



Universidade do Minho
Escola de Engenharia

MESTRADO INTEGRADO EM ENGENHARIA DE TELECOMUNICAÇÕES E
INFORMÁTICA

MICRODISPOSITIVOS DE RF PARA COMUNICAÇÃO SEM-FIOS

SISTEMA DE MONITORIZAÇÃO DE ESTUFAS DOMÉSTICAS
RELATÓRIO INTERMÉDIO

Grupo 9:

Augusto César da Silva Mota - A76563

Hugo Daniel da Costa Cunha Machado - A80662

João Fernando Alonso Barbosa - A68453

José Pedro Afonso Rocha - A70020

Miguel Ângelo Teixeira Pereira - A82875

Tiago Martins Teles - A76266

Guimarães, 14 de novembro de 2020

Índice

| | | |
|----------|--|-----------|
| 1 | Introdução | 5 |
| 2 | Síntese do projeto | 6 |
| 3 | Conceitos teóricos | 7 |
| 3.1 | NodeMCU-32 | 7 |
| 3.2 | Sensores | 8 |
| 3.2.1 | <i>DHT 22</i> | 8 |
| 3.2.2 | <i>Moisture Sensor Hygrometer</i> | 9 |
| 3.2.3 | <i>KY-018 Photo LDR Resistor</i> | 10 |
| 3.3 | Atuadores | 11 |
| 3.3.1 | <i>Mini Water Pump DC 3V</i> | 11 |
| 3.3.2 | <i>Step Motor</i> | 11 |
| 4 | Planeamento | 12 |
| 4.1 | Ferramentas utilizadas | 13 |
| 5 | Arquitetura do sistema | 14 |
| 5.1 | Esquema geral | 14 |
| 5.2 | Sistema Sensor Estufa | 15 |
| 5.3 | <i>Cloud IoT</i> | 16 |
| 5.3.1 | Broker MQTT | 16 |
| 5.3.2 | API | 16 |
| 5.3.3 | BD | 18 |
| 5.4 | Aplicação <i>web</i> e <i>mobile</i> | 18 |
| 5.4.1 | <i>User Interface (UI)</i> | 18 |
| 6 | Funcionalidades | 20 |
| 7 | Requisitos | 21 |
| 7.1 | Requisitos funcionais | 21 |
| 7.2 | Requisitos não funcionais | 21 |
| 8 | Conclusão | 22 |

9 Referências

24

Índice de Imagens

| | | |
|----|--|----|
| 1 | NodeMCU-32 [2]. | 7 |
| 2 | Estrutura interna sensor humidade [3]. | 8 |
| 3 | Coefficiente de Temperatura Negativo [5]. | 8 |
| 4 | Pinout do Moisture Sensor [7]. | 9 |
| 5 | Pinout do photo LDR resistor[9]. | 10 |
| 6 | Mini Water Pump[11]. | 11 |
| 7 | Esquema do step motor | 11 |
| 8 | Diagrama de Gantt da primeira fase do projeto. | 12 |
| 9 | Diagrama de Gantt da segunda fase do projeto. | 12 |
| 10 | Arquitetura do sistema. | 14 |
| 11 | Arquitetura do sistema sensor. | 15 |
| 12 | Base de dados. | 18 |

Índice de Tabelas

| | | |
|---|---------------------------|----|
| 1 | Recursos da API | 17 |
|---|---------------------------|----|

1. Introdução

Na unidade curricular de Microdispositivos de RF para Comunicações Sem Fios, foi-nos proposto realizar um projeto que consiste na criação de um sistema de comunicação sem fios para a monitorização de parâmetros de um sistema com base na utilização de sensores e dispositivos RF.

Nos dias que vivemos, tem-se dado cada vez mais valor e importância aos produtos biológicos. Como tal, existe uma maior procura e também uma subida dos preços desses produtos. São indicados como produtos que trazem vantagens para a saúde do consumidor por terem uma grande quantidade de antioxidantes e, como são produzidos sem recurso a produtos químicos, são menos prejudiciais para o planeta.

O nosso grupo decidiu, em unanimidade, criar uma pequena estufa de uso doméstico que, tal como nos foi pedido, fosse monitorizada e controlada com um sistema de comunicação sem fios.

Como a agricultura é uma área que ainda tem pouco investimento tecnológico ao nível da monitorização e registo de variáveis, este projeto pode, mais tarde, ser adaptado para ambientes maiores e com características diferentes daquelas que vão ser implementadas.

Neste relatório intermédio será apresentado todo o planeamento necessário para o desenvolvimento deste sistema proposto pelo grupo.

2. Síntese do projeto

Globalmente, este sistema contém um conjunto de dispositivos e sistemas críticos tais como os microcontroladores, servidores *web* e de base de dados e dispositivos de comunicação. Para este projeto foi feita uma divisão em 3 momentos de avaliação, sendo que para cada momento estão associadas várias etapas específicas:

- **Fase Intermédia** - Planeamento temporal do projeto, desenvolvimento conceptual do sistema, identificação de tecnologias/recursos e relatório intermédio;
- **Fase Final** - Implementação do sistema, testes/análise de resultados e relatório final;
- **Demonstração prática.**

3. Conceitos teóricos

Um dos principais objectivos deste projeto é a capacidade de controlar uma estufa através do uso do poder da Internet das Coisas(IoT) através de ferramentas e softwares. A nossa estufa será composta por sensores que monitorizam o estado em tempo real e atuadores para realizar acções quando necessárias.

Considerando as funcionalidades do nosso projeto, os materiais foram divididos em três categorias.

3.1. NodeMCU-32

Placa de desenvolvimento baseada no módulo ESP32 Wi-Fi, um equipamento electrónico desenvolvido com o intuito de conectar projectos á Internet com grande facilidade e baixo custo.

Dispositivo muito eficiente, constituído por um módulo ESP-WROOM-32 que contem o ESP32, um cristal de 40MHz, antena embutida e uma porta micro USB. O NodeMCU-32 também contem Bluetooth v4.2 embutido e um micro controlador *dual core* 32-bit, o que torna este dispositivo mais prático e aumenta as possibilidade de uso.

Esta placa de desenvolvimento é extremamente adequada para iniciar aplicativos de IoT como é pretendido no nosso projecto [1].



Figura 1: NodeMCU-32 [2].

3.2. Sensores

3.2.1. *DHT 22*

Este modulo com baixo consumo de corrente é constituído por um sensor de temperatura e humidade, e também um conversor analógico/digital para realizar comunicação com o micro-controlador. Uma das grandes vantagens da utilização deste módulo é oferecer uma leitura de valores com precisão de valores decimais.

O componente de sensor humidade, é constituído por dois eléctrodos com substrato de retenção de humidade. No substrato, os íons são liberados á medida que o vapor de água é absorvido por ele, aumentando a condutividade entre os eletrodos. As alterações na resistência entre os dois eletrodos vai ser proporcional á humidade relativa, sendo que a humidade relativa mais alta diminui a resistência entre os eletrodos, e a humidade relativa mais baixa aumenta a resistência entre os eletrodos.

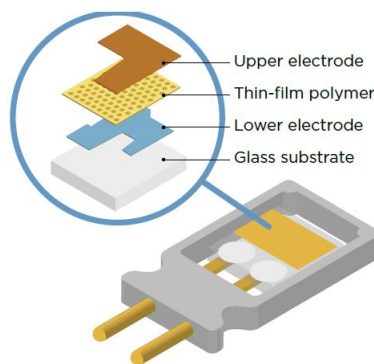


Figura 2: Estrutura interna sensor humidade [3].

A componente de sensor temperatura é constituído por um termistor, sendo um resistor térmico que muda a sua resistência com a temperatura. Este termistor é feito de forma a que a sua resistência mude drasticamente com a temperatura [4].

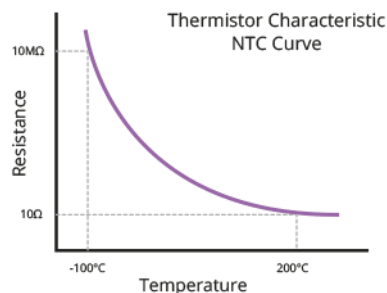


Figura 3: Coeficiente de Temperatura Negativo [5].

3.2.2. *Moisture Sensor Hygrometer*

Este sensor é constituído por uma sonda em forma de garfo com dois condutores que se vão encontrar expostos no solo em qualquer tipo de lugar. Funciona com a sonda em forma de garfo com dois condutores expostos atuando como um resistor variável onde a resistência varia de acordo com o teor de água no solo, sendo esta resistência inversamente proporcional á humidade do solo:

- Quanta mais água se encontrar no solo, melhor condutividade resultando um valor menor na resistência
- Quanta menos água se encontrar no solo, má condutividade resultando um valor maior na resistência.

Este sensor gera um tensão de saída de acordo com o valor da resistência, e se for medido é possível obter o nível de humidade.

O sensor, é composto por um módulo eletrónico que vai realizar uma conexão da sonda com o arduino, módulo que vai produzir uma tensão de saída de acordo com o valor da resistência da ponta prova e de seguida é disponibilizado num pino de saída analógica. Este sinal analógico de seguida é enviado para um comparador de alta precisão para digitalizar e disponibilizar num pino de saída digital [6].

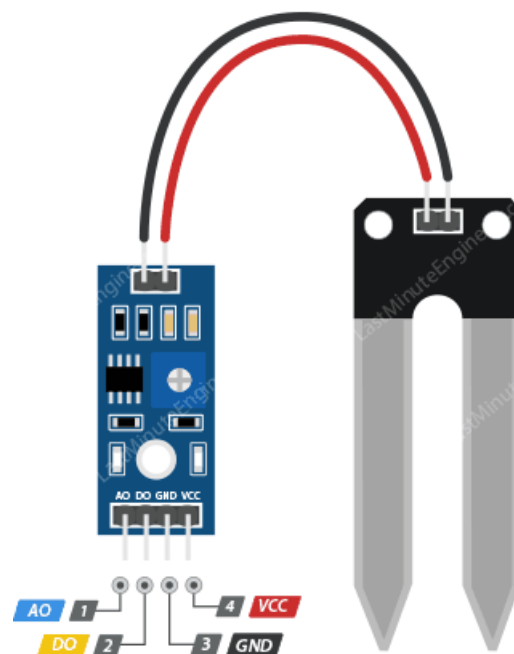


Figura 4: Pinout do Moisture Sensor [7].

3.2.3. *KY-018 Photo LDR Resistor*

Este tipo de sensor, é um dispositivo sensível á luz que é utilizado para indicar a presença ou ausência de luz para medir a sua intensidade. Quando se encontra na presença de um ambiente escuro, o valor da sua resistência é muito alto, mas quando se encontra numa zona iluminada, o valor da resistência diminui drasticamente. Este sensor é constituído por um resistor depende de luz que tem uma sensibilidade que varia com o comprimento de onda da luz que é aplicada[8].

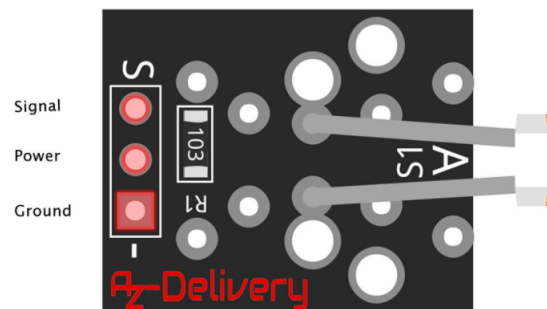


Figura 5: Pinout do photo LDR resistor[9].

3.3. Atuadores

3.3.1. *Mini Water Pump DC 3V*

O *Water pump* é basicamente um motor DC que é alimentado com 3V. Quando se encontra ligado, a bomba suga a água pela lateral do invólucro de plástico e liberta para fora da porta do tubo. Mudar a polaridade deste atuador não vai transformar num dispositivo de sucção, este actuador apenas tem a função de bombear a água e apenas só. Atuador que vai ter a função de regar os nossos vegetais que se encontram na estufa[10].



Figura 6: Mini Water Pump[11].

3.3.2. *Step Motor*

Os step motor já se encontram disponíveis no mercado á muito tempo, sendo utilizados em bastante aplicações. Atuador com pequeno tamanho, mas com grande eficiência em termos de energia.

Este atuador é constituído por um pequeno motor, potenciómetro e um circuito de controlo. À medida que este motor gira a resistência do potenciómetro muda de modo a que o circuito de controlo possa regular precisamente quando existe movimento. Quando este motor se encontrar na posição desejada, a alimentação que é fornecida ao motor vai ser interrompida, se ainda não se encontrar na posição desejada, o motor continua a girar na direcção desejada. Em baixo encontra-se um esquema de demonstração de como este atuador funciona[12].

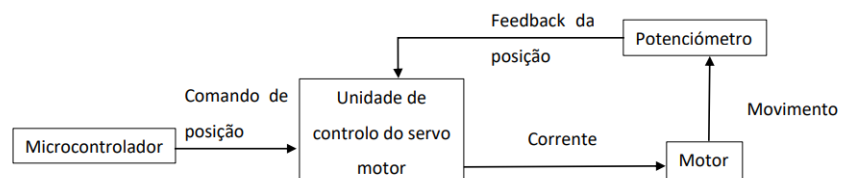


Figura 7: Esquema do step motor

4. Planeamento

Nesta secção encontra-se disponível a planificação temporal, do nosso grupo para as diferentes fases do projeto, bem como o conjunto de ferramentas que foram utilizadas.

Em baixo, são apresentados os diagramas de Gantt - figuras 8 e 9.

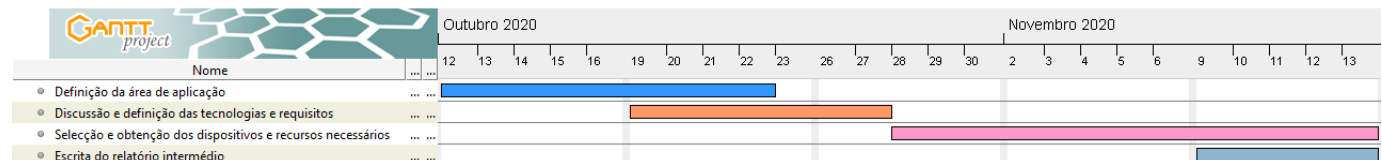


Figura 8: Diagrama de Gantt da primeira fase do projeto.

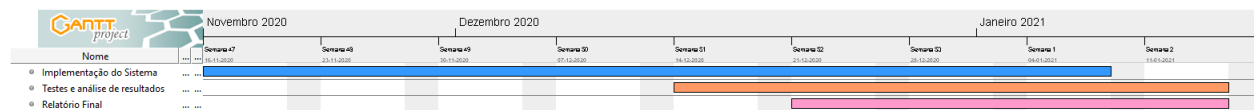


Figura 9: Diagrama de Gantt da segunda fase do projeto.

4.1. Ferramentas utilizadas

As ferramentas utilizadas serão as seguintes:

- Programa **GanttProject** para planeamento temporal das tarefas do grupo;
- Programa **Arduino IDE**, para editar, compilar e enviar código para a placa Arduino;
- Programa **MySQL**, para criação e gestão de bases de dados;
- Programa **Visual Studio Code**, para editar e compilar código;
- Programa **draw.io**, para elaboração de diagramas e esquemas gráficos;
- Plataforma **GitHub**, para partilha e organização do código desenvolvido pelo grupo;
- Plataforma **OverLeaf**, para elaboração de relatórios em LaTeX;
- Plataforma **circuito.io**, para desenho de esquemáticos;
- Plataforma **Discord**, para reuniões de equipa.

5. Arquitetura do sistema

5.1. Esquema geral

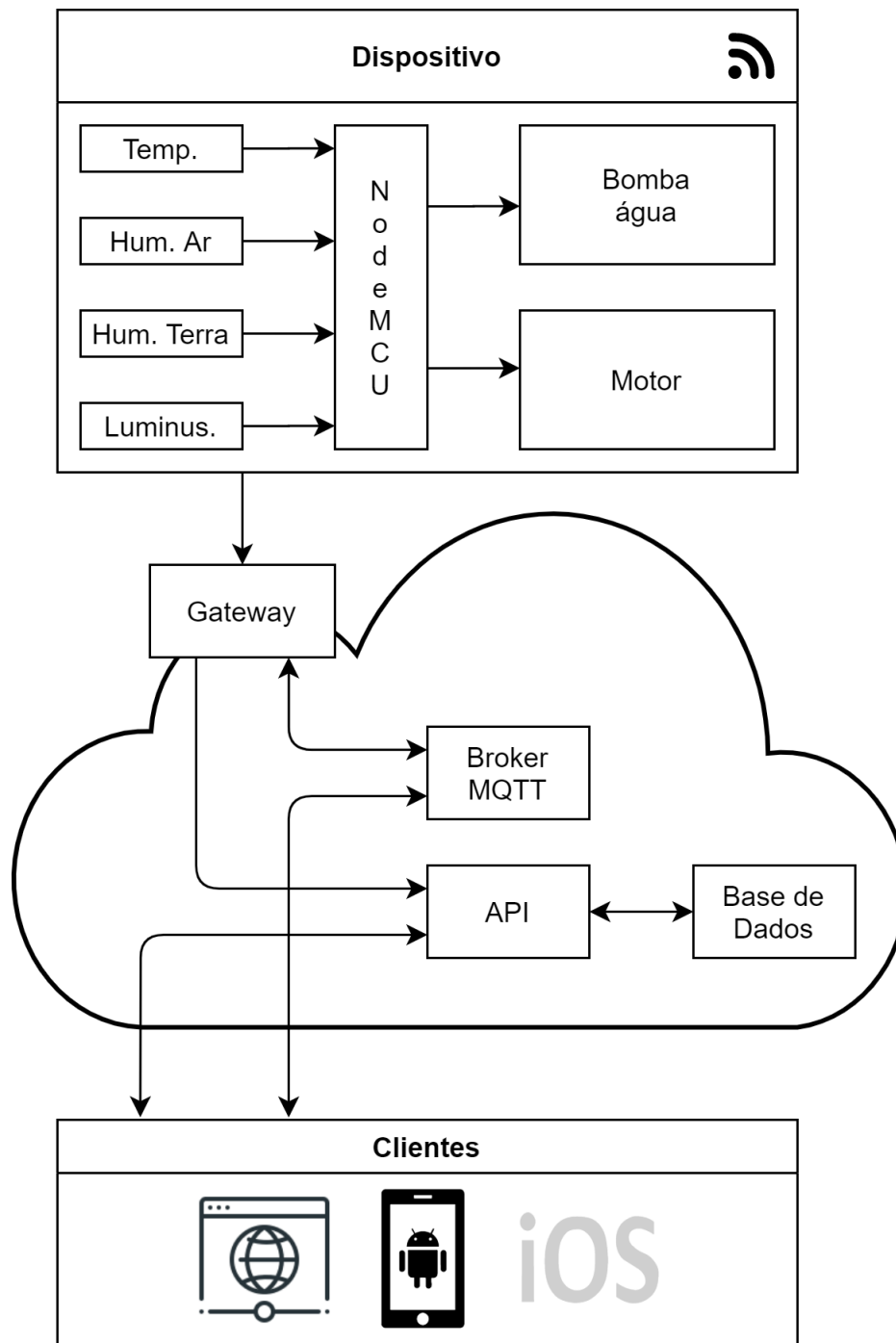


Figura 10: Arquitetura do sistema.

5.2. Sistema Sensor Estufa

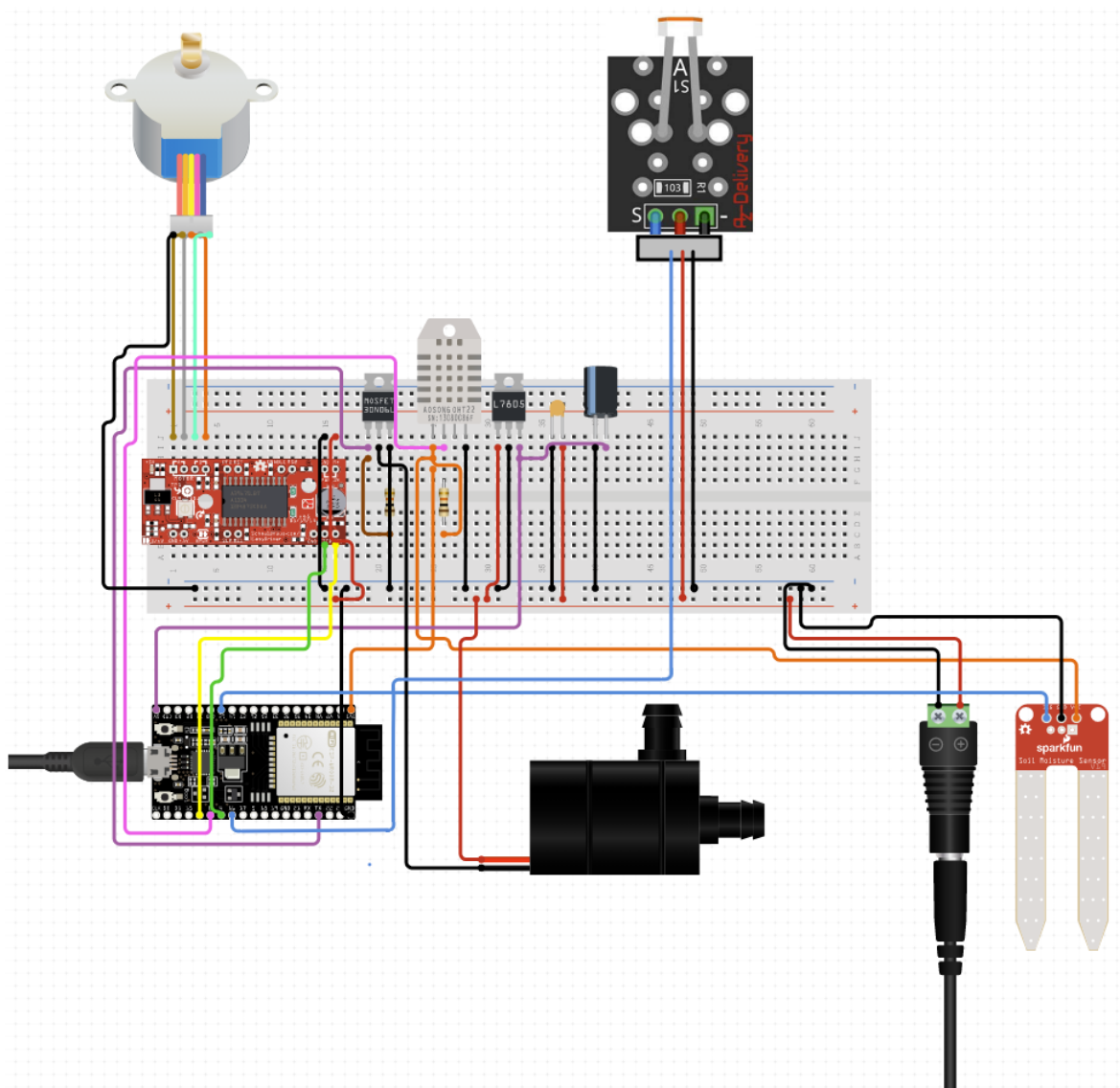


Figura 11: Arquitetura do sistema sensor.

O sistema sensor estufa vai ser constituído por um conjunto de módulos dos sensores de luminosidade, temperatura e humidade do ar e humidade da terra, um conjunto de módulos atuadores para rega e abertura de estufa e por fim um respetivo microcontrolador para a gestão destes módulos.

Estes componentes serão responsáveis pela recolha de dados, controlo do cultivo em questão e serão posteriormente enviados para a *Cloud* IoT definida na arquitetura, onde serão processados e disponibilizados para os utilizadores.

Este sistema sensor receberá da *Cloud* um conjunto de parâmetros a cumprir para funcionar de forma autónoma.

5.3. *Cloud* IoT

5.3.1. Broker MQTT

A sigla MQTT traduz-se em *Message Queuing Telemetry Transport* e refere-se a um protocolo emergente nesta área dos IoT baseado na pilha TCP/IP. Este protocolo permite uma comunicação assíncrona entre as partes intervenientes, geralmente de máquina para máquina, de uma forma simples, leve e eficaz utilizando um *broker* que serve de intermediário.

Este é um protocolo com baixos requisitos de largura de banda e de *hardware*, poderá também ser considerado pouco confiável, por razões de alta latência. Mas visto que na nossa aplicação não possui esse tipo de exigências, este protocolo é a escolha ideal. Além disso, as mais recentes versões prevêm a possibilidade de utilizar diferentes níveis de QoS, garantindo uma qualidade superior na entrega das mensagens.

O *broker* é responsável por gerir as publicações e as subscrições permitindo que uma publicação seja entregue a vários subscritores. Este protocolo traz vantagens para o nosso projeto porque irá ser por este meio que iremos obter os estados dos sensores em *near-real-time* bem como a execução de algumas operações.

Existem atualmente vários *brokers* de utilização pública e gratuita que serão utilizados neste projeto ou então poderá ser implementado um *broker* privado. Consideramos que inicialmente e para efeitos demonstrativos, a implementação de um *broker* privado não é um requisito essencial nem é o foco do projeto, apesar de trazer uma camada de segurança adicional que fará sentido numa vertente mais profissional.

5.3.2. API

Para a comunicação entre duas aplicações de software, onde existe uma requisição de serviços entre essas aplicações, é necessário código que consiga realizar corretamente esta troca de dados.

O grupo decidiu que iria usar uma *Rest* API, um tipo de API que utiliza pedidos HTTP para a gestão de dados na comunicação de aplicações do tipo *web*.

As metodologias HTTP utilizadas por este tipo de API são as seguintes:

- Método **GET** - obter um recurso;
- Método **PUT** - mudar ou atualizar um recurso;
- Método **POST** - criar um recurso;
- Método **DELETE** - eliminar um recurso.

Salienta-se que estas metodologias são definidas pelo protocolo RFC2616(*Hypertext Transfer Protocol*).

Este tipo de API é muito usado em situações *web* devido às rotinas que são usadas serem *stateless*, ou seja, sem estado associado onde os pedidos entre servidor *web* e serviços associados são baseados na informação gerada na interação entre os dois. Isto torna-se determinante pois permite torna este sistema escalável e de fácil reutilização de serviços em caso de falha de algum deles [14].

O grupo já definiu a maior parte dos recursos da API desenvolvida e, na tabela a seguir representada, podemos visualizar esses mesmos recursos:

Tabela 1: Recursos da API

| URI | Método HTTP | Descrição | Parâmetros |
|------------------|-------------|--|--|
| /login | POST | Efetua o login | -username -password |
| /logout | GET | Efetua logout | |
| /user | GET | Obtém um <i>user</i> | |
| /user/:id | GET | Obtém um <i>user</i> através do id | |
| /user | POST | Insere novo <i>user</i> | -nome -username -password |
| /user/:id | PUT | Atualiza <i>user</i> | -nome -password |
| /user/:id | DELETE | Remove sujeito | |
| /device/user | GET | Obtém todos os dispositivos associados | |
| /device/user/:id | GET | Obtém dispositivo através do id | |
| /device/user | POST | Introduzir novo dispositivo | -id -SN(Serial Number) -designação |
| /device/user/:id | DELETE | Remove dispositivo | |

5.3.3. BD

Para complementar o sistema, a base de dados irá servir para guardar a relação entre os utilizadores e os dispositivos, bem como um histórico de medições que cada dispositivo efectua regularmente.

Irá guardar também uma colecção de perfis predefinidos para controlar diversos tipos de culturas que posteriormente serão enviados para os dispositivos. Estes perfis poderão ser usados tal como definido ou com os parâmetros alterados antes de serem enviados para os dispositivos.

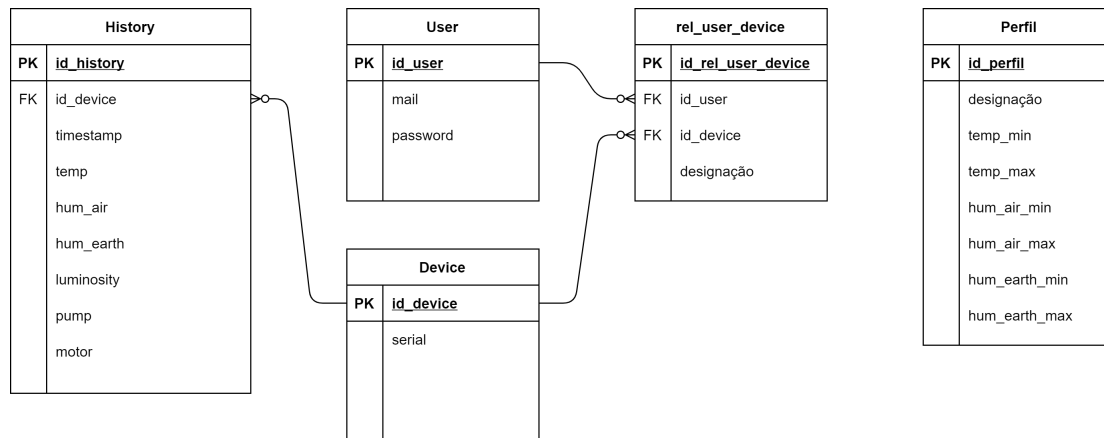


Figura 12: Base de dados.

5.4. Aplicação *web* e *mobile*

5.4.1. *User Interface* (UI)

Ao iniciar a aplicação, os utilizadores serão acolhidos por um ecrã de autenticação onde podem proceder ao seu novo registo ou, caso já possuam, conta podem fazer *login*. Os utilizadores da nossa aplicação poderão interagir com ela através de uma *dashboard* principal, onde são apresentadas informação relativas às várias estufas do utilizador em sessão, podendo também tomar decisões sobre como se comporta cada estufa, instruindo assim a mesma numa variedade de ações. Por estufa, são apresentados os seus dados de identificação, bem como informações sobre estado da estufa (temperaturas e níveis de humidade) e a sua atual configuração (níveis mínimos e máximos de humidade e perfis comportamento).

Cada estufa apresentada possui também ferramentas para que o cliente facilmente possa tomar ações sobre ela, como por exemplo, abrir e fechar a mesma, ou mesmo mudar o atual perfil de configuração .

Paralelamente aos dados em tempo real, irá ser possível consultar o histórico de cada estufa, isto é, consultar diversos gráficos que mostram a evolução do estado da estufa e dos seus parâmetros ao longo do tempo. O design será responsivo e poderá ser usado quer em *web*, quer em *mobile*.

6. Funcionalidades

Neste projeto temos como objetivo um produto funcional para o consumidor final. Isto é, funcionalidades praticas, fácil interação com o produto e fácil montagem.

Em primeiro lugar, para o produto funcionar é necessário a ligação á Internet. E para que seja possível a configuração à rede de casa, o micro-dispositivo possui uma sessão inicial(esta sessão poderá também ser forçada) que permite a configuração das credenciais através de uma pagina *Web*. Isto impede que as credenciais tenham de ser programadas pelo utilizador final.

Após a ligação á Internet, o sistema sensor deverá ser registado na conta do utilizador, na plataforma IoT. Desta forma o utilizador poderá começar a monitorizar e controlar o seu cultivo remotamente.

Através da plataforma IoT será possível, através de um painel, monitorizar a temperatura do ar, humidade do ar e da terra e a luminosidade diária. Também será possível criar e editar perfis de cultivo, isto é, opções para regar todos os dias à hora especificada, ou para regar caso a terra esteja pouco húmida, ou caso a temperatura interior ultrapasse um dado valor é necessário abrir a estufa.

Com base nestas opções, poderá ser criado um perfil perfeito, de acordo com as condições pretendidas e a planta em questão, sendo possível também usar perfis públicos já existentes ou partilhados por outros membros da plataforma.

7. Requisitos

7.1. Requisitos funcionais

Para o correto funcionamento deste sistema na sua globalidade, foi necessário que os seguintes requisitos fossem cumpridos:

- Captação correta dos valores dos sensores de temperatura, humidade do ar, humidade da terra e de luminosidade;
- Interpretação correta dos dados enviados e recebidos pelo microcontrolador;
- Qualidade do serviço entre microcontrolador e a plataforma IoT;
- Correto armazenamento dos dados na base de dados;
- Interface que satisfaça todos os requisitos para a boa utilização deste sistema e de acordo com o mesmo.

7.2. Requisitos não funcionais

Os requisitos deste sistema que serão descritos a seguir são não funcionais, isso implica que estejam mais relacionados como o tempo de resposta e a fiabilidade do sistema logo, eles estão ligados com as funcionalidades do sistema implementado. Os requisitos não funcionais definidos pelo grupo foram:

- Fiabilidade dos dados recolhidos pelos sensores;
- Fiabilidade da comunicação entre os os sensores, microcontroladores e atuadores;
- Fiabilidade da comunicação entre *Gateways*, *Cloud* e aplicações *Web*;
- Configuração da obtenção das amostras vindas dos sensores;
- Configuração dos parâmetros da comunicação entre o *microcontrolador* e *Gateway*;
- Configuração dos parâmetros da comunicação entre o *Gateway* e *Cloud*.

8. Conclusão

Nesta fase do projeto debruçamo-nos na planificação da implementação da nossa aplicação. Após a decisão sobre a área de aplicação, foi feita uma pesquisa na área da agricultura com o objetivo de escolher os dispositivos RF necessários para a elaboração de um trabalho adequado e eficiente. Tendo tudo isso acertado, procedeu-se à escolha e respetiva compra dos dispositivos e dos recursos necessários.

Finda esta primeira fase do projeto, podemos identificar alguns objetivos que foram cumpridos com sucesso: a familiarização com os conceitos base da nossa aplicação, o planeamento do projeto até ao final do semestre, a elaboração de uma arquitetura e a definição concisa dos objetivos finais. De notar que, por diferentes fatores, este planeamento poderá sofrer alterações que irão tornar este projeto ainda mais desafiante.

O grupo ficou satisfeito com planeamento delineado para o sistema a ser desenvolvido e acredita estar num bom caminho para criar um projeto completo, robusto e proveitoso.

9. Referências

- [1] **Usinainfo. 2020. ESP32 Nodemcu Placa De Desenvolvimento Iot ESP-32S - Usinainfo. [online]** Disponível em: <https://www.usinainfo.com.br/nodemcu/esp32-nodemcu-iot-com-wifi-e-bluetooth-30-pinos-5147.html> [Acedido em 14 Novembro 2020].
- [2] **Blogmasterwalkershop.com.br. 2020. [online]** Disponível em: https://blogmasterwalkershop.com.br/wp-content/uploads/2017/05/img02_conhecendo_o_nodemcu-32s_esp32_esp-32_wifi_bluetooth_esp8266_arduino_esp-wroom-32.jpg [Acedido em 14 Novembro de 2020].
- [3] **Vaisala.com. 2020. [online]** Disponível em: <https://www.vaisala.com/sites/default/files/inline-images/capacitive-humidity-sensor.jpg> [Acedido em 14 Novembro 2020].
- [4] **Last Minute Engineers. 2020. Insight Into How DHT11 DHT22 Sensor Works Interface It With Arduino. [online]** Disponível em: <https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/> [Acedido em 14 Novembro 2020].
- [5] **Lastminuteengineers.com. 2020. [online]** Disponível em: <https://lastminuteengineers.com/wp-content/uploads/arduino/NTC-Thermistor-Temperature-Resistance-Characteristic-Curve.png> [Acedido em 14 Novembro 2020].
- [6] **Lastminuteengineers.com. 2020. [online]** Disponível em: <https://lastminuteengineers.com/wp-content/uploads/arduino/NTC-Thermistor-Temperature-Resistance-Characteristic-Curve.png> [Acessed 14 November 2020].
- [7] **Lastminuteengineers.com. 2020. [online]** Disponível em: <https://lastminuteengineers.com/wp-content/uploads/arduino/NTC-Thermistor-Temperature-Resistance-Characteristic-Curve.png> [Acedido em 14 Novembro 2020].
- [8] **ArduinoModulesInfo. 2020. KY-018 Photoresistor Module - Arduinomodulesinfo. [online]** Disponível em: <https://arduinomodules.info/ky-018-photoresistor-module/> [Acedido em 14 Novembro 2020].
- [9] **Images-na.ssl-images-amazon.com. 2020. [online]** Disponível em: <https://images-na.ssl-images-amazon.com/images/I/61U81clfr3L.SL1500.jpg> [Acedido em 14 Novembro 2020].

- [10] **2020. [online]** Disponível em: <<https://www.digikey.com/catalog/en/partgroup/submersible-3v-dc-water-pump-horizontal-or-vertical-type/107104>> [Acedido em 14 Novembro 2020].
- [11] **5.imimg.com. 2020. [online]** Disponível em: <<https://5.imimg.com/data5/KQ/RZ/XM/SELLER-65880656/dc-3-6v-submersible-pump-mini-water-pump-500x500.jpg>> [Acedido em 14 Novembro 2020].
- [12] **Jameco.com. 2020. Servo Motors Work | How Servo Motors Work. [online]** <https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>> [Acedido em 14 Novembro 2020].
- [13] **Lastminuteengineers.com. 2020. [online]** Disponível em: <https://lastminuteengineers.com/wp-content/uploads/arduino/NTC-Thermistor-Temperature-Resistance-Characteristic-Curve.png> [Acedido em 14 Novembro 2020].
- [14] **Lastminuteengineers.com. 2020. [online]** Disponível em: <https://lastminuteengineers.com/wp-content/uploads/arduino/NTC-Thermistor-Temperature-Resistance-Characteristic-Curve.png> [Acedido em 14 Novembro 2020].
- [15] **Docente de Microdispositivos de RF para comunicação sem Fios, "Trabaho Prático",** Universidade do Minho, 2020