Engenharia de Sistemas e Serviços

Relatório do Segundo Trabalho Laboratorial

Solução IoT de monitorização de áreas agrícolas

**Protótipo e Serviços**



Índice

[1. Resumo 3](#_Toc514960324)

[2. Proposta 3](#_Toc514960325)

[3. Solução 4](#_Toc514960326)

[4. Arquitetura do Protótipo da solução proposta 5](#_Toc514960327)

[5. Protótipo 6](#_Toc514960328)

[5.1. Local Network (Wireless) 6](#_Toc514960329)

[5.1.1. MQTT 6](#_Toc514960330)

[5.1.2. Comunicações em área fechada (sensores) 7](#_Toc514960331)

[5.2. Serial Communication between Controller and Micro Controller 7](#_Toc514960332)

[5.2.1. Controlo de acessos (sensores) 7](#_Toc514960333)

[5.3. LoRa Network 8](#_Toc514960334)

[5.3.1. Comunicação longa distancia (sensores) 8](#_Toc514960335)

[6. Serviços 9](#_Toc514960336)

[6.1. Monitorização de áreas abertas com longas distâncias 9](#_Toc514960337)

[6.2. Monitorização em áreas fechadas (estufa) 9](#_Toc514960338)

[6.3. Controlo de acessos 10](#_Toc514960339)

[6.4. Disponibilização da informação *AnyTime*/*AnyWhere* 10](#_Toc514960340)

[6.4.1. API 10](#_Toc514960341)

[6.4.2. Base dados 11](#_Toc514960342)

[6.4.3. Website 11](#_Toc514960343)

[6.5. Alertas por e-mail 13](#_Toc514960344)

[6.6. Estado dos equipamentos 13](#_Toc514960345)

[7. Desafios/Dificuldades/Trabalho a Desenvolver/Motivação pessoal 14](#_Toc514960346)

[8. Conclusão 15](#_Toc514960347)

[9. Referências bibliográficas 16](#_Toc514960348)

[10. Anexos 17](#_Toc514960349)

# Resumo

O presente documento vem no seguimento do desenvolvimento da solução IoT apresentada no primeiro trabalho prático\* e divide-se nos seguintes tópicos:

- Referida novamente a proposta com o fim de enquadramento dos docentes na nossa solução IoT.

- Referida novamente a solução IoT composta pelos esquemas gerais da rede com diferentes níveis de arquitetura e respetivas alterações ao cenário inicial;  
 - Protótipo de alta-fidelidade;

- Referido os serviços que foram implementados no protótipo e como se enquadram na realidade;  
 - Plataforma WEB *AnyTime*/*AnyWhere*;

- Desafios, dificuldades, trabalho a desenvolver e motivação pessoal.

**\*NOTA:** O presente relatório não substitui o primeiro trabalho desenvolvido pelo que deverá ter em conta o conhecimento total sobre o mesmo.

# Proposta

As tecnologias de informação (TI) massificaram-se de tal forma nas nossas vidas que hoje em dia é imprescindível a sua utilização nas mais diversas áreas de negócio. Quando associamos a tecnologia ao conceito de automação, associamos rapidamente outro conceito de internet das coisas (IoT – *Internet of things*). Estas componentes possibilitam assim, um vasto conjunto de serviços (*AnyService*), a qualquer hora (*AnyTime*), em qualquer lugar (*AnyWhere*) e em qualquer dispositivo (*AnyDevice*).

A nossa solução proposta permite ter acesso à informação recolhida através de nós sensoriais de uma forma eficiente, pretendendo-se dinamizar as tecnologias da informação na indústria agrícola com o objetivo de introduzir o conceito de "Agricultura Inteligente". Seguindo este princípio, iremos apresentar uma solução composta por uma rede de “coisas” (sensores ambientais, presenciais, entre outros) que são monitorizados através de um microcontrolador. Posteriormente os dados recolhidos serão enviados em tempo real (*AnyTime*) para um serviço de *cloud* onde serão manipulados e apresentados numa plataforma WEB. Desta forma a plataforma permitirá consultar a informação em qualquer lugar (*AnyWhere*) com qualquer dispositivo (*AnyDevice*).

Neste sentido, os objetivos deste projeto são:

1. Estudar o funcionamento e configurações IoT 1.
2. Dinamizar arquitetura que permita autoconfigurações de equipamentos à distância.
3. Estudar a implementação de formas de comunicação entre áreas monitorizadas ao ar livre 2.
4. Criar cenários de implementação e de testes com Arduino, Raspberry Pi, ESP.
5. Realizar testes e apresentar os resultados.

A solução IoT que apresentamos está dividida em quatro partes:

*Sensing*,

*Actuation*,

Disponibilização informação (*AnyTime*/*AnyWhere*),

Autoconfiguração de equipamentos. **(proposta de trabalho não desenvolvida)**

Do ponto vista sensorial iremos analisar: temperatura, humidade, proximidade, movimentos, entre outros. Do ponto vista da atuação poderão ser despoletadas ações mediante a leitura dos nós sensoriais, assim como através de uma plataforma WEB. Os nós sensoriais fazem parte de uma rede local que estão diretamente ligados ou que possuem a capacidade de comunicar com um microcontrolador (arduino/raspberry). A disponibilização da informação será feita através de um Raspberry Pi (com acesso à internet) para um serviço na cloud. Os equipamentos que desempenham funções de gestão de dados (Arduino/Raspberry/ESP) são autoconfiguráveis numa perspetiva plug & play. Uma vez instalados poderão ser configurados sempre que necessário à distância.

# Solução

Do ponto vista solução a nossa arquitetura divide-se nos seguintes pontos:

- Três Thing Network distintas que comunicam com um Thing Controller;

- Uma rede Wireless Local;

- Uma ligação ponto a ponto entre o Thing Controller e uma Thing com sensores;

- Uma Low-Power Wide-Area network de comunicações longa distância ao ar livre e respetivo Gateway que faz a comunicação entre a rede de longas distancia e a rede local;

- Ligação à internet;

- Plataforma de acesso à informação *AnyTime/AnyWhere*

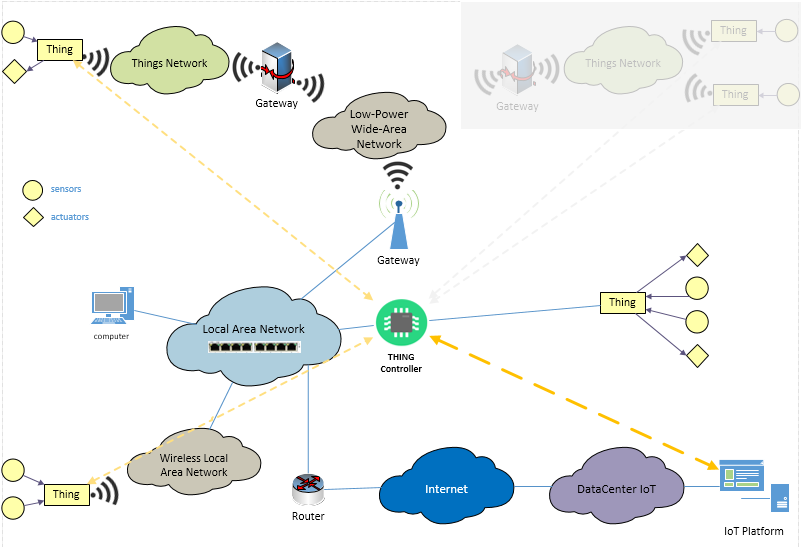


Figura 1 - Arquitetura geral de rede

A arquitetura geral de rede não sofreu uma alteração direta, mas sim uma simplificação para fins de prototipagem uma vez que não foi possível adquirir mais material para um segundo ponto de acesso Low-Power Wide-Area Network com as respetivas Things.

De salientar que o ponto-chave da arquitetura é o THING Controller que é responsável por receber os dados de todas as things network, geri-los e enviar toda essa informação para plataforma EB (*AnyTime*, *AnyWhere*). A plataforma permite também o sentido inverso de enviar ações para o THING Controller que por sua vez envia para as redes das coisas.

# Arquitetura do Protótipo da solução proposta

Do ponto vista de protótipo a nossa arquitetura divide-se nos seguintes pontos:

- Três Thing Network distintas que comunicam com um Thing Controller;

- Uma rede Wireless Local (**IEEE802.11b/g/n**);

- Uma ligação ponto a ponto (**Serial**) entre o Thing Controller (RPI3) e uma Thing (Arduino) com sensores;

- Uma Low-Power Wide-Area network e respetivo Gateway (**LoRa**);

- Plataforma de acesso à informação *AnyTime/AnyWhere* (**Laravel**)

- Ligação à internet;

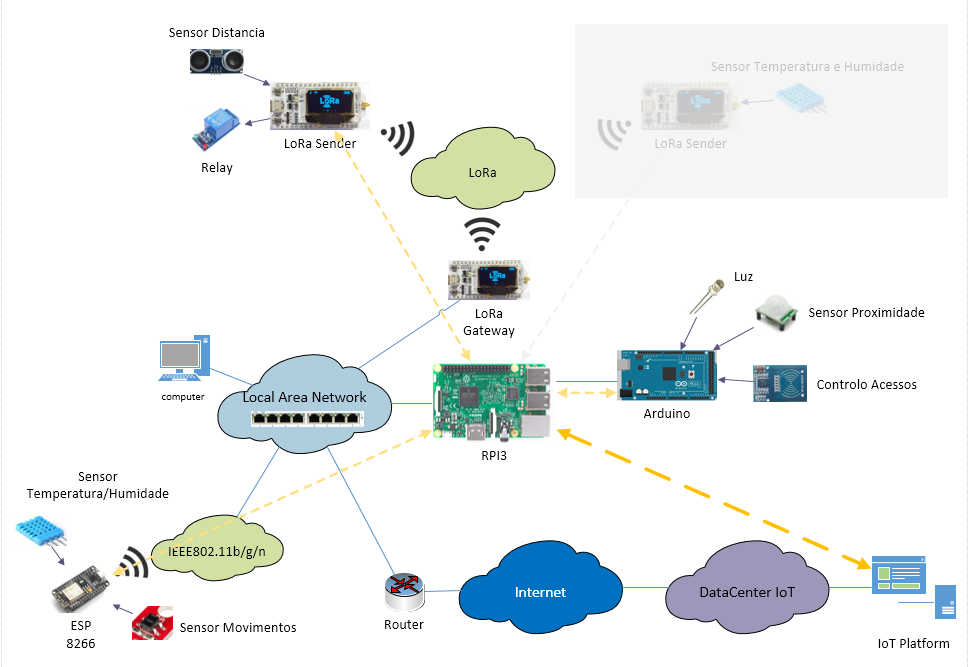


Figura 2 - Arquitetura geral de rede nível 2 (protótipo)

Face ao descrito na página anterior a arquitetura sofreu uma simplificação do protótipo por falta de material, sendo que foi removida uma das duas áreas LoRa. Para além disso um dos sensores do cenário foi trocado. O LM35 que estava associado ao ESP8266 foi substituído pelo DHT11 uma vez que já não o íamos usar na área removida e este permite para além de obter temperatura, adquirir também a humidade.

# Protótipo

Este ponto descreve todo o protótipo realizado e vamos dividi-lo em partes para que seja mais fácil a compreensão do trabalho desenvolvido.

# Local Network (Wireless)

A nossa rede local do protótipo foi criada através de recurso ao hotspot do Windows 10.

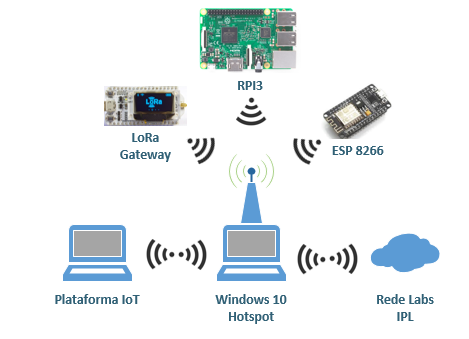
Numa breve explicação, aquilo que o hotspot nos permite fazer, é criar uma rede wifi funcionando como um router que faz forwarding de todos os pedidos para uma interface existente no laptop seja ela ethernet ou wireless. Desta forma conseguimos garantir conectividade entre todos os equipamentos, assim como à plataforma WEB, de uma forma segura, com o nosso endereçamento IPv4 e sem interferir com outros equipamentos na rede do IPL.

Figura 3 - Rede local (wireless)

# MQTT

Face à necessidade de enviar informação entre equipamentos e uma vez que estamos a lidar com things que possuem baixos recursos computacionais, optamos por usar a tecnologia MQTT para troca de mensagens. Toda a comunicação neste ponto (rede wireless) entre LoRa Gateway, RPI3 e ESP8266 é feita através de MQTT, sendo que o nosso broker é o Raspberry Pi 3. O broker é ainda responsável por enviar/receber informação da plataforma com recurso a uma Restful API.

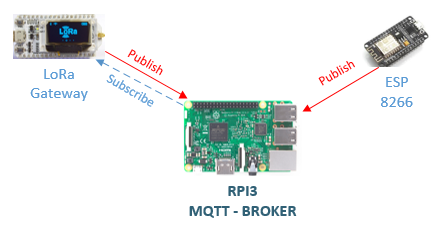


Figura 4 - MQTT

# Comunicações em área fechada (sensores)

O ESP8266 que se encontra diretamente ligado à rede local é responsável por fazer o controlo sensorial da estufa através dos seguintes sensores:

**DHT11** – controlo de temperatura e humidade na estufa;

**QRE** – deteção de movimentos/pragas (exemplo: roedores)

Os dados lidos são enviados pelo ESP8266 via MQTT para o Raspberry Pi 3 que posteriormente os envia com recurso para a plataforma.

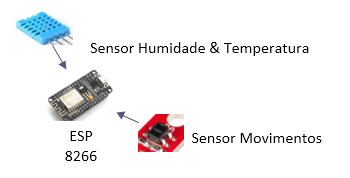


Figura 5 - Estufa sensors

# Comunicação serial entre Controller e Micro Controller

A ligação entre o Arduino mega e o Raspberry Pi 3 foi feita em Serial através de USB.

O arduino é responsável pela leitura sensorial, envia os dados para serial, o Raspberry Pi 3 fica à escuta, analisa os dados e envia para a plataforma WEB. O script de comunicação foi desenvolvido em Python e está presente no código em anexos.

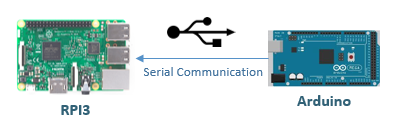


Figura 6 - Serial Communication

# Controlo de acessos (sensores)

A nossa thing no controlo de acessos é um Arduino Mega 2560 e possui os seguintes nós ligados:

**PIR** – possibilita controlo de presenças quando a porta está fechada;

**RC522** – possibilita a abertura de uma porta com um cartão RFID, ativação do mecanismo de configuração de cartões secundários com o cartão master;

**3x LEDs** – mostram feedback visual do sistema de abertura de cartões;

**Buzzer –** feedback sono do sistema de cartões.

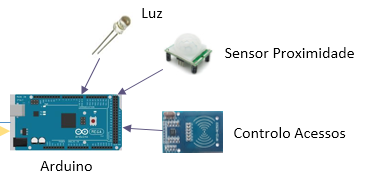


Figura 7 - Controlo de acessos (sensores)

# LoRa Network

Neste ponto temos um ponto de acesso LoRa gateway que funciona como receiver de qualquer sender que exista na rede LoRa.

O LoRa sender é o responsável por ler dados de um sensor, controlar um atuador, enviar/receber informação do LoRa gateway.

O gateway para além de um módulo de comunicação LoRa possui ainda uma interface WiFi. Desta forma conseguimos que um único equipamento seja capaz de interligar as duas redes (LoRa e Rede área local wireless) garantindo assim comunicação entre as mesmas.

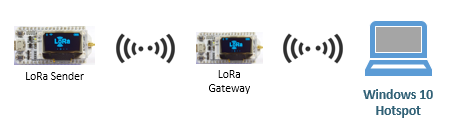


Figura 8 - LoRa Network

# Comunicação longa distancia (sensores)

Na área LoRa o único responsável pela aquisição de dados sensoriais é o sender pois o gateway não executa mais nenhuma função para além de fazer a comunicação entre a área LoRa e a rede local Wireless.

Os nós diretamente ligados ao LoRa sender são:

**Sensor distância** - mede o nível do silo em centímetros;

**Relay** - é despoletado através da plataforma para simular a ativação de uma bomba de água.

Modelo funcionamento:

Os dados são enviado por LoRa do sender para o gateway;

O gateway recebe os dados e envia por MQTT para o Raspberry Pi 3.

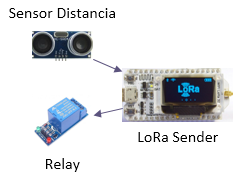


Figura 9 – LoRa sender (sensores)

# Serviços

Com recurso a toda a arquitetura descrita anteriormente o nosso projeto que se integra no âmbito de uma solução IoT Agrícola possui os seguintes serviços:

# Monitorização de áreas abertas com longas distâncias

Com recurso à tecnologia LoRa conseguimos abranger grandes distâncias ao ar livre (3-4Km áreas urbanas e 12Km em áreas rurais).



Figura 10 - Representação real da aplicação LoRa

O nosso foco na implementação da tecnológica LoRa surge pela necessidade de interligar áreas agrícolas onde existem silos, pequenos edifícios de madeira, motores de rega, estufas, casa do agricultor, etc., que podem estar dispersos num campo agrícola com vários quilómetros. Desta forma o responsável pelo campo agrícola tem acesso à informação em todos estes pontos, sem ter de se deslocar junto dos mesmos.

No nosso protótipo de alta-fidelidade simulamos a verificação do nível de um silo através de um sensor de distâncias e a ativação de um sistema de rega com recurso a um relay.

# Monitorização em áreas fechadas (estufa)

Por outro lado em áreas fechadas recorremos a uma tecnologia de comunicações sem fios que está presente em praticamente todas as casas que tenham um ponto de acesso à internet.

Uma rede Wireless 2.4GHz tem um ótimo funcionamento indor quando não existem obstáculos como paredes de betão. Neste sentido podemos afirmar que esta tecnologia permite-nos ter desempenhos e fiabilidade muito superior numa estufa comparado ao uso numa habitação.



Figura 11 - Representação real de uma estufa

Neste ponto o nosso protótipo controla a temperatura, humidade e deteção de roedores com recurso a sensor de infravermelhos (QRE).

# Controlo de acessos

Como em tudo, cada vez mais o controlo de acessos está presente nos mais diversos setores de mercado e a agricultura não foge à regra. Neste sentido o nosso protótipo inclui um sistema de abertura de porta através de cartões RFID e controlo de presenças através de infravermelhos. De salientar que caso seja detetado movimento no espaço, será enviado um e-mail de alerta através da plataforma. O nosso sistema possui ainda um mecanismo de registo de cartões SLAVE para abertura de portas, sendo que é necessário usar o cartão MASTER para ativar este modo.

# Disponibilização da informação *AnyTime*/*AnyWhere*

A plataforma de acesso aos dados foi desenvolvida com recurso à framework Laravel.

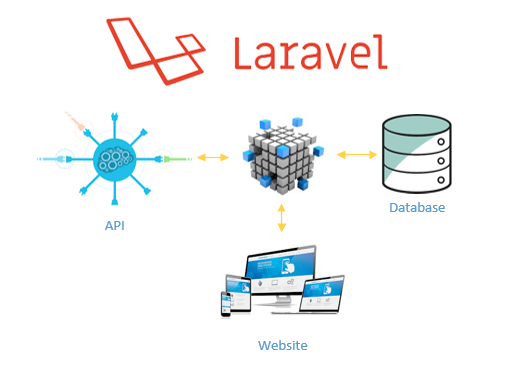


Figura 12 – Laravel Framework

# API

Tirando partido da Restful API incorporada na Framework Laravel, criamos rotas API para que o nosso THING Controller (Raspberry Pi 3) seja capaz de enviar dados para a plataforma. Estes dados que chegam à plataforma são geridos por métodos que programamos nos respetivos controladores onde armazenamos a informação na base dados.

De salientar novamente que também fizemos o processo inverso onde o Raspberry Pi 3, possui uma Resful API em Node.Js com o fim de receber dados da plataforma e enviar para as nossas things/atuadores.

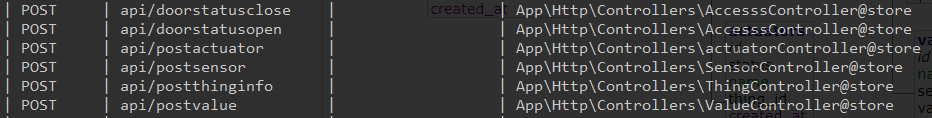


Figura 13 - Laravel API Routes

# Base dados

A nossa base de dados regista todos os dados de cada sensor, thing, e atuador.

Com recurso à framework Laravel, criamos scripts em PHP (migrations) que são capazes de gerar automaticamente todas as tabelas em qualquer servidor MySql (mariaDB). Para efeitos de testes durante o desenvolvimento criamos ainda seeds que são capazes de popular todas as tabelas com dados dummy.

A imagem abaixo representa o esquema da nossa base de dados com todas as tabelas e dependências.

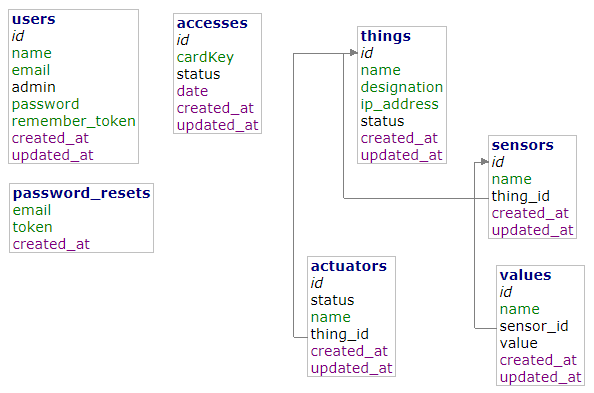


Figura 14 - Esquema base de dados

# Website

O website é a plataforma IOT que nos permite ter acesso aos dados *AnyTime*/*AnyWhere.*

As funcionalidades que integramos na nossa plataforma são as seguintes:

- Estatísticas que fazem a contagem do número de things ligados, sensores e atuadores;

- Visualizar a lista de Sensores;

- Visualizar a lista de Atuadores;

- Visualizar a lista de Things;

- Visualizar os valores sensoriais;

- Visualizar o estado dos equipamentos (ligados/desligados);

- Despoletar uma ação num atuador através de um botão (ligar/desligar);

- Controlo de acessos (IT room);

- Envio de avisos por e-mail;

- Autenticação da plataforma (login);

O código da nossa plataforma está ainda otimizado tanto para computador como dispositivo móvel onde as dimensões do ecrã são inferiores.

Para o desenvolvimento da plataforma foi usada a estrutura Model-View-Controller (MVC) para separar a parte lógica/backend da plataforma das vistas/frontend. O backend foi totalmente desenvolvido na linguagem PHP e o frontend em Blade que são templates de HTML onde é possível a programação de PHP.

A metodologia de desenvolvimento usada para o desenvolvimento foi a Agile, na medida em que separámos a plataforma por processos e user stories para ajudar a desenvolver passo a passo e ver as dependências surgiam ao longo do tempo.

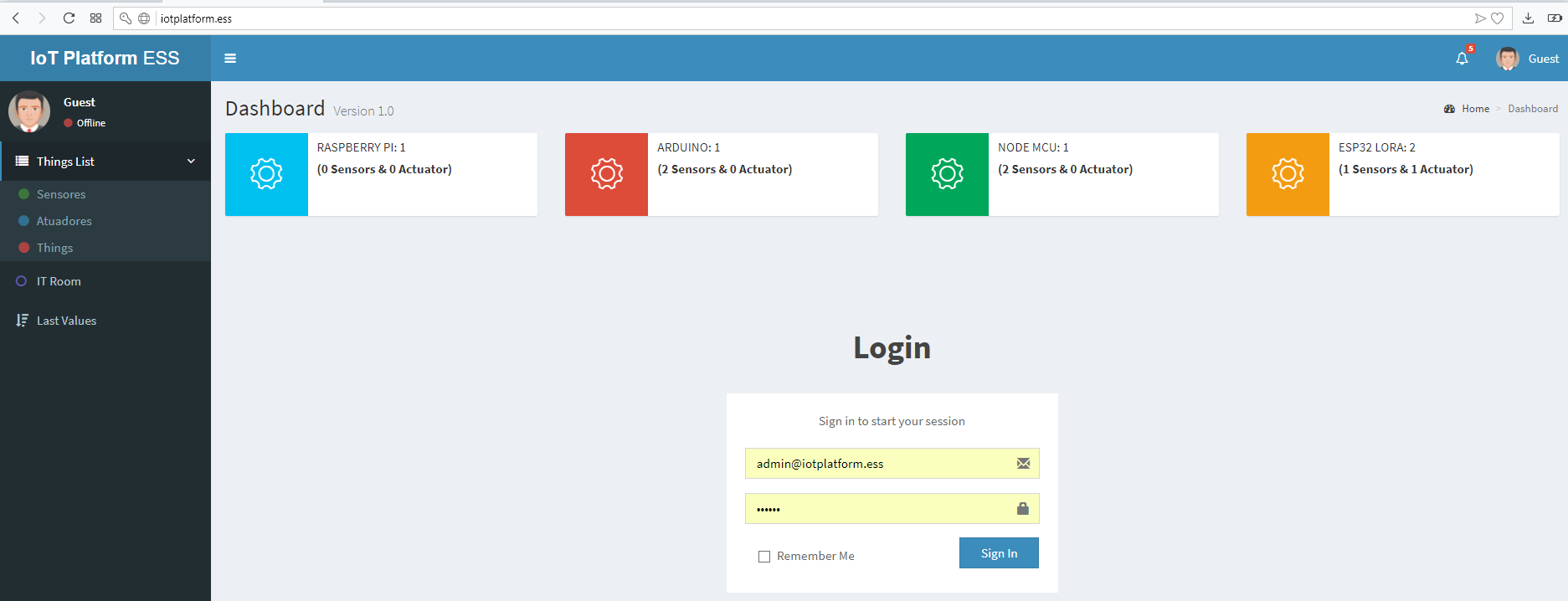


Figura 15 - Plataforma IoT (1)

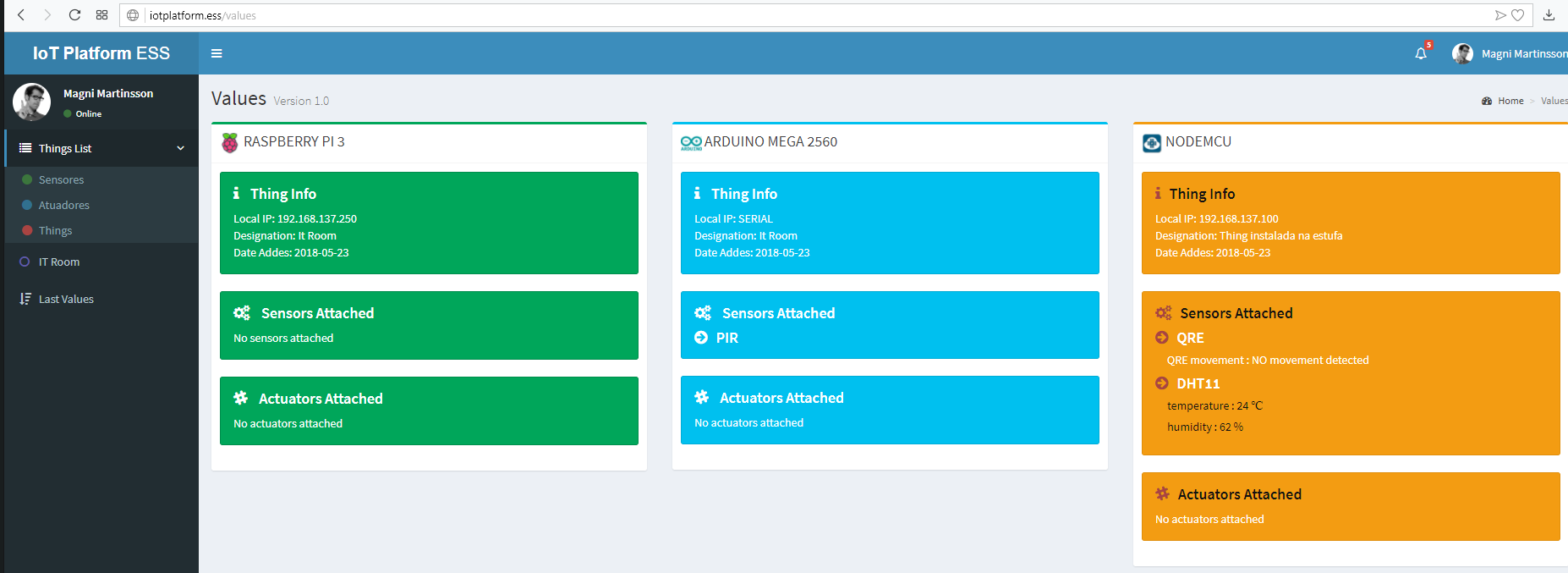


Figura 16 - Plataforma IoT (2)

# Alertas por e-mail

Como forma de despoletar um aviso de acontecimentos importantes, decidimos implementar na plataforma o envio de e-mails. Tal acontece quando o sensor PIR conectado ao Arduino Mega 2560 deteta movimento movimento na IT room quando a porta está fechada e também quando é detetado movimento de intrusão pelo sensor de infravermelhos QRE conectado ao Node MCU.



# Estado dos equipamentos

Para saber os estados dos equipamentos da rede IoT a partir da plataforma, foi desenvolvido um script que usa a função Echo do ICMP, também conhecido por ping para saber o estado dos equipamentos que têm um IPv4. Este script após recolher o estado dos equipamentos envia via API disponibilizada pela plataforma o seu estado. Para conhecer o estado do Arduino Mega 2560 que está ligado ao Raspberry Pi 3 via Serial, é feito um varrimento às portas USB do Raspberry Pi 3 para saber se o equipamento se encontra ativo. No fim, para conhecer o estado do ESP32 Lora, o ESP32 Lora Gateway envia de “x” em “x” tempo via API o estado do equipamento. Esse estado é recebido pelo Lora Gateway como forma de pacote lora. Quando não é recebido esse pacote, o Lora Gateway envia para a plataforma a informação de que o Lora sender se encontra inativo.

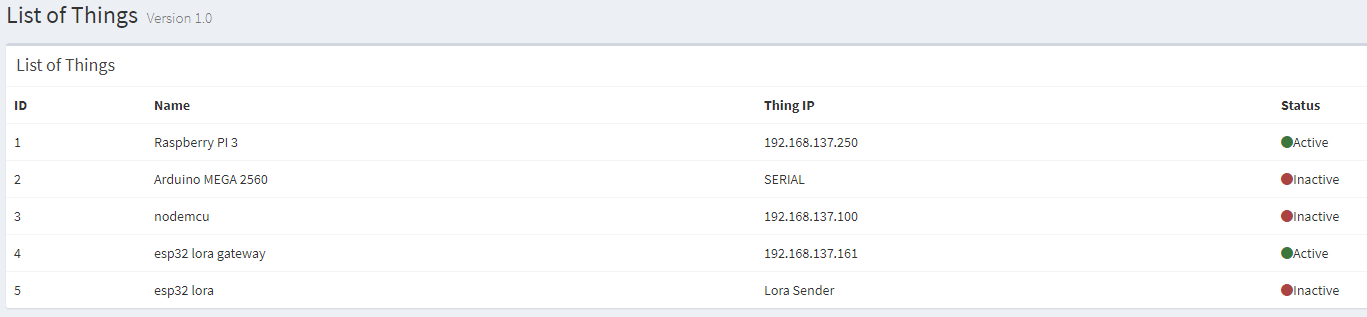


Figura 17 - lista de estados (plataforma)

# Desafios/Dificuldades/Trabalho a Desenvolver/Motivação pessoal

Durante o processo de desenvolvimento da solução IoT descrita, o maior desafio foi conseguir dominar a comunicação LoRa entre as things, na medida em que não tínhamos tal conhecimento e por isso teve que ser feito um trabalho de investigação em antemão para que fosse possível ter uma rede LoRa em que se pudesse transmitir os dados de um equipamento a outro.

Não tivemos grandes dificuldades a desenvolver a solução porque para além do LoRa, já dominávamos as restantes tecnologias usadas.

No futuro, para tornar o desenvolvimento deste projeto mais robusto, seria uma mais-valia a autoconfiguração das things, na medida em que o utilizador final não tem que se preocupar em fazer o setup dos equipamentos e sensores. Toda a parte de software seria enviada remotamente. O utilizador final apenas tem que os ligar a uma fonte de energia, na respetiva zona de rede e os dados são transmitidos para a plataforma.

Outra possível solução na área da agricultura seria a implementação de sensores que comunicassem via Bluetooth Low Energy (BLE). Este tipo de equipamentos tem uma particularidade interessante que passa pelo facto possuírem um modo acordar/desligar (ligam para comunicar dados e entram em sleep mode quando terminam de enviar) poupando assim bateria. A vantagem desta solução é a possibilidade de ter um campo agrícola com vários equipamentos espalhados que comunicam num estilo de ‘full mesh’, não tendo a necessidade de ter um ponto central de comunicação.

Por fim seria necessário tornar a comunicação dos dados da rede IoT para a Cloud cifrada. Uma possível solução seria a implementação de JSON WEB Token (JWT), em que quando o gateway da rede IoT envia dados pela API, envia junto um token de autenticação que é verificado na plataforma.

O que nos motivou mais foi o facto de conseguir juntar um vasto leque de tecnologias numa solução IoT que consiga funcionar em conjunto com a plataforma criada de raiz. O desenvolvimento da solução deu-nos a possibilidade de aprender novas tecnologias, como é o caso do LoRa e que nos enriqueceu como futuros engenheiros na área.

# Conclusão

O desenvolvimento deste projeto deu-nos a possibilidade de trabalhar com várias tecnologias para criar uma solução IoT que por sua vez comunica dados para uma plataforma WEB alojada na Cloud e que está disponível *Anytime*/*Anywhere*. Ajudou-nos também a conhecer novas tecnologias que enriquecem o nosso conhecimento e que nos será de valor para o futuro, na medida em que seremos capazes de propor soluções IoT Low-Cost e eficazes com dados disponíveis na Cloud. Conseguimos alcançar com sucesso a nossa proposta inicial da solução IoT, no entanto, para fins de prototipagem e estando já referido neste presente relatório, optámos por criar apenas uma área LoRa em vez de duas, omitindo uma delas por falta de equipamentos (LoRa).

# Referências bibliográficas

1. Vasisht D, Kapetanovic Z, Won J, et al. FarmBeats: An IoT Platform for Data-Driven Agriculture. Nsdi ’17. 2017:515-529. https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/vasisht.

2. Centenaro M, Vangelista L, Zanella A, Zorzi M. Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. IEEE Wirel Commun. 2016;23(5):60-67. doi:10.1109/MWC.2016.7721743

3. Augustin A, Yi J, Clausen T, Townsley W. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. Sensors. 2016;16(9):1466. doi:10.3390/s16091466

4. Prathibha SR, Hongal A, Jyothi MP. IOT Based Monitoring System in Smart Agriculture. In: 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT). ; 2017:81-84. doi:10.1109/ICRAECT.2017.52

5. Familiar B. Microservice Architecture. In: Microservices, IoT, and Azure. ; 2015. doi:10.1007/978-1-4842-1275-2\_3

6. Stanford-clark A, Truong HL. MQTT For Sensor Networks ( MQTT-S ) Protocol Specification. mqttorg. 2008.

7. Orenstein D. Application Programming Interface. Computerworld. 2000.