



# Sistema de Aquisição e Tratamento de Dados de um Sistema Hídrico Residencial

**NUNO MANUEL PINTO DE SOUSA**

Outubro de 2022

# SISTEMA DE AQUISIÇÃO E TRATAMENTO DE DADOS DE UM SISTEMA HÍDRICO RESIDENCIAL

**Nuno Manuel Pinto de Sousa**



**Mestrado em Engenharia Eletrotécnica e de Computadores**

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2022



Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Nuno Manuel Pinto de Sousa, Nº 1170992, 1170992@isep.ipp.pt  
Orientação científica: Lino Manuel Baptista Figueiredo, lbf@isep.ipp.pt



**Mestrado em Engenharia Eletrotécnica e de Computadores**

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

23 de outubro de 2022



---

# Agradecimentos

---

Primeiramente pretendo agradecer ao professor Lino Figueiredo pela pronta disponibilidade e orientação durante o desenvolvimento da dissertação.

Em seguida agradeço aos meus amigos por todos os momentos de diversão e a partilha de ideias durante todo o percurso acadêmico, dando notoriedade ao meu colega Diogo Fernando pela cedência de material para o desenvolvimento do projeto.

Quero agradecer também ao meu avô Júlio, à minha avó Rosalina e ao meu tio Maximino pela ajuda nas alterações do sistema hidráulico, na construção de diversos dispositivos e na discussão de ideias.

Por fim queria agradecer aos meus avós aos meus pais e ao meu irmão pela educação, apoio e suporte que me proporcionaram desde sempre ininterruptamente.



---

# Resumo

---

Portugal atravessa atualmente um período de seca agravado o que condiciona a vida quotidiana da população. Notou-se recentemente que para a sustentabilidade ser mantida, houve a necessidade da alteração das políticas de utilização dos recursos hídricos por parte das pessoas.

Este projeto visa criar uma ferramenta de auxílio de gestão dos recursos hídricos, analisando e reportando dados relativos a parâmetros que possibilitam caracterizar os consumos dos residentes de uma habitação. Permite também centralizar a informação recolhida por diferentes sensores disponibilizando-a através de uma interface gráfica acessível remotamente, facilitando a sua gestão e controlo.

O sistema divide-se em três secções distintas: na hierarquia mais baixa estão os módulos implementados que recolhem dados e atuam dispositivos. Em seguida, foi implementada uma aplicação de gestão do sistema hidráulico, responsável por realizar um controlo dos atuadores e processar os dados recebidos. Por fim, o último constituinte do sistema consiste na configuração e implementação de diversos *softwares* numa máquina central que disponibiliza diversos serviços de suporte da interface homem-máquina.

**Palavras-Chave:** Internet das coisas, Comunicação, Monitorização, Controlo, Condicionamento de sinal, Sustentabilidade



---

# Abstract

---

Portugal is currently going through a period of severe drought, which affects the daily life of the population. It was recently noted that to maintain sustainability, there was a need to change people's policies for the use of water resources.

This project aims to create a tool to help manage water resources, analyzing and reporting data on parameters that make it possible to characterize the consumption of residents of a dwelling. It also allows centralizing the information collected by different sensors, making it available through a graphical interface accessible remotely, facilitating its management and control.

The system is divided into three distinct sections: in the lowest hierarchy are the implemented modules that collect data and actuate devices. Then, a hydraulic system management application was implemented, responsible for controlling the actuators and processing the data received. Finally, the last component of the system consists of the configuration and implementation of several software in a central machine that provides several services to support the human-machine interface.

**Keywords:** Internet of Things, Communication, Monitoring, Control, Signal conditioning, Sustainability



---

# Índice

---

<b>Agradecimentos</b>	<b>v</b>
<b>Índice</b>	<b>i</b>
<b>Índice de Figuras</b>	<b>v</b>
<b>Índice de Tabelas</b>	<b>ix</b>
<b>Acrónimos</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Calendarização . . . . .	3
1.4 Organização do Relatório . . . . .	3
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Panorama Atual . . . . .	5
2.2 Cidades Inteligentes . . . . .	6
2.2.1 Características de uma Cidade Inteligente . . . . .	6
2.2.2 Riscos das cidades inteligentes . . . . .	8
2.3 Tecnologias de Comunicação Sem Fios . . . . .	9
2.3.1 Long Range . . . . .	9
2.3.2 Wireless Fidelity . . . . .	10
2.3.3 Rede Celular . . . . .	11
2.3.4 Sigfox . . . . .	12
2.3.5 ZigBee . . . . .	13
2.3.6 Narrowband Internet of Things . . . . .	13
2.3.7 Comparação das Diferentes Tecnologias . . . . .	14
2.4 Tecnologias de Comunicação Com Fios . . . . .	14

2.4.1	Controller Area Network . . . . .	14
2.4.2	RS-485 . . . . .	15
2.4.3	Local Interconnection Network . . . . .	17
2.4.4	Single-Edge Nibble Transmission . . . . .	19
2.4.5	MSSP, I2C e SPI . . . . .	19
2.4.6	Comparação das Diferentes Tecnologias . . . . .	21
2.5	Aquisição e Monitorização de Parâmetros . . . . .	21
2.5.1	Temperatura . . . . .	21
2.5.2	Humidade Atmosférica . . . . .	23
2.5.3	Corrente . . . . .	25
<b>3</b>	<b>Arquitetura do Sistema</b>	<b>29</b>
3.1	Arquitetura Geral . . . . .	29
3.1.1	Arquitetura do Controlador Remoto de Controlo da Eletrobomba . . . . .	30
3.1.2	Arquitetura do Controlador Remoto dos Sensores de Sobrecarga e Pressões . . . . .	37
3.1.3	Arquitetura do Controlador Remoto de Recolha de Informação Meteorológica . . . . .	40
3.1.4	Arquitetura do Controlador Remoto do Sistema de Rega Auxiliar . . . . .	45
3.1.5	Arquitetura do controlador remoto principal . . . . .	48
3.2	Conclusão do Capítulo 3 . . . . .	52
<b>4</b>	<b>Hardware dos Controladores</b>	<b>53</b>
4.1	Hardware do Controlador Remoto da Eletrobomba . . . . .	53
4.1.1	Condicionamento de sinais digitais de tensão alternada . . . . .	54
4.1.2	Condicionamento do sinal analógico de tensão alternada . . . . .	55
4.2	Hardware do Controlador Remoto dos Sensores de Sobrecarga e Pressões . . . . .	56
4.2.1	Controlo do sensor de sobrecarga e amplificação do sinal resultante . . . . .	56
4.2.2	Teste do sensor de sobrecarga . . . . .	57
4.3	Hardware do Controlador Remoto de Recolha de Informação Meteorológica . . . . .	57
4.3.1	Sensores da radiação solar visível e ultravioleta . . . . .	58
4.3.2	Sensores de temperatura, humidade e pressão atmosférica . . . . .	58
4.3.3	Sensores de precipitação . . . . .	58
4.4	Hardware do Controlador Remoto do Sistema de Rega Auxiliar . . . . .	59
4.4.1	Relógio de tempo real . . . . .	60
4.4.2	Indicador luminoso . . . . .	60
4.4.3	Sensor de humidade do solo . . . . .	60

4.5	Hardware do Controlador Remoto Principal . . . . .	61
4.5.1	Hardware das válvulas servo-motorizadas . . . . .	62
4.6	Conclusão do Capítulo 4 . . . . .	63
<b>5</b>	<b>Software dos Controladores Remotos</b>	<b>65</b>
5.1	Software do Controlador Remoto da Eletrobomba . . . . .	65
5.1.1	Medição de valores analógicos . . . . .	66
5.1.2	Interrupção temporizada pela rede elétrica . . . . .	67
5.1.3	Deteção do estado de sinais digitais e contagem do número de arranques . . . . .	69
5.1.4	Permissão de arranque e limite do número de arranques por período . . . . .	71
5.1.5	Comunicação com o ESP8266 . . . . .	73
5.1.6	<i>Software</i> do ESP8266 . . . . .	75
5.2	Software do Controlador Remoto dos Sensores de Sobrecarga e Pressões . . . . .	78
5.2.1	Microcontrolador principal . . . . .	79
5.2.2	Microcontrolador de expansão . . . . .	81
5.3	Software do Controlador Remoto de Recolha de Informação Mete- orológica . . . . .	82
5.4	Software do Controlador Remoto do Sistema de Rega Auxiliar . . . . .	88
5.4.1	Processamento principal . . . . .	88
5.4.2	Processamento de comunicações . . . . .	93
5.5	Software do Controlador Remoto Principal . . . . .	95
5.5.1	Leitura de dados do módulo de recolha dos níveis dos re- servatórios . . . . .	96
5.5.2	Comunicação com o servidor . . . . .	98
5.6	Conclusão do Capítulo 5 . . . . .	100
<b>6</b>	<b>Implementação do Servidor</b>	<b>101</b>
6.1	Configuração do DNS, <i>router</i> e encriptação web . . . . .	102
6.1.1	Configuração do DNS . . . . .	102
6.1.2	Configuração do <i>Router</i> . . . . .	102
6.1.3	Encriptação web . . . . .	103
6.2	Configuração do Apache . . . . .	105
6.3	Configuração do <i>Broker</i> MQTT . . . . .	107
6.4	Base de Dados . . . . .	109
6.4.1	Configuração da base de dados . . . . .	109
6.4.2	Arquitetura da base de dados . . . . .	109
6.5	Software de Gestão do Sistema . . . . .	111
6.5.1	Algoritmo de Controlo Principal . . . . .	112
6.5.2	Algoritmo de Interface com a Base de Dados . . . . .	140

6.6	Menu Principal da Interface Gráfica . . . . .	144
6.6.1	Ligação ao <i>broker</i> MQTT . . . . .	144
6.6.2	Recolha de outras informações . . . . .	145
6.7	Conclusão do Capítulo 6 . . . . .	146
<b>7</b>	<b>Testes e Resultados</b>	<b>147</b>
7.1	Interface Web . . . . .	147
7.1.1	Aplicação web desenvolvida . . . . .	147
7.1.2	Configuração do Apache . . . . .	157
7.2	Notificação dos utilizadores . . . . .	158
7.3	Controladores Remotos . . . . .	159
7.3.1	Controlador Remoto da Eletrobomba . . . . .	159
7.3.2	Controlador Remoto do Sistema de Rega Auxiliar . . . . .	160
7.3.3	Interface de Configuração dos Controladores Remotos . . . . .	163
7.4	Conclusão do Capítulo 7 . . . . .	164
<b>8</b>	<b>Conclusões</b>	<b>165</b>
8.1	Trabalho Futuro . . . . .	166
	<b>Referências Bibliográficas</b>	<b>167</b>
<b>A</b>	<b>Esquema do controlador remoto da eletrobomba</b>	<b>173</b>
<b>B</b>	<b>Protótipo do PCB do controlador remoto da eletrobomba</b>	<b>175</b>
<b>C</b>	<b>Esquema do controlador remoto dos sensores de sobrecarga e pressões</b>	<b>177</b>
<b>D</b>	<b>Esquema do controlador remoto de recolha de informação meteorológica</b>	<b>179</b>
<b>E</b>	<b>Esquema do controlador remoto do sistema de rega auxiliar</b>	<b>181</b>
<b>F</b>	<b>Prototipo do PCB do controlador remoto do sistema de rega auxiliar</b>	<b>183</b>
<b>G</b>	<b>Arquitetura do hardware das válvulas servo-motorizadas</b>	<b>185</b>
<b>H</b>	<b>Prototipo do PCB das válvulas servo-motorizadas</b>	<b>187</b>

---

# Índice de Figuras

---

1.1	Consumo médio de água por pessoa por dia na união europeia[1]	1
1.2	Calendarização	3
2.1	População residente em áreas urbanas e rurais de 1950-2050 [2]	5
2.2	Arquitetura de uma rede LoRa do tipo <i>mesh</i> [3]	10
2.3	Cobertura da rede LoRa (a) Porto[4] (b) Lisboa[5]	10
2.4	Cobertura da rede Sigfox. [6]	13
2.5	Tipologias de barramentos desaconselhados (vermelho), aceitáveis (verde) e preferenciais (azul). [7]	17
2.6	Diferentes métodos de terminação de um barramento RS-485. [7]	18
2.7	Configuração minimalista de um barramento LIN. [8]	18
2.8	Sensor de humidade capacitivo. [9]	23
2.9	Sensor de humidade de estado sólido. [9]	25
2.10	Resistência de <i>shunt</i> com ligação em Kelvin. [10]	26
2.11	Diagrama de blocos do ACS712. [11]	27
3.1	Arquitetura geral do sistema	30
3.2	Diagrama de blocos do controlador remoto da eletrobomba	31
3.3	Conversão e condicionamento do sinal da corrente em tensão[12]	32
3.4	Diagrama de blocos do controlador remoto dos sensores de sobrecarga e pressões	38
3.5	Diagrama de blocos do controlador remoto de recolha de parâmetros meteorológicos	40
3.6	Desenho mecânico do sensor de precipitação[13]	44
3.7	Sensor de precipitação[14]	44
3.8	Diagrama de blocos do controlador remoto do sistema de rega auxiliar	46
3.9	Sensores de humidade do solo (a) Capacitivo (b) Resistivo[15, 16]	47
3.10	Diagrama de blocos do controlador remoto principal	48
3.11	Sensor de medição do nível da água[17]	50

4.1	Condicionamento de sinais digitais de tensão alternada . . . . .	55
4.2	Condicionamento do sinal analógico de tensão alternada . . . . .	55
4.3	Esquema do sensor de humidade do solo . . . . .	61
4.4	Arquitetura do <i>hardware</i> do controlador remoto principal . . . . .	62
4.5	Adaptador entre o eixo da válvula e do motor (a) Visto de frente (b) Visto de traseira e, (c) Motor . . . . .	63
5.1	Diagrama do código do controlador remoto da eletrobomba da interrupção do ADC . . . . .	66
5.2	Fluxograma simplificado do tratamento dos valores recolhidos pelo ADC	69
5.3	Contagem do número de arranques da eletrobomba . . . . .	70
5.4	Permissão de um novo arranque da eletrobomba . . . . .	72
5.5	Execução do arranque da eletrobomba . . . . .	73
5.6	Processamento de um comando recebido e envio de uma resposta . . .	74
5.7	Trama de resposta do controlador remoto a um comando recebido . .	74
5.8	Código de configuração do ESP8266 durante o arranque . . . . .	75
5.9	Ciclo principal do ESP8266 do controlador remoto da eletrobomba . .	76
5.10	Transmissão de dados do controlador remoto da eletrobomba para o ESP8266 . . . . .	77
5.11	Transmissão de dados do ESP8266 para o controlador remoto da eletrobomba . . . . .	78
5.12	Rotina principal do controlador remoto de sobrecarga . . . . .	80
5.13	Rotina de segurança do controlador remoto de sobrecarga . . . . .	80
5.14	Rotina principal da expansão do sistema de sobrecarga . . . . .	81
5.15	Ciclo principal do controlador remoto de recolha de informação meteorológica . . . . .	83
5.16	Recolha dos valores dos sensores da estação meteorológica . . . . .	84
5.17	Interrupção de resposta a um evento no sensor de medição de pluviosidade . . . . .	85
5.18	Algoritmo de processamento de um comando recebido pelo servidor . .	86
5.19	Algoritmo de resposta a um comando recebido pelo servidor . . . . .	87
5.20	Formato da trama de resposta enviada pelo controlador remoto da estação meteorológica . . . . .	87
5.21	Ciclo principal do controlador remoto do sistema de rega auxiliar . . .	89
5.22	Rotina executada durante uma interrupção de um <i>timer</i> interno . . .	90
5.23	Algoritmo de verificação da validade de um programa . . . . .	91
5.24	Algoritmo de decisão de ações consoante a programação semanal . . .	92
5.25	Trama de dados recebida para atualizar o sistema com o estado da ligação . . . . .	93
5.26	Trama recebida para alterar o modo de funcionamento do sistema e o estado das válvulas em modo manual . . . . .	93
5.27	Trama recebida para realizar o acerto do relógio . . . . .	94

5.28	Trama de dados recebida para alterar um dado da programação do sistema de rega . . . . .	94
5.29	Trama de dados recebida responsável por restaurar sinalização de anomalias . . . . .	94
5.30	Armazenamento de eventos originados pelos caudalímetros . . . . .	95
5.31	Trama enviada ao módulo dos sensores de níveis para pedir informação . . . . .	96
5.32	Pedido de informação ao módulo dos sensores de níveis . . . . .	97
5.33	Processamento da resposta enviada pelo módulo dos sensores de níveis . . . . .	98
5.34	Processamento de comandos enviado pelo servidor . . . . .	99
6.1	Configuração do sistema de nomes de domínio . . . . .	102
6.2	Atribuição de um endereço IP estático pelo <i>router</i> ao servidor . . . . .	103
6.3	Regras de reencaminhamento do <i>router</i> . . . . .	104
6.4	Encriptação e desencriptação assimétrica . . . . .	104
6.5	Arquitetura do tráfego de mensagens de dados . . . . .	108
6.6	Tabelas da base de dados . . . . .	111
6.7	Arquitetura da aplicação de controlo desenvolvida . . . . .	113
6.8	Rotina de inicialização do sistema . . . . .	114
6.9	Primeira parte do ciclo infinito principal . . . . .	114
6.10	Segunda parte do ciclo infinito principal . . . . .	115
6.11	Ciclo que determina o número de clientes ligados . . . . .	116
6.12	Alteração de valores inválidos por valores estimados . . . . .	119
6.13	Separação dos dados em secções contínuas . . . . .	121
6.14	Serviço de controlo do controlador remoto da eletrobomba . . . . .	123
6.15	Algoritmo de resposta a mensagens enviadas pelo controlador remoto da eletrobomba . . . . .	124
6.16	Processamento de comandos enviados pela aplicação web (a) Que alteram a programação dos temporizadores (b) Que alteram o estado de funcionamento do sistema e das válvulas em modo manual . . . . .	125
6.17	Serviço de recolha de informação do controlador remoto do sistema de rega auxiliar . . . . .	126
6.18	Serviço de processamento de informação do controlador remoto de recolha de informação meteorológica . . . . .	127
6.19	Representação da função de temporização dos serviços de escrita na base de dados . . . . .	128
6.20	Diferenciador principal de decisão de execução de rotinas . . . . .	129
6.21	Rotina executada com o sensor no modo automático . . . . .	131
6.22	Primeira parte da rotina executada com o sensor no modo de teste . . . . .	132
6.23	Segunda parte da rotina executada com o sensor no modo de teste . . . . .	133
6.24	Serviço de controlo do modo de operação do sistema . . . . .	134
6.25	Serviço de alteração do modo de operação do sistema . . . . .	135

6.26	Serviço de registo de parâmetros para realizar uma atuação temporizada nas válvulas . . . . .	137
6.27	Serviço de ação para realizar uma atuação temporizada nas válvulas . . . . .	138
6.28	Serviço de decisão de prioridades . . . . .	139
6.29	Exemplo de serviço de adição de dados a uma tabela . . . . .	142
6.30	Serviço que guarda dados provenientes do IPMA . . . . .	143
7.1	Página principal da aplicação web . . . . .	148
7.2	Informação sobre o controlador remoto da eletrobomba . . . . .	149
7.3	Informação sobre o controlador remoto do sistema de rega auxiliar . . . . .	150
7.4	Informação sobre o controlador remoto de recolha de informação meteorológica . . . . .	151
7.5	Apresentação dos caudais das fontes, consumos e balanço . . . . .	151
7.6	Origem do volume de água consumida nos últimos 7 dias (a) De forma geral (b) Apresentação diária por fonte . . . . .	152
7.7	Gráficos retornados pelos algoritmos que calculam os caudais das fontes de água . . . . .	153
7.8	Gráficos do volume de água gasta e balanço diário . . . . .	153
7.9	Gráficos com o histórico (a) Do balanço do volume de água (b) Da variação do nível do reservatório superior . . . . .	154
7.10	Volume de água gasto para rega e nível de água em reserva . . . . .	154
7.11	Caudais das minas e do furo e precipitação medida . . . . .	155
7.12	Precipitação medida pelo sensor implementado e precipitação determinada pelo IPMA . . . . .	155
7.13	Variação do volume do reservatório principal e do furo . . . . .	156
7.14	Visualização dos clientes conectados ao servidor . . . . .	157
7.15	Resposta do servidor (a) Com listagem ativada (b) Com listagem desativada . . . . .	158
7.16	Notificação dos utilizadores (a) Por Email (b) Por SMS . . . . .	159
7.17	Menu principal do controlador remoto da eletrobomba . . . . .	160
7.18	Menus do controlador remoto da eletrobomba (a) Primeira página (b) Segunda página . . . . .	160
7.19	Menu principal do controlador remoto do sistema de rega auxiliar . . . . .	161
7.20	Menus do controlador do sistema de rega auxiliar (a) Configuração do relógio (b) Configuração do modo de operação . . . . .	162
7.21	Configuração dos programas de rega do controlador remoto . . . . .	162
7.22	Interface de configuração dos ESP8266 . . . . .	163

---

# Índice de Tabelas

---

2.1	Riscos das cidades inteligentes . . . . .	8
2.2	Tecnologias de comunicação sem fios . . . . .	14
2.3	Tecnologias de comunicação com fios . . . . .	21
3.1	Tabela de estados dos sinais da eletrobomba . . . . .	31
3.2	Características de candidatos para a medição da corrente[11, 12] . . . . .	33
3.3	Preço e disponibilidade de candidatos para a medição da corrente[18, 19, 20, 21] . . . . .	34
3.4	Características dos microcontroladores de 8 bits[22, 23] . . . . .	34
3.5	Características dos microcontroladores de 32 bits[24, 25] . . . . .	35
3.6	Preço e disponibilidade dos microcontroladores[18, 19, 20, 21] . . . . .	35
3.7	Características das tecnologias de comunicação sem fios[26, 27] . . . . .	36
3.8	Preço e disponibilidade das tecnologias de comunicação sem fios[18, 19, 20, 21, 28] . . . . .	36
3.9	Características dos microcontroladores apresentados[29, 30] . . . . .	38
3.9	Características dos microcontroladores apresentados[29, 30] . . . . .	39
3.10	Preço e disponibilidade dos microcontroladores em diferentes fornecedores[18, 19, 20, 21] . . . . .	39
3.11	Características de candidatos para a medição da temperatura e humidade [31, 32, 33, 34] . . . . .	41
3.12	Preço e disponibilidade de candidatos para a medição da temperatura e humidade [18, 19, 20, 21] . . . . .	41
3.13	Características de candidatos para a medição da pressão atmosférica [35, 36] . . . . .	42
3.14	Preço e disponibilidade de candidatos para a medição da pressão atmosférica [18, 19, 20, 21] . . . . .	42
3.15	Sensores utilizados para a medição da intensidade solar visível e UV[37, 38] . . . . .	43
3.16	Sensores utilizados para medir o nível da humidade do solo[39, 40, 41] . . . . .	47
3.17	Métodos possíveis para medir o nível da água no furo[42] . . . . .	50

3.18	Métodos possíveis para controlar a passagem da água[43, 44, 45, 46]	51
5.1	Estado do sensor de sobrecarga e da eletroválvula consoante o valor dos bits de controlo	79
6.1	Modos de operação do sistema	112
6.2	Custos associados à Twilio	117
6.3	Possíveis mensagens enviadas para o controlador remoto do sistema de rega auxiliar	124
6.3	Possíveis mensagens enviadas para o controlador remoto do sistema de rega auxiliar	125

---

# Índice de Listas

---

6.1	Configuração do <i>VirtualHost</i> HTTP . . . . .	105
6.2	Configuração do <i>VirtualHost</i> HTTPS . . . . .	106
6.3	Ficheiro <i>htaccess</i> . . . . .	106
6.4	Ficheiro de configuração do <i>broker</i> MQTT . . . . .	108
6.5	Exemplo de ficheiro de controlo de acessos do <i>broker</i> MQTT . . . . .	109
6.6	Configuração do MySQL . . . . .	109
6.7	Resumo da função utilizada para verificar se podem ser enviadas tramas para a aplicação web . . . . .	116
6.8	Função utilizada para determinar a taxa de enchimento do reser- vatório e o caudal da fonte . . . . .	120
6.9	<i>Query</i> SQL de análise dos hábitos semanais . . . . .	144
6.10	Declaração de um cliente . . . . .	145
6.11	<i>Callback</i> de uma função após uma ligação . . . . .	145
6.12	Recolha de informação apresentada nos gráficos . . . . .	146



---

# Acrónimos

---

Acrónimo	Descrição	Página
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks	11
ACME	Automatic Certificate Management Environment	105
ADC	Analog to Digital Converter	32
CAN	Controller Area Network	14
CRC	Cyclic Redundancy Check	73
CSMA	Carrier Sense Multiple Access	13
CSV	Comma-Separated Values	142
DNS	Domain Name System	102
EEPROM	Electrically Erasable Programmable Read-Only Memory	59
FSK	Frequency shifting key	9
GSM	Global System for Mobile Communications	11
HTTP	Hypertext Transfer Protocol	101
I2C	Inter-Integrated Circuit	19
ICANN	Internet Corporation for Assigned Names and Numbers	102
ICTs	Information and Communication Technologies	6
IoT	Internet of Things	9
IP	Internet Protocol	102
IPMA	Instituto Português do Mar e da Atmosfera	110
ISP	Internet Service Provider	102
ISRG	Internet Security Research Group	105
LAN	Local Area Network	11
LCD	Liquid-Crystal Display	19
LIN	Local Interconnection Network	17
LoRa	Long Range	9
LoRaWAN	Long Range Wide Area Network	9
LPUART	Low Power Universal Asynchronous Receiver-Transmitter	35
MAC	Media Access Control	103

Acrónimo	Descrição	Página
MOS	Metal Oxide Semiconductor	22
MQTT	Message Queuing Telemetry Transport	65
MSSP	Master Synchronous Serial Port	19
NB-IoT	Narrowband Internet of things	13
NRZ	Non Return to Zero	15
NTC	Negative Temperature Coefficient	22
PHP	Hypertext Preprocessor	105
PTC	Positive Temperature Coefficient	22
RMS	Root Mean Square	67
RTC	Real Time Clock	45
RTD	Resistance Temperature Detectors	22
SDIO	Secure Digital Input/Output Interface	35
SENT	Single-Edge Nibble Transmission	19
SMS	Short Message Service	113
SPI	Serial Peripheral Interface	19
SQL	Structured Query Language	109
SSL	Secure Sockets Layer	106
SVG	Scalable Vector Graphics	144
TBJ	Transistor Bipolar de Junção	22
TCP	Transmission Control Protocol	103
TLS	Transport Layer Security	106
UART	Universal Asynchronous Receiver-Transmitter	16
UNB	Ultra Narrow Band	12
URL	Uniform Resource Locator	105
USART	Universal Synchronous and Asynchronous Receiver-Transmitter	34
WEP	Wired Equivalent Privacy	11
WiFi	Wireless Fidelity	10

# Capítulo 1

## Introdução

### 1.1 Contextualização

Como está demonstrado na figura 1.1, Portugal é um dos países com o valor médio de consumo de água mais elevado na união europeia, surgindo em 9º lugar. Recentemente o país tem ultrapassado uma crise de seca prolongada que remota à perda de água por parte dos solos a períodos superiores a 12 meses o que obriga a que a população mude as práticas de consumo dos recursos hídricos, para garantir um equilíbrio e a sustentabilidade.



Figura 1.1: Consumo médio de água por pessoa por dia na união europeia[1]

## 1.2 Objetivos

Este projeto desenvolvido visa implementar conectividade de diversos sistemas eletrônicos que compõe o circuito hidráulico de abastecimento de uma residência, centralizando a sua interação através de uma aplicação web desenvolvida que forneça os serviços sobre a rede de Internet.

Devido à habitação não ter possibilidade de realizar uma ligação à rede de abastecimento de água, obtém os recursos hídricos maioritariamente de duas fontes distintas, umas minas e um furo. Atualmente o presente sistema contém bastantes inconveniências tais como a necessidade de precisar de bastante mão de obra e elevadíssima atenção necessária durante a sua operação.

Com a presente seca, as nascentes que alimentam a residência fornecem cada vez mais um volume de água menor por dia, pelo que atualmente é necessário adotar novos hábitos diários que não comprometam a sustentabilidade do sistema.

O objetivo principal deste projeto é criar um serviço que forneça assistência aos residentes, avaliando os gastos diários e demarcando um balanço entre o volume de água consumida diariamente e os volumes disponíveis. Pretende-se que este valor ajude a determinar a viabilidade das ações ou a necessidade de ajustar as políticas de poupança.

Outra necessidade urgente é a implementação de um sistema que determine se está a ocorrer alguma sobrecarga durante o processo de enchimento dos reservatórios, visto esta advertência ser recorrente sobretudo durante a operação manual do sistema, sendo que o desperdício de água tem de ser mínimo visto ter um impacto grande no total de água disponível.

Pretende-se também implementar um serviço que centralize informação recolhida por diversos sensores espalhados pela habitação, fornecendo informação ao utilizador sobre variáveis relacionadas com o ambiente ou com o sistema hidráulico. Para além da centralização da informação, deve ser agrupado e implementado atuação dos diferentes subsistemas, permitindo um controlo simplificado e até mesmo autónomo do sistema hidráulico.

A implementação dos subsistemas devem estar preparados para aceitar desenvolvimentos futuros, recolhendo-se e armazenando-se toda a informação recolhida.

## 1.3 Calendarização

Este projeto dividiu-se em seis fases diferentes, sendo estas: avaliação de requisitos, elaboração do estado da arte, implementação dos controladores, implementação do servidor, testes e resultados e elaboração da tese, conforme pode ser observado na figura 1.2.

Tarefa	2021			2022									
	Out.	Nov.	Dez.	Jan.	Feb.	Mar.	Abr.	Mai.	Jun.	Jul.	Ago.	Set.	Out.
Avalliação de requisitos													
Elaboração do estado da arte													
Implementação dos controladores													
Implementação do servidor													
Testes e resultados													
Elaboração da tese													

Figura 1.2: Calendarização

## 1.4 Organização do Relatório

O relatório é constituído por oito capítulos, que descrevem a projecção e implementação do projeto.

O primeiro capítulo apresenta as motivações de realização deste projeto, os principais objetivos delineados, apresentado-se a calendarização e duração de cada etapa.

No segundo capítulo é abordado o estado da arte de algumas tecnologias presentes no projeto, sendo efetuado um estado de conceitos relacionadas com a IoT (*Internet of Things*), as tecnologias de comunicação e de monitorização de parâmetros.

No terceiro capítulo é efetuada a apresentação da arquitetura do sistema, sendo efetuada a descrição do *hardware* a ser implementado, sendo apresentada a arquitetura geral do sistema.

O quarto capítulo aborda a projecção do *hardware* dos módulos que realizam a monitorização e controlo do sistema, sendo que o desenvolvimento do *software* de cada dispositivo desenvolvido foi descrito no capítulo cinco.

O capítulo seis aborda o desenvolvimento da máquina central do projeto, que concentra a informação da implementação global e controla os dispositivos subordinados, fornecendo também outros serviços úteis sobretudo ao utilizador.

No sétimo capítulo apresentam-se os resultados obtidos através da realização de testes de funcionamento do sistema.

O oitavo capítulo apresenta as considerações finais e os próximos passos a efetuar num trabalho futuro.



## Capítulo 2

---

# Estado da Arte

---

### 2.1 Panorama Atual

Em 2007, um recorde foi atingido, uma vez que, globalmente, passaram a existir mais pessoas a viver em zonas urbanas do que em zonas rurais. Esta diferença tem vindo a aumentar cada vez mais, como mostra a figura 2.1. Em 1950, cerca de 2/3 da população mundial vivia em zonas rurais. Já em 2050, os papéis inverter-se-ão e, no futuro, 2/3 da população mundial residirá em zonas urbanas[2][47].

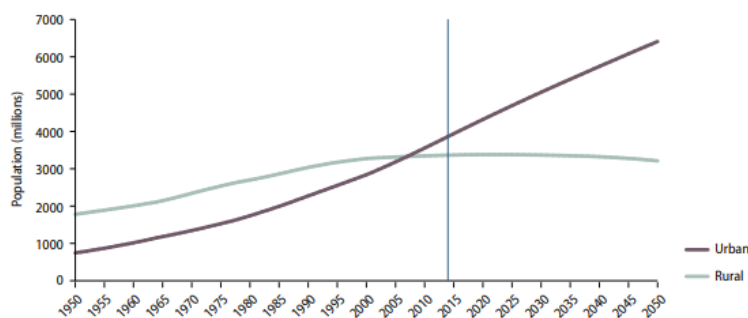


Figura 2.1: População residente em áreas urbanas e rurais de 1950-2050 [2]

Os continentes com a maior proporção de residentes em zonas urbanas e residentes em zonas rurais são: América, tendo a América do norte um índice mais elevado; seguindo-se da Europa e, por fim, a Oceânia. A África e a Ásia são os continentes que sofrem a maior transformação, no entanto, continuam a ser maioritariamente rurais[2].

Segundo as Nações Unidas, espera-se que para 2030, o número de cidades com mais de 10 milhões de habitantes suba para um total de 41, sendo este valor mais

de quatro vezes superior em relação ao ano de 1990. Esta transformação promove uma elevada quantidade de novas oportunidades, no entanto, acarreta também os seus desafios físicos, sociais, comportamentais, económicos e infraestruturais[48]. Em destaque, encontram-se os desafios ambientais, sendo as zonas urbanas as principais consumidoras de energia e a principal fonte referente à libertação de gases de efeito de estufa, como o CO<sub>2</sub>. Isto implica que, as cidades, sejam um dos primeiros sectores a dar o próximo passo[49].

## 2.2 Cidades Inteligentes

O conceito de cidade inteligente é difícil de definir, variando, consideravelmente, entre autores. A maioria das definições ronda que uma cidade inteligente envolva o uso de tecnologias de informação e comunicação (ICTs) com o intuito de melhorar o uso de energia, a mobilidade, a saúde, a gestão de recursos e, fundamentalmente, a qualidade de vida das pessoas[49][50].

As ICTs são normalmente integradas de forma estratégica com a ideia de possibilitar a recolha de informação que, posteriormente, é analisada e processada, podendo ser distribuída aos utilizadores em tempo real.

Estas tecnologias permitem também a troca de informação entre diversos dispositivos, monitorizando e melhorando o ambiente urbano relativamente a infraestruturas tais como os transportes, gestão de resíduos, balanceamento energético, entre outros[49][51].

### 2.2.1 Características de uma Cidade Inteligente

De forma a criar um termo de comparação relativamente à evolução de diferentes cidades, foram destacadas seis condições diferentes que tem como objetivo expor os pontos fortes e fracos de cada centro urbano[50].

#### **Economia Inteligente**

Com o crescimento das cidades, existe implicitamente um aumento dos recursos requeridos para acompanhar esta expansão de forma sustentável. Desta forma, é importante, por exemplo, encorajar o desenvolvimento de novos projetos com o objetivo de melhorar a economia e trocas internacionais, incentivar o empreendimento, inovação e flexibilidade.

Um dos pontos com mais destaque, é o crescimento do mercado de trabalho ligado à construção, estando-se à espera de um aumento de novas infraestruturas.

### **População Inteligente**

É fundamental existir uma população suficientemente educada que retire partido do que é oferecido por uma cidade inteligente. Assim, é necessário continuar a investir na educação tal como reduzir a dificuldade ao seu devido acesso. Como já foi dito anteriormente, a inovação e criatividade da população são pontos importantes que tem de ser encorajados de forma a reduzir o impacto das dificuldades que surjam e de forma a promover a inovação, colocando a sociedade num novo patamar.

### **Governo Inteligente**

É evidente que cada urbanização tem os seus objetivos delineados, sendo que estes tendem a diferir de caso para caso, tornando as cidades únicas. Tendo isto em conta, fatores que impliquem um governo inteligente são, por exemplo, a participação na tomada de decisão, serviços públicos e sociais, estratégias políticas e perspetivas futuras.

### **Mobilidade Inteligente**

O sistema de transportes tem um peso bastante alargado, uma vez que são responsáveis por auxiliar a movimentação de cidadãos nas cidades de forma rápida e eficiente. Com a evolução tecnológica, espera-se que haja a incorporação de métodos de transporte mais eficientes e ecologicamente menos impactantes. Para além disto, a informação sobre os sistemas de transporte deve ser disponibilizada em tempo real em plataformas que facilitem o seu acesso.

### **Impacto Ambiental**

Atualmente existe um esforço para reduzir o impacto de diversos sectores no ambiente. Vários movimentos estão a ser continuamente realizados para tal, como, por exemplo, o abandono de veículos de combustão interna tradicionais e a adoção de veículos mais “verdes”, nomeadamente veículos elétricos.

Outros fatores que reduzem o impacto ambiental da urbanização são, a redução do uso de água e de energias, uma gestão sustentável de recursos, entre outros.

Um exemplo prático ocorre em Amesterdão, onde os edifícios são arrefecidos pela água utilizada no consumo doméstico, não se verificando desperdício de energia de outras fontes, com o objetivo de atingir o mesmo fim. Outro exemplo acontecesse na mesma cidade, onde os resíduos resultantes da atividade urbana são convertidos em energia elétrica e térmica, ou em materiais de construção de alta qualidade[52].

## Qualidade de Vida

O aumento da qualidade de vida é fundamental. Isto implica o desenvolvimento de uma diversidade de sectores, tais como o sector cultural, saúde, educação, segurança, entre outros. Este desenvolvimento deve ter como objetivo promover e proporcionar a coesão social.

### 2.2.2 Riscos das cidades inteligentes

Numa cidade inteligente, a informação é recolhida por sensores e sistemas embebidos que a transmitem através de um ou vários sistemas de comunicação disponíveis atualmente. Ao longo do percurso da transmissão de informação estas comunicações costumam integrar soluções que são do tipo cablada ou sem-fios. A ciber-segurança é um dos aspetos mais críticos do desenvolvimento de um sistema robusto que permita manter a integridade dos dados, e, acima de tudo, a privacidade dos utilizadores. Para tal é importante assegurar bons meios de transporte de informação, assim como métodos seguros de encapsulamento de mensagens, tais como a encriptação por chave privada[48].

Para além dos aspetos anteriores, é importante reter que as vulnerabilidades não se restringem a ciber-ataques. Um exemplo de uma segurança deficiente é, por exemplo, certos dispositivos, ou nós, poderem ser acedidos fisicamente, tornando-os vulneráveis[48].

Contudo, ainda existem outros fatores, não maliciosos, que podem comprometer o desenvolvimento duma cidade inteligente. Alguns destes fatores encontram-se apresentados em seguida[50]:

Tabela 2.1: Riscos das cidades inteligentes

<b>Tipo de risco</b>	<b>Descrição</b>
Tecnológico	Novas tecnologias em circulação podem não atingir a performance esperada
Operacional	Desaproveitamento de uma tecnologia por falta de educação de operários
Construção	Eventual aparecimento de dificuldades ou atrasos não esperados
Mercado	Novas tecnologias podem ser um fracasso quando postas em circulação.
Político	Alterações na legislação ou regulamentos podem levar a inviabilidade de uma tecnologia

## 2.3 Tecnologias de Comunicação Sem Fios

Existem no mercado diversas tecnologias sem fios de elevado alcance, tecnologias estas capazes de fornecer conectividade a um sistema de Internet das coisas (IoT) integrante no âmbito das cidades inteligentes, como é o caso do presente projeto. Para tal, foi realizado um estudo com o intuito de expor as diferentes características de algumas redes sem fios.

### 2.3.1 Long Range

LoRa, acrónimo do inglês *Long Range* é uma rede sem fios conhecida fundamentalmente pelo seu baixo consumo e longo alcance. Para realizar a transmissão de informação na camada física, é utilizado um método vulgarmente utilizado pela indústria militar e aeroespacial nas comunicações de longa distancia. Este método é conhecido como *chirp spread spectrum*. Este tipo de comunicação tem características semelhantes, em termos de consumo, ao vulgar *frequency shifting key* (FSK) que é normalmente utilizado por dispositivos sem fios convencionais. No entanto, o *chirp spread spectrum* demonstra desempenhos superiores em termos de alcance[3].

O LoRa transmite dados com uma frequência do sinal portador situada na banda dos 868 MHz e com uma largura de banda que ronda os 125 kHz ou os 250 kHz. Outras características técnicas importantes são, por exemplo, o facto do LoRa permitir no máximo a transmissão de 243 bytes por pacote, realizar as transmissões a uma taxa máxima de 50 kbps e ter um alcance máximo que ronda os 20 km nas zonas rurais ou os 5 km nas zonas urbanas[53].

O protocolo que efetua o controlo e gestão da rede LoRa denomina-se LoRaWAN (*Long Range Wide Area Network*). Este protocolo tem uma elevada importância no controlo e gestão de diversas variáveis tais como: os consumos de cada nó, a qualidade do serviço, a capacidade, a abrangência e a segurança da uma rede. Muitas das redes desenvolvidas implementam a tipologia *mesh* sendo que cada nó liga a um conjunto de outros nós permitindo, naturalmente, estender o alcance da rede. No entanto, passa a haver um consumo energético superior, visto que cada nó pode retransmitir pacotes que lhe são alheios. Outras desvantagens são, a redução da capacidade da rede e aumento elevado da complexidade. Quando o alcance é assegurado, é preferível utilizar redes em estrela. A interface com outros serviços, tais como, a *cloud* ou outros tipos de servidores, é realizada por *gateways*. Cabe ao serviço efetuar a eliminação de pacotes repetidos e proceder à realização de testes que assegurem a integridade dos dados. Na figura 2.2 está representada a arquitetura comum de uma rede LoRa[3].

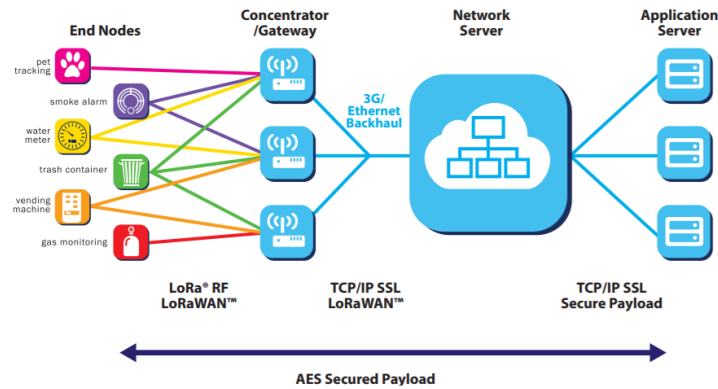


Figura 2.2: Arquitetura de uma rede LoRa do tipo *mesh* [3]

Na figura 2.3 encontra-se representada a cobertura da rede LoRa nas cidades do Porto e de Lisboa, sendo que no Porto é possível aceder a 6 *gateways*, em Lisboa este número sobe para 14. O acesso aos serviços é livre, no entanto, não se conhecem as entidades que os disponibilizam.

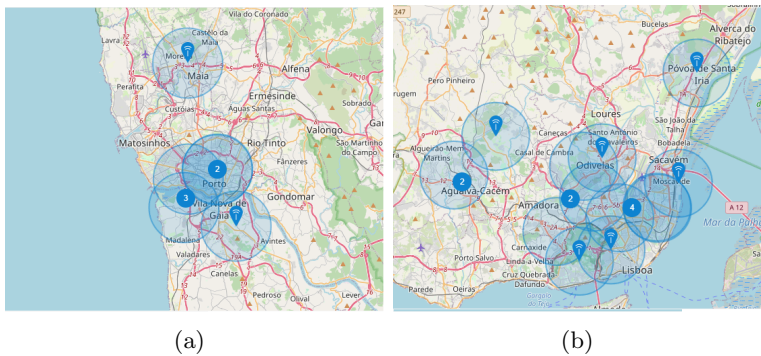


Figura 2.3: Cobertura da rede LoRa (a) Porto[4] (b) Lisboa[5]

### 2.3.2 Wireless Fidelity

*Wireless Fidelity*, tradicionalmente conhecida como WiFi é a rede local sem fios mais usada a nível mundial e baseia-se nos protocolos IEEE 802.11a/b/g/n. Existem algumas variações entre as diferentes versões, por exemplo, a versão 802.11n oferece a maior taxa de comunicação, no entanto, tem um consumo energético bastante elevado. Observando esse fator com um ponto de vista sobre o presente trabalho, esse fenómeno condiciona a sua escolha para produtos IoT que tenham uma bateria como fonte de energia. Para dispositivos no âmbito da Internet das Coisas, as variantes b ou g são as mais aconselhadas visto terem um consumo mais reduzido[54].

Alguns requisitos padrão apresentados pela rede são[55]:

- Facilidade no uso. O WiFi foi desenvolvido com o objectivo de acrescentar funcionalidades a redes *LANs* cabladas já existentes, adicionando o mínimo de inconveniências possíveis;
- Segurança. Naturalmente que uma rede sem fios tem notoriamente uma fraca segurança durante a transmissão de informação, visto que pode ser recebida facilmente por um dispositivo que se situe na vizinhança. A vulnerabilidade da rede foi reduzida substancialmente com a introdução do WEP *Wired Equivalent Privacy* que trouxe encriptação às comunicações;
- Mobilidade. Existem vários factores que contribuem para a mobilidade sendo que a interoperabilidade entre marcas é claramente fundamental;
- Manutenção da rede. Monitorizar o tráfego de um ponto de acesso, a saúde do dispositivo e as condições do meio de transmissão, são factores importantes de modo a que a transmissão de um sinal seja realizado de modo a que se possa transmitir com a maior potência possível, ainda que de modo a que não ocorra interferências com equipamentos vizinhos.

As comunicações costumam ser realizadas com uma frequência do sinal portador na ordem dos 2.4 GHz e 5 GHz e têm um alcance não superior a 50 m. As taxas de transmissão são largamente dependentes de diversos fatores e costumam variar de 54 a 600 Mbps[54][56].

Outra tecnologia que se encontra em ascensão é o IPv6 *over Low-Power Wireless Personal Area Networks* (6LoWPAN). Esta rede fornece uma ligação de uso geral, simples, de baixo custo e baixo consumo. No entanto, ainda não se encontra disponível com frequência em diversos locais. O alcance da rede costuma variar de 10 a 100 m e a taxa de transmissão não excede os 250 kbps[56][57].

A organização que gere a rede, WiFi Alliance, apresentou recentemente um novo padrão conhecido como IEEE 802.11ah ou WiFi HaLow dedicado exclusivamente para tecnologias IoT. Esta tecnologia permite realizar comunicações em faixas que rondam os 900 MHz, possibilitando alcances de até 1 km, baixos consumos energéticos o que permite ser alimentado por baterias e, não requer *hubs* especiais.[56].

### 2.3.3 Rede Celular

Uma rede móvel bastante famosa é o sistema global para comunicações móveis (GSM). Este sistema apareceu na Finlândia em 1991 e opera em diferentes bandas do espectro, sendo estas 850 MHz, 900 MHz, 1800 MHz e 1900 MHz. Esta rede

tem taxas de transmissão modestas, desde os 35 kbps a 384 kbps com alcances de cerca de 35 km[58].

Posteriormente, com a evolução tecnológica, apareceram outros sistemas semelhantes ao GSM, sendo estes o 3G e 4G. Estas tecnologias tem um alcance e taxas de transmissão de dados consideravelmente mais elevadas, sendo que conseguem transmitir dados a uma velocidade que pode variar de 384 kbps a 10 Mbps com um alcance de até 200 km.

Actualmente, estão a aparecer as primeiras implementações do 5G que trás vantagens como latências de comunicações extremamente reduzidas que rondam valores inferiores a 1 ms. Esta tecnologia vem acompanhada também de um aumento na taxa de transmissão de dados que é de 10 a 100 vezes superior em relação à tecnologia antecessora[59].

Um ponto negativo da rede celular é o elevado consumo energético, que condiciona a sua aplicação a sistemas que operem a bateria. Em contra partida, a cobertura a nível mundial da rede móvel é bastante alargada[60].

#### 2.3.4 Sigfox

Em 2010, Ludovic Le Moan e Christophe Fourtet fundaram a rede Sigfox. Ambos tinham como objetivo criar uma rede IoT capaz de interligar objectos do mundo físico ao digital. Actualmente, a rede encontra-se espalhada por 75 países, cobrindo em torno de 1.6 mil milhões de pessoas, existindo um total de 79 milhões de mensagens a circular na rede a cada dia. A tecnologia é conhecida pelo seu baixo consumo e baixa taxa de transmissão. Em termos de alcance ronda os 10 km em zonas urbanas e 40 km em zonas rurais, dependendo, claro, da topografia.[6][53].

Esta rede opera na banda publica dos 868 Mhz, utilizando apenas 192 kHz desta banda para transmitir mensagens. A modulação é conhecida como sendo de banda ultra estreita (UNB), sendo que cada mensagem é transmitida com uma largura de 100 Hz a uma taxa de 100 a 600 bits por segundo. Para garantir uma elevada qualidade de transmissão, cada nó transmite um pacote três vezes em tempos e frequências diferentes realizando um acesso ao meio de forma aleatória[61].

Na figura 2.4, é possível encontrar um mapa onde estão representados os locais em que o Sigfox se encontra em implementação, estando estes destacados na cor roxa. Também é possível observar os locais onde já existe um serviço disponível, estando estes destacados com a cor azul.

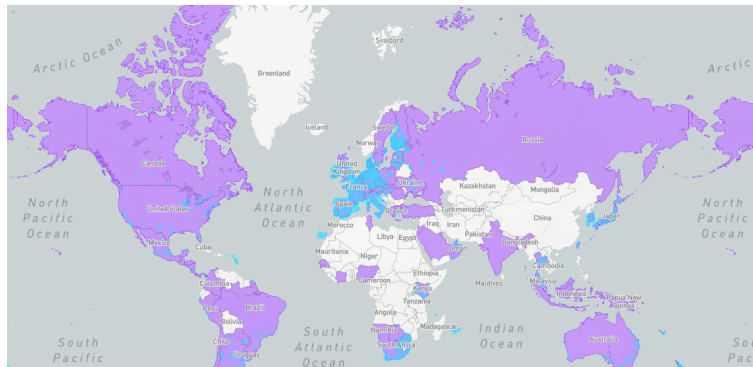


Figura 2.4: Cobertura da rede Sigfox. [6]

### 2.3.5 ZigBee

ZigBee é uma rede de comunicação sem fios que opera na faixa dos 868 MHz ou 2.4 GHz. É uma tecnologia de baixo consumo energético e possui uma elevada imunidade a ruídos e capacidade de construir uma rede de forma automática, que suporta até 65536 nós individuais. O facto da rede ser de construção automática facilita imenso a introdução de nós na rede e aumenta drasticamente a sua robustez, uma vez que tem a capacidade de se auto-regenerar em caso de avaria ou remoção de um dispositivo, tornando-a bastante fiável[58].

Com o intuito de evitar colisões nas transmissões dos pacotes, é utilizado o protocolo *Carrier Sense Multiple Access* (CSMA), que consiste na escuta do meio de transmissão e consecutivo envio caso este meio não esteja ocupado. Em termos de tamanho da trama, cada transmissão realizada pelo ZigBee tem um tamanho máximo a 128 Bytes. O alcance máximo desta tecnologia ronda os 140 metros e tem uma taxa de transmissão de 250 kbps[58].

### 2.3.6 Narrowband Internet of Things

Narrowband Internet of things (NB-IoT) é uma rede sem fios concebida especialmente para fornecer conectividade a equipamentos IoT. Esta tecnologia opera na mesma banda que a rede 4G e ocupa um largura de banda de 200 kHz. A transmissão pode ser realizada em blocos utilizados pela rede 4G, ou na faixa de guarda da rede anteriormente mencionada. As mensagens são limitadas a um máximo de 200 kbps de recepção e 20 kbps de envio, cada mensagem não pode conter mais de 1600 bytes[53].

Uma das grandes vantagens da rede é o facto de poder ser construída sobre a infraestrutura que dá suporte ao 4G, sendo apenas necessário a alteração do seu software de gestão. O seu alcance característico não costuma alcançar valores superiores a 1 km nas zonas urbanas e 10 km nas zonas rurais[53].

Pelo facto de operar na banda utilizada pela rede 4G, a tecnologia NB-IoT está suscetível a interferências.

### 2.3.7 Comparação das Diferentes Tecnologias

Na tabela 2.2 são apresentadas as características técnicas mais importantes das tecnologias anteriormente descritas, sendo estas a taxa de transmissão e o seu alcance máximo. Além disso, é apresentada também uma desvantagem de cada sistema de comunicação.

Tabela 2.2: Tecnologias de comunicação sem fios

Tecnologia	Alcance	Velocidade	Desvantagens
LoRa	5-20 km	50 kbps	Baixa taxa de transmissão
WiFi	50 m	54-600 Mbps	Elevado consumo energético
6LoWPAN	10-100 m	250 kbps	Rede raramente encontrada
GSM	35 km	35-384 kbps	Elevado consumo energético
SigFox	10-40 km	100-600 bps	Número de mensagens limitado
ZigBee	140 m	250 kbps	Curto alcance
NB-IoT	1-10 km	20-200 kbps	Vulnerável a interferências

## 2.4 Tecnologias de Comunicação Com Fios

Com o intuito de transmitir os dados dos diferentes módulos desenvolvidos, é requerida a implementação de uma comunicação com fios que interligue os diferentes subsistemas.

### 2.4.1 Controller Area Network

O protocolo *Controller Area Network* (CAN) foi originalmente desenvolvido pela Bosch, com o intuito de fornecer conectividade a diferentes sensores implementados num automóvel no entanto, cresceu rapidamente, abrangendo hoje diversos sectores, desde o medico, industrial, entre outros.

Esta tecnologia é do tipo determinística e implementa as seguintes capacidades[8]:

- Existência de prioridades nas mensagens;
- Sincronismo orientado ao bit;

- Limite de 64 nós por rede;
- Sistema *multimaster* que suporta *multicast*;
- Detecção, sinalização e retransmissão automática de mensagens;
- Detecção de avaria permanente em nós e consecutivo isolamento destes.

Pelo facto de na camada física ser implementada uma codificação dos dados do tipo *Non Return to Zero* (NRZ), existe um mínimo possível de transições do estado lógico do barramento, resultando numa imunidade ao ruído bastante boa. Como a codificação dos bits no meio físico são do tipo dominante (estado lógico 0) e recessivo (estado lógico 1), existe a possibilidade de implementar um sistema de deteção de colisões. Tendo em conta que o bit dominante prevalece sempre, na eventualidade de dois módulos tentarem transmitir informação em simultâneo, no momento em que sejam enviados bits com o estados diferentes, o módulo que enviou o bit correspondente ao estado lógico recessivo, lê, do barramento, o estado lógico dominante, detetando, assim, uma colisão[62].

Existe um grande número de velocidades numa rede CAN. Estas variam tipicamente entre: 5 kbps, 10 kbps, 20 kbps, 50 kbps, 125 kbps, 250 kbps, 500 kbps, 800 kbps e 1 Mbps. A distância da linha de transmissão depende essencialmente da velocidade de operação da rede, e pode variar desde 1000 m a uma velocidade máxima de 40 kbps, até 40 m a uma velocidade máxima de 1 Mbps. O número máximo de dispositivos interligados por barramento é de 127[8].

Normalmente podem ser encontrados quatro tipos de tramas diferentes em circulação numa rede CAN, estando estas apresentadas em seguida[62]:

- Trama de dados - contem informação convencional, transmitida entre os diferentes dispositivos;
- Trama de pedidos - utilizado para requerer dados a um dispositivo;
- Trama de erro - utilizada caso algum nó detete erros;
- Trama de *overflow* - utilizada para realizar o controlo do fluxo de dados na rede, pedindo um atraso adicional antes de serem enviadas tramas de dados ou pedidos.

### 2.4.2 RS-485

O RS-485 é uma tecnologia bastante utilizada conhecida pela sua simplicidade e fiabilidade. Este meio de comunicação é do tipo *half-duplex*, ou seja, permite comunicações bidirecionais, mas nunca em simultâneo. É importante realçar que o standard que especifica esta tecnologia, o TIA/EIA-485-A, descreve apenas

as características técnicas do meio de transmissão, nunca especificando nada sobre o protocolo de comunicação em si, sendo da responsabilidade do utilizador implementar o protocolo de comunicação que achar mais apropriado. Encontrase frequentemente associado ao RS-485, o protocolo de comunicação *Universal Asynchronous Receiver-Transmitter* (UART) para transmitir a informação, embora outros protocolos possam ser utilizados[7].

Esta tecnologia transmite a informação sobre um par entrançado, de forma diferencial, existindo portanto uma imunidade ao ruído excelente. Para além dos dois condutores utilizados na transmissão, costuma-se utilizar um terceiro para estabelecer o potencial de referência entre os diversos dispositivos. Este condutor tem como objetivo reduzir as tensões em modo comum que são prejudiciais à boa qualidade do barramento[7].

O RS-485 permite realizar comunicações a distâncias que se estendem até aos 1200 m, sendo que não devem de ultrapassar os 100 kbps. A taxa de transferência de dados máxima é de 10 Mbps com um barramento que se estende por uma distância inferior a 20 m. É importante salientar que ambos os parâmetros anteriormente mencionados são fortemente influenciados pela qualidade do barramento, ruídos externos, entre outros[7].

Esta tecnologia permite uma arquitetura *multidrop* isto é, existe a possibilidade de serem interligados vários nós numa rede, embora, o número de nós não deve ser superior a 32.

A figura 2.5 demonstra uma diversidade de barramentos frequentemente utilizada em diversas redes. Na rede em causa, tipologias como em anel, estrela e barramento/estrela, devem ser evitadas pois costumam ser de difícil mitigação de reflexões de sinais. Arquiteturas semelhantes à do barramento com longas derivações são aceitáveis mas requerem especial atenção. A tipologia que deve ser preferencialmente escolhida é a *daisy chain*, visto ser fácil de implementar pois problemas como a reflexão de sinais são de simples controlo[7].

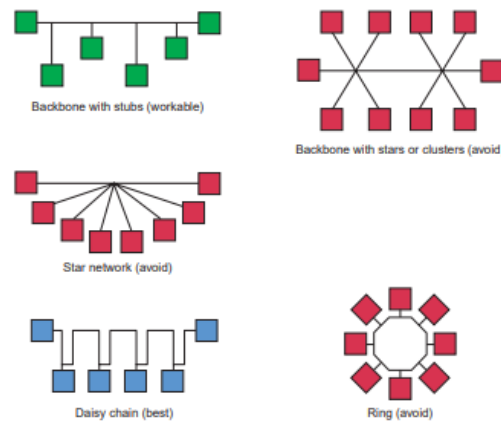


Figura 2.5: Tipologias de barramentos desaconselhados (vermelho), aceitáveis (verde) e preferenciais (azul). [7]

Caso o barramento de dados seja de elevada dimensão podem começar a surgir problemas, tais como a distorção do sinal, o que implica perda de fiabilidade da comunicação. Na figura 2.6, encontram-se representados diversos métodos de implementação de um barramento[63].

O primeiro método consiste em linhas sem terminadores, costumam ser de baixo custo e de baixo consumo energético. No entanto, têm a desvantagem de estarem limitadas a uma velocidade de transmissão muito reduzida.

O segundo método consiste em linhas com terminação paralela, são características pelo facto do terminador ser colocado no ponto mais distante possível do transmissor, no entanto implicam que a linha opere no modo unidirecional.

O terceiro método consiste em linhas com terminação em ambas as extremidades. É o método que mais energia usa, mas, é também o método mais fiável pois oferece a melhor integridade do sinal.

O quarto método é bastante mais complexo pois, faz uso de um condensador para bloquear correntes contínuas com o intuito de reduzir o consumo energético.

### 2.4.3 Local Interconnection Network

O protocolo *Local Interconnection Network* (LIN), encontra-se frequentemente em aplicações incorporantes do ramo automotivo. O Lin apresenta uma arquitetura bastante simples e de muito baixo custo, podendo ser executada por qualquer microcontrolador que implemente a UART. Para tal, é necessário que seja realizada uma implementação semelhante à da figura 2.7[8].

Após uma breve observação da figura 2.7, que representa uma típica implementação do protocolo LIN, chega-se à conclusão que o sistema só permite

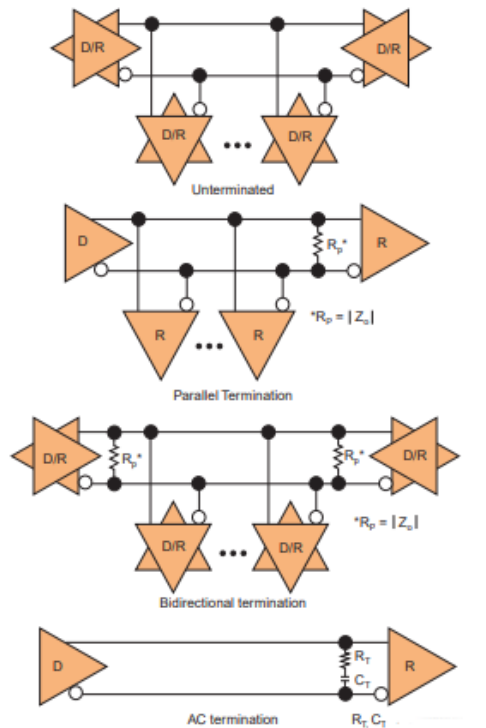


Figura 2.6: Diferentes métodos de terminação de um barramento RS-485. [7]

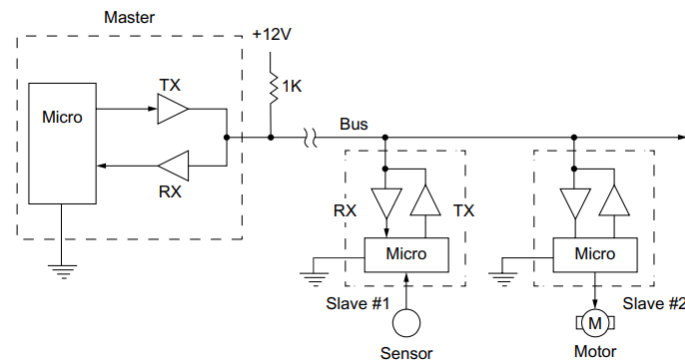


Figura 2.7: Configuração minimalista de um barramento LIN. [8]

comunicações *half-duplex*, visto que o meio de transmissão de dados é compartilhado, tanto para o envio como para a recepção de informação. Pode-se observar também que deve ser implementado um circuito que permita realizar o condicionamento de sinal entre o barramento e o microcontrolador, como sugere a imagem, provavelmente porque as tensões de funcionamento do barramento e de um microcontrolador convencional costumam ser de gamas diferentes, não sendo estas

compatíveis.

As características chave deste protocolo encontram-se apresentadas em seguida[8]:

- Comunicação do tipo *Master-Slave*;
- Pode conter até cerca de 16 *slaves*;
- Requer unicamente um condutor e um sinal de referência;
- Velocidade típica de 2400 bps, 9600 bps e, mais frequentemente 19.2 kbps.
- Alcance máximo de cerca de 40 m a 19.2 kbps.

#### 2.4.4 Single-Edge Nibble Transmission

O protocolo *Single-Edge Nibble Transmission* (SENT) foi desenvolvido com o objetivo de proporcionar um meio de transporte de informação de muito baixo custo. É um sistema ponto-a-ponto baseado em ciclos descendentes sucessivos que codificam a informação fazendo variar a largura de um impulso, resultando na codificação de um conjunto de quatro bits conhecidos como *nibble*[64].

Cada transmissão é iniciada enviando um sinal de sincronismo seguindo-se de um campo com informação sobre o estado do dispositivo. O último conjunto de dados a serem enviados consiste num valor utilizado para a verificação da integridade dos dados. Cada conjunto de quatro bits, demora pelo menos quinze milissegundos a ser transmitido[65], pelo facto de a largura do impulso ser variável, não existe uma taxa de transmissão fixa.

#### 2.4.5 MSSP, I2C e SPI

O *Master Synchronous Serial Port* (MSSP) é um módulo normalmente integrado nos microcontroladores da Microchip e pode ser encontrado, por exemplo, em modelos pertencentes à família PIC24F. Este módulo implementa uma interface série de 8 bits frequentemente utilizada para comunicar com outros microcontroladores, controladores LCD, conversores, entre outros.

Este módulo por si só não realiza comunicações, mas permite implementar duas interfaces de comunicação bastante conhecidas e frequentemente encontradas em diversos microcontroladores, sendo estas o *Inter-Integrated Circuit* (I2C), e o *Serial Peripheral Interface* (SPI)[66].

## I2C

A interface I2C consiste num sistema de comunicação *full-duplex*, isto é, permite comunicações em dois sentidos. No entanto, as duas direções do fluxo de dados não podem ocorrer em simultâneo. Encontra-se frequentemente implementado com as seguintes características[67]:

- Um ou vários *masters*;
- Um ou vários *slaves*;
- Endereçamento dos *slaves* disponível em modos de 7 ou 10 bits;
- Opera frequentemente à velocidade de 100 kbps, 400 kbps ou 1 Mbps;
- Utiliza dois condutores sendo um para a transmissão de um sinal de sincronismo e o outro para os dados.

## SPI

O SPI é característico por ser um sistema *half-duplex* síncrono, isto é, permite enviar e receber dados em simultâneo, visto existir um condutor dedicado para cada sentido de transmissão de informação. Além disto, é partilhado pelo *master* um sinal comum de sincronismo entre os diferentes dispositivos e um sinal para endereçamento do recetor único por dispositivo.

Não existe normalmente uma velocidade específica de funcionamento do SPI, no entanto, é possível encontrar tipicamente variações entre os 20 Mbps até aos 100 Mbps[8].

### Extensores de distancia

Tanto o SPI como o I2C são interfaces de comunicação de curto alcance, estando limitados a transmitir dados para recetores não a mais de 1 metro e 10 metros, respetivamente. Porém, existe sempre a possibilidade de ser implementado um sistema capaz de aumentar o alcance máximo do sinal.

Com o intuito de demonstrar essa possibilidade, apresenta-se um exemplo implementado pela NXP Semiconductors que utiliza um circuito integrado, o P82B715, que permite prolongar uma comunicação realizada em I2C, podendo alcançar-se distâncias de cerca de 50 m[68]. Outro exemplo semelhante pode ser observado utilizando um circuito integrado da Linear Technology, o LTC4311, que permite estender a comunicação I2C em até 30 m[69].

### 2.4.6 Comparação das Diferentes Tecnologias

A tabela 2.3 tem como objetivo agrupar as diferentes tecnologias apresentadas no tópico da comunicação com fios, possibilitando expor um termo de comparação das suas características, tais como, o alcance, a taxa de transmissão e os seus pontos negativos.

Tabela 2.3: Tecnologias de comunicação com fios

Tecnologia	Alcance	Velocidade	Desvantagens
CAN	40-1000 m	5-1000 kbps	Máximo 64 <i>slaves</i>
RS-485	20-1200 m	até 10 Mbps	Máximo 32 <i>slaves</i>
LIN	40 m	2.4-19.2 kbps	Máximo 16 <i>slaves</i>
SENT	-/-/-	variável	Ponto-a-ponto
MSSP-I2C	50 m	100-1000 kbps	Requer extensores de distância
MSSP-SPI	1 m	20-100 Mbps	Distância de transmissão curta, um sinal de endereçamento por <i>slave</i>

## 2.5 Aquisição e Monitorização de Parâmetros

A monitorização e parâmetros revela-se importante, não só para fornecer informação a subsequentes algoritmos de cálculo, como também para avaliar o correto funcionamento do sistema em si.

### 2.5.1 Temperatura

Em seguida, apresentam-se seis tipos de sensores de diferentes tecnologias, capazes de realizar a medição de uma temperatura.

#### Termopares

Um termopar consiste na junção de dois metais diferentes num ponto formando uma junção, que quando aplicada num meio que se pretende realizar uma medição térmica, gera uma tensão dependente da temperatura. Esta tensão pode ser medida e interpretada podendo-se obter o valor da temperatura a que a junção foi submetida. Existe uma grande variedade de metais utilizados num termopar, tais como:

- Tipo E - Cromel / Constantan;
- Tipo J - Ferro / Constantan;

- Tipo K - Cromel / Alumel;
- Tipo R, S, B - Platina / Ródio (variam no teor de cada elemento);
- Tipo T - Cobre / Constantan.

### **Detetores de temperatura de resistência**

Detetores de temperatura de resistência (RTD) operam com base num fenómeno conhecido como termo-resistividade, que consiste na variação da resistência eléctrica de um elemento metálico, consoante a variação da temperatura. As características técnicas de um RTD variam consoante o metal utilizado na sua construção, utilizando-se frequentemente platina ou cobre.

Os RTDs providenciam sensibilidade, repetibilidade e desvio, que são uma ordem de grandeza melhor que os termopares[70].

### **Termistores**

Os termistores são semelhantes a um RTD, sendo que a principal diferença é o material utilizado na sua construção, que costuma ser baseado em substâncias semicondutoras. São caracteristicamente de resposta muito rápida e bastante sensíveis, no entanto, sofrem com a instabilidade química e eléctrica. Além disso, são de fraca linearidade, fator que não é grande adversidade e que é facilmente resolvido por um controlo inteligente.

Existem dois grandes grupos de termistores, os de coeficiente de temperatura negativo (NTC), que têm uma resposta inversamente proporcional ao aumento de temperatura; e de coeficiente de temperatura positivo (PTC), que têm uma resposta diretamente proporcional ao aumento de temperatura[70][71].

### **Pirômetros**

Estes dispositivos costumam ser mais avançados. Destinam-se a medir temperaturas sem que haja contacto com o objeto ao qual será realizada a medição. Para realizar a medida, esta tecnologia faz uso da intensidade da radiação térmica emitida por um objeto.[72].

### **Semicondutores**

Transístores bipolares de junção (TBJ) e semicondutores de oxido metálico (MOS) sofrem de uma dependência das características técnicas em função da temperatura a que operam, com os devidos métodos, as alterações das características técnicas podem ser detetadas possibilitando a quantificação do valor da temperatura. No caso de um MOS, a temperatura a que a junção se sujeita, faz variar a sua tensão de *threshold*. Já num TBJ, a influência da temperatura verifica-se na

tensão entre a base e o emissor e na corrente de saturação. Colocando ambos num termo de comparação, as dependências dos TBJ's colocam-nos em superioridade uma vez que são mais fiáveis[73].

### 2.5.2 Humidade Atmosférica

Existe uma panóplia de opções para realizar a quantificação da humidade do ar, sendo que, em seguida, encontram-se descritas seis tecnologias[9]:

#### Capacitivos

O vapor de água é um dos gases que consegue alterar a permissividade dielétrica entre dois condutores, tornando possível realizar a sua quantificação consoante a severidade da sua influência. Este tipo de sensores representa, atualmente, cerca de 75% dos sensores existentes no mercado, e são o único tipo capaz de realizar uma medição total da escala e que mantêm boa precisão nos valores próximos de 0% de humidade relativa.

Na figura 2.8 encontra-se o diagrama de um sensor capacitivo plano, em que é possível observar os dois elétrodos metálicos, situados nas faces opostas de uma substância detetora de humidade.

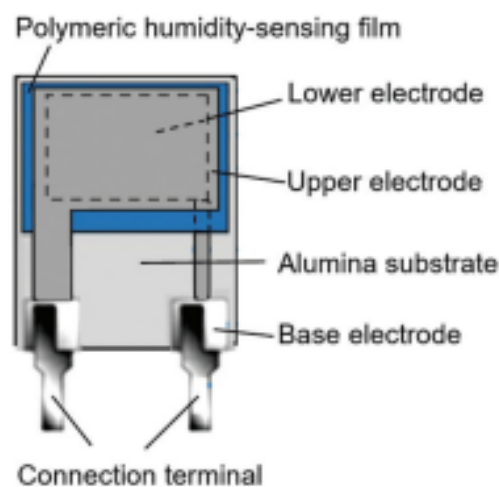


Figura 2.8: Sensor de humidade capacitivo. [9]

### **Resistivos**

Os sensores resistivos tiram proveito da alteração do valor da condutividade elétrica de certos materiais, quando expostos à humidade. A mudança da condutividade tem origem em diferentes reações tais como: absorção, reação química e difusão.

### **Gravimétrico**

Os sensores gravimétricos são sensores sensíveis à perturbação da sua massa. Estes sensores tiram partido de um oscilador piezoelétrico no qual foi depositado uma camada de um elemento detector de vapor de água. Com a alteração da concentração deste gás na atmosfera, há uma mudança da sua quantidade presente na camada detetora, implicando uma alteração na massa do disco piezoelétrico, resultando na alteração da sua frequência de ressonância.

### **Condutividade térmica**

Estes higrómetros detectam a concentração do vapor de água pela diferença da condutividade térmica do ar húmido e seco. Dois termistores com um coeficiente de temperatura negativo, são colocados em duas câmaras, uma com nitrogénio seco, o elemento de controlo, e a outra com ar atmosférico, o elemento que realiza a medição. Para realizar uma medição estes sensores são aquecidos com recurso a uma corrente e regista-se a diferença de temperatura do sensor exposto ao ar em relação ao sensor de controlo. Esta diferença torna-se notável visto que o sensor exposto tem mais dificuldades em aquecer devido à maior condutividade térmica do ar, que dissipa melhor o calor.

### **Transístor de filme fino**

Nestes sensores, a camada semicondutora é exposta ao ar atmosférico, havendo uma reação química com as moléculas de água. Com esta reação, ocorre uma alteração da concentração de portadores no semicondutor, resultando numa variação da corrente que circula no transístor.

### **Estado sólido**

O funcionamento destes sensores implica a colocação de ar atmosférico e ar seco nos dois lados de um óxido metálico condutor de prótons (fig. 2.9), dando início ao movimento de átomos de hidrogénio ( $H^+$ ) de forma a balancear as concentrações de água em ambos os lados. A origem do  $H^+$  provem da decomposição da água (2.1a) no lado do ar atmosférico. No lado oposto da membrana, constata-se a reação demonstrada em 2.1b.

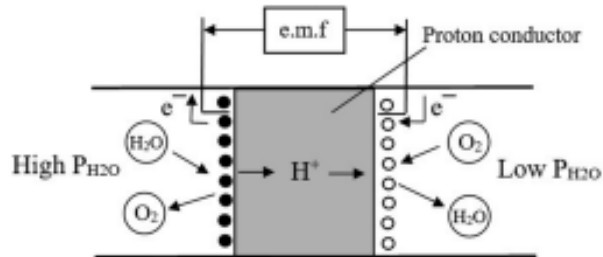
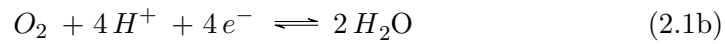
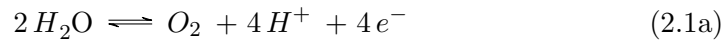


Figura 2.9: Sensor de humidade de estado sólido. [9]

É importante salientar que em ambos os lados da membrana existem duas reações químicas. Numa há sobra de eletrões e na outra há a uma deficiência destes, o que resulta numa diferença de potencial. Com este valor é possível chegar à concentração do vapor de água.

### 2.5.3 Corrente

A corrente é um parâmetro importante a medir, visto ser essencial para realizar o cálculo da produção energética de cada gerador, assim como para determinar o seu bom funcionamento. Duas tecnologias bastantes utilizadas para medir a corrente encontram-se em seguida.

#### Resistência de shunt

Este método consiste na colocação, em série com o circuito, de uma resistência de potência de elevada precisão e baixo valor. Com a passagem da corrente pela resistência, cria-se uma pequena queda de tensão insignificativa ao funcionamento do circuito. Este valor da tensão é entretanto medido e utilizado para determinar a intensidade da corrente, recorrendo-se à lei de Ohm.

Na figura 2.10 encontra-se representada a ligação típica de uma resistência de *shunt*. Note-se que normalmente os condutores que recolhem o sinal estão ligados ao *shunt* por conectores separados das ligações de potência, com o intuito de eliminar erros introduzidos pela resistência parasita dos terminais[74][75].

Algumas desvantagens deste método encontram-se apresentadas em seguida[74]:

- Não oferecem isolamento entre o circuito a ser medido e o circuito que realiza a medição;

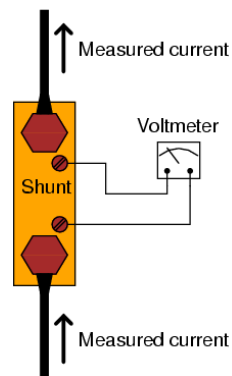


Figura 2.10: Resistência de *shunt* com ligação em Kelvin. [10]

- Introduz algumas perdas energéticas sobre o efeito de Joule;
- O valor da resistência varia com a temperatura, introduzindo erros;
- O valor em tensão do sinal resultante é pequeno, pelo que requer circuitos de amplificação de sinal.

### Efeito de Hall

Os sensores de efeito de Hall são capazes de detetar e quantificar campos magnéticos. Visto que a passagem de corrente num condutor gera um campo magnético proporcional a intensidade da corrente, com a colocação de um sensor anteriormente mencionado, sobre o condutor, cria-se um circuito capaz de responder em tensão de forma proporcional com a variação da corrente[74][75].

Existem atualmente circuitos integrados que contêm no seu interior um condutor de cobre, o sensor de efeito de Hall e um circuito de compensação e amplificação. São muito utilizados pelo facto de serem bastante precisos ao longo do tempo. Mesmo com oscilações de temperatura, oferecem isolamento galvânico entre o circuito sob teste e a saída do sinal. No entanto, são suscetíveis a interferências magnéticas externas[11][74].

Na figura 2.11, encontra-se apresentado o diagrama de blocos de um sensor de corrente de efeito de Hall comum, o ACS712. Observando a imagem, é possível isolar visualmente o condutor onde passa a corrente a ser medida, estando este marcado com cor azul. Junto a este encontra-se colocado um sensor que mede a intensidade do campo magnético com origem no condutor, seguindo-se de um circuito de compensação e amplificação de sinal, colocando no terminal de saída um valor em tensão proporcionalmente linear à corrente[11].

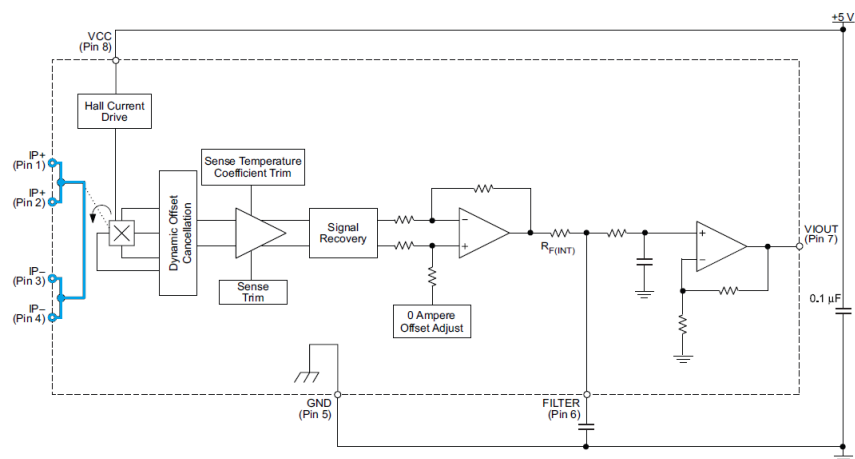


Figura 2.11: Diagrama de blocos do ACS712. [11]



## Capítulo 3

---

# Arquitetura do Sistema

---

*Neste capítulo são apresentadas descrições detalhadas da arquitetura do sistema. Este estudo demonstra as exigências envolvidas no projeto, assim como comparações de possíveis soluções, possibilitando realizar decisões que permitam uma implementação satisfatória do sistema.*

### 3.1 Arquitetura Geral

A figura 3.1 apresenta um diagrama de blocos da arquitetura geral do sistema desenvolvido.

Este sistema consiste fundamentalmente num computador central que disponibiliza diversos serviços a outros dispositivos ligados a si, assim como aos utilizadores. Esta máquina é normalmente referida apenas como servidor.

A grande maioria do controlo do sistema provém de algoritmos executados nesta máquina que tomam todas as decisões, dependendo de configurações realizadas pelos utilizadores, assim como de informação recolhida.

Cada módulo IoT realiza diferentes tarefas, variando desde a recolha de informação atmosférica local, níveis de água nos reservatórios e no furo, caudais, entre outros. Após a recolha dos dados ter sido realizada pelos diversos módulos, são enviados para o servidor os resultados obtidos, aquando o pedido para tal. O método utilizado, comum entre eles, foi a transmissão de um pacote de dados MQTT sob a rede Wi-Fi.

Por fim, o servidor serve o principal propósito de fornecer um serviço web que permita a um utilizador aceder, em qualquer parte do mundo através de uma interface gráfica, à informação recolhida, permitindo também que seja realizado algum controlo sobre o sistema. Informação mais específica, tais como os relató-

rios ou avisos importantes, são enviados através de outros meios para notificar o utilizador, tais como Email e, em caso de urgência, o serviço pago de SMS.

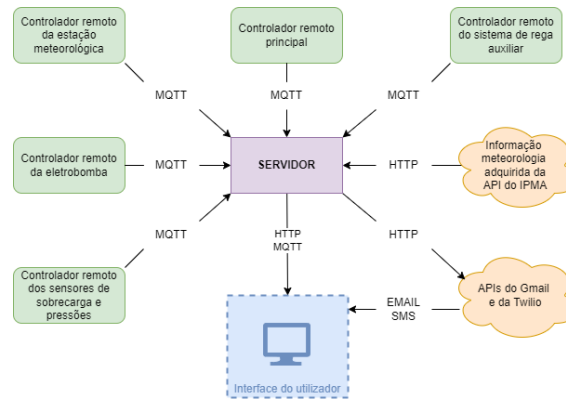


Figura 3.1: Arquitetura geral do sistema

### 3.1.1 Arquitetura do Controlador Remoto de Controlo da Eletrobomba

O controlador remoto da eletrobomba, é um módulo responsável por realizar o controlo da eletrobomba que retira água para consumo, de um furo situado na residência.

Até ao momento, o único método de atuação do dispositivo era de forma manual, sendo necessário ligar o quadro através de um botão presente na lateral. Este controlador tem como principal função permitir atuar o dispositivo de forma remota, assim como, implementar diversos métodos de segurança, tais como, sobre-corrente, sub-corrente e limite de arranques por hora.

Este controlador, representado através de um diagrama de blocos na figura 3.2, conta com um microcontrolador central responsável pela interpretação e computação de dados e informação enviada e recebida. Esta informação pode ser de origem local ou remota.

Existem dois sinais analógicos que devem de ser recolhidos, sendo estes correspondentes ao valor da corrente consumida pela bomba e, da tensão presente no quadro de controlo. O valor da corrente permite realizar uma análise sobre o consumo instantâneo, desligando a bomba caso sejam detetados valores fora dos limites definidos pelo utilizador.

Para além disso, existem três sinais digitais provenientes de um interruptor de seleção do modo Manual/Automático e da presença das linhas de fase e neutro no contactor.

Consoante os resultados destes sinais digitais, recorrendo-se a uma tabela semelhante à tabela 3.1, é possível determinar o estado de funcionamento da eletrobomba.

Tabela 3.1: Tabela de estados dos sinais da eletrobomba

Sinais	Eb. ligada	Sem água	Térmico	Eb. desligada
Fase do contactor	1	0	1	0
Neutro do contactor	1	1	0	0

O penúltimo constituinte a destacar consiste num conjunto de botões e um LCD, que permite realizar a navegação entre os diferentes parâmetros configuráveis e consultar a informação recolhida. Por fim, o último elemento de interface é o relé de saída que permite alimentar o relé de nível com 230 V atuando, por sua vez, o contactor da eletrobomba.

Durante o desenvolvimento, é do interesse deixar espaço disponível e preparado, que possibilite implementar algum tipo de conectividade sem fios, permitindo a este módulo comunicar com o exterior.

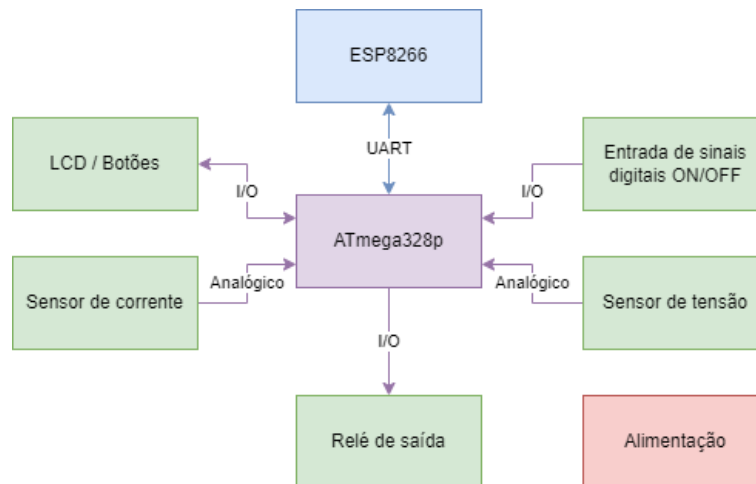


Figura 3.2: Diagrama de blocos do controlador remoto da eletrobomba

### Medição da tensão

Este parâmetro é o mais simples de medir, visto que o conversor analógico-digital (ADC) do microcontrolador e o sinal a medir são da mesma unidade. Apesar disto, a amplitude do sinal não é de todo semelhante à gama de medição do ADC, no entanto, esse fator é facilmente resolvido com um circuito para realizar o condicionamento de sinal.

### Medição da corrente

Para realizar a medição da corrente, é necessário recorrer a um circuito de condicionamento de sinal que altere a unidade do valor medido, isto é, é necessário implementar um sensor de corrente com saída em tensão de forma a permitir ao microcontrolador medir a intensidade da corrente.

Encontra-se apresentada na figura 3.3 um circuito de transimpedância desenvolvido pela Texas Instruments, que tira partido de um transformador de corrente capaz de relacionar a corrente que circula pelo seu circuito primário, com uma corrente que circula no seu circuito secundário. Note-se que o sinal à saída deste dispositivo ainda não se encontra na unidade nem na gama pretendida, pelo que é necessário implementar um circuito de condicionamento de sinal recorrendo a uma resistência de *burden* e um circuito de condicionamento de sinal. A escolha da resistência de *burden* e do transformador de corrente influenciam diretamente a gama de medições que podem ser realizadas com o circuito.

Algumas vantagens da utilização de um transformador de corrente é o desperdício residual de energia, assim como o fornecimento de isolamento galvânico da linha sobre teste.

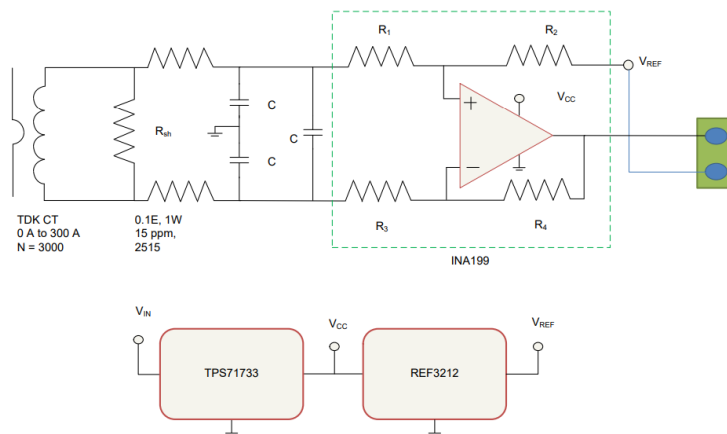


Figura 3.3: Conversão e condicionamento do sinal da corrente em tensão[12]

Outro método considerado foi a medição da intensidade da corrente recorrendo-se a um sensor de efeito de *Hall*. Este sensor tem a capacidade de realizar a medição da corrente de forma indireta, recorrendo à intensidade do campo magnético gerado. Assim como o transformador de corrente, um circuito integrado que meça a corrente a partir do efeito de *hall*, costuma ter um bom isolamento entre a linha de teste e a linha de alimentação e de saída de sinal. Para além disso, o desperdício de energia sobre o efeito de joule é mínimo.

Na tabela 3.2, são apresentadas características técnicas das tecnologias anteriormente apresentadas. Para realizar esta comparação, recorreu-se à escolha de dois possíveis candidatos que poderiam realizar a medição da corrente neste circuito.

Note-se que os valores apresentados no caso da implementação usando um transformador de corrente, foram recolhidos seguindo a implementação padrão da Texas Instruments, pelo que o valor máximo de corrente medida pelo circuito deveria de ser ajustado para um valor mais apropriado as necessidades. É importante lembrar que todos os parâmetros apresentados englobam o circuito de condicionamento de sinal.

Tabela 3.2: Características de candidatos para a medição da corrente[11, 12]

Sensor	ACS712	Transformador Corrente
Tensão de alimentação	4.5 a 5.5 V	3.5 a 6.5 V
Gama de medição	+/-20 A	100 A
Corrente quiescente	10 mA	5 mA
Sensibilidade	100 mV/A	25 mV/A
Resistência interna	1.2 mΩ	n/a

Apesar da gama de medição do circuito com o transformador de corrente ter uma capacidade máxima cerca de 5 vezes superior, em relação ao ACS712, a sua sensibilidade é apenas 4 vezes inferior. Isto deve-se ao facto do circuito integrado de efeito de *Hall* desperdiçar cerca de 20% da amplitude de medição do ADC pois a tensão analógica na sua saída encontra-se limitada a cerca de 0.5 V das linhas de alimentação.

Na tabela 3.3 encontra-se o custo proveniente de diversos fornecedores, tanto para o sensor de corrente de efeito de *Hall*, como também para os componentes necessários para construir o circuito com o transformador de corrente. O transformador de corrente escolhido foi o AC1020, visto ser da mesma gama de medição que o ACS712.

Tabela 3.3: Preço e disponibilidade de candidatos para a medição da corrente[18, 19, 20, 21]

Distribuidor	ACS712	Trf. Corr.
Farnell	Sem stock	Sem stock do INA199
Mouser	4.67€	Sem stock do T. C.
Digikey	Sem stock	1.01€ + 5,76€
LCSC	4.89€	Sem stock do T. C.

Determinou-se que o dispositivo mais apropriado para realizar a medição da corrente foi o ACS712, sendo que os principais pontos de consideração foram a sua facilidade de implementação e sobretudo, o seu diminuto tamanho.

### Microcontrolador

O microcontrolador representa um dos papeis mais importantes no módulo desenvolvido, uma vez que é responsável por acrescentar a inteligência ao sistema.

Foi realizado um estudo relativamente a quatro candidatos da vasta gama de produtos existentes no mercado, sendo que são apresentados os resultados na tabela 3.4 para os microcontroladores de 8 bits de baixa performance e, na tabela 3.5 para os microcontroladores de 32 bits, com uma maior capacidade.

Tabela 3.4: Características dos microcontroladores de 8 bits[22, 23]

Microcontrolador	ATtiny3227	ATmega328p
Tensão de alimentação	1.8 a 5.5 V	1.8 a 5.5 V
Tamanho da memória Flash	32 kB	32 kB
Tamanho da memória RAM	3 kB	2 kB
Frequência máxima de operação	20 MHz	20 MHz
Numero de I/Os	24	32
Resolução do ADC	12 bits	10 bits
Hardware de comunicação implementado	1 x I2C 1 x SPI 2x USART	1 x I2C 1 x SPI 2x USART

Tabela 3.5: Características dos microcontroladores de 32 bits[24, 25]

<b>Sensor</b>	<b>STM32G431</b>	<b>ESP32</b>
Tensão de alimentação	1.71 a 3.6 V	2.5 a 3.6 V
Tamanho da memória Flash	128 kB	Interna - 0 B Externa - 16 MB
Tamanho da memória RAM	22 kB	320 kB
Frequência máxima de operação	170 MHz	240 MHz
Numero de I/Os	86	34
Resolução do ADC	16 bits	12 bits
Hardware de comunicação implementado	3 x I2C 1 x FDCAN 4 x USART 1 x LPUART 3 x SPI	1 x SDIO 2 x I2C 3 x UART 3 x SPI

Na tabela 3.6 é possível encontrar os microcontroladores e os seus respetivos preços em diferentes fornecedores.

Tabela 3.6: Preço e disponibilidade dos microcontroladores[18, 19, 20, 21]

<b>Distribuidor</b>	<b>ATtiny3227</b>	<b>ATmega324p</b>	<b>STM32G431</b>	<b>ESP32</b>
Farnell	0.95€	6.54€	Sem stock	Sem stock
Mouser	0.97€	5.43€	Sem stock	4.31€
Digikey	Sem stock	5.39€	Sem stock	2.80€
LCSC	Não existe	10.58€	7.30€	3.78€

### Conectividade

De forma a implementar conectividade com outros dispositivos, inicialmente pensou-se em desenvolver uma pequena rede que permitisse a troca de informação. De todas as tecnologias avaliadas, decidiu-se aprofundar a pesquisa sobre dois dispositivos, presentes em 3.7.

Tabela 3.7: Características das tecnologias de comunicação sem fios[26, 27]

Módulo	RFM95W	HC-12
Tensão de alimentação	1.7 a 3.7 V	3.2 a 5.5 V
Corrente média de funcionamento	10.3 mA	16 mA
Frequência de banda	868/915 MHz	433.4/473.0MHz
Alcance	2 Km	1 Km
Taxa de transmissão	0.11 a 37.5 kbps	1.2 a 115.2 Kbps

A tabela 3.8 apresenta uma comparação dos preços das diferentes tecnologias em diferentes fornecedores.

Tabela 3.8: Preço e disponibilidade das tecnologias de comunicação sem fios[18, 19, 20, 21, 28]

Módulo	RFM95W	HC-12
Farnell	Sem stock	Não existe
Mouser	20.45€	Não existe
Digikey	20.59€	Não existe
LCSC	Não existe	Não existe
eBay	9.57€	3.83€

Durante a projeção do sistema, decidiu-se utilizar um ATmega328p sendo a principal razão da sua escolha o facto de já ter sido adquirido para outros projetos o que resultou em exemplares sobrantes. Com isto, decidiu-se, inicialmente, utilizar o módulo HC-12 para realizar a transmissão de dados sem fios, pois é de custo bastante reduzido, e tem taxas de transmissão relativamente altas.

### Fonte de alimentação

Uma vez que este controlador foi projetado para funcionar no interior do quadro de controlo e proteção da eletrobomba, o método mais simples e seguro de realizar a alimentação do circuito consistiu em instalar uma fonte de alimentação que utilizasse energia da rede elétrica para satisfazer as necessidades do circuito.

A projeção deste sistema aconteceu algum tempo antes da introdução de novos dispositivos na rede de comunicações, pelo que se considerava ser suficiente. Tal não se comprovou com a introdução do controlador remoto do sistema de rega auxiliar. Como se pretendia ter uma taxa de atualização do sistema bastante elevada, e visto os módulos utilizados funcionarem a uma taxa de transmissão relativamente baixa devido ao alcance, começou a haver uma elevada congestão da rede, pelo que se decidiu abandonar esta abordagem, tirando partido de uma segunda rede previamente instalada, o Wi-fi.

### 3.1.2 Arquitetura do Controlador Remoto dos Sensores de Sobrecarga e Pressões

Durante o enchimento de um reservatório, caso ocorra uma falha não detetável no sensor ou no sistema de fim de curso de enchimento, torna-se real a possibilidade de acontecer uma sobrecarga do depósito de água, sendo que o excesso do abastecimento é desperdiçado.

Foram considerados dois métodos para contrabalançar este problema, sendo que o primeiro consistia na implementação de uma temporização que limitasse o tempo de enchimento. Caso não houvesse uma ordem do sistema de fim de curso durante um determinado intervalo de tempo, o abastecimento do reservatório seria parado e, ativada uma anomalia. Este método não previne o desperdício de grandes quantidades de água, apenas limita-o. Devido à possibilidade de continuar a haver perdas que podem chegar à ordem do metro cúbico, este método foi imediatamente abandonado, tendo em conta a escassez do recurso.

O segundo método, consiste na implementação de um segundo sensor de fim de curso que detete o nível da água, semelhante ao primeiro. No entanto, apesar de ser muito mais improvável, pode ocorrer um evento que consiste numa avaria indetetável de dois sensores independentes, em simultâneo. Para combater esta possibilidade, existe a necessidade de implementar um método que permita realizar testes ao segundo fim de curso. O método pensado consiste na descarga de uma quantidade minúscula de água sobre o sensor, através da abertura de uma eletroválvula, permitindo realizar testes periódicos ao último nível de defesa contra sobrecargas.

O diagrama de blocos que demonstra a arquitetura deste módulo do sistema, encontra-se representado na figura 3.4. Note-se que existe *hardware* adicional,

pois durante o desenvolvimento do sistema, houve a necessidade de recolher dois sinais digitais e um sinal analógico relativos a pressões em diferentes pontos do circuito hidráulico. Estes sinais não são importantes para o funcionamento do sensor de sobrecarga, mas devido à falta de entradas disponíveis noutros módulos, adicionou-se um microcontrolador a este sistema capaz de realizar a tarefa pretendida, partilhando a comunicação ao servidor com o sistema de deteção de sobrecargas.

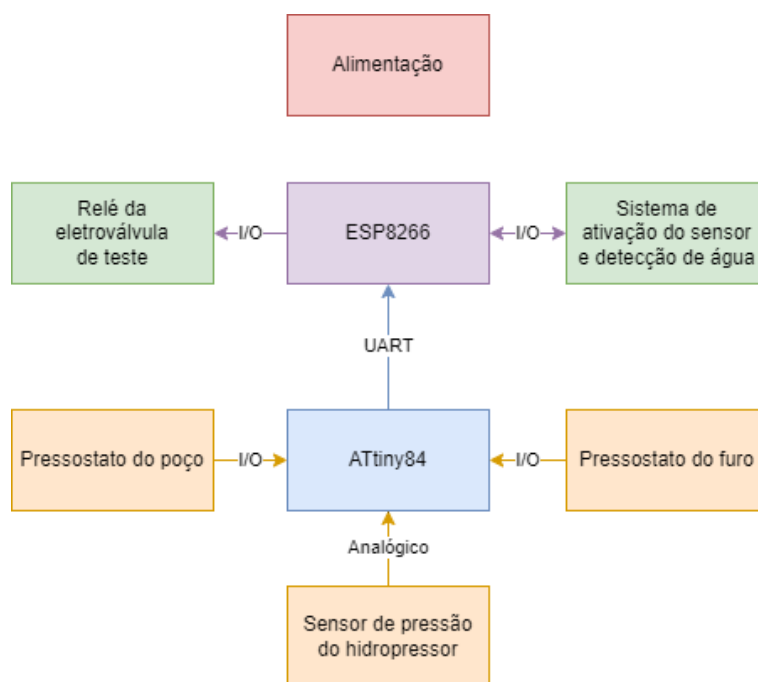


Figura 3.4: Diagrama de blocos do controlador remoto dos sensores de sobrecarga e pressões

### Microcontrolador

Na tabela 3.9 encontra-se apresentado um estudo adicional relativamente a dois microcontroladores.

Tabela 3.9: Características dos microcontroladores apresentados[29, 30]

Módulo	ESP8266	ATtiny84
Tensão de alimentação	2.5 a 3.6 V	1.8 a 5.5 V
Tamanho da memória Flash	Interna - 0 B Externa - 16 MB	8 kB

Tabela 3.9: Características dos microcontroladores apresentados[29, 30]

<b>Módulo</b>	<b>ESP8266</b>	<b>ATtiny84</b>
Tamanho da memória RAM	50 kB	512 B
Frequência máxima de operação	160 MHz	16 MHz
Numero de I/Os	17	12
Resolução do ADC	10 bits	10 bits
Hardware de comunicação implementado	1 x SDIO 1 x I2C 1 x UART + 1 TX 1 x SPI	1 x Universal Serial Interface

A tabela 3.10 apresenta uma comparação dos preços dos diferentes controladores em diferentes fornecedores.

Tabela 3.10: Preço e disponibilidade dos microcontroladores em diferentes fornecedores[18, 19, 20, 21]

<b>Módulo</b>	<b>ESP8266</b>	<b>ATtiny84</b>
Farnell	Sem stock	1.46€
Mouser	2.85€	2.58€
Digikey	2.80€	2.55€
LCSC	1.90€	4.49€

Relacionando todos os microcontroladores apresentados até este ponto, destaca-se o ESP8266 para realizar o controlo do sensor de sobrecarga do reservatório, pelo seu baixo custo e pela resolução de todas as necessidades apresentadas. O modelo utilizado foi o ESP-01.

Como foi indicado anteriormente, devido à necessidade posterior de recolher dois novos sinais digitais e um sinal analógico, independentes do sensor de sobrecarga, decidiu utilizar-se um ATtiny84 como expansão, sendo que este realiza a recolha de informação e transmite-a ao ESP8266 que, por sua vez, envia os dados para o servidor. A escolha deste microcontrolador de 8 bits foi principalmente influenciada pelo facto de existirem unidades sobrantes.

### 3.1.3 Arquitetura do Controlador Remoto de Recolha de Informação Meteorológica

Na figura 3.5 encontra-se representado o diagrama de blocos do *hardware* do módulo responsável pela recolha de dados atmosféricos. Os diferentes sensores incorporados neste módulo são responsáveis por recolher informação sobre a pressão, temperatura e humidade atmosférica, assim como a intensidade da radiação solar visível e ultravioleta. Um último parâmetro atmosférico recolhido, de maior importância, é o valor da precipitação.

Para além destes dados recolhidos, encontra-se incorporado do módulo um sensor de temperatura e humidade responsável por medir os parâmetros atmosféricos dentro da caixa, onde se situa a maioria do *hardware*, permitindo determinar as condições de operação do sistema.

Por fim, este módulo contém um botão de pressão que permite colocar o módulo num estado de configuração e um LED, que permite identificar visualmente a tarefa que está a ser executada pelo controlador, assim como a existência ou ausência de alguma anomalia.

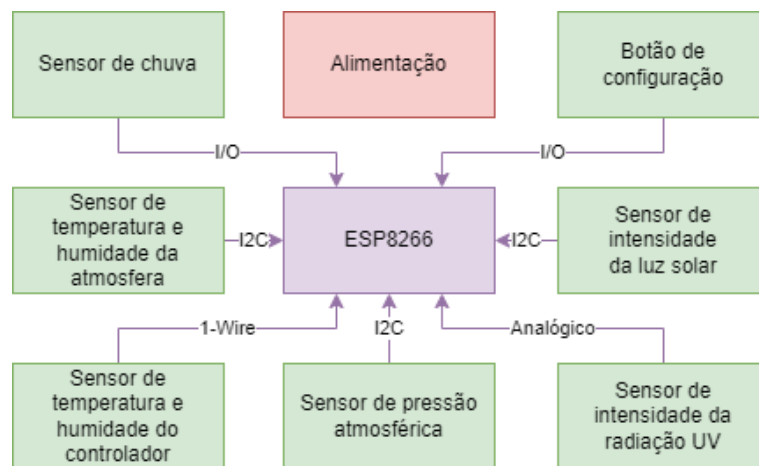


Figura 3.5: Diagrama de blocos do controlador remoto de recolha de parâmetros meteorológicos

#### Sensores de temperatura, humidade e pressão atmosférica

Pretende-se que este módulo realize a medição da temperatura e da humidade em duas localizações diferentes, permitindo obter conhecimento sobre as condições ambientais às quais o sistema está submetido internamente, assim como as condições atmosféricas.

Sendo que estes sensores destinam-se a medir parâmetros atmosféricos, a gama de operação deve de ser apropriada, tanto para a temperatura, como para a

humidade. É também importante ter em conta o balanço entre o custo e a precisão destes dispositivos, logo, por exemplo, a escolha do sensor de humidade está limitada à tecnologia capacitiva ou resistiva.

Na tabela 3.11 encontram-se apresentados três candidatos possíveis, já na tabela 3.12 encontram-se apresentados os preços de cada sensor em diversos fornecedores, assim como a sua disponibilidade.

Tabela 3.11: Características de candidatos para a medição da temperatura e humidade [31, 32, 33, 34]

Sensor	DHT11	DHT22	HDC1080	HTS221
Tensão de alimentação	3.3 a 6 V	3.3 a 6 V	2.7 a 5.5 V	1.7 a 3.6 V
Gama de operação	20 a 80 % 0 a 50 °C	0 a 100 % -40 a 80 °C	0 a 100 % -20 a 85 °C	0 a 100 % -40 a 120 °C
Precisão	+5 % +2 °C	+2 % +0.5 °C	+2 % +0.2 °C	+3.5 % +1 °C
Protocolo de comunicação	1-Wire	1-Wire	I2C	I2C

Tabela 3.12: Preço e disponibilidade de candidatos para a medição da temperatura e humidade [18, 19, 20, 21]

Distribuidor	DHT11	DHT22	HDC1080	HTS221
Farnell	7.71€	11.07€	Sem stock	16.57€
Mouser	4.29€	Sem stock	3.92€	4.67€
Digikey	5.21€	Sem stock	7.40€	4.60€
LCSC	1.28€	1.54€	Sem stock	Sem stock

Em relação à medição da pressão atmosférica, destacou-se dois sensores da Bosh sendo que as suas características encontram-se apresentadas em 3.13, já o preço dos respetivos sensores encontram-se apresentados em 3.14.

Tabela 3.13: Características de candidatos para a medição da pressão atmosférica [35, 36]

Sensor	BMP280	BME280
Tensão de alimentação	3.3 a 6 V	1.71 a 3.6 V
Gama de operação	-40 a 85 °C	0 a 100 %
	300 a 1100 hPa	-40 a 85 °C
		300 a 1100 hPa
Precisão	+1 °C	+3 %
	+1 hPa	+1 °C
		+1 hPa
Protocolo de comunicação	I2C	I2C

Tabela 3.14: Preço e disponibilidade de candidatos para a medição da pressão atmosférica [18, 19, 20, 21]

Distribuidor	BMP280	BME280
Farnell	5.76€	Sem stock
Mouser	5.90€	7.59€
Digikey	Sem stock	9.39€
LCSC	2.30€	5.28€

De todos os sensores apresentados, decidiu-se escolher o DHT11 para realizar a análise interna da temperatura e humidade do controlador, já para realizar a medição dos parâmetros atmosféricos, existem duas vertentes possíveis, ou se escolhe sensores de temperatura e humidade em conjunto com um sensor de pressão atmosférica, ou se opta pelo BME280 que realiza todas as funções pretendidas.

Optou-se pela primeira opção, usando-se o HDC1080 para realizar a medição da temperatura e humidade, sendo que tem uma boa precisão e um baixo custo, conjugando-se com o BMP280, para realizar a medição da pressão.

### Medição da intensidade solar visível e ultravioleta

Os sensores utilizados para realizar a medição de intensidade solar visível e ultravioleta encontram-se apresentados em 3.15, sendo que já se encontravam adquiridos antes da projeção do sistema.

Note-se que, no caso do BH1750, existe um registo de configuração que determina o multiplicador do tempo de exposição do sensor, sendo que este registo deve de ser configurado para o mínimo possível, obtendo-se assim um valor máximo de 100000 LUX lidos pelo sensor.

Tabela 3.15: Sensores utilizados para a medição da intensidade solar visível e UV[37, 38]

Sensor	GUVA-S12SD	BH1750
Tensão de alimentação	2.7 a 5.5 V	2.4 a 3.6 V
Gama de operação	240-370nm	0 a 65535 LUX.
Protocolo de comunicação	Analógico	I2C

### Medição da precipitação

De forma a medir a pluviosidade, utilizou-se um sensor adquirido da MISOL, o WH-SP-RG.

Uma representação da estrutura mecânica deste sensor encontra-se apresentada em 3.6. Este sensor recolhe a água da chuva por meio de um funil, reencaminhando-a para um sistema de baldes oscilantes que tombam quando atingem a sua capacidade máxima de volume em reservatório. Sempre que há uma mudança de balde, há também uma mudança do circuito da água, o que provoca o movimento oscilatório detetado por um sensor encastrado na estrutura.

Note-se que o volume dos baldes é fixo, assim como a área do funil de recolha, pelo que é possível calcular o incremento do nível, em milímetros, de chuva recolhida por cada impulso.

Uma imagem ilustrativa do sensor utilizado encontra-se em 3.7.

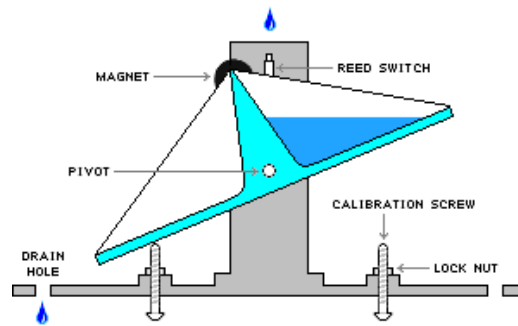


Figura 3.6: Desenho mecânico do sensor de precipitação[13]



Figura 3.7: Sensor de precipitação[14]

### Microcontrolador e fonte de alimentação

De modo a realizar o controlo do sistema, houve a necessidade de escolher um microcontrolador que proporcionasse o *hardware* necessário para recolher a informação fornecida pelos sensores. Para realizar a interação com todo o sistema é necessário, pelo menos, um canal de I2C, uma entrada analógica e quatro entradas de uso geral.

De forma a realizar a comunicação com o servidor, o dispositivo de controlo deve possuir capacidades que o permitam ligar-se a rede Wi-fi disponível no local. Como foi analisado anteriormente, o dispositivo mais adequado é o ESP8266, sendo que se utilizou a versão do módulo o ESP-12E.

Em termos de alimentação do circuito, considerou-se implementar uma fonte de alimentação com recurso a um transformador, recolhendo-se energia da rede. Esta decisão baseou-se em alguns fatores, como por exemplo facto desta abordagem ser mais simples e barata pois não é necessário nenhum sistema de armazenamento e gestão de energia. Outro fator condicionante é o facto do dispositivo IoT estar baseado num microcontrolador que utiliza Wi-fi como meio de comunicação,

e de ter uma elevada taxa de atividade, o que agrava ainda mais o consumo de energia.

### 3.1.4 Arquitetura do Controlador Remoto do Sistema de Rega Auxiliar

O controlador remoto do sistema de rega auxiliar é responsável por realizar a rega de uma pequena porção de área verde da habitação. A necessidade de implementação deste sistema surgiu de emergência, pelo que aquando ao seu desenvolvimento, em simultâneo com o resto do projeto, não foi aprofundado o método de implementação de comunicação com o exterior, sendo que apenas foi disponibilizado um modo que facilitasse a instalação de uma interface que realizasse comunicações, *a posteriori*.

O diagrama da figura 3.8 representa, em formato de diagrama de blocos, os constituintes principais do controlador remoto auxiliar de rega.

Um bloco importante a destacar é o relógio em tempo real (RTC), sendo responsável por manter a hora e data do sistema, permitindo que este realize uma correta temporização dos programas de rega. Este componente possibilita também que o controlador opere de forma “*offline*”, ou seja, de forma isolada e independente do servidor. Para além deste constituinte, o controlador de rega contém um LCD e um conjunto de botões que permitem realizar a visualização, navegação e configuração de diversos parâmetros.

Outros constituintes a destacar, são os relés de saída, que atuam as eletroválvulas e, um LED de sinalização que permite determinar a operação que está a ser executada pelo controlador, assim como a existência ou ausência de anomalias.

Este controlador contém um sensor de temperatura e humidade que permite medir os parâmetros atmosféricos dentro da caixa, indicando as condições a que o controlador se encontra sujeito.

Como referido anteriormente, este sistema encontra-se preparado para aceitar um módulo de comunicação sem fios, que comunique com o microcontrolador de controlo principal, por via UART. Isto permite implementar um método que possibilite a transmissão da informação recolhida, assim como aceitar ordens provenientes de origem remota.

Em termos de programação das eletroválvulas, pretende-se que seja possível realizar uma programação que permite definir o ou os dias da semana em que uma válvula possa ativar, assim como a sua hora e minuto de disparo de cada programa. O tempo máximo de funcionamento das válvulas deve de ser definido por dois registos que simbolizem os minutos e os segundos, sendo que o seu valor pode ser influenciado pelo valor da humidade do solo e pela previsão meteorológica.

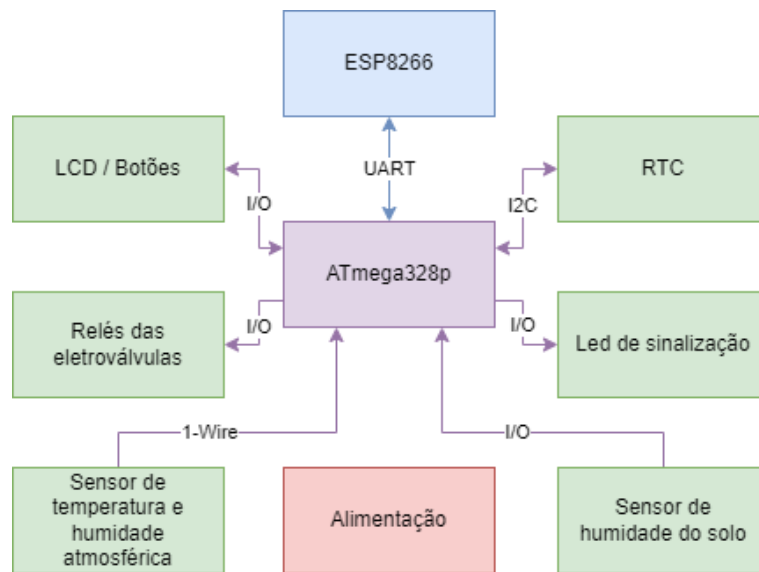


Figura 3.8: Diagrama de blocos do controlador remoto do sistema de rega auxiliar

As últimas configurações que devem de ser consideradas resultam na possibilidade de determinar qual ou quais das válvulas devem de operar em cada programa, assim como, se o programa em si, deve estar habilitado ou desabilitado.

### Sensor de humidade do solo

Existem dois grandes grupos de sensores que permitem realizar a medição da humidade do solo dividindo-se em duas categorias, os sensores resistivos e os sensores capacitivos. Os sensores resistivos costumam apresentar um baixo custo e relativa simplicidade, ao passo que sofrem corrosão galvânica enquanto são operados. Em contrapartida, os sensores capacitivos costumam ser mais complexos, mais caros e menos robustos, no entanto, como estão isolados eletricamente do solo, não sofrem grande deterioração durante o seu funcionamento.

Na tabela 3.16 encontram-se apresentados alguns sensores analisados.

Note-se, primeiramente, a elevada diferença de preço entre o sensor resistivo e o último sensor capacitivo, sendo que a explicação para o baixo custo do primeiro sensor capacitivo deve-se à sua fraca robustez, visto ter o circuito eletrónico exposto aos elementos.

Além disso, repare-se que o sensor resistivo é o dispositivo que mais aproveita a amplitude da possível variação de tensão na saída. Caso o sinal fosse recolhido por um ADC que convertesse valores analógicos dos 0 aos 5 V, o aproveitamento da gama total do ADC não passaria, respetivamente, dos 84%, 26% e 58%.

Tabela 3.16: Sensores utilizados para medir o nível da humidade do solo[39, 40, 41]

Sensor	SEN0114	SEN0193	SEN0308
Tipo de sensor	Resistivo	Capacitivo	Capacitivo
Tensão de alimentação	3.3 a 5 V	3.3 a 5.5 V	3.3 a 5.5 V
Comunicação	Analógica	Analógica	Analógica
Tensão de saída $V_{cc}=5V$	0 a 4.2 V	1.2 a 2.5 V	0 a 2.9 V
Preço (Farnell)	2.64€	5.76€	18.44€

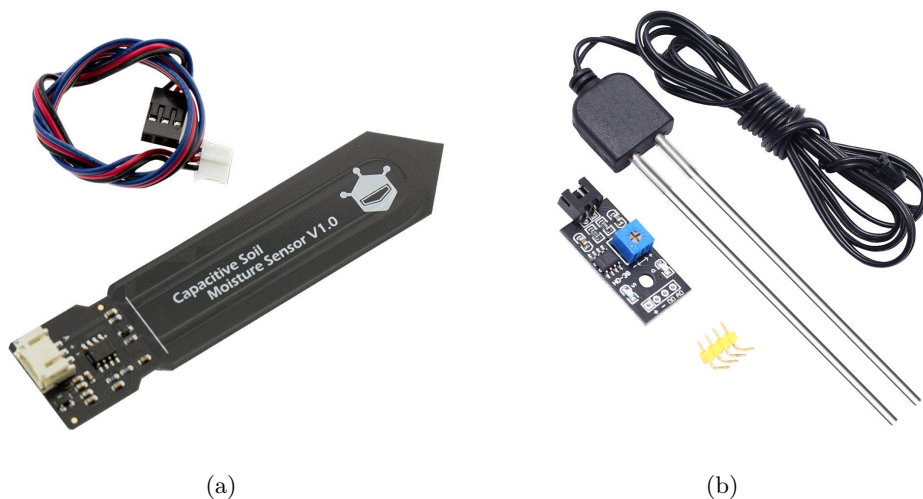


Figura 3.9: Sensores de humidade do solo (a) Capacitivo (b) Resistivo[15, 16]

### Microcontrolador e sensor de temperatura e humidade

Como referido anteriormente, devido à urgência da construção do controlador, a escolha de material foi condicionada pela disponibilidade dos componentes adquiridos. Desta forma, utilizou-se um ATmega328p para realizar o controlo, decidindo-se deixar um espaço para a implementação de um módulo de comunicação HC-12 que seria efetuado posteriormente. Além disso, tal como foi referido, a rede com o HC-12 foi abandonada pela incapacidade de manobrar o elevado tráfego, pelo que se decidiu utilizar um ESP8266 para realizar a *interface* entre o servidor e o controlador.

Para realizar a recolha de informação sobre a temperatura e humidade, no interior da caixa, utilizou-se um DHT11, por se considerar que o seu baixo custo compensa sobre a baixa precisão e gama de medição.

### 3.1.5 Arquitetura do controlador remoto principal

Este controlador é responsável por recolher um conjunto de valores relacionados com o estado do circuito hidráulico da residência. O diagrama de blocos que o representa encontra-se apresentado em 3.10.

Com este dispositivo pretende-se realizar a recolha de informação dos caudais que circulam nos 6 caudalímetros integrantes do circuito hidráulico, a recolha dos níveis de água de ambos os reservatórios, assim como do furo da residência. Pretende-se também realizar a atuação de um compressor de ar e das válvulas atuadoras do sistema de rega dos espaços verdes principais, assim como recolher valores de pressões em diferentes localizações do circuito. Por último, este controlador deve realizar o controlo de uma eletrobomba auxiliar do sistema.

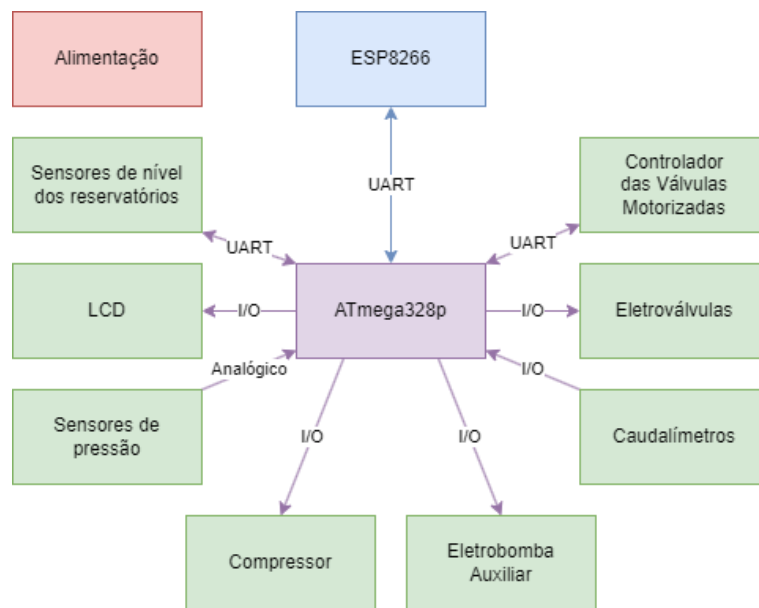


Figura 3.10: Diagrama de blocos do controlador remoto principal

Devido ao elevado custo deste controlador, em relação aos outros dispositivos projetados, decidiu-se implementar primeiramente um prototipo de validação do conceito. Para tal reaproveitou-se parte de um sistema antigo e obsoleto que realizava o controlo da rega e leitura dos níveis dos reservatórios.

Este controlador funcionava à base de um ATmega328p, sendo que tinha como função permitir a configuração de programas de rega, atuando nas eletroválvulas. Este sistema continha um controlador subordinado que realizava a recolha dos

níveis de água dos reservatórios, transmitindo por RS-485 a informação recolhida, sendo esta apresentada num LCD.

Para realizar a nova implementação, aproveitou-se parte do *hardware* disponível, uma vez que foram removidos componentes tais como os botões de navegação e o RTC, dado que foram realizadas novas ligações aos sensores implementados, aproveitando as conceções vazias. A maior conveniência do sistema é a disponibilidade de uma linha de alimentação de 5 V que foi utilizada para alimentar a maior parte dos dispositivos.

Naturalmente que o código antigo do sistema foi descartado, desenvolvendo-se um novo. A comunicação com o exterior realiza-se através de uma porta UART que comunica com um EP8266, tal como o controlador remoto da eletrobomba.

### Sensor de nível do furo

Na medida em que um dos dados mais importantes apresentados pelo sistema é o caudal proveniente do furo, deve-se realizar a recolha do valor do nível de água no seu interior, sendo que a integração deste valor no tempo resulta no caudal.

Posto que o furo tem cerca de 20 centímetros de diâmetro por 57 metros de profundidade, sensores de nível que se baseiem em métodos ultrassónicos estão fora de questão.

Investigaram-se duas implementações distintas de medição do nível, apresentadas em seguida.

O primeiro método consiste na implementação de um sensor de pressão no fundo do furo, tal como o sensor da figura 3.11, de modo a que consiga medir a pressão da água nesta região. A relação entre o valor medido e a altura manométrico da água encontra-se representado pela formula da pressão hidrostática 3.1.

$$P_{(Pa)} = \rho gh \quad (3.1)$$

O segundo método consiste na introdução de um tubo de pequeno calibre ao longo do furo, sendo que pretende-se pressurizar com ar comprimido. O objetivo da injeção do ar é realizar a expulsão da água de dentro do tubo, até ser encontrado o equilíbrio, que coincide com o alcance da bolsa de ar com a extremidade do tubo no fim do furo. Neste momento, não é possível incrementar a pressão pois o excesso de ar escapa no fundo do furo.

A relação entre a pressão do ar e a altura manométrico da água pode ser deduzida pela relação entre a lei de Arquimedes, em 3.2a e pela lei de Pascal, em 3.2b, obtendo-se a formula da pressão hidrostática em 3.1.



Figura 3.11: Sensor de medição do nível da água[17]

Note-se que a massa do corpo de ar foi desprezada durante os cálculos, uma vez que a sua densidade é múltiplas ordens de grandeza inferior à massa de água.

$$I_{(N)} = \rho g V \quad (3.2a)$$

$$F_{(N)} = PA \quad (3.2b)$$

Para realizar a medição da pressão do ar comprimido, basta colocar um transdutor de pressão junto ao compressor. Na tabela 3.17 encontra-se uma comparação dos diferentes métodos.

Tabela 3.17: Métodos possíveis para medir o nível da água no furo[42]

Método	1	2
Alcance	200 m	55 m <sup>1</sup>
Tensão de alimentação	10 a 30 V	Compressor - 12 V Transdutor - 5 V
Comunicação	4-20 mA	0.5 a 4.5 V
Precisão	0.5 % FS	2 % FS
Preço	110.48€ <sup>2</sup>	Tubo - 32.86€ Compressor - 26.79€ Transdutor - 12.87€

Nota 1: O alcance do sensor depende de vários fatores, sendo que o principal contribuinte é o comprimento vertical do tubo. Os limites de pressão do transdutor e de compressão do compressor também devem ser considerados.

Nota 2: Cabo com apenas 25m, sendo que um exemplar equivalente para realizar a extensão não foi encontrado. Costuma ser de construção pouco comum pois contem 2 condutores de cobre e um micro-tubo para realizar a referência do sensor em relação à atmosfera.

### Válvulas atuadoras

De forma a controlar o fluxo e destino do caudal de fluido, teve-se a necessidade de implementar um método que permite realizar alguma atuação sobre um comando.

Na tabela 3.18 apresentam-se alguns candidatos possíveis para controlar a passagem da água.

Tabela 3.18: Métodos possíveis para controlar a passagem da água[43, 44, 45, 46]

Válvula	Membrana	Pilotada a ar	Motorizada	Servo-motorizada
Tensão de alimentação	24 VAC	Atuador: 24 VDC	230 VAC	6 VDC
Tempo de atuação	< 5s	Ajustável	15 s	300 ms
Preço	22.86€	Válvula: 39.15€ Atuador: 11.75€	32.58€	Válvula: 8.00€ Atuador: 9.02€

Dois requisitos para a escolha da válvula de controlo são, o diminuto tempo de atuação e a possibilidade de realizar o corte de forma bidirecional.

Observando os candidatos apresentados, uma das melhores soluções aparenta ser a válvula de membrana, uma vez que realiza a sua atuação de forma relativamente rápida e apresentar um custo baixo. No entanto, devido à sua construção interna, estes mecanismos de controlo apenas operam com o fluido a circular numa direção, sendo que na eventualidade deste ser invertido, a válvula perde a capacidade de realizar uma atuação, falhando de forma aberta.

Relativamente ao segundo candidato, as válvulas pilotadas a ar comprimido, existem alguns pontos negativos que levam à desconsideração das mesmas. Primeiramente, são de alto custo. Em segundo lugar, o atuador pneumático que

realiza o seu controlo também é bastante caro e, por fim, estas válvulas requerem um compressor de ar comprimido.

As válvulas motorizadas aparentam ser a primeira solução capaz de realizar um controlo como pretendido, de forma bidirecional, sem que sejam de elevado custo. No entanto, de todas as soluções apresentadas, são as que demoram mais tempo a realizar uma atuação. Devido ao seu sistema interno de fins de curso de controlo do motor, a válvula requer o término do ciclo de abertura, antes de poder iniciar um ciclo de fecho e vice versa. Ou seja, caso um dispositivo de controlo inicie uma operação de abertura da válvula, sendo que momentos depois decida fecha-la, a válvula só vai concluir a operação de movimentação passados 30 segundos.

O último método apresentado, consiste na aplicação de um servo-motor numa válvula de esfera convencional. Esta abordagem aparenta ser a de mais baixo custo sendo que fornece um alargado controlo, no entanto, é de complexo desenvolvimento.

## **3.2 Conclusão do Capítulo 3**

Neste capítulo foram abordados diversos temas sobre a constituição física do projeto, assim como as suas necessidades. Efetuou-se um estudo e comparação de diferentes soluções possíveis apresentando um processo discriminatório que destacasse a solução mais enquadrada na resolução do problema.

## Capítulo 4

---

# Hardware dos Controladores

---

*Neste capítulo são apresentadas as soluções de hardware implementadas através de diversos módulos, de forma a permitir realizar a recolha de informação através de diversos sensores, assim como realizar a atuação de dispositivos. Estes dispositivos comunicam com uma máquina central, frequentemente denominada apenas como servidor, que comanda todo o sistema*

### 4.1 Hardware do Controlador Remoto da Eletrobomba

Neste controlador foi implementado condicionamento de sinais digitais, a optoacopladores, que permite ao microcontrolador perceber o estado lógico de sinais alternados de 230 V. Além disso, existe também o condicionamento do sinal analógico de corrente, sendo disponibilizado espaço para montagem de um ACS712 ou um ACS750. O sinal de tensão, foi condicionado com recurso a um circuito discreto com amplificadores operacionais. Em termos de atuação, existem 3 saídas a relés, no entanto, apenas uma é utilizada.

Em termos de comunicação com o exterior, é possível incorporar um HC-12, um módulo de conversão UART-RS485, ou outro dispositivo qualquer, sendo que se encontram disponibilizadas ligações aos pinos de comunicação. Acabou-se por utilizar um ESP8266, sendo que está montado num placa própria, junto de um circuito de condicionamento de sinal e de um regulador de tensão, estando encaixado sobre o controlador remoto, fornecendo-lhe conectividade com o servidor.

Em termos de visualização da informação, foi implementado um LCD-20x04 com comunicação por 4 linhas. Relativamente à navegação, foram implementados dois botões retro-iluminados.

O esquema do controlador remoto da eletrobomba encontra-se representado no anexo A, o desenho da placa de circuito impresso encontra-se no anexo B.

Os aspetos mais importantes referentes ao desenvolvimento e implementação do *hardware*, encontram-se discutidos nos pontos em seguida.

#### 4.1.1 Condicionamento de sinais digitais de tensão alternada

O esquema utilizado para realizar o condicionamento de sinais digitais de tensão alternada encontra-se representado na figura 4.1. O principal constituinte é um isolador ótico, responsável por isolar galvânicamente um sinal, mas também transmitir o seu estado lógico. Visto que o transmissor do isolador ótico é um LED, foi necessário apropriar as características do sinal que chega ao dispositivo.

De forma a realizar a atuação do LED do isolador ótico, desenhou-se o circuito para que fosse percorrido, por ele, uma corrente na ordem dos 7 mA. Dado que era necessário realizar uma queda de tensão na ordem dos 225 V, a utilização de uma resistência convencional resultaria na perda de energia por efeito de Joule na ordem dos 1.6 W. Para mitigar esse problema, e uma vez que o sinal é de característica alternada, fez-se uso do comportamento de um condensador sobre um sinal alternado.

Utilizando a lei de Ohm, determinou-se a impedância necessária que tem de ser fornecida pelo condensador, sendo esta de cerca de 31.8 k $\Omega$ . Para determinar a capacidade do condensador, utilizou-se a formula representada em 4.1, sendo que  $X_c$  é o valor da impedância necessária e que  $f$  é o valor da frequência da rede, 50 Hz. Com estes parâmetros, determinou-se que o valor do condensador ronda o valor padrão de 100 nF.

Foi colocada uma resistência de elevado valor em paralelo com o condensador, encarregue de descarregar alguma tensão presente no componente, caso ocorra uma interrupção na circulação da corrente. Devido ao valor deste dispositivo ser de mais de uma ordem de grandeza superior à impedância apresentada pelo condensador, a sua influencia no valor da corrente foi negligenciado.

Para realizar a retificação do sinal, utilizou-se um par de díodos que realiza uma retificação de meio arco, sendo que o segundo dispositivo apenas serve para conduzir a corrente de forma segura na arcada invertida. Este passo é importante para proteger o LED de tensões inversas muito elevadas. Um dos díodos é de Zener para proteger o circuito contra surtos.

$$X_c = \frac{1}{2\pi fC} \quad (4.1)$$

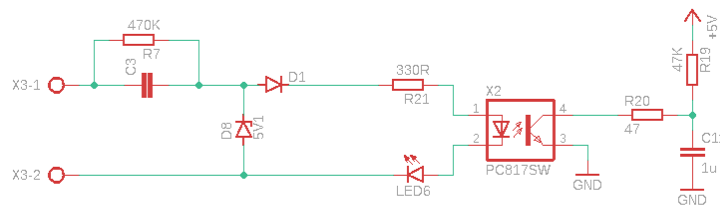


Figura 4.1: Condicionamento de sinais digitais de tensão alternada

#### 4.1.2 Condicionamento do sinal analógico de tensão alternada

O circuito de condicionamento de sinal (Fig. 4.2) é composto por um amplificador inversor com adição de *offset*, responsável por aplicar uma atenuação ao sinal da rede e um deslocamento em relação à referência.

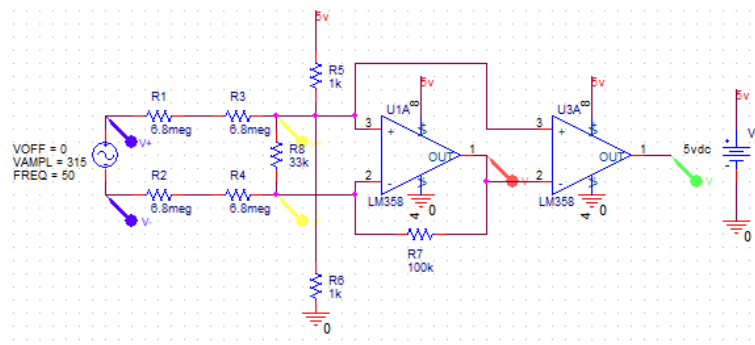


Figura 4.2: Condicionamento do sinal analógico de tensão alternada

O valor do ganho do amplificador pode ser calculado segundo a formula 5.1, sendo que o seu valor é de  $1/272$  V/V. Estes valores foram escolhidos de forma a que acomodassem um valor de entrada máximo de 250 V, uma vez que o sinal na saída do amplificador passa a ser um sinal alternado de amplitude 1.3 V, com valor médio de 2.5 V. O valor de pico máximo e mínimo é de 3.8 V e 1.2 V, respectivamente, estando dentro dos limites do ADC do microcontrolador e, não entrando na região de saturação do AMP-OP.

O valor escolhido para as resistências R1 a R4, encontra-se na ordem de grandeza do mega ohm, permitindo um isolamento entre as linhas da rede, e o resto do circuito, de cerca de  $13.6$  M $\Omega$ , o que resultaria na circulação de uma corrente negligente de cerca de  $16$   $\mu$ A, caso houvesse um caminho entre a terra e o circuito desenvolvido. Esta mesma corrente de fuga caracteriza o dispositivo como um sistema de classe II e. O valor de fuga é mais de uma ordem de grandeza inferior ao limite máximo estabelecido pela norma internacional IEC-950 [76].

$$\frac{V_{Out}}{V_{In}} = \frac{R_7}{R_1 + R_2 + R_3 + R_4} \quad (4.2)$$

O amplificador U3A, compara o valor de tensão à saída do primeiro amplificador, com a tensão média deste sinal (2.5 V), determinando assim o momento exato em que a tensão da rede muda de quadrante. Este resultado é aplicado numa interrupção do microcontrolador e usado, pelo código, para realizar temporizações. O valor da frequência da rede foi confiado como uma boa fonte de relógio, em virtude da sua frequência de oscilação ser bastante estável.

## 4.2 Hardware do Controlador Remoto dos Sensores de Sobrecarga e Pressões

No anexo C é possível encontrar o esquema elétrico do controlador remoto dos sensores de sobrecarga e de pressões. No diagrama, está apresentado o estágio de controlo do sensor de sobrecarga, que permitem ativá-lo e desativá-lo, assim como o estágio de amplificação do sinal resultante deste sensor.

Foi também implementado um circuito com um segundo microcontrolador, um ATtiny84, que recolhe alguns sinais do circuito hidráulico. Existem dois pressostatos que detetam pressões em locais do sistema hidráulico, sendo que fornecem resultados digitais, isto é, sempre que a pressão a que estão sujeitos ultrapassa um nível predefinido no parafuso de configuração do pressostato, o seu contacto interno abre. Este microcontrolador também lê valores de um transdutor de pressão analógico que recolhe valores do sistema do hidropressor da habitação,

O ATtiny84 envia por UART, todos os dados recolhidos por si, para o controlador principal deste módulo baseado num ESP8266. Toda a informação é enviada como resposta ao servidor, sempre que este pedir. Em cada pedido, é transmitida informação destinada ao controlo da eletroválvula e do sensor de deteção de sobrecargas.

### 4.2.1 Controlo do sensor de sobrecarga e amplificação do sinal resultante

Como o sensor de sobrecarga, para detetar água, faz circular uma pequena corrente pelo fluido, inevitavelmente irá ocorrer uma corrosão galvânica dos elétrodos. De forma a adiar esse problema, decidiu-se implementar um circuito que permita desligar o sensor, evitando a corrosão prematura dos elétrodos.

Na figura do anexo C, no circuito delimitado pela caixa assinalada com (1), constata-se o circuito responsável por ativar e desativar o sensor. Este circuito recebe uma ordem de atuação proveniente do ESP8266.

Sempre que se verificar um estado lógico alto neste sinal, é colocado, pelo circuito em (1), uma tensão de cerca de 12 V num dos elétrodos. Na presença de

um estado lógico negativo no sinal de comando, o potencial do eletrodo ligado a este estágio do circuito, é trazido a 0 V.

No circuito delimitado a (2), encontra-se o estágio de amplificação do sinal em corrente que atravessa os eletrodos. Esta corrente apenas é significativa com a passagem de água, havendo um fecho do circuito entre os eletrodos alterando-se o valor de tensão à saída do circuito de amplificação.

Caso o controlador dê uma ordem para desativar o sensor, ambos os eletrodos são trazidos ao mesmo potencial, de 0 V, pelas resistências de *pulldown* de cada. Em consequência de não haver potencial nos eletrodos, não há possibilidade de existir corrente a circular no circuito, eliminando o problema da corrosão galvânica.

Note-se que o sensor está sempre ativo, no entanto, na presença de água, é desligado ao fim de 5 segundos de submersão, sendo reativado passados 2 minutos, repetindo-se o processo. Isto proporciona que, na presença de água, os eletrodos estejam ativados cerca de 4% do tempo em que estão submersos.

#### **4.2.2 Teste do sensor de sobrecarga**

Uma vez que este sensor é a última linha de defesa contra o desperdício de água, é necessário ter a certeza do seu bom funcionamento. Para tal, implementou-se um método que permite realizar um teste prático ao sensor.

Este teste consiste na injeção de uma pequena quantidade de água à montante do sensor, verificando-se o seu comportamento. Para tal, foi montada uma eletroválvula no circuito hidráulico destinada a exercer esta tarefa. O comando da eletroválvula provém do circuito em (3), que é atuado consoante o estado lógico de um sinal, proveniente do ESP8266.

### **4.3 Hardware do Controlador Remoto de Recolha de Informação Meteorológica**

O esquema do controlador remoto de recolha de informação meteorológica encontra-se apresentado no anexo D. O mesmo permite interligar os diferentes sensores com o ESP8266, assim como acomodar também uma fonte de alimentação que permite alimentar o circuito.

Esta placa encontra-se montada dentro de uma caixa resistente aos elementos atmosféricos. Na sua parte superior, foram realizados dois furos que foram tapados com um vidro, de forma a criar uma janela. Esta janela é transparente desde a região infravermelha, até à região ultravioleta do espectro eletromagnético. No entanto, impede a entrada de água da chuva para o seu interior. De

forma a poder determinar as condições ambientais a que o controlador está sujeito, montou-se um sensor de temperatura e humidade, o DHT-11, que permite medir tal parâmetros no interior da caixa.

#### 4.3.1 Sensores da radiação solar visível e ultravioleta

Na parte inferior de cada orifício, foram colocados os sensores de medição da radiação ultravioleta e de medição da intensidade solar visível. O sinal analógico resultante do sensor ultravioleta é atenuado pelo divisor resistivo implementado por um par de resistências. Esta divisão da tensão é importante, pois o conversor analógico para digital do ESP8266 apenas permite medir uma tensão analógica máxima de 1 V. O sensor de intensidade solar comunica por I2C.

#### 4.3.2 Sensores de temperatura, humidade e pressão atmosférica

Os sensores de temperatura, humidade e pressão atmosférica encontram-se montados dentro de um invólucro plástico que fornece proteção mecânica. No entanto, permite a passagem de correntes de ar pelas suas paredes. Estes sensores encontram-se montados dentro de dois escudos que permitem a livre circulação de correntes de ar, mas protegem os sensores contra outros elementos. O escudo exterior é um escudo térmico que impede que a radiação solar atinja os sensores, aquecendo-os e medindo valores errados. Já o escudo interior impede que a água da chuva e correntes de ar quentes geradas pelo escudo exterior, atinjam os sensores. Um cabo de quatro linhas fornece alimentação e comunicação I2C aos sensores, provenientes do PCB.

#### 4.3.3 Sensores de precipitação

O sensor de precipitação realiza uma medição da precipitação em incrementos de cerca de 0.3 mm por impulso. Estes impulsos provém do fecho e conseqüente abertura de um interruptor eletromecânico operado por um campo magnético, decorrente da mudança brusca da inclinação do sistema de baldes no seu interior.

Sempre que o contacto fecha, a linha denominada no esquema elétrico como *RAIN*, é colocada a 0 V, ativando uma interrupção no microcontrolador. Esta linha é forçada a 3.3 V, por uma resistência de *pullup* configurada internamente no ESP8266.

## 4.4 Hardware do Controlador Remoto do Sistema de Rega Auxiliar

O controlador remoto do sistema de rega auxiliar é um sistema que permite realizar a configuração de temporizadores calendarizados à semana, hora e minuto, que permitem criar programas para atuação de eletroválvulas de rega. Este módulo tem a capacidade de trabalhar de forma independente, sem necessitar de conectividade à rede Wi-fi nem ao servidor, pelo que, nesse caso, qualquer alteração que se pretenda realizar, vai necessitar de um utilizador presente no local a interagir com o sistema.

O esquema do controlador encontra-se presente no anexo E. No anexo F encontra-se uma imagem do prototipo da placa de circuito impresso.

Neste diagrama encontram-se especificas as ligações do microcontrolador utilizado para realizar todo o controlo do sistema, sendo este um ATmega328p. Toda a informação e parâmetros configuráveis podem ser observados através de um LCD-2004 que comunica com o microcontrolador por 4 linhas. Para realizar a navegação e a reconfiguração de valores seleccionáveis, dedicaram-se 3 botões para este propósito. Todos os dados alterados e gravados, são guardados na memória programável apagável apenas de leitura (EEPROM), sendo que a cada reinício do sistema, verifica-se a integridade desta informação, procedendo-se seguidamente ao seu carregamento para a memória RAM.

Em termos de atuadores é possível controlar até 4 dispositivos de forma independente, sendo que se dedicou quatro contactos normalmente abertos, provenientes de relés comandados pelo ATmega328p. Aquando o fecho de um contacto de um relé, é atuada uma eletroválvula de 230 VAC.

De forma a permitir implementar conectividade com outros dispositivos, dedicou-se espaço para poder implementar um conversor de UART para RS-485. Dedicou-se também espaço para permitir a montagem de um HC-12 ou de outro dispositivo de comunicação sem fios, fornecendo-se alimentação de 5 V e comunicação UART. No lugar do conversor UART para RS-485, foi montado um sensor de temperatura e humidade atmosférica. Para determinar as condições de funcionamento do sistema, implementou-se também um sensor de humidade do solo e um LED que permite ao utilizador determinar o estado de funcionamento do circuito.

No presente desenvolvimento, foi utilizado um ESP8266 para permitir a transmissão de informação com o exterior. A sua única tarefa, é receber informação proveniente da rede Wi-Fi e, enviá-la por UART para o microcontrolador. O percurso inverso também é realizado, sendo que em vez de receber uma ordem, é enviada uma resposta com informação sobre o controlador.

#### 4.4.1 Relógio de tempo real

O dispositivo responsável por manter, com elevada precisão, a data e a hora, é o RTC DS3231. Este circuito integrado é consultado e configurado com recurso ao protocolo de comunicação I2C. A alimentação principal deste dispositivo provém da regulação e retificação da rede elétrica, sendo que no caso da sua interrupção, o RTC muda automaticamente a sua alimentação para uma pilha de reserva, permitindo assim continuar a estar atualizado, mantendo a contagem do tempo.

Este dispositivo é crucial pois a informação fornecida por este é comparada com os programas de rega gravados na memória do microcontrolador, sendo que o resultado obtido pode condicionar o estado de atuação das eletroválvulas, caso o sistema esteja no modo automático.

#### 4.4.2 Indicador luminoso

Aplicou-se no circuito um LED de indicação que pode indicar até quatro estados de funcionamento do sistema. Estes estados encontram-se descritos em seguida:

- Sem atividade luminosa - a falta de atividade do LED indica a pouca atividade do sistema, ou seja, não existem válvulas abertas e não existem transmissões de informação;
- Impulso de curta duração - sempre que é gerada uma resposta a um comando recebido, tal operação é sinalizada com um rápido impulso luminoso;
- Impulso de longa duração - os impulsos luminosos longos indicam que existe pelo menos uma válvula aberta;
- LED ligado continuamente - nesta situação, é realizada a indicação, por parte do sistema, que existe uma situação de anomalia. Esta anomalia pode ter origem na falta de coerência dos dados da EEPROM ou na avaria de um sensor.

Note-se que a intensidade do indicador luminoso é controlável, embora apenas pelo servidor.

#### 4.4.3 Sensor de humidade do solo

Na figura 4.3 encontra-se apresentado o circuito desenvolvido para realizar medições sobre o valor da humidade do solo. O sensor consiste num cabo elétrico de duas linhas com um comprimento de cerca de 5 metros, que se encontra estendido de forma circular sob a superfície do solo.

De forma a possibilitar a medição da humidade do solo, por parte do microcontrolador, desenvolveu-se um circuito capaz de relacionar o valor da capacidade



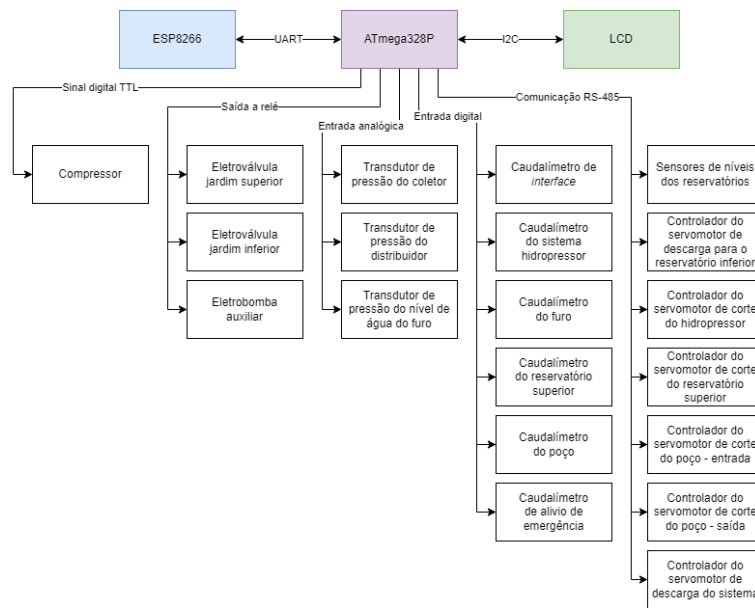


Figura 4.4: Arquitetura do *hardware* do controlador remoto principal

Em termos de controlo, encontram-se disponíveis saídas a relé que permitem realizar o controlo de duas eletroválvulas de membrana, dedicadas para a rega dos espaços verdes principais, assim como o controlo do compressor do sistema de medição do nível da água do furo.

Este sistema realiza comunicações com alguns controladores subordinados, sendo que um recolhe informação sobre os níveis de água dos reservatórios, já os restantes atuam algumas válvulas de esfera implementadas mas não montadas no sistema. Estas transferências de dados é efetuada sobre o protocolo RS-485.

Em termos de comunicação com o servidor, utilizou-se novamente o método usado no controlador remoto da eletrobomba, sendo que se tira partido das capacidades de um ESP8266 para realizar a interface das comunicações.

#### 4.5.1 Hardware das válvulas servo-motorizadas

Como foi referido anteriormente, desenvolveu-se um módulo que permitisse controlar umas válvulas de esfera do sistema hidráulico. O principal constituinte, o motor, tem incorporado um sistema de rodas dentadas que permitem reduzir a velocidade de rotação do eixo de saída, aumentando o seu binário. No eixo está montado um potenciómetro que permite determinar o seu ângulo.

Para realizar a ligação mecânica entre a válvula de esfera e o motor desenhouse e imprimiu-se com recurso a uma impressora 3D, o objeto apresentado em 4.5. Note-se que foi adquirido um servo-motor convencional, mas o circuito de controlo foi substituído.

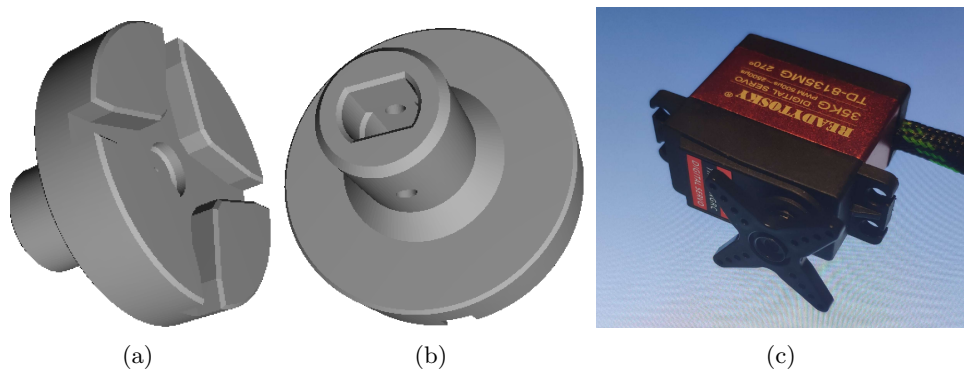


Figura 4.5: Adaptador entre o eixo da válvula e do motor (a) Visto de frente (b) Visto de traseira e, (c) Motor

Na imagem G apresenta-se o esquema implementado de cada módulo de controlo do motor, no anexo H encontra-se uma captura do protótipo da placa de circuito impresso desenvolvida.

Este dispositivo é controlado com recurso a um circuito integrado que contém uma ponte H no seu interior, sendo que apenas requer do microcontrolador, o ATtiny84, dois sinais que indicam a direção de rotação do motor.

O sinal de retorno da posição é recebido diretamente pelo microcontrolador, vindo do potenciômetro encastrado no eixo de saída do atuador. A alimentação do atuador provém de um regulador comutado, o LM2596, sendo que foi disponibilizada uma ligação que permite controlar o seu estado de funcionamento, uma vez que, na prática, apenas é necessário que este funcione quando for realizada uma atuação do motor, podendo ser desligado quando não há nenhum uso, economizando-se assim energia.

## 4.6 Conclusão do Capítulo 4

Neste capítulo foi explicada a implementação projetada para os diferentes controladores, assim como foi realizada uma descrição dos problemas e das soluções desenvolvidas para realizar o condicionamento de sinal ou atuação de dispositivos.



## Capítulo 5

---

# Software dos Controladores Remotos

---

*Neste capítulo são apresentadas as soluções de software desenvolvidas para cada controlador remoto, fazendo com que cada dispositivo opere de modo desejado*

Todos os controladores, aquando energizados, retornam, da sua memória interna, as credenciais sobre a rede Wi-fi a que se devem ligar e informação sobre o servidor de transporte de telemetria de filas de mensagens (MQTT). Logo em seguida, tentam ligar-se à rede LAN e consecutivamente ao servidor.

O principal método de funcionamento da transmissão de informação ocorre sobre a ideologia *master-slave*, isto é, é retornada para o servidor uma resposta por parte dos controladores somente quando houver um pedido para tal, ou um comando a que devem obedecer. Este método permite ao servidor obter informação e enviar comandos apenas quando é necessário, reduzindo o tráfego desnecessário e a sobrecarga da rede. No entanto, tal processo não é regra. Sempre que um controlador determinar que deve de atualizar o servidor de imediato, é despoletada uma transmissão de informação, sem ter havido por parte do servidor tal intenção.

### 5.1 Software do Controlador Remoto da Eletrobomba

O quadro que controla a eletrobomba tem dois modos de operação distintos, o modo manual e o modo automático. Estes modos só podem ser alterados fisicamente, mudando o estado de um interruptor de 3 posições e são completamente independentes do modo de funcionamento do sistema geral.

O modo manual permite controlar o estado da bomba, realizando uma sobreposição sobre os comandos do servidor, ignorando-os. Na eventualidade do

interruptor selecionador estar a selecionar o modo automático, toda a atuação é realizada pelo controlador remoto, sendo que as ordens de operação tem origem no servidor.

### 5.1.1 Medição de valores analógicos

Existem 4 valores analógicos recolhidos e tratados pelo microcontrolador, sendo que dois são o valor da tensão e da corrente da rede elétrica, os restantes dependem de sinais digitais dos botões de interface do controlador. Os botões são analisados de forma analógica, sendo que se pretende obter estado digital (ligado ou desligado), porque os pinos utilizados no microcontrolador que interligam a estes dispositivos, não tem ligação a nenhum *hardware* interno, apenas ao multiplexador analógico.

No diagrama da figura 5.1 encontra-se representado o código executado dentro da interrupção despoletada no fim de uma conversão do ADC. Note-se que sempre que se muda o canal do ADC, a primeira conversão costuma demorar mais tempo, no entanto, como cada período de recolha de amostras demora 40 milissegundos, mas apenas se guardam valores nos últimos 20 milissegundos, a influência desta e outras inconveniências são mitigadas. Todos os valores recolhidos, são guardados num *array*, sendo que sempre que se guarda um novo valor, incrementa-se um valor que aponta para a localização desta última leitura.

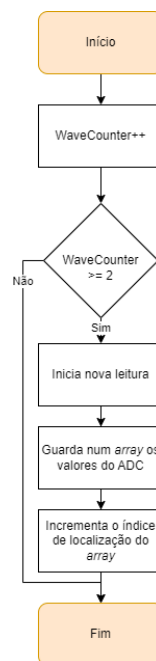


Figura 5.1: Diagrama do código do controlador remoto da eletrobomba da interrupção do ADC

### 5.1.2 Interrupção temporizada pela rede elétrica

Na figura 5.2 apresenta-se um diagrama simplificado do código executado na eventualidade de existir uma interrupção externa. Relembre-se que esta interrupção tem origem no detetor da mudança de quadrante do sinal da tensão, abordado na secção 4.1.2.

Configurou-se o disparo de uma interrupção sempre que o sinal analisado mudar de estado. Uma vez que a frequência da rede elétrica nacional é de 50 Hz, o período de cada meio ciclo dura 10 milissegundos, assim como a cadencia de cada interrupção.

No presente código desenvolvido, existe a computação de resultados recolhidos e a alteração da entrada analógica convertida, a cada 40 milissegundos. Sempre que o contador que realiza esta desmultiplicação, o *Wavecounter*, atingir o seu teto, é limpo o seu registo, e iniciado o cálculo da conversão dos valores recolhidos pelo ADC, para um valor eficaz. Relembrando que no condicionamento de sinal, é somado um *offset* de 2.5 V ao sinal alternado, o primeiro passo no cálculo do valor eficaz é realizar uma remoção da componente DC, visto afetar o resultado final erradamente. A formula utilizada para realizar o cálculo do valor eficaz (RMS) encontra-se apresentada em 5.1[77], sendo que  $n$  é o número de amostras analógicas recolhidos e  $x_i$  representa cada amostra.

$$RMS = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} x_i^2} \quad (5.1)$$

Existe um contador que identifica o sinal analógico que está a ser recolhido, podendo variar entre, valor da tensão, corrente, botão de *start* e *stop*. Após o cálculo do valor RMS, verifica-se através de um conjunto de operações lógicas, a que variável pertence o valor calculado.

Na eventualidade do registo que aponta para o valor a ser analisado apontar para a tensão, altera-se de imediato o seu valor para analisar o valor da corrente no ciclo seguinte. Todos os valores RMS da tensão são guardados num *array* circular, ou seja, o novo valor recolhido é escrito sempre por cima do valor mais antigo guardado em memória.

Caso o registo que aponta para o valor a ser analisado apontar para a corrente, altera-se o seu valor para analisar o estado do botão de *stop* no ciclo seguinte. Em seguida, é executado um algoritmo semelhante ao descrito no paragrafo anterior, guardando o valor RMS calculado recentemente, num *array* circular que guarda os valores da corrente.

Considera-se que o sistema está em repouso na eventualidade da retro-iluminação do visor estar apagada. Uma operação possível pelo utilizador é ligar apenas a

retro-iluminação para consultar os valores apresentados, tal operação apenas é possível realizar carregando num dos botões de interface, note-se que não é pretendida realizar nenhuma operação de arranque ou paragem da bomba, apenas ligar o *display*. Para tal, implementou-se um algoritmo que determina se o sistema está em repouso ou ativado, pelo que se existir uma alteração do estado de um botão, primeiramente o sistema é colocado no modo ativado e, apenas na presença de uma segunda operação, é que é registada a presença de atividade nos contactos.

Na eventualidade de um botão ser pressionado, lembrando que estão ligados ao plano de massa, o valor medido pelo ADC passa a ser próximo de nulo. Caso esta condição se verifique, o sistema é colocado num modo pendente se estiver previamente em repouso, isto é, indica-se que houve algum evento no sinal analisado, e que se está a espera que o botão seja solto para que o sistema seja alterado para o modo ativo.

Retornando-se ao fluxograma, se o sinal em análise for o sinal de retorno do botão de *stop*, muda-se o registo de indicação de análise para o botão de *start*, mais uma vez, indicando-se o novo sinal a analisar no ciclo seguinte. Em seguida, caso o sistema esteja ativado, inicia-se o incremento de um contador de forma a medir o tempo de pressão deste botão. Quando o botão for solto, analisa-se o tempo total de pressão com um valor predefinido, na eventualidade de ser inferior, assinala-se um simples impulso, caso o resultado contrário se verifique, assinala-se um *longpress*, isto é, indica-se ao controlador que se pretende aceder aos seus sub-menus.

Caso o registo apontador para o sinal em análise apontar para o botão de *start*, o valor deste registo é mudado indicando que no ciclo seguinte deverá de ser analisado novamente, o valor da tensão, reiniciando-se todo este processo. Em termos de decisão de eventos no botão, o algoritmo é semelhante ao anterior, no entanto, não existe medição de *longpress*.

O registo que determina se o sistema está no modo de repouso ou ativado, é decrementado a cada 40ms sempre que for superior a 0, sendo que quando atingir o valor nulo, o sistema é colocado novamente em repouso, retornando-se para o menu principal e apagando-se a retro iluminação do *display*.

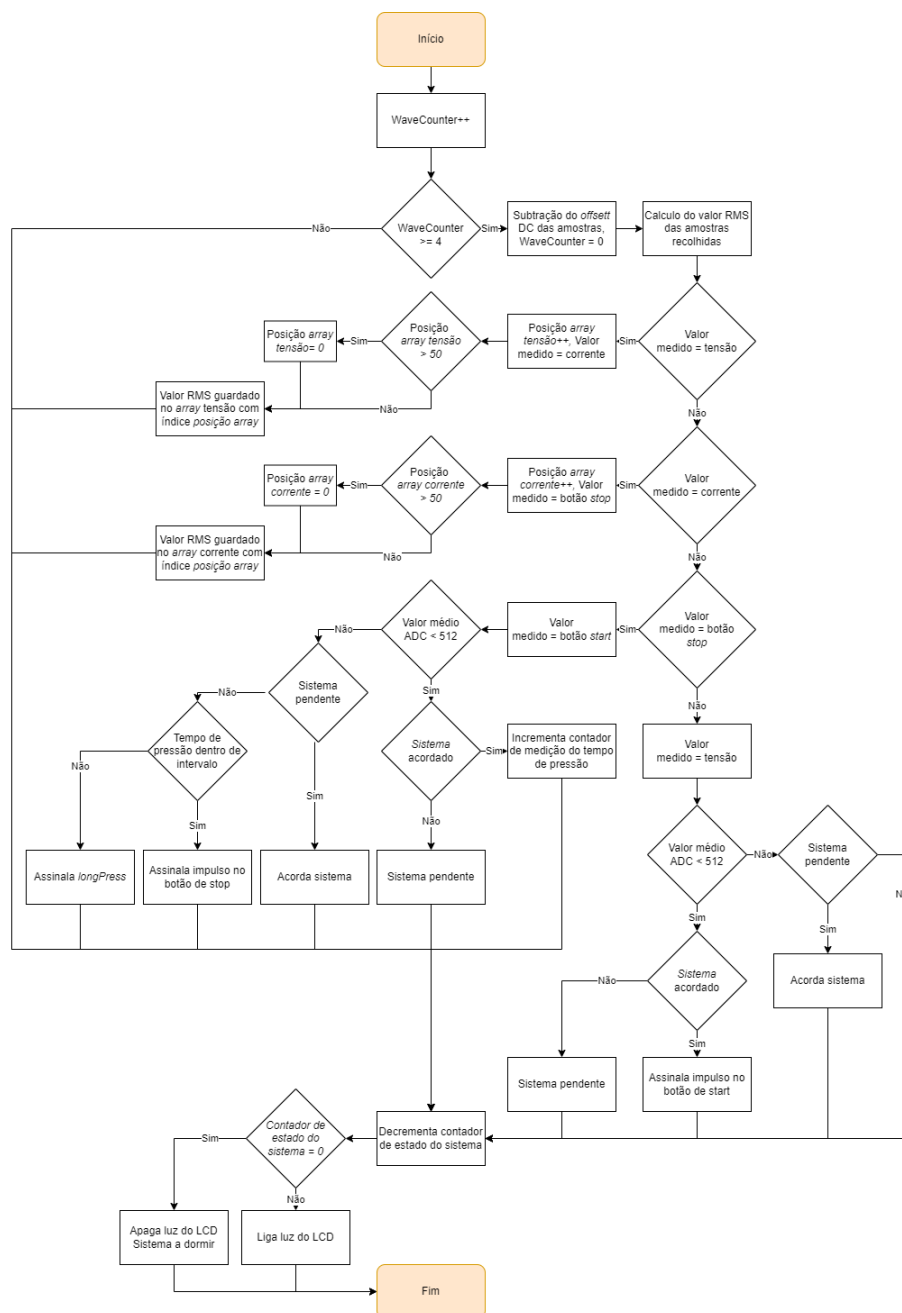


Figura 5.2: Fluxograma simplificado do tratamento dos valores recolhidos pelo ADC

### 5.1.3 Detecção do estado de sinais digitais e contagem do número de arranques

Foi implementado dentro da interrupção externa do microcontrolador um algoritmo que lê o estado digital de três entradas do dispositivo e, guarda o resultado

em memória. Com isto, é possível determinar se o quadro do controlador está no modo manual ou automático, se o disjuntor térmico de proteção da eletrobomba disparou, se o relé de nível detetou a falta de água e, conjugando alguns sinais, pode-se determinar se a bomba está em funcionamento ou parada.

O fluxograma que representa o algoritmo de contagem do número de arranques da eletrobomba encontra-se representado em 5.3. Para determinar o arranque ou paragem do dispositivo elétrico, utilizou-se os sinais provenientes do quadro de controlo, sendo que na presença de ambos os sinais, a bomba encontra-se em funcionamento, já no caso de ausência de um sinal, o dispositivo encontra-se parado.

Note-se que para um arranque ser contado apenas uma vez, necessita-se de guardar informação que diga que o contador já foi atualizado. Para tal, usou-se o mesmo registo de contagem, distinguindo-se os valores de contagem pares e ímpares em relação ao estado de funcionamento da bomba, isto é, caso a bomba esteja ligada e o valor do número de arranques for ímpar, incrementa-se o registo. Caso a bomba esteja parada, apenas é realizada uma adição enquanto o valor da variável do número de arranques for par.

Note-se que com este algoritmo, apenas se necessita de uma variável para realizar a contagem do número de arranques, no entanto o seu valor é sempre o dobro do valor real do número de arranques, pelo que tem de ser dividido por 2 antes de ser apresentado.

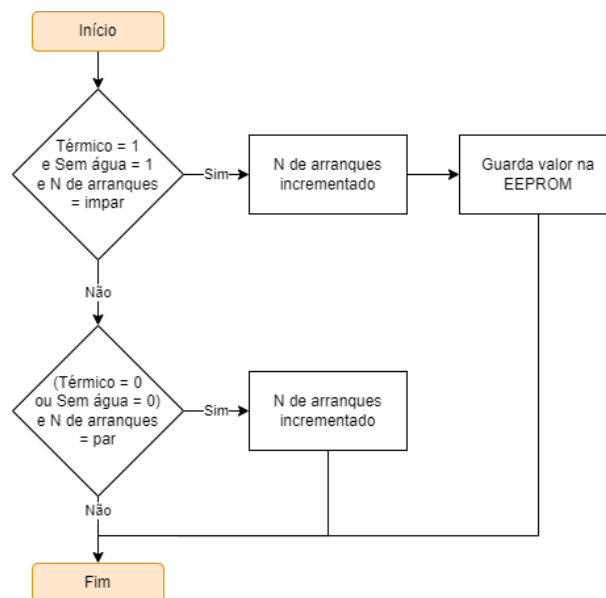


Figura 5.3: Contagem do número de arranques da eletrobomba

#### 5.1.4 Permissão de arranque e limite do número de arranques por período

Sempre que há um pedido de arranque do dispositivo, existem três regras importantes que têm de ser obedecidas para haver a sua aceitação:

- *cooldown* = 0 - a última paragem da bomba teve de ser há mais de um determinado tempo, no caso da implementação, este valor foi definido para 3 segundos. Esta função visa impedir paragens e arranques repentinos;
- Número de arranques no último minuto menor que 5 - o número máximo de arranques da bomba no último minuto não pode ultrapassar de 5 vezes, caso contrário terá de esperar um determinado tempo. Este algoritmo visa proteger a eletrobomba contra um elevado número de arranques e paragens num curto espaço de tempo;
- Número de arranques na última hora menor que 20 - o número máximo de arranques da bomba na última hora não pode ultrapassar de 20 vezes, caso contrário terá de esperar um determinado tempo, assim como o exemplo anterior. Esta função visa proteger a bomba contra um elevado número de arranques e paragens num curto espaço de tempo de forma a evitar desgaste prematuro e sobreaquecimento, valor sugerido pelo fabricante;

Na imagem 5.4 encontra-se o fluxograma do algoritmo implementado que visa responder às necessidades anteriormente expostas. Este código é executado sempre que houver a presença de um pedido de arranque da eletrobomba. Verificando-se primeiramente se a última paragem da bomba foi há mais de um determinado tempo predefinido.

A contagem do número de arranques por hora e por minuto é semelhante, sendo que existem dois *arrays*, um de 20 e outro de 5 posições que vêm uma das suas posições preenchidas com um valor, sempre que a bomba é ligada. No algoritmo de aceitação do pedido de arranque, é verificado em cada um destes *arrays*, se existe um espaço disponível, ou seja, se existe uma célula com o valor 0. Caso esta condição não se verifique, significa que o número de arranques por hora ou por minuto já foi atingido, sendo necessário esperar que uma posição fique livre.

Na eventualidade de todas as condições se verificarem, é indicado por meio de uma *flag*, a aceitação do pedido de arranque, sendo que o valor do registo de *cooldown* é preenchido com o número 3, sendo que é decrementado uma vez por seguindo quando a bomba for parada.

Para terminar a explicação deste algoritmo, é importante acrescentar que existem duas bases de tempo, uma que opera uma vez por segundo e outra que

opera uma vez por minuto, sendo que sempre que há a sua operação, os *arrays* que limitam a bomba a um determinado número de arranques, são percorridos e, todas as células que contem um valor diferente de 0 são decrementadas.

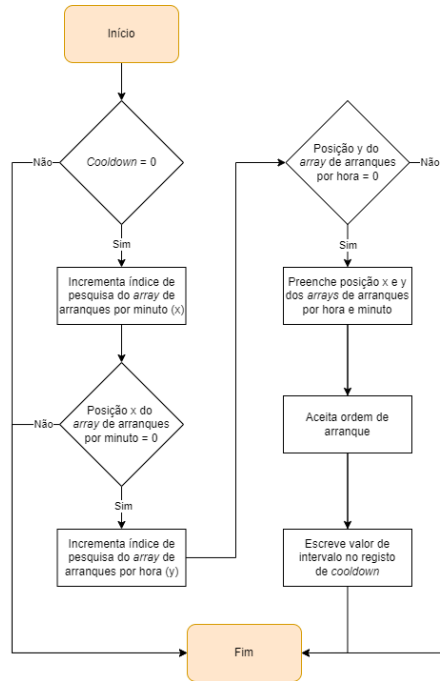


Figura 5.4: Permissão de um novo arranque da eletrobomba

Após o pedido de arranque da eletrobomba ser aceite, a sua atuação ainda vai depender de mais duas condições, apresentadas no fluxograma 5.5. A primeira condição é que a conexão com o servidor não pode estar impedida, tal é verificado pois espera-se que exista uma atualização da ordem de arranque por parte do servidor, a cada 300 milissegundos, portanto, na eventualidade da última atualização foi há mais de 1500 milissegundos, é sinalizada uma perda da comunicação, parando-se a bomba e considerando-se que o sistema está no modo *offline*.

A segunda condição necessária a ser obedecida é o valor da corrente consumida pela eletrobomba, que tem de ser superior a um mínimo e inferior a um máximo, previamente configurados, sendo que só pode sair destes limites durante um determinado tempo durante o arranque.

A falha do teste das condições de verificação de falha da comunicação, a verificação de um valor de corrente normal, assim como o alcance do número máximo de arranques por hora ou por minuto, despoletam a criação de uma anomalia.

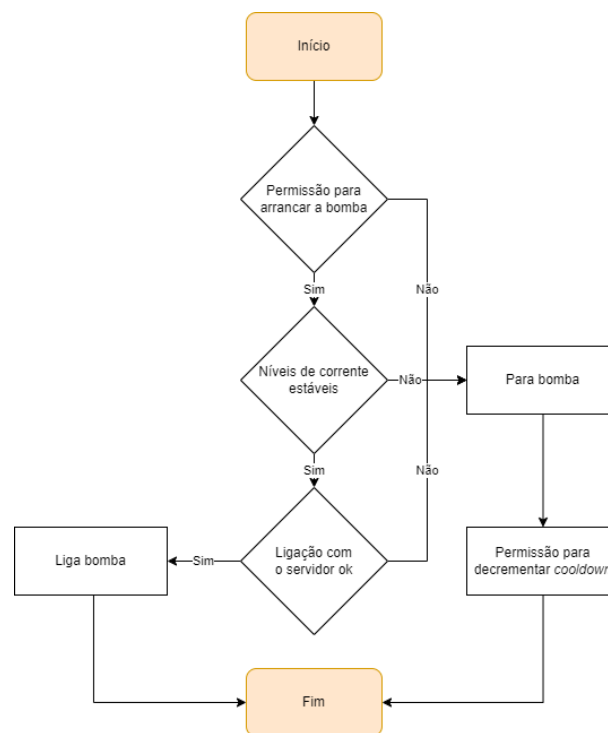


Figura 5.5: Exucução do arranque da eletrobomba

### 5.1.5 Comunicação com o ESP8266

Após uma trama enviada pelo ESP ser recebida pelo controlador remoto, é executado o código representado pelo fluxograma em 5.6. Primeiramente é calculada a verificação cíclica de redundância (CRC) da trama recebida, na eventualidade do resultado do cálculo realizado coincidir com o valor transmitido pelo microcontrolador de *interface* com a rede local, a informação recebida é considerado como válida, seguindo-se o processamento do comando.

Um comando tem sempre 5 caracteres de comprimento, sendo que os 3 primeiros indicam ao controlador remoto, o nível da água presente no furo, enviado um valor da altura com uma casa decimal. Em seguida, é enviado por meio de um byte, os estados que devem de ser colocados nos relés de saída. O último carácter indica que o servidor já reconheceu o estado dos botões de *start* e *stop*, ordenando que o seu estado deve retornar ao valor padrão.

Sempre que uma ordem for considerada como válida, um contador que realiza a contagem da idade da última receção, é restaurado com o valor 0. Este valor é importante para parar a atuação da bomba, caso o servidor não atualize o controlador num tempo inferior a um valor predefinido.

O pacote de dados de resposta do controlador remoto da eletrobomba é de tamanho variável e tem um formato semelhante ao da figura 5.7. Este pacote

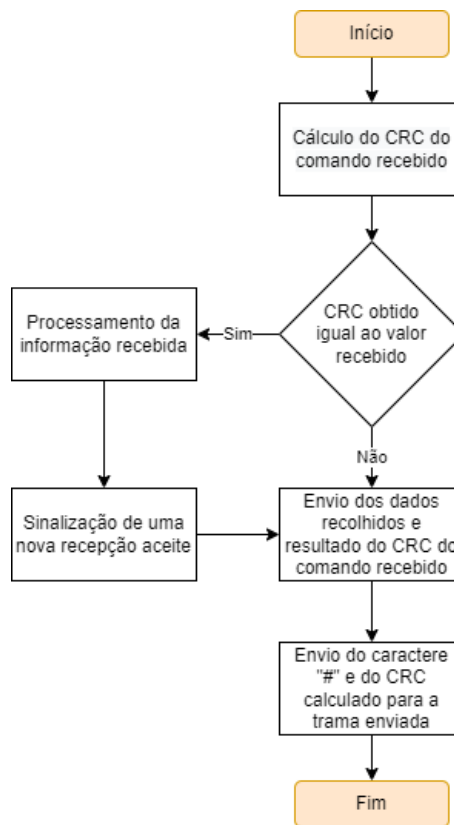


Figura 5.6: Processamento de um comando recebido e envio de uma resposta

encontra-se encapsulado dentro de um carácter de inicialização e outro de finalização, sendo que todas as variáveis são separadas pelo carácter “:”, ajudando na leitura dos dados por parte do ESP8266.

A verificação da integridade é calculada e enviada depois do último carácter. No campo de dados, são enviados todos os dados recolhidos pelo controlador, assim como dados determinados como anomalias, número de arranques, entre outros.

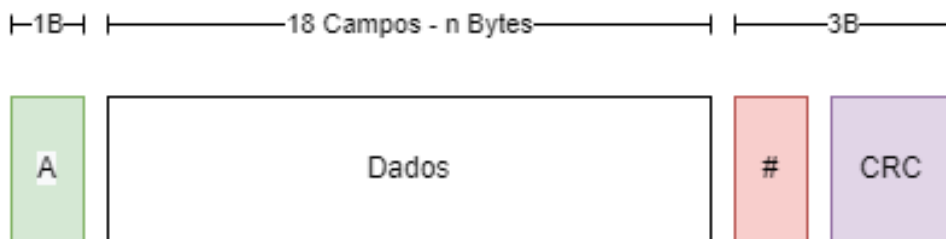


Figura 5.7: Trama de resposta do controlador remoto a um comando recebido

### 5.1.6 Software do ESP8266

A figura 5.8 apresenta a rotina de inicialização do ESP8266, sendo que primeiramente, inicia-se a UART com uma taxa de transferência de 9600 bps, seguindo-se da recolha da memória das credenciais da rede e do *broker* MQTT. Com estes valores, pode-se iniciar a ligação do dispositivo à rede Wi-fi e tentar estabelecer uma ligação com o servidor.

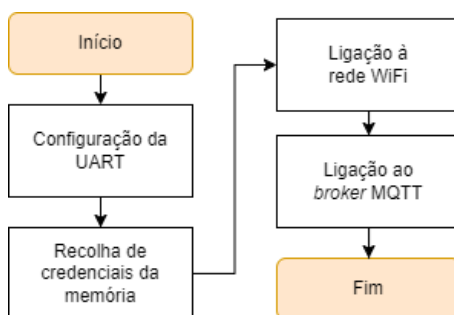


Figura 5.8: Código de configuração do ESP8266 durante o arranque

No fluxograma 5.9 é apresentado o ciclo principal executado pelo microcontrolador. Primeiramente é verificado se o ESP está no modo de configuração das credenciais ou no modo de execução normal. Caso o modo de configuração esteja ativado, o seu *hardware* de rede é reconfigurado como ponto de acesso, gerando uma *hotspot* Wi-fi ao qual um utilizador pode realizar uma ligação. Seguindo-se esta etapa, é apresentada uma página web estática que permite introduzir novas configurações, sendo que a sua apresentação só termina quando o utilizador submeter nova informação. Após a informação ser submetida é validada e guardada em memória, reiniciando-se a execução do *software* do dispositivo.

Normalmente o ESP8266 está com o modo de configuração desligado, pelo que é executado outro ciclo diferente. Primeiramente verifica-se se existe uma ligação ao *broker* MQTT, caso tal condição se verifique, é executado um ciclo de refrescamento da ligação, processando-se mensagens recebidas e enviando-se informação publicada. Na eventualidade da primeira condição não se verificar, é enviado um aviso para o controlador remoto da eletrobomba, iniciando-se um ciclo que tenta realizar uma reconexão a cada 5 segundos.

A comunicação realizada entre o ESP8266 e o ATmega328p é realizada utilizando o protocolo UART, sendo que este é bidirecional e pode operar de forma simultânea. Para tal foram implementados dois algoritmos distintos, um que realiza o envio de informação do ESP para o ATmega e um segundo algoritmo que realiza a transação inversa. Ambas as comunicações são verificadas com recurso a um cálculo de CRC de 16 bits, sendo que o seu valor é verificado aquando a receção da informação.

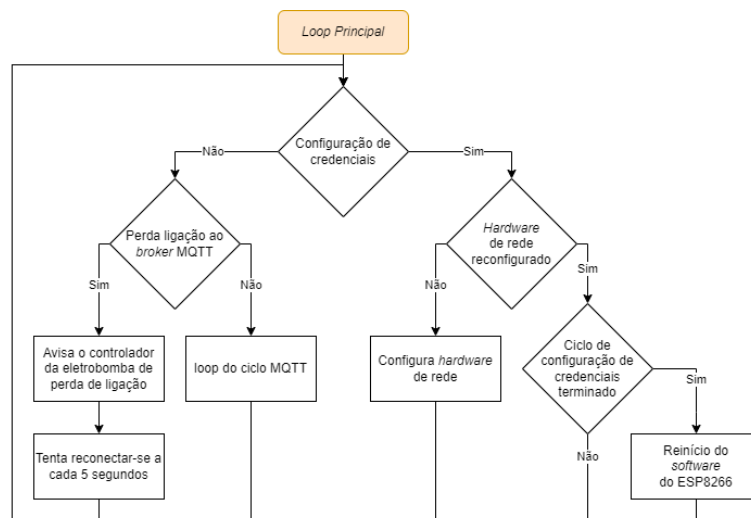


Figura 5.9: Ciclo principal do ESP8266 do controlador remoto da eletrobomba

Na figura 5.10 encontra-se um fluxograma que representa a implementação do ciclo de resposta a transações UART. Este algoritmo é despoletado por uma função que deteta eventos na receção de pacotes pela comunicação serie.

Primeiramente é realizada uma leitura da informação recebida, sendo que de seguida realiza-se uma verificação sobre o seu tamanho. Na eventualidade de serem recebidos 4 *bytes* e caso sejam precisamente “B#B#”, significa que o utilizador selecionou no controlador remoto a opção que ordenasse o ESP a entrar no modo de configuração das credenciais.

Na eventualidade desta condição não se verificar, espera-se pela receção de mais informação, até ser lido o carácter “#”, simbolizando o fim de uma transmissão por parte do remetente. Após este byte ser encontrado, é realizado o cálculo do CRC da informação recebida, sendo que é comparado com o CRC calculado pelo remetente para a mesma trama, sendo que ambos têm de coincidir. Caso esta condição se verifique, é testado para o valor do byte imediatamente antes de “#”, que simboliza o resultado da verificação do CRC da mensagem enviada pelo ESP para o controlador remoto da eletrobomba, se algum destes testes anteriores falhar, é realizado um pedido de retransmissão de informação ao ATmega328p, até que os CRCs da mensagem enviada e da mensagem recebida sejam corretamente validados em simultâneo. Nesta situação, é publicada uma resposta para o servidor.

A transmissão de dados do ESP8266 para o microcontrolador subordinado é realizada sempre que for recebida uma mensagem enviada através do *broker* MQTT, ou na eventualidade do ESP achar que deve de atualizar o controlador remoto com informação sobre o seu estado.

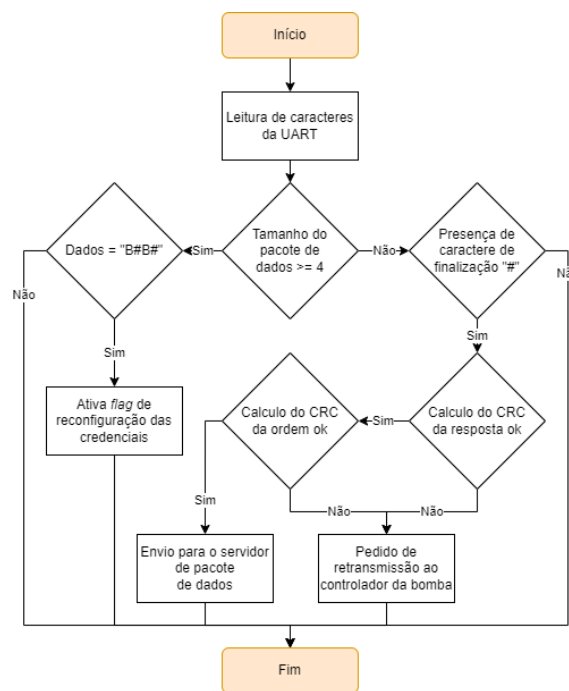


Figura 5.10: Transmissão de dados do controlador remoto da eletrobomba para o ESP8266

O estado do ESP é caracterizado por meio de um registo que pode variar entre os seguintes valores:

- “1” - indica que há uma falha de comunicação com o servidor, estando no modo *offline*;
- “2” - indica que o sistema está a funcionar normalmente, estando no modo *online*;
- “3” - que o microcontrolador de conexão Wi-fi está com o modo de configuração de credenciais ativado.

O fluxograma 5.11 exemplifica o processo executado para transmitir informação do ESP8266 para o controlador remoto da eletrobomba.

Sempre que é recebida uma mensagem através do MQTT, o seu conteúdo é guardado em memória, permitindo que a informação seja retransmitida na eventualidade de ser preciso. A trama de dados é encapsulada entre os caracteres “A” e “#”, sendo que após o carácter de inicialização, são enviados os 5 bytes de dados, mais 1 carácter com o estado do ESP.

Sempre que algo de anómalo se verifique, tal como a perda da ligação Wi-fi ou ao *broker*, a informação em memória é alterado para um valor padrão de “99900”,

sendo que este valor irá colocar o controlador remoto em modo de paragem. Após a trama estar completa, antes de ser enviada, é realizado o cálculo do CRC, sendo que o resultado é transmitido após o envio do “#” ser realizado.

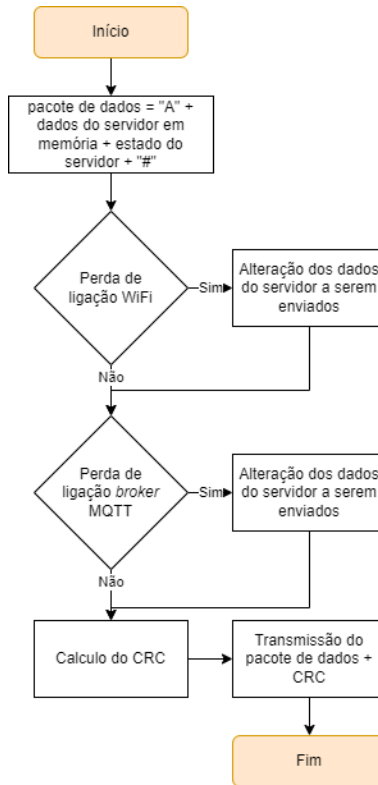


Figura 5.11: Transmissão de dados do ESP8266 para o controlador remoto da eletrobomba

## 5.2 Software do Controlador Remoto dos Sensores de Sobrecarga e Pressões

O controlador remoto dos sensores de sobrecarga e pressões tem a principal função de verificar se o reservatório inferior não se encontra em sobrecarga. Para verificar o bom funcionamento do circuito de deteção, implementou-se uma eletroválvula que permite injetar água a montante do sensor.

Neste controlador existe um circuito auxiliar implementado com um AT-tiny84, que funciona como expansão de entradas, sendo que a sua única função é realizar a leitura de um conjunto de sinais analógicos e digitais, reportando o resultado por UART a cada 500 milissegundos ao microcontrolador principal, um EP8266.

### 5.2.1 Microcontrolador principal

A figura 5.12 apresenta um fluxograma da rotina principal de atuação do ESP. Esta função é executada sempre que é recebido um comando por parte do servidor.

Primeiramente é preparada uma *string* que guarda os dados encapsulados, prontos para serem enviados como resposta para o servidor. O primeiro conjunto de informação acrescentado a esta trama são os dados recebidos pelo microcontrolador de expansão responsável por ler alguns sinais digitais e um sinal analógico.

A informação enviada pelo dispositivo de expansão costuma ser guardada em memória volátil, sendo este armazenamento crucial pois, os algoritmos de comunicação do dispositivo principal com o servidor e com o dispositivo de expansão, funcionam de forma assíncrona.

Seguidamente, realiza-se a leitura dos dados recebidos pelo *broker* MQTT, sendo que o seu tamanho e integridade são verificados. É sabido que o tamanho do comando recebido é fixo, sendo composto por 5 bytes, e que a sua construção obedece a um conjunto de regras, sendo estas que os 2 primeiros bytes tem de ser “#\$” e que os últimos 3 bytes têm de ser iguais entre si.

O estado de atuação do sensor e da eletroválvula dependem dos dois bits menos significativos, sendo que é executada uma verificação de cada um, produzindo-se um resultado dependente dos estados lógicos, semelhante à tabela 5.1

Tabela 5.1: Estado do sensor de sobrecarga e da eletroválvula consoante o valor dos bits de controlo

Bits	Sensor	Eletroválvula	Observação
00	Desligado	Desligado	Estado utilizado quando se deteta água, impede a corrosão do sensor
01	Ligado	Desligado	Estado utilizado normalmente, sensor á espera de água para sinalizar uma deteção
10	N/A	N/A	Não é possível realizar uma injeção de teste com o sensor desligado
11	Ligado	Ligado	Situação utilizada para testar o sensor

Após o comando enviado pelo servidor ter sido processado, é acrescentado um valor de confirmação do estado dos atuadores à trama de resposta, sendo que esta é, em seguida, publicada num tópico específico, sinalizando-se também que a idade da última receção é de 0 segundos.

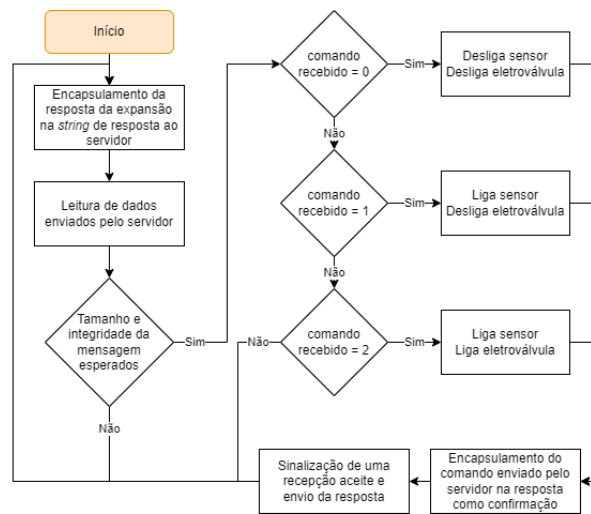


Figura 5.12: Rotina principal do controlador remoto de sobrecarga

Na figura 5.13 encontra-se um algoritmo implementado que aplica um procedimento de segurança que condiciona o comportamento do controlador, em caso de falha de comunicações com o servidor. Existe um *timer* que despoleta o processamento da rotina em seguida, uma vez a cada segundo, sendo que primeiramente é verificada se a idade da última atualização é superior a 20 segundos, na eventualidade desta condição se verificar, todos os atuadores ligados são trazidos a um estado de repouso, por segurança.

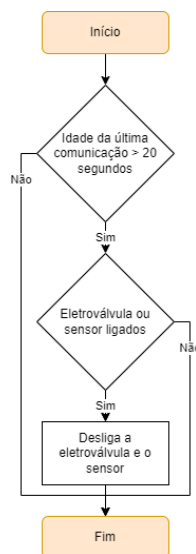


Figura 5.13: Rotina de segurança do controlador remoto de sobrecarga

### 5.2.2 Microcontrolador de expansão

A imagem 5.14 demonstra a rotina principal executada pelo microcontrolador de expansão aplicado no controlador remoto de detecção de sobrecargas. O seu principal propósito é recolher informação de uma entrada analógica e três entradas digitais, empacotando-as e enviando-as para o ESP8266.

Em cada ciclo, inicia-se um pedido de conversão de valores pelo ADC. Após a conversão estar finalizada, o resultado convertido é recolhido e guardado num *array*, seguindo-se do incremento do registo que aponta para a sua próxima posição livre, preparando-se para receber um novo valor na próxima execução.

Após o pedido de conversão ser realizado, inicia-se um comando de espera de 10 milissegundos. Após a finalização deste tempo, é verificado se o registo que aponta as posições do *array* de leituras do ADC, já foi incrementado 50 vezes. Na eventualidade deste teste retornar positivo, inicia-se o processamento dos dados recolhidos, e a sua transmissão.

Primeiramente realiza-se a média de todos os valores presentes no *array* de leituras do ADC, seguindo-se do restauro do seu registo apontador. Com isto, são lidas as 3 entradas digitais, sendo que toda a informação é encapsulada, e transmitida em conjunto com o cálculo do CRC.

Para realizar a transmissão de informação, utilizou-se a UART, sendo que esta foi desenvolvida por *software* no microcontrolador de expansão, visto este não conter *hardware* interno que o suporte.

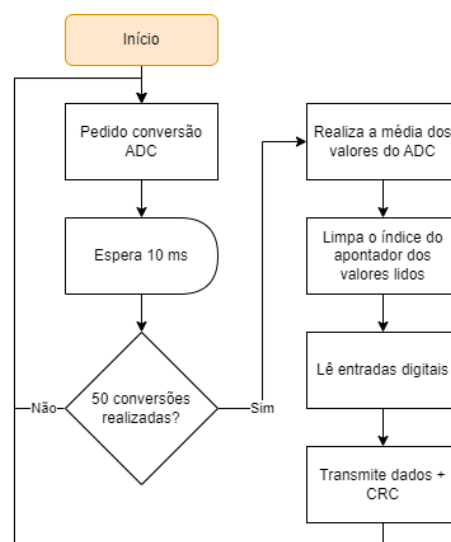


Figura 5.14: Rotina principal da expansão do sistema de sobrecarga

### 5.3 Software do Controlador Remoto de Recolha de Informação Meteorológica

Este controlador está montado no exterior, sendo que é responsável por recolher algumas variáveis atmosféricas, sendo que as mais importantes são o nível de precipitação, a temperatura e a humidade. O circuito desenvolvido conta com um controlador, o ESP12E, baseado no microcontrolador ESP8266 que realiza a recolha de informação dos diversos sensores, enviando-a para o servidor, sempre que este requisitar.

No fluxograma da figura 5.15 encontra-se apresentada a rotina principal executada pelo dispositivo, sendo que é semelhante a outras rotinas principais utilizadas noutros ESPs de controlo. Primeiramente é realizado um teste sobre o estado lógico de um interruptor de pressão dedicado a colocar o sistema no modo de configuração, guardando-se o resultado deste teste numa variável.

Na eventualidade de existir um pedido para realizar a configuração das credenciais do dispositivo, primeiramente altera-se o modo de funcionamento do *hardware* de rede, permitindo criar um ponto de acesso ao qual o utilizador se pode ligar. Após um cliente submeter a nova configuração das credenciais, estas são guardadas em memória seguindo-se de um reinício do dispositivo.

Numa operação normal do módulo, estando a sinalização de um pedido de configuração desativado, são executadas umas rotinas diferentes, sendo que primeiramente é validada a ligação com o servidor. Caso esta validação falhe, tenta-se reestabelecer a comunicação a cada 5 segundos, caso a validação seja bem sucedida, é executado um ciclo de processamento de novas mensagens MQTT.

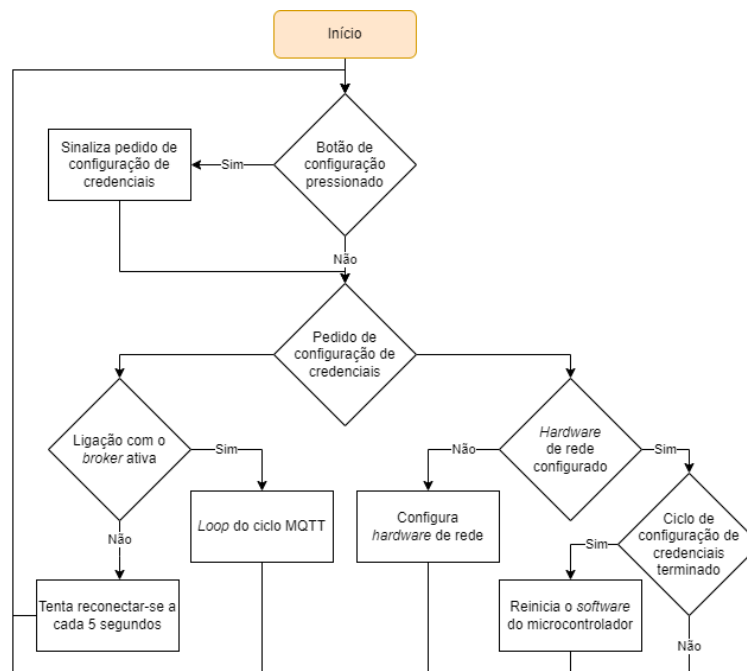


Figura 5.15: Ciclo principal do controlador remoto de recolha de informação meteorológica

Após o término do algoritmo anterior, é executado o ciclo apresentado na figura 5.16, sendo que este é responsável principalmente pela recolha de dados.

Retornando ao fluxograma, repare-se que é verificada a existência de um comando que ordena a ativação de uma resistência de evaporação da condensação que se pode acumular no sensor de recolha da humidade atmosférica. No caso de tal teste ser retornado como positivo, a resistência é ligada por 100 milissegundos, sinalizando-se tal operação no contador que ordena a sua ativação.

Note-se que esta resistência só é ativada sempre que for realizada uma comunicação com o sensor de humidade, pelo que a função que a liga sobrecarrega o barramento I2C com comandos que provoquem atividade no sensor em causa. A duração desta sobrecarga apenas dura 100 milissegundos de cada vez, para permitir a execução do restante código e, a leitura dos restantes sensores ligados por I2C.

Terminando-se a etapa anterior, executa-se uma rotina de recolha e armazenamento de informação. Toda a informação recolhida pelos sensores, à exceção do sensor de pluviosidade, é guardada num conjunto de *arrays* circulares de 50 posições cada. O princípio de funcionamento deste método baseia-se num apontador que indica a posição que contem o valor mais antigo no *array*, sendo que a cada leitura, este valor é substituído por um valor novo. Sendo que são realizadas

leituras a cada 100 milissegundos e sendo que estão reservadas em memória 50 células por tipo de variável, espera-se que o *array* seja completamente preenchido com valores novos a cada 5 segundos.

Sempre que o sistema é iniciado pela primeira vez, é sinalizada através de uma *flag* a possibilidade da existência de espaços por preencher dentro de cada *array*, sendo que se for executada uma média destes valores, podem ocorrer o retorno de resultados inválidos. Após a quinquagésima leitura ser realizada, é sinalizado pela *flag* em causa, que o *array* já se encontra completamente preenchido.

Após a rotina anterior terminar, é realizada uma recolha de todos os valores analógicos dos sensores utilizados na implementação, sendo que são guardados num conjunto de *arrays* circulares.

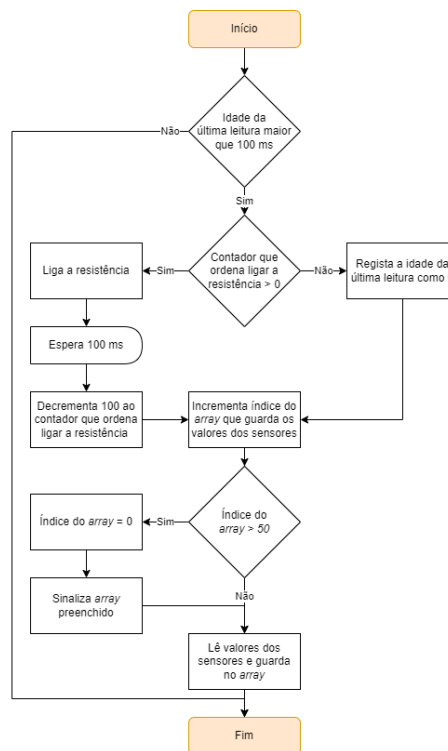


Figura 5.16: Recolha dos valores dos sensores da estação meteorológica

A medição da pluviosidade foi implementada através de um método representado em 5.17. Primeiramente é importante relembrar que o sensor envia o seu resultado através de um sinal digital que muda brevemente o seu estado sempre que detetar uma precipitação de cerca de 0.3 mm de chuva. A cada evento, a medição da precipitação é reiniciada pelo sensor, repetindo o ciclo anteriormente descrito. De forma a determinar o nível de chuva a que o sensor esteve sujeito, basta contar o número de impulsos enviados por este.

Como a geração deste sinal digital é baseada na atuação de um contacto mecânico, houve a obrigatoriedade de ser implementada uma rotina de *debounce* dentro da interrupção que responde aos eventos do sensor. Esta rotina funciona fundamentalmente, verificando se a idade da última deteção é superior a 10 milissegundos. Caso este teste se verifique, é incrementado um registo, caso contrário, o evento não produz resultado nenhum.

A escolha do valor limite de idade do último sinal foi de 10 milissegundos pois permite uma correta leitura de um sinal recebido, no entanto não é demasiado longo, impedindo a rejeição doutros sinais que poderiam ser válidos.

Para existirem dois sinais válidos num intervalo menor que 10ms, a precipitação instantânea teria de ser superior a 30 mm/s. Valor praticamente impossível de ser alcançado.

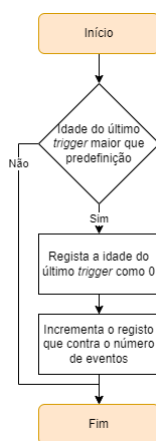


Figura 5.17: Interrupção de resposta a um evento no sensor de medição de pluviosidade

A função despoletada para processamento de informação recebida encontra-se representada em 5.18. Primeiramente é realizada a leitura dos dados recebidos, seguindo-se da verificação da sua integridade. No caso da trama recebida ser considerada como válida, são extraídos do seu conteúdo, dois bytes de dados.

O primeiro byte contém o nível de brilho que deve de ser colocado no LED de sinalização, sempre que este for atuado. O segundo byte lido, contém informação sobre o estado lógico de operação da resistência de eliminação de condensação do sensor de humidade atmosférico.

Sempre que é recebido um sinal que implica a ativação da resistência, é guardado o valor 5000 num registo indicando que este dispositivo deverá ser ativado pelos próximos 5 segundos. Na eventualidade desta ordem ser recebida de novo, na comunicação seguinte, este registo é preenchido com o valor 5000 outra vez.

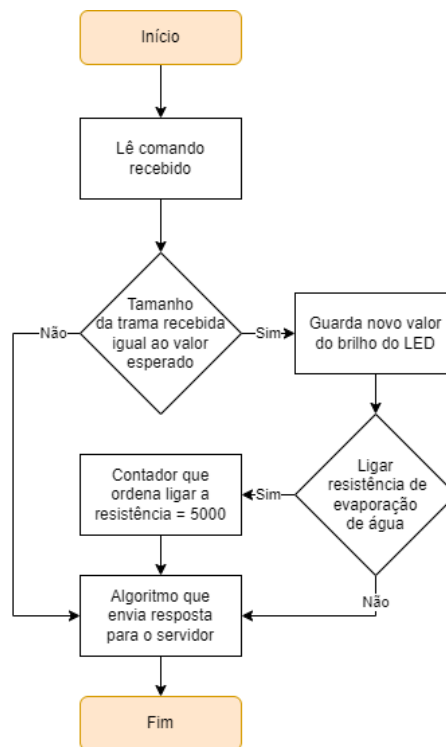


Figura 5.18: Algoritmo de processamento de um comando recebido pelo servidor

No fim do processamento dos comandos recebidos, é executada uma função que irá produzir e enviar uma resposta para o servidor com a informação recolhida.

O fluxograma da imagem 5.19 representa o código responsável por produzir uma resposta, sendo executado após a receção de uma mensagem proveniente do servidor.

A sua execução apenas se limita a realizar a média de cada *array* circular, sendo que em seguida ordena os resultados obtidos, preparando-os para uma transmissão com destino o servidor. Aquando a publicação da resposta, o LED de sinalização de atividade é piscado, indicando a ocorrência de uma transação de informação.

Todos os dados recolhidos, tais como as temperaturas, humidades, valores da intensidade solar e UV, entre outros, são organizados sempre pela mesma ordem, sendo que são separados pelo carácter “:”. Esta separação apenas foi realizada para facilitar a separação dos valores do lado do servidor, podendo ser evitada, visto a trama ser de tamanho fixo.

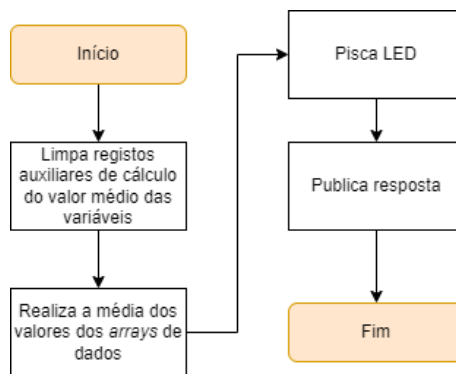


Figura 5.19: Algoritmo de resposta a um comando recebido pelo servidor

A figura 5.20 representa o pacote de dados enviada pelo controlador para o servidor, como resposta.

Note-se que o valor do contador de precipitação é enviado sem nenhum processamento, sendo que este é realizado do lado do servidor, recorrendo a um algoritmo desenvolvido em Python. Este processamento visa converter a variação de um número inteiro que representa o número de deteções realizadas pelo sensor de pluviosidade, num nível de precipitação.

Um último parâmetro a destacar na trama enviada é a sinalização da possibilidade da existência de incongruência dos dados enviados, pois os *arrays* circulares ainda não estão totalmente preenchidos. Esta sinalização é realizada através de um byte enviado após o carácter delimitador de informação, o “P”.

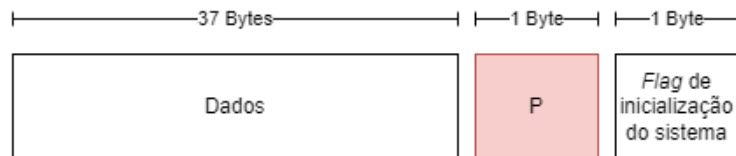


Figura 5.20: Formato da trama de resposta enviada pelo controlador remoto da estação meteorológica

## 5.4 Software do Controlador Remoto do Sistema de Rega Auxiliar

O controlador remoto do sistema de rega auxiliar é responsável por realizar a rega de uma pequena porção de área verde da habitação, com recurso a quatro eletroválvulas implementadas. Apesar deste dispositivo ser considerado como módulo subordinado, é o único controlador com capacidade para operar de forma independente e isolada, sendo que a maioria das comunicações com o servidor apenas servem para este recolher dados.

Com a implementação desenvolvida, é possível configurar até 5 programas de rega, nos quais podem ser configurados 18 parâmetros diferentes, sendo estes:

- Dias da semana em que deve de ser ativado;
- Válvulas que devem de ser atuadas;
- Hora e minuto de disparo do sistema;
- Tempo de atuação das válvulas em minutos e segundos;
- Dependência do tempo de atuação sobre a humidade do solo;
- Dependência do tempo de atuação sobre a probabilidade de precipitação;
- Estado de ativação do programa.

Os algoritmos que relacionam o tempo de atuação das válvulas com as condições ambientais e previsões atmosféricas, não se encontram implementados.

### 5.4.1 Processamento principal

No fluxograma da figura 5.21 apresenta-se o ciclo de controlo principal executado pelo microcontrolador de controlo do sistema de rega auxiliar.

Primeiramente verifica-se se a última recolha de dados ocorreu a um tempo superior a um valor predefinido, caso esta condição se verifique, os valores dos sensores de temperatura e humidade da atmosfera presente dentro da caixa onde o sistema opera, são recolhidos. É também realizada uma média dos valores de frequência retornados pelo sensor de humidade do solo. Na atual implementação, esta rotina é executada a cada 5 segundos.

Em seguida, é verificada se houve uma receção de um comando enviado pelo servidor. Caso tal se verifique, é executada uma rotina de processamento da trama recolhida. Posto isto, é recolhido o tempo do RTC, sendo este importante para realizar a alteração dos registos do estado das válvulas em modo automático, consoante a programação inserida pelo utilizador.

Neste sistema, existem dois conjuntos de registos que guardam o estado das quatro eletroválvulas de saída. Cada conjunto de é dedicado ao modo de operação do sistema, sendo que o primeiro grupo guarda o resultado do algoritmo de atuação das válvulas em modo automático, já o segundo quarteto guarda o estado das válvulas em modo manual, inserido pelo utilizador.

A última etapa da rotina principal, é verificar o modo de operação do sistema e atuar nas válvulas consoante o resultado da condição anterior e o estado do registo referente.

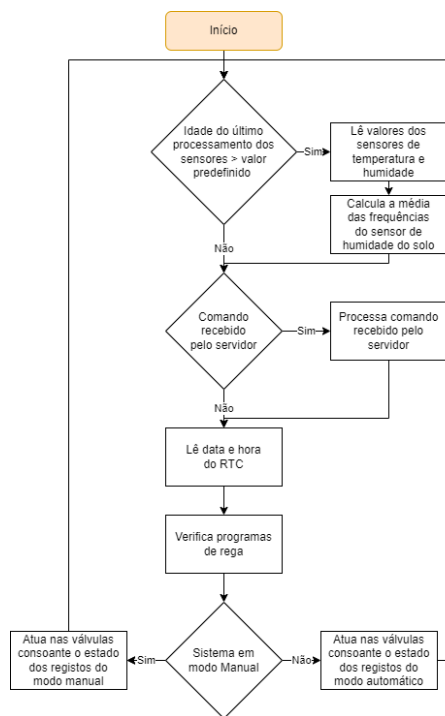


Figura 5.21: Ciclo principal do controlador remoto do sistema de rega auxiliar

O fluxograma apresentado em 5.22 apresenta o código executado aquando a ocorrência de uma interrupção de um *timer* do microcontrolador, que opera a uma frequência de 1 kHz. Primeiramente é realizada a divisão da base de tempo para uma escala com uma base de tempo de 1 segundo.

Verificando-se a condição anterior, primeiramente incrementa-se um registo que mede a idade da última recolha dos valores de sensores implementados. Seguindo-se tal operação, guarda-se o valor de oscilações do sensor de humidade do solo no último segundo, num *array* dedicado.

De forma a economizar energia e vida útil da retro-iluminação do LCD, após um determinado tempo sem deteção de atividade nos botões de controlo do sistema, entra-se num estado de repouso, desligando-se a iluminação. Devido a tal

operação, o utilizador pode perder a capacidade de conseguir realizar a leitura de informação do sistema, portanto, sempre que o sistema está num estado de dormência, a primeira ação detetada nos botões de navegação é ignorada, sendo que a única reação do sistema resulta no acendimento da retro-iluminação da *display*. Após a iluminação estar ligada, o sistema é considerado como ativo.

Na atual implementação, o tempo máximo que o sistema pode estar ativado, sem nenhuma atuação, são de 50 segundos, sendo que no último instante, quando o registo que indica que o sistema está acordado atinge o valor 0, é executada uma função que descarta eventuais valores reconfigurados, não guardados em memória, voltando em seguida para o menu principal.

Na função executada dentro da interrupção do temporizador, a última rotina verificada apenas determina se a última atualização do servidor atingiu um tempo limite. Na eventualidade de tal se verificar, caso alguma válvula esteja aberta por indicação em modo manual, é fechada, assinalando-se uma anomalia. Este algoritmo apenas impede que uma válvula fique permanentemente aberta caso um utilizador esteja a operar o sistema em modo manual remotamente e perca a ligação com o controlador remoto.

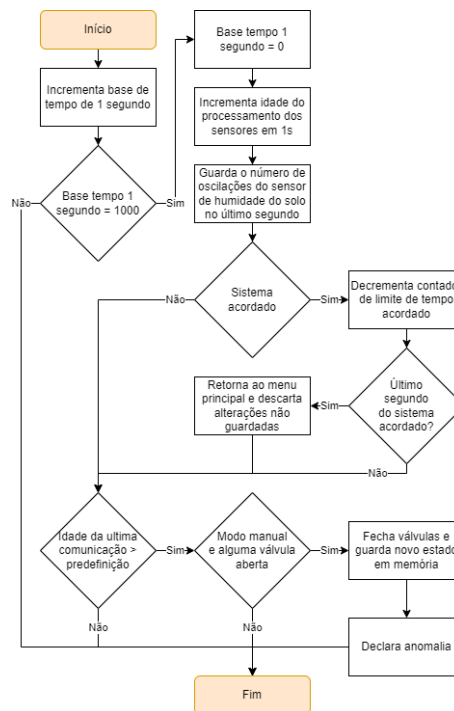


Figura 5.22: Rotina executada durante uma interrupção de um *timer* interno

A imagem 5.23 apresenta o código executado em certas situações, durante o funcionamento do código, em que pode ser útil verificar a integridade de um programa. Existe uma regra básica que tem de ser seguida por coerência lógica, isto é, um programa não pode ser considerado como ativo se as três condições seguintes não se verificarem:

- Tem de haver uma ativação das válvulas, pelo menos um dia da semana;
- Tem de haver uma ativação de pelo menos uma válvula;
- O tempo mínimo de execução do programa tem de ser de 30 segundos.

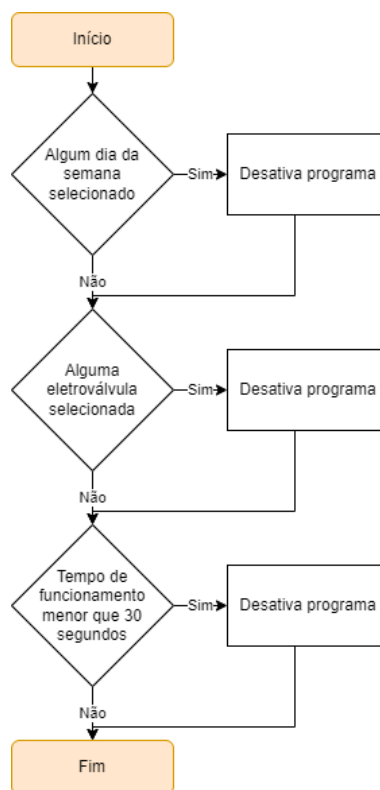


Figura 5.23: Algoritmo de verificação da validade de um programa

Na figura 5.24 encontra-se apresentado sob formato de fluxograma a rotina responsável por determinar o estado de atuação das eletroválvulas, consoante a programação semanal definida pelo utilizador.

Inicialmente, por segurança, é verificada a integridade do programa a ser testado, recorrendo-se ao algoritmo explicado anteriormente. Posto isto, verifica-se se o programa se encontra ativado, caso a condição seja considerada como válida, inicia-se a conversão, para segundos, do tempo retornado pelo RTC, do tempo de disparo do programa e da sua duração de execução.

Posto isto, é executada uma subtração do tempo de disparo há hora atual. Com este cálculo, pretende-se determinar o tempo de desfasamento do RTC com o horário de disparo do programa. Na eventualidade destes dois valores serem iguais, no momento em que as válvulas poderão de ser ativadas, o resultado na variável de retorno é de 0. Note-se que esta operação matemática é executada sobre uma variável de 32 bits, sem sinal. Para acomodar a mudança de dia no RTC e, conseqüente *rollover* do resultado da variável, sempre que o resultado for superior a 86400, é-lhe somado 86400.

As válvulas devem de estar abertas enquanto o registo de retorno da diferença de tempos, tiver um valor inferior ao tempo de duração de um programa.

Na eventualidade de todas as condições se verificarem como válidas até este ponto, antes de ser realizado uma sinalização de um pedido de abertura das eletroválvulas de forma automática, é verificado se o dia da semana atual foi configurado como dia de atuação do sistema de rega.

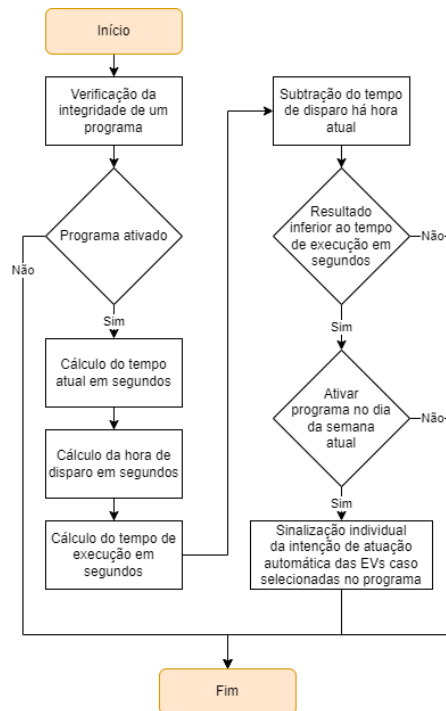


Figura 5.24: Algoritmo de decisão de ações consoante a programação semanal

### 5.4.2 Processamento de comunicações

Note-se que a comunicação implementada neste dispositivo tira partido do protocolo UART, pelo que existe um microcontrolador destinado a realizar a interface entre a UART e a rede local, permitindo assim a realização de uma comunicação entre o servidor e o controlador remoto.

Existe um total de cinco tramas que podem ser enviadas pelo servidor, que se destinam a alterar diferentes configurações do módulo implementado. Após se considerar como terminada a receção de um pacote de dados, calcula-se e verifica-se o CRC da informação, seguindo-se de uma verificação do indicador da trama. Dependendo deste valor do indicador, são executados uns algoritmos responsáveis por interpretar a sua informação.

A trama com indicador “A”, representada em 5.25, destina-se a informar o controlador remoto, sobre o estado da ligação com o servidor e do estado de operação do microcontrolador de interface. Esta é a única trama que não produz nenhuma alteração do banco de dados do sistema, pelo que é utilizada maioritariamente para realizar uma simples consulta do módulo.

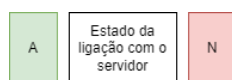


Figura 5.25: Trama de dados recebida para atualizar o sistema com o estado da ligação

A primeira trama de configuração do sistema de forma remota apresentada, encontra-se ilustrada em 5.26 e, permite alterar o modo de funcionamento do dispositivo de automático para manual, assim como alterar os registos que guardam informação sobre o estado das eletroválvulas em modo manual.

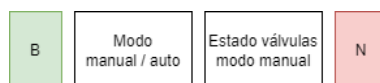


Figura 5.26: Trama recebida para alterar o modo de funcionamento do sistema e o estado das válvulas em modo manual

A trama de configuração “C”, representada em 5.27, permite ao servidor enviar para o controlador, novos valores de data e hora. Antes destes valores serem escritos no RTC, é verificada uma *flag* que indica se tal operação foi configurada como permitida. A configuração desta *flag* só pode ser realizada localmente, sendo que na eventualidade do servidor ser proibido de acertar o valor do relógio do controlador, todas as tentativas realizadas por este não produzem nenhum efeito.



Figura 5.27: Trama recebida para realizar o acerto do relógio

O diagrama 5.28 representa a trama de dados recebida para alterar um dado dos registos de programação do sistema de rega. É importante relembrar que existem no total 5 programas diferentes, em que podem ser configurados até 18 registos por cada um, sendo um total de 90 variáveis distintas. O índice da variável a ser configurada é enviado pelo primeiro byte de dados da trama, sendo que o novo valor que deve de ser escrito neste registo encontra-se armazenado no segundo byte. Antes de um novo valor ser escrito e guardado, é realizada uma verificação da sua validade, por exemplo, se for escolhido o índice 14, que representa a hora de disparo do primeiro programa, o novo valor para a hora tem de pertencer ao intervalo 0 a 23, sendo que todos os outros valores são descartados.

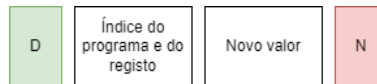


Figura 5.28: Trama de dados recebida para alterar um dado da programação do sistema de rega

A trama representada em 5.29 é enviada pelo servidor para indicar ao controlador remoto que já teve conhecimento do acontecimento de uma anomalia. Esta anomalia provém da perda de comunicações após a ordem de abertura de uma válvula de forma manual.



Figura 5.29: Trama de dados recebida responsável por restaurar sinalização de anomalias

Sempre que é recebida uma comunicação, é construída e enviada uma resposta que contem todos os dados recolhidos pelo sistema, estado das válvulas em modo automático e manual, modo de operação do sistema e os cinco programas, entre outros.

O pacote de dados pode ser iniciado por um byte que pode conter dois valores. Na eventualidade de ser escrito o valor “A”, significa que a comunicação enviada é resultante de uma ordem recebida do servidor, na eventualidade do valor deste carácter ser “B”, significa que a trama teve origem num evento que aconteceu internamente neste módulo.

Todos os dados do sistema são agrupados sempre com a mesma ordem e separados pelo carácter “:”, o que permite ao servidor distingir os valores. Após a transmissão do carácter de finalização de dados do sistema, é enviado o resultado da verificação do CRC da trama de comando, sendo que este registo é utilizado normalmente para pedir uma retransmissão da ordem.

Após a transmissão do carácter de finalização da trama, é enviado o valor calculado do CRC deste pacote de dados, permitindo ao microcontrolador de interface verificar a sua integridade.

## 5.5 Software do Controlador Remoto Principal

No fluxograma 5.30 encontra-se apresentado o algoritmo desenvolvido para realizar a determinação da frequência de oscilação dos sinais provenientes dos caudalímetros. Este valor da frequência consegue ser relacionado com o caudal do fluido que atravessa o sensor.

Primeiramente, determina-se uma variação do estado do sinal, caso uma mudança seja detetada, é incrementado o registo do caudalímetros associado, na eventualidade de existir uma subida do valor lógico da linha do sensor.

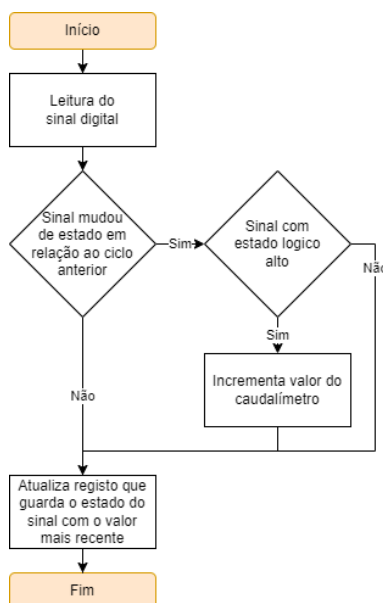


Figura 5.30: Armazenamento de eventos originados pelos caudalímetros

Os valores analógicos provenientes dos sensores de pressão incorporados no sistema, são recolhidos em intervalos de 40 milissegundos. Sempre que ocorre uma recolha dos valores, estes são guardados em *arrays* circulares dedicados a cada sensor. A cada nova escrita, o índice que aponta para o registo com o dado

mais antigo é incrementado, sendo que quando atinge o limite do tamanho do *array*, retorna para o início.

São lidos valores de três transdutores de pressão, o transdutor do coletor, do distribuidor e do compressor responsável por determinar o nível de água no furo.

A cada intervalo de 1 segundo, é executada a rotina responsável por atualizar os registos que guardam os valores recolhidos e calculados pelo controlador. No caso dos caudalímetros, cada valor dos registos dos acumuladores, são guardados em memória. Após esta operação, os acumuladores são limpos, permitindo a recolha de novos valores. Ao conteúdo guardado em memória, é aplicado um cálculo matemático que relaciona a frequência de oscilação do sinal dos caudalímetros com o caudal em litros por minuto.

Relativamente aos valores dos sensores de pressão guardados nos *arrays* circulares, são-lhes aplicados uma média e mudança de escala que permite converter os valores medidos para a unidade certa.

### 5.5.1 Leitura de dados do módulo de recolha dos níveis dos reservatórios

A comunicação com o módulo de recolha dos níveis dos reservatórios realiza-se através do protocolo de comunicação RS-485, sendo que é necessário enviar um conjunto de dados, estando estes apresentados em 5.31.

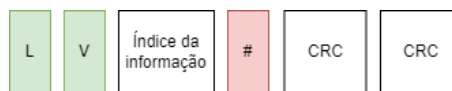


Figura 5.31: Trama enviada ao módulo dos sensores de níveis para pedir informação

Para indicar que é pretendida realizar uma leitura de informação, inicia-se a comunicação enviando-se dois bytes com o identificador do módulo, sendo que neste caso é enviado “LV”. Em seguida identifica-se qual dos dados se pretende recolher, sendo que podem ser lidos os valores dos níveis dos reservatórios e o estado do fim de curso do reservatório inferior. Por fim, é enviado o byte de finalização “#”, seguindo-se de dois bytes com o valor do CRC da trama.

O algoritmo em 5.32 é responsável por realizar um pedido periódico dos valores recolhidos pelo módulo de recolha dos níveis dos reservatórios.

Esta função é executada a cada 300 milissegundos, sendo que, durante a sua execução, primeiramente é incrementado um registo que indica o valor que se pretende recolher. Após o valor deste registo ser identificado, é enviada uma trama semelhante à representada em 5.31, seguindo-se da atualização da variável

que contem o valor da idade do último pedido efetuado ao módulo de hierarquia inferior.



Figura 5.32: Pedido de informação ao módulo dos sensores de níveis

Sempre que se verifique um evento nos terminais da UART deste dispositivo, realiza-se a leitura do byte recebido seguindo-se da sua adição à trama de receção. Quando se verificar o aparecimento do byte de finalização, indica-se que apenas se espera ler mais dois dados, podendo iniciar-se o processamento da trama logo de seguida. Os dois bytes em falta são o resultado do CRC calculado pelo módulo transmissor.

No momento em que a transmissão é dada como completada, inicia-se o processo de tratamento dos dados recebidos. Sempre que é recebida uma resposta enviada pelo módulo dos sensores de níveis, é executada uma função de processamento, semelhante à de 5.33. Primeiramente realiza-se uma verificação da integridade da trama recebida recorrendo-se ao cálculo e teste do resultado de um CRC. Na eventualidade desta condição retornar um resultado válido, é verificado o valor presente no indicador de trama. Esta operação pretende distinguir o dono da informação recebida.

Após a conclusão da rotina anterior, caso o transmissor dos dados seja validado, guarda-se o valor recebido num registo próprio e dependente do indicador

de dados transmitido.

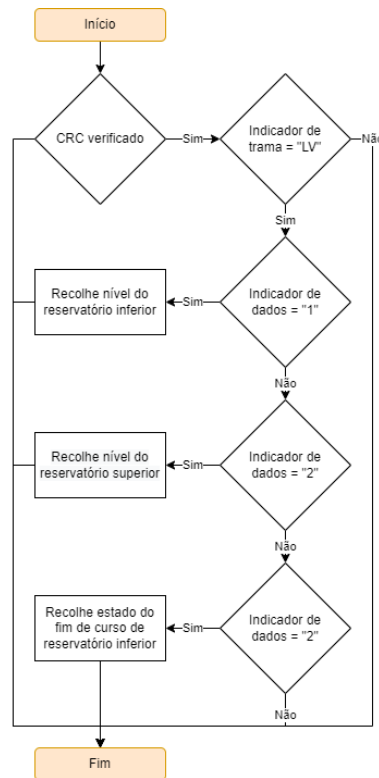


Figura 5.33: Processamento da resposta enviada pelo módulo dos sensores de níveis

### 5.5.2 Comunicação com o servidor

Uma receção de um comando bem sucedida consiste na leitura de 3 caracteres, os dois primeiros têm de ser obrigatoriamente iguais e determinam o estado lógico dos atuadores do sistema. O último byte indica ao sistema se deve de limpar a anomalia da eletrobomba auxiliar.

Esta anomalia aparece quando há uma ordem de arranque de uma eletrobomba auxiliar do sistema, sem que se verifique um caudal de manutenção mínimo num dos caudalímetros, durante um determinado tempo. Esta rotina pretende prevenir o sobreaquecimento da máquina.

Na ocorrência de uma anomalia, é gerado um valor aleatório de 1 a 9, sempre diferente do valor de erro gerado anteriormente, enviando-se este resultado para o servidor. Quando o valor for reencaminhado de volta para o controlador, por meio da trama de comando, o erro é limpo, podendo-se arrancar a bomba novamente.

Por defeito, caso o tempo desde a última mensagem enviada pelo servidor exceda um valor predefinido, todos os atuadores são desativados.

O diagrama de blocos que processa a informação recebida pelo servidor, encontra-se apresentada em 5.34. Sempre que é recebido um novo carácter pela UART, este é guardado numa trama, de forma sequencial. No momento em que se encontra o carácter de finalização, o “\n”, executa-se um algoritmo responsável por realizar a verificação da mensagem recebida.

A verificação da integridade da trama consiste na verificação da igualdade dos dois primeiros bytes, sendo que são os valores mais críticos enviados pelo servidor, uma vez que permitem realizar uma atuação nos dispositivos eletrónicos. Caso o teste da integridade da trama seja bem sucedido, primeiramente é realizada uma atuação dos componentes acoplados ao sistema, em concordância com o estado dos quatro bits menos significativos do byte de comando.

Após a finalização deste algoritmo, o registo que assinala a idade da última comunicação é escrito com o valor 0. Este valor é utilizado para parar o sistema caso haja uma falha nas comunicações.

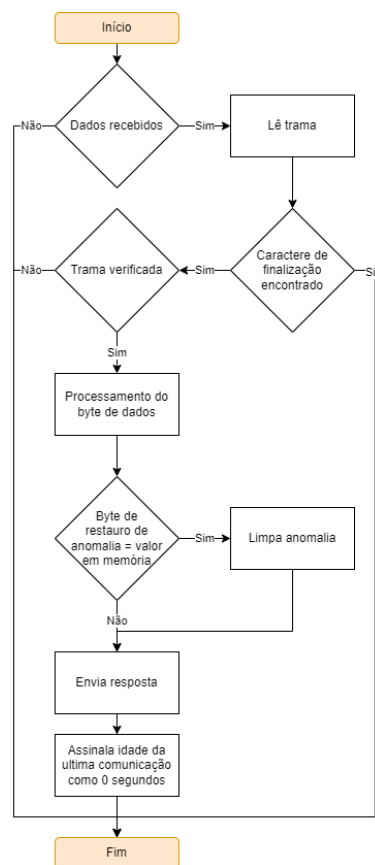


Figura 5.34: Processamento de comandos enviado pelo servidor

## 5.6 Conclusão do Capítulo 5

Neste capítulo foram apresentadas as soluções de *software* desenvolvidas para os diferentes controladores remotos de forma a que os módulos tivessem um comportamento próximo do idealizado. Esta implementação permite assim realizar a recolha de dados, e executar algum controlo e proteção de diferentes dispositivos.

## Capítulo 6

---

# Implementação do Servidor

---

*Neste capítulo são abordados diferentes tópicos relativos à implementação do servidor, estendendo-se desde a explicação dos serviços instalados, assim como a sua configuração, terminando-se com a explicação das aplicações desenvolvidas*

De forma a disponibilizar uma plataforma de suporte às tecnologias de comunicação e controlo, instalou-se o sistema operativo Fedora 34 Workstation numa máquina com as seguintes características:

- Processador - Intel(R) Core(TM) i3-3240 @ 3.40GHz
- Armazenamento - Kingston 120GB SSD SATA III 500/320
- Memória RAM - 8GB @ 1600 MHz DDR3
- Memória Swap - 8GB (No armazenamento principal)
- Conectividade - Placa de rede 1 GB/s

Neste computador correm serviços como os servidores do protocolo de transferência de hipertexto (HTTP) e MQTT, uma base de dados MySQL e as aplicações de gestão e controlo do sistema desenvolvidas em Python.

Por questões de segurança, todo o tráfego de informação entre o servidor e os controladores encontra-se isolado, por *software*, do tráfego que circula para a Internet. Apesar deste servidor estar localizado na rede doméstica, há a possibilidade de ser instalado remotamente, no entanto, as configurações que implementam a separação das comunicações tem de ser levantadas.

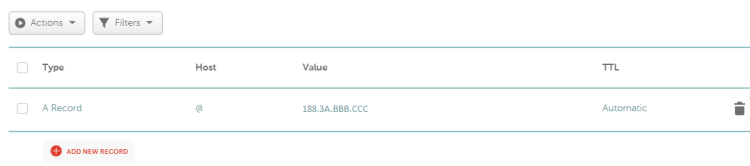
## 6.1 Configuração do DNS, router e encriptação web

De forma a melhorar e a assegurar uma boa experiência durante a interação com a aplicação desenvolvida, adquiriu-se um domínio e realizou-se a configuração de um software que realizasse a instalação de certificados, permitindo encriptar a informação trocada entre um cliente e um servidor.

### 6.1.1 Configuração do DNS

Na consequência de ser pouco prático decorar o endereço do protocolo de rede (IP) que localize na Internet, o fornecedor de um serviço web, procedeu-se à compra de um domínio abrangido pelo domínio de nível de topo “.xyz”. A obtenção do domínio foi realizada através da empresa *Namecheap*, que é uma entidade que permite realizar o registo de nomes de domínio, autorizada pela corporação da Internet para atribuição de nomes e números (ICANN). A escolha desta empresa e deste domínio de nível de topo deveu-se unicamente ao seu baixo custo.

Após o registo do domínio, foi necessário realizar a configuração de um sistema de nomes de domínio (DNS) que permitisse traduzir o nome do domínio num endereço IPv4 numérico, este processo denomina-se por resolução de nomes. Tal operação foi configurada no serviço DNS fornecido pela *Namecheap*, demonstrada na figura 6.1, sendo que se adicionou uma entrada do tipo “A” na tabela de configuração do DNS. Esta entrada indica o endereço IPv4 que deve de ser retornado sempre se realizar uma pesquisa pelo domínio registado.



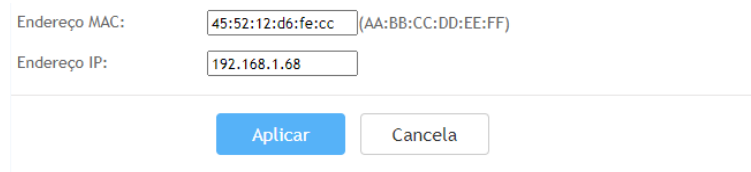
Type	Host	Value	TTL
A Record	@	188.3A.BBB.CCC	Automatic

Figura 6.1: Configuração do sistema de nomes de domínio

### 6.1.2 Configuração do Router

Note-se que o endereço IP utilizado para aceder à aplicação desenvolvida não corresponde ao endereço da máquina que disponibiliza os serviços, mas sim ao endereço público atribuído a um *router* fornecido pelo provedor de serviços de Internet (ISP). Neste dispositivo foi realizada uma configuração indicando as regras de reencaminhamento, que permitem direcionar o tráfego implícito ao servidor.

A primeira configuração realizada consiste na atribuição de um endereço estático à máquina em causa, semelhante à figura 6.2. Com esta operação pretende-se impedir que o *router* atribua novos endereços aleatórios ao servidor, inviabilizando a configuração das regras de reencaminhamento.



Endereço MAC:  (AA:BB:CC:DD:EE:FF)

Endereço IP:

Figura 6.2: Atribuição de um endereço IP estático pelo *router* ao servidor

Na figura 6.3 apresenta-se a configuração realizada no *router* de forma a reencaminhar o tráfego do servidor para a Internet. A primeira declaração realizada foi indicar o endereço IP atribuído ao servidor, na rede privada, sob o campo “anfitrião interno”, neste campo especificou-se também o endereço de controlo de acesso ao meio (MAC) do computador. Em seguida realizou-se o reencaminhamento de tráfego em apenas 3 portos específicos. Visto que a maioria das comunicações são realizadas sob HTTP ou HTTPS, que utilizam os portos padrão 80 e 443 respetivamente, reencaminhou-se todo o tráfego destes portos do *router* para o mesmo destino no computador.

Para possibilitar a atualização de valores em tempo real na aplicação web com baixa latência, foram abandonados métodos de *polling* por pedidos HTTP e foram utilizadas *websockets* que permitem abrir e manter um canal de comunicação *full-duplex*. Tais comunicações utilizam a porta 8081 do *router* e do computador.

Uma vez que a comunicação funciona sobre o protocolo de controle de transmissão (TCP), apenas a este foi concedida a autorização para reencaminhamento.

### 6.1.3 Encriptação web

Dado que a aplicação desenvolvida troca alguma informação sensível entre o servidor e o navegador, houve a necessidade de implementar uma tecnologia que permitisse a encriptação das trocas de informação.

Na figura 6.4 encontra-se um modelo que pretende explicar o funcionamento da encriptação assimétrica, método usado para estabelecer uma comunicação segura. Para poder ser executada tal operação, há a necessidade de, previamente, criar-se duas chaves, denominadas de chave pública e privada. A chave pública é partilhada pelo destinatário, sendo que é utilizada pelo remetente nos algoritmos de encriptação das mensagens. Em seguida, depois da informação ser transmitida, é utilizada a chave privada, por parte do destinatário, para realizar a operação de desencriptação, de modo a obter-se a mensagem original.

O par de chaves também pode ser utilizado para realizar a assinatura de uma mensagem. Para realizar este processo, utiliza-se um algoritmo semelhante ao da figura 6.4, no entanto, no sentido inverso. Este processo é útil para um destinatário verificar a autenticidade da origem de uma mensagem e a sua integridade.

	Nome de aplicação	Nome da WAN	Anfitrião interno	Anfitrião externo	Ativar
<input type="checkbox"/>	SmartHome	Internet	192.168.1.68	--	Ativar

Tipo:  Definido pelo utilizador  Aplicação

Aplicação:

Ativar aplicação de portas:

Nome de aplicação:

Nome da WAN:

Anfitrião interno:

Endereço IP externo de origem:

---

Protocolo:  Número da porta interna:

Número da porta externa:   Número da porta da fonte externa:

---

Protocolo:  Número da porta interna:

Número da porta externa:   Número da porta da fonte externa:

---

Protocolo:  Número da porta interna:

Número da porta externa:   Número da porta da fonte externa:

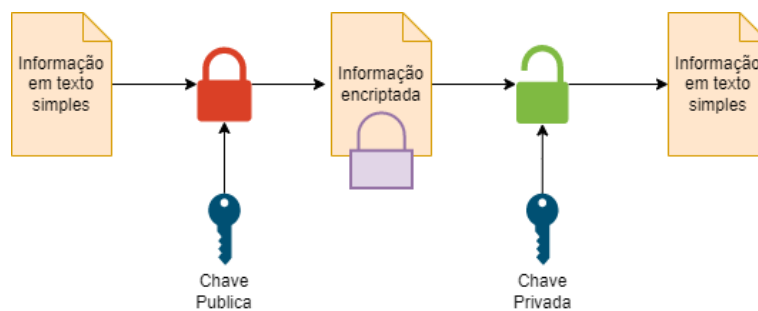
Figura 6.3: Regras de reencaminhamento do *router*

Figura 6.4: Encriptação e descriptação assimétrica

Existem dois métodos principais para se obter uma certificação, o primeiro consiste na utilização de um *software*, por exemplo o OpenSSL, que permita criar um par de chaves localmente, a este tipo de certificação costuma-se denominar de auto atribuída. Por questões de segurança, muitos navegadores questionam a fidedignidade de tais certificados, pelo que costumam avisar o utilizador que navega para um site com chaves auto atribuídas. Devido a este motivo, optou-se por mudar para uma certificação através de uma autoridade de certificação publica[78].

De forma a conseguir-se a atribuição de certificados, recorreu-se à autoridade de certificação *Let's Encrypt* que fornece serviços sem custo associado, providenciados pelo grupo de pesquisa de segurança na Internet (ISRG).

Para se obter um certificado para um domínio é importante demonstrar a existência de controlo sobre este, tal demonstração é realizada utilizando um ambiente de gerência automática de certificados (ACME) que permite automatizar a transação de certificados entre o servidor e a autoridade de certificação. A *Let's Encrypt* recomenda a utilização de um cliente ACME, o *Certbot*, que permite gerenciar a certificação em aplicações administradas manualmente, sem tempo de inatividade[79, 80, 81].

Após a primeira configuração do *Certbot*, inicia-se um processo de gestão automática dos certificados. Esta gestão consistem na realização periódica, atualmente a cada três meses, de um pedido à autoridade de certificação, que consiste na emissão de um novo certificado, revogando o antigo.

Após a obtenção de um novo par de chaves estas são instaladas pelo cliente ACME no servidor, seguindo-se de um reinício dos serviços que tiram partido das chaves, tais como o Apache e o *broker* MQTT.

## 6.2 Configuração do Apache

O Apache é um servidor HTTP instalado no computador e visa fornecer a um cliente resposta a pedidos realizados por este que são na sua maioria páginas web. Toda a estrutura da aplicação web encontra-se desenvolvida e integrada neste *software*, pelo que permite ao utilizador realizar uma navegação. São fornecidos outros serviços tais como a execução de código num pré-processador de hipertexto (PHP), que permite realizar sobretudo a autenticação de um utilizador e a leitura e formatação de informação da base de dados.

Como foi explicado, decidiu-se instalar no servidor um método que permitisse encriptar as comunicações. Para implementar tal funcionalidade a nível do Apache, procedeu-se a uma configuração semelhante à apresentada no excerto 6.1. No código, da linha 3 à linha 6 especifica-se uma diretiva de retificação do localizador uniforme de recursos (URL), no caso da diretiva apresentada, a sua função consiste no redirecionamento de todo o tráfego HTTP para HTTPS, retornando ao browser do cliente o código de resposta 301, indicando que houve um redirecionamento permanente.

```
1 <VirtualHost *:80>
2     ...
3     RewriteEngine on
4     RewriteCond %{SERVER_NAME} =domain.xyz [OR]
5     RewriteCond %{SERVER_NAME} =www.domain.xyz
```

```

6 RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=
  permanent]
7 </VirtualHost>

```

Lista 6.1: Configuração do *VirtualHost* HTTP

No excerto de código apresentado em 6.2 encontra-se apresentado um excerto de configuração de um *VirtualHost* HTTPS, repare-se imediatamente na linha 1 a indicação ao Apache para incluir o módulo *mod\_ssl* que providencia encriptação sobre a camada de *sockets* seguros (SSL) e a camada de transporte segura (TLS). Na declaração do *VirtualHost*, na linha 2, aponta-se para a porta associada normalmente a transações HTTPS, a porta 433, seguindo-se de uma declaração do *hostname* da aplicação desenvolvida, semelhante ao excerto de código 6.1.

Note-se que este *VirtualHost* difere do anterior pela ausência da diretiva de retificação do URL e, pela presença da declaração do local de armazenamento do par de chaves utilizados na encriptação. Esta declaração encontra-se apresentada nas linhas 5 e 6.

```

1 <IfModule mod_ssl.c>
2 <VirtualHost *:443>
3 ...
4 SSLCertificateFile /etc/letsencrypt/live/domain.xyz/fullchain.pem
5 SSLCertificateKeyFile /etc/letsencrypt/live/domain.xyz/privkey.pem
6 </VirtualHost>

```

Lista 6.2: Configuração do *VirtualHost* HTTPS

Uma última configuração importante do Apache corresponde à criação e aplicação de um ficheiro de configuração contido à aplicação desenvolvida, que permite obrigar a determinados comportamentos por parte do servidor. Uma porção do código incluído neste ficheiro apresenta-se no excerto 6.3, na linha 1 é indicado ao Apache a proibição da listagem do seu conteúdo a um cliente, permitindo acesso a diferentes páginas web, mas recusando a apresentação hierárquica do site e seus constituintes.

Nas linhas 2 e 3 são apresentados os locais dos documentos que devem de ser enviados para o *browser* do cliente no caso de obtenção do código de erro 404, informação solicitada não encontrada e o código de erro 403, acesso recusado a informação solicitada. Estas duas configurações apenas permitem apresentar uma página personalizada no caso da obtenção de algum destes erros.

Por fim, nas linhas 5 a 8, são apresentadas as diretivas que redirecionam todos os acessos ao site por endereçamento IP direto, retornando o URL que deve de ser seguido, acompanhado com o código 301 que indica um redirecionamento permanente.

```

1 Options -Indexes
2 ErrorDocument 404 /Errors/404.php

```

```
3 ErrorDocument 403 /Errors/403.php
4
5 RewriteEngine On
6 RewriteBase /
7 RewriteCond %{HTTP_HOST} ^188.3A.BBB.CCC
8 RewriteRule ^(.*)$ https://domain.xyz/$1 [R=301,L]
```

Lista 6.3: Ficheiro *htaccess*

## 6.3 Configuração do Broker MQTT

O *software* utilizado para suportar as comunicações através de MQTT, foi o *broker* Mosquitto. Ele é responsável por interligar os três constituintes deste sistema: os controladores remotos, os algoritmos de decisão no servidor e os utilizadores que acedem ao sistema através da aplicação web.

Na figura 6.5 encontram-se representadas as principais rotas de tráfego indicando a sua origem e direção. É claramente possível realizar uma troca de mensagens direta entre a aplicação web e os controladores remotos, no entanto, note-se que decidiu-se separar todas as mensagens referentes ao *software* de *frontend*, das mensagens referentes aos controladores, tal decisão partiu fundamentalmente por dois motivos, sendo o primeiro a segurança, a separação das comunicações permite realizar um isolamento total do servidor e dos controladores com o exterior, impedindo a escuta de informação e controlo do sistema.

Outro motivo que levou à separação das comunicações foi devido ao facto de muita informação necessitar de um pré processamento antes de poder ser apresentada ao utilizador.

No excerto 6.4 encontra-se apresentado um sumário da configuração do *broker* MQTT, a separação das mensagens da aplicação web e dos controladores ocorrem logo num dos mais baixos níveis, sendo que dedicou-se dois portos diferentes que permitem realizar o acesso ao *mosquitto*, o porto 1883 destina-se única e exclusivamente a tráfego dentro da rede privada, que consiste maioritariamente em comunicações realizadas pelos controladores remotos.

Para implementar, de forma segura, a separação da informação, primeiramente esclareceu-se na linha 1 que todas as configurações realizadas destinam-se a um único porto previamente especificado, prevenindo assim que declarações de informação realizadas num local, sejam aplicadas noutra, por acidente.

Para a porta 1883, especifica-se na linha 4 indica-se a localização do ficheiro que armazenadas as credenciais de cada utilizador, contendo apenas o nome de utilizador e um *hash* da sua palavra-passe. Por fim, na linha 5 aponta-se para um ficheiro de configuração que permite definir os limites de liberdade de cada utili-

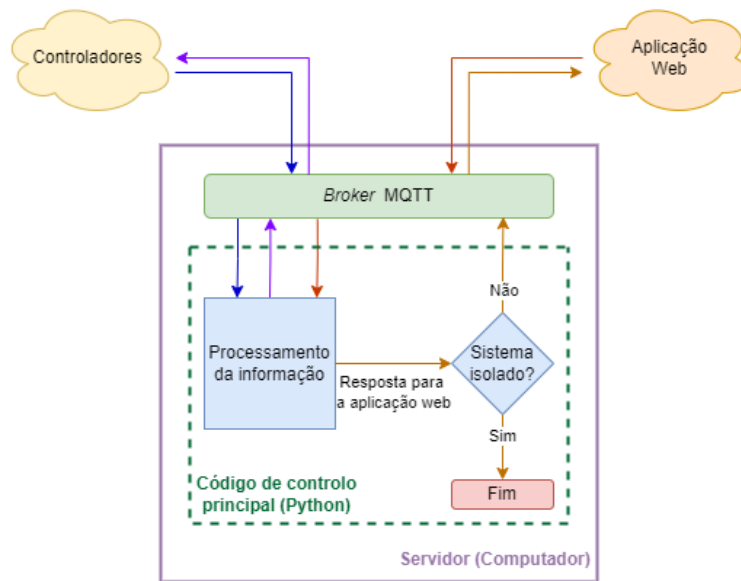


Figura 6.5: Arquitetura do tráfego de mensagens de dados

zador, isto é, definem-se os tópicos e sub-tópicos a que estes podem subscrever, condicionando o envio e receção de informação.

Para a porta 8081, as configurações dos ficheiros que contem as permissões e credenciais de cada utilizador são semelhantes ao porto anterior. Indicações adicionais realizadas para esta porta são, primeiramente, a indicação do tipo de protocolo a utilizar na linha 9, uma vez que a troca de mensagens com a aplicação web funciona sobre *websockets*, a utilização deste protocolo tem de ser especificada. Por fim, nas últimas 2 linhas, indicou-se o local de armazenamento de credenciais que permitem realizar a encriptação das mensagens.

```

1 per_listener_settings true
2
3 listener 1883
4 password_file /etc/mosquitto/pwfile.pwd
5 acl_file /etc/mosquitto/aclfile_Lan.acl
6
7 listener 8081
8 ...
9 protocol websockets
10 certfile /etc/letsencrypt/live/domain.xyz/cert.pem
11 cafile /etc/letsencrypt/live/domain.xyz/fullchain.pem

```

Lista 6.4: Ficheiro de configuração do *broker* MQTT

Um exemplo de um ficheiro de controlo de acessos utilizado na configuração do *mosquitto*, encontra-se apresentado no excerto 6.5, sendo que para declarar uma nova regra, indica-se primeiramente o nome do utilizador, seguindo-se dos

tópicos a que pode realizar uma subscrição. Note-se que a utilização do carácter “#” indica que os limites de subscrição de um cliente alargam-se para sub-tópicos hierarquicamente descendentes de um tópico principal.

```
1 # This only affects clients with username "well_remCtl".
2 user well_remCtl
3 topic remote_controllers/drill/#
4
5 ...
```

Lista 6.5: Exemplo de ficheiro de controlo de acessos do *broker* MQTT

## 6.4 Base de Dados

Devido à necessidade de armazenar certa informação de forma não volátil, como por exemplo, palavras-passe, dados recolhidos anteriormente para realizar gráficos, entre outros, necessitou-se de uma ferramenta que armazenasse dados de forma fiável e estruturada. Para tal, escolheu-se o MySQL, que é um *software* de gerenciamento de banco de dados.

### 6.4.1 Configuração da base de dados

A instalação e configuração do MySQL não necessitou de grande atenção, tendo em conta que é um processo simples e bem estruturado. Uma das únicas alterações realizadas que merece ser notada encontra-se apresentada no excerto 6.6. O único propósito da execução de tais comandos de linguagem de consulta estruturada (SQL), foi devido a haver a necessidade de desligar o sistema de *logging* do *software* e, eliminar relatórios antigos, visto estar a consumir muitos recursos de armazenamento. Idealmente deviam de ser alteradas as configurações que limitam o tamanho de disco usado e, a idade que ativa uma remoção de *logs* antigos.

```
1 SET sql_log_bin = 0;
2 PURGE BINARY LOGS
```

Lista 6.6: Configuração do MySQL

### 6.4.2 Arquitetura da base de dados

O armazenamento de informação é realizado, principalmente, sobre dois métodos. Ora é guardado, de forma volátil, em variáveis comuns em Python, pelos algoritmos de controlo. Ora o armazenamento é realizado com recurso a um banco de dados. A escolhas do local de armazenamento depende da influência, sobre o sistema, dos dados a serem armazenados, assim como o seu impacto no caso da sua perda. A figura 6.6 apresenta a estrutura da base de dados implementada. Uma descrição das tabelas encontra-se em seguida:

- “auth” - tabela que guarda informação utilizada em autenticações necessárias na aplicação web, contendo quatro campos, os dois primeiros guardam as credenciais de acesso à aplicação, os dois últimos guardam palavras-passe utilizadas para realizar operações especiais.
- “wtr\_main\_data” - contem informação sobre o estado do sistema, por exemplo, o modo de operação, o número total de comunicações, intensidade do indicador luminoso de alguns controladores remotos, entre outros.
- “wtr\_timers” - guarda os programas utilizados para operar o sistema de rega principal, note-se que o sistema de rega auxiliar, operado pelo controlador remoto guarda os seus próprios programas independentes destes.
- “wtr\_2d\_lineGraphs” - informação armazenada com uma periodicidade de novas entradas de 5 segundos, utilizada para realizar a construção dos gráficos de informação do sistema. O segundo campo desta tabela indica o momento em que foram recolhidos os valores, os cinco campos seguintes guardam informação meteorológica, tal como a temperatura, humidade intensidade solar, entre outros. Em seguida, é armazenada, em seis células, informação sobre os volumes de água que atravessam os caudalímetros. Nas seis células finais apresentam-se os níveis de água do furo e dos reservatórios, seguindo-se de pressões analógicas recolhidas em diferentes locais do sistema.
- “wtr\_1y\_lineGraphs” - tabela com uma estrutura idêntica à “wtr\_2d\_lineGraphs”, com a principal diferença de que os valores são armazenados com um intervalo de 5 minutos. Estes valores são utilizados para diferentes operações tais como a realização de regressões lineares que permitem transformar variações de níveis em caudais e balanços.
- “wtr\_src\_flw\_analysis” - tabela de histórico dos volumes de água obtido das diferentes nascentes, com uma resolução de 1 dia. Nesta tabela também são armazenadas imagens que demonstram o resultado dos algoritmos de regressão.
- “IPMA\_data” - contem informação fornecida pelo Instituto Português do Mar e da Atmosfera (IPMA) sobre o valor de precipitação mínimo e máximo, referente ao concelho onde o sistema opera. Nova entrada realizada diariamente.
- “wtr\_home\_lineGraphs” - local de armazenamento de informação apresentada sobre a forma de gráfico de linhas no menu principal. Contem, maioritariamente, dados relativos à temperatura e humidade dos controladores e atmosféricos, guardando também informação sobre a humidade do solo

e valores de unidades elétricas medidos pelo controlador remoto da eletrobomba.

- “wtr\_main\_btms” - tabela que armazena o estado de válvulas e outros dispositivos operáveis do sistema.

Table Name	Fields and Data Types
IoT_auth	user : text main_pass : text wtr_special_auth_1 : text wtr_special_auth_2 : text
IoT_wtr_main_data	id : int sysMode : text TT : text WthLghtInt : text WthReMstRem_Mode : text IrrLghtInt : text
IoT_wtr_timers	id : int info : text TrTmeH : int TrTmeM : int DrTmeM : int DrTmeS : int MstDp : int MetDp : int State : int Bnd_1 : int Bnd_2 : int Bnd_3 : int Bnd_4 : int Dow_1 : int Dow_2 : int Dow_3 : int Dow_4 : int Dow_5 : int Dow_6 : int Dow_7 : int
IoT_wtr_src_fw_analysis	id : int timestamp : bigint mines_img : blob drill_img : blob mines_fw : float drill_fw : float
IoT_IPMA_data	id : int timestamp : text min : float max : float
IoT_wtr_2d_lineGraphs	id : int timestamp : bigint Tatm : float Hatm : float PrecMd : float Light : float UVa : float int_caud : float hp_caud : float well_caud : float drill_caud : float topres_caud : float emg_caud : float drillLvl : float topres_lvl : float botres_lvl : float Patm : float HpPSns : float ColPSns : float DisPSns : float
IoT_wtr_1y_lineGraphs	id : int timestamp : bigint Tatm : float Hatm : float PrecMd : float PrecGet : float Light : float UVa : float int_caud : float hp_caud : float well_caud : float drill_caud : float topres_caud : float emg_caud : float drillLvl : float topres_lvl : float botres_lvl : float Patm : float HpPSns : float ColPSns : float DisPSns : float
IoT_wtr_home_lineGraphs	id : int timestamp : bigint Tatm : float Hatm : float PrecMd : float Hsolo : float rawFreq : float Vlt : float Curr : float Pwr : float Lvl : float Tatm_wth : float Hatm_wth : float Tctrn_wth : float Hctrn_wth : float
IoT_wtr_main_btms	id : int MotVacBr : int ManVacBRHp : int ManVacBRSys : int MotDisBotRes : int MotColBotRes : int ManDrill : int ManHp : int MotHp : int ManTopRes : int MotTopRes : int ManWell : int MotDisWell : int MotColWell : int ManOvfTest : int MotWaste : int AuxPmp : int

Figura 6.6: Tabelas da base de dados

A razão que levou à construção de três tabelas de registo de dados ao longo do tempo, foi unicamente o fraco desempenho do sistema em recolher largas quantidades de valores de uma tabela, assim como o fraco desempenho do navegador do cliente em apresentá-los. Pelo que foram criados agrupamentos com bases de tempo diferentes.

## 6.5 Software de Gestão do Sistema

Como já tinha sido referido anteriormente, as comunicações realizadas através do *broker* MQTT regem-se por uma arquitetura do tipo ponto-a-ponto, sendo que existe uma aplicação desenvolvida em Python que funciona como nó central e intermediário entre os diferentes dispositivos. Esta configuração anormal foi escolhida pela simplicidade na realização de um controlo de segurança de cada subsistema, sendo que, por este método podem ser isolados e ignorados.

A aplicação implementada tem o papel de realizar a maior parte do controlo e da toma de decisões no sistema, consoante a informação recebida por parte dos utilizadores e dos módulos desenvolvidos, enviando as decisões tomadas para os controladores.

Na figura 6.7 encontra-se representada sob a forma de um diagrama, a arquitetura da aplicação desenvolvida. Note-se que a aplicação consiste fundamentalmente em dois programas que funcionam de forma separada, sendo que trocam informação entre si, através do *broker* MQTT. Ambos os processos lançam múltiplos sub-processos que funcionam de forma independente e assíncrona, sendo que cada um realiza um controlo diferente.

O algoritmo de interface da base de dados apenas realiza escritas e leituras periódicas das tabelas da base de dados, sempre que há um pedido por parte do código principal. Já o *software* principal realiza o controlo de todo o sistema, assim como realiza o processamento de pedidos efetuados pelos utilizadores através da aplicação web.

### 6.5.1 Algoritmo de Controlo Principal

Neste algoritmo encontram-se a grande maioria das rotinas implementadas de controlo do sistema. Nos tópicos seguintes encontram-se descritos os diferentes processos desenvolvidos.

#### Rotina principal

Aquando a execução do código desenvolvido, primeiramente é executada a rotina do diagrama 6.8 realizando-se a inicialização do sistema. O primeiro processo realizado, e o mais importante, é a recolha de informação da base de dados sobre o estado do sistema, pretendendo-se saber qual dos modos da tabela 6.1 sistema deve de operar.

Tabela 6.1: Modos de operação do sistema

Modo	Observação
Automático	Modo de operação normal
Manual	O sistema apenas realiza operações ordenadas pelo utilizador
Parado	O sistema desliga todos os atuadores e impede novas operações
Isolado	O sistema funciona de forma automática mas não realiza comunicações com a aplicação web

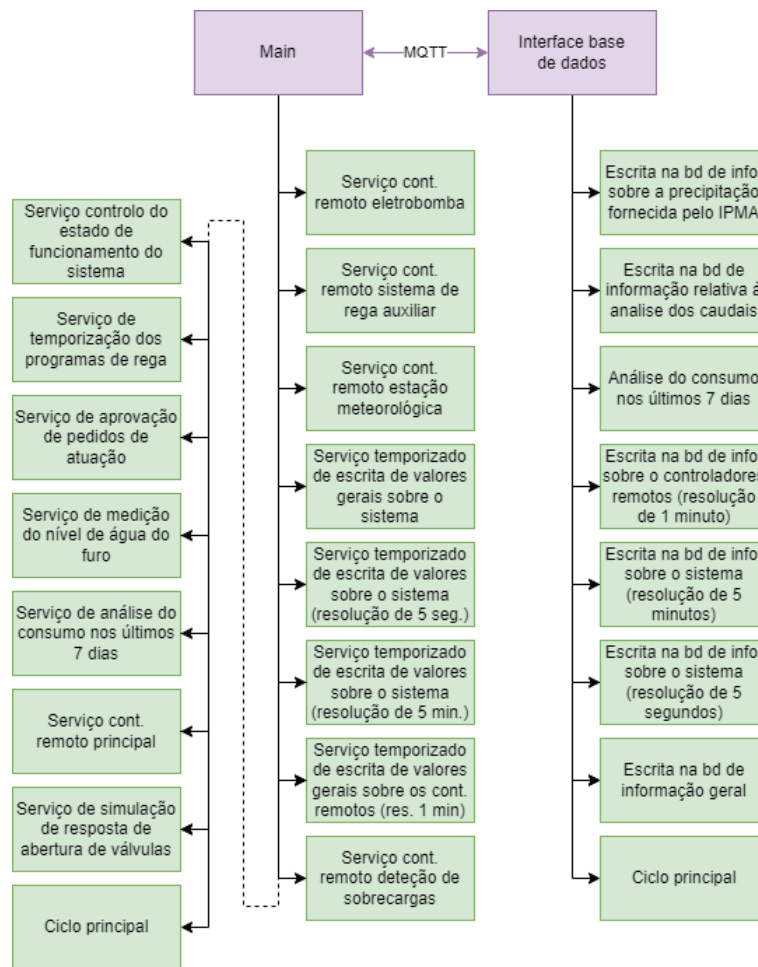


Figura 6.7: Arquitetura da aplicação de controlo desenvolvida

Após determinar o modo de funcionamento do sistema, são recolhidas informações tais como as configurações dos programas de rega e, o saldo da API de serviço de mensagens curtas (SMS) fornecida pela Twilio.

Em seguida, seguindo-se a continuidade do fluxograma são lançados múltiplos sub-processos, sendo que estes serão explicados com detalhe em tópicos seguintes neste capítulo. Após a inicialização das diferentes funções terminar, considera-se que o sistema está pronto para operar, restando apenas realizar a recolha de dois valores relacionados com os caudais das minas e do furo da habitação seguindo-se do lançamento do ciclo infinito principal.

No fluxograma apresentado em 6.9 encontra-se um diagrama de blocos da primeira parte do ciclo infinito principal que corre uma vez a cada 30 segundos.

Na primeira condição, inicia-se o processo com a passagem do tempo do relógio do computador para um valor com uma resolução em minutos. Seguindo-se este

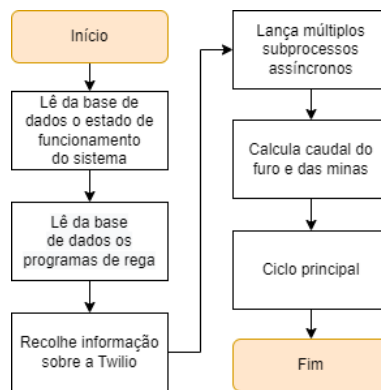


Figura 6.8: Rotina de inicialização do sistema

cálculo, verifica-se se o valor obtido é divisível inteiramente por 30. Isto implica que a condição se verifique a cada 30 minutos estando em sincronismo com a mudança da hora.

Na eventualidade desta condição se verificar, é executada uma vez uma função que calcula o caudal das minas e do furo. Por fim, a última operação apresentada neste fluxograma, realiza a recolha de informação da API de envio de SMS.

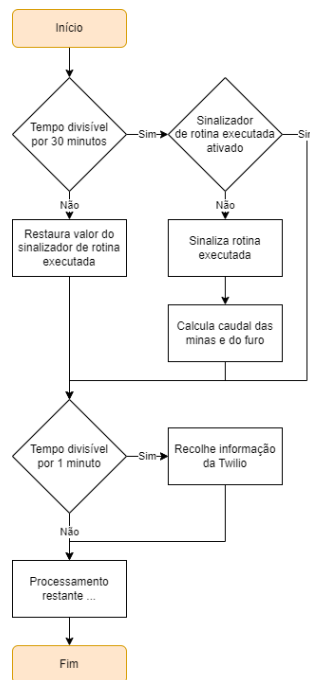


Figura 6.9: Primeira parte do ciclo infinito principal

A última parte do ciclo principal apresenta-se em 6.10, sendo que inicialmente é realizado um teste que pretende determinar se o dia da semana mudou. Caso esta

operação se verifique e na eventualidade do presente dia ser o domingo, é enviado para o email, um relatório com informação relativa ao sistema nos últimos 7 dias.

A parte final do ciclo principal, apenas testa algumas condições para tentar detetar anomalias graves.

A primeira anomalia consiste na deteção de numa falha na receção de uma mensagem periódica que possibilita determinar o bom funcionamento do código de interação com a base de dados e o bom funcionamento do *broker* MQTT.

A segunda e terceira anomalia são semelhantes, sendo que ambas sinalizam a deteção de água por parte do sistema de deteção de sobrecargas do reservatório inferior. Caso a deteção anterior se verifique, esta é conjugada com o sinal do fim de curso do módulo de medição dos níveis dos reservatórios, sendo que é enviado um SMS ao utilizador indicando a ocorrência de uma sobrecarga no teste válido de ambos os sensores. Caso o fim de curso falhe a deteção de água, envia-se um aviso indicando uma possível sobrecarga.

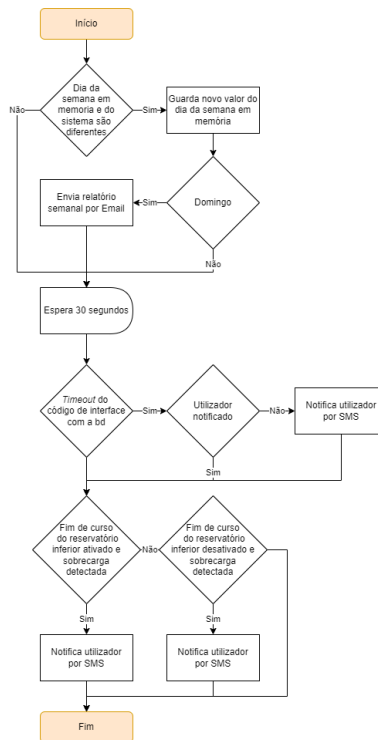


Figura 6.10: Segunda parte do ciclo infinito principal

### Lançamento de serviços de processamento de tramas MQTT e eventual resposta

Cada cliente envia de forma temporizada, uma trama de “ping” ao sistema, responsável por determinar o tempo de viagem de ida e volta de uma mensagem. Sempre que o sistema recebe uma destas mensagens, é executado o algoritmo apresentado em 6.11, sendo que esta rotina consiste num ciclo “For” que percorre uma lista que guarda informação sobre os clientes ligado, quando for encontrado um cliente com o mesmo identificador, assinala-se que a idade da última atualização passou a ser 0.

Sempre que este ciclo é executado, verifica-se se a última atualização de cada cliente não ultrapassou os 3 segundos. Esta é a condição que pretende eliminar utilizadores desconectados. Após a lista ter sido percorrida, sem que o identificador testado não tenha sido encontrado em nenhuma posição, significa que houve uma nova ligação, sendo que é adicionada uma nova entrada à tabela.

Acompanhado a cada cliente, acrescenta-se alguma informação adicional, sendo que esta contem o seu “ping” e o dispositivo eletrónico que está a utilizar para realizar a ligação, podendo este variar entre um telemóvel ou computador. Este algoritmo serve para realizar a contagem do número de utilizadores ligados ao sistema, valor que é reportado entretanto na aplicação web.

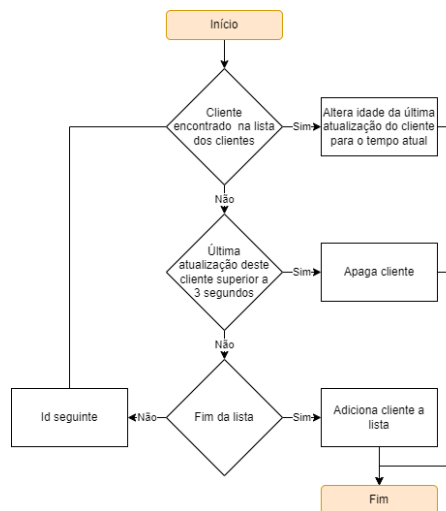


Figura 6.11: Ciclo que determina o número de clientes ligados

Todas as respostas geradas pelos serviços com destino à aplicação web, passam primeiramente por um teste semelhante ao do excerto 6.7, que se destina a verificar se o sistema está em modo isolado ou não. Este teste é logo realizado no início da execução da função, a publicação dos dados em concreto, ocorre na linha 4.

```

1 def web_publish(where, data):
2     if MainVars.SysMode_State != 3:
3         client.publish(where, send)

```

Lista 6.7: Resumo da função utilizada para verificar se podem ser enviadas tramas para a aplicação web

### API de SMS

Como foi referido anteriormente, utilizou-se um serviço fornecido pela Twilio para enviar mensagens de texto para o telemóvel do utilizador. Como este serviço é pago, ele é utilizado apenas quando há um alerta muito grave que requer atenção imediata.

Os custos associados ao envio e receção de mensagens dependem do número de segmentos, conjunto de 160 caracteres, enviados em cada transmissão e podem ser consultado em 6.2. Note-se que optou-se por realizar uma reserva de um número do Reino Unido já que era bastante mais económico. Não houve nenhum acrescento de taxas relativo a realizar-se um envio de mensagens internacionais.

Na presente implementação não houve utilização do serviço de receção de SMS, apenas envio. No entanto reconhece-se que poderia ser implementado um método de controlo do sistema de último recurso por mensagem, apesar de ser um meio de comunicação bastante lento.

Tabela 6.2: Custos associados à Twilio

	Reino Unido	Portugal
Custo de reserva do número	2€/mês	15€/mês
Custo de envio de SMS	0.045€	0.045€
Custo de receção de SMS	0.0075€	0.045€

As funções utilizadas para realizar a interação entre a aplicação desenvolvida e a Twilio foram baseadas na documentação fornecida, sendo que apenas se utilizam duas funcionalidades. A primeira consiste no pedido de transmissão de uma mensagem, a segunda consiste na recolha do saldo restante da conta.

### Cálculo do caudal das minas e do furo

Uma das operações mais importantes realizadas pelo sistema, é a determinação do caudal proveniente das diferentes fontes que alimentam a habitação.

No presente tópico, será realizada uma explicação do algoritmo que trata os dados recolhidos da variação do nível do reservatório superior, local que armazena o volume de água providenciado pelas minas. Note-se que o algoritmo que determina o caudal proveniente do furo é semelhante.

Para determinar o caudal, realiza-se uma regressão linear dos valores do nível do reservatório, consoante o tempo. Naturalmente que os valores da variação utilizados não podem depender de mais nenhum processo a não ser o enchimento realizado pelas minas, sendo que, sempre que for realizada alguma operação que influencie o volume de água do reservatório, os valores lidos do seu nível nesses instantes não podem ser utilizados para os cálculos.

A primeira operação realizada pelo algoritmo que calcula o caudal é a recolha de dados da base de dados. Os dados são recolhidos da tabela que guarda a informação sobre o sistema com uma resolução de 5 minutos, sendo que são recolhidos os últimos 2016 dados presentes. Tal operação resulta na leitura de dados recolhidos nos últimos 7 dias.

Da tabela recolhe-se especificamente 3 dados: o “timestamp”, dado relativo à data e hora de recolha dos dados, o “topres\_caud”, informação sobre o caudal que circula no caudalímetro de comunicação com o reservatório, permitindo determinar se está a ocorrer uma operação de descarga que irá influenciar o seu nível e, o “topres\_lvl”, dado relativo ao nível da água do reservatório superior.

Após a recolha da informação ter sido realizada, é aplicado uma primeira rotina que pretende remover informação inválida, isto é, valores marcados como “null”, sendo que são substituídos por um valor de nível ou caudal estimado. Note-se que o aparecimento de valores inválidos costuma ser esporádica e de baixa extensão. Da forma como foi implementado o sistema, não é possível aparecer um valor de data e hora inválido.

O algoritmo em 6.12 é executado dentro de um ciclo *For* sendo que são testados todos os valores recolhidos da base de dados. Inicialmente é verificada a invalidade do dado testado, sendo que em caso de retorno positivo da condição, é iniciado o processo de determinação do seu valor esperado.

Primeiramente recua-se uma posição na lista de dados recolhendo-se este valor, sendo que é frequentemente válido, em seguida, é executado um novo ciclo *For* que pretende determinar o dado válido seguinte, recolhendo-se o seu conteúdo e o índice da sua posição. Em seguida, realiza-se um cálculo que pretende determinar a distancia entre o valor válido anterior e o valor válido seguinte, determinando-se também a diferença entre estes.

O cálculo do valor esperado depende da distancia entre os valores válidos, isto é, caso apenas falte um número na sequencia, o resultado esperado é calculado sendo meia variação superior ao dado de índice imediatamente anterior. Caso exista uma distancia de, por exemplo, dois campos inválidos, o resultado esperado é determinado como um terço da variação.

Após o valor inválido ser substituído por um resultado esperado, este é guardado numa nova lista.

É importante notar que a primeira condição realizada neste algoritmo, o teste da invalidade de um número, caso falhe, significa que foi lido um dado bom, sendo que todo este processo é ignorado, guardando-se a informação diretamente na nova lista sem haver nenhuma alteração.

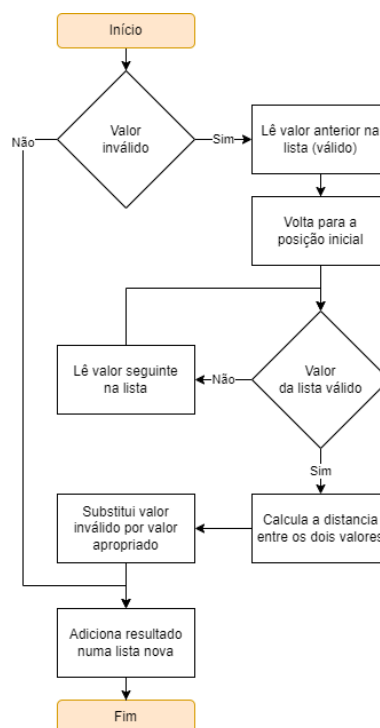


Figura 6.12: Alteração de valores inválidos por valores estimados

Como foi referido anteriormente, apenas é aplicada uma regressão linear aos valores do nível do reservatório que não foram perturbados. Para tal, divide-se a trama total em múltiplas secções, sendo que o início e o fim de cada secção é caracterizada pela medição de algum caudal medido pelo caudalímetro da saída do reservatório.

Esta operação é realizada pois se houver alguma descarga deste reservatório, o valor recolhido do nível torna-se pouco fiável pois a flutuação do volume de água não deve apenas ao enchimento provocado pelas minas.

O fluxograma em 6.13 representa a porção de código que divide a trama principal em secções. Primeiramente é verificado se o nível do reservatório está abaixo de um patamar predeterminado, sendo que não se calcula a variação do nível com valores superiores a 93% da capacidade total do depósito. Esta operação tende a compensar, com uma larga margem, dados inválidos retornados pelo sensor devido a possíveis descalibrações ou desvios. Em simultâneo verifica-se se o caudal de saída deste reservatório é inferior a 0.1 litros por minuto.

Na eventualidade de ambas as condições se verificarem, executa-se uma condição que pretende determinar se as listas auxiliares que armazenam as secções, estão a ser preenchidas pela primeira vez. Caso se verifique, é guardado o tempo em que foi recolhido o primeiro dado presente nesta lista.

Sempre que é realizada uma escrita de um novo valor recolhido, é subtraído ao seu tempo de captura, o valor previamente guardado em memória, isto permite referenciar a evolução temporal de todas as secções a 0.

Após a finalização anterior, é verificado se foi atingido o fim da trama principal, na eventualidade desta condição se verificar, ou na eventualidade de existir um valor de caudal superior a 0.1 litros por minuto, testa-se se o comprimento das listas auxiliares é superior a 10 elementos guardados. A aprovação em todos estes testes significa que foi recolhida mais uma secção, sendo esta guardada numa lista de resultado deste código.

Com a finalização das rotinas anteriores, surge neste ponto uma lista que contém as várias secções de valores, sendo que estas são introduzidas num algoritmo semelhante ao 6.8, sendo que na linha 2 é realizada uma regressão linear para cada secção, em seguida, na linha 3 é guardado num *array*, o valor da variação do nível marcado por cada secção.

O cálculo do caudal é realizado na linha 4, utilizando-se o declive da regressão de cada secção, o volume total do reservatório e o tempo total de recolha de dados, resultando um valor expresso em litros por dia. Este valor é guardado numa lista de saída em conjunto com um dado que determina o comprimento desta secção,

Por fim, na linha 5 e 6, são adicionados a um gráfico, os dados brutos de cada secção e a linha resultante da regressão linear, pretendendo-se demonstrar visualmente, ao utilizador, o resultado da operação efetuada por este algoritmo.

```

1  for index in range(len(y1)): # corre todas as listas da lista de
    resultados
2      fit = np.polyfit(x1[index], y1[index], 1) # regressao linear
3      variation.append(fit[0] * x1[index][len(x1[index]) - 1]) #
    calculo da variacao do nivel em %
4      total.append([len(x1[index]), fit[0] * x1[index][len(x1[index])
    ] - 1] * 24 / (len(x1[index]) * 5 * 60 / 3600) / 100 * 3250])
    # calculo do caudal
5      mines_graph_lin_reg_x.append([0, x1[index][len(x1[index]) -
    1]]) # grafico - linhas usadas para realizar a regressao

```

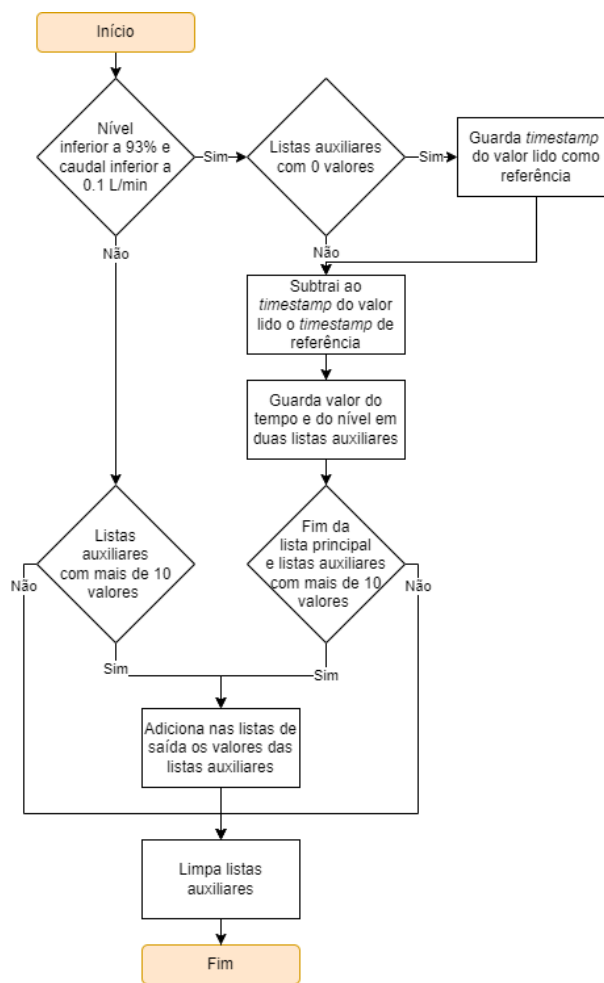


Figura 6.13: Separação dos dados em secções contínuas

```

6 mines_graph_lin_reg_y.append([fit[1], fit[1] + fit[0] * x1[index][len(x1[index]) - 1]]) # grafico - linhas resultantes da regressao

```

Lista 6.8: Função utilizada para determinar a taxa de enchimento do reservatório e o caudal da fonte

Com os valores da variação do nível de cada secção, guardados na lista de variações, determina-se se houve uma alteração superior a 8% do nível de água no reservatório, sendo que em seguida, precede-se ao cálculo do valor médio dos caudais determinados.

Neste cálculo final, a influência de cada valor tem um peso dependente do comprimento da secção que o originou, sendo que este quando maior o comprimento da lista, mais peso terá este valor sobre o resultado da média. Esta influência rege-se por uma operação cúbica.

### Serviço do controlador remoto da eletrobomba

Este conjunto de funções assíncronas pretendem controlar e responder a transmissões enviadas pelo controlador remoto da eletrobomba.

No fluxograma presente em 6.14 envia mensagens periódicas com informação e comandos para o controlador remoto. Primeiramente é verificado o estado da máquina, caso esta se encontre ligada, a cadencia de envio de mensagens é drasticamente reduzida, para permitir um tempo de resposta menos demorado. Na eventualidade da bomba estar parada, a velocidade de comunicação com o controlador remoto é reduzido para um valor padrão de 2 segundos, de forma a libertar recursos utilizados.

Seguidamente, verifica-se se existe alguma intenção de ligar a eletrobomba por parte do programa. Esta operação é verificada por um algoritmo que determina se a operação pode ser executada ou não. Caso se verificar a validade de um pedido de arranque da bomba, a sua aceitação e o modo de funcionamento do sistema diferente de parado, é registado numa *flag* uma ordem para atuar na eletrobomba.

No final deste algoritmo, é codificada e enviada numa mensagem o nível da água no furo, mostrado no LCD do controlador, seguindo-se da codificação da *flag* de controlo da bomba e dois sinalizadores de reconhecimento do estado dos botões.

No fluxograma 6.15 apresenta-se a resposta enviada pelo controlador após a receção de uma mensagem enviada pelo algoritmo anterior.

A primeira operação realizada é a marcação da idade da última transmissão como 0, seguindo-se do processamento de dados que simplesmente descodifica a trama recebida e guarda os diferentes valores em memória.

Em seguida, é testado o valor lógico do térmico de proteção e do relé de nível, sendo que se ambos forem verdadeiros significa que a bomba está em operação, guardando-se o resultado deste teste em memória. Por fim, são executados duas rotinas semelhantes que pretendem determinar se algum botão foi pressionado do lado do controlador, provocando uma reação.

Sempre que um botão é pressionado do lado do controlador, esta informação é fixada num registo interno e, enviada na trama de dados. Após o servidor responder ao pedido, é sinalizado através de uma *flag* o reconhecimento da atuação do botão, sendo que quando o controlador receber esta informação, colocará o seu registo interno com um valor padrão, aceitando novos pedidos a partir deste ponto.

Podem ser atuados os botões de *start* e *stop* implementados no módulo desenvolvido, que pretendem pedir um arranque ou paragem da eletrobomba, sendo que esta decisão é tomada por outro algoritmo de controlo de prioridades.

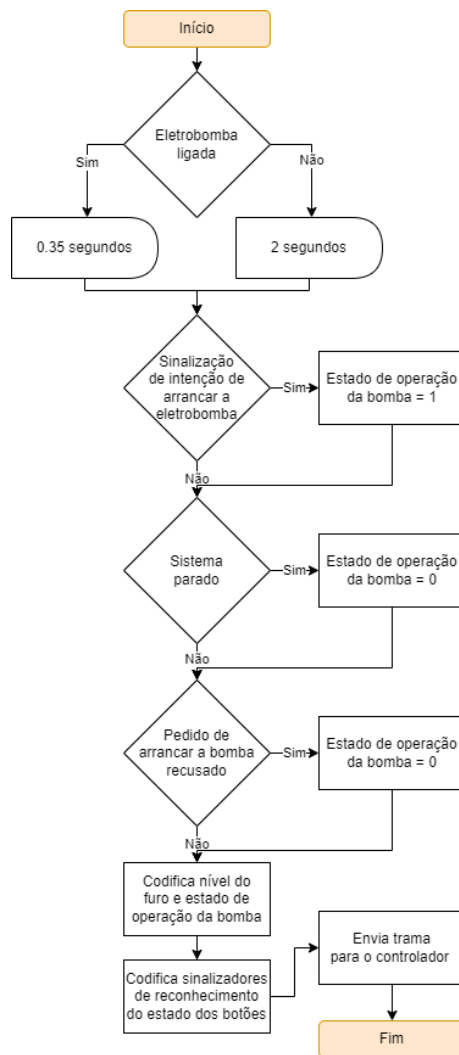


Figura 6.14: Serviço de controlo do controlador remoto da eletrobomba

Existe uma terceira função que apenas responde a comandos provenientes da aplicação web. Este comando indica que o utilizador pediu o arranque ou a paragem da eletrobomba e só pode ser enviado com o sistema em modo manual.

A tarefa da função não é nada mais nada menos que inverter o estado do registo de sinalização de intenção de atuação na bomba.

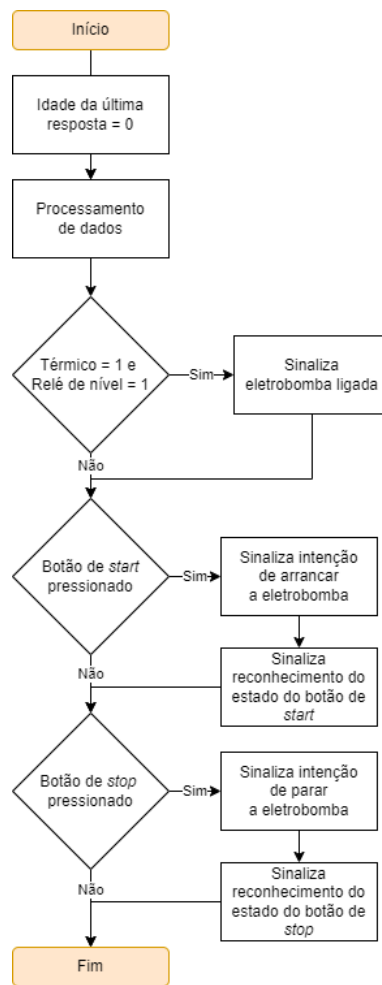


Figura 6.15: Algoritmo de resposta a mensagens enviadas pelo controlador remoto da eletrobomba

### Serviço do controlador remoto do sistema de rega auxiliar

As funções apresentadas em seguida são responsáveis por recolher informação do controlador remoto do sistema de rega auxiliar, assim como enviar comandos de controlo.

Como foi descrito num capítulo anterior, podem ser enviadas até 5 mensagens diferentes para o controlador, sendo que estas estão apresentadas na tabela 6.3.

Tabela 6.3: Possíveis mensagens enviadas para o controlador remoto do sistema de rega auxiliar

Indicador de trama	Função
“A”	Apenas pede informação e envia intensidade do brilho

Tabela 6.3: Possíveis mensagens enviadas para o controlador remoto do sistema de rega auxiliar

Indicador de trama	Função
“B”	Altera modo de funcionamento e estado manual das válvulas
“C”	Acerta o relógio interno
“D”	Altera programação dos temporizadores de rega
“E”	Limpa anomalias

Os dois fluxogramas em 6.16 são exemplos de algoritmos responsáveis por realizar um processamento de comandos recebidos da aplicação web e enviar uma ordem para o controlador remoto.

Ambos iniciam o processo realizando a separação dos comandos enviados pela aplicação. Em seguida, no caso do algoritmo (a), parte desta informação é convertida de texto para índices numéricos que indicam ao controlador qual dos registos devem de ser alterados, já no caso do algoritmo (b), é realizada uma troca lógica do valor indicado no comando enviado.

Por fim, em ambas as rotinas, é enviado para o controlador uma mensagem que se inicia com o indicador de trama, indicando a operação que deve de ser realizada, em seguida são enviados os novos dados processados.

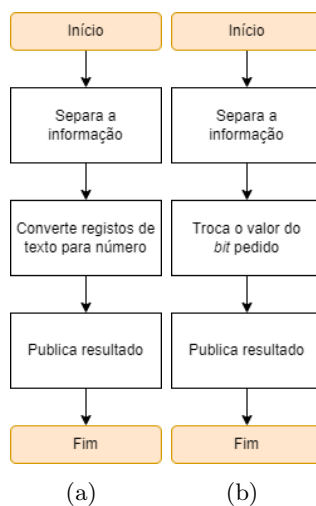


Figura 6.16: Processamento de comandos enviados pela aplicação web (a) Que alteram a programação dos temporizadores (b) Que alteram o estado de funcionamento do sistema e das válvulas em modo manual

O algoritmo em 6.17 simplesmente pede informação ao módulo, de forma temporizada, sendo que a cada pedido é enviado o valor do brilho do indicador luminoso.

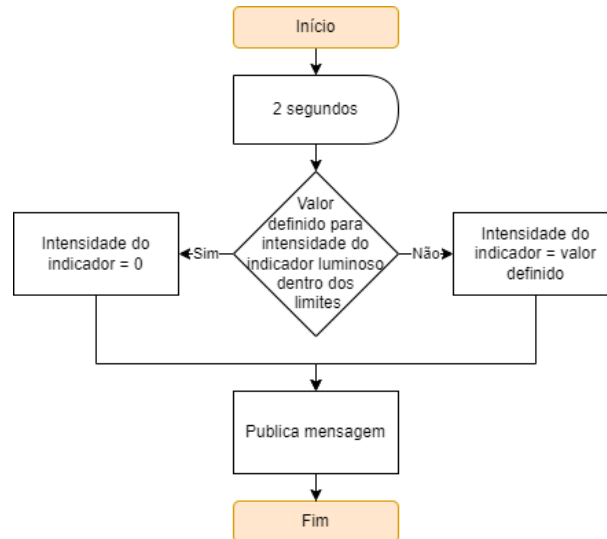


Figura 6.17: Serviço de recolha de informação do controlador remoto do sistema de rega auxiliar

Note-se que todas as mensagens produzidas por estas funções são publicadas no mesmo tópico ao qual o controlador remoto subscreve. Sendo que após receber e processar um comando, envia informação recolhida por si.

O algoritmo que processa essa informação da parte do servidor, simplesmente isola os diferentes dados, separados por “:” e guarda a informação em memória.

### Serviço do controlador remoto de recolha de informação meteorológica

O serviço deste controlador remoto é semelhante aos serviços dos controladores anteriormente apresentados, pelo que não merece explicação. O seu propósito principal é pedir ao controlador remoto que envie a informação que este recolheu, sendo que este pedido é realizado de forma temporizada sendo que é enviado um pedido a cada 2 segundos.

Existem duas funções que foram implementadas, que respondem a pedidos realizados através da aplicação web. Estes pedidos resultam na alteração do brilho e no estado de funcionamento da resistência de remoção de condensação. Sendo que ambos os valores são guardados na base de dados aquando alterados.

A única função a que foi dada notoriedade encontra-se apresentada no fluxograma 6.18. Este algoritmo é responsável por processar e armazenar a informação enviada pelo controlador remoto. Primeiramente é verificado se o sistema está

a iniciar, isto é, se ainda se encontra a recolher o primeiro conjunto de valores, sendo que na primeira recolha os valores resultantes podem variar demasiado.

Na eventualidade do sistema assinalar a conclusão da sua inicialização, os dados recolhidos são finalmente guardados. A grande maioria das variáveis é guardada diretamente na memória, com uma exceção que requer um tratamento diferente, sendo este o contador de precipitação, uma vez que retorna um incremento de eventos e pretende-se obter a pluviosidade em relação ao tempo.

Primeiramente verifica-se através de uma condição, se o valor deste contador é superior a 0 e se a última atualização do sistema não ultrapassou os 15 segundos. Na verificação desta condição, é realizada a diferença entre o valor atual deste contador e o seu valor na atualização anterior, sendo que o resultado é entretanto multiplicado por um valor para converter a escala, retornando o valor da precipitação em milímetros. Por fim, guarda-se em memória o valor atual do contador.

A condição explicada no paragrafo anterior é importante para prevenir a leitura de valores errados no caso do reinício do servidor ou do controlador remoto.

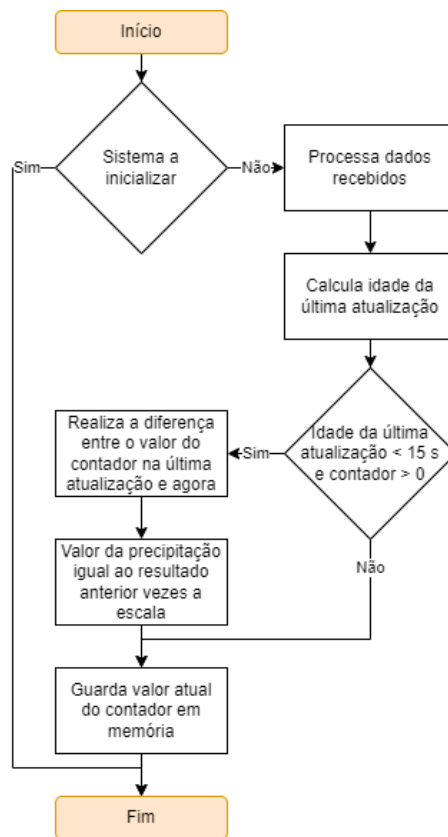


Figura 6.18: Serviço de processamento de informação do controlador remoto de recolha de informação meteorológica

### Serviços temporizados de ordenação de escrita de valores na base de dados

Existem múltiplos serviços que realizam um pedido temporizado ao algoritmo desenvolvido de escrita na base de dados, que se resume numa ordem de introdução de um nova entrada, sendo que realizam o processamento e preparação dos valores, assim como o cálculo do tempo de referência da entrada.

Normalmente cada serviço controla o pedido de atualização de uma única tabela de registo de dados, sendo que a presente implementação contém três agrupamentos principais de *datalog*. A principal razão de separação dos algoritmos deve-se ao facto de terem bases de tempo diferentes, sendo estas de 5 segundos, 1 minuto e 5 minutos.

A função que pretende sincronizar a execução da rotina com o relógio do sistema, encontra-se apresentada em 6.19. Esta função pretende determinar o tempo restante até à próxima execução, iniciando um temporizador que adormece o algoritmo durante o tempo calculado.

Para determinar este valor, começa-se por realizar a recolha do valor atual do relógio do sistema. Em seguida, realiza-se uma divisão inteira com arredondamento por defeito, do valor do relógio pela base de tempo pretendida. Os passos seguintes consistem na soma de 1 a esse valor e a multiplicação do resultado pela base de tempo inicial. Estas ações retornam o valor que estará presente no relógio do sistema, no momento em que deve de ser iniciado um novo processamento.

O passo final da rotina de temporização, consiste na determinação do tempo que falta para atingir esse objetivo, iniciando-se um *delay* com o valor obtido.

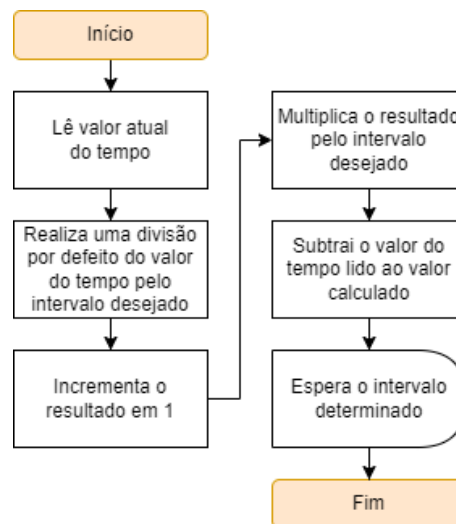


Figura 6.19: Representação da função de temporização dos serviços de escrita na base de dados

### Serviço do controlador remoto dos sensores de sobrecarga e pressões

O controlador remoto dos sensores de sobrecarga e pressões tem a principal tarefa de detetar uma sobrecarga do reservatório inferior. O serviço de controlo do sensor é responsável por pedir informação recolhida por este, assim como enviar ordens que resultam na atuação da eletroválvula de teste e do sensor em si, lembrando que este pode ser ligado ou desligado.

No algoritmo de decisão deste serviço é inicialmente realizada uma consulta de um registo que indica qual dos três modos de funcionamento, está o controlador a operar. O fluxograma que demonstra esta porção de código está representado em 6.20.

Os três modos de funcionamento do controlador estão explicados em seguida:

- Modo automático - Liga o sensor para detetar água, desligando-o por um determinado tempo no caso de deteção.
- Modo de teste - Executa determinados passos que permitem determinar a validade do correto funcionamento do sensor.
- Modo de hibernação - Desliga o sensor e a eletroválvula e mantém-se sempre neste estado.

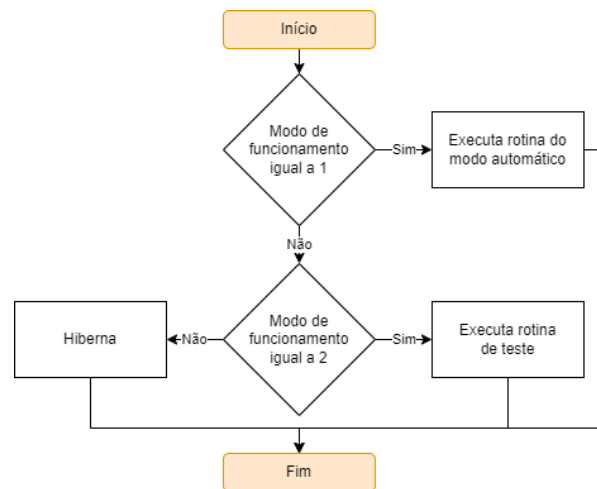


Figura 6.20: Diferenciador principal de decisão de execução de rotinas

No fluxograma 6.21 apresenta-se a rotina simplificada que é executada no modo automático. Como já tinha sido referido, este sensor monitoriza constantemente a presença de água na instalação de alívio de sobrecargas do reservatório principal. Na presença de água num período superior a 5 segundos, contínuos ou não, é despoletada uma anomalia avisando-se o utilizador. Em seguida, o sensor ativa-se uma vez a cada minuto monitorizando novamente o sistema.

Na eventualidade do sistema estar caracterizado com estando em sobrecarga, se ao fim de cinco minutos no estado de anomalia não tenha havido nenhuma deteção de água, é restabelecido o funcionamento do sensor, limpando-se a sinalização estado anómalo e retornando à sinalização do estado normal.

Para implementar tal processo, iniciou-se com a ativação da *flag* que ordena a atuação do sensor, seguindo-se de uma verificação da presença de água.

Caso seja encontrada água, o contador que restabelece o sistema é limpo, incrementando o contador que determina a presença de água. Caso esta condição não se verifique, incrementa-se o contador de reposição do sistema até que este atinja o valor 300, coincidindo com um tempo de espera de 5 minutos. Após este valor ser alcançado, força-se a limpeza da anomalia de sobrecarga.

Na possibilidade do registo que conta o número de deteções do sensor atingir o valor 5, desliga-se o sensor, iniciando-se o incremento de um registo auxiliar que pretende reativar o dispositivo. Na eventualidade desta condição ser negada, limpa-se o registo auxiliar, sendo que se encontra preparado para um novo ciclo.

Quando o registo auxiliar atingir o valor 60 que coincide com a passagem de 1 minuto desde a sua primeira incrementação, restaura-se a variável que conta o número de deteções de água, sendo que se inicia o algoritmo descrito novamente.

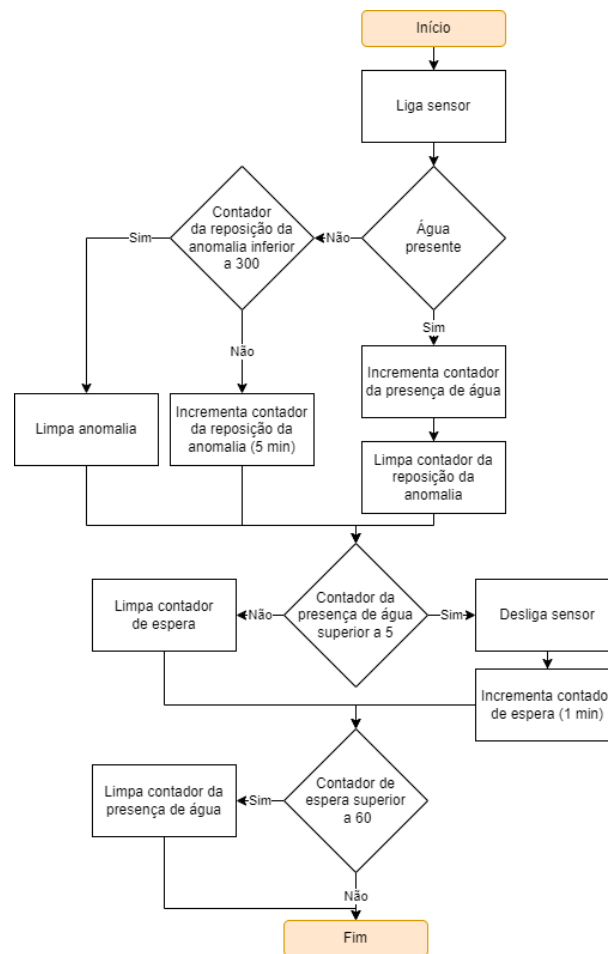


Figura 6.21: Rotina executada com o sensor no modo automático

Se o sensor for colocado no modo de teste, é executada uma rotina que consiste fundamentalmente em três passos, primeiro verifica-se se não há presença de água no sistema. Após esta operação, liga-se a eletroválvula até existir uma deteção por parte do sensor. Por fim, inicia-se um ciclo de finalização do teste que consiste na desativação da válvula e na verificação da ausência de água novamente.

O ciclo inicial que verifica a ausência de água encontra-se representado em 6.22. Primeiramente liga-se o sensor, verificando-se o seu resultado. Na presença de água, limpa-se o valor presente num registo auxiliar, na ausência, incrementa-se o seu valor em 1 por execução. Em seguida, verifica-se se o valor deste registo auxiliar já foi incrementado 30 vezes seguidas. Caso esta condição se verifique, passa-se para a operação de teste seguinte, caso contrário, incrementa-se um contador de *timeout*. O contador de *timeout* pretende definir um tempo máximo de execução deste ciclo inicial.

Na presente implementação, definiu-se que o sensor não pode encontrar água durante 30 segundos seguidos, sendo que tem no máximo 5 minutos para o realizar. No fim do tempo máximo de execução da rotina assinala-se uma anomalia de falha deste primeiro estágio.

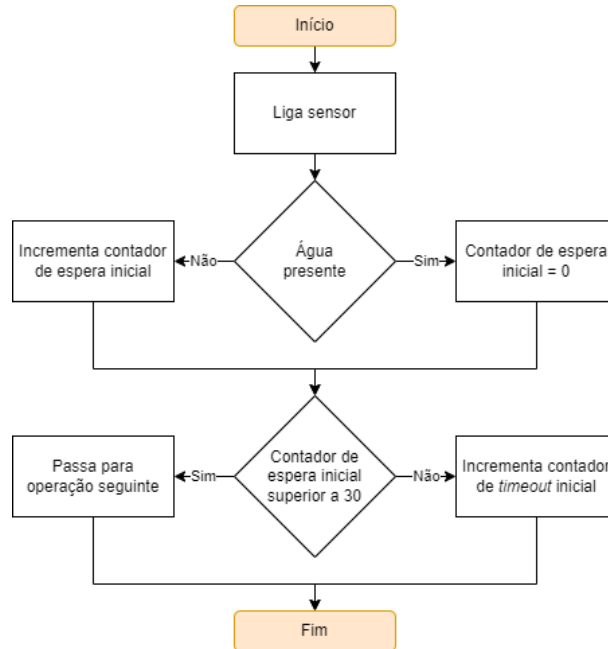


Figura 6.22: Primeira parte da rotina executada com o sensor no modo de teste

No diagrama apresentado em 6.23, apresenta-se a segunda parte do teste do sensor, neste momento, pretende-se ativar a válvula de teste até que o sensor detete água. Para tal, primeiramente pede-se permissão à rotina de gestão do sistema, quando o pedido for aceite, ativa-se a válvula e o sensor.

Na presença de água, termina-se este processo e salta-se para o estágio final do teste. Na ausência de água, é incrementado um contador de erro. Quando este contador atingir o valor 30, que coincide com a passagem de 30 segundos desde a ativação da eletroválvula, na ausência da deteção de água, ativa-se uma anomalia.

No terceiro estágio do teste é executada uma rotina semelhante à primeira que pretende determinar a ausência de água durante um determinado tempo, no entanto, foram mudados os tempos das metas, sendo que apenas é preciso ficar 10 segundos sem detetar água, ao invés de 30 e, a anomalia ocorre ao fim de 40 segundos, ao invés de 5 minutos.

Note-se que na presença de uma anomalia, o sistema é parado sendo ativado o estado de hibernação do sensor. Caso o teste termine com sucesso, o estado de funcionamento do módulo é alterado para o estado automático.

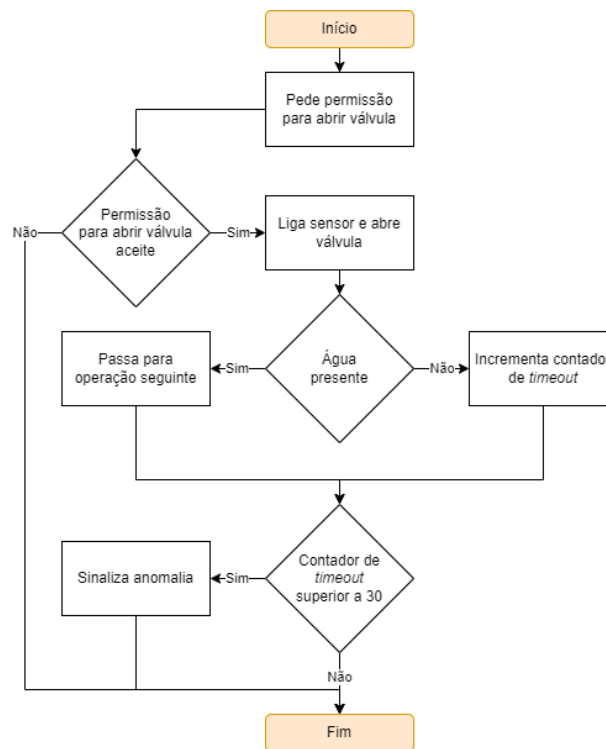


Figura 6.23: Segunda parte da rotina executada com o sensor no modo de teste

Todos os comandos resultantes dos algoritmos anteriormente executados, são condicionados pelo modo de funcionamento do sistema, isto é, o modo de funcionamento “Parado” tem uma prioridade mais alta que qualquer outro algoritmo de decisão. No fim de execução do serviço, é publicado num tópico específico, o resultado de processamento, enviando para o controlador remoto a decisão tomada.

### Serviços de controlo e alteração do modo de operação do sistema

Estes serviços são responsáveis por determinar o modo de operação do sistema, sendo que alteram o registo mais influente na aplicação desenvolvida.

No algoritmo 6.24 apresenta-se a rotina desenvolvida que controla o registo do modo de funcionamento do sistema. Este registo pode conter quatro estados diferentes como já tinha sido referido anteriormente, estando estes relacionados aos modos de funcionamento. O modo padrão de operação é o automático, em que o sistema é livre de tomar as próprias decisões, normalmente este é alterado para o modo manual, sendo que é enviado pela aplicação web um pedido de alteração em conjunto com o tempo requerido para ativar esta funcionalidade. Os dois modos restantes são o modo parado, em que o sistema para todas as operações e, por fim, o modo físico em que todas as comunicações com a aplicação web são

ignoradas, passando o sistema a funcionar num estado semelhante ao automático.

Voltando ao fluxograma, a primeira condição verificada é se o sistema se encontra em modo manual com uma sessão expirada, caso a condição se verifique, altera-se o registo imediatamente para automático. Em seguida, verifica-se se o tempo de sessão ainda é válido mas inferior a 2 segundos, sendo que se a condição for verificada, guarda-se num registo auxiliar o tempo de sessão restante. Esta operação permite realizar um novo processamento, no exato momento em que uma sessão do modo manual termine, revogando-a e alterando de imediato o estado do sistema para modo automático

Em seguida, publica-se num tópico específico o modo de funcionamento do sistema que irá indicar aos diferentes utilizadores ligados através da aplicação web, o estado de operação do código de controlo central. Este algoritmo corre normalmente 1 vez a cada 2 segundos, sendo que a única exceção de alteração do tempo de dormência acontece caso uma sessão esteja a terminar.

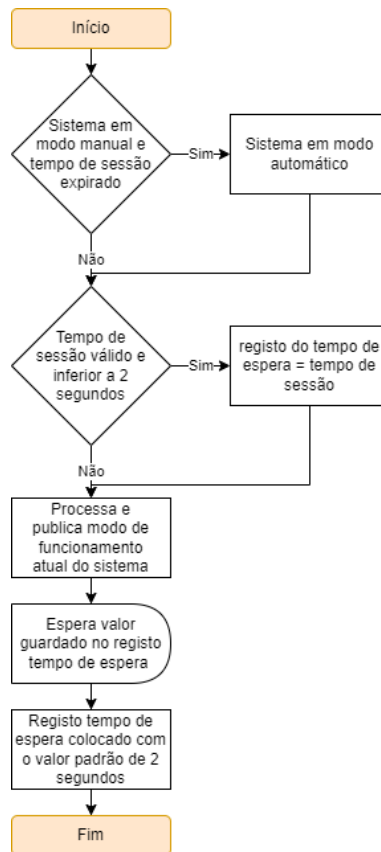


Figura 6.24: Serviço de controlo do modo de operação do sistema

O algoritmo apresentado em 6.25 é responsável por responder a pedidos realizados por um utilizador para realizar alterações no modo de funcionamento do sistema. Após ser recebido um comando, é executada a seguinte rotina que pretende processá-lo, sendo que primeiramente é determinada a operação pretendida por um conjunto de condições.

A única operação que se considera relevante explicar consiste na realização de um pedido para colocar o sistema em modo manual. Primeiramente é importante esclarecer que esta operação só pode acontecer com o sistema previamente em modo automático ou manual, sendo que com a receção de um pedido, regista-se o tempo de sessão requerido, configurando-se um registo com o tempo que o relógio da máquina marcará no fim da sessão, sendo que esta é considerada como expirada quando for ultrapassada.

Todas as alterações realizadas por ambos os fluxogramas são guardadas de forma permanente com recurso à base de dados, sendo que é importante relembrar que este é o primeiro registo carregado aquando a inicialização do sistema.

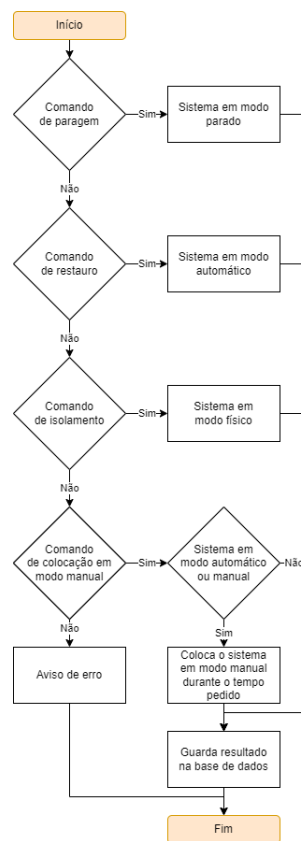


Figura 6.25: Serviço de alteração do modo de operação do sistema

### **Rotina de temporização do sistema de rega principal**

O controlo das eletroválvulas de rega dos espaços verdes principais da habitação é realizado pelo algoritmo em python desenvolvido no servidor.

Existem bastantes funções que realizam operações semelhantes ao sistema de rega do controlador remoto auxiliar, tais como as rotinas de alteração de um temporizador ou da sua validação, pelo que não serão cobertas novamente.

Foram implementados 5 temporizadores programáveis semelhantes aos do controlador anteriormente referido, sendo que a única diferença entre ambos é o local de armazenamento. Os programas do sistema de rega auxiliar são guardados no microcontrolador sendo que este realiza um controlo semi-independente, já os programas do sistema de rega auxiliares são armazenados na base de dados do servidor.

O único algoritmo que merece ser destacado é o algoritmo responsável por pedir a abertura das válvulas, sendo que este é consideravelmente diferente do controlador remoto.

Uma vez que este sistema funciona sobre uma hierarquia de prioridades, é necessário determinar se no momento em que são atingidas todas as condições de ativação um programa, pode haver uma atuação nas válvulas, pois o sistema pode estar a desempenhar outra função. Caso não haja uma permissão para abrir as válvulas, é registada a necessidade de adiar a atuação do programa, sendo que este é executado quando aparecer a primeira oportunidade.

Para combater este problema, são guardadas num registo auxiliar, todas as informações relativamente a um programa, como está representado na figura 6.26. Esta atualização do registo auxiliar acontece sempre no momento em que um programa deve de iniciar, apagando configurações antigas que estejam guardadas. Em seguida, é calculado um tempo de execução dependentes das configurações dos campos de execução de cada programa. Note-se que o código está preparado para influenciar este tempo consoante a previsão meteorológica e o valor da humidade do solo, no entanto, atualmente os algoritmos de influência apenas multiplicam o tempo total por 1, sendo que não existe nenhuma alteração real.

Verificando-se uma ativação do programa, o dia da semana atual foi configurado para ativar o sistema e um tempo de execução inferior a 45 segundos, atualizam-se certos valores num registo auxiliar, sendo que este contem o tempo de ativação do temporizador, o tempo de execução e as válvulas que devem de ser atuadas.

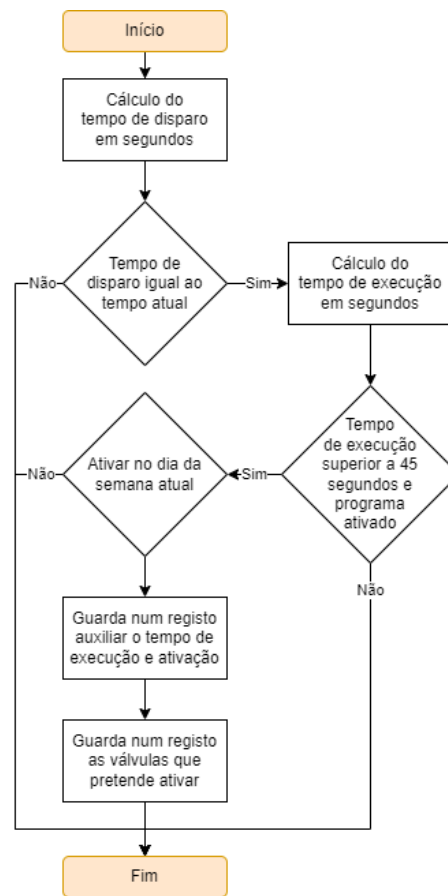


Figura 6.26: Serviço de registo de parâmetros para realizar uma atuação temporizada nas válvulas

O serviço que realiza uma atuação nas válvulas tendo em conta o sistema de prioridade encontra-se representado em 6.27. Este serviço é responsável por realizar os eventos programados, assim que houver oportunidade. Observando o fluxograma, verifica-se que primeiramente é realizado um teste sobre o modo de funcionamento do sistema, como já foi verificado em várias rotinas anteriores sendo que este algoritmo não é exceção, o programa só é atuado se o sistema não estiver parado.

Em seguida, verifica-se se foi dada permissão para realizar uma execução do restante algoritmo, que consiste fundamentalmente na testarem da ultrapassagem do tempo de disparo pelo relógio da máquina e na verificação do valor do registo do tempo de execução que tem de ser superior a 0.

Existe uma última condição que merece importância, sendo esta a verificação do tempo de atraso na execução do algoritmo, isto é, caso não seja dada permissão de atuação logo que esta for pedida, existirá um desfasamento entre o horário que deveria de atuar e, o eventual horário de funcionamento. Este desfasamento pode

ser tão grande, por exemplo quase 24h, que poderá resultar numa abertura das válvulas, momentos antes de ser executado o programa do dia seguinte, pelo que esta condição pretende limitar o atraso máximo impedindo uma aproximação demasiado grande do mesmo programa.

Sempre que todas as condições se verificarem significa que existem válvulas que devem de ser atuadas, após ser sinalizado um pedido de atuação dos dispositivos marcados, o registo que guarda o tempo de execução é decrementado, sendo que este ciclo repete-se sempre 1 vez por segundo, até o contador atingir o valor 0, sendo que neste momento uma das condições falha e dá-se como terminado o programa.

Note-se que a qualquer momento a permissão de execução deste algoritmo pode ser levantada, pelo que o decremento do contador e a atuação das válvulas são paradas de imediato. O ciclo resume-se após ser restabelecida a permissão de funcionamento, a não ser que o tempo de execução restante seja inferior a 45 segundos, nesse caso da-se como terminado o programa.

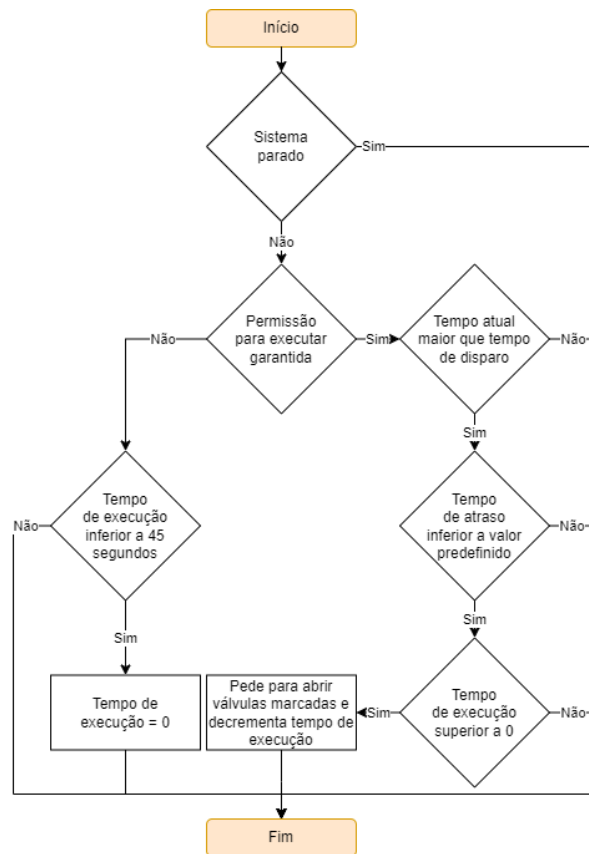


Figura 6.27: Serviço de ação para realizar uma atuação temporizada nas válvulas

### Serviço de prioridades

Destaca-se a importância deste serviço sobretudo para determinar se uma rotina deve de ser executada ou parada. Esta decisão depende sobretudo do estado de funcionamento do sistema.

Observando-se o algoritmo apresentado em 6.28, verifica-se inicialmente que é realizada uma primeira destinação sobre o modo de funcionamento do sistema, caso este esteja a operar em modo automático ou isolado, são executadas as rotinas de decisão de permissões de forma normal.

No caso apresentado, apenas se representa o processamento de um exemplo, sendo este o pedido realizado por parte do controlador remoto do sensor de sobrecargas, que pretende pedir uma ativação do sensor e/ou da válvula de teste, prosseguindo com a execução da sua rotina. De momento como as válvulas motorizadas não estão montadas, não existe nenhuma condição que se oponha a tal pedido em modo automático, pelo que é sempre aceite.

Caso o sistema não esteja a funcionar em modo automático ou isolado, volta-se a realizar um teste para determinar se o modo de operação é o manual, caso tal se verifique, apenas é recusada a execução de rotinas, sendo que todos os pedidos para abertura de válvulas são aceites. Estes pedidos apenas aparecem em resposta a uma ordem dada por parte do utilizador.

Por fim, se o sistema estiver em modo parado, são recusados todos os pedidos de execução de rotinas e de atuação de válvulas.

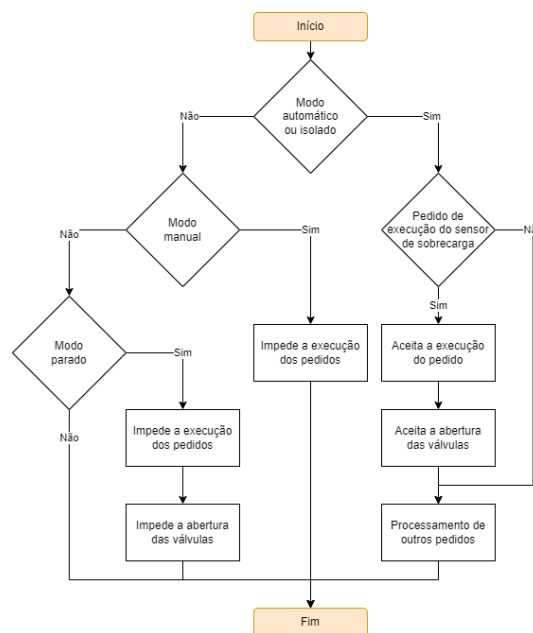


Figura 6.28: Serviço de decisão de prioridades

## Outros serviços

Existem outros serviços implementados que são de simplicidade bastante elevada, pelo que apenas será realizada uma menção rápida.

O primeiro serviço a ser mencionado é o de controlo do estado do compressor do sensor de nível do furo. Este desempenha duas funções simples, sendo a primeira a realização de um pedido de ativação do compressor durante 7 segundos, repetindo-se a cada 30 minutos. A segunda função corresponde à resposta de um pedido realizado pelo utilizador através da aplicação web, resultando numa alteração do estado do compressor e do estado de funcionamento do algoritmo. O resultado desta alteração depende do modo de funcionamento do sistema, caso esteja em modo automático, pode-se ativar ou desativar a rotina de controlo temporizado do compressor, caso o sistema esteja no modo manual, para além de se poder realizar a alteração anterior, é adicionado um terceiro estado que força a atuação do compressor.

Outros serviços que merecem ser reconhecidos de fora breve são: o de controlo do controlador remoto principal e, o de simulação das válvulas. Sendo que o primeiro, semelhante aos restantes controladores, apenas codifica o estado das válvulas do sistema de rega, do compressor e da eletrobomba auxiliar, enviando para o módulo desenvolvido e, aguardando uma resposta.

Já o serviço restante, apenas realiza uma alteração simulada do estado das válvulas motorizadas, sendo que o seu estado deveria de ser remetido pelo controlador principal, no entanto, estas ainda não se encontram montadas.

### 6.5.2 Algoritmo de Interface com a Base de Dados

Este algoritmo foi desenvolvido para realizar trocas de informação com a base de dados que costumam ocorrer com maior frequência ou que são bastante demoradas a executar. A troca de informação com o serviço principal acontece através do *broker* MQTT, o que torna os serviços completamente independentes, isto é, ambos funcionam sem a necessidade da outra implementação estar em execução, no entanto, as funcionalidades da aplicação desenvolvida são drasticamente afetadas.

O motivo que levou à separação do controlo do sistema e da interface com a base de dados em dois algoritmos diferentes, foi o facto de ter ocorrido uma advertência que não se conseguiu solucionar das várias vezes que foi tentado. Este problema ocorria, esporadicamente, numa escrita nas tabelas de *datalog*, em que a pesquisa de informação demorava várias ordens de grandeza a mais do que o normal, afetando o funcionamento do resto dos serviços.

É importante lembrar que vários sistemas atuam sobre dispositivos sob uma rotina de prevenção de falha de comunicações que despoletava várias vezes devido

ao problema na escrita na base de dados.

A solução encontrada consiste na separação de ambas as tarefas em algoritmos diferentes, sendo que se pretendia implementar um sistema de filas de mensagens, para não perder informação numa eventualidade do sistema ficar pendurado. Esta fila não foi implementada pela sua falta de necessidade, devido ao problema anteriormente descrito nunca mais se ter verificado.

### **Serviço de adição de dados a uma tabela**

Como foi referido várias vezes anteriormente, existem serviços que pedem ciclicamente que certos dados sejam armazenados numa tabela. O fluxograma em 6.29 representa o algoritmo que responde a tais pedidos.

Sempre que é recebida uma nova trama de dados através do *broker* MQTT, para que sejam inseridos numa tabela, o primeiro passo do algoritmo de processamento consiste na realização de uma recolha da última entrada inserida na base de dados. Com esta entrada e com a informação enviada na trama de dados, é possível determinar a diferença de idades de ambas. Este valor é entretanto dividido pela base de tempo usada para escrever na lista, lembrando que podem haver inserções de 5 em 5 segundos, de 1 em 1 minuto, entre outros.

Com esta divisão, é possível determinar a distancia entre ambas as entradas, ou seja, é possível determinar se são dados consecutivos, ou se faltou inserir algum pacote de dados anteriormente.

Em seguida, inicia-se um ciclo de inserção de dados na tabela, note-se que normalmente apenas é introduzido o pacote de dados recebido, mas se houver a deteção de falha de inserções anteriores, preenchem-se primeiramente os dados mais antigos com valores nulos e o tempo no qual deveria de ter ocorrido a sua recolha. Este ciclo serve para que haja coerência na base de dados.

No fim de toda a informação ter sido guardada na base de dados, termina-se este serviço, sendo que apenas será iniciado quando for recebido um novo pacote de dados.

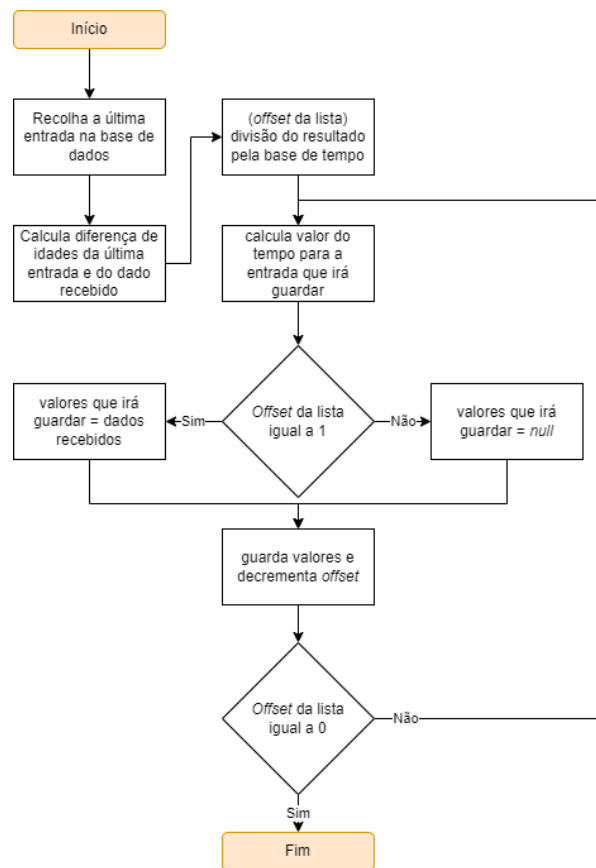


Figura 6.29: Exemplo de serviço de adição de dados a uma tabela

### Serviço de recolha e armazenamento de dados do IPMA

A rotina de recolha e armazenamento de dados do IPMA, apresentada em 6.30, é semelhante à função anteriormente explicada, no entanto, os dados armazenados na tabela provém do IPMA. O objetivo principal deste algoritmo é guardar a informação recolhida pelo instituto sobre a precipitação mínima e máxima registada, sendo que é fornecida em formato de dados separados por virgula (CSV) contendo informação dos últimos 2 meses.

Sempre que há uma execução do algoritmo, inicia-se por recolher os dados da API, seguindo-se da leitura da última entrada escrita na base de dados. Em seguida, é procurada na lista de dados da precipitação, o local onde se encontra a informação da última atualização da base de dados.

Após esta localização ser determinada, procede-se por armazenar na base de dados, todos os valores restantes na lista do IPMA, caso existam.

Este algoritmo tem o papel de armazenar na tabela todos os dados fornecidos pelo IPMA, de forma ordenada, impedido que haja a repetição de valores

previamente escritos.

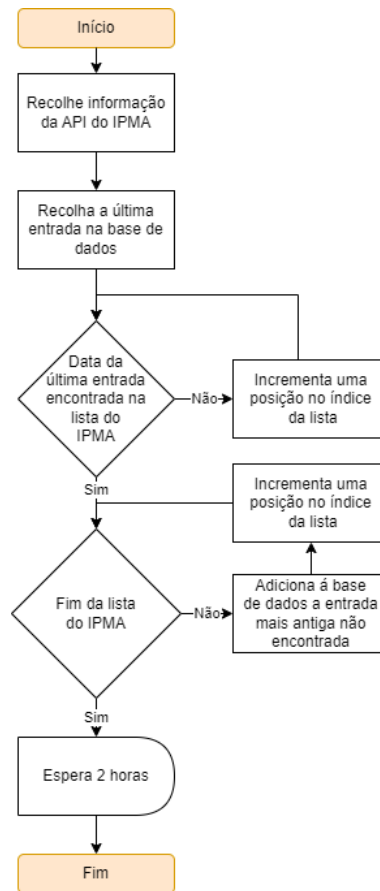


Figura 6.30: Serviço que guarda dados provenientes do IPMA

### Serviço de análise dos hábitos diários

No excerto seguinte, apresenta-se a *query* SQL responsável por agrupar e recolher dados que caracterizem os hábitos diários com uma extensão de 7 dias. Apesar da expressão estar simplificada, resume-se que o grupo de informação recolhida varia entre os consumos de água medidos pelos caudalímetros. A este conjunto de dados também é adicionado o valor de precipitação medida na extensão do tempo analisado, em virtude de ser um dado enviado nos relatórios semanais, aproveitando a expressão.

Esta informação é recolhida da tabela que armazena valores com uma base de tempo de 5 minutos, sendo que na expressão se encontra com o seu nome simplificado para “*table*”. O primeiro passo executado na *query* consiste na recolha de 2016 entradas com todos os dados que se prolonguem com início na data definida em “*start*”. O valor 2016 resulta no número de grupos de 5 minutos

presentes numa semana.

Toda a informação recolhida da tabela é agrupada em conjuntos que contêm o mesmo valor para a data, sendo que são realizadas operações matemáticas nos valores dentro de cada grupo. É demonstrado um exemplo na linha 2, em que no caso dos valores recolhidos pelos caudalímetros, é realizada a soma de todos os valores superiores a 0.2 litros por minuto, permitindo determinar o volume de água total que atravessou cada sensor, por dia.

Em seguida, na linha 5, os dados são organizados por ordem de idade, apresentando assim o formato final.

```

1 SELECT * FROM (
2     SELECT DATE(FROM_UNIXTIME('time')) AS ForDate, min('time') AS
   time, ... , sum(if('Caud' >= 2, 'Caud', 0)) AS Caud FROM (
3     SELECT * FROM table WHERE 'time' >= UNIX_TIMESTAMP(
   STR_TO_DATE('" + start + "', '%M %d %Y')) LIMIT 2016) table
4     GROUP BY DATE(FROM_UNIXTIME('time')) ORDER BY 'ForDate' DESC)
5 table ORDER BY 'ForDate' ASC;
```

Lista 6.9: Query SQL de análise dos hábitos semanais

## 6.6 Menu Principal da Interface Gráfica

Para facilitar a visualização dos valores recolhidos e calculados pelo sistema, assim como proporcionar um meio de configuração mais facilitado, procedeu-se à implementação de uma interface gráfica. Apesar de terem sido desenvolvidas diversas páginas diferentes, tais como de *login*, de verificação dos *datalogs*, entre outros, apenas se decidiu apresentar o menu principal, visto ser o único que merece notoriedade.

Neste menu é onde se localiza a maioria dos controlos que podem ser realizados através da aplicação, foi construído um esquema do circuito hidráulico recorrendo-se a gráficos vetoriais escalonáveis (SVG). É possível encontrar também diversos gráficos que apresentam vários valores recolhidos pelos módulos e pelos algoritmos de controlo.

Os únicos algoritmos que merecem notoriedade encontram-se em seguida.

### 6.6.1 Ligação ao broker MQTT

A vasta maioria de informação que é apresentada em tempo real na aplicação web, é transmitida através do *broker* MQTT.

No excerto de código 6.10 apresenta-se na linha 2, a declaração da criação de um cliente, note-se a definição do porto de ligação, o 8081 é referente ao *router*, sendo que as mensagens são entretanto reencaminhadas para o computador. Para além disso, repare-se na linha 1, que é criado inicialmente um identificador do

cliente gerado de forma aleatória. Este passo é crucial pois o *broker* não aceita ligações de clientes com a mesma identificação.

Este identificador é também utilizado pela aplicação de controlo desenvolvida para distinguir os diferentes utilizadores conectados.

```
1 var webpage_client_id = guidGenerator();
2 var client = new Paho.Client('domain.xyz', Number(8081),
  webpage_client_id);
```

Lista 6.10: Declaração de um cliente

No excerto 6.11 apresenta-se a função executada após ser estabelecido um canal de comunicação com o servidor. Após ser estabelecida tal comunicação, procede-se por realizar uma subscrição aos tópicos destinados à receção de informação. O tópico presente na linha 2 é onde circula toda a informação enviada pela aplicação desenvolvida em Python,.

Já no tópico da linha 3, é o local onde são enviados pacotes que pretendem determinar a latência da comunicação com o servidor, sendo também o local utilizado pelo algoritmos e controlo do sistema, para reconhecer os diferentes clientes.

No fim da execução desta função, são executadas duas novas, sendo que a primeira compromete-se a pedir ao servidor uma atualização de todos os valores apresentados na página web. Já a função seguinte, na linha 5, lança um serviço que calcula a latência da comunicação.

```
1 function onConnect() {
2     client.subscribe("world/wtr/home/data");
3     client.subscribe("world/wtr/ping/" + webpage_client_id);
4     mqtt_refresh_data();
5     Ping();
6 }
```

Lista 6.11: *Callback* de uma função após uma ligação

## 6.6.2 Recolha de outras informações

Existem várias informações apresentadas graficamente que são recolhidas através de pedidos HTTP convencionais. No excerto 6.12 apresenta-se a realização de um pedido GET, demonstrado na linha 7, à API desenvolvida no lado do servidor. Esta API apenas recolhe informação da base de dados e envia para o *browser* do utilizador sob o formato de Json.

Na linha 4 é realizada uma verificação da resposta recebida. Na eventualidade de ser bem sucedida, ou seja, no caso de ser recebido o código 200, procede-se ao processamento da informação na linha seguinte, começando-se por processar o resultado recebido num objeto Json utilizado pelo resto do código em JavaScript, seguindo-se da execução de uma função que utiliza os dados recebidos.

```
1 function httpGetAsync(theUrl) {
2     var xmlHttp = new XMLHttpRequest();
3     xmlHttp.onreadystatechange = function() {
4         if (xmlHttp.readyState == 4 && xmlHttp.status == 200)
5             func_handler(JSON.parse(xmlHttp.responseText))
6     }
7     xmlHttp.open("GET", theUrl, true);
8     xmlHttp.send(null);
9 }
```

Lista 6.12: Recolha de informação apresentada nos gráficos

## 6.7 Conclusão do Capítulo 6

Neste capítulo foi apresentado o desenvolvimento do controlador central do sistema que faz a maioria da gestão. Iniciou-se por instalar e configurar diferentes *softwares* que permitissem disponibilizar diferentes serviços, seguindo-se da explicação da aplicação de gestão desenvolvida.

## Capítulo 7

---

# Testes e Resultados

---

*Neste capítulo apresentam-se alguns testes de maior importância assim como a interpretação dos resultados obtidos, pretendendo-se comprovar o bom comportamento referente à solução proposta.*

### 7.1 Interface Web

Nos tópicos em seguida, apresenta-se a aplicação desenvolvida que permite consultar a informação recolhida pelos controladores remotos, assim como efetuar algum controlo. Em seguida, é apresentado o resultado da configuração do Apache, tal como outros *softwares* dedicados ao fornecimento de serviços web.

#### 7.1.1 Aplicação web desenvolvida

Na imagem 7.1 encontra-se a página principal da aplicação desenvolvida, sendo que é neste local onde é possível realizar a maior parte do controlo dos atuadores, assim como verificar, em tempo real, o comportamento do sistema.

Os caudalímetros de recolha de informação sobre os caudais que circulam no sistema encontram-se apresentados dentro das caixas violetas. Destaca-se a importância do sensor simbolizado pelo número 1, sendo este responsável por determinar o caudal da bomba de auxílio representada no interior da caixa branca. Este valor é importante pois é utilizado para verificar se existe a passagem de um caudal mínimo de manutenção na eletrobomba, de forma a proteger o dispositivo.

Nas caixas a azul, simbolizadas com o número 2, encontram-se representadas as eletroválvulas do sistema de rega principal. Note-se que no momento da captura, estes dispositivos estavam ligados encontrando-se com o corpo representado

a verde. Esta atuação é comprovada uma vez que verifica um caudal de cerca de 70 litros por minuto a circular no sistema.

Nas caixas a vermelho encontram-se representadas as eletroválvulas motorizadas desenvolvidas, sendo que junto ao corpo de cada dispositivo, encontra-se uma caixa que representa o ângulo de abertura de cada atuador. O servidor reconhece e responde à atuação destes dispositivos no entanto, as válvulas ainda não se encontram implementadas no sistema.

Por fim, na caixa verde, encontra-se uma representação do sensor de deteção de sobrecargas do reservatório inferior, assim como a eletroválvula utilizada para realizar testes ao dispositivo.

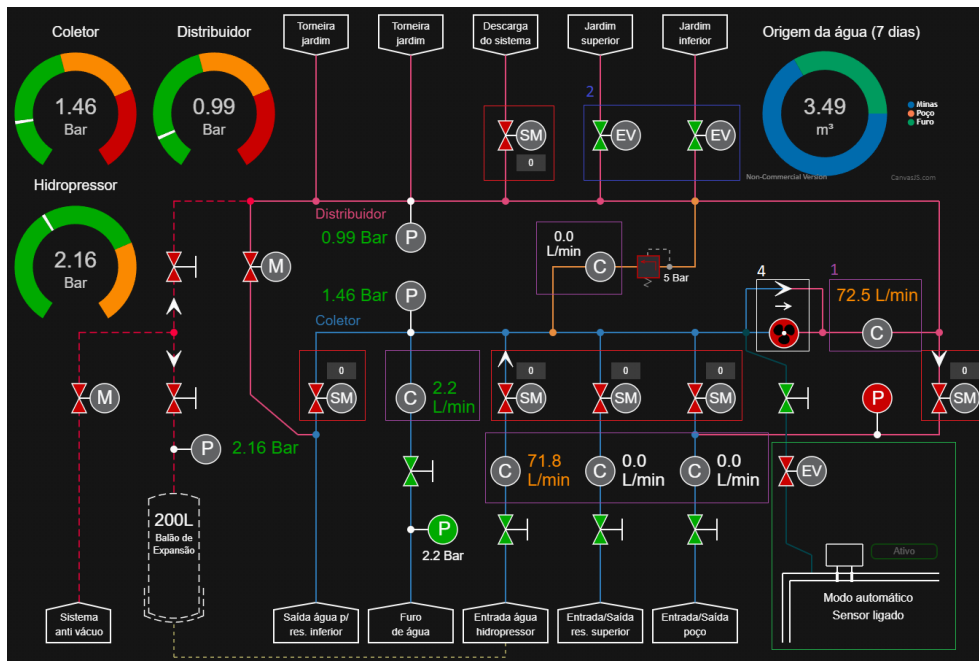


Figura 7.1: Página principal da aplicação web

### Informação sobre o controlador remoto da eletrobomba

Na imagem 7.2 encontra-se apresentada alguma informação relativa ao controlador da eletrobomba. No campo denominado como “LUA” verifica-se a idade da última atualização do controlador, indicando se a informação disponível está atualizada. Na linha seguinte, apresenta-se o tempo de duração da última operação da eletrobomba, o número total de arranques do dispositivo e o número total de horas de funcionamento.

Observando os indicadores coloridos, repara-se inicialmente que o controlador está no modo de funcionamento automático, isto é, a atuação da bomba depende de comandos enviados pelo servidor. O outro modo possível está representado

pela cor cinza e com o texto “Físico”, o que significa que o controlo da bomba tem de ser realizado localmente.

Em seguida, localiza-se um botão que permite ao utilizador pedir o arranque ou paragem da bomba. Note-se que este processo só pode ser realizado com o sistema em modo manual e com o controlador remoto em modo automático.

As últimas quatro linhas apresentam indicadores de anomalias sendo que a única que merece notoriedade é o estado do controlador de potência que indica se o quadro de controlo da eletrobomba está no modo operacional, ou se o relé de nível, ou o térmico de proteção, dispararam.

Controlador remoto da eletrobomba do furo		LUA: 0 s
Tempo: 00:05:51	Narr.: 601	Tempo total: 12 h
Modo de funcionamento do controlador	Automático	
Pedir arranque / paragem da eletrobomba	Arrancar	
Estado da eletrobomba	Parada	
Estado do controlador de potência	Ok	
Limite máximo de arranques por minuto (5)	Ok	
Limite máximo de arranques por hora (20)	Ok	
Proteção contra flutuações da corrente da eletrobomba	Ok	

Figura 7.2: Informação sobre o controlador remoto da eletrobomba

### Informação sobre o controlador remoto do sistema de rega auxiliar

Na imagem 7.3 apresenta-se um bloco com informação referente ao controlador remoto do sistema de rega auxiliar.

A apresentação das anomalias e da informação é semelhante ao controlador remoto da eletrobomba, pelo que apenas se dará relevância a duas exceções. Primeiramente, note-se que a alteração do modo de funcionamento do controlador requer a introdução de um código de confirmação de quatro dígitos gerado aleatoriamente. A reação do apertado do botão “submiter” apenas é desencadeada caso se verifique a correta introdução do código apresentado, sendo que a rotina executada envia uma indicação para o servidor e altera o valor do número de confirmação apresentado. Este sistema foi implementado para prevenir apertos acidentais do botão.

Por fim, repare-se que o último conjunto de informação apresentada é o estado das eletroválvulas, determinado pelo algoritmo implementado no controlador remoto e o estado dos atuadores requerido pelo utilizador, em modo manual. O fundo verde indica o estado do controlador. Uma vez que se verifica que este

módulo esta a operar em modo automático, nas saídas espera-se que apareçam os estados representados no primeiro conjunto de indicadores.

O estado das válvulas em modo manual pode ser alterado pressionando o atuador desejado na fila de indicadores inferior.

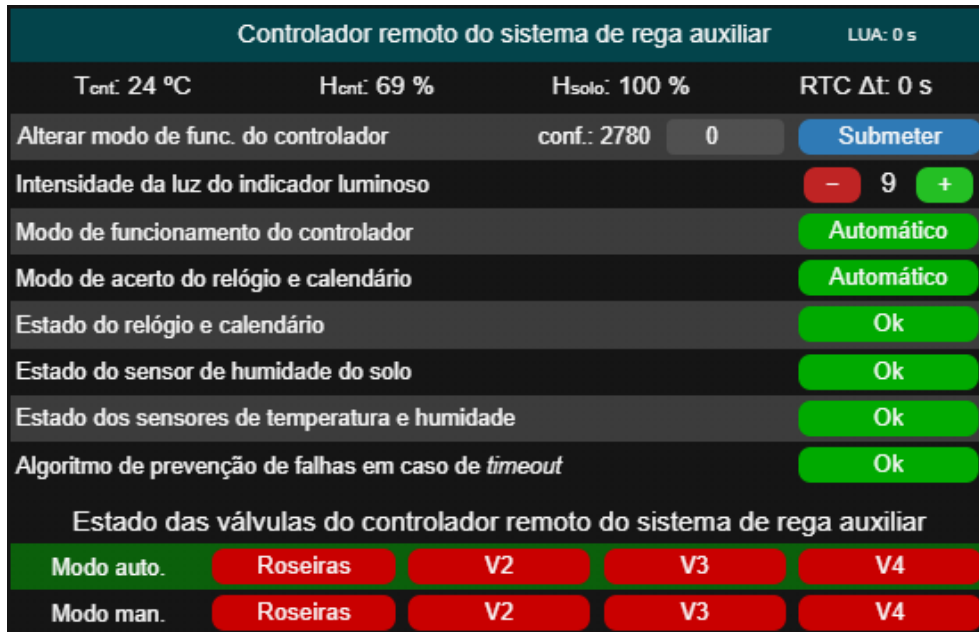


Figura 7.3: Informação sobre o controlador remoto do sistema de rega auxiliar

### Informação sobre o controlador remoto de recolha de informação meteorológica

Na imagem 7.4 encontra-se apresentada informação sobre o controlador remoto de recolha de informação meteorológica, sendo que este dispositivo apresenta um conjunto de informações menos diversificado que os controladores explicados anteriormente. O único controlo que é possível realizar nesta secção é a determinação do modo de funcionamento do algoritmo de atuação da resistência de remoção de condensação, podendo variar entre o modo manual e automático. Também é possível determinar o estado de atuação deste dispositivo caso o algoritmo de controlo esteja configurado como modo manual.

Note-se que o sistema reconhece o pedido de controlo da resistência de forma automática no entanto, a rotina executada nesta situação não desempenha nenhum controlo.

Sistema de recolha de informação meteorologia				LUA: 0 s
T <sub>atm</sub> : 18.1 °C	H <sub>atm</sub> : 93.0 %	T <sub>ent</sub> : 33.7 °C	H <sub>ent</sub> : 10.0 %	
P <sub>atm</sub> : 981.77 hPa	1 UVi	Lux: 6716.47	Prec.: 0.6 mm	
Modo de controlo da remoção da condensação			Manual	
Estado da resistência de remoção da condensação			Desligada	
Intensidade da luz do indicador luminoso			- 5 +	

Figura 7.4: Informação sobre o controlador remoto de recolha de informação meteorológica

### Apresentação dos caudais das fontes, consumos e balanço

Na figura 7.5 apresenta-se uma imagem que ilustra os resultados das regressões lineares de determinação dos caudais das fontes que alimentam a habitação. Em seguida, encontra-se o valor médio do consumo diário de água, sendo que se o valor for pressionado, será apresentado o consumo médio de água utilizado das minas e do furo.

Caso o valor do consumo médio de uma fonte for superior ao seu valor do caudal determinado, a caixa onde o resultado da regressão é apresentado altera a cor de fundo de verde para vermelho. Este ato indica que a prática de consumo da água fornecida pela nascente não é sustentável.

O último valor apresentado corresponde ao balanço do volume de água que apresenta a diferença entre o volume de água consumida e o volume total de água disponível. Este valor é de elevada importância pelo que as práticas diárias do uso de água são condicionadas pelo resultado do seu cálculo. Note-se que a apresentação de um valor negativo traduz-se na perda de volume de reserva, o que é, claramente, insustentável, pelo que se o seu valor for inferior a 0 por um alargado período de tempo, corre-se o risco de ficar sem água disponível nos reservatórios.

Caudal de água proveniente das minas	343 L/dia
Caudal de água proveniente do furo	179 L/dia
Média do volume de água gasta diariamente (Geral)	481 L
Balanço do volume de água	+41 L/dia

Figura 7.5: Apresentação dos caudais das fontes, consumos e balanço

Na imagem 7.6 está apresentado, graficamente, o volume de água gasto nos últimos 7 dias, sendo que o gráfico apresentado em (a) destaca o volume total utilizado por nascente. Pressionando-se a porção relativa a cada fonte, obtém-se um gráfico de colunas com informação mais detalhada, apresentando-se o consumo de água diário relativo à nascente escolhida.

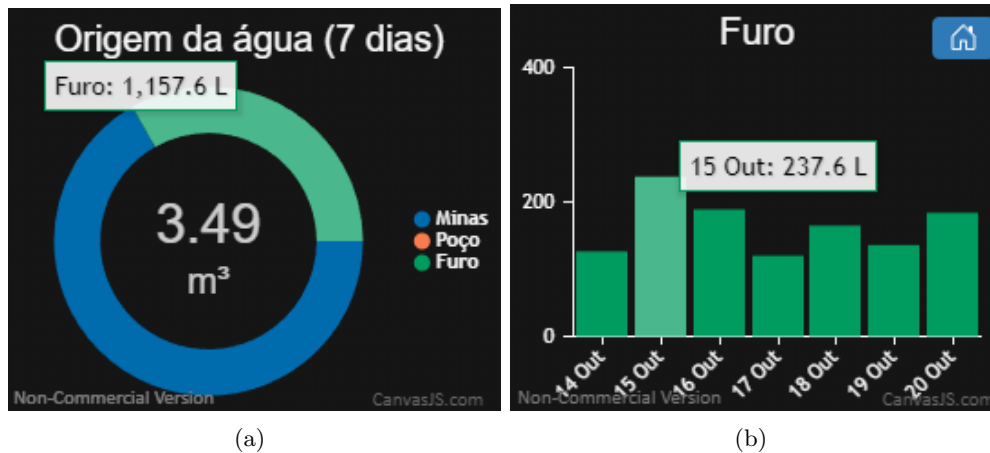


Figura 7.6: Origem do volume de água consumida nos últimos 7 dias (a) De forma geral (b) Apresentação diária por fonte

### Representação gráfica do cálculo dos caudais das minas e do furo

Na imagem 7.7 apresentam-se os gráficos retornados pelos algoritmos que determinam o caudal de água proveniente das diferentes fontes que abastecem a residência. Note-se que o cálculo efetuado utiliza valores dos últimos 7 dias, sendo que o processamento dos dados apresentados foi realizado no dia 18 de outubro de 2022, retornando um caudal das minas de 320,4 litros por dia e 177,0 litros por dia para o furo.

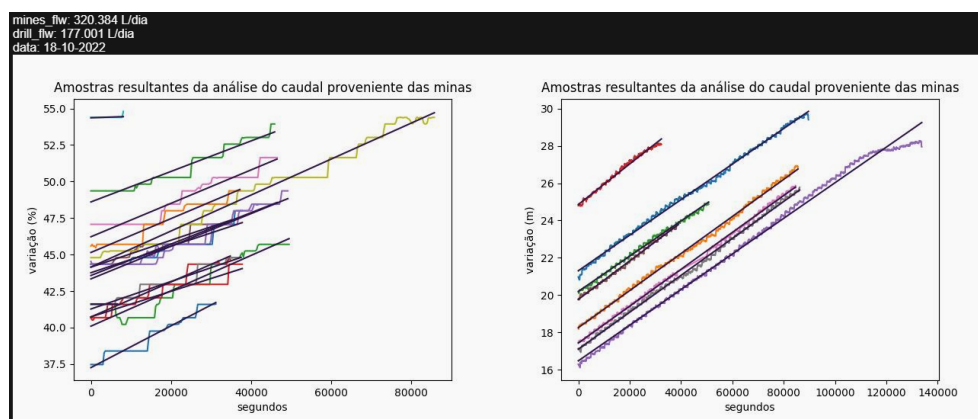


Figura 7.7: Gráficos retornados pelos algoritmos que calculam os caudais das fontes de água

### Análise de gráficos de histórico

Na imagem 7.8 apresenta-se uma captura de um gráfico que disponibiliza informação relativamente ao volume de água gasto diariamente, visualizado através das colunas verdes. Também é possível consultar o histórico diário do balanço de água através da representação em cascata.

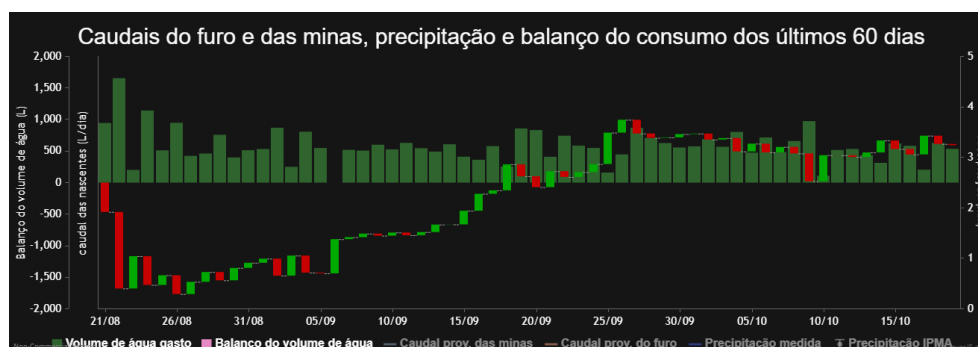


Figura 7.8: Gráficos do volume de água gasta e balanço diário

As imagens em 7.9 refletem a influência das práticas diárias de consumo sobre o cálculo do balanço e, conseqüentemente, do volume em reserva.

Analisando a captura (a) nota-se que nos primeiros 4 dias houve um balanço positivo de elevada amplitude, pelo que se reflete no gráfico em (b) um aumento acentuado do seu nível. Em seguida, houve uma alteração nas políticas de poupança dos recursos hídricos que levaram a um decréscimo do balanço diário. Uma vez que este valor foi maioritariamente inferior a 0 durante um longo período de tempo, verificou-se uma diminuição do nível do reservatório.

Como contra medida, os hábitos de consumo tiveram de ser ajustados para

assegurar a sustentabilidade do sistema, verificando-se, em seguida, um ligeiro aumento do volume de água em reserva.



Figura 7.9: Gráficos com o histórico (a) Do balanço do volume de água (b) Da variação do nível do reservatório superior

As decisões de poupança de água incidiram sobretudo na diminuição dos períodos de rega dos espaços verdes da habitação. Tal ação verifica-se na imagem 7.10 em que a linha vermelha, representante do volume de água gasto para rega, foi diminuindo de tamanho ao longo do tempo. No dia 11 de agosto, o sistema de rega foi desligado pois uma nova redução do tempo de operação implicava que houvesse uma paragem por completo do sistema por não cumprir o requisito mínimo de operação de 45 segundos. Note-se que até ao dia 26 de setembro não havia nenhuma operação do sistema de rega devido ao mesmo problema.

A linha verde representa o volume em reserva do reservatório superior, note-se o seu decréscimo constante com a ativação dos programas de rega.

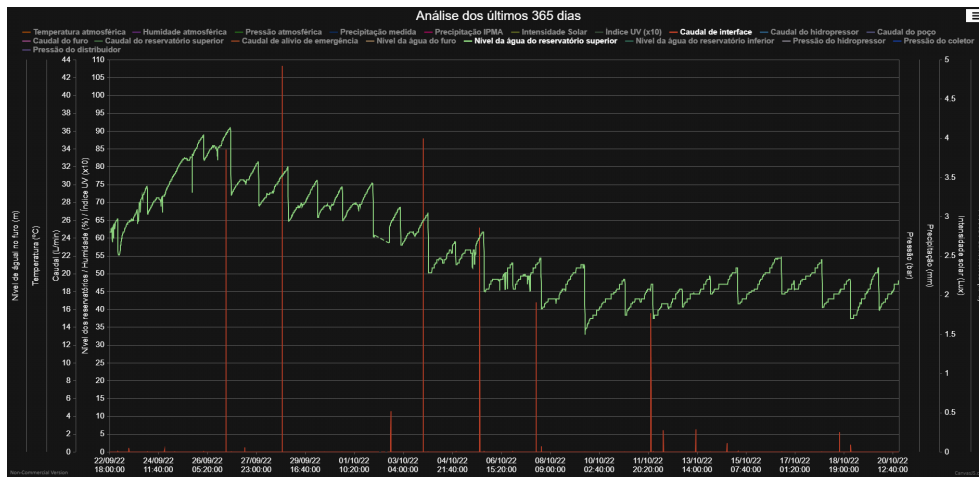


Figura 7.10: Volume de água gasto para rega e nível de água em reserva

Na imagem 7.11 constata-se os valores determinados do caudal das minas e do furo da residência, pelo que também está apresentado o valor da precipitação medida. Observando o gráfico verifica-se uma diminuição gradual dos caudais ao longo do tempo, no entanto, observa-se um aumento gradual do caudal das minas com início no mês de novembro. Este fenómeno explica-se devido à elevada precipitação poucos dias antes.

Devido à falta de valores, ainda não é possível tentar realizar-se uma previsão sobre o caudal das minas em relação ao valor da precipitação medido no entanto, nota-se uma possível tendência.

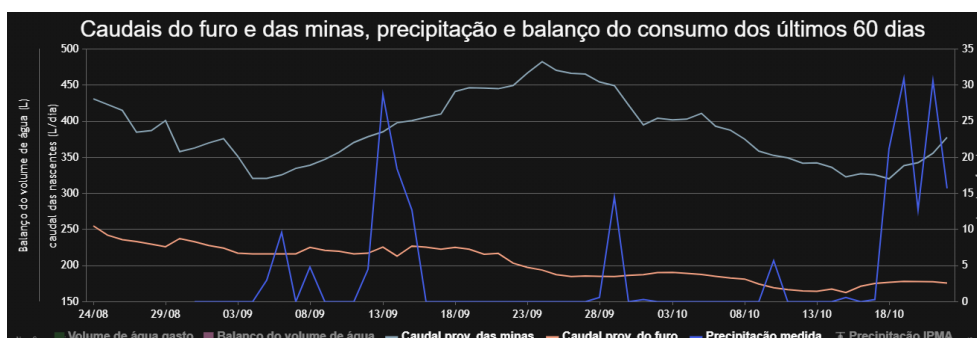


Figura 7.11: Caudais das minas e do furo e precipitação medida

A imagem 7.12 demonstra os valores da precipitação recolhidos pelo sensor implementado, assim como os valores determinados pelo IPMA para o concelho onde se situa a residência. Os dados fornecidos pelo instituto português do mar e da atmosfera encontram-se apresentados sobre forma de um intervalo que contém a precipitação mínima e máxima determinada para a mesma região.

Existem alguns valores medidos que não encaixam no intervalo do IPMA, pelo que se pensa que a razão para tal imprevisto seja o facto das estações meteorológicas do instituto e o sensor implementado localizarem-se em sítios diferentes.



Figura 7.12: Precipitação medida pelo sensor implementado e precipitação determinada pelo IPMA

Por fim, na imagem 7.13 encontra-se um gráfico que relaciona o nível do furo (linha clara) com o nível do reservatório principal (linha turquês), sendo que estava a ser efetuada uma operação de enchimento com recurso a eletrobomba do furo.

A linha a rosa representa o caudal medido pelo caudalímetro da bomba do furo, valores estes que foram utilizados para determinar o volume total de água do furo gasta. Continua-se a verificar um aumento do nível do reservatório após a paragem da bomba, no entanto, com um declive menos acentuado. Este fenómeno explica-se devido ao facto de estar a ocorrer uma transferência de água do reservatório superior para o reservatório inferior, apesar de esta operação não estar representada graficamente.

Verifica-se um pequeno salto periódico do nível do furo, devido a uma atuação temporizada no compressor utilizado para medir o nível de água do furo.

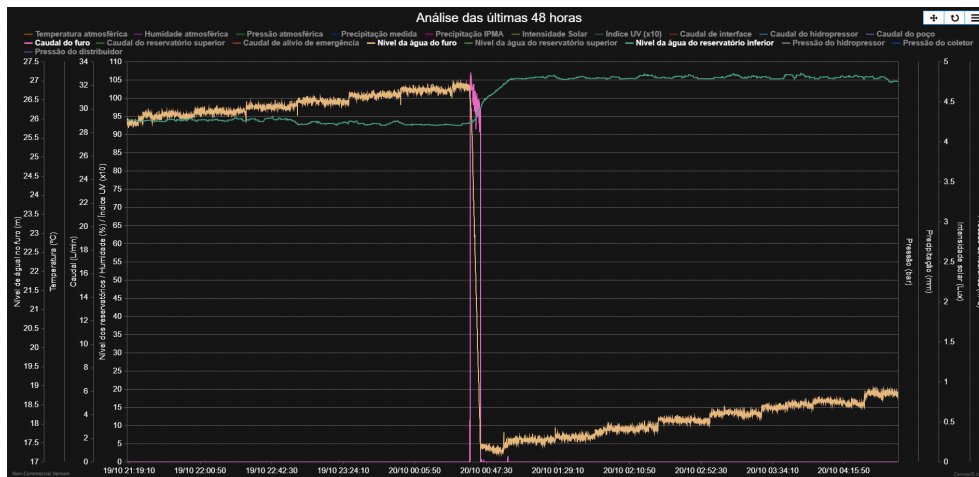


Figura 7.13: Variação do volume do reservatório principal e do furo

### Visualização dos clientes conectados ao servidor

Na imagem 7.14 está representado um esquema que apresenta todos os dispositivos ligados ao *broker* MQTT que comunicam com a aplicação de controlo desenvolvida.

É possível dividir a imagem em quatro grupos distintos explicados em seguida:

- Utilizadores - dispositivos do lado esquerdo, que indicam o número de utilizadores ligados, o aparelho que utilizaram para realizar a ligação e a latência da sua comunicação.

- API do IMPA e do Gmail - apenas são representados como constituintes do sistema, pelo que ainda não existe nenhum algoritmo de exposição de informação.
- API da Twilio - API utilizada para realizar avisos de elevada importância recorrendo a SMS. Na imagem apenas é apresentado o saldo restante da conta, este valor é atualizado automaticamente, no entanto, pode-se realizar uma atualização manual pressionando o símbolo laranja.
- Controladores remotos - dispositivos do lado direito, que informam sobre a idade da última atualização de dados, valor de elevada importância para determinar o correto funcionamento de um módulo.
- Servidor - representa a máquina central, indicando o número de comunicações realizadas até ao momento pela aplicação de controlo desenvolvida. Verifique-se que existe uma indicação do lado superior direito da imagem, avisando de novas entradas nos ficheiros de *logs* de erros, pedindo atenção do utilizador.

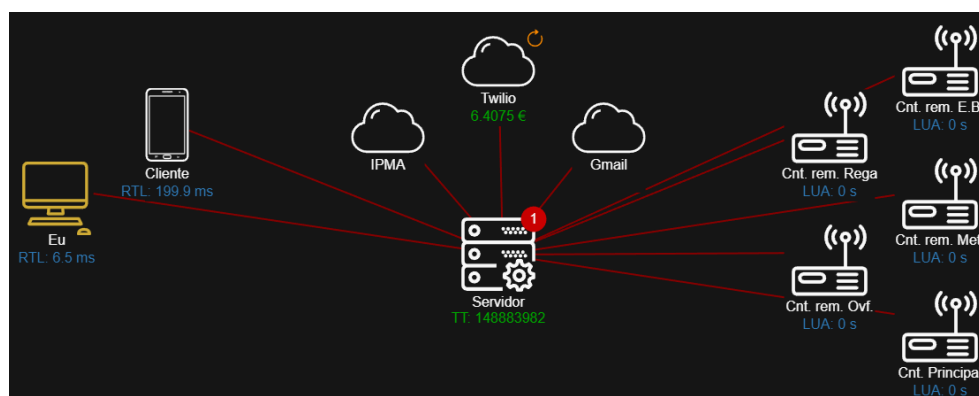








Figura 7.14: Visualização dos clientes conectados ao servidor

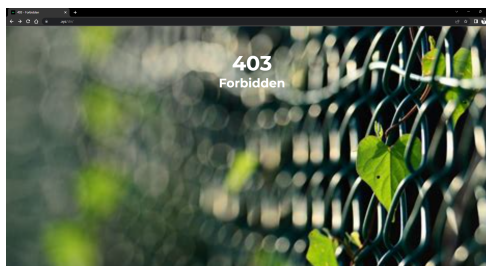
### 7.1.2 Configuração do Apache

Com o intuito de demonstrar o resultado da configuração de algumas regras do ficheiro de configuração do *htaccess*, apresentam-se as imagens em 7.15. Estas figuras representam a ativação (a) e desativação (b) da listagem do conteúdo disponível no servidor. A configuração (a) encontra-se definida por defeito, pelo que foi criada uma regra que impede a listagem de ficheiros e diretórios, sendo que, na tentativa de acesso de um diretório, é retornada a indicação da proibição de acesso, representada pelo erro 403 em conjunto com uma página ilustrativa (b).

## Index of /dir

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">dir_1/</a>	2022-09-30 18:42	-	
 <a href="#">dir_2/</a>	2022-09-30 18:42	-	
 <a href="#">nuno.html</a>	2022-09-30 18:41	0	
 <a href="#">sousa.php</a>	2022-09-30 18:41	0	
 <a href="#">teste.pdf</a>	2022-09-30 18:41	0	

(a)



(b)

Figura 7.15: Resposta do servidor (a) Com listagem ativada (b) Com listagem desativada

## 7.2 Notificação dos utilizadores

Nas imagens 7.16 apresentam-se as notificações enviadas aos utilizadores, sendo que na figura (a) apresenta-se um Email enviado sempre que termina uma semana. Esta notificação é de reduzida importância, apenas serve de relatório semanal informando os consumos diários e totais de água, divididos por nascente. é enviado também o valor da precipitação diária medido pelo sensor de precipitação.

Na imagem (b) apresentam-se as notificações enviadas por SMS, que requerem a atenção do utilizador de forma mais breve possível. Observando a figura verifica-se que se encontram três notificações diferente, a primeira apenas avisa que o serviço de escrita na base de dados parou, precisando que o utilizador resolva o problema e reinicie a aplicação.

Em seguida, indica-se que está a haver uma sobrecarga do reservatório inferior, evento despoletado pela deteção de água pelo sensor de sobrecarga.

Por fim, a última notificação é semelhante à anterior, houve uma deteção de água pelo sensor de sobrecarga no entanto, o fim de curso do sensor de nível não detetou água, havendo uma incongruência nos valores. Esta mensagem costuma ser enviada quando a válvula de teste é atuada em modo manual, havendo uma deteção de água sem que haja sobrecarga no reservatório.

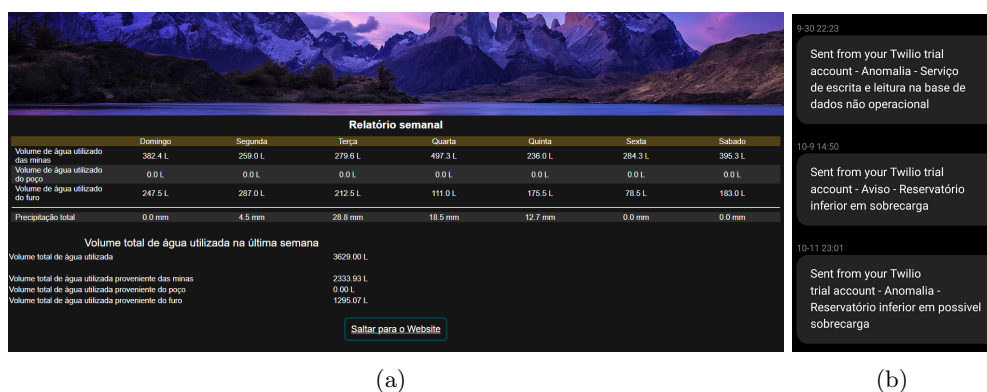


Figura 7.16: Notificação dos utilizadores (a) Por Email (b) Por SMS

## 7.3 Controladores Remotos

Nos tópicos em seguida, apresentam-se os resultados obtidos resultantes das implementações dos controladores remotos. Note-se que apenas são referidos os dispositivos que permitem a consulta de resultados localmente, pelo que não dependem da aplicação web para disponibilizarem os dados.

### 7.3.1 Controlador Remoto da Eletrobomba

Na figura 7.17 constata-se o menu principal disponível no LCD do controlador da eletrobomba. Observando a imagem, é possível extrair alguma informação sobre o dispositivo. Primeiramente, repare-se que o controlador encontra-se a operar em modo automático, isto é, atua sobre o circuito de potência consoante os comandos enviados pelo servidor.

Com o circuito de condicionamento de sinais analógicos, tornou-se possível conhecer os valores da tensão e corrente que circulam no circuito de potência. Note-se que no momento da captura da imagem, a bomba encontrava-se parada, uma vez que os valores da corrente e da potência eram residuais. Os valores não são exatamente 0 pois o sensor de corrente mede a intensidade à entrada do quadro e não somente da bomba, portanto, os valores apresentados correspondem ao consumo da fonte de alimentação que alimenta o circuito implementado.

No presente menu do LCD é possível extrair mais 3 variáveis. Primeiramente, o tempo de duração da última operação da bomba foi de 46 segundos, em seguida, conclui-se que o furo possui 22.6 metros de água no seu interior. Por fim, determina-se que o controlador remoto apresenta uma boa comunicação com o servidor, uma vez que está no modo “online”.

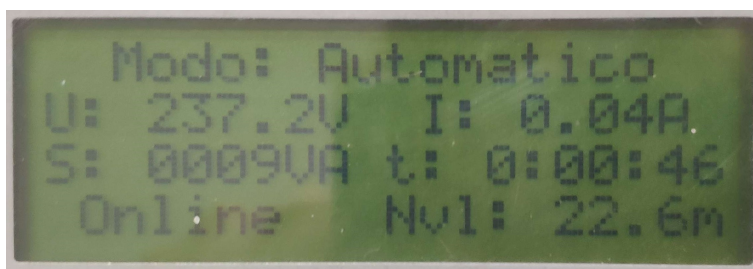


Figura 7.17: Menu principal do controlador remoto da eletrobomba

Pressionando-se de forma longa o botão de *stop*, acede-se a menus secundários, apresentados em 7.18, que permitem realizar a configuração dos limites da corrente medida à entrada do quadro quando a bomba estiver ligada e do dispositivo de comunicação Wi-fi.

Neste local, também é permitido realizar a consulta de informação relativa ao número de arranques da máquina, assim como o número total de horas de funcionamento.

A nova configuração dos valores dos limites da corrente só são armazenados em memória se for realizada uma navegação até à opção “Sair e guardar”, seguindo-se da sua seleção. Se a opção “Sair sem guardar” for seleccionada, ou se houver um tempo de inatividade superior a 45 segundos, todos os novos valores inseridos são descartados.

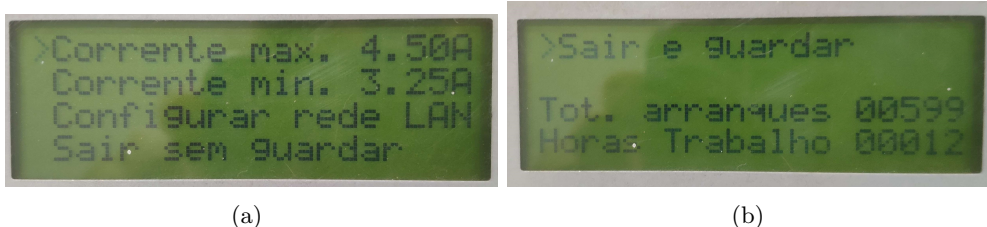


Figura 7.18: Menus do controlador remoto da eletrobomba (a) Primeira página (b) Segunda página

### 7.3.2 Controlador Remoto do Sistema de Rega Auxiliar

Na figura 7.19 encontra-se uma imagem do menu principal do LCD do controlador remoto do sistema de rega auxiliar.

Analisando a imagem, são facilmente extraíveis várias informações. Primeiramente, repara-se na linha 1 do LCD, onde é possível verificar o modo de funcionamento do sistema, estando este a operar em modo automático no momento da captura da imagem.

Já na segunda e terceira linha do *display* está-se apresentada a informação fornecida pelo relógio do controlador, assim como o estado da ligação do módulo com o servidor. Visto ler-se a indicação “online”, percebe-se que existe uma boa comunicação entre o servidor e o controlador.

Na última linha recolhe-se uma grande variedade de dados. Primeiramente, consegue-se ler os valores da temperatura e da humidade da atmosfera no interior da caixa. Em seguida, verifica-se os estado de funcionamento das quatro válvulas, sendo que o valor “0000” indica que estão todas fechadas. Por fim, apresenta-se o valor da frequência lida do sensor de humidade do solo. Note-se que a conversão de frequência para uma escala adequada é realizada no lado do servidor, pelo que a razão da apresentação deste dado no *display* justifica-se unicamente pelo aumento na facilidade da realização do *debug* do controlador.



Figura 7.19: Menu principal do controlador remoto do sistema de rega auxiliar

Na figura 7.20 apresentam-se dois menus utilizados para realizar a configuração de diversos parâmetros do controlador. Na imagem (a), verifica-se o local onde é possível realizar um acerto do relógio do controlador localmente, sendo que também é possível definir a permissão ou o bloqueio da função que permite efetuar o acerto do relógio remotamente.

Na imagem (b) apresenta-se o último menu de configuração de variáveis do sistema. Na linha 1 verifica-se a localização da rotina que ordena o lançamento da página de configuração das credenciais do controlador. Já na linha 2, está exibida a opção que permite alterar o modo de funcionamento do sistema.

Por fim, nas restantes linhas, encontram-se os estados de atuação das válvulas caso o sistema esteja a operar em modo manual. Se alguma válvula estiver selecionada, aparece o símbolo “>” antes de cada opção. Desta forma, conclui-se que estão todas desligadas.



Figura 7.20: Menus do controlador do sistema de rega auxiliar (a) Configuração do relógio (b) Configuração do modo de operação

A imagem 7.21 apresenta o último menu implementado neste controlador. Neste local é possível realizar a configuração dos cinco programas de rega. Observando a imagem, percebe-se, através da leitura da informação da linha 1, que encontra-se apresentada a informação referente ao primeiro programa.

Na linha 3, verifica-se que o programa foi configurado para operar apenas aos domingos, quartas e sextas porém, a sua execução encontra-se desativada.

Saltando para a linha 4, encontra-se inicialmente o horário de disparo do programa, seguindo-se as opções que determinam as válvulas que devem de ser operadas. Uma válvula é ativada se o registo que a representa estiver apresentado com um “A” no *display*. Na eventualidade de não se pretender perturbar o seu estado de repouso, configura-se o registo com um “n”.

Em seguida, na mesma linha, configura-se a dependência do valor de atuação do programa pela humidade do solo e pela probabilidade de precipitação. Ambas as opções encontram-se desativadas sendo que aparecerem caracteres minúsculos em cada opção. Por fim, apresenta-se a opção que permite configurar o tempo de execução do programa.

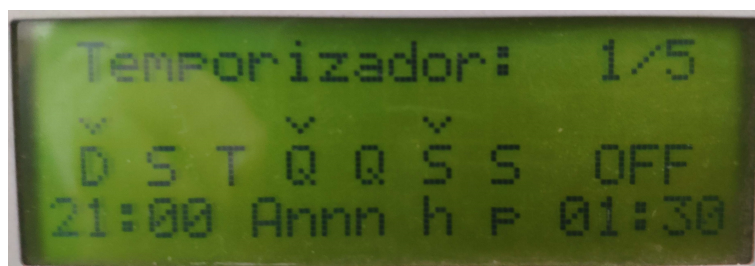


Figura 7.21: Configuração dos programas de rega do controlador remoto

### 7.3.3 Interface de Configuração dos Controladores Remotos

A interface de configuração comum entre os diferentes controladores está apresentada na figura 7.22 e possibilita ao utilizador atualizar as credenciais guardadas na memória de cada dispositivo.

É importante lembrar que esta interface é disponibilizada apenas quando for recebida uma ordem enviada por parte do utilizador. Após o comando ser recebido, a configuração de *hardware* é alterada do modo estação para o modo ponto de acesso, disponibilizando uma rede à qual o utilizador se pode ligar.

Após ser efetuada uma ligação à rede criada, através do acesso ao endereço dos dispositivos, sendo este 192.168.4.1, passa a ser possível aceder à página web.

Nesta interface é possível definir-se as credenciais da rede de LAN a qual se pretende que o dispositivo realize uma conexão para comunicar com o servidor. A informação relativa ao *broker* MQTT encontra-se apresentada no segundo conjunto de campos. Neste espaço, é necessário definir-se a localização do *broker* na rede, definido-se, também, o nome do utilizador e a palavra-passe que o cliente usará para realizar uma ligação.

Após pressionar o botão de confirmar, a informação é enviada para o ESP8266 através de um pedido POST, sendo que o microcontrolador guarda as credenciais em memória, reiniciando-se em seguida.



The image shows a mobile application interface for configuring an ESP8266. The screen has a dark theme. At the top, there is a status bar with 'No internet connection' and a signal strength icon. Below that, a browser-like header shows a warning icon, the IP address '192.168.4.1', and a lock icon. The main heading is 'Configuração' in large white text. Underneath, there are two sections: 'Wi-Fi' (highlighted with a yellow border) and 'MQTT' (highlighted with a blue border). The 'Wi-Fi' section contains two text input fields labeled 'SSID' and 'Password'. The 'MQTT' section contains three text input fields labeled 'Endereço do servidor', 'Username', and 'Password'. At the bottom of the screen, there is a prominent green button labeled 'Confirmar'.

Figura 7.22: Interface de configuração dos ESP8266

## 7.4 Conclusão do Capítulo 7

Neste capítulo apresentaram-se alguns testes e resultados, assim como uma explicação dos valores obtidos referentes à solução implementada no capítulo anterior.

Deste modo, apresentou-se a interface gráfica do utilizador desenvolvida, que permite realizar a consulta de diversos valores, assim como o controlo de diversos dispositivos. Foi também demonstrada a visualização e possível controlo dos módulos que permitem realizar uma interação local

## Capítulo 8

---

# Conclusões

---

O trabalho desenvolvido apresenta um sistema capaz de realizar a recolha e o tratamento de dados de uma residência com o intuito de analisar as práticas do consumo dos recursos hídricos, assim como a sua sustentabilidade, permite também armazenar dados de diversos sensores possibilitando uma grande oportunidade de desenvolvimento no futuro.

Para se conseguir atingir os objetivos, implementou-se uma variedade de módulos IoT ligados por rede a um servidor, sendo que são responsáveis por recolher os dados, assim como realizar algum controlo.

Implementou-se um controlo autónomo dos sistemas de rega mas, lamentavelmente não foram implementados alguns atuadores críticos para o funcionamento autónomo dos ciclos de carga dos reservatórios. Isto levou a que haja uma continuação da necessidade do sistema ser operado de forma manual, no entanto, esta operação tornou-se muito mais prática, sem que necessite de um utilizador constantemente no local, visto poder consultar o estado do sistema de forma remota. Os desperdícios avultados de recursos hídricos também foram mitigados devido ao aviso do utilizador por mensagem.

Em relação à recolha e tratamento dos dados, considera-se que o sistema desenvolvido é uma ótima ferramenta de apoio na decisão e condicionamento das políticas de poupança de água, sendo que os dados fornecidos são de grande utilidade na verificação da sustentabilidade das práticas diárias. Esta informação é ainda mais importante em Portugal uma vez atravessa-se um período de seca considerável.

Em relação ao consumo dos recursos hídricos na residência, o valor mais recente determinado pelo sistema foi de 481 litros por dia. Sendo que o agregado familiar consiste em 6 elementos, calcula-se que há um consumo médio de cerca

de 80 litros por pessoa por dia, valor bastante abaixo do consumo médio de um português e, inferior ao consumo médio europeu.

Com a análise efetuada, conclui-se que, até ao momento, as políticas de consumo dos recursos hídricos são sustentáveis, no entanto já houve necessidade de aplicar múltiplas ações que visaram reduzir o volume de água consumido diariamente para combater a diminuição do caudal de água proveniente das nascentes.

Com isto, conclui-se que o sistema de abastecimento de água da residência não foi capaz de alimentar, em simultâneo, os sistemas de rega dos espaços verdes e os consumos relacionados com a atividade do quotidiano, pelo que devem ser consideradas alternativas para resolver o problema.

## 8.1 Trabalho Futuro

Como trabalho futuro, reconhece-se principalmente a necessidade de implementar as válvulas motorizadas no sistema hidráulico, permitindo um controlo totalmente autónomo.

Para além disso, sendo que a atual implementação do controlador remoto principal cumpre as necessidades para o qual foi projetado, deve-se abandonar o protótipo de teste atual e desenvolver uma solução final que substitua o controlador em causa. Esta solução deve conter sistemas de segurança tais como uma fonte de alimentação alternativa e uma bateria de modo a realizar-se o fecho das válvulas motorizadas em segurança numa eventual falha de luz.

A recolha de dados como a precipitação, a temperatura e humidade atmosférica e a humidade do solo permitem que eventualmente seja implementado um algoritmo que determine o volume de água necessário para realizar a rega dos espaços verdes, sem que haja a necessidade de desperdiçar água provocado pelo prolongado e desnecessário tempo de atuação.

Em relação especificamente à precipitação, nota-se uma dependência direta entre este valor medido e o caudal das minas, mas a insuficiência de valores ainda não permite realizar previsões.

---

## Referências Bibliográficas

---

- [1] J. D. Guillot, “Água potável na ue: melhor qualidade e acesso,” p. 3, 12 2020. [cited on p. v, 1]
- [2] U. Nations, *World Urbanization Prospects: The 2014 Revision*. Department of Economic and Social Affairs, Population Division, 2014. [cited on p. v, 5]
- [3] LoRa Alliance, “What is LoRaWAN.” [https://lora-alliance.org/resource\\_hub/what-is-lorawan/](https://lora-alliance.org/resource_hub/what-is-lorawan/), Último Acesso: 18 março, 2022. [cited on p. v, 9, 10]
- [4] <https://www.thethingsnetwork.org/community/porto/>, Último Acesso: 18 março, 2022. [cited on p. v, 10]
- [5] <https://www.thethingsnetwork.org/community/lisbon/>, Último Acesso: 18 março, 2022. [cited on p. v, 10]
- [6] Sigfox, “What is Sigfox?.” <https://www.sigfox.com/en/sigfox-story>, Último Acesso: 18 março, 2022. [cited on p. v, 12, 13]
- [7] B. Perrin, “The art and science of rs-485,” *Circuit Cellar*, pp. 1–6, July 1999. [cited on p. v, 16, 17, 18]
- [8] L. Frenzel, *Handbook of Serial Communications Interfaces: A Comprehensive Compendium of Serial Digital Input/Output (I/O) Standards*. 225 Wyman Street, Waltham, MA 02451, USA: Newnes, 2016. [cited on p. v, 14, 15, 17, 18, 19, 20]
- [9] G. Korotcenkov, *Handbook of Humidity Measurement*. CRC Press, 2020. [cited on p. v, 23, 25]
- [10] <https://i.stack.imgur.com/28IoT.png>, Último Acesso: 27 março, 2022. [cited on p. v, 26]

- [11] Allegro MicroSystems, *ACS712 - Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor*, 1 2020. Rev. 19. [cited on p. v, ix, 26, 27, 33]
- [12] Texas Instruments, *Three-Phase AC Current Measurement Using Current Transformer Reference Design*, 4 2016. [cited on p. v, ix, 32, 33]
- [13] <https://www.weathershack.com/images/tipping-bucket-rain-gauge.gif>, Último Acesso: 9 outubro, 2022. [cited on p. v, 44]
- [14] <https://www.robotics.org.za/image/cache/wp/gj/generic/WH-SP-RG/WH-SP-RG-000-500x500.webp>, Último Acesso: 9 outubro, 2022. [cited on p. v, 44]
- [15] [https://pt.farnell.com/productimages/large/en\\_GB/2946124-40.jpg](https://pt.farnell.com/productimages/large/en_GB/2946124-40.jpg), Último Acesso: 9 outubro, 2022. [cited on p. v, 47]
- [16] <https://bt-electronic.com/wp-content/uploads/2022/03/ARDUINO-SENSOR-SOIL-MOISTURE-HD-38.jpg>, Último Acesso: 9 outubro, 2022. [cited on p. v, 47]
- [17] [https://ae01.alicdn.com/kf/H7d8d31ef19f74a8eb2c808a87f138c3bh/Transmissor-nivelado-submers-vel-qdy30a-do-sensor-4-20ma-0-10v-do-n-vel-l-quido.jpg\\_Q90.jpg\\_.webp](https://ae01.alicdn.com/kf/H7d8d31ef19f74a8eb2c808a87f138c3bh/Transmissor-nivelado-submers-vel-qdy30a-do-sensor-4-20ma-0-10v-do-n-vel-l-quido.jpg_Q90.jpg_.webp), Último Acesso: 9 outubro, 2022. [cited on p. v, 50]
- [18] <https://pt.farnell.com/>, Último Acesso: 8 outubro, 2022. [cited on p. ix, 34, 35, 36, 39, 41, 42]
- [19] <https://eu.mouser.com/>, Último Acesso: 8 outubro, 2022. [cited on p. ix, 34, 35, 36, 39, 41, 42]
- [20] <https://www.digikey.pt/>, Último Acesso: 8 outubro, 2022. [cited on p. ix, 34, 35, 36, 39, 41, 42]
- [21] <https://lsc.com/>, Último Acesso: 8 outubro, 2022. [cited on p. ix, 34, 35, 36, 39, 41, 42]
- [22] Microchip Technology Inc., *8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*, 1 2015. Rev.: 7810D-AVR-01/15. [cited on p. ix, 34]
- [23] Microchip Technology Inc., *ATtiny3224/3226/3227*, 9 2021. Rev.B. [cited on p. ix, 34]
- [24] Espressif Systems, *ESP32 WROOM 32 Datasheet*, 8 2016. Version 3.3. [cited on p. ix, 35]

- [25] ST, *Arm Cortex-M4 32-bit MCU+FPU, 170 MHz /213 DMIPS, up to 128 KB Flash, 32 KB SRAM, rich analog, math accelerator*, 10 2021. Rev 6. [cited on p. ix, 35]
- [26] HOPERF Electronic, *RFM95/96/97/98(W) - Low Power Long Range Transceiver Module*, 2006. V1.0. [cited on p. ix, 36]
- [27] HC-Tech, *HC-12 Wireless RF UART Communication Module*, 12 2016. V2.4. [cited on p. ix, 36]
- [28] <https://ebay.com/>, Último Acesso: 8 outubro, 2022. [cited on p. ix, 36]
- [29] Atmel Corporation, *8-bit Microcontroller with 2/4/8K Bytes In-System Programmable Flash*, 12 05. Rev. 8006K-AVR-10/10. [cited on p. ix, 38, 39]
- [30] Espressif Systems, *ESP8266EX Datasheet*, 12 2015. Version 6.7. [cited on p. ix, 38, 39]
- [31] OSEPP Electronics, *DHT11 Humidity Temperature Sensor*. [cited on p. ix, 41]
- [32] Aosong Electronics, *Digital-output relative humidity temperature sensor/module DHT22*. [cited on p. ix, 41]
- [33] Texas Instruments, *HDC1080 Low Power, High Accuracy Digital Humidity Sensor with Temperature Sensor*, 11 2015. [cited on p. ix, 41]
- [34] ST, *Capacitive digital sensor for relative humidity and temperature*, 8 2016. Rev 4. [cited on p. ix, 41]
- [35] Bosh, *BMP280 Digital Pressure Sensor*, 5 2015. Rev.1.14. [cited on p. ix, 42]
- [36] Bosh, *BME280 Combined humidity and pressure sensor*, 7 2015. Rev.1.1. [cited on p. ix, 42]
- [37] ROHM Semiconductor, *Digital 16bit Serial Output Type Ambient Light Sensor IC*, 11 2011. Rev.D. [cited on p. ix, 43]
- [38] Roithner lasertechnik gmbh, *GUVA-S12SD*, 2 2011. [cited on p. ix, 43]
- [39] <https://www.farnell.com/datasheets/2700152.pdf>, Último Acesso: 9 outubro, 2022. [cited on p. ix, 47]
- [40] <https://www.farnell.com/datasheets/2700164.pdf>, Último Acesso: 9 outubro, 2022. [cited on p. ix, 47]
- [41] <https://www.farnell.com/datasheets/3216156.pdf>, Último Acesso: 9 outubro, 2022. [cited on p. ix, 47]

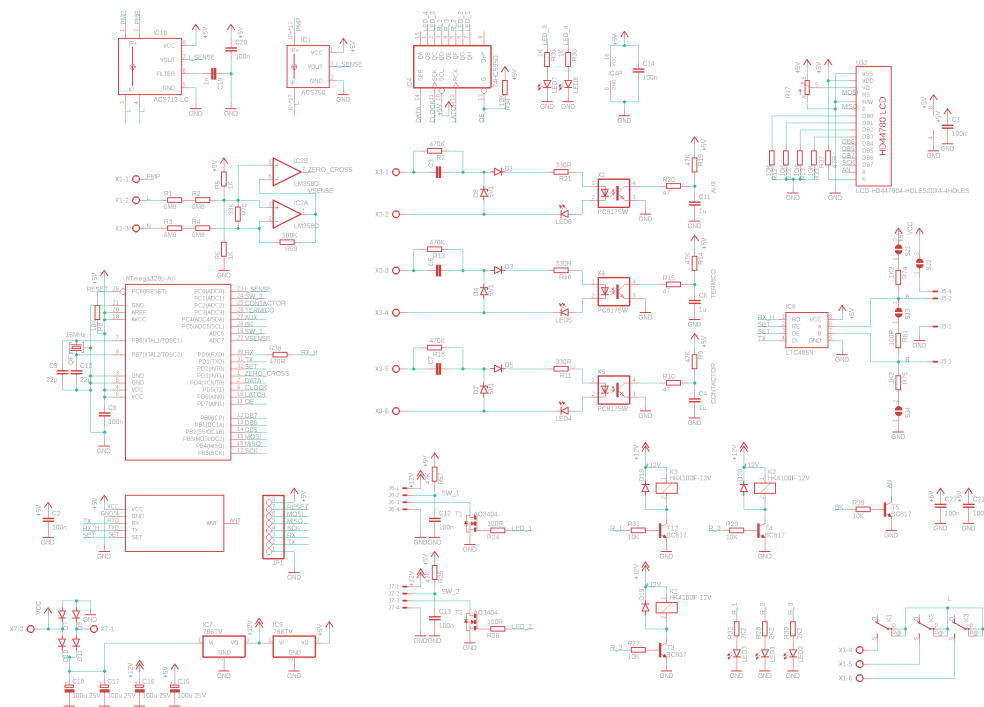
- [42] <https://pt.aliexpress.com/item/32861308733.html>, Último Acesso: 9 outubro, 2022. [cited on p. ix, 50]
- [43] Hunter Industries, *11/2" and 2" PGV*, 10 2022. [cited on p. x, 51]
- [44] <https://www.ebay.com/itm/232285662749>, Último Acesso: 9 outubro, 2022. [cited on p. x, 51]
- [45] <https://pt.aliexpress.com/item/4000767933598.html?spm=a2g0o.cart.0.0.43fb7f06T32W6n&mp=1&gatewayAdapt=glo2bra>, Último Acesso: 9 outubro, 2022. [cited on p. x, 51]
- [46] [https://pt.aliexpress.com/item/4000553188040.html?spm=a2g0o.order\\_list.0.0.5ef7caa4DfLdc1&gatewayAdapt=glo2bra](https://pt.aliexpress.com/item/4000553188040.html?spm=a2g0o.order_list.0.0.5ef7caa4DfLdc1&gatewayAdapt=glo2bra), Último Acesso: 9 outubro, 2022. [cited on p. x, 51]
- [47] N. Ozaki, *Smart Sensing for Traffic Monitoring*. London, United Kingdom: The Institution of Engineering and Technology, 2020. [cited on p. 5]
- [48] T. S. S. J. Houbing Song, Ravi Srinivasan, *Smart Cities: foundations, principles, and applications*. 111 River Street, Hoboken, NJ 07030, USA: John Wiley Sons, Inc, June 2017. [cited on p. 6, 8]
- [49] F. F. Andrea Vesco, *Handbook of Research on Social, Economic, and Environmental Sustainability in the Development of Smart Cities*. 701 E. Chocolate Avenue Hershey PA, USA: IGI Global, August 2013. [cited on p. 6]
- [50] A. A. Waleed Ejaz, *Internet of Things for Smart Cities Technologies, Big Data and Security*. New York, USA: Springer, October 2018. [cited on p. 6, 8]
- [51] G. R. Halegoua, *Smart Cities*. Cambridge, Massachusetts: The MIT Press, 2020. [cited on p. 6]
- [52] <https://amsterdamsmartcity.com>, Último Acesso: 16 março, 2022. [cited on p. 7]
- [53] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT Express*, vol. 5, no. 1, pp. 2–3, 2019. [cited on p. 9, 12, 13]
- [54] IBM, "Connecting all the things in the Internet of Things." <https://developer.ibm.com/articles/iot-lp101-connectivity-network-protocols/>, Último Acesso: 24 março, 2022. [cited on p. 10, 11]
- [55] P. Henry and H. Luo, "Wifi: what's next?," *IEEE Communications Magazine*, vol. 40, no. 12, pp. 66–69, 2002. [cited on p. 11]

- [56] I. Froiz-Míguez, T. M. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, “Design, implementation and practical evaluation of an iot home automation system for fog computing applications based on mqtt and zigbee-wifi sensor nodes,” *Sensors*, vol. 18, no. 8, 2018. [cited on p. 11]
- [57] H. A. A. Al-Kashoash and A. H. Kemp, “Comparison of 6lowpan and lpwan for the internet of things,” *Australian Journal of Electrical and Electronics Engineering*, vol. 13, no. 4, pp. 268–274, 2016. [cited on p. 11]
- [58] A. P. Anjali Kulkarni, “A review on zigbee, gsm and wsn based home security by using embedded controlled sensor network,” *International Journal of Embedded systems and Applications*, vol. 6, December 2016. [cited on p. 12, 13]
- [59] W. C. F. T. C.-L. L. Jiangfeng Cheng, Li Da Xu, “Industrial iot in 5g environment towards smart manufacturing,” *Journal of Industrial Information Integration*, pp. 6–8, 2018. [cited on p. 12]
- [60] RS Components, “11 Internet of Things (IoT) Protocols You Need to Know About.” <https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about>, Último Acesso: 19 outubro, 2021. [cited on p. 12]
- [61] Sigfox, “Sigfox: Technical Overview.” <https://www.avnet.com/wps/wcm/connect/onesite/03aebfe2-98f7-4c28-be5f-90638c898009/sigfox-technical-overview.pdf?MOD=AJPERES&CVID=magVa.N&CVID=magVa.N&CVID=magVa.N>, Último Acesso: 18 março, 2022. [cited on p. 12]
- [62] P. G. A. G. Marco Di Natale, Haibo Zeng, *Understanding and Using the Controller Area Network Communication Protocol*. New York, USA: Springer, December 2011. [cited on p. 15]
- [63] Texas Instruments, *AN-903 A Comparison of Differential Termination Techniques*, August 1993. [cited on p. 17]
- [64] Microchip Technology Inc., “Single-Edge Nibble Transmission (SENT) Module.” <https://ww1.microchip.com/downloads/en/DeviceDoc/70005145b.pdf>, Último Acesso: 21 março, 2022. [cited on p. 19]
- [65] K. Borgeest, *EMC and Functional Safety of Automotive Electronics*. London, United Kingdom: The Institution of Engineering and Technology, 2018. [cited on p. 19]
- [66] Microchip Technology Inc., “Section 58. Master Synchronous Serial Port (MSSP).” <https://ww1.microchip.com/downloads/en/DeviceDoc/30627a.pdf>, Último Acesso: 20 março, 2022. [cited on p. 19]

- [67] C. Wootton, “Inter-integrated circuit (i2c),” *Apress, Berkeley, CA*, pp. 321–323, November 2016. [cited on p. 20]
- [68] NXP Semiconductors, *P82B715 - I2C-bus extender*, November 2009. Rev. 08. [cited on p. 20]
- [69] Linear Technology, *LTC4311 - Low Voltage I2C/SMBus Accelerator*. [cited on p. 20]
- [70] G. K. McMillan, *Advanced Temperature Measurement and Control Second Edition*. International Society of Automation, 2011. [cited on p. 22]
- [71] H. J. Stolten, *The Use Of Thermistors In Precise Temperature Measurement Thermometric Determination Of Molecular Weights*. New York University, 1951. [cited on p. 22]
- [72] M. Nau, *Electrical Temperature Measurement: with thermocouples and resistance thermometers*. JUMO, August 2002. [cited on p. 22]
- [73] F. F. Gerard C. M. Meijer, Guijie Wang, “Temperature sensors and voltage references implemented in cmos technology,” *IEEE SENSORS JOURNAL*, vol. 1, p. 1, October 2001. [cited on p. 23]
- [74] D. Andrea, *Battery Management Systems for Large Lithium-Ion Battery Packs*. 685 Canton Street, Norwood, MA 02062: Artech House, September 2010. [cited on p. 25, 26]
- [75] F. J. Charles Platt, “Encyclopedia of electronic components,” vol. 3, pp. 199–203, April 2016. [cited on p. 25, 26]
- [76] Schaffner EMV AG, *IEC 950 CENELEC EN 60950*, 8 2002. [cited on p. 55]
- [77] <https://forum.digikey.com/t/adxl11005z-eval-measure-g-rms/9919/3>, Último Acesso: 4 outubro, 2022. [cited on p. 67]
- [78] <https://www.openssl.org/>, Último Acesso: 6 outubro, 2022. [cited on p. 104]
- [79] <https://letsencrypt.org/>, Último Acesso: 6 outubro, 2022. [cited on p. 105]
- [80] <https://certbot.eff.org/>, Último Acesso: 6 outubro, 2022. [cited on p. 105]
- [81] D. M. J. K. Richard Barnes, Jacob Hoffman-Andrews, *Automatic Certificate Management Environment (ACME)*. IETF, Março 2019. [cited on p. 105]

## Anexo A

# Esquema do controlador remoto da eletrobomba



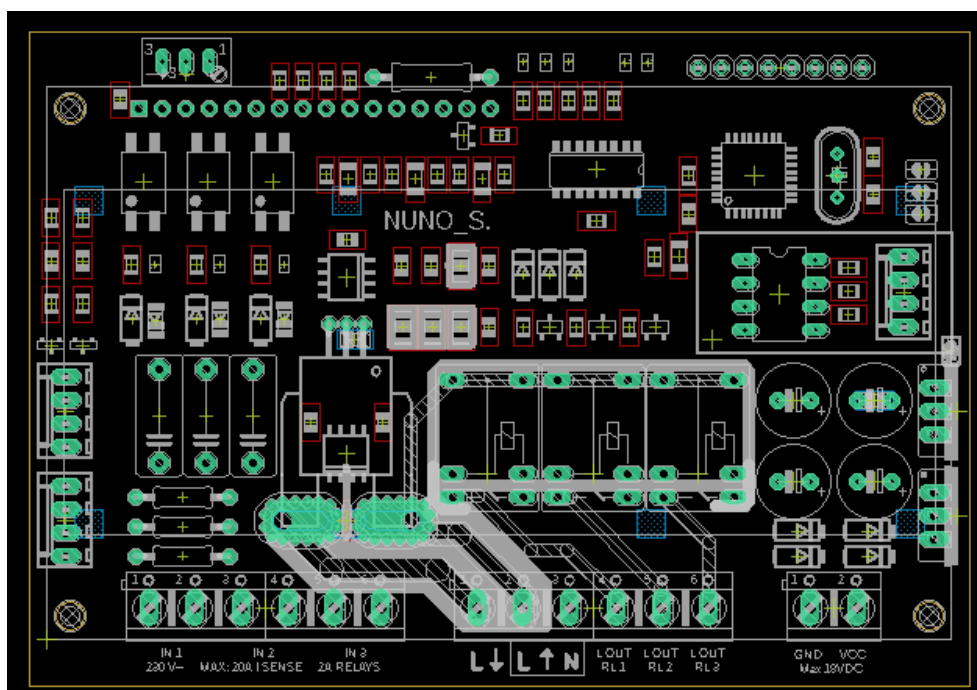


## Anexo B

---

# Protótipo do PCB do controlador remoto da eletrobomba

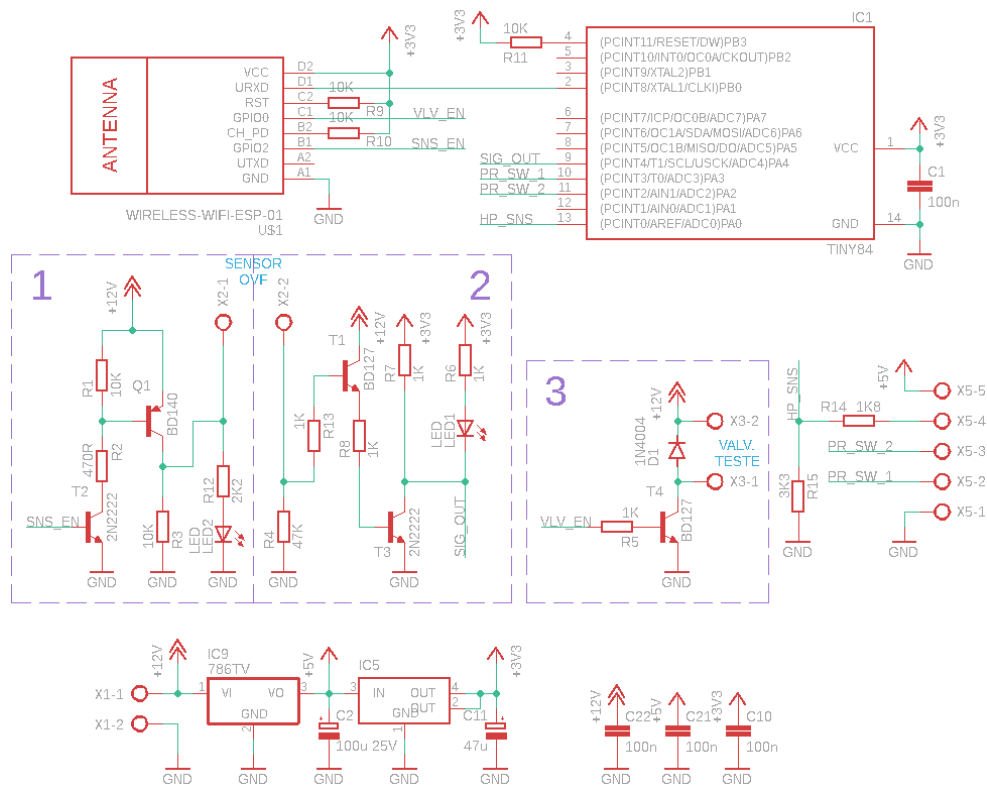
---





## Anexo C

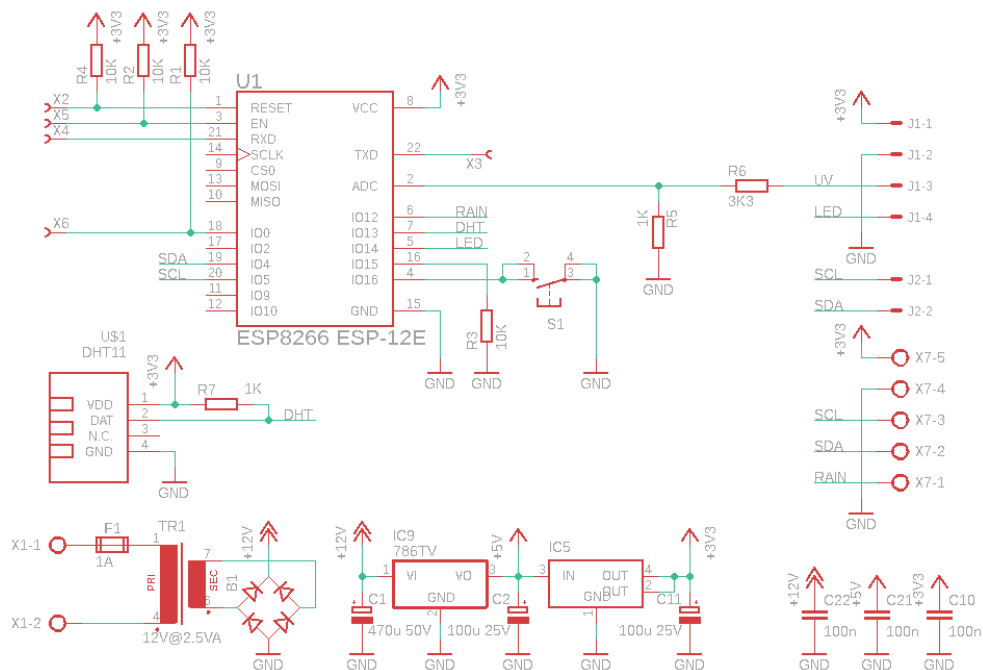
# Esquema do controlador remoto dos sensores de sobrecarga e pressões





## Anexo D

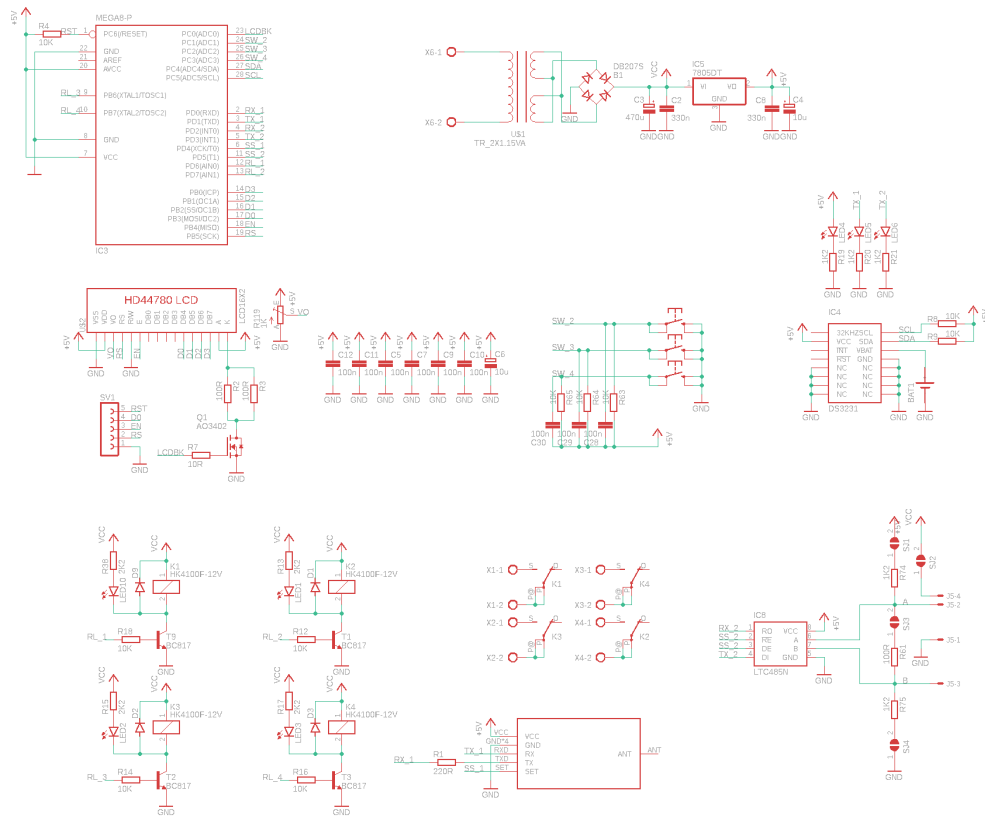
# Esquema do controlador remoto de recolha de informação meteorológica





## Anexo E

# Esquema do controlador remoto do sistema de rega auxiliar



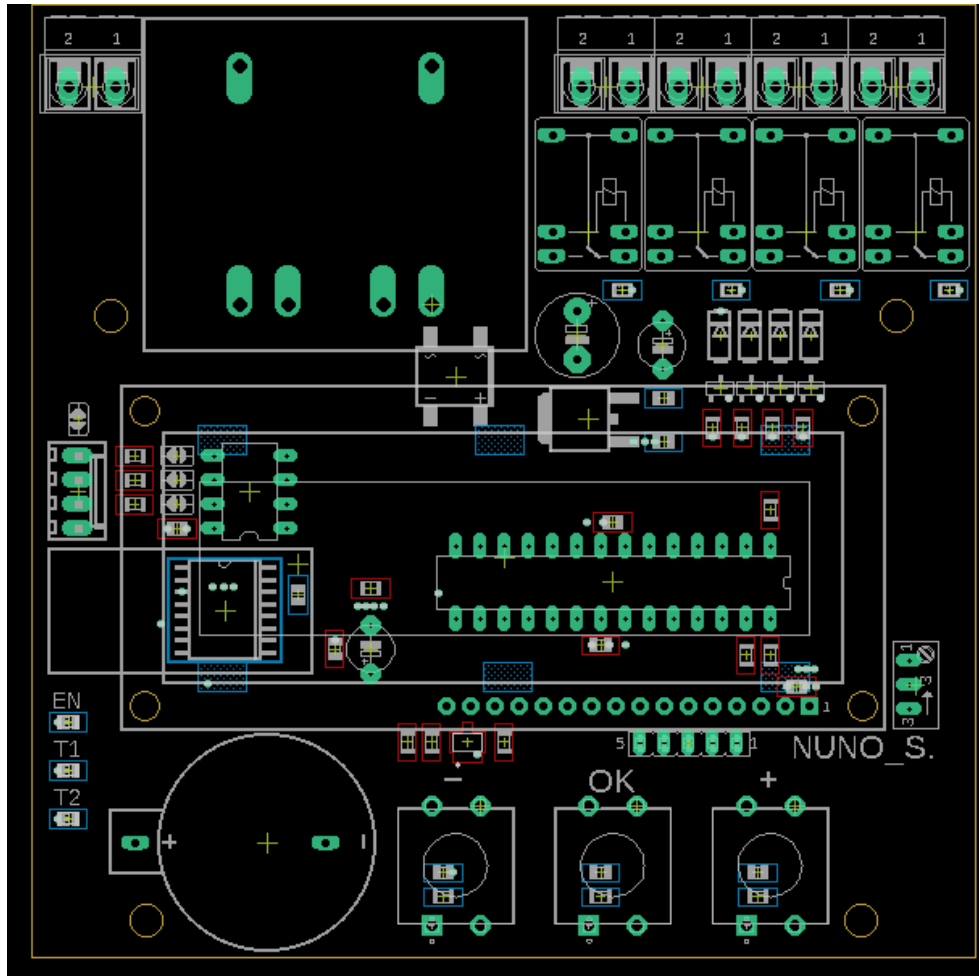


Anexo F

---

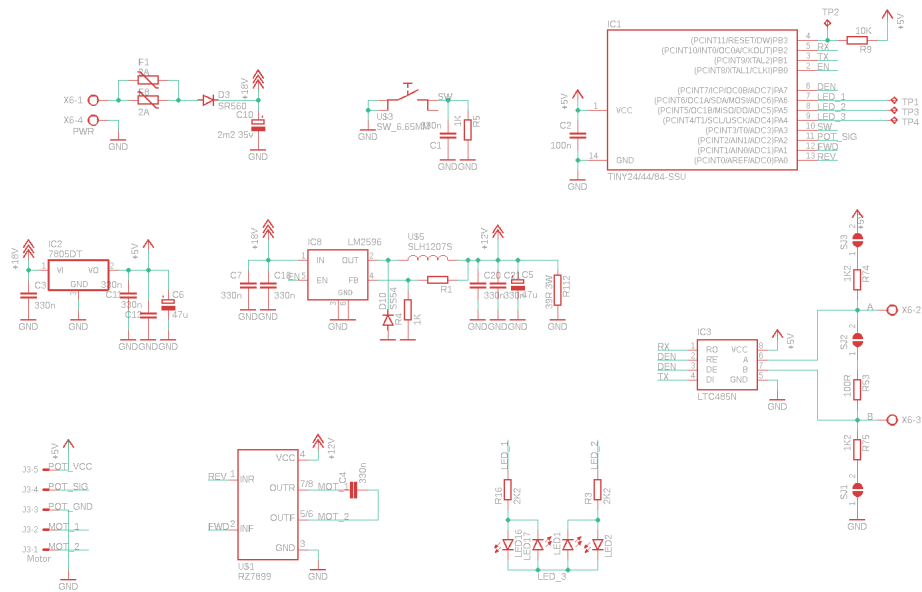
**Prototipo do PCB do controlador  
remoto do sistema de rega  
auxiliar**

---



## Anexo G

# Arquitetura do hardware das válvulas servo-motorizadas





## Anexo H

---

# Prototipo do PCB das válvulas servo-motorizadas

---

