataforma online, mesmo que transmitissem dados da mesma localização.

Apesar destas diferenças naturais e esperadas, a experiência de monitorização diária confirmou a robustez do sistema na recolha de dados ambientais.

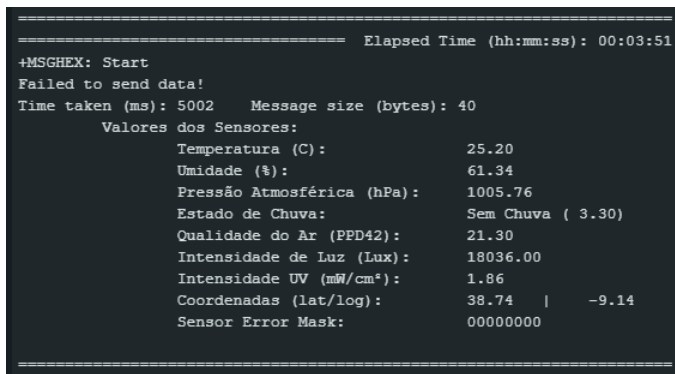


Figura 22 - Leitura 1

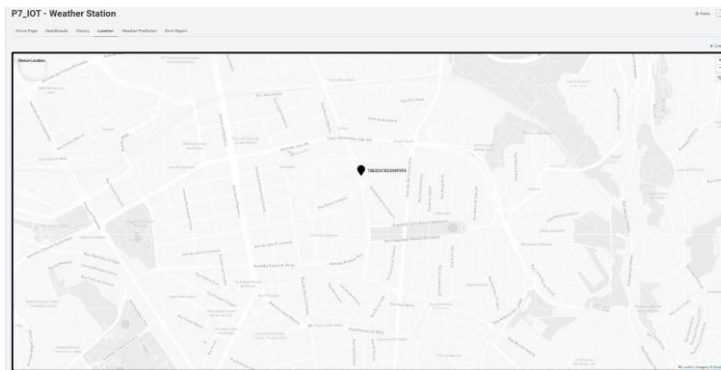


Figura 23 - Localização Leitura 1

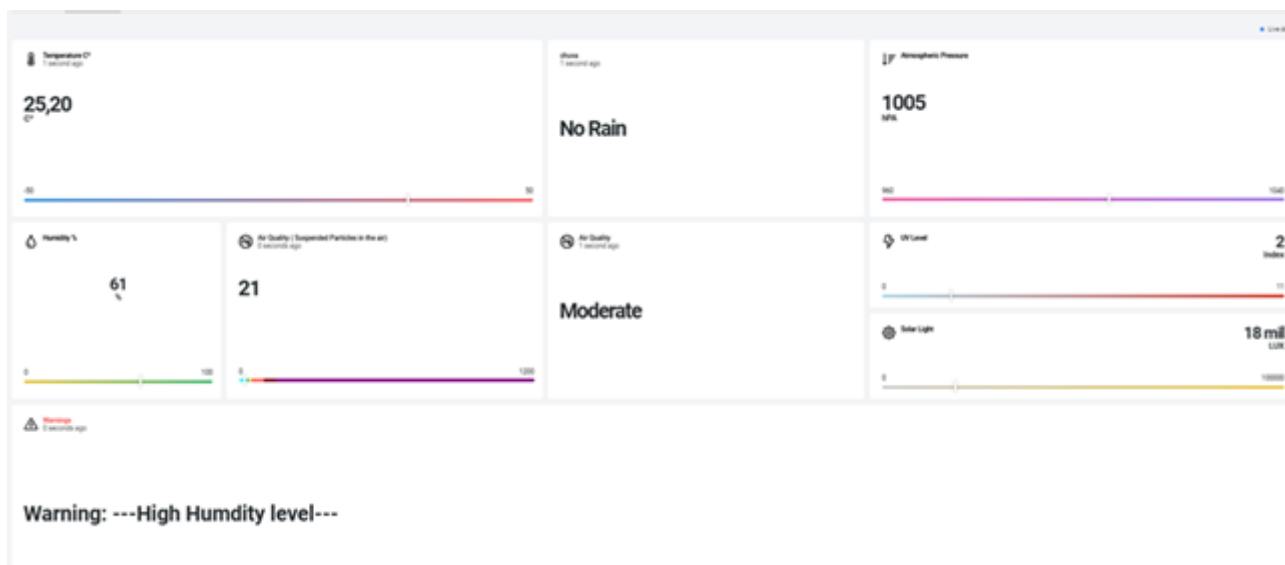


Figura 24 - Dashboards de leitura 1

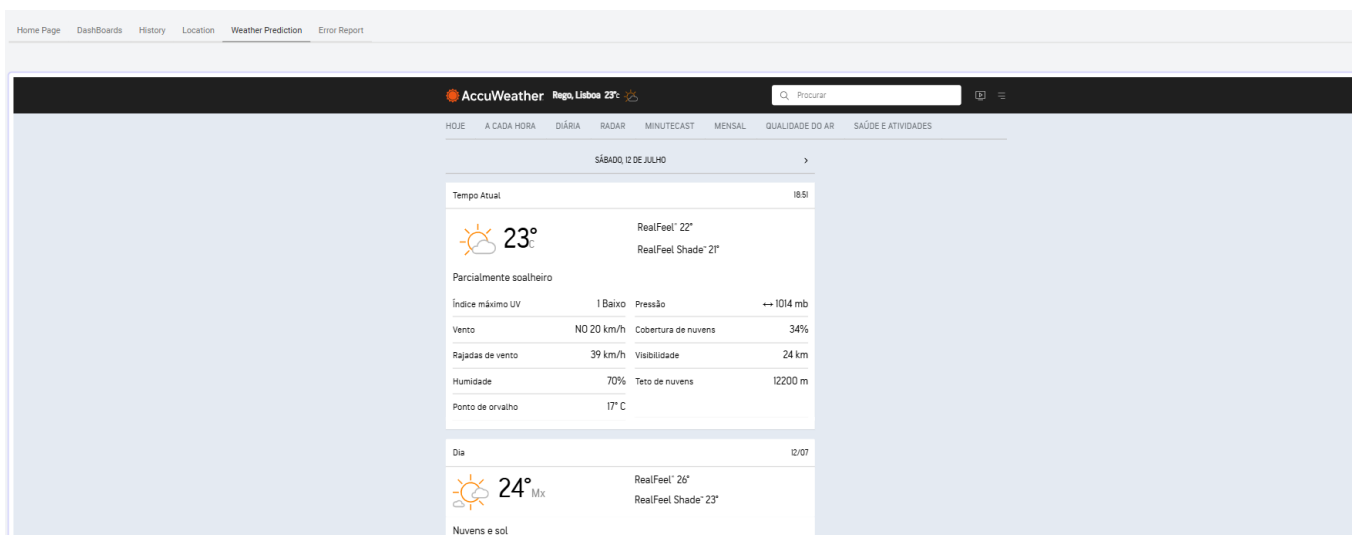


Figura 25 - Weather Prediction leitura 1

5. Plataforma The Things Network

A plataforma *The Things Network (TTN)* é uma infraestrutura global de rede colaborativa que utiliza a tecnologia *LoRaWAN* para conectar dispositivos *IoT*.

O seu propósito principal é permitir a comunicação de dispositivos de baixo consumo energético e longo alcance, promovendo a troca de dados de forma eficiente e acessível, sem a necessidade de redes privadas ou custos elevados.

No âmbito da Internet das Coisas, a utilização da *TTN* assume um papel fundamental por permitir a construção de um ecossistema aberto e descentralizado, que facilita o desenvolvimento de projetos públicos, privados ou semiprivados de forma acessível e colaborativa.

Permite-nos realizar serviços como a gestão de dispositivos, transmissão de dados em tempo real, integração com outras plataformas (como bases de dados ou sistemas analíticos) e ferramentas para monitorização da rede. Além disso, por ser baseada numa comunidade global, a *TTN* incentiva a colaboração entre utilizadores e desenvolvedores. No nosso projeto, a *TTN* permite que os dados recolhidos pela placa *Seeeduno LoRaWAN* sejam transmitidos de forma eficiente e integrados numa plataforma acessível para análise e monitorização. Isso simplifica o processo de desenvolvimento e garante que os dados estejam disponíveis para validação e tomada de decisões. Dessa forma, a *TTN* não apenas viabiliza o funcionamento do sistema, mas também contribui para a sua escalabilidade e impacto. As imagens abaixo apresentam o decoder (Fig.26) que descripta os dados recebidos, vindos da consola, durante uma troca de mensagem, com os valores dos sensores, bem-sucedida (Fig. 26).

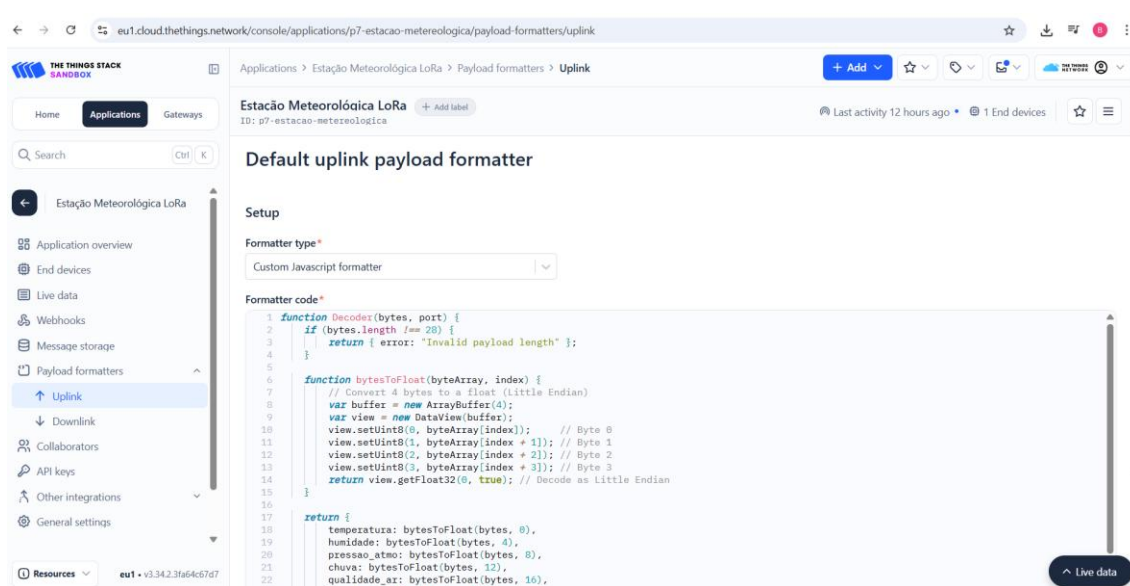


Figura 26 - Payload Decoder TTN

```

5      {
6        "device_ids": {
7          "device_id": "seeed-23",
8          "application_ids": {
9            "application_id": "p7-estacao-metereologica"
10          },
11          "dev_eui": "70B3D57ED006F0F8",
12          "join_eui": "8CF9572000079A3E",
13          "dev_addr": "260B9551"
14        }
15      }
16    ],
17    "data": {
18      "@type": "type.googleapis.com/ttn.lorawan.v3.ApplicationUp",
19      "end_device_ids": {
20        "device_id": "seeed-23",
21        "application_ids": {
22          "application_id": "p7-estacao-metereologica"
23        },
24        "dev_eui": "70B3D57ED006F0F8",
25        "join_eui": "8CF9572000079A3E",
26        "dev_addr": "260B9551"
27      },
28      "correlation_ids": [
29        "gs:uplink:01JZZSCQ8HFK0RX0J0CY2TGXBA"
30      ],
31      "received_at": "2025-07-12T17:02:13.980969441Z",
32      "uplink_message": {
33        "session_key_id": "AZf/lcHW+LvTjT5JUzc0/Q==",
34        "f_port": 8,
35        "frm_payload": "
36          +OnWQS59VkJnXMRHjDHGPixF8jwAgOJDkRnfPgAAAAAAAAAAAAAAAA====",
37        "decoded_payload": {
38          "chuva": 0.3870967626571655,
39          "humidade": 53.62224578857422,
40          "luz": 0.43574193120002747,
41          "pressao_atmo": 100536.8046875,
42          "qualidade_ar": 0.029573999345302582,
43          "temperatura": 26.864242553710938,
44          "uv": 453
45        },
46        "rx_metadata": [
47          {

```

Estação Meteorológica LoRa

+ Add Label

ID: p7-estacao-metereologica

⌕ Last activity 2 minutes ago

📶 1 End devices

☆

☰

TIME	ENTITY ID	TYPE	DATA PREVIEW			
↑ 18:02:13	seeed-23	Forward uplink data message	DevAddr: 26 0B 95 51	Payload: { chuva: 0.3870967626571655, humidade: 53.62224578857422, luz: 0.43574193120002747, pressao_atmo: 100536.8046875, qualidade_ar: 0.029573999345302582, temperatura: 26.864242553710938, uv: 453 }		
↑ 18:01:35	seeed-23	Forward join-accept message	DevAddr: 26 0B 95 51	JoinEUI: 8C F9 57 20 00 07 9A 3E	DevEUI: 70 B3 D5 7E D0 06 F0 F8	
↑ 18:01:34	seeed-23	Successfully processed join-request	DevAddr: 26 0B 69 10	JoinEUI: 8C F9 57 20 00 07 9A 3E	DevEUI: 70 B3 D5 7E D0 06 F0 F8	
⌵ 18:01:34	seeed-23	Accept join-request	DevAddr: 26 0B 95 51	JoinEUI: 8C F9 57 20 00 07 9A 3E	DevEUI: 70 B3 D5 7E D0 06 F0 F8	

Verbose stream

⌵

📄 Export as JSON

00

🗑 Clear

Figura 27 - Troca de mensagens bem sucedida

6. DataCake

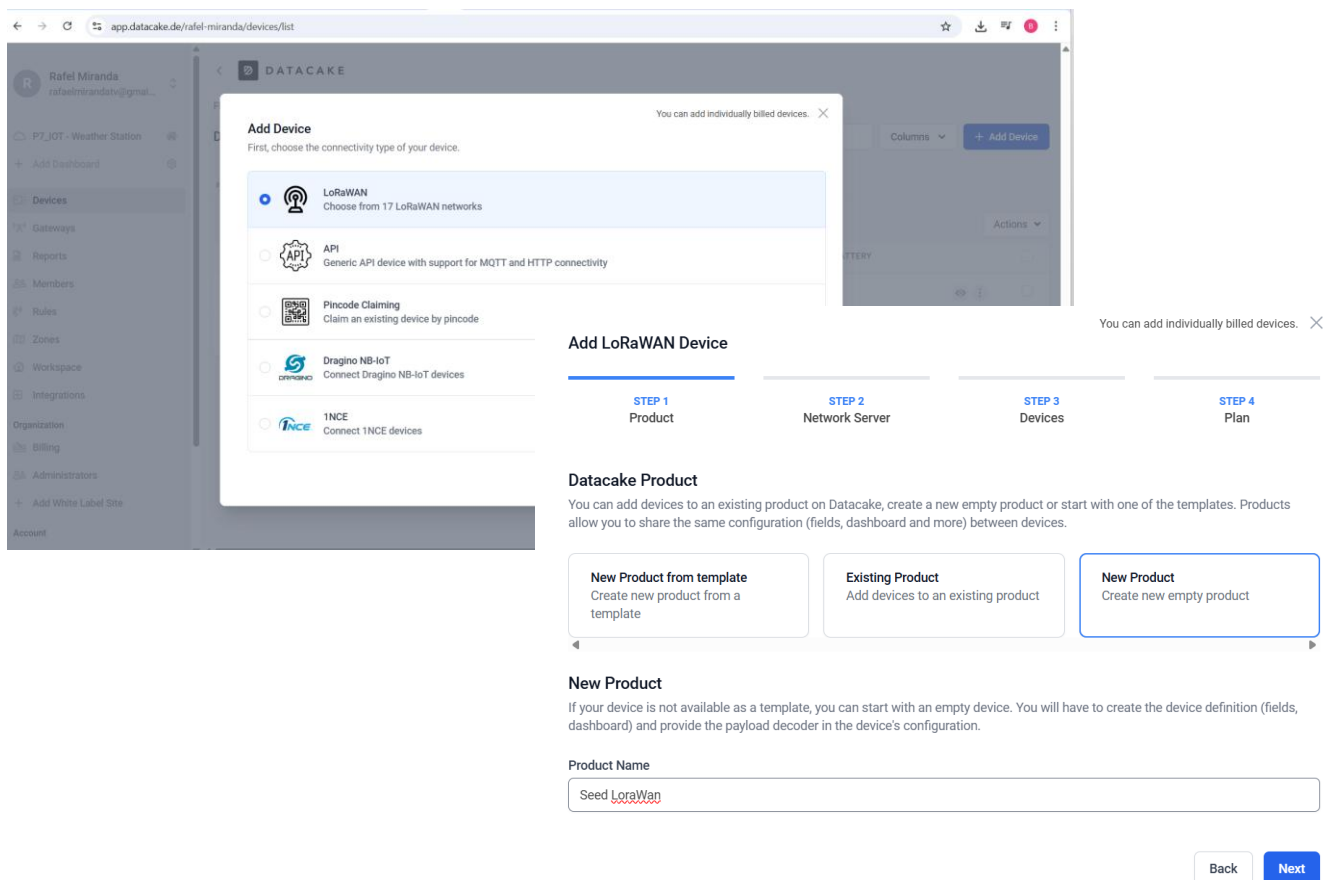


Figura 28 - Integração TTN e DataCake

Para permitir a visualização e análise intuitiva dos dados recolhidos pelo sistema de monitorização ambiental desenvolvido, optou-se pela utilização da plataforma *DataCake*. Esta escolha baseou-se primariamente na sua capacidade de integração direta com a rede *TTN* e na facilidade de criação de painéis interativos (*Dashboards*) personalizados, tornando a consulta da informação acessível e apelativa ao utilizador final.

O fluxo de dados estabelecido no projeto é iniciado pela recolha de informações ambientais pelos sensores conectados à placa microcontrolador. Estes dados são então processados pela placa, que funciona como um nó *LoRaWAN*. A placa transmite os dados, encapsulados em mensagens *uplink*, para a rede *TTN* (especificamente para a aplicação do nosso projeto associado ao nosso dispositivo). Por sua vez, a integração entre a *TTN* e o *DataCake* é um ponto crucial desta arquitetura, implementada através de mecanismos de payload forwarding. Para integrarmos o *DataCake* com o *TTN*, bastou seguir uma série de passos muito simples, o que tornou a integração fácil e prática. Em primeiro lugar, foi necessário adicionar um novo

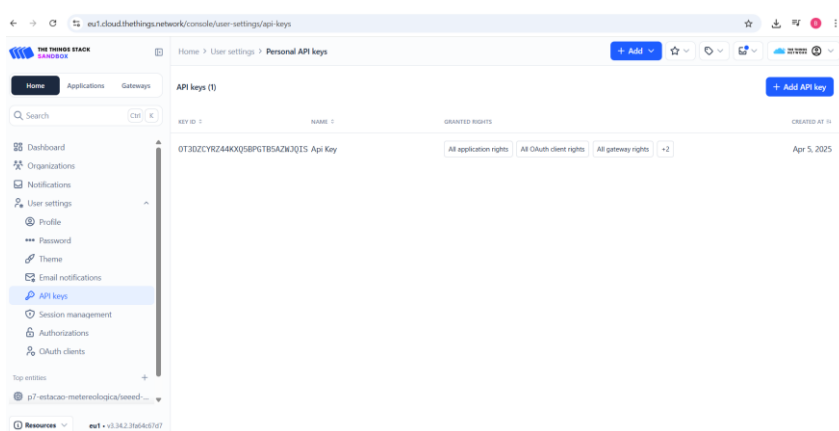
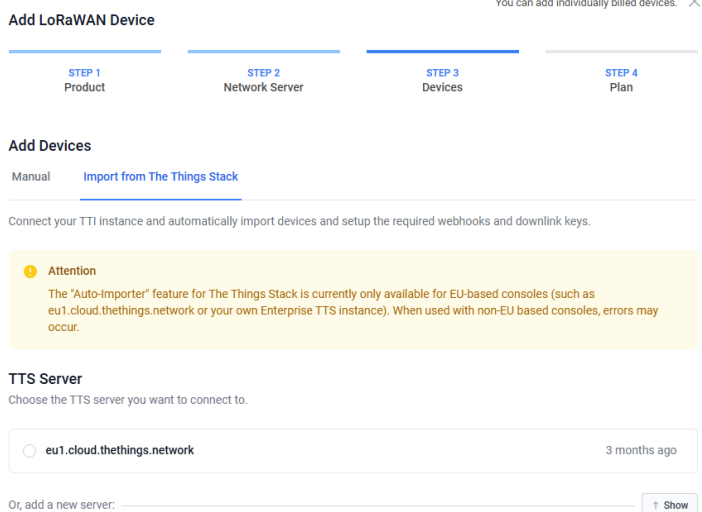
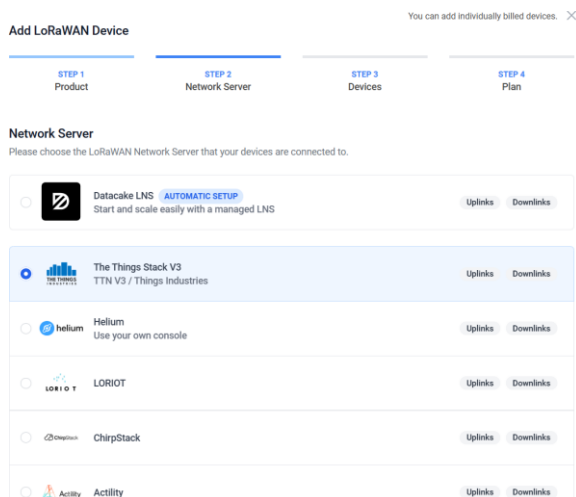


Figura 29 - Continuação do processo de integração

dispositivo no *DataCake*, identificando o seu tipo de conectividade (*LoRaWAN*) e especificando a rede *LoRa* à qual o dispositivo estava ligado (*TTN*).

Indicámos também o nome do dispositivo e definimos com detalhe qual o servidor *TTN* ao qual pretendíamos conectar o *DataCake*. De seguida, fornecemos ao *DataCake* a nossa chave *Application Programming Interface (API)* do *TTN*.

Por fim, seleccionámos, entre os dispositivos e aplicações que já havíamos criado no *TTN*, aqueles que queríamos importar para o *DataCake*.

Este processo encontra-se ilustrado nas Figura 28 e 29.

Payload Decoder

Product-wide setting

When your device sends data, the payload will be passed to the payload decoder, alongside the event's name. The payload decoder then transforms it to measurements.

```
2 function Decoder(bytes, port) {
3   /* if (bytes.length !== 36) {
4     return { error: "Invalid payload length" };
5   } */
6
7   function bytesToFloat(b0, b1, b2, b3) {
8     var bits = (b3 << 24) | (b2 << 16) | (b1 << 8) | b0;
9     var sign = (bits >>> 31) ? -1 : 1;
10    var exponent = ((bits >>> 23) & 0xFF) - 127;
11    var mantissa = bits & 0x7FFFFFFF;
12
13    if (exponent === 128) {
14      return mantissa !== 0 ? NaN : sign * Infinity;
15    } else if (exponent === -127) {
16      return sign * mantissa * Math.pow(2, -149);
17    }
18
19    return sign * (1 + mantissa / Math.pow(2, 23)) * Math.pow(2, exponent);
20  }
21
22
23  function getStatus(value) {
24    if (value >= 0 && value <= 5) {
25      return "Good";
26    } else if (value > 6 && value <= 15) {
27      return "Fair";
28    } else if (value > 16 && value <= 50) {
29      return "Moderate";
30    } else if (value > 51 && value <= 90) {
31      return "Poor";
32    } else if (value > 91 && value <= 140) {
33      return "Very poor";
34    } else if (value > 140) {
35      return "Extremely poor";
36    } else {
37      return "Invalid value";
38    }
39  }
40}
```

Payload

Port

Figura 30 - Payload Decoder DataCake

Ao receber o *payload* bruto da TTN, o *DataCake* necessita de o interpretar. Para tal, é configurado um *Payload Decoder*(Figura 30) como o da figura acima (em anexo) dentro da plataforma *DataCake* (alternativamente, poderia ser configurado na TTN e os dados já decodificados enviados). Este decoder, escrito em código (*JavaScript*), converte os *bytes* brutos do *payload* em unidades de medida legíveis e estruturadas. Estes valores decodificados são então automaticamente mapeados em campos (*Fields*) ou medidas (*Measures*) definidos para o dispositivo no *DataCake*, permitindo que a plataforma os armazene em base de dados e os disponibilize para visualização. Um dos pontos fortes que reforçou a escolha do *DataCake* foi a sua funcionalidade de partilha pública de *dashboards*, que permite o acesso gratuito e direto aos dados em tempo real sem a necessidade de o utilizador criar uma conta na plataforma.

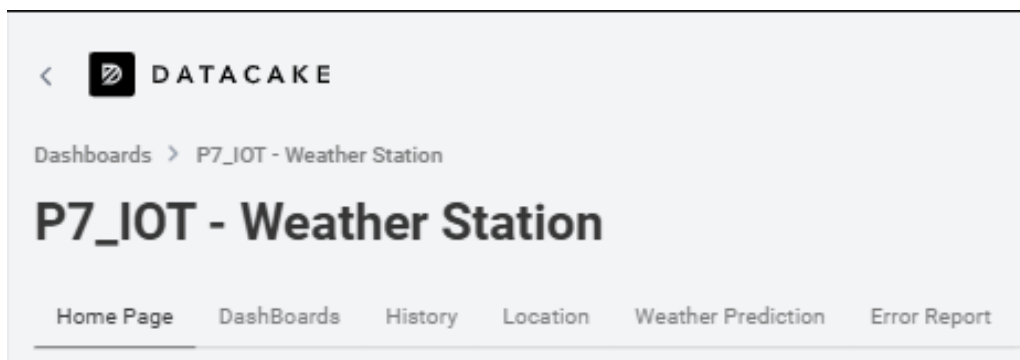


Figura 31 - Separadores Webpage

Atualmente, a interface web dedicada ao nosso dispositivo no *DataCake* encontra-se estruturada em seis abas principais (Figura 31), cada uma com uma funcionalidade específica:

- **Home Page:** Página de boas-vindas, onde é apresentado ao utilizador o nosso projeto (Figura 32).
- **Dashboards:** Contém os painéis interativos customizados, apresentando os dados recolhidos em tempo real (Figura 30).
- **History:** Permite consultar o histórico detalhado dos dados recebidos ao longo do tempo.
- **Location:** Apresenta a localização geográfica do dispositivo, utilizando os dados de *GPS* recolhidos pela placa.
- **Weather Prediction:** Página onde podemos consultar os valores das previsões meteorológicas e assim comparar com os valores recolhidos pelo nosso sistema.
- **Error report:** Regista eventuais erros ou eventos invulgares reportados pelo dispositivo ou pela comunicação.

Esta estrutura permite uma navegação clara e um acesso rápido às diferentes perspetivas dos dados monitorizados.

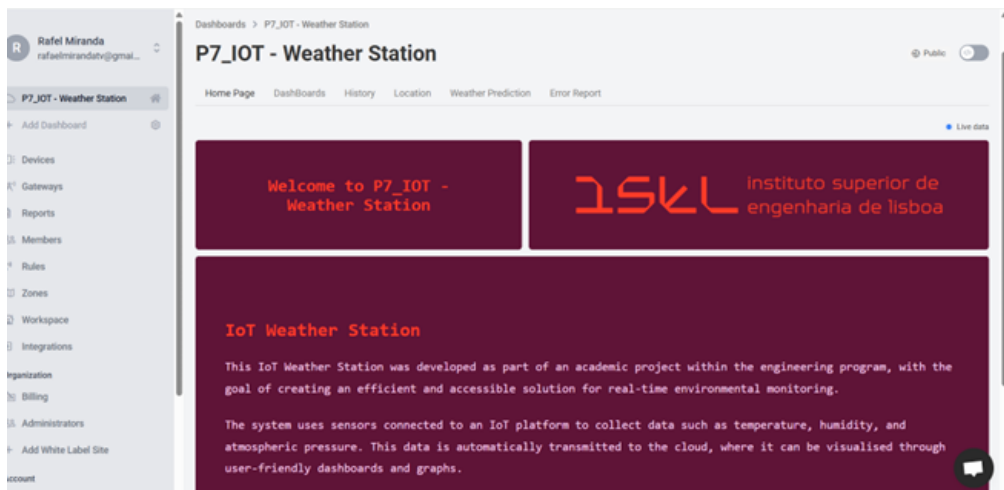


Figura 32 - HomePage

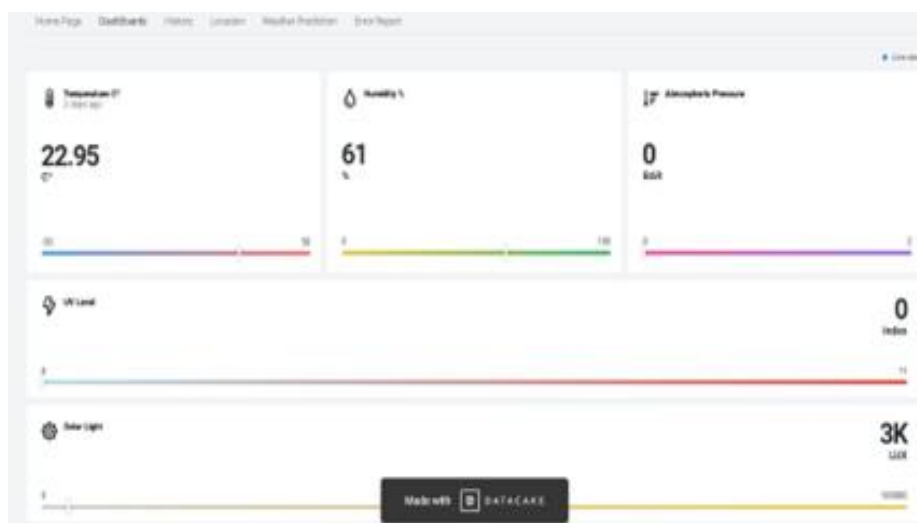


Figura 33 - Dashboards

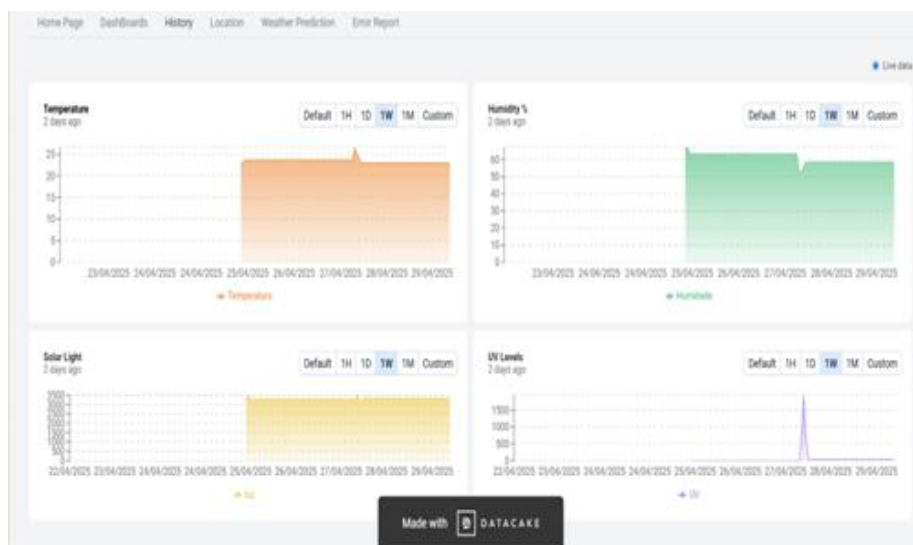


Figura 34 - History

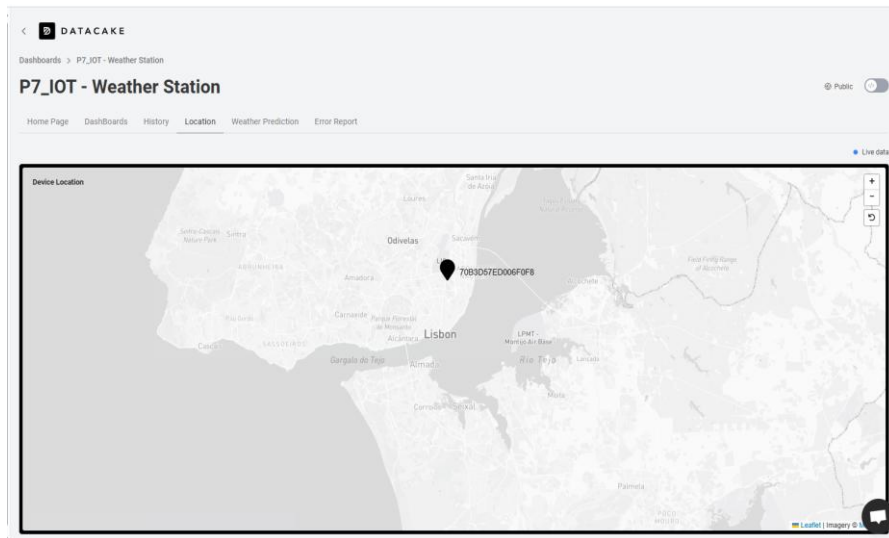


Figura 35 - Location

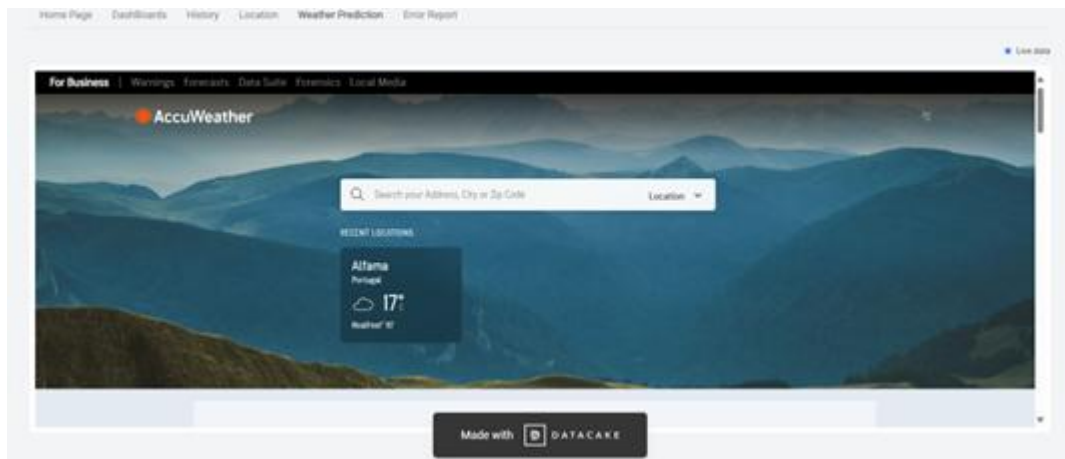


Figura 36 - Weather Prediction

7. Caixa de Suporte e proteção do sistema

Para garantir a durabilidade e o funcionamento fiável da estação meteorológica em diversos ambientes, a conceção e construção de uma caixa de suporte e proteção para os componentes eletrónicos e sensores revelou-se essencial. Esta estrutura física tem múltiplos propósitos, incluindo salvaguardar os componentes contra intempéries (chuva, humidade excessiva), poeira, impactos físicos e potenciais atos de vandalismo ou interferência. Adicionalmente, a caixa serve para organizar e fixar de forma segura todos os elementos, prevenindo danos causados por movimento ou conexões soltas.

A Figura 35 ilustra uma representação tridimensional da estrutura interna da caixa de suporte e proteção que foi concebida para este projeto através da modelagem 3D na aplicação *Sketchup3D* e posteriormente impresso numa impressora 3D. Nela, são visíveis os diferentes níveis ou áreas de montagem pensadas para acomodar a placa *Seeeduino LoRaWAN*, os diversos sensores, a bateria *LiPo*, painel solar e outros componentes de suporte ou cablagem.

O design procurou otimizar o espaço interno e facilitar a instalação e manutenção dos componentes, enquanto considera as necessidades específicas de cada sensor, permitindo a exposição adequada do sensor de chuva ou qualidade do ar, se necessário, enquanto protege a eletrónica mais sensível). A estrutura serve como base para a montagem final da estação, garantindo que todos os elementos permaneçam no seu lugar e funcionem corretamente dentro do invólucro protetor.

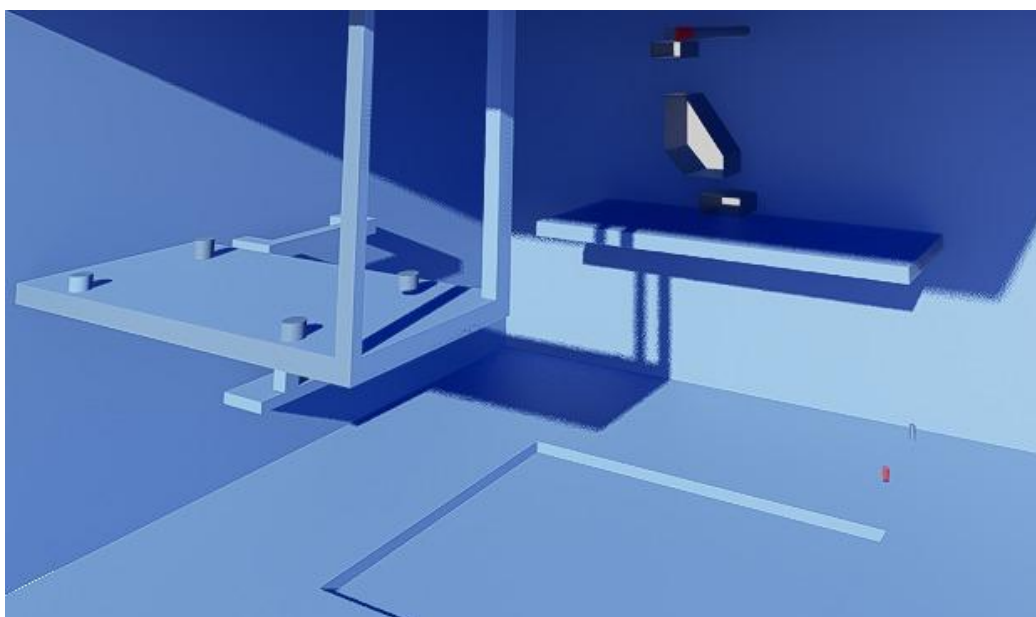


Figura 37 - Render

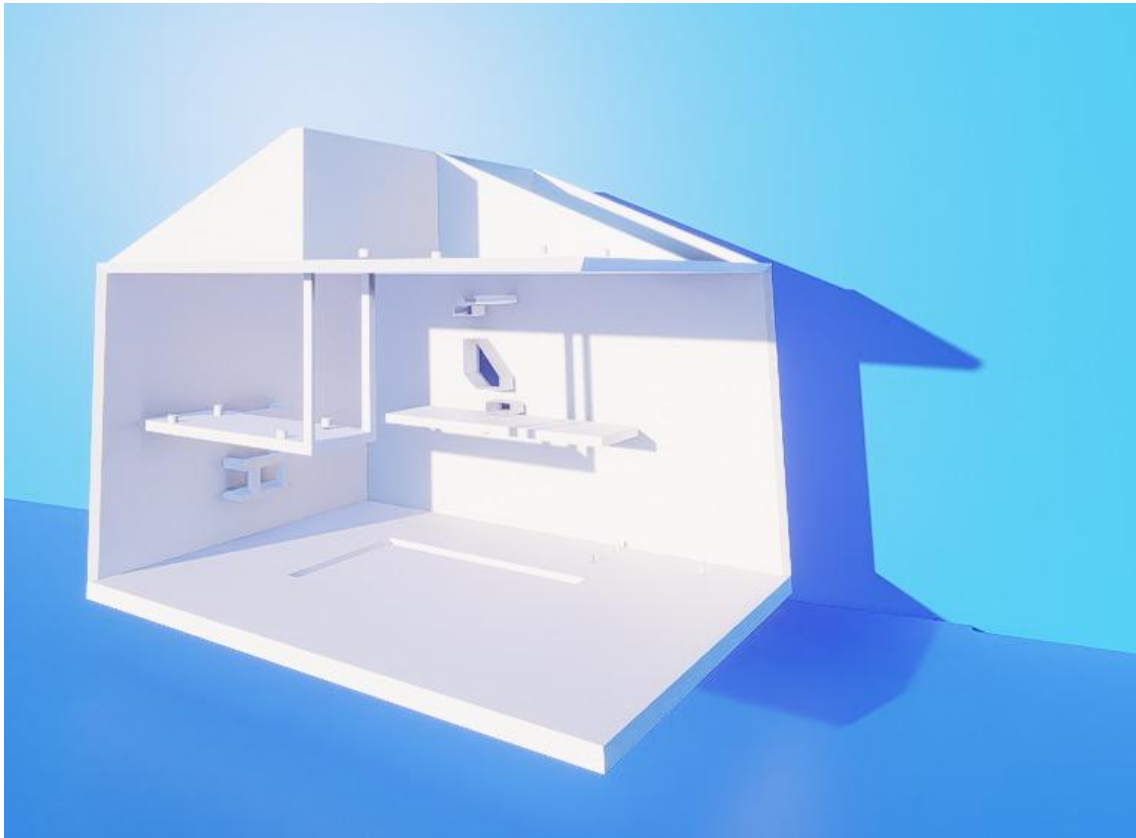
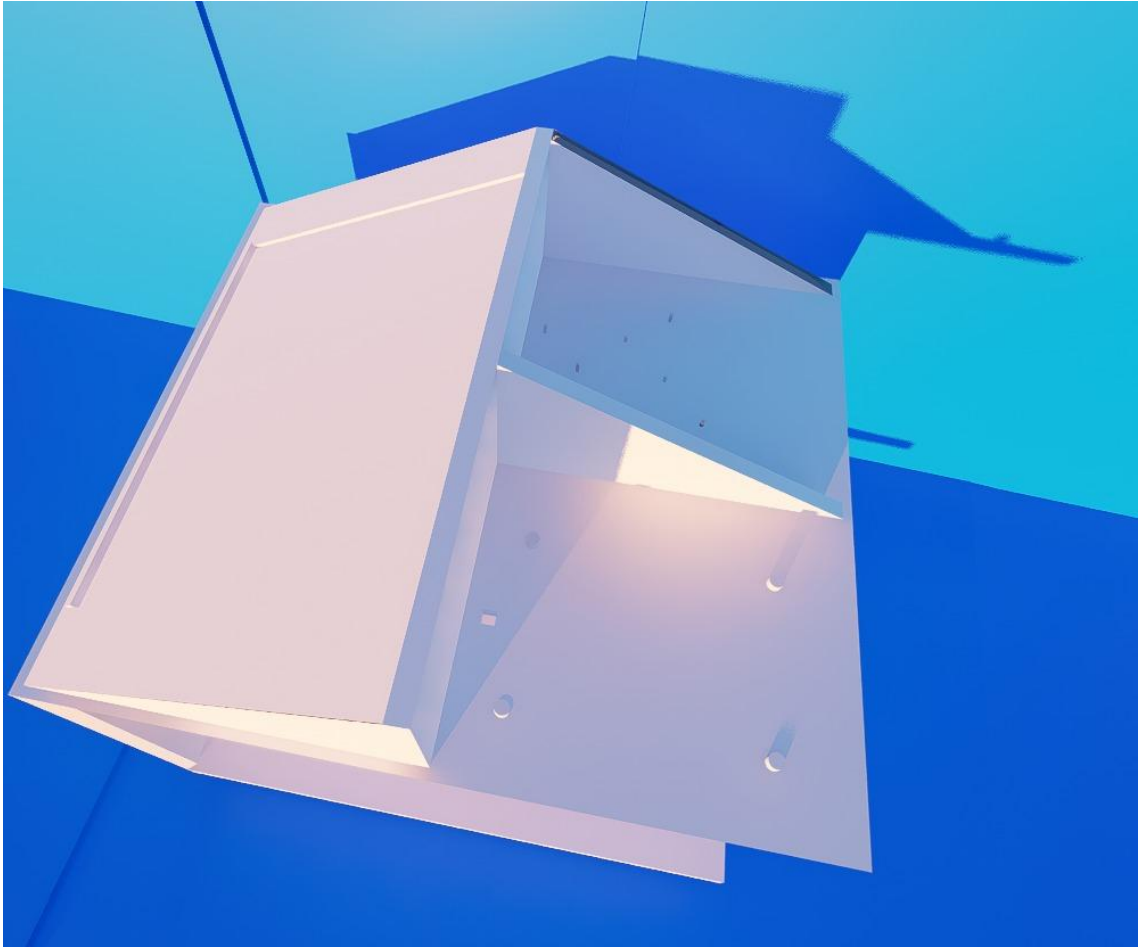


Figura 38 - Render 2

8. Conclusão

Este relatório marca a culminação da fase de desenvolvimento do projeto "Estação Meteorológica *LoRa*", representando a terceira e última etapa de entrega no âmbito da Licenciatura em Engenharia Informática, Redes e Telecomunicações (LEIRT). Durante este percurso, foi implementada e estruturada com sucesso uma versão funcional de uma estação meteorológica baseada em Arduino.

A arquitetura do sistema foi definida e materializada através da integração da placa *Seeedduino LoRaWAN* com seis sensores essenciais: temperatura e humidade (DHT20), pressão atmosférica (DPS310), qualidade do ar (SHINYEI PPD42NS), radiação ultravioleta (*Grove- UV Sensor*), luz solar (*Grove- Digital Light Sensor*) e chuva (M445C). A escolha destes componentes foi orientada pela sua relevância para a monitorização ambiental e pela sua compatibilidade com a plataforma. A comunicação dos dados recolhidos foi estabelecida com sucesso utilizando o protocolo *LoRaWAN*, através do método OTAA para a plataforma TTN. Esta abordagem demonstrou ser eficaz na autenticação automática e segura do dispositivo na rede, simplificando a gestão em larga escala.

A integração com a plataforma *DataCake* permitiu a criação de painéis interativos (*dashboards*) que proporcionam uma visualização clara e acessível das informações meteorológicas. A estrutura modular destes *dashboards* facilita a consulta de dados em tempo real e o acesso a históricos, cumprindo o objetivo de disponibilizar informações úteis aos utilizadores finais. Os resultados obtidos após análise validam a viabilidade e o potencial do projeto. A estação é capaz de monitorizar eficientemente variáveis ambientais críticas e transmiti-las através de uma rede de longo alcance e baixo consumo energético. A integração bem-sucedida com plataformas de visualização de dados assegura que as informações recolhidas sejam acessíveis e processáveis.

Embora a fase de desenvolvimento principal esteja concluída, reconhece-se a oportunidade para futuras melhorias.

A avaliação do desempenho do sistema em cenários simulados, através de ferramentas como o CupCarbon, permanece um objetivo perspicaz para aprofundar a análise da sua robustez e escalabilidade em ambientes controlados principalmente das ligações entre o sistema, *gateways* e a rede.

Em suma, o projeto demonstrou a aplicação e consolidação de conhecimentos em engenharia de sistemas, comunicação sem fios e plataformas *IoT*, resultando num protótipo funcional que cumpre os objetivos estabelecidos para esta fase final do curso.

9. Bibliografia

- [1]. Adafruit. DHT20 - AHT20 Pin Module - I2C Temperature and Humidity Sensor. Adafruit.
[<https://www.adafruit.com/product/5183>, Acedido em: Julho 2025].
- [2]. Adafruit Learning System. Connecting to a DHTxx Sensor. Adafruit.
[<https://learn.adafruit.com/dht/connecting-to-a-dhtxx-sensor>, Acedido em: Junho 2025].
- [3]. Alfacom.net Blog. LoRaWAN: Ativação de end-devices. Alfacom.net.
[<https://alfacom.net/2023/12/11/lorawan-ativacao-de-end-devices/>, Acedido em: Julho 2025].
- [4]. All About Circuits. The Basics Behind a MEMS Capacitive Humidity Sensor. All About Circuits. [<https://www.allaboutcircuits.com/technical-articles/mems-ic-capacitive-humidity-sensor-basics/>, Acedido em: Maio 2025].
- [5]. Analog Devices. Thermistor-Based Temperature Sensing System—Part 1: Design Challenges and Circuit Configuration. Analog Dialogue, Analog Devices.
[<https://www.analog.com/en/resources/analog-dialogue/articles/thermistor-temperature-sensing-system-part-1.html>, Acedido em: Junho 2025].
- [6]. Arduino. Comunicação via Wire. Documentação Arduino.
[<https://docs.arduino.cc/learn/communication/wire/>, Acedido em: Julho 2025].
- [7]. Arduino. Software Arduino IDE. Arduino Website.
[<https://www.arduino.cc/en/software>, Acedido em: Maio 2025].
- [8]. ASAIR. Data Sheet DHT20 Humidity and Temperature Module. (Sem link direto disponível).
- [9]. Acta Mechanica Slovaca. Dust Mass Concentration Measurement Based on Light Scattering Principle. Acta Mechanica Slovaca, 2013.
[<https://www.actamechanica.sk/pdfs/ams/2013/03/01.pdf>, Acedido em: Julho 2025].
- [10]. Chris Nafis. Air Quality Monitoring. howmuchsnow.com.
[<http://www.howmuchsnow.com/arduino/airquality/>, Acedido em: Junho 2025].
- [11]. DataCake. WebPage DataCake. Plataforma DataCake.
[<https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>, Acedido em: Julho 2025].
- [12]. Davide Gironi. An optical dust meter that uses. Blogspot.
[<http://davidegironi.blogspot.gr/2013/08/an-optical-dust-meter-that-uses.html>, Acedido em: Maio 2025].
- [13]. DSPMU Ranchi. Thermistors. DSPMU Ranchi (PDF).
[<https://dspmuranchi.ac.in/pdf/Blog/thermistor.pdf>, Acedido em: Julho 2025].
- [14]. EPJ Web of Conferences. Light Scattering of optical Components and their Imperfections: Measurement, Modelling, and System Analysis. EPJ Web of Conferences, 2024.
[https://www.epj-conferences.org/articles/epjconf/pdf/2024/19/epjconf_eosam2024_03015.pdf, Acedido em: Junho 2025].
- [15]. GitHub (Infineon). DPS310-Pressure-Sensor/README.md. GitHub.
[<https://github.com/Infineon/DPS310-Pressure-Sensor/blob/dps310/README.md>, Acedido em: Julho 2025].
- [16]. GitHub (RobTillaart). Arduino library for DHT20 I2C temperature and humidity sensor. GitHub.
[<https://github.com/RobTillaart/DHT20>, Acedido em: Maio 2025].

- [17]. Han X., Naeher L.P. (2006). A review of traffic-related air pollution exposure assessment studies in the developing world. *Environment International*, Vol. 32, pp. 106 – 120.
- [18]. Health Effects Institute. (2010). Traffic-Related Air Pollution: a Critical Review of the Literature on Emissions, Exposure, and Health Effects Special Report 17. Health Effects Institute.
[<http://pubs.healtheffects.org/view.php?id=334>, Acedido em: Junho 2025].
- [19]. Infineon. DPS310 Digital XENSIV™ Barometric Pressure Sensor for Portable Devices. (Datasheet - Sem link direto disponível).
- [20]. Instructables. Air Quality Test Box. Instructables,
[<https://www.instructables.com/id/Air-Quality-Test-Box/>, Acedido em: Julho 2025].
- [21]. Interline.pl. What Technology Behind LoRa Frequency. Interline.pl.
[<https://interline.pl/Information-and-Tips/What-Technology-Behind-LoRa-Frequency>, Acedido em: Maio 2025].
- [22]. Jiao, W., Liu, Y., Yuan, J., Pope III, C.A., Zhang, J. (2019). Low-cost PM2.5 Sensors: An Assessment of their Suitability for Various Applications. *Aerosol and Air Quality Research*, Vol. 19, No. 8, pp. 1837-1848.
[<https://aaqr.org/articles/aaqr-18-10-lcs-0390.pdf>, Acedido em: Junho 2025].
- [23]. Kumar, R., Pandey, R. (2019). A review of principles of MEMS pressure sensing with its aerospace applications. *International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)*, Vol. 9, Issue 4, pp. 593-602. [Disponível via ResearchGate: https://www.researchgate.net/publication/334399274_A_review_of_principles_of_MEMS_pressure_sensing_with_its_aerospace_applications, Acedido em: Julho 2025].
- [24]. Li, J., Wang, H., Yang, C., Ding, Y., Chen, J., Lv, Z., Zhang, L. (2024). Evaluation of Long-Term Performance of Six PM2.5 Sensor Types. *Atmosphere*, 15(4), 423.
[<https://pmc.ncbi.nlm.nih.gov/articles/PMC11861664/>, Acedido em: Maio 2025].
- [25]. LoRa Alliance®. LoRa Alliance®. LoRa Alliance Website.
[<https://lora-alliance.org/>, Acedido em: Junho 2025].
- [26]. lora.readthedocs.io. Documentação LoRa. Read the Docs.
[<https://lora.readthedocs.io/en/latest/>, Acedido em: Julho 2025].
- [27]. Newton C. Braga. Modulação LoRa. Newton C. Braga Website.
[<https://www.newtonbraga.com.br/index.php/telecomunicacoes/16353-modulacao-lora-tel263.html>, Acedido em: Maio 2025].
- [28]. Ozdisan. Grove - High Precision Barometric Pressure Sensor (DPS310). Ozdisan (PDF).
[https://cdn.ozdisan.com/ETicaret_Dosya/650333_502618.pdf, Acedido em: Julho 2025].
- [29]. Beginning LoRa Radio Networks with Arduino de Pradeeka Seneviratne.
- [30]. ROTRONIC. The Capacitive Humidity Sensor – How it Works & Attributes of the Uncertainty Budget. ROTRONIC (PDF).
[https://www.rotronic.com/media/productattachments/files/c/a/capacitive_humidity_sensor_final.pdf, Acedido em: Junho 2025].
- [31]. Seeed Studio. DHT20 Product Manual - Temperature and humidity sensor. Seeed Studio (PDF Manual).
[<https://files.seeedstudio.com/products/101020932/DHT20-Product-Manual.pdf>, Acedido em: Julho 2025].
- [32]. Seeed Studio Wiki. Seeeduino LoRAWAN. Seeed Studio.
[https://wiki.seeedstudio.com/Seeeduino_LoRAWAN/, Acedido em: Maio 2025].

- [33]. Seinfeld H. John, Pandis N. Spyros. (2006). Atmospheric chemistry and physics, From air pollution to climate change. Segunda edição. John Wiley & Sons, INC.
- [34]. Semtech. What is LoRa. Semtech Website.
[<https://www.semtech.com/lora/what-is-lora>, Acedido em: Junho 2025].
- [35]. Senraco. What are the distinctions between OTA (Over-The-Air Activation) and ABP (Activation By Personalization) in LoRaWAN. Senraco Support Portal.
[<https://support.senraco.com/portal/en/kb/articles/what-are-the-distinctions-between-ota-over-the-air-activation-and-abp-activation-by-personalization-in-lorawan>, Acedido em: Julho 2025].
- [36]. SensorsONE. How does the accuracy of pressure measurement devices change over time. SensorsONE Blog.
[<https://www.sensorsone.com/accuracy-change-over-time-pressure-measurement-devices/>, Acedido em: Maio 2025].
- [37]. SHINYEI TECHNOLOGY CO.,LTD. PRODUCT SPECIFICATIONS PPD42NJ. (Datasheet).
- [38]. SHINYEI Technology. Particle Sensor Unit. SHINYEI Website.
[<https://www.shinyei.co.jp/stc/eng/products/optical/dust.html>, Acedido em: Junho 2025].
- [39]. SHINYEI Technology. PPD42NJ Particle Sensor Unit. SHINYEI Website.
[<https://www.shinyei.co.jp/stc/eng/products/optical/ppd42nj.html>, Acedido em: Julho 2025].
- [40]. Sousan, S., Majeed, M.A., Stauffer, T.V., Farr, N., Keegan, P., Cavallari, J.M., Leaderer, B.P., Holford, T.R., Fine, P.M., Slade, M.D. (2020). A Review of Low-Cost Particulate Matter Sensors from the Developers' Perspectives. Sensors 2020, 20(23), 6819.
[<https://pmc.ncbi.nlm.nih.gov/articles/PMC7730878/>, Acedido em: Maio 2025].
- [41]. System Plus Consulting - Yole Group. Infineon DPS310 pressure sensor. System Plus Consulting - Yole Group (Report Sample PDF).
[https://medias.yolegroup.com/uploads/2019/02/Yole_SP17308-Infineon_DPS310_Pressure_sensor_Sample.pdf, Acedido em: Julho 2025].
- [42]. The Things Industries. ABP vs OTAA. The Things Industries Documentation.
[<https://www.thethingsindustries.com/docs/hardware/devices/concepts/abp-vs-otaa/>, Acedido em: Junho 2025].
- [43]. The Things Network. Community Lisbon. The Things Network Website.
[<https://www.thethingsnetwork.org/community/lisbon/>, Acedido em: Julho 2025].
- [44]. Texas Instruments. Why long-term consistent performance matters for relative humidity sensors. Texas Instruments (Literature).
[<https://www.ti.com/lit/SSZT152>, Acedido em: Maio 2025].
- [45]. Tracy Allen. (2013). De-construction of the Shinyei PPD42NS dust sensor. EME Systems LLC (PDF).
[http://takingspace.org/wp-content/uploads/ShinyeiPPD42NS_Deconstruction_TracyAllen.pdf, Acedido em: Junho 2025].
- [46]. WFSensors. Differences in Measurement Accuracy of Different Types of Air. WFSensors Blog. [<https://www.wfsensors.com/blog/differences-in-measurement-accuracy-of-different-types-of-air-pressure-sensors.html/#:~:text=The%20accuracy%20of%20barometric%20pressure,turn%20affects%20the%20measured%20value.>, Acedido em: Julho 2025].

- [47]. World Health Organization. (2010). Air quality and health. Fact sheet No 313, updated September 2011. WHO. [<http://www.who.int/mediacentre/factsheets/fs313/en/>, Acedido em: Junho 2025].
- [48]. YouTube. (Sem título específico fornecido). [<https://www.youtube.com/watch?v=CAvawEcxoPU>, Acedido em: Maio 2025].
- [49]. Zhao, T., Liu, Z., Chen, W., Huang, C., Wu, H., Liu, S. (2022). A Further Theoretical Study of Capacitive Pressure Sensors Based on Thin Film Elastic Deflection and Parallel Plate Capacitor: Refined Closed-Form Solution and Numerical Calibration. *Sensors* 2022, 22(8), 2848. [<https://www.mdpi.com/1424-8220/22/8/2848>, Acedido em: Julho 2025].
- [50]. 50. Zhang K., Batterman S. (2013). Air pollution and health risks due to vehicle traffic. *Science of the Total Environment*, pp: 450–451.
- [51]. [<https://www.seeedstudio.com/document/pdf/Introduction%20to%20Grove.pdf>, Acedido em Março 2025]
- [52]. [https://www.researchgate.net/publication/298344488_A_Novel_Design_and_Implementation_of_Electronic_Weather_Station_and_Weather_Data_Transmission_System_Using_GSM_Network , Acedido em Maio 2025]
- [53]. [<https://www.scielo.br/j/babt/a/9MHbV7JNswtsMN9MnmNJcCv/>, Acedido em Julho 2025]
- [54]. [<https://files.seeedstudio.com/products/101020932/DHT20-Product-Manual.pdf>, Acedido em Julho 2025]

Bibliografia de Imagens

- Figura 1 - Up-chirp e Down-chirp ao longo do tempo
Fonte: <https://link.springer.com/article/10.1007/s11277-024-10911-z>
- Figura 2 - Mensagem Lora no domínio da frequência
Fonte: <https://link.springer.com/article/10.1007/s11277-024-10911-z>
- Figura 3 - Mapa de antenas TTN disponíveis na área do ISEL
Fonte: <https://www.thethingsnetwork.org/community/lisbon/>
- Figura 4 - Distribuição de Comunidades em Portugal e Gateways de Lisboa
Fonte: <https://www.thethingsnetwork.org/community/lisbon/>
- Figura 5 - Ilustração do processo "Join"
Fonte: <https://alfacomp.net/2023/12/11/lorawan-ativacao-de-end-devices/>
- Figura 6 - Principais Componentes Arduino IDE
Fonte: Autoria própria - Aplicação Arduino IDE.
- Figura 7 - Menu Sketch e Tools Arduino IDE26
Fonte: Autoria própria - Aplicação Arduino IDE.
- Figura 8 - Componentes da Estação meteorológica
Fonte: Autoria própria - Fotografia do Sistema.
- Figura 9 - Placa Seeeduino LoRaWAN
Fonte: https://wiki.seeedstudio.com/Seeeduino_LoRAWAN/
- Figura 10 - Sensor Grove DHT20
Fonte: <https://www.seeedstudio.com/blog/2024/09/27/home-assistant-sensors/>
- Figura 11 - Modulo DHT20
Fonte: <https://news.sparkfun.com/3918>
- Figura 12 . Sensor de pressão atmosférica Grove DPS310 30
Fonte: https://wiki.seeedstudio.com/grove_barometer_sensor_spa06_003/
- Figura 13 - SHINYEI PPD42NS
Fonte: https://link.springer.com/chapter/10.1007/978-1-4842-7234-3_15
- Figura 14 - Ilustração do mecanismo de leitura de partículas
Fonte: <https://aqicn.org/sensor/shinyei/pt/>
- Figura 15 - Sensor Grove UV sensor e cabo Grove
Fonte: https://wiki.seeedstudio.com/Grove-IR_Distance_Interrupter_v1.2/
- Figura 16 - Sensor Grove Digital Light v1.1
Fonte: <https://wiki.seeedstudio.com/Grove-Analog-Microphone/>

- **Figura 17 - Ilustração da variação de resistência**
Fonte: https://lastminuteengineers.com/rain-sensor-arduino-tutorial/?utm_source=chatgpt.com
- **Figura 18 - Painel Solar 1W 5V 200mA - 110x80mm**
Fonte: <https://www.botnroll.com/en/solar-panels/314-5w-solar-panel-55x70.html>
- **Figura 19 - Variação do sinal SDA e SCL**
Fonte: <https://www.analog.com/en/resources/technical-articles/i2c-primer-what-is-i2c-part-1.html>
- **Figura 20 - Figura Exemplo**
Fonte: Autoria Própria
- **Figura 21 - Resultados Índice qualidade do Ar em experiencia1**
Fonte: Autoria Própria
- **Figura 22 - Leitura 1**
Fonte: Autoria Própria
- **Figura 23 - Dashboards de leitura 1**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
Figura 24 - Weather Prediction leitura 1
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 25 - Localização Leitura 1**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 26 - Payload Decoder TTN**
Fonte: <https://console.cloud.thethings.network/>
- **Figura 27 - Troca de mensagens bem sucedida**
Fonte: <https://console.cloud.thethings.network/>
- **Figura 28 - Integração TTN e DataCake**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 29 - Continuação do processo de integração**
Fonte: <https://console.cloud.thethings.network/>
- **Figura 30 - Payload Decoder DataCake**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>

- **Figura 31 - Separadores Webpage**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 32 – Dashboards**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 33 – HomePage**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 34 – History**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 35 - Weather Prediction**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 36 - Location**
Fonte: <https://app.datacake.de/dashboard/d/748d28bb-ca6c-42a9-9cb6-fb3538893f0c>
- **Figura 37 - Render**
Fonte: Autoria Própria
- **Figura 38 – Render 2**
Fonte: Autoria Própria