



João Bonfim Ribeiro Carou Cunha

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

Controlador de Aquário de Água Salgada

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Anikó Katalin Horváth da Costa, Professora Auxiliar FCT-UNL

Júri:

Presidente: Doutor André Teixeira Bento Damas Mora

Arguentes: Doutor Rui Manuel Leitão Santos Tavares

Vogais: Doutora Anikó Katalin Horváth da Costa

Setembro, 2019



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Controlador de Aquário de Água Salgada

Copyright © João Bonfim Ribeiro Carou Cunha, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Esta dissertação apresenta a etapa final do meu percurso acadêmico e deste modo pretendo agradecer a algumas pessoas que me apoiaram nesta fase.

Em primeiro lugar, quero agradecer à minha orientadora, Anikó Katalin Horváth da Costa, por toda a envolvimento na minha dissertação.

De seguida, agradeço aos meus amigos, uns que conheci no meio académico e com quem partilhei esta vivência e outros que presenciaram a minha evolução nesta etapa.

Quero ainda agradecer à Daniela Rocha por todo o apoio incondicional durante este percurso e que sempre esteve presente e me ajudou a superar as minhas dificuldades e encorajou a lutar pelos meus objetivos.

Por fim, agradeço a toda a minha família, em particular aos meus pais que me apoiaram incondicionalmente e me proporcionaram esta oportunidade.

Resumo

O mundo da aquariofilia de água salgada é muito concorrido. O desejo de ter uma porção do oceano em casa cativa muitas pessoas. No entanto, os ecossistemas de água salgada são muito complexos e de elevado custo, sendo por isso difíceis de manter. Dado o elevado custo dos elementos que compõem um sistema aquático, é importante evitar que estes avariem, até porque uma falha destes pode causar perdas de vida animal.

Os principais problemas são falhas em termostatos, falhas de iluminação, falhas nos sistemas de limpeza e falhas que gerem uma rápida alteração do pH da água.

Todos estes problemas podem ser solucionados através da utilização do controlador desenvolvido, ao qual se encontram ligados diversos módulos e sensores, de forma a que tanto os parâmetros da água, como os vários sistemas de controlo da qualidade do ecossistema sejam controlados por uma única entidade. Este controlador possui assim conhecimento de tudo o que se passa no ecossistema. Através da utilização de uma outra placa programável, conectada ao controlador, que funcione como um servidor de internet, é possível colocar todos estes valores numa base de dados e, posteriormente, numa interface web, proporcionando assim um fácil acesso a todas as informações do ecossistema.

Palavras-chave: Ecossistemas de água salgada, Falhas, Controlador, Base de dados, Interface Web, Controlo Remoto, Vídeo em direto

Abstract

There is a lot of interest regarding the world of saltwater aquariums. The desire to have a piece of ocean at home captivates a lot of people. However, saltwater ecosystems are very complex and have a very high cost, which makes them hard to maintain. Given the high cost of the elements that make up an aquatic system, it is important to prevent them from malfunctioning, as failure to do so can lead to loss of life.

The main problems are thermostat failures; lighting failures; failures in cleaning systems and failures that generate a rapid change in water pH

All these problems are solved using the developed controller, to which various modules and sensors are connected, so that both water parameters and ecosystem quality control systems are controlled by a single entity. This way the controller has knowledge of everything that goes on in the ecosystem. By using another programmable board, connected to the controller, that functions as a web server, it is possible to place all these values in a database and later in a web interface, thus providing easy access to all information in the ecosystem.

Keywords: Saltwater ecosystems, Failure, Controller, Database, Web interface, Remote Control, Live Video.

Índice Geral

| | |
|--|-------------|
| Agradecimentos | ii |
| Resumo | iii |
| Abstract | v |
| Índice Geral..... | vii |
| Índice de Tabelas | xi |
| Índice de Figuras | xiii |
| Acrónimos e Simbologia..... | xvii |
| 1 Introdução | 1 |
| 1.1 Motivação..... | 1 |
| 1.2 Objetivos | 2 |
| 1.3 Contribuição | 3 |
| 1.4 Estrutura | 3 |
| 2 Estado da Arte..... | 5 |
| 2.1 Soluções comerciais | 5 |
| 2.1.1 <i>Inkbird ITC-308</i> | 5 |
| 2.1.2 <i>Digital Aquatics Reefkeeper Lite Net</i> | 8 |
| 2.2 Comparação entre soluções | 11 |
| 3 Tecnologia de Suporte | 13 |
| 3.1 Família de sensores úteis em controladores de água salgada | 13 |
| 3.1.1 <i>Sensor de Temperatura</i> | 14 |
| 3.1.2 <i>Sensor de pH</i> | 15 |
| 3.1.3 <i>Sensor de Nível de Fluido</i> | 16 |
| 3.1.4 <i>Sensor de Imagem</i> | 16 |

| | | |
|----------|--|-----------|
| 3.1.5 | <i>Sensor Ultrassónico</i> | 18 |
| 3.1.6 | <i>Sensor de Salinidade</i> | 19 |
| 3.2 | Plataformas para controladores | 20 |
| 3.2.1 | <i>Plataforma Arduino</i> | 20 |
| 3.2.2 | <i>Plataforma RaspberryPi 3 Modelo B+</i> | 24 |
| 3.3 | Métodos de comunicação utilizados por controladores de água salgada..... | 25 |
| 3.3.1 | <i>Comunicação Série</i> | 25 |
| 3.3.2 | <i>Comunicação por Radiofrequência</i> | 27 |
| 3.3.3 | <i>Comunicação por Wi-Fi</i> | 30 |
| 4 | Solução Proposta | 31 |
| 4.1 | Estrutura geral | 31 |
| 4.1.1 | <i>Controlador Principal-Master</i> | 35 |
| 4.1.2 | <i>Controlador Secundário Slave_Sensores</i> | 38 |
| 4.1.3 | <i>Controlador Secundário Slave_Relês</i> | 40 |
| 4.1.4 | <i>Servidor de internet</i> | 42 |
| 5 | Componentes e Implementação da Solução Desenvolvida | 45 |
| 5.1 | Controlador principal – <i>Master</i> | 45 |
| 5.1.1 | <i>Componentes Controlador Principal-Master</i> | 45 |
| 5.1.2 | <i>Ligações dos componentes do controlador principal-Master</i> | 49 |
| 5.1.3 | <i>Implementação</i> | 50 |
| 5.2 | Controlador Secundário- <i>Slave_Sensores</i> | 57 |
| 5.2.1 | <i>Componentes Controlador Secundário-Slave_Sensores</i> | 57 |
| 5.2.2 | <i>Ligações dos componentes do controlador Secundário-Slave_Sensores</i> ... | 62 |
| 5.2.3 | <i>Implementação</i> | 63 |
| 5.3 | Controlador Secundário- <i>Slave_Relês</i> | 66 |
| 5.3.1 | <i>Componentes do Controlador Secundário-Slave_Relês</i> | 67 |
| 5.3.2 | <i>Ligações dos componentes do controlador Secundário-Slave_Relês</i> | 69 |
| 5.3.3 | <i>Implementação</i> | 70 |
| 5.4 | Servidor de Internet..... | 72 |
| 5.4.1 | <i>Componentes do Servidor de Internet</i> | 72 |
| 5.4.2 | <i>Ligações dos componentes do Servidor de Internet</i> | 73 |
| 5.4.3 | <i>Implementação</i> | 74 |
| 6 | Validação do funcionamento | 83 |
| 6.1 | Validação..... | 83 |
| 6.1.1 | <i>Validação do correto funcionamento em situações normais</i> | 84 |
| 6.1.2 | <i>Validação do correto funcionamento em situações extraordinárias</i> | 91 |
| 7 | Conclusões e Trabalho Futuro | 98 |
| 7.1 | Conclusões | 98 |
| 7.2 | Trabalho Futuro | 100 |

Referências 101

Índice de Tabelas

| | |
|--|-----|
| Tabela 2-1 - Comparação entre soluções existentes e solução proposta | 12 |
| Tabela 3-1 - Características técnicas do Arduino Uno | 21 |
| Tabela 3-2 - Características do Arduino Micro | 22 |
| Tabela 3-3 - Principais características do Arduino Mega 2560..... | 23 |
| Tabela 3-4 - Características RaspberryPi 3 Modelo B+ | 25 |
| Tabela 5-1 - Codificação informações enviadas pelo Servidor de Internet..... | 53 |
| Tabela 5-2 - Ligações relês..... | 54 |
| Tabela 5-3 - Horas a que as luzes de ligam e desligam..... | 55 |
| Tabela 5-4 - Funcionalidades do Sistema..... | 56 |
| Tabela 5-5 - Valores registados na calibração do módulo de pH..... | 65 |
| Tabela 5-6 - Estrutura da mensagem recebida pelo Servidor de Internet..... | 74 |
| Tabela 5-7 - Configurações necessárias de realizar na biblioteca <i>Motion</i> | 80 |
| Tabela 8 - Custos dos diversos elementos do controlador desenvolvido | 100 |

Índice de Figuras

| | |
|--|----|
| Figura 2-1 - Constituintes Inkbird ITC-308 [6] - Adaptado | 6 |
| Figura 2-2 - Fluxograma do funcionamento do Inkbird ITC-308 | 7 |
| Figura 2-3 - Controlador principal do Digital Aquatics Reefkeeper Lite Net | 8 |
| Figura 2-4 - Controlador central e módulo de controlo de potência | 9 |
| Figura 2-5 - Sonda de leitura de temperatura | 9 |
| Figura 2-6 - Guia de conexão dos diversos módulos e sondas | 10 |
| Figura 3-1 - Termistor | 15 |
| Figura 3-2 - Exemplo de sistema com duas sondas | 16 |
| Figura 3-3 - Sensor de nível de fluído digital | 16 |
| Figura 3-4 - Câmara USB HP HD-2200 | 17 |
| Figura 3-5 - Picamera – Módulo Câmara RaspberryPi V2 | 18 |
| Figura 3-6 - Sensor ultrassónico | 18 |
| Figura 3-7 - Esquema de funcionamento do sensor ultrassónico | 19 |
| Figura 3-8 - Sensor de salinidade | 20 |
| Figura 3-9 - Arduino Uno | 21 |
| Figura 3-10 - Arduino Micro | 22 |
| Figura 3-11 - Arduino Mega 2560 VER3 | 23 |
| Figura 3-12 - RaspberryPi 3 Modelo B+ | 24 |
| Figura 3-13 - Esquema de comunicação <i>full-duplex</i> | 26 |
| Figura 3-14 - Comunicação com 1 transmissor para 2 recetores | 26 |
| Figura 3-15 - Comunicação com 2 transmissores para 1 recetor | 26 |
| Figura 3-16 - Linha bus do protocolo I2C | 27 |
| Figura 3-17 - Envio da informação no I2C | 27 |

| | |
|--|----|
| Figura 3-18 - Esquema representativo do funcionamento da comunicação radiofrequência | 28 |
| Figura 3-19 - Módulos recetor e emissor de RF | 28 |
| Figura 3-20 - Módulo transceiver de RF | 29 |
| Figura 3-21 - Ilustração do funcionamento de uma rede Wi-Fi | 30 |
| Figura 3-22 - <i>Wireless router TP-LINK</i> | 30 |
| Figura 4-1 - Configuração geral do controlador | 33 |
| Figura 4-2 - Caso de uso do controlador | 34 |
| Figura 4-3 - Diagrama comportamental do controlador principal- <i>Master</i> | 37 |
| Figura 4-4 - Diagrama comportamental do controlador secundário <i>Slave_Sensores</i> | 39 |
| Figura 4-5 - Diagrama comportamental do controlador secundário <i>Slave_Relês</i> | 41 |
| Figura 4-6 - Diagrama comportamental do servidor de internet | 43 |
| Figura 5-1 - Botões de pressão | 46 |
| Figura 5-2 - Funções dos botões de pressão | 47 |
| Figura 5-3 - Ecrã LCD..... | 47 |
| Figura 5-4 - Módulo de I2C ligado ao LCD | 48 |
| Figura 5-5 - Ligações módulo I2C para LCD..... | 48 |
| Figura 5-6 - Montagem Controlador Principal- <i>Master</i> | 49 |
| Figura 5-7 - Estrutura correspondente à informação lida pelo <i>Slave_Sensores</i> | 51 |
| Figura 5-8 - Configuração comunicação por radiofrequência..... | 51 |
| Figura 5-9 - Fluxograma da lógica de leitura de informações por radiofrequência | 52 |
| Figura 5-10 - Fluxograma da lógica de receção de informações enviadas pelo Servidor de Internet | 54 |
| Figura 5-11 - Módulo DM PH-4502C | 58 |
| Figura 5-12 - Sonda do módulo DM PH-4502C | 58 |
| Figura 5-13 - Ligações DM PH-4502C | 58 |
| Figura 5-14 - Módulo DS3231 | 59 |
| Figura 5-15 - Ligações DS3231 | 59 |
| Figura 5-16 - Sensor nível de água..... | 60 |
| Figura 5-17 - Termístor à prova de água | 60 |
| Figura 5-18 - Módulo HC-SR04..... | 61 |
| Figura 5-19 - Ligações HC-SR04..... | 61 |
| Figura 5-20 - Ligações NRF24L01 | 61 |
| Figura 5-21 - Montagem Controlador Secundário- <i>Slave_Sensores</i> | 62 |
| Figura 5-22 - Estrutura do Controlador Secundário- <i>Slave_Sensores</i> | 63 |
| Figura 5-23 - Gráfico de calibração do módulo de pH..... | 65 |
| Figura 5-24 - Módulo relês..... | 67 |

| | |
|--|----|
| Figura 5-25 - Ligações módulo relê | 68 |
| Figura 5-26 - Montagem ficha fêmea ao relê | 68 |
| Figura 5-27 - Montagem Controlador Secundário- <i>Slave_Relês</i> | 69 |
| Figura 5-28 - Estrutura usada pelo Controlador Secundário- <i>Slave_Relês</i> | 70 |
| Figura 5-29 - Fluxograma do funcionamento do envio e receção de informações do Controlador Secundário- <i>Slave_Relês</i> | 71 |
| Figura 5-30 - Montagem do Servidor de Internet..... | 73 |
| Figura 5-31 - Interface de início de sessão na ferramenta <i>phpmyadmin</i> | 75 |
| Figura 5-32 - Base de dados e tabela para o controlador..... | 76 |
| Figura 5-33 - Parte da tabela "Sensores" da base de dados "Controlador Aquário" | 76 |
| Figura 5-34 - Página Principal da interface web | 77 |
| Figura 5-35 - Página inicial da interface web..... | 78 |
| Figura 5-36 - Modo de funcionamento do servidor <i>Flask</i> | 78 |
| Figura 5-37 - Página Funcionalidades da interface web | 79 |
| Figura 5-38 - Fluxograma do funcionamento de um pedido via a página web Funcionalidades | 79 |
| Figura 5-39 - Página Câmara da interface web | 80 |
| Figura 6-1 Validação dos sistemas de iluminação às 13:15H | 84 |
| Figura 6-2 - Validação dos sistemas de iluminação às 13:59H..... | 85 |
| Figura 6-3 - Validação dos sistemas de iluminação às 16:48H..... | 85 |
| Figura 6-4 - Validação dos sistemas de iluminação às 21:48H..... | 85 |
| Figura 6-5 - Validação dos sistemas de iluminação às 22:48H..... | 86 |
| Figura 6-6 - Estados dos relês de controlo no modo limpeza (1)..... | 86 |
| Figura 6-7 - Estados dos relês de controlo no modo limpeza (2)..... | 87 |
| Figura 6-8 - Estado dos reles de controlo apos a ativação do modo normal (1) | 87 |
| Figura 6-9 - Estado dos reles de controlo após ativação do modo normal (2) | 87 |
| Figura 6-10 - Desativação manual do termostato | 88 |
| Figura 6-11 – Ativação e desativação do rele correspondente à luz do aquário de filtração..... | 88 |
| Figura 6-12 - Desativação do rele correspondente ao escumador sob ordem do utilizador | 89 |
| Figura 6-13 - Alteração dos estados de ativação dos reles correspondentes ao motor retorno e escumador..... | 89 |
| Figura 6-14 - Comportamento do sistema perante uma diminuição e posterior aumento do nível de água..... | 90 |
| Figura 6-15 - Estados dos reles de controlo do sistema todos desativados devido a ordem do utilizador..... | 90 |

| | |
|---|----|
| Figura 6-16 – Alteração estado rele correspondente ao termostato devido ao aumento da temperatura da água..... | 91 |
| Figura 6-17 - Indicação na base de dados da temperatura da água às 17:01H..... | 92 |
| Figura 6-18 - Email recebido pelo utilizador às 17:02H a informar da temperatura da água elevada | 92 |
| Figura 6-19 - Detecção de falha no motor retorno às 17:26H e consequente desativação do rele correspondente ao escumador..... | 93 |
| Figura 6-20 - Ativação do rele correspondente às ventoinhas de arrefecimento decido ao aumento de temperatura ambiente..... | 93 |
| Figura 6-21 - Registo de temperatura ambiente elevada às 18:15H..... | 94 |
| Figura 6-22 - Email recebido pelo utilizador às 18:15H a avisar da temperatura ambiente elevada | 94 |
| Figura 6-23 – Alteração do estado do rele do escumador devido à diminuição do valor de pH | 95 |
| Figura 6-24 - Registo na base de dados do valor de pH às 18:37H..... | 95 |
| Figura 6-25 - Email recebido pelo utilizador às 18:38H relativo ao baixo valor de pH | 95 |
| Figura 6-26 - Registo da não ativação do rele correspondente à bomba de retorno de água evapora perante uma diminuição no nível de água do aquário | 96 |
| Figura 6-27 - Email recebido pelo utilizador quando o nível de água no reservatório é baixo | 96 |
| Figura 6-28 - Registo da ativação dos reles correspondentes ao termostato principal e termostato de segurança perante um valor de temperatura da água muito baixo ... | 97 |

Acrónimos e Simbologia

| | |
|-------------|--|
| CSI | Camera Serial Interface |
| CE | Chip Enable |
| CSN | Chip Enable Not |
| DC | Direct Current |
| GPIO | General Purpose Input/Output |
| GND | Ligação à Terra (Ground) |
| HDMI | High Definition Media Interface |
| I/O | Entrada/Saída (Input/Output) |
| ICSP | In Circuit Serial Programming |
| I2C | Inter-Integrated Circuit |
| LED | Light Emitting Diode |
| LCD | Liquid Crystal Display |
| LDR | Light dependent resistor |
| MOSI | Master Output Serial Input |
| MISO | Master Input Serial Output |
| NTR | Negative Temperature Coefficient of Resistance |
| PWM | Pulse Width Modulation |
| PTR | Positive Temperature Coefficient of Resistance |

| | |
|----------------------|---|
| RX | Recetor |
| RF | Radiofrequência |
| SCL | Serial Clock Line |
| SDA | Serial Data Line |
| SCK | Serial Clock |
| SPI | Serial peripheral interface |
| SG | Specific Gravity |
| SDA | Serial Data Line |
| TX | Transmissor |
| T^a | Temperatura |
| UART | Universal Asynchronous Receiver Transmitter |
| USB | Universal Serial Bus |



Introdução

Neste capítulo é apresentado o problema por detrás da motivação em que esta dissertação se baseia, tal como os seus objetivos e contribuições para o melhoramento do problema em causa. Por fim, encontra-se apresentada a estrutura do documento.

1.1 Motivação

O mundo está repleto de paisagens e lugares estonteantes, desde ravinhas a oceanos. No entanto, estes nem sempre são de fácil acesso. Acontece que, através da aquariofilia, é possível ter um pedaço do oceano onde quisermos [1], pois dentro de um aquário existe um ecossistema completo, isto é, há um sistema onde existem diversos organismos, desde peixes, a corais e a diversos microrganismos que interagem entre si de forma a que seja gerada uma harmonia na sua sobrevivência [2].

Existem dois tipos de ecossistemas na aquariofilia, os de água doce e os de água salgada. Esta dissertação foca-se nos ecossistemas de água salgada. No entanto, ambos possuem um grande problema, se não for realizada uma correta manutenção, ou ao ocorrer uma falha num dos vários aparelhos essenciais ao seu funcionamento, estes acabam por ser gravemente afetados.

Acontece que, muitas vezes os erros na manutenção dos ecossistemas são resultado de falhas de um, ou mais, aparelhos do sistema. Ao falhar, por exemplo, um aparelho de iluminação do aquário, todos os seres vivos que necessitam de luz para sobreviver irão ser afetados caso a falha não seja detetada e corrigida em pouco tempo.

Caso a falha de manutenção seja ao ponto do não seguimento do planeamento da troca de água, os parâmetros da água, como o pH, podem sofrer alterações. Tais alterações, afetam todos os seres vivos do ecossistema e devem ser evitadas a qualquer custo.

1.2 Objetivos

De modo a que todos os problemas associados ao mantimento e manutenção de ecossistemas de água salgada sejam evitados, o objetivo desta dissertação é arranjar um método de prever e solucionar todos estes problemas através de um método de baixo custo. Para tal existem diversos objetivos que devem ser cumpridos, nomeadamente:

- Desenvolver um controlador que seja capaz de ler e avaliar todos os parâmetros do ecossistema de água salgada, que seja capaz de controlar todos os aparelhos que garantem o correto funcionamento do ecossistema e que seja autónomo na resolução e prevenção de problemas.
- Desenvolver um sistema que seja capaz de disponibilizar informações vitais do ecossistema, como a sua temperatura, quantidade de água e valor de pH, numa interface web.
- Desenvolver um sistema capaz de ser controlável tanto localmente, como via uma interface web.
- Desenvolver um sistema capaz, através de comunicações *master-slave* [3], de ler e enviar para um *master* informações lidas por vários *slaves*. Este sistema deve também ser capaz de enviar ordens para os *slaves* vindas do *master* relativas à avaliação dos valores lidos pelos diversos sensores instalados no sistema.
- Desenvolver um sistema capaz, através de ligações *série* [4], de enviar informações lidas pelos sensores para uma base de dados e, posteriormente, para uma interface web. A interface web deve proporcionar ao utilizador um controlo total das diversas funcionalidades do sistema.

1.3 Contribuição

A dissertação pretendeu desenvolver um controlador para aquários de água salgada que fosse capaz de ler e processar diversos parâmetros da água, controlar todos os elementos de manutenção, gerar modos de funcionamento que tenham como objetivo simplificar a manutenção e interação com o aquário e gerir uma interface web que se encontra ligada a uma base de dados, sendo que todos os objetivos foram cumpridos.

Inicialmente foi criado um sistema mais simples no qual não havia qualquer ligação a um servidor de internet. Esta versão apenas lia e processava as informações lidas pelos diversos sensores e tomava ações com base nestas. Posteriormente, foi incorporado um servidor de internet nesta versão, o que possibilitou o desenvolvimento de uma interface web capaz de ilustrar o valor de diversos parâmetros do sistema e controlar todas as funcionalidades desenvolvidas para o sistema. A interface web desenvolvida possibilita também ao seu utilizador um visionamento de um vídeo em direto do aquário de água salgada, o que gera uma outra forma de o utilizador avaliar o estado do ecossistema.

Uma outra funcionalidade desenvolvida no controlador foi a criação de uma base de dados e posterior interligação com todos os elementos do sistema. Desta forma, o utilizador possui uma forma de guardar e visualizar todas as ações realizadas no aquário e alterações nos diversos parâmetros medidos.

O controlador possui também uma interface física, através da qual o utilizador pode controlar todas as funcionalidades dos sistema e visualizar informações sobre este.

1.4 Estrutura

Esta dissertação está dividida em seis capítulos, através dos quais são apresentados em detalhe os diversos componentes e métodos utilizados no controlador desenvolvido.

- **Capítulo 1: Introdução.** No primeiro capítulo é feita uma introdução ao tema e são indicados os principais objetivos e contribuições desta dissertação.
- **Capítulo 2: Estado da arte.** O segundo capítulo apresenta as soluções existentes para o problema apresentado na introdução, tal como os seus prós e contras. São também apresentados os diversos tipos de sensores, métodos

de comunicação e placas programáveis utilizáveis em controladores para aquários de água salgada.

- **Capítulo 3: Tecnologia de Suporte.** No terceiro capítulo são apresentadas todas as tecnologias de suporte.
- **Capítulo 4: Solução Proposta.** No quarto capítulo é introduzida a solução proposta e apresentada a sua estrutura geral, tal como o funcionamento de cada um dos sub-controladores utilizados.
- **Capítulo 5: Componentes e Implementação da Solução Desenvolvida.** O quinto capítulo apresenta os diversos componentes de cada sub-controlador, tal como a sua montagem. São também apresentados os métodos de implementação usados para cada um dos sub-controladores.
- **Capítulo 6: Validação.** Neste capítulo são testadas todas as funcionalidades do sistema de forma a que estas sejam validadas. São validadas tanto funcionalidades autónomas, como funcionalidades forçadas pelo utilizador.
- **Capítulo 7: Conclusões e Trabalho Futuro.** Este capítulo apresenta comentários e críticas relativas à implementação, o funcionamento do controlador, os custos associados ao desenvolvimento do controlador e o trabalho futuro que pode ser feito relativo ao controlador desenvolvido.



Estado da Arte

Serve o presente capítulo para apresentar duas soluções comerciais existentes relativas ao tema abordado, seguidas de uma comparação entre estas e a solução proposta. Posteriormente são apresentadas várias famílias de sensores, microcontroladores e comunicações úteis para este tipo de solução.

2.1 Soluções comerciais

2.1.1 Inkbird ITC-308

Um dos controladores existentes no mercado é o Inkbird ITC-308. O Inkbird ITC-308 é um controlador de temperatura que tem como principal função manter a temperatura do ecossistema o mais constante possível. No entanto, ao contrário de um termostato normal, o Inkbird ITC-308 não tem apenas como funcionalidade o aquecimento do ecossistema. Acontece que, o Inkbird ITC-308 controla tanto o aquecimento como o arrefecimento do ecossistema. Para tal, este possui uma extensão própria com duas tomadas elétricas, às quais devem ser ligados sistemas de arrefecimento e aquecimento do ecossistema. É de salientar que estes têm de ser ligados em tomadas específicas, caso contrário o controlador não terá o funcionamento desejado [5]. Na Figura 2-1 estão representados os diferentes componentes do Inkbird ITC-308.

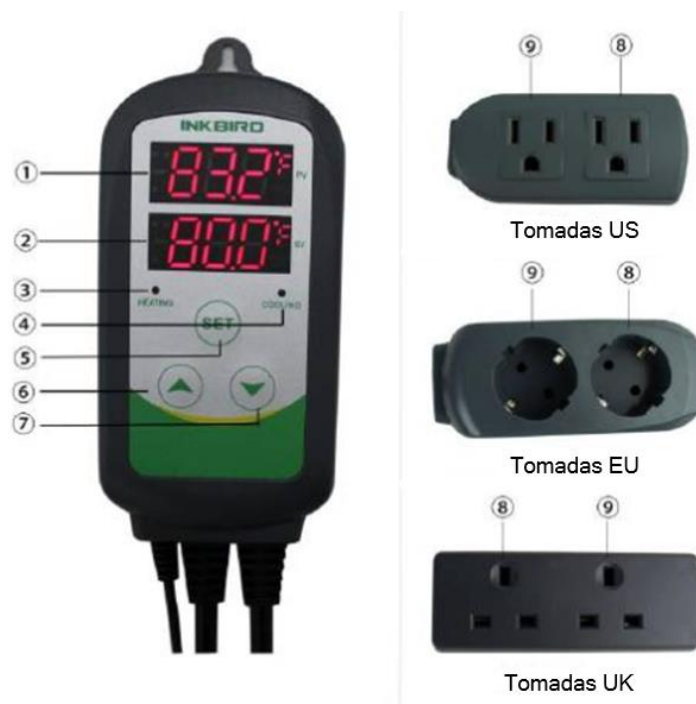


Figura 2-1 - Constituintes Inkbird ITC-308 [6] - Adaptado

1. *Display* onde mostra a temperatura do ecossistema.
2. *Display* onde mostra a temperatura objetivo.
3. Luz indicadora de funcionamento do sistema de arrefecimento.
4. Luz indicadora de funcionamento do sistema de aquecimento.
5. Botão de confirmação da temperatura introduzida.
6. Botão para aumentar a temperatura objetivo.
7. Botão para diminuir a temperatura objetivo.
8. Tomada para ligar um sistema de aquecimento do ecossistema.
9. Tomada para ligar um sistema de arrefecimento do ecossistema.

As ligações do Inkbird ITC-308 são todas realizadas por cabo. Este possui um sensor de temperatura à prova de água, o qual é usado para determinar a temperatura exata do ecossistema. Possui também um controlador central à prova de água com dois *displays* de 4 dígitos de 7 segmentos onde são mostradas duas temperaturas, a temperatura atual e a temperatura objetivo. Quando a temperatura atual é inferior à objetivo, é ativado o relê correspondente à tomada do aquecimento. No entanto, quando a temperatura atual é superior à objetivo, é ativado o relê correspondente à tomada de arrefecimento. O seu funcionamento é exemplificado no fluxograma na Figura 2-2. [5]

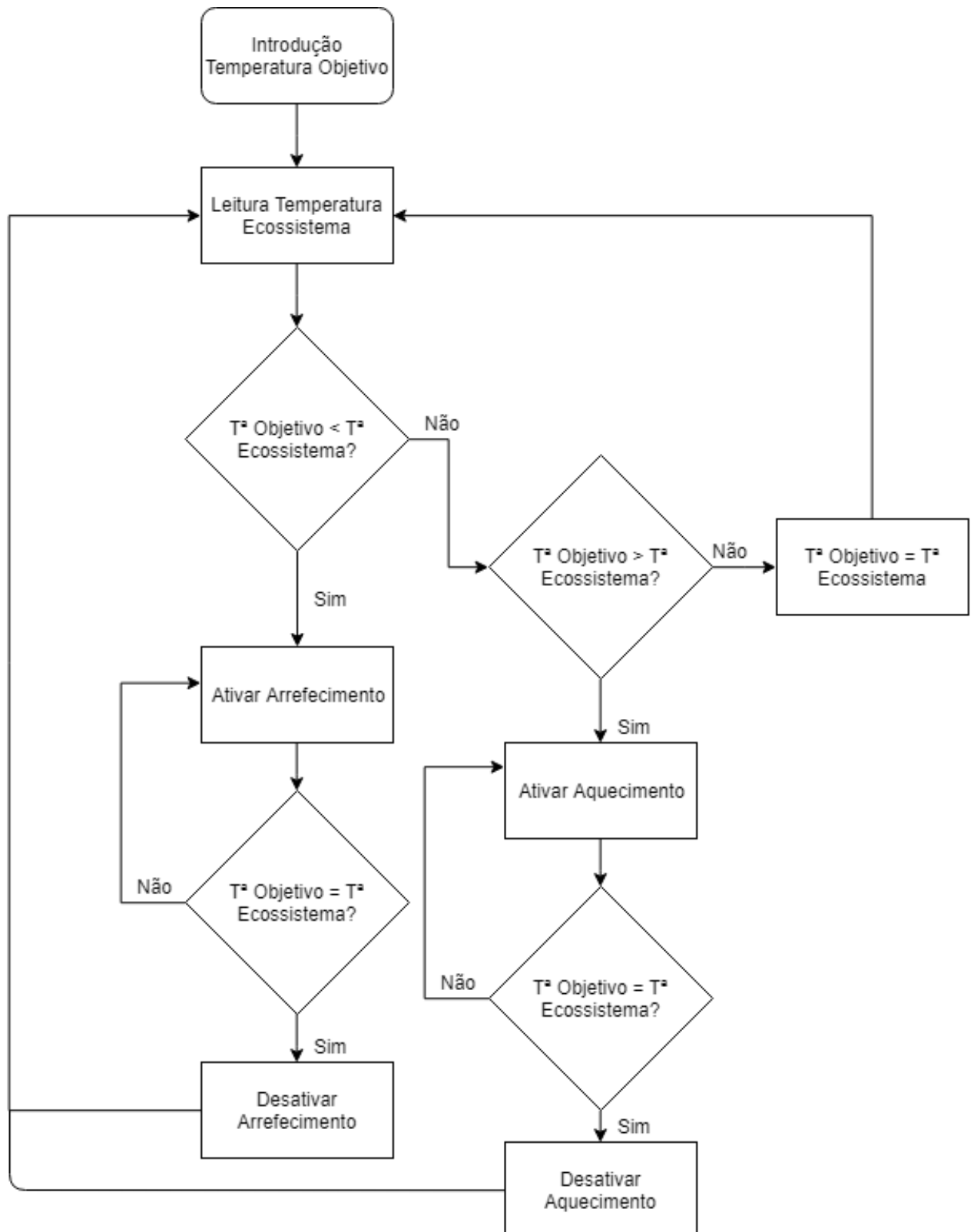


Figura 2-2 - Fluxograma do funcionamento do Inkbird ITC-308

2.1.2 Digital Aquatics Reefkeeper Lite Net

O Digital Aquatics Reefkeeper Lite Net é um controlador de aquário capaz de controlar uma bomba de retorno, um termostato e um sistema de iluminação de um ecossistema através do uso de diversos módulos e sondas. Esses módulos, através do uso das sondas, possibilitam também a leitura, em tempo real, da temperatura e do nível de pH do ecossistema, tal como a sua disponibilização numa interface web. Para além dos módulos, este é constituído por um controlador principal, ilustrado na Figura 2-3, onde são mostradas todas as informações do ecossistema e onde são definidos os objetivos do controlador, e um controlador central, ilustrado na Figura 2-4, para onde são enviados todos os valores lidos e efetuadas as ações necessárias para manter o correto funcionamento do ecossistema. [7]



Figura 2-3 - Controlador principal do Digital Aquatics Reefkeeper Lite Net [8]

Existem diversos módulos que, quando usados em conjunto com o controlador central, permitem ao utilizador ter um controlo total do ecossistema.

- Módulo AP1 – Usado para ler e enviar valores de temperatura ambiente e humidade relativa.
- Módulo NET – Usado para enviar informação do sistema, em tempo real, para uma interface web.
- Módulo SW5 – Módulo ao qual se ligam sensores de nível de água. É usado para impedir que o nível de água do ecossistema desça em demasia.

- Módulo SL2v2 – Módulo ao qual se ligam os sensores de pH, temperatura e salinidade.

De forma a que a informação enviada pelos módulos seja processada, estes têm de ser ligados fisicamente ao controlador central. Este controlador irá depois processar a informação recebida pelos sensores e pelo controlador principal. Com base nessa informação irá depois realizar uma, ou mais, ações de forma a manter o funcionamento desejado do ecossistema.



Figura 2-4 - Controlador central e módulo de controlo de potência [8]

A leitura da temperatura do ecossistema é feita através de uma sonda, ilustrada na Figura 2-5. A sonda regista o valor lido e envia-o para o controlador central. Esse valor é depois comparado com o introduzido no controlador principal e, seguidamente, é tomada uma ação sobre se é necessário, ou não, ligar o sistema de aquecimento do ecossistema.



Figura 2-5 - Sonda de leitura de temperatura [8]

Em suma, o Digital Aquatics Reefkeeper Lite Net permite um controlo dos principais elementos do ecossistema de uma forma simples. É de salientar o facto de este permitir um acesso às informações lidas numa interface web. Um guia de conexões dos diversos componentes do Reefkeeper Lite Net é apresentado na Figura 2-6. [7]

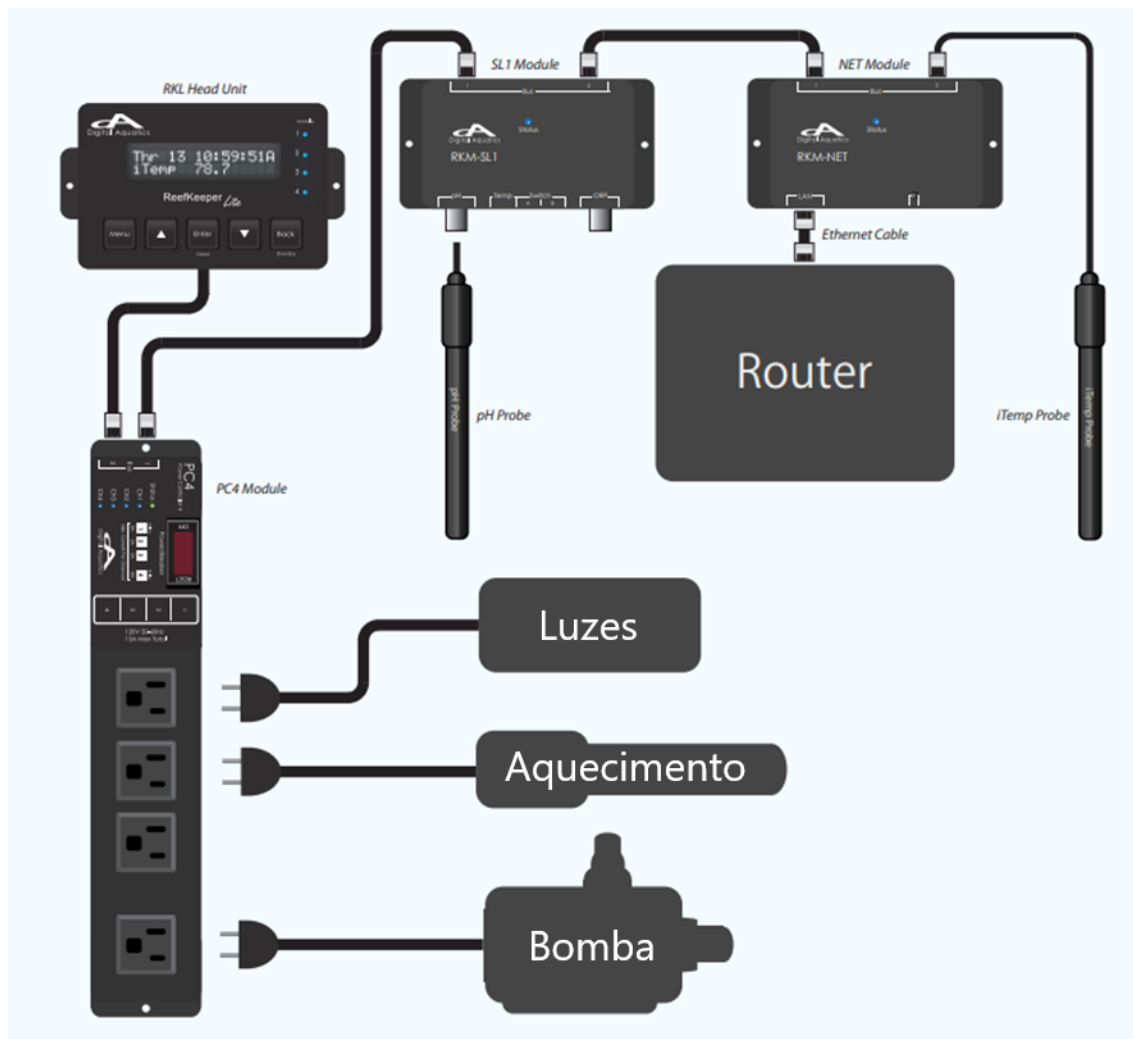


Figura 2-6 - Guia de conexão dos diversos módulos e sondas [7] – Adaptado

2.2 Comparação entre soluções

Ao longo deste capítulo foram apresentadas diversas soluções já existentes no mercado. Acontece que, as soluções apresentadas possuem alguns problemas, tanto ao nível dos sensores utilizados, como ao nível das suas funcionalidades.

Os sensores usados nas soluções existentes utilizam uma ligação física para se conectarem ao controlador principal. Tal escolha gera problemas ao serem instalados em sistemas de grandes dimensões, pois a dimensão do cabo de ligação pode não ser suficiente para chegar ao controlador principal após a instalação do sensor no sistema. Na solução desenvolvida, são utilizados sensores que comunicam com o controlador principal por meio de tecnologias sem fios, não tendo assim este problema.

Outro problema associado aos controladores existentes é a não existência de sensores de temperatura ambiente. Os sensores de temperatura ambiente são fundamentais para informar o utilizador da existência de uma maior probabilidade de o sistema sobreaquecer, caso este não possua um sistema de arrefecimento instalado, ou arrefecer em demasia, caso a temperatura ambiente desça demasiado. No caso do aumento da temperatura ambiente, se existir um sistema de arrefecimento, o controlador pode ativá-lo a tempo de minimizar a flutuação de temperaturas no ecossistema. Para o caso da diminuição da temperatura, o sistema pode ligar um ou mais termostatos de emergência de forma a garantir que a temperatura do ecossistema se mantém o mais constante possível.

Algumas das soluções existentes apenas possuem a funcionalidade de colocar alguns valores medidos numa interface web, para que o utilizador os possa verificar. No entanto, estes valores não são guardados em nenhum local. O que significa que caso tenha havido uma flutuação de valores num instante em que o utilizador não os estivesse a verificar, este não tem forma de saber que tal tenha acontecido. Na solução desenvolvida todos os valores medidos são guardados numa base de dados, permitindo assim ao utilizador verificar todos os valores ao longo de um período de tempo. Estes valores são também avaliados desta forma. Caso exista algum problema com um destes, o controlador avisa, através de um e-mail, o utilizador. A solução desenvolvida possibilita também ao utilizador ter um controlo direto com o sistema. Se este pretender ligar ou desligar algum ou alguns componentes do sistema, pode realizá-lo através da interface web em tempo real. Existe também a possibilidade de visualizar o sistema em tempo real na interface web, através da ativação da câmara existente no controlador.

A Tabela 2-1 ilustra estas diferenças, tanto do ponto de vista de hardware como de software:

Tabela 2-1 - Comparação entre soluções existentes e solução proposta

| | | Inkbird ITC-308 | Digita Aquatics Reefkeeper Lite Net | Solução Proposta |
|---------------|--|-----------------|-------------------------------------|------------------|
| Sensores | Sensor Tª ambiente | | | Sem fios |
| | Sensor ph | | Com fios | Sem fios |
| | Sensor nível água | | | Sem fios |
| | Sensor Tª água | | Com fios | Sem fios |
| | Sensor Nível de água do reservatório da água a repor | | | Sem fios |
| | Câmara | | | Com fios |
| Praticalidade | Mostrador de informações | | | |
| | Base de dados | | | |
| | Controlável pela Web | | | |
| | Interface Web | | | |
| | Avisos por e-mail | | | |

3

Tecnologia de Suporte

Serve o presente capítulo para apresentar todas as tecnologias de suporte existentes no mercado. São apresentadas todas as famílias de sensores utilizáveis neste tipo de controladores, todas as tecnologias de troca de informação e comunicação utilizadas e as principais plataformas usadas de forma a proporcionar um controlador final de baixo custo e elevada eficiência.

3.1 Família de sensores úteis em controladores de água salgada

Um aquário de água salgada possui diversos parâmetros que necessitam de ser monitorizados de forma a garantir o seu correto funcionamento, tais como a sua temperatura, a temperatura do meio em que se encontra, o seu valor de pH, o seu valor de salinidade, o nível de água existente no aquário e o nível de água existente no reservatório de reposição de água evaporada.

De forma a garantir que a monitorização do sistema é realizada o mais eficientemente possível, podem ser utilizados sensores de imagem de forma a proporcionar ao utilizador um visionamento em tempo real do sistema.

3.1.1 Sensor de Temperatura

Existem diversos tipos de sensores de temperatura e todos possuem diferentes características, dependendo do tipo de aplicação para a qual foram desenhados:

- **Sensor de temperatura de contacto** – Este tipo de sensores necessita de estar em contacto direto com o objeto cuja temperatura pretende ser medida. Utiliza condução para monitorizar mudanças na temperatura. Pode ser usado para detetar sólidos, líquidos ou gases numa grande gama de variação de temperaturas.
- **Sensores de temperatura de não-contacto** – Este tipo de sensores não necessita de estar em contacto com os objetos cuja temperatura pretende ser medida. Utiliza convecção e radiação para monitorizar variações na temperatura. Pode ser usado para detetar líquidos e gases que emitam energia radiante à medida que o calor sobe e o frio desce nas correntes de convecção. Pode também detetar a energia radiante transmitida de um objeto na forma de radiação infravermelho, como por exemplo o sol. [9]

3.1.1.1 O Termistor

O termistor, ilustrado na Figura 3-1, é um sensor de contacto que consiste num tipo especial de resistor que varia a sua resistência física quando exposto a alterações de temperatura. São normalmente feitos de materiais de cerâmica, como óxidos de níquel, manganês ou cobalto revestido em vidro, sendo por isso em norma frágeis. É o principal tipo de sensor utilizado em controladores para aquários devido ao seu baixo custo e simplicidade de funcionamento, o que não seria o caso se este fosse um sensor de não-contacto.

A maior parte dos termístores possuem coeficiente de temperatura negativo de resistência, ou *Negative Temperature Coefficient of Resistance* (NTR), quer isto dizer que a sua resistência diminui com um aumento de temperatura. Existem outros que possuem um coeficiente de temperatura positivo de resistência, ou *Positive Temperature Coefficient of Resistance* (PTR). Deste modo, a sua resistência aumenta com o aumento da temperatura.



Figura 3-1 - Termistor [10]

Os termístores são classificados pelo seu valor de resistência à temperatura ambiente (normalmente 25°C) e pelo tempo que levam a reagir a uma variação na temperatura. São dispositivos resistivos passivos, o que significa que é necessário que haja corrente a passar por eles para que estes produzam alguma tensão de saída mensurável. Quando ligados a um controlador, o seu valor de saída é enviado para umas das portas analógicas. [11]

3.1.2 Sensor de pH

Por norma um sensor de pH possui dois componentes básicos, uma sonda e um mostrador, onde é indicado o valor medido. [12] No entanto, existem sensores que em vez de possuírem um mostrador, enviam o valor medido para a porta analógica de um controlador. Este último tipo de sensores são os mais usados em aplicações para controladores de aquários devido à sua fácil integração em sistemas informáticos.

O medidor pode possuir uma ou duas sondas. De forma a que seja possível determinar o valor de pH da solução de teste é necessário fazer com que esta seja atravessada por eletricidade. Para tal, têm de ser inseridos na solução dois elétrodos. Caso o sensor possua duas sondas, como ilustrado no exemplo da Figura 3-2, cada sonda é um eletrodo. No entanto, caso apenas possua uma sonda, ambos os elétrodos se situam dentro da única sonda, o que gera uma maior simplicidade e conveniência. Os dois elétrodos são constituídos por diferentes componentes, o que faz com que se comportem de diferentes formas perante uma mesma solução. É esta diferença em comportamento que permite determinar o nível de pH de uma solução. [13]

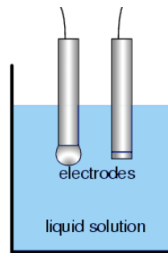


Figura 3-2 - Exemplo de sistema com duas sondas [14]

3.1.3 Sensor de Nível de Fluido

Existem diversos tipos de sensores para medir o nível de um fluido. Alguns sensores utilizam uma boia magnética que flutua consoante o nível de fluido e que, ao atingir um certo nível, ativa um interruptor magnético. Este tipo de sensor está ilustrado na Figura 3-3 e pode conter um interruptor magnético tanto para um nível máximo de fluido, como para um nível mínimo de fluido, sendo por isso o tipo de sensor mais usado em aplicações para controladores de nível de água. Um outro tipo de sensor é o sensor de condutividade. Este tipo de sensor é usado apenas em fluidos que conduzam eletricidade. Utiliza dois elétrodos de diferentes comprimentos que são colocados num fluido, quando este atinge um certo nível, entra em contacto com ambos os elétrodos, tanto o longo como o curto. Desta forma o circuito é fechado e um sinal é enviado. Quando ligados a um controlador, o sinal é enviado para a sua porta analógica, caso o sensor seja analógico, ou digital, caso o sensor seja digital. [15]



Figura 3-3 - Sensor de nível de fluido digital [16]

3.1.4 Sensor de Imagem

Qualquer sistema que possua como objetivo o visionamento em tempo real ou a obtenção de imagens de si próprio, utiliza uma câmara. Existem diversas câmaras para diversas aplicações. Algumas câmaras são desenhadas para funcionarem em situações de

baixa luminosidade, outras para situações de elevada luminosidade, algumas utilizam lentes de forma a conseguirem fotografar ou filmar elementos que estejam a uma grande distância, outras utilizam lentes de forma a fotografar ou filmar elementos que estejam a uma curta distância.

Um estilo de câmara possível de ser utilizada é a webcam, isto é, uma câmara que envia toda a informação captada via uma ligação USB (*Universal Serial Bus*). Estas câmaras são de baixo custo e relativamente pequena dimensão, sendo isso uma boa razão para serem escolhidas como sensor de imagem. A sua qualidade de imagem pode variar desde uma definição medíocre, a uma definição muito elevada, dependendo do tipo de sensor utilizado. Um exemplo deste tipo de câmara encontra-se na Figura 3-4.



Figura 3-4 - Câmara USB HP HD-2200

Outra câmara possível de ser utilizada é a Picamera, ilustrada na Figura 3-5. A Picamera foi recentemente atualizada, utilizando agora um sensor Sony IMX219 de 8 *megapixéis*, em vez do sensor OmniVision OV5647 de 5 *megapixéis* usado na versão anterior. Esta câmara pode ser usada tanto para filmar vídeos em alta resolução como para tirar fotografias. [17]



Figura 3-5 – Picamera – Módulo Câmara RaspberryPi V2 [17]

3.1.5 Sensor Ultrassónico

O sensor de ultrassom, ilustrado na Figura 3-6, é muito usado como meio para determinar a distância até um objeto através de um método sem contacto. Este funciona através do uso de ondas ultrassónicas.

De forma a que o sensor seja capaz de emitir e detetar ondas ultrassónicas, este possui um transmissor e um recetor. Tal como os nomes indicam, o transmissor é responsável por emitir ondas ultrassónicas e o recetor é responsável por detetar ondas ultrassónicas.



Figura 3-6 - Sensor ultrassónico [18]

O funcionamento do sensor ultrassónico consiste no envio de uma onda ultrassónica, por parte do transmissor, no sentido do objeto até à qual a distância pretende ser descoberta. A onda irá posteriormente refletir no objeto em causa e se redirecionar de volta para o sensor. Dado que o sensor possui também um recetor, é capaz de detetar a onda refletida. Tal procedimento está exemplificado na Figura 3-7. [18]

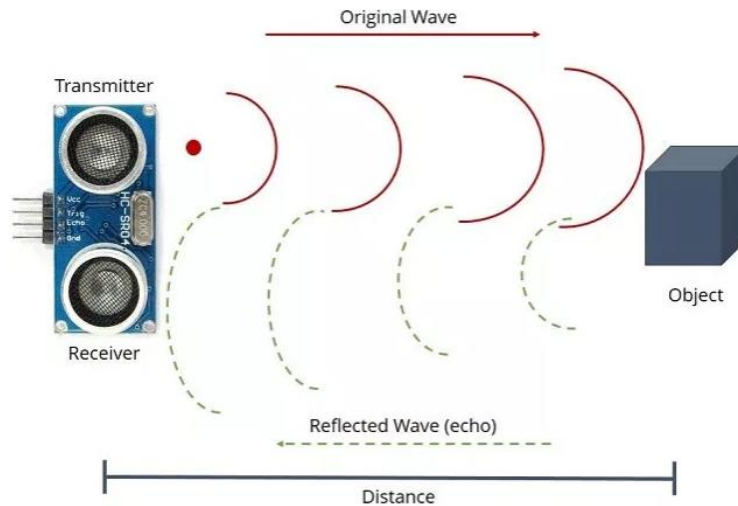


Figura 3-7 - Esquema de funcionamento do sensor ultrassônico [18]

Dado que a velocidade do som no ar é conhecida, através da aplicação da fórmula da velocidade é possível detectar a distância exata ao objeto em questão.

$$v = \frac{d}{T}$$

Sabendo que $v = 334$ (m/s) e que $T = 0.5 \cdot \text{Tempo transmissão (s)}$, consegue-se determinar a distância d pretendida.

3.1.6 Sensor de Salinidade

Um sensor de salinidade, ilustrado na Figura 3-8, é usado para determinar a salinidade de uma solução. Para tal, este possui uma sonda que mede a quantidade de corrente elétrica que está presente na solução, pois uma solução que possua sal dissolvido conduz eletricidade melhor que uma solução que não possua sal dissolvido. Deste modo, quanto mais sal estiver dissolvido na solução, melhor esta vai conduzir eletricidade. O valor da salinidade é calculado utilizando o valor de condutividade medido. Este valor é posteriormente enviado para uma porta analógica, quando ligado a um controlador, para ser processado. [19]



Figura 3-8 - Sensor de salinidade [20]

3.2 Plataformas para controladores

Um controlador para um aquário de água salgada necessita de ser capaz de ler e interpretar valores lidos por diversos sensores. Como tal, necessita de utilizar um controlador que possua pinos de entrada e saída digitais, pinos de saída analógicos e um microcontrolador. Existem diversos controladores capazes de realizar tais ações, no entanto cada um deles está desenhado para ser utilizado numa situação específica.

3.2.1 Plataforma Arduino

A plataforma Arduino é uma plataforma *open source* [21] muito utilizada em diversos projetos devido à sua simplicidade de utilização, suficiente capacidade de processamento e elevado número de módulos existente. No entanto, existem diferentes formatos em que esta plataforma se apresenta, isto é, a plataforma pode ter mais ou menos capacidade de processamento, mais ou menos portas analógicas ou digitais e uma maior ou menor dimensão física.

3.2.1.1 *Arduino Uno Rev3*

O Arduino Uno, ilustrado na Figura 3-9, proporciona um bom equilíbrio entre dimensão física, número de portas analógicas e digitais existentes e capacidade de processamento.



Figura 3-9 - Arduino Uno [22]

A plataforma Arduino Uno baseia-se no microcontrolador ATmega328P. Contem 14 pinos digitais de entrada/saída (I/O) onde 6 destes podem ser usados como saídas PWM (*Pulse Width Modulation*), 6 pinos de entrada analógica, uma conexão USB, um botão de reinício, 6 pinos ICSP (*In Circuit Serial Programming*) header e uma conexão para fornecimento de energia. Tem um design que permite uma utilização imediata. Basta ligar o fornecimento de energia à placa e esta a um computador, carregar o código através do software do Arduino e esta está pronta a funcionar. [22] Na Tabela 3-1 estão identificadas as principais características técnicas do Arduino Uno.

Tabela 3-1 - Características técnicas do Arduino Uno [22]

| | |
|--------------------------------------|-------------------|
| Microcontrolador | ATmega328P |
| Tensão de funcionamento | 5V |
| Tensão de entrada (recomenda) | 7-12V |
| Tensão de entrada (limite) | 6-20V |
| Pinos digitais Entrada/Saída | 14 |
| Pinos de entrada analógicos | 6 |
| Tensão DC por pino I/O | 20mA |
| Tensão DC para o pino de 3.3V | 50mA |
| Memória Flash | 32KB (ATmega328P) |
| SRAM | 2KB (ATmega328P) |
| EEPROM | 1KB(ATmega328P) |
| Velocidade do relógio | 16MHz |

3.2.1.2 Arduino Micro

O Arduino Micro, ilustrado na Figura 3-10, possui uma capacidade de processamento igual à do Arduino Uno e um número de portas analógicas e digitais elevado, no entanto possui uma dimensão física muito inferior.



Figura 3-10 - Arduino Micro [23]

A plataforma Arduino Micro baseia-se no microcontrolador ATmega32U4. Esta possui 20 pinos digitais de entrada/saída (I/O), dos quais 7 podem funcionar como saídas PWN e 12 como entradas analógicas. Possui também, uma conexão micro USB, 6 pinos de ICSP *header* e um botão de reinício. Ao contrário do Arduino Uno, o Micro não possui uma conexão para fornecimento de energia. Este possui apenas uma conexão USB a qual é usada para carregar o programa gerado. Deste modo, após o carregamento do programa a plataforma deve ser alimentada diretamente através da porta USB. [23] Na Tabela 3-2 estão identificadas as principais características do Arduino Micro.

Tabela 3-2 - Características do Arduino Micro [23]

| | |
|---|--------------------|
| Microcontrolador | ATmega32U4 |
| Tensão de funcionamento | 5V |
| Tensão de entrada (recomenda) | 7-12V |
| Tensão de entrada (limite) | 6-20V |
| Pinos digitais I/O | 20 |
| Pinos de entrada analógicos | 7 |
| Tensão DC por pino Entrada/Saída | 20mA |
| Tensão DC para o pino de 3.3V | 50mA |
| Memória Flash | 32KB (ATmega32U4) |
| SRAM | 2.5KB (ATmega32U4) |
| EEPROM | 1KB(ATmega32U4) |
| Velocidade do relógio | 16MHz |

3.2.1.3 Arduino Mega

O Arduino mega, ilustrado na Figura 3-11, possui a mesma capacidade de processamento do Arduino Uno e Nano, no entanto possui um número de portas analógicas e digitais muito mais elevado.

Tal diferença no número de portas analógicas e, especialmente, digitais tornam o Arduino Mega mais adequado para ser usado em aplicações de controle de diversos elementos em simultâneo, sendo interpretados como elementos, sensores ou módulos com diversas funcionalidades.



Figura 3-11 - Arduino Mega 2560 VER3 [24]

O Arduino Mega 2560 é uma placa de microcontrolador baseada no ATmega2560. Possui 54 pinos de entrada/saída (I/O), dos quais 15 podem funcionar como saídas PWM, 16 entradas analógicas, 4 portas série UART, um oscilador de cristal de 16 MHz, uma conexão USB, uma ficha de alimentação elétrica, um ICSP *header* e um botão de reinício. [24] Na Tabela 3-3 estão identificadas as principais características do Arduino Mega 2560 VER3.

Tabela 3-3 - Principais características do Arduino Mega 2560

| | |
|--------------------------------------|------------|
| Microcontrolador | ATmega2560 |
| Tensão de funcionamento | 5V |
| Tensão de entrada (recomenda) | 7-12V |
| Tensão de entrada (limite) | 6-20V |
| Pinos digitais I/O | 54 |

| | |
|---|--------------------|
| Pinos de entrada analógicos | 16 |
| Tensão DC por pino Entrada/Saída | 20mA |
| Tensão DC para o pino de 3.3V | 50mA |
| Memória Flash | 256KB (ATmega2560) |
| SRAM | 8KB (ATmega2560) |
| EEPROM | 4KB(ATmega2560) |
| Velocidade do relógio | 16MHz |

3.2.2 Plataforma RaspberryPi 3 Modelo B+

O RaspberryPi, ilustrado na Figura 3-12, é um computador de baixo custo originalmente desenhado para a educação [25]. No entanto, devido à sua capacidade, baixo custo, pequena dimensão física e maior poder de computação que o Arduino, este começou a ser utilizado também por entusiastas em diversos e variados projetos.



Figura 3-12 - RaspberryPi 3 Modelo B+ [26]

Esta plataforma possui um sistema operativo sem qualquer custo associado, o *Raspbian*. É alimentada por uma conexão USB e possui conexões para um rato, teclado, ecrã HDMI (*High Definition Media Interface*), CSI (*Camera Serial Interface*), USB e 40 pinos GPIO (*General Porpuse Input Output*). Pode ser usada através de uma interface própria ou através de acessos remotos. [26] Na Tabela 3-4 estão representadas as principais características do RaspberryPi 3 Modelo B+.

Tabela 3-4 - Características RaspberryPi 3 Modelo B+ [27]

| | |
|--------------------------|--|
| Processador | Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz |
| Memória | 1GB LPDDR2 SDRAM |
| Conexão | <ul style="list-style-type: none"> • 1 x Ethernet • 1 x CSI • 1 x HDMI • 4 x USB 2.0 |
| Acesso | 40 pinos GPIO |
| Tensão de entrada | <ul style="list-style-type: none"> • 5V/2.5A DC (<i>Direct Current</i>) via conector micro USB |
| Ambiente | Temperatura de funcionamento: 0-50°C |

3.3 Métodos de comunicação utilizados por controladores de água salgada

Um controlador de água salgada necessita de comunicar entre todos os seus elementos, sejam eles sensores, módulos ou controladores. Para tal, estes podem utilizar diversos métodos, desde métodos sem fios a métodos baseados em ligações físicas.

3.3.1 Comunicação Série

Uma das várias comunicações utilizadas é a comunicação série. Este tipo de comunicação pode ser utilizada através de diferentes protocolos. Dois dos principais protocolos são o UART e o I2C.

3.3.1.1 Protocolo UART

O UART (Universal Asynchronous Receiver Transmitter) utiliza uma comunicação assíncrona, o que significa que numa comunicação entre dois dispositivos ambos necessitam de possuir o mesmo ritmo de transmissão de data. Tal noção é importante, pois na comunicação UART não é usado um *clock* em comum pelos vários dispositivos.

Através do protocolo UART é possível de se obter uma comunicação intitulada de *full-duplex*. Neste tipo de comunicação existe um transmissor, TX, e um recetor, RX,

como representado na Figura 3-13. Este tipo de comunicação é caracterizada pela capacidade de todos os dispositivos serem capazes de comunicar entre eles, tanto do ponto de vista de transmissão, como de recepção. Tomando como exemplo um sistema composto por dois dispositivos, ambos podem comunicar entre eles, tanto no ponto de vista de transmissão como de recepção, em vez de um deles funcionar apenas como transmissor e o outro como recetor em todos os momentos de funcionamento do sistema.

O protocolo UART também pode ser usado na interligação de dois ou mais dispositivos. Pode-se dar o caso de existir mais de um emissor, ou de um recetor. [4] Tal situação é ilustrada na Figura 3-14 e na Figura 3-15.

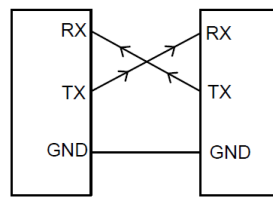


Figura 3-13 - Esquema de comunicação *full-duplex*

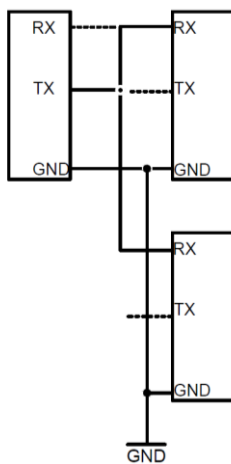


Figura 3-14 - Comunicação com 1 transmissor para 2 recetores

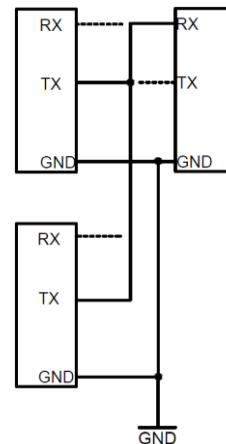


Figura 3-15 - Comunicação com 2 transmissores para 1 recetor

3.3.1.2 Protocolo I2C

O I2C (*Inter-Integrated Circuit*) utiliza uma comunicação síncrona, ilustrada na Figura 3-16, e utiliza um *clock* em comum por todos os dispositivos do sistema. Utiliza um bus de 2 cabos, o SCL (*Serial Clock Line*) e o SDA (*Serial Data Line*), ao qual cada dispositivo está ligado em paralelo. As comunicações são do tipo *master-slave*, no entanto pode existir mais do que um só *master*. Cada dispositivo possui um id pré-definido, de

forma a que o master saiba com quem comunicar. As linhas do bus necessitam de ter uma ligação do *pull-up* de forma a que estejam sempre a *HIGH*, tal deve-se ao facto dos dispositivos serem *active low*.

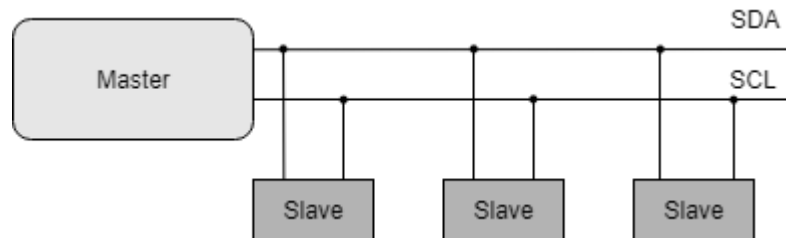


Figura 3-16 - Linha bus do protocolo I2C

O master inicia o *clock* e a comunicação com o *slave*, que por sua vez recebe o *clock* e responde quando contactado pelo master. A transmissão é realizada em sequências de 8 bits. No início é enviada uma condição especial de início, a qual é seguida do envio dos primeiros 8 bits que indicam o *address* do *slave* para o qual a informação irá ser enviada. Esta troca de informação inicial termina com o envio do *acknowledge* por parte do *slave*. Após o *acknowledge* são enviados outros 8 bits, mas neste caso com informação relativa ao *address* dos registos internos do *slave*. Seguidamente é enviado novamente um *acknowledge*, após o qual é enviada a informação. Este passo pode não ser sempre necessário dependendo do *slave* em causa. No final do envio, é transmitido novamente um *acknowledge* e, caso seja o final da transmissão, é enviado uma condição especial de término de comunicação. Na Figura 3-17 está ilustrado este funcionamento. [28]

| | | | | | | | | | |
|---|----|----|----|----|----|----|----|-----|---|
| S | A6 | A5 | A4 | A3 | A2 | A1 | A0 | R/W | A |
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | A |
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | A |
| | | | | | | | | | S |

Figura 3-17 - Envio da informação no I2C

3.3.2 Comunicação por Radiofrequência

A comunicação por radiofrequência é outra comunicação utilizada. Esta funciona de uma forma muito semelhante à comunicação série, sendo a principal diferença o facto da informação ser enviada através de ondas rádio, em vez de cabos físicos. Para tal, possui como base de funcionamento uma comunicação entre dois módulos, um transmissor (TX) e um recetor (RX). O módulo transmissor recebe as informações a enviar através de uma

comunicação direta com o microprocessador, enviando-as por RF. As informações são posteriormente recebidas pelo módulo recetor. Na Figura 3-18 está representado um esquema de funcionamento do envio de informação entre dois dispositivos usando radio-frequência.

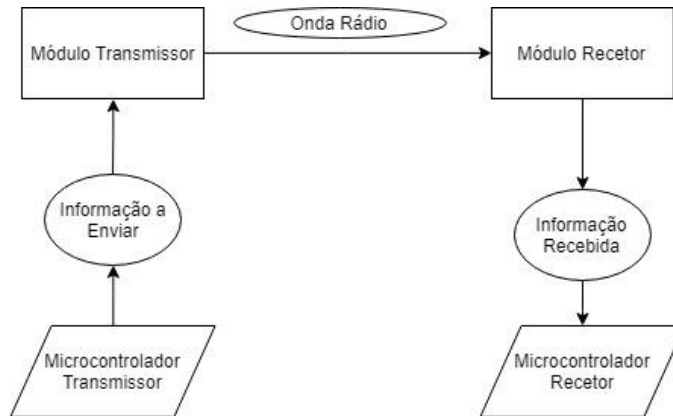


Figura 3-18 - Esquema representativo do funcionamento da comunicação radiofrequência

Neste tipo de comunicação existem diversos tipos de módulos, módulos que apenas funcionam como emissores, módulos que apenas funcionam como recetores e módulos que funcionam como recetores e emissores. Estes últimos denominam-se de *transceiver*. Na Figura 3-19 estão representados um módulo transmissor e recetor.



Figura 3-19 - Módulos recetor e emissor de RF [29]

Caso os módulos usados sejam *transceivers*, um mesmo módulo pode ser usado como recetor ou transmissor, o que permite uma constante troca de informação entre diferentes entidades sem a necessidade da utilização de um grande número de módulos

transmissores e recetores, o que seria o caso num cenário em que estes módulos não sejam utilizados. [30]

Na Figura 3-20 está representado um módulo transceiver. Na instalação de um transceiver necessitam de ser indicados no software quais as portas ligadas aos pinos CE (*chip enable*) e CSN (*chip select not*), sendo o CE usado para controlar o envio e a receção de informação e o CSN é o pino responsável pelo barramento SPI (*Serial peripheral interface*), que é o barramento responsável pela troca de informação entre dispositivos. O CSN funciona como *active low*, isto é, está ativo quando ler *low*.



Figura 3-20 - Módulo transceiver de RF [30]

Um módulo pode ser usado para enviar informações para um, ou mais, módulos e esses módulos podem enviar uma resposta não só para o módulo que enviou a informação original, mas também para qualquer outro módulo. [31]

3.3.3 Comunicação por Wi-Fi

O Wi-Fi permite uma comunicação sem fios entre dois ou mais elementos. Este utiliza ondas rádio para providenciar uma conectividade de rede, isto é, uma zona na qual vários elementos se conseguem ligar e interagir, como ilustrado na Figura 3-21.



Figura 3-21 - Ilustração do funcionamento de uma rede Wi-Fi [32] – Adaptado

Uma rede é formada através do uso de um adaptador sem fios, um *wireless router*, ilustrado na Figura 3-22, para criar um *hotspot*, isto é, uma zona próxima de um *wireless router* ligada à rede, que permite aos seus utilizadores se conectarem à rede em causa e utilizar todos os seus serviços.



Figura 3-22 - *Wireless router TP-LINK* [33]

A transmissão de informação na rede é feita via ondas rádio. Para tal, cada elemento necessita de possuir um adaptador *wireless*, de forma a que seja capaz de transformar a informação enviada num sinal de rádio e posteriormente enviá-la, através do uso de uma antena. Este sinal é enviado para um *router*, que por sua vez funciona como um *decoder*, isto é, retira a informação contida nas ondas rádio e envia-a para o destinatário em causa. [34]



Solução Proposta

Neste capítulo são inicialmente dados a conhecer mais em detalhe os problemas e necessidades de um sistema de água salgada. Seguidamente, é apresentada a estrutura do controlador correspondente à solução proposta, tal como o seu caso de uso. Por fim é apresentada a estrutura e funcionamento dos diversos sub-controladores do controlador relativo à solução proposta.

4.1 Estrutura geral

Um sistema de água salgada requer o uso de diversos subsistemas de manutenção de forma a que este consiga manter um estado de funcionamento que possibilite o correto funcionamento do ecossistema contido nele. Estes subsistemas são responsáveis por funções como a filtragem de impurezas da água, a oxigenação e movimentação da água, o aquecimento da água, a iluminação adequada do sistema, entre outros. Dado que um sistema por si só não é capaz de controlar todos estes elementos, existe a necessidade para o uso de um controlador.

O controlador proposto a ser desenvolvido possui como principal função a administração do funcionamento de todos os subsistemas do aquário, de forma a que o seu funcionamento seja o melhor possível perante as condições do meio em que este se encontra e perante os valores de vários parâmetros medidos no próprio aquário. De forma a que não seja complicado de instalar, o controlador divide-se em quatro sub-controladores, um Controlador Principal-*Master*, dois Controladores Secundários, o *Slave_Sensores* e o *Slave_Relês*, e um Servidor de Internet. A comunicação entre o controlador principal e os secundários é realizada por um meio de comunicação sem fios, de forma a que estes sejam mais fáceis de instalar e que seja minimizada a probabilidade de algum

deles se danificar pela água, pois desta forma apenas o controlador secundário responsável por ler valores da água necessita de estar perto desta. Por outro lado, a comunicação entre o Controlador Principal-*Master* e o Servidor de Internet é realizada por cabo, pois não existe necessidade de estes estarem separados fisicamente. Desta forma, ambos partilham a mesma caixa rígida exterior, diminuindo o número de sub-controladores necessários de instalar, o que facilita a instalação.

De forma a que certas tarefas de manutenção do sistema sejam mais cómodas e mais rápidas, o controlador possui um Modo Limpeza. Este modo desliga todos os elementos necessários para que seja feita a limpeza e manutenção do sistema. Após esta terminar, o utilizador apenas necessita de o desativar e o aquário volta ao seu estado anterior de funcionamento. Outros modos de melhoramento e simplicidade da interação com os subsistemas de manutenção do sistema são existentes no controlador, tais como a possibilidade de ligar e desligar qualquer subsistema individualmente ou ligar e desligar o sistema por completo. Neste cenário existe memória dos estados do sistema antes de este ser desligado, de forma a que ao ser novamente ligado, volte ao estado de funcionamento em que se encontrava.

O controlador é capaz de manter um estado de funcionamento ideal num ecossistema autonomamente, através do controlo de diversos relês, aos quais se encontram ligados diversos aparelhos de manutenção do sistema. O controlo destes relês tem como base valores lidos por diversos sensores e módulos instalados no sistema. O controlador possibilita também aos seus utilizadores um acesso remoto a todas as funções que podem ser realizadas sob o sistema e a todas as informações lidas pelos diversos sensores instalados neste, através de uma interface web. Estas informações são também guardadas numa base de dados.

Através da interface web é também possível visualizar em tempo real o sistema, através da transmissão de imagens da câmara que este possui instalada. O controlador disponibiliza também na sua interface web informações em tempo real relativas aos valores lidos pelos diversos sensores e módulos e aos estados de ativação de cada um dos diversos relês utilizados para controlar o sistema. É também feita uma análise dos valores lidos pelos diversos sensores e, caso haja alguma anomalia com um dos valores lidos, é enviado um e-mail ao utilizador a avisar do sucedido, de forma a que sejam minimizados os danos causados ao ecossistema.

Na Figura 4-1 é representada a configuração geral de todos os constituintes do controlador.

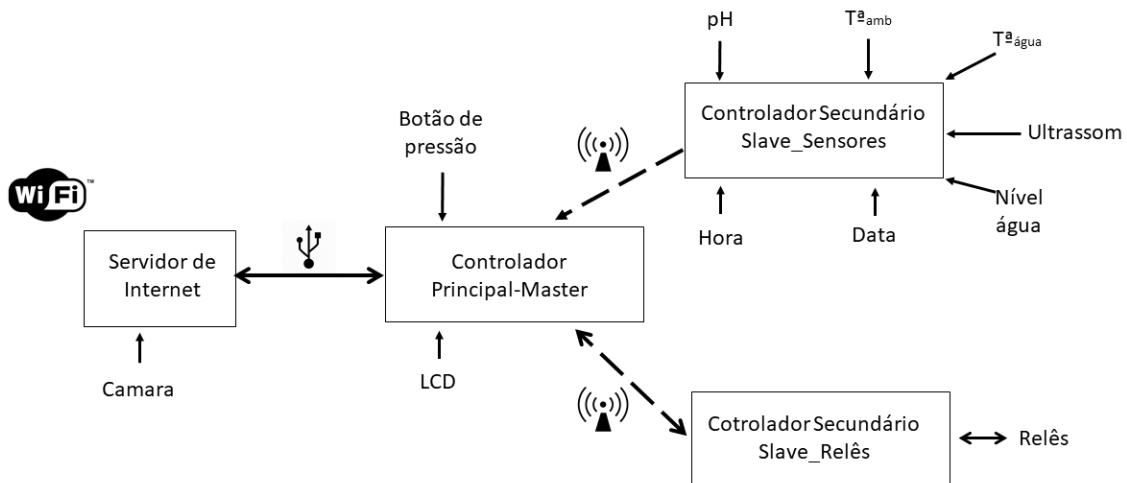


Figura 4-1 - Configuração geral do controlador

A Figura 4-1 ilustra 4 principais sub-controladores, um controlador principal-*Master*, dois controladores secundários e um servidor de internet. Cada sub-controlador possui pelo menos um sensor ou módulo ligado a ele próprio.

Os sensores ou módulos são ligados aos respectivos sub-controladores através de uma ligação física. O mesmo tipo de ligação é usada na comunicação entre o controlador principal-*Master* e o servidor de internet, sendo o método de comunicação usado o de comunicação série, pois é um método de comunicação de simples aplicação e fiável. No entanto, a comunicação entre o controlador principal-*Master* e os controladores secundários é efetuada via radiofrequência, de forma a facilitar o processo de instalação do controlador num sistema de água salgada. Ao não ser necessária a utilização de ligações físicas entre os sub-controladores, estes podem ser colocados em locais de mais fácil acesso para o utilizador, o que facilita a sua instalação

O servidor de internet utiliza um protocolo de comunicação série na sua ligação ao controlador principal-*Master* e um protocolo de ligação sem fios a uma rede Wi-Fi, ligação essa que possibilita o acesso remoto a todos os valores e funcionalidades do sistema através de uma interface web desenvolvida.

O funcionamento conjunto dos diversos elementos gera o comportamento desejado do controlador. Na Figura 4-2 pode-se observar o caso de uso do sistema.

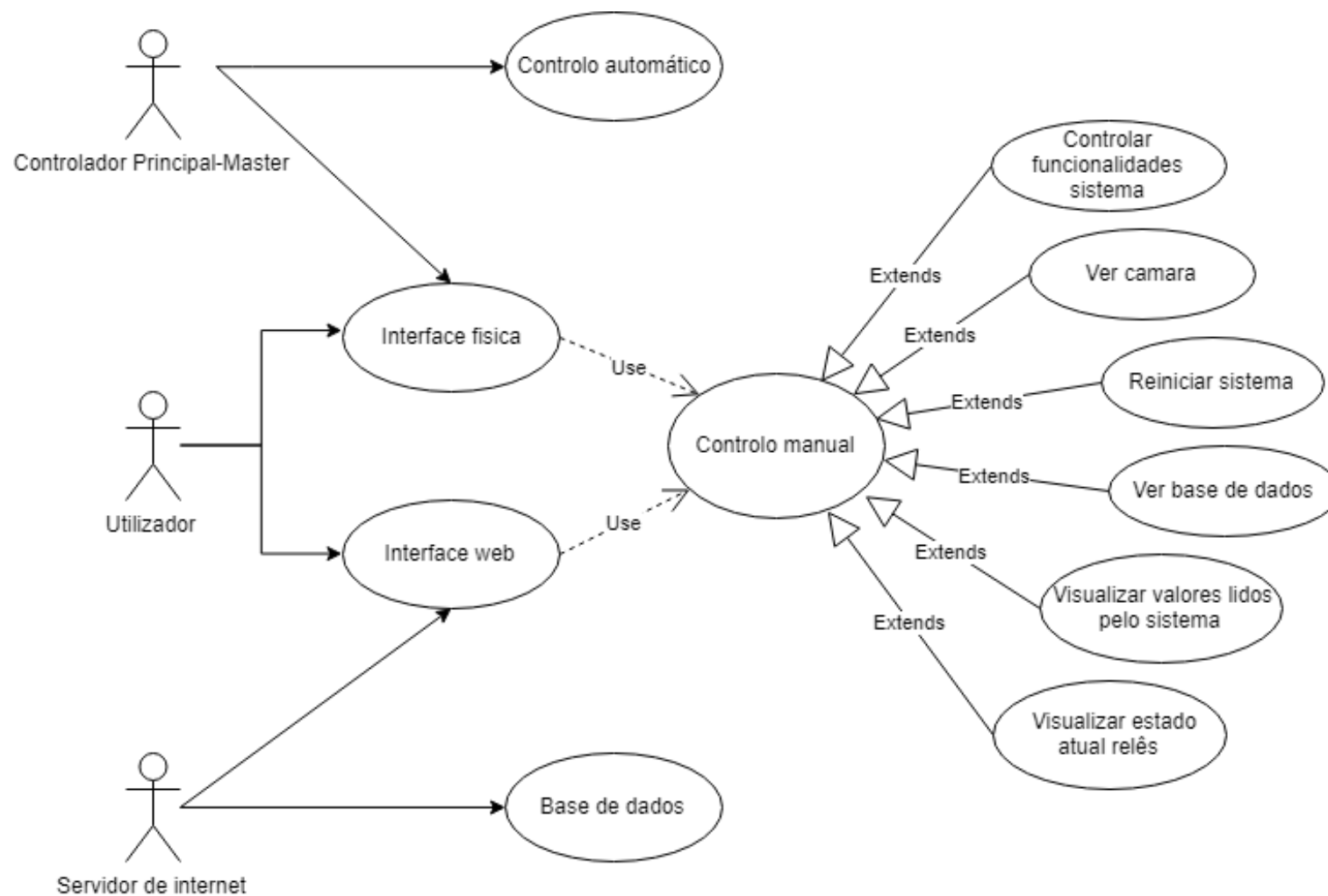


Figura 4-2 - Caso de uso do controlador

4.1.1 Controlador Principal-Master

O controlador principal-Master funciona como um *master*, isto é, recebe informações vindas de um segundo controlador com funções mais limitadas que as dele, ou seja, um *slave*. [3]

O controlador principal-Master possui quatro principais funções no sistema, processar a informação recebida de ambos os controladores secundários, enviar informações para o controlador secundário *Slave_Relês*, enviar informações para o servidor de internet e manter o funcionamento desejado do sistema de forma a não perturbar o ecossistema.

O processamento da informação recebida dos controladores secundários realiza um papel importante no mantimento do funcionamento desejado do sistema. O controlador principal-Master recebe os valores lidos por todos os sensores e módulos instalados no controlador secundário *Slave_Sensores* sob a forma de uma estrutura e, através de um algoritmo desenvolvido no desenrolar desta dissertação, utiliza-os de forma a tomar decisões relativas aos vários aparelhos de manutenção e mantimento do ecossistema. Decisões essas que são posteriormente enviadas para o controlador secundário *Slave_Relês*, também sob a forma de uma estrutura, de forma a que este as aplique nos diversos relês de controlo do sistema. De forma a que seja possível detetar falhas no cumprimento destas decisões, o controlador principal-Master recebe informações, sob a forma de uma estrutura, do controlador secundário *Slave_Relês* relativas aos estados atuais dos relês. Desta forma, o controlador principal-Master consegue detetar se algum dos relês de controlo do sistema não cumpriu a decisão que lhe foi imposta, o que pode significar que este possui uma avaria.

O envio de informações para o servidor de internet é outra das principais funções do controlador principal-Master. As informações enviadas são relativas aos valores lidos pelos diversos sensores e módulos instalados no controlador secundário *Slave_Sensores* e ao estado atual dos diversos relês de controlo do sistema. Adicionalmente, são enviadas informações relativas a erros detetados nalgum ou nalguns dos relês de controlo do sistema, para que o utilizador seja posteriormente informado.

De forma a que o sistema seja capaz de garantir um correto funcionamento do ecossistema, este necessita de analisar diversos parâmetros e realizar diferentes funções. Como já foi referido anteriormente, este algoritmo utiliza os valores recebidos do controlador secundário *Slave_Sensores* para garantir um correto funcionamento do ecossistema. No entanto, estes não são os únicos fatores a considerar pelo controlador. Este tem de ter

também em consideração o tipo de função que o sistema se encontra a realizar. Se o sistema se encontrar, por exemplo, em modo de limpeza o controlador não pode enviar de- cisões relativas ao acionamento de certos motores, como por exemplo o de reposição de água evaporada ou do escumador, por razões de segurança tanto para o ecossistema como para o utilizador que se encontra a realizar funções de manutenção do sistema. As diversas funções em que o sistema se pode encontrar são escolhidas pelo utilizador. Este pode escolhê-las através de 3 botões de pressão situados no controlador principal-*Master* ou através da interface web associada ao servidor de internet. No caso do uso dos botões de pressão o utilizador utiliza o LCD incorporado no controlador principal-*Master*, através do qual são expostas informações relativas ao sistema, para escolher o tipo de função que pretende que o sistema realize. No caso da interface web, o servidor de internet envia um caracter para o controlador principal-*Master*, através da ligação série que existe entre am- bos, indicando que tipo de função deve ser realizada. O controlador principal-*Master* irá depois identificar o tipo de função a executar através do seu algoritmo de funcionamento.

É importante realçar que os métodos de comunicação utilizados tanto no envio como na receção de dados no controlador principal-*Master* diferem. No caso das comu- nicações entre o controlador principal-*Master* e os controladores secundários o método utilizado é a comunicação por radiofrequência. No entanto, na comunicação entre o con- trolador principal-*Master* e o servidor de internet a comunicação utilizada é a comunica- ção série. Tal é devido à necessidade de diferentes controladores necessitarem de estar situados em diferentes locais. O controlador secundário *Slave_Relês* necessita de ser co- locado junto aos cabos de ligação dos diversos elementos do sistema. No entanto, o con- trolador secundário *Slave_Sensores* necessita de estar situado junto ao aquário de filtra- ção, pois este possui diversos sensores e módulos dos quais alguns necessitam de estar em contacto direto com a água do sistema. Este deve ser ligado ao aquário de filtração e não ao aquário principal, de forma a que não existam cabos visíveis, o que torna o sistema mais limpo visivelmente. Já o controlador principal-*Master* e o servidor de internet, que possuem uma ligação física entre eles, podem ser colocados onde o utilizador pretender, visto que estes não necessitam de estar em contacto direto com nenhum outro elemento do sistema. A única ligação que estes possuem é com os controladores secundários, mas como esta é feita por um método sem fios, estes não necessitam de estar perto deles.

Na Figura 4-3 encontra-se o diagrama comportamental do controlador principal-*Master*.

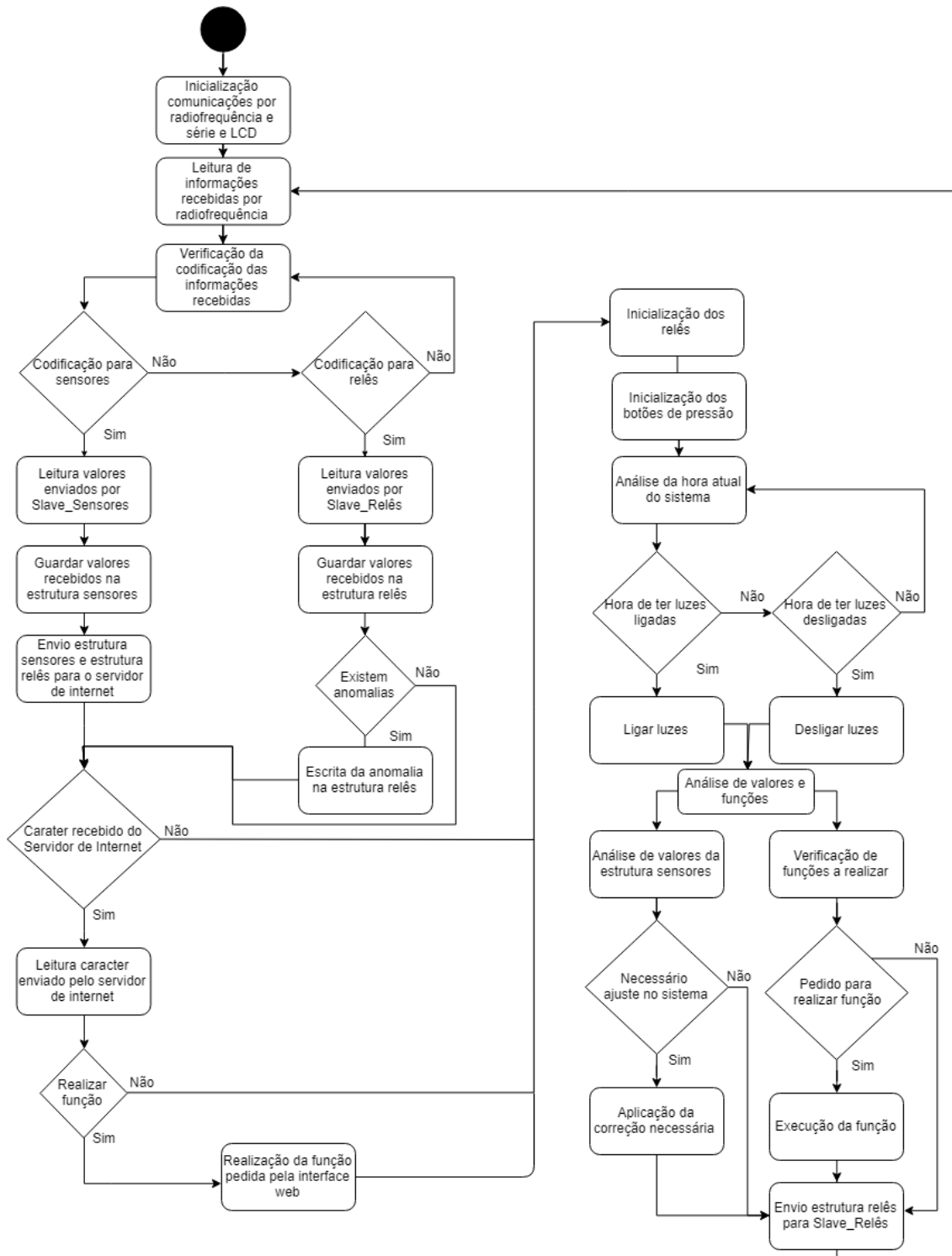


Figura 4-3 - Diagrama comportamental do controlador principal-Master

4.1.2 Controlador Secundário *Slave_Sensores*

O controlador secundário *Slave_Sensores* funciona como um *slave*, isto é, é um controlador com funções limitadas que tem como principal função enviar informações importantes para um outro controlador com mais funcionalidades, ou seja, um *master*. [3]

Este controlador possui duas principais funcionalidades, a leitura dos valores enviados pelos diversos sensores e módulos ligados a ele mesmo e o envio desses valores para o controlador principal-*Master*. Os diversos sensores usados encontram-se ligados diretamente ao *Slave_Sensores*, através de uma ligação física. Deste modo, toda a informação enviada pelos sensores é recebida por ele mesmo.

A informação enviada pelos sensores e módulos para o *Slave_Sensores* é processada a muito baixo nível, isto é, o *Slave_Sensores* não tem de processar a informação de forma a detetar se o sistema está a funcionar como desejado ou se existe alguma falha, este apenas tem de verificar se a informação recebida é do tipo esperado e se esta não é nula, pois pode ter-se dado o caso de ter havido um problema associado à leitura dos valores enviados pelos sensores e módulos. Caso haja algum problema associado aos valores lidos pelos sensores e módulos, o *Slave_Sensores* deve descartar a informação recebida. No entanto, caso a informação recebida tenha o formato correto e não seja nula, o *Slave_Sensores* guarda-a numa estrutura e envia-a para o controlador principal-*Master*.

O *Slave_Sensores* comunica unicamente com o controlador principal-*Master* numa comunicação de um só sentido, isto é, o *Slave_Sensores* envia valores para o controlador principal-*Master*, mas nunca recebe valores deste. A comunicação é realizada através de um método de comunicação sem fios, a radiofrequência. Ao ser usada uma comunicação sem fios para comunicar entre o controlador principal-*Master* e o controlador secundário *Slave_Sensores*, é gerada uma maior simplicidade de instalação do *Slave_Sensores*. É importante notar que este terá de ser colocado junto aos sistemas de manutenção do ecossistema, de forma a que seja possível a instalação dos sensores dentro de água sem que sejam visíveis cabos pendurados no aquário principal. Logo, ter um controlador que pode ser colocado em qualquer local do sistema é uma mais valia do ponto de vista de simplicidade e limpeza de instalação.

Na Figura 4-4 pode-se ver o diagrama comportamental do controlador secundário *Slave_Sensores*.

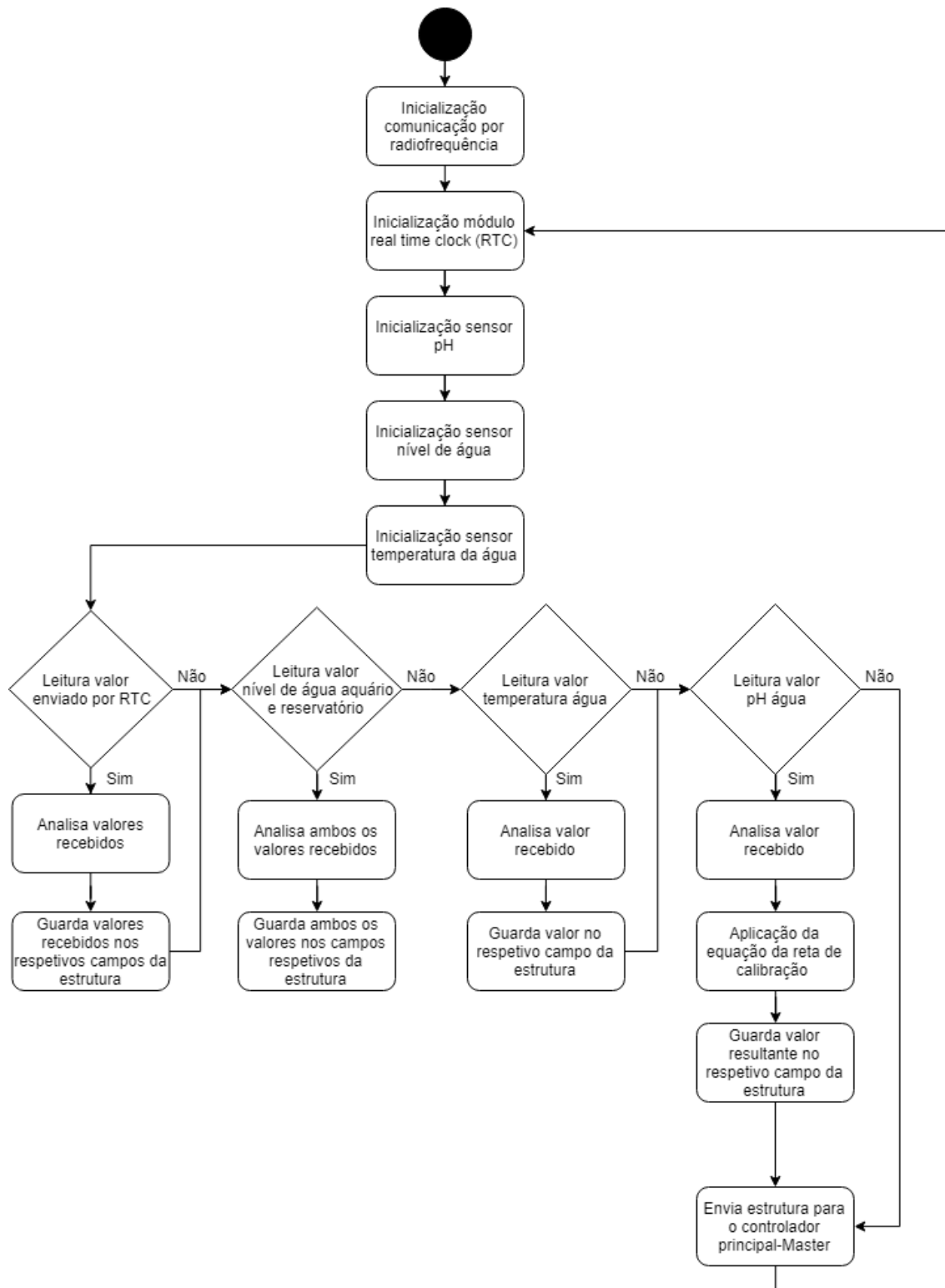


Figura 4-4 - Diagrama comportamental do controlador secundário *Slave_Sensores*

4.1.3 Controlador Secundário *Slave_Relês*

O controlador secundário *Slave_Relês*, tal como o *Slave_Sensores*, funciona como um *slave*, isto é, é um sub-controlador com funções limitadas que tem como principal função enviar informações importantes para um outro sub-controlador com mais funcionalidades, ou seja, um *master* [3].

O *Slave_Relês* tem duas principais funcionalidades, alterar os estados dos relês de controlo do sistema de acordo com as informações enviadas pelo controlador principal-*Master* e enviar os estados atuais no momento de envio, para o controlador principal-*Master*.

A alteração dos estados dos relês de controlo do sistema depende unicamente da informação que o *Slave_Relês* recebe. Este recebe, via radiofrequência, uma estrutura, enviada pelo controlador principal – *Master*, na qual se encontram os estados desejados para os relês de controlo do sistema. É importante notar que, como foi referido no subcapítulo 3.1.1, relativo ao controlador principal-*Master*, os estados recebidos pelo *Slave_Relês* resultam da aplicação do algoritmo de controlo existente no controlador principal-*Master*. Logo, estes são resultado da análise de todos os valores lidos pelos vários sensores e módulos, interações através dos botões de pressão do controlador principal-*Master* e interações através da interface web situada no servidor de internet.

A leitura dos estados atuais dos relês de controlo é uma outra função principal do controlador secundário *Slave_Relês*. Tal funcionalidade consiste em ler os estados de cada um dos relês de controlo e guardá-los numa estrutura correspondente. Esta estrutura é depois enviada para o controlador principal-*Master* e é fundamental para a identificação de erros de funcionamento por parte dos relês. É importante notar que é efetuada primeiro a alteração dos estados atuais dos relês para os estados enviados pelo controlador principal-*Master* e só depois são lidos os estados atuais dos relês de controlo e guardados na respetiva estrutura. Deste modo, se existir alguma diferença entre o estado atual de um relê de controlo e o estado objetivo enviado pelo controlador principal-*Master*, é identificável um problema com o relê em causa e o utilizador é posteriormente avisado. No entanto, esta tarefa não é realizada no *Slave_Relês*. Este apenas lê os valores atuais dos relês de controlo após aplicação da estrutura enviada pelo controlador principal-*Master* e os envia, sob a forma de uma estrutura, para o controlador principal-*Master*.

Na Figura 4-5 encontra-se o diagrama comportamental do controlador secundário *Slave_Relês*.

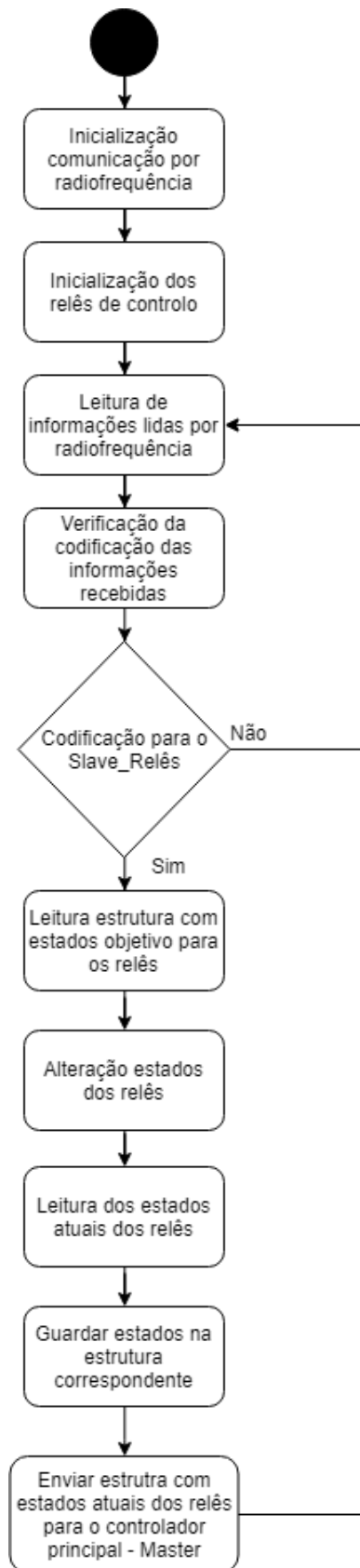


Figura 4-5 - Diagrama comportamental do controlador secundário *Slave_Relês*

4.1.4 Servidor de internet

O servidor de internet consiste numa placa programável ligada a uma rede Wi-Fi local que tem duas principais funcionalidades, comunicar com o controlador principal-*Master* e funcionar como servidor para uma interface web que possibilite ao utilizador ter um acesso remoto a todas as funcionalidades e parâmetros do sistema.

A comunicação com o controlador principal-*Master* é a razão pela qual é possível haver não só um controlo remoto do sistema, como também um acesso completo a todos os valores lidos no sistema. Esses valores ao serem recebidos pelo servidor de internet, são imediatamente gravados numa base de dados. O controlador tem de ser capaz de identificar o tipo de informação que recebe e de a colocar no campo correspondente da base de dados. Após serem gravados, são avaliados de forma a que sejam detetadas falhas no funcionamento do sistema. Caso seja detetada alguma falha é imediatamente enviado um e-mail para o utilizador com o intuito de o informar em relação à falha ocorrida, para que este a possa resolver. Caso haja mais do que uma falha, são enviados mais do que um e-mail.

De forma a que haja uma interface através da qual o utilizador possa visualizar os parâmetros e estados dos diversos relês de controlo mais atuais, visualizar um vídeo em direto ou controlar remotamente as diversas funcionalidades do sistema, tem de existir uma interface web. Para tal, o servidor de internet funciona como servidor para uma interface web. Para que seja possível dispor na interface web os valores mais recentes dos parâmetros e estados de relês de controlo, a interface web tem de se ligar à base de dados. Desta, a interface web retira os valores mais recentes de todas as entradas da base de dados e coloca-as em duas tabelas, uma relativa aos parâmetros do sistema e outra relativa aos estados dos diversos relês de controlo do sistema, tabelas essas que são posteriormente dispostas na interface web. Por outro lado, o vídeo em direto do sistema resulta da utilização de uma câmara USB ligada diretamente ao servidor de internet, que através da utilização de uma biblioteca própria, é possível emitir vídeo em direto, através da câmara, para a interface web. No entanto, para que este possa realizar ações de controlo de todas as funcionalidades do sistema, tem de ser utilizada uma comunicação no sentido do servidor de internet para o controlador principal-*Master*, de forma a que ao ser escolhida uma funcionalidade pelo utilizador na interface web, seja enviado um código para o controlador principal-*Master* que lhe indique qual a função a executar. Desta forma é proporcionado ao utilizador um controlo remoto a todas as funcionalidades do sistema.

Na Figura 4-6 podemos visualizar o diagrama comportamental do servidor de internet.

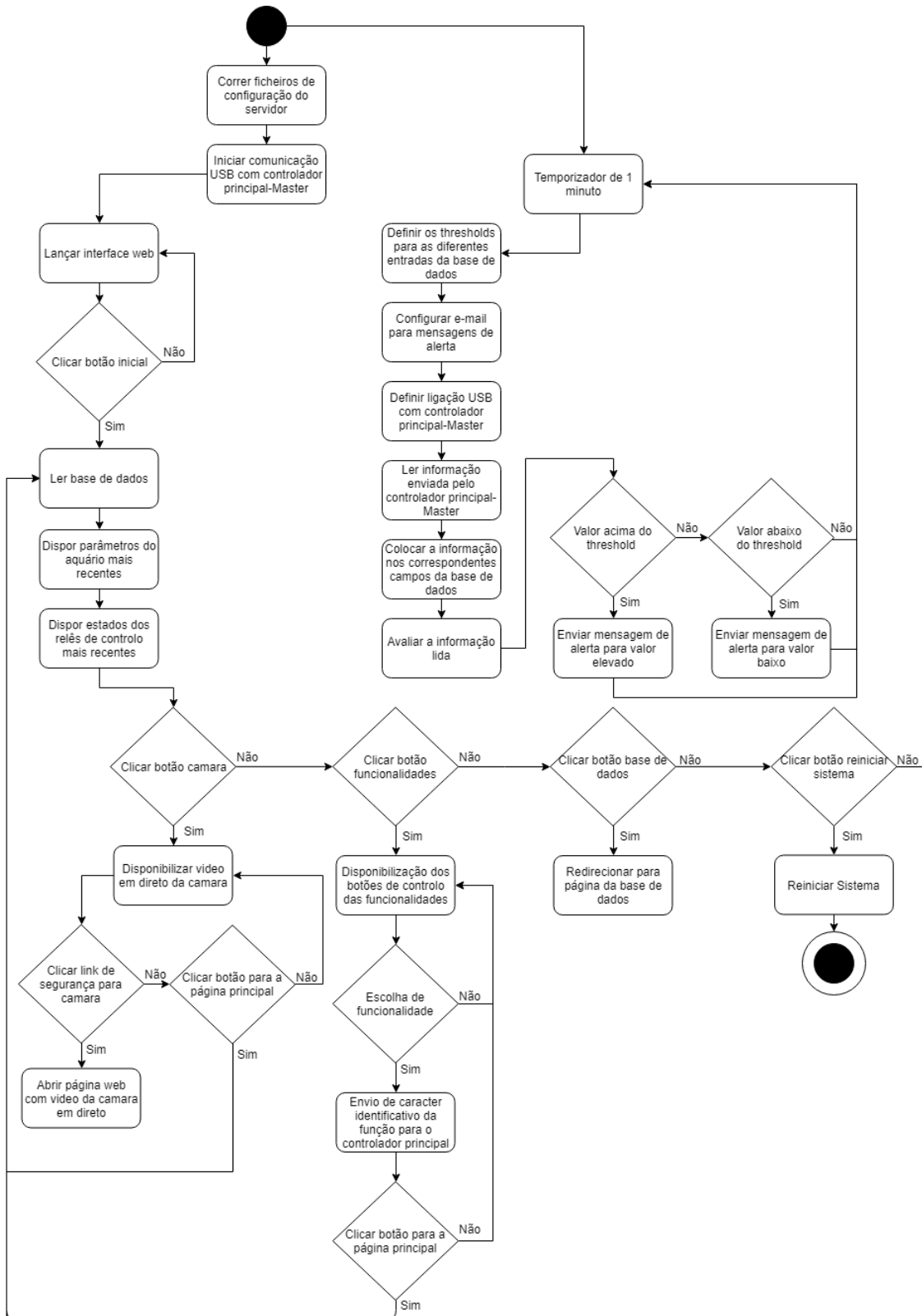


Figura 4-6 - Diagrama comportamental do servidor de internet

5

Componentes e Implementação da Solução Desenvolvida

Serve o presente capítulo para apresentar detalhadamente os diversos componentes de cada um dos sub-controladores do controlador relativo à solução desenvolvida. Adicionalmente, são apresentadas as ligações entre os diversos componentes que constituem cada um dos sub-controladores. Por fim, é apresentada uma explicação da implementação por detrás de cada um dos sub-controladores.

5.1 Controlador principal – *Master*

O controlador principal-*Master* é o principal componente do controlador. Este comunica com todos os outros componentes. Como tal, necessita de ser eficiente e de fácil utilização, mas ao mesmo tempo ser capaz de contribuir para um baixo custo final do controlador.

5.1.1 Componentes Controlador Principal-*Master*

De forma a garantir o funcionamento desejado, o controlador principal-*Master* utiliza um microcontrolador, um módulo LCD (*liquid crystal display*), botões de pressão e um módulo de radiofrequência.

5.1.1.1 Microcontrolador

É usado um microcontrolador no controlador principal-*Master* de forma a que todas as informações que este recebe sejam processadas juntamente com todas as informações que este possui, de forma a que sejam gerados resultados e ações desejadas.

O microcontrolador necessita de ser rápido de forma a que o processamento de informações em maior quantidade não torne o sistema lento, necessita de possuir pequenas dimensões de forma a não gerar um controlador final de grandes dimensões e necessita de ter um custo baixo, de forma a não gerar um controlador final com um custo elevado. Tendo estas considerações em causa é então usado como microcontrolador uma placa Arduino Uno, como exposta na Figura 3-9.

5.1.1.2 LCD e botões de pressão

Como referido no capítulo 3 o controlador principal oferece a possibilidade de ter as suas funcionalidades controláveis diretamente através deste, isto é, uma interface física, ou através de uma interface web. Para o primeiro caso são usados 3 botões de pressão que em conjunto com um LCD permitem a escolha de entre diversos modos de funcionamento.

Os botões de pressão utilizados estão expostos na Figura 5-1 e encontram-se dispostos em linha. Estes consistem em interruptores que fecham o circuito quando pressionados, permitindo assim que o microcontrolador detete o seu acionamento.

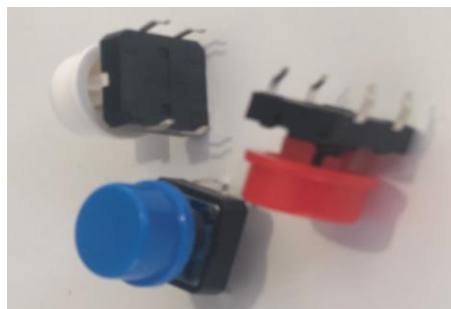


Figura 5-1 - Botões de pressão

O seu funcionamento é ilustrado no esquema da Figura 5-2. Estes permitem a passagem entre os diversos menus de escolha de funcionalidade do sistema e a respetiva ativação ou desativação da funcionalidade em causa.

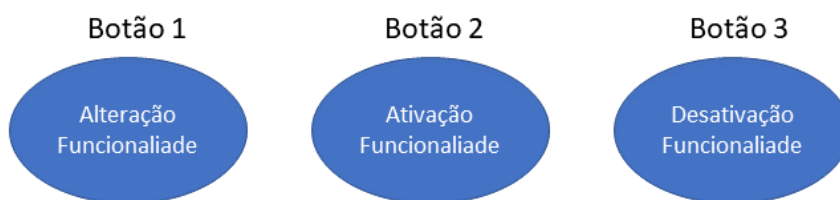


Figura 5-2 - Funções dos botões de pressão

Como referido anteriormente é utilizado um LCD para expor algumas informações relevantes ao sistema, como o pH da água, as horas do sistema e a temperatura da água.

O LCD utilizado possui 2 linhas e 16 colunas e está ilustrado na Figura 5-3. Como é possível observar, este possui diversos pinos que necessitam de ser ligados ao micro-controlador usado para garantir o seu correto funcionamento.

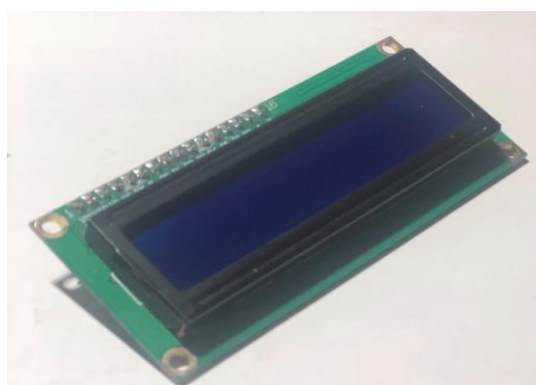


Figura 5-3 - Ecrã LCD

Ora, como a placa utilizada para o Controlador Principal-*Master* é um Arduino Uno e este não possui muitos pinos de controlo, não seria possível efetuar todas as ligações necessárias para a utilização do LCD em causa junto com os botões de pressão referidos anteriormente. Como tal, é utilizado um módulo de I2C, ilustrado na Figura 5-4, que quando ligado aos diversos pinos do LCD permite que este seja utilizado e completamente controlado através de apenas 2 pinos de sinal analógico e 2 pinos de alimentação.

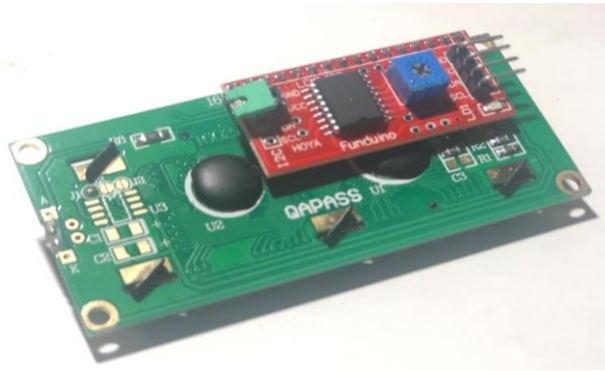


Figura 5-4 - Módulo de I2C ligado ao LCD

Após a instalação do módulo de I2C apenas existem 4 pinos que necessitam de ser ligados para garantir o correto funcionamento do LCD, como é possível visualizar na Figura 5-5.



Figura 5-5 - Ligações módulo I2C para LCD

Estes são ligados diretamente ao Arduino Uno nos pinos correspondentes a uma alimentação de 5V, GND e os pinos analógicos SDA e SCL. Deste modo sobram pinos suficientes para efetuar todas as restantes ligações necessárias.

5.1.1.3 Módulo de radiofrequência e comunicação Série

Como visto no capítulo 3 o Controlador Principal-*Master* utiliza duas diferentes formas de comunicação para comunicar com todos os outros elementos do controlador. Para comunicar com o Servidor de Internet utiliza comunicação série. Para tal, é utilizado um cabo USB para efetuar a ligação entre o microcontrolador do Controlador Principal-*Master* e o Servidor de Internet. No entanto, a comunicação entre o Controlador Principal-*Master* e os dois Controladores Secundários-*Slave* é efetuada via radiofrequência. Para tal, é utilizado o módulo de radiofrequência NRF24L01, ilustrado na Figura 3-20.

Este módulo é ligado diretamente ao Arduino Uno através de 5 pinos digitais e 2 portas de alimentação. As ligações deste encontram-se na Figura 5-20.

5.1.2 Ligações dos componentes do controlador principal-*Master*

O Controlador Principal-*Master* utiliza diversos elementos, como visto no capítulo 4.1.1. Todos estes elementos são utilizados de forma a gerar um comportamento desejado para o controlador. De forma a que tal comportamento seja possível, estes têm de ser instalados devidamente no sistema. A Figura 5-6 ilustra a ligação de todos os elementos do Controlador Principal-*Master*.

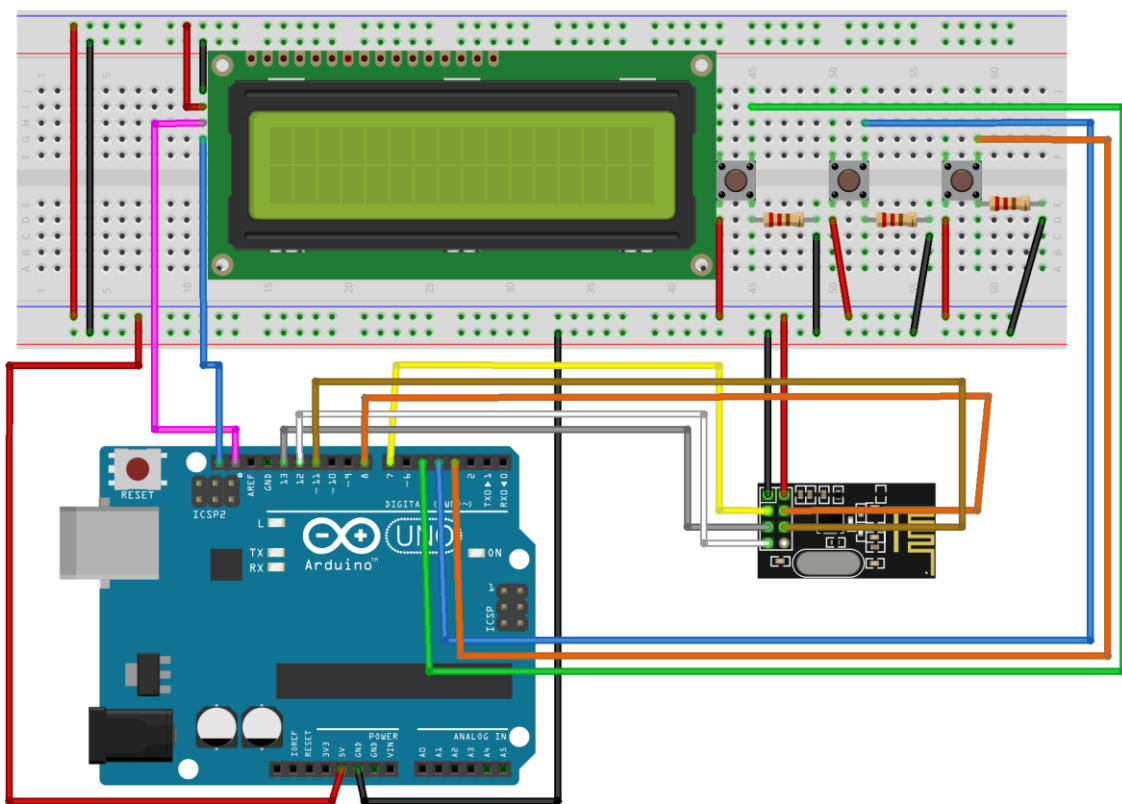


Figura 5-6 - Montagem Controlador Principal-*Master*

O Arduino Uno é o centro de toda a instalação, pois este controla todos os elementos instalados através das suas portas analógicas e digitais. De forma a facilitar a instalação é usada uma *breadboard*.

O LCD é ligado diretamente às portas analógicas SDA e SCL, de forma a receber informação relativa ao que dispôr no visor. Este é alimentado por 5V e GND.

Os botões de pressão necessitam de uma ligação de *pull-down*, realizada através da instalação de uma resistência de $1k\Omega$ entre o botão de pressão e o GND (*Ground*), de forma a que a linha que liga à porta digital do Arduino esteja sempre a *LOW*. Deste modo, quando o botão é pressionado a linha fica a *HIGH* e o Arduino deteta que o botão correspondente à porta em causa foi premido, podendo assim realizar a reação correspondente a esta ação.

O módulo NRF24L01 é ligado diretamente às portas analógicas 7, 8, 11, 12 e 13 e é alimentado por 3.3V e GND. Através da conexão às portas 7 e 8, que correspondem aos pinos CE e CSN, é possível enviar as informações desejadas por radiofrequência para qualquer destinatário que esteja a ouvir e a utilizar a encriptação correta.

5.1.3 Implementação

Os principais objetivos do Controlador Principal-*Master* são lidar com a informação recebida dos Controladores Secundários-*Slave*, realizar funções desejadas pelo utilizador ou necessárias para manter o correto funcionamento do sistema e comunicar com o Servidor de Internet. Esta comunicação é utilizada de forma a que o Controlador Principal-*Master* receba informações do Servidor de Internet relativas a funções requisitadas pelo utilizador através da interface web. Serve ainda para enviar informações relativas aos valores lidos pelos sensores e módulos e estados atuais dos reles de controlo do sistema.

5.1.3.1 Estruturas

Dado que o Controlador Principal-*Master* necessita de comunicar com diferentes Controladores Secundários-*Slave*, este necessita de utilizar diferentes estruturas. Uma estrutura é um conjunto de variáveis onde são guardadas informações lidas pelos controladores. Um exemplo de uma estrutura recebida pelo Controlador Principal-*Master* está exposta na Figura 5-7.

```

struct sensor_package
{
    float tempAgua, pH, temp;
    int sensoragua;
    char data[13];
    char hora[13];
};
typedef struct sensor_package s_package;
s_package s_data;

```

Figura 5-7 - Estrutura correspondente à informação lida pelo *Slave_Sensores*

Como é possível observar são usadas diferentes variáveis dentro da estrutura, sendo que cada uma das variáveis guarda o valor lido pelos vários sensores e módulos do controlador. O Controlador Principal-*Master* recebe constantemente estruturas enviadas pelos dois controladores secundários, de forma a que os seus valores estejam sempre atualizados.

5.1.3.2 *Receção e envio de informação por radiofrequência*

Como visto no ponto 4.1.3.1 o Controlador Principal-*Master* recebe constantemente estruturas com informação relativa aos diversos sensores e módulos dos controladores secundários. A comunicação entre estes controladores é feita por radiofrequência. No entanto, para que seja possível haver a troca de informações uma comunicação tem de ser configurada. Para tal, é utilizada uma biblioteca intitulada de RF24. Através desta, é possível utilizar funções já existentes para iniciar comunicações e enviar e receber informações via radiofrequência. De forma a que esta biblioteca possa ser utilizada, têm de ser realizadas duas configurações prévias, indicar as portas digitais correspondentes aos pinos CE e CSN do módulo de radiofrequência e definir um vetor de encriptação para a comunicação. A Figura 5-8 mostra esta configuração.

```

RF24 radio(7, 8);
const byte rxAddr[] = {00003, 00002, 00001};

```

Figura 5-8 - Configuração comunicação por radiofrequência

Após a configuração da comunicação, esta necessita de ser iniciada. Para tal, é utilizada a função *setup()* do Arduino. Esta função tem como principal característica apenas ser executada uma vez ao arranque da placa. Desta forma, é o local indicado para realizar

inicializações. A leitura de informações, que é realizada pela função *radio.read()*, segue a lógica do fluxograma da Figura 5-9 e é realizada na função *loop()* do Arduino, que tal como o seu nome indica realiza todo o código no seu interior de uma forma cíclica.

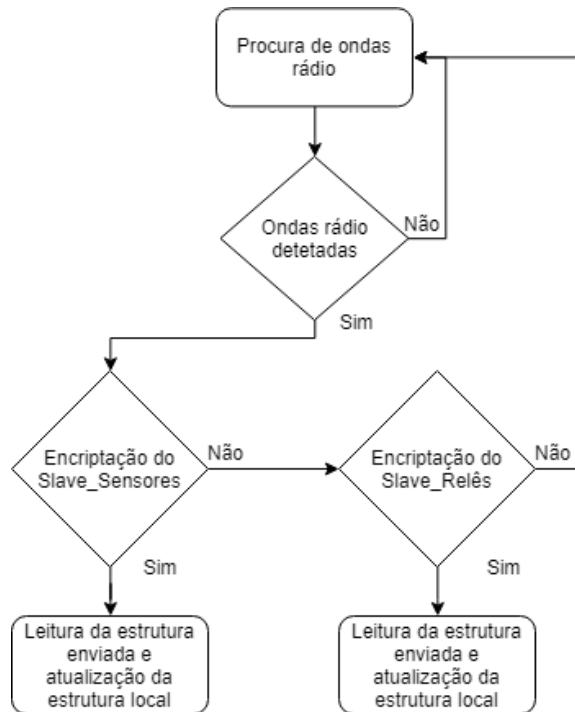


Figura 5-9 - Fluxograma da lógica de leitura de informações por radiofrequência

É importante notar que as funções de escrita, *radio.write()*, e leitura, *radio.read()*, requerem que seja alterado o modo de funcionamento do módulo, de modo leitura para modo escrita, previamente ao seu uso. Além disso, possuem a capacidade de comunicar apenas numa encriptação de entre as 3 definidas na variável *rxAddr* da Figura 5-8. Desta forma, apenas consegue receber a informação enviada o recetor que use a mesma encriptação usada para enviar.

O envio de informações via radiofrequência é realizada no final da função *loop()*, pois tem como principal função atualizar os estados dos relês de controlo do sistema. Para tal, é enviada a estrutura atualizada correspondente aos estados destino dos relês de controlo para o Controlador Secundário-*Slave_Relês*. Essa estrutura é atualizada no desenrolar do código correspondente à função *loop()* do Arduino. Neste, através da análise da informação recebida, vão sendo atualizados os estados destino dos vários relês de controlo à medida que são analisados. Desta forma é garantida a automação do sistema.

5.1.3.3 Receção e envio de informações via ligação série

A comunicação série é usada pelo Controlador Principal-*Master* de forma a comunicar com o Servidor de Internet. Para que seja possível ser utilizada esta, tal como a comunicação por radiofrequência, tem de ser inicializada. Para tal, é mais uma vez usada a função *setup()* do Arduino. Ao ser inicializada tem de ser indicado o ritmo de comunicação utilizado, que no caso do Controlador Principal-*Master* é 9600bps (*bits per second*).

Após a sua inicialização a comunicação série usufrui das estruturas atualizadas pela comunicação por radiofrequência para enviar os valores lidos pelos sensores e os estados mais atuais dos relés de controlo do sistema para o Servidor de Internet. Para tal, é utilizada uma das bibliotecas incluídas no Arduino, a biblioteca *Serial*.

De forma a que a informação seja facilmente identificável e separável à chegada ao Servidor de Internet, são utilizados espaçamentos específicos entre cada valor enviado. Após o último valor enviado, é utilizado o método de inserção de parágrafo do modo *print* da biblioteca *Serial*, permitindo assim que o Servidor de Internet detete o final da transmissão atual de informação.

O Servidor de Internet também envia informações para o Controlador Principal-*Master*. Para tal, utiliza a ligação série estabelecida entre os dois. A lógica de receção destas informações está ilustrada no fluxograma da Figura 5-10.

A informação enviada pelo Servidor de Internet utiliza uma codificação de forma a que o Controlador Principal-*Master* saiba que funcionalidade o utilizador pediu através da interface web. Esta codificação está exposta na Tabela 5-1.

Tabela 5-1 - Codificação informações enviadas pelo Servidor de Internet

| Codificação | a / b | c / d | e / f | g / h |
|--------------------------|---------|---------------------------|-------------|---------------|
| Função de Ligar/Desligar | Sistema | Escumador | Termostato | Modo Limpeza |
| Codificação | i / j | k / l | m / n | o / p |
| Função de Ligar/Desligar | Luzes | Reposição automática água | Luz filtros | Motor retorno |

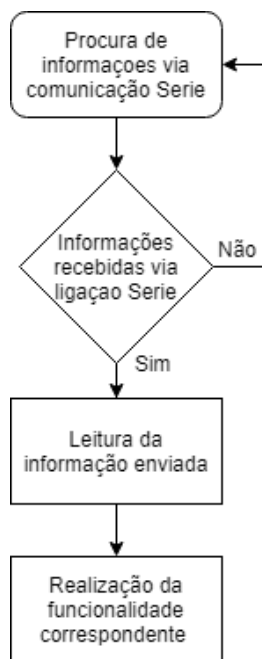


Figura 5-10 - Fluxograma da lógica de recepção de informações enviadas pelo Servidor de Internet

5.1.3.4 Controlo relês

O controlo dos relês permite que existam alterações no comportamento do sistema. Estes podem ser alterados a pedido do utilizador ou através de lógicas de controlo implementadas. A cada relê está ligado um diferente elemento do sistema, como é possível visualizar na Tabela 5-2.

Tabela 5-2 - Ligações relês

| | | | | |
|----------|-----------|------------------------------|-----------------------|--------------------------|
| Relês | 1 | 2 | 3 | 4 |
| Elemento | Led Azul | Led Branco | Luzes fortes | Motor retorno |
| Relês | 5 | 6 | 7 | 8 |
| Elemento | Escumador | Motor movimento cima | Motor movimento baixo | Termostato |
| Relês | 9 | 10 | 11 | 12 |
| Elemento | Ventoíhas | Bomba retorno água evaporada | Luzes filtros | Termostato de emergência |

Quando o Controlador Principal-*Master* é iniciado este executa um código que inicia no modo ativo os relês 4, 5, 6, 7 e 8. De seguida, é lido na estrutura correspondente o valor mais recente das horas do sistema. Através dessa informação, são testadas várias condições de lógica para verificar que luzes devem ser ligadas ou desligadas, sabendo que estas seguem a lógica da Tabela 5-3. O seu estado é atualizado na estrutura correspondente aos estados destino atualizados dos relês de controlo do sistema.

Tabela 5-3 - Horas a que as luzes de ligam e desligam

| Hora | 13:00 | 13:30 | 14:00 | 22:00 | 22:30 | 23:00 |
|--------------|-------|-------|-------|---------|---------|---------|
| Luz | | | | | | |
| Led azul | Liga | | | | | Desliga |
| Led Branco | | Liga | | | Desliga | |
| Luzes fortes | | | Liga | Desliga | | |

Através da análise da estrutura atualizada pelo Controlador Secundário-*Slave_Sensores*, é possível para o Controlador Principal-*Master* detetar se o nível de água do aquário é baixo. Caso o seja, é atualizado o estado do relê de controlo, na estrutura correspondente aos estados destino atualizados dos relês de controlo, correspondente à bomba de retorno de água evaporada, para ativo. Desta forma, quando no final do *loop()* a estrutura atualizada dos estados destino dos relês for enviada para o Controlador Secundário-*Slave_Relês*, a bomba de retorno de água evaporada será ativada. Posteriormente, quando a informação da estrutura atualizada pelo Controlador Secundário-*Slave_Sensores* possuir informação de que o nível de água já não se encontra baixo, o relê de controlo da bomba de retorno de água evaporada, na estrutura correspondente aos estados destino atualizados dos relês de controlo, será desligado. Logo, ao ser enviado o novo estado para o Controlador Secundário-*Slave_Relês* a bomba será desligada. É importante notar que este funcionamento apenas se dá quando o nível de água no reservatório de água a repor é suficiente, quando este não o é, o sensor de distância instalado no Controlador Secundário-*Slave_Sensores* coloca na sua estrutura uma informação relativa a tal acontecimento. Posteriormente, quando o Controlador Principal-*Master* recebe e lê a estrutura atualizada, avalia se existe alguma indicação relativa ao nível de água no reservatório de água não ser suficiente e, caso exista, não ativa o motor de retorno. Neste cenário é também enviado para o Servidor de Internet uma indicação de que é necessário encher o reservatório de água a repor, para que o utilizador seja posteriormente informado.

Esta mesma lógica é usada para a alteração dos estados de todos os outros relês de controlo, seja a alteração autónoma, como é o caso do nível da água, ou requisitada, como é o caso do menu de funcionalidades. Um outro valor avaliado autonomamente é a temperatura do aquário. Caso esta seja mais elevada que 27°C o relê correspondente ao termostato é desligado, para prevenir situações em que tenha sido uma anomalia no funcionamento deste a causa do aquecimento do sistema. Um outro valor avaliado é a temperatura ambiente. Caso esta seja maior que 26°C as ventoinhas de arrefecimento são ativadas. Posteriormente, estas são desativadas caso a temperatura desça dos 26°C.

O último método de alteração dos estados dos relês de controlo consiste num menu de funcionalidades em que o utilizador tem ao seu dispor diversas funcionalidades do sistema, tanto através do uso dos botões de pressão e LCD do Controlador Principal-*Master*, como através da interface web. A Tabela 5-4 ilustra todas as funcionalidades do sistema.

Tabela 5-4 - Funcionalidades do Sistema

| | | | | |
|-----------------|---------|--------------------------------|-------------|---------------|
| Funcionalidades | | | | |
| Ligar/Desligar | Sistema | Escumador | Termostato | Modo limpeza |
| Funcionalidades | | | | |
| Ligar/Desligar | Luzes | Sistema retorno água evaporada | Luz filtros | Motor retorno |

O utilizador ao escolher uma das funcionalidades acima através da interface web é, como visto no capítulo 4.1.3.3, utilizada uma codificação que consiste em enviar um carácter específico via a ligação série que existe entre os dois controladores. Essa informação é posteriormente lida e processada de forma a que a função pedida seja executada. No entanto, caso o utilizador utilize os botões de pressão e o LCD instalados diretamente no Controlador Principal-*Master* para escolher a funcionalidade desejada, já não é utilizada a mesma codificação. Neste caso, como o utilizador apenas pode visualizar uma funcionalidade de cada vez à medida que avança pelo menu, o Controlador Principal-*Master* consegue saber diretamente qual a funcionalidade que o utilizador está a visualizar quando este decide ativar ou desativar, conseguindo desta forma ativar ou desativar os relês correspondentes à funcionalidade em causa.

5.1.3.5 Configuração LCD

O LCD usado utiliza a biblioteca `LiquidCrystal_I2C` de forma a simplificar a sua implementação para o funcionamento desejado.

Inicialmente tem de ser definido o tamanho do LCD, que neste caso é de 16 colunas por 2 linhas. Posteriormente, são inicializadas as funções da biblioteca correspondentes à iniciação das funções de escrita no ecrã e alimentação devida do sistema para questões de iluminação do ecrã.

Na altura de escrever valores no ecrã, através da função `lcd.print()`, têm primeiro de ser indicadas as coordenadas nas quais vai ser colocado o primeiro carácter ou algarismo da variável a imprimir, através da função `lcd.setCursor()`. Esta mesma lógica é

usada para todas as variáveis que sejam impressas. Quando necessário, o LCD pode ser reiniciado através da função *lcd.clear()*.

5.2 Controlador Secundário-*Slave_Sensores*

O Controlador Secundário-*Slave_Sensores* é responsável pela leitura e envio para o Controlador Principal-*Master* dos valores lidos pelos diversos sensores e módulos ligados a si. Este não processa a informação da mesma forma que o Controlador Principal-*Master* processa, isto é, não realiza ações sobre o sistema com base nos valores lidos, apenas verifica se os valores lidos são do tipo esperado e, caso o sejam, envia-os.

5.2.1 Componentes Controlador Secundário-*Slave_Sensores*

5.2.1.1 *Microcontrolador*

Como visto anteriormente, o Controlador Secundário-*Slave_Sensores* necessita de receber e enviar valores lidos por diversos sensores e módulos. Para tal, necessita de utilizar as capacidades de processamento de um microcontrolador. Dado que, é necessária a utilização de pinos digitais e analógicos para receber os valores lidos pelos diversos sensores e módulos necessários, é utilizada uma placa que já os contenha instalados. A placa utilizada é o Arduino Micro, disposta na Figura 3-10. Esta possui um número suficiente de pinos analógicos e digitais para lidar com os diversos módulos e sensores utilizados e, tal como o Arduino Uno, utiliza a mesma lógica de programação da utilização das funções *setup()* e *loop()* previamente explicadas.

5.2.1.2 *Sensor de pH*

O pH é um parâmetro muito importante nos aquários de água salgada. Uma pequena variação do seu valor pode causar reações imediatas no bem estar da vida animal no ecossistema. Deste modo, é importante que o seu valor seja verificado em tempo real.

De forma a que seja possível possuir uma leitura em tempo real do pH do ecossistema, é necessário instalar um módulo que seja capaz de ler esse valor. Para tal, é utilizado o módulo DM PH-4502C, disposto na Figura 5-11.

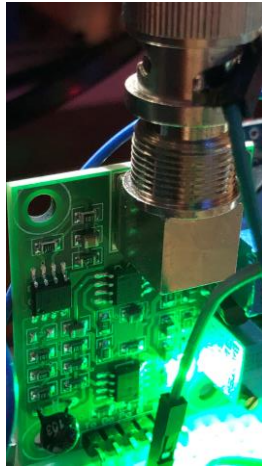


Figura 5-11 - Módulo DM PH-4502C

Este módulo utiliza uma sonda para detetar o nível de pH na água e 4 pinos para se conectar à placa utilizada. A sonda utilizada está disposta na Figura 5-12 e as ligações necessárias para a instalação do módulo na Figura 5-13.



Figura 5-12 - Sonda do módulo DM PH-4502C

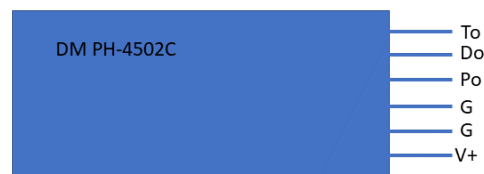


Figura 5-13 - Ligações DM PH-4502C

5.2.1.3 Módulo Real Time Clock

O sistema necessita de informações relativamente à hora atual para que possa realizar certas ações, como por exemplo, o controlo das luzes. Desta forma, é utilizado um módulo de relógio de tempo real de alta precisão, o módulo DS3231.

O DS3231, disposto na Figura 5-14, é um relógio de tempo real de alta precisão que utiliza um chip que é capaz de lidar com situações como anos bissextos, meses com menos de 31 dias e faltas de energia. Esta última é possível devido à instalação de uma

pilha no módulo que é carregada em funcionamento normal, mas ativada numa situação de falha de energia. Desta forma, após uma falha de energia não muito prolongada o sistema é capaz de, ao se voltar a ligar, realizar ações correspondentes à hora atual e não à hora a que o sistema foi abaixo. Neste módulo está também incluído um sensor de temperatura ambiente. As informações lidas pelo sensor são transferidas via protocolo I2C.



Figura 5-14 - Módulo DS3231

As ligações necessárias para o seu funcionamento em conjunto com um microprocessador estão expostas na Figura 5-15.



Figura 5-15 - Ligações DS3231

5.2.1.4 Sensor nível de água

Um problema comum nos sistemas de água salgada é a evaporação de água. Para lidar com este problema é utilizado um flutuador, disposto na Figura 5-16, que, conforme o nível de água, abre ou fecha um interruptor eletromagnético. Este comportamento gera valores binários que são posteriormente enviados para o Controlador Principal-*Master* de forma a serem processados.



Figura 5-16 - Sensor nível de água

5.2.1.5 *Sensor de temperatura da água*

Um outro campo tão importante como o valor do pH é valor da temperatura da água do sistema. Tal como o pH, uma alteração neste valor pode ter consequências imediatas na vida animal do sistema. Desta forma, o seu valor deve ser verificado em tempo real.

Para tal, é instalado no aquário de filtração um termístor à prova de água, ilustrado na Figura 5-17.



Figura 5-17 - Termístor à prova de água

5.2.1.6 *Sensor de distância*

A evaporação de água de um sistema de água salgada é um problema, tal como referido no capítulo 4.2.1.4. Neste capítulo é apresentada uma solução que utiliza um interruptor eletromagnético na forma de um flutuador que, dependendo da sua posição, ativa ou desativa uma bomba de retorno que se encontra situada num recipiente com água limpa. No entanto, existe um problema com este sistema. Caso esse recipiente fique sem água, a bomba irá ficar a trabalhar em seco, o que com o tempo a acabará por estragar e o sistema ficará com falta de água.

Para resolver este problema é usado um módulo de medição de distância. O módulo usado é o HC-SR04 e está ilustrado na Figura 5-18.

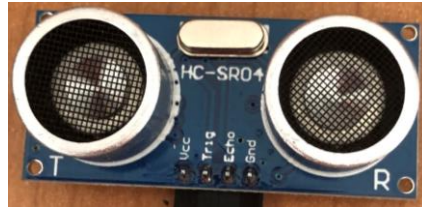


Figura 5-18 - Módulo HC-SR04

As ligações que este módulo utiliza para funcionar em conjunto com um microcontrolador estão ilustradas na Figura 5-19.

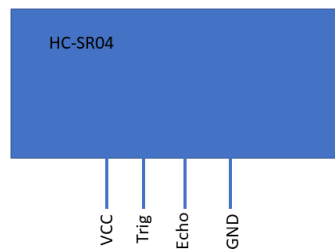


Figura 5-19 - Ligações HC-SR04

5.2.1.7 Módulo de Radiofrequência

Para que os valores lidos pelos diversos sensores e módulos sejam enviados para o Controlador Principal-*Master* via radiofrequência é necessário o uso do módulo NRF24L01, tal como é usado no Controlador Principal-*Master*. As ligações deste módulo estão disponíveis na Figura 5-20.

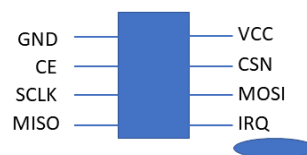


Figura 5-20 - Ligações NRF24L01

5.2.2 Ligações dos componentes do controlador Secundário-*Slave_Sensores*

O controlador Secundário-*Slave_Sensores* utiliza diversos sensores e módulos ligados ao seu microcontrolador Arduino Micro, de forma a que seja capaz de realizar todas as ações necessárias para o correto funcionamento do controlador. Na Figura 5-21 estão ilustradas todas as ligações deste.

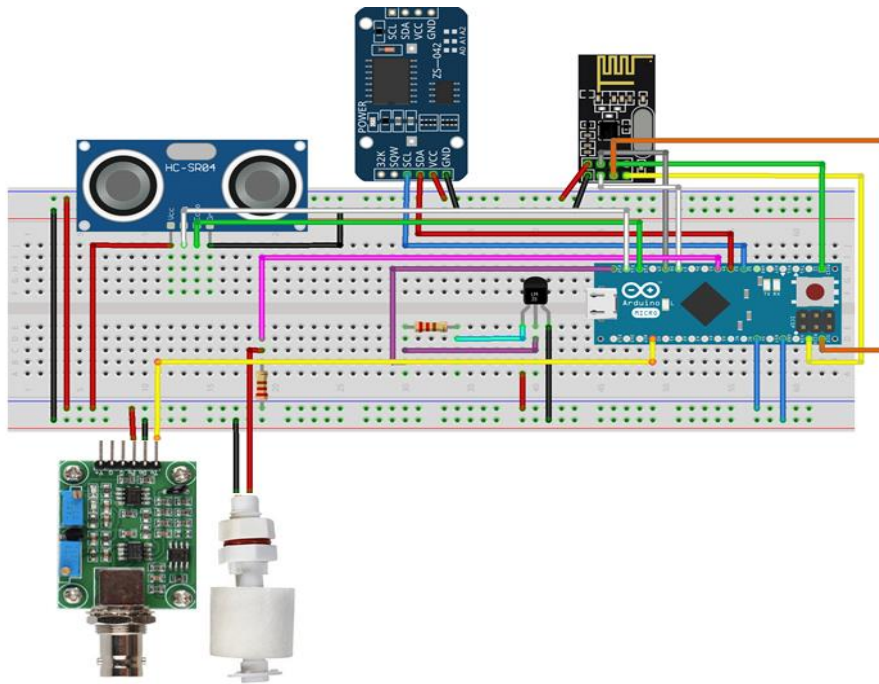


Figura 5-21 - Montagem Controlador Secundário-*Slave_Sensores*

O Arduino Micro é o centro de toda a instalação, como se pode comprovar na Figura 5-21, pois é ele quem processa todas as informações lidas pelos diversos sensores e módulos.

O NRF24L01 liga-se aos pinos 7 e 8, que correspondem às portas CE e CSN respetivamente, e aos pinos SCK (*Serial Clock*), MOSI (*Master Out Slave In*) e MISO (*Master In Slave Out*). O módulo é alimentado diretamente pela porta de 3.3V do Arduino Micro e liga-se ao GND. O módulo de distância liga-se diretamente aos pinos digitais 10 e 11 enquanto é alimentado por 5V e GND.

As conexões do módulo real time *clock* não são todas utilizadas, pois apenas são necessárias as ligações SDA, SCL, VCC e GND. Estas são ligadas respetivamente aos

pinos 2, 3, 5V e GND. Um outro módulo que não necessita de utilizar todas as suas conexões para funcionar devidamente é o de pH. Este apenas utiliza os pinos Po, G e V+, que se ligam respetivamente aos pinos A0, GND e VCC.

Tanto o termístor como o sensor de nível de água necessitam de ligações *pull-up*, através do uso de resistências de $1K\Omega$, para funcionarem corretamente. Desta forma a linha que os conecta ao Arduino Micro está sempre a *HIGH*, o que elimina o fenómeno de *floating*. Fenómeno esse que consiste em haver uma alteração do estado da ligação à porta digital, o que impossibilita uma correta leitura dos valores enviados pelos sensores.

5.2.3 Implementação

Dado que o objetivo principal do Controlador Secundário-*Slave_Sensores* é ler e enviar os valores lidos pelos diversos sensores e módulos a ele ligados, a complexidade da sua lógica de funcionamento é mais baixa que a do Controlador Principal-*Master*.

5.2.3.1 Estrutura

Como foi referido no capítulo 4.1.3, o controlador utiliza uma estrutura de forma a simplificar a troca de informações entre os seus sub-controladores. Visto que o Controlador Secundário-*Slave_Sensores* lê e guarda os valores lidos pelos seus módulos e sensores, a sua estrutura consiste inteiramente em variáveis que guardem estes vários valores, como se pode comprovar pela Figura 5-22.

```
struct package
{
    float tempAgua, pH, temp;
    int sensoragua, distSensoragua;
    char data[13];
    char hora[13];
};
typedef struct package Package;
Package data;
```

Figura 5-22 - Estrutura do Controlador Secundário-*Slave_Sensores*

O Controlador Principal-*Master* possui uma estrutura igual, de forma a que quando este recebe a estrutura enviada pelo Controlador Secundário-*Slave_Sensores*, nenhum valor seja perdido.

5.2.3.2 Envio de informações via radiofrequência

O envio de informações via radiofrequência pelo Controlador Secundário-*Slave_Sensores* segue exatamente a mesma lógica do Controlador Principal-*Master*. Isto é, é usada a mesma biblioteca RF24, através da qual têm de ser indicados os pinos aos quais se ligam as saídas CE e CSN e definido um vetor de encriptação a ser usado.

Posteriormente, tem de ser inicializada a comunicação na função *setup()* do Arduino Micro. Quando todos os valores tiverem sido lidos e for altura de enviar a estrutura para o seu destinatário, é usada a função *radio.write()* da biblioteca RF24 para a enviar. Previamente ao uso desta função, o módulo tem de ser colocado em modo de escrita.

5.2.3.3 Leitura de valores

Existem diversos módulos ligados ao Controlador Secundário-*Slave_Sensores* e todos eles necessitam de diferentes métodos para que os seus valores sejam corretamente lidos.

O termistor à prova de água utiliza a biblioteca OneWire e DallasTemperature de forma a simplificar a sua utilização. Inicialmente, tem de ser indicada a porta à qual se encontra ligado o termistor, através do uso da biblioteca OneWire. Seguidamente, através da biblioteca DallasTemperature, tem de ser chamada a função de leitura do valor lido pelo sensor, *requestTemperatures()* e, por último, é usada a função *getTempIndex()* para ler o valor lido pelo sensor. Este valor é posteriormente gravado na estrutura do controlador.

O sensor de nível de água não necessita de qualquer biblioteca para a sua utilização, dada a sua simplicidade. Basta ser definida a porta digital à qual este se encontra ligado e utilizando a função *digitalRead()* da biblioteca do Arduino, é lido o valor do sensor. Este valor é depois gravado diretamente na estrutura do controlador para que seja posteriormente analisado pelo Controlador Principal-*Master*.

A receção dos valores lidos pelo módulo *real time clock* é efetuada através do uso da biblioteca DS3231. Através desta, inicialmente, têm de ser indicados os pinos aos quais são ligadas as saídas SDA e SCL do módulo. Posteriormente, na função *setup()* do Arduino são inicializadas as funções de controlo do módulo.

Na função *loop()* são realizadas as leituras de todos os valores lidos pelo módulo. As horas são lidas através da função *getTimeStr()*, a data é recebida através do uso da função *getDateStr()* e a temperatura ambiente é recebida através do uso da função *getTemp()*.

No caso dos módulos de pH e distância não são necessárias bibliotecas para que sejam lidos os seus valores. No entanto, ambos necessitam de uma calibração prévia.

No caso do módulo de pH, para que seja possível ler o valor enviado pelo sensor basta definir a porta analógica à qual o pino Po do módulo se encontra ligado. Seguidamente, utiliza-se a função *analogRead()* do Arduino para ler o seu valor e, através de uma simples lógica, é obtido o valor lido pela sonda.

Após a instalação da sonda são medidos vários valores de pH em diversas soluções. Posteriormente, é colocada a sonda dentro de cada uma destas soluções e registado o valor lido por esta. Após algumas medições é gerada a Tabela 5-5.

Tabela 5-5 - Valores registados na calibração do módulo de pH

| Valor Teste (pH) | Valor Medido pela Sonda (mV) |
|------------------|------------------------------|
| 8,2 | 3,54 |
| 9,5 | 5,56 |
| 7 | 1,38 |
| 5,7 | 0,69 |

Após a realização das medições é realizado um gráfico correspondente aos valores da Tabela 5-5, através do qual é retirada a equação da reta da regressão linear dos valores. O gráfico e a equação obtida encontram-se dispostos no gráfico da Figura 5-23.

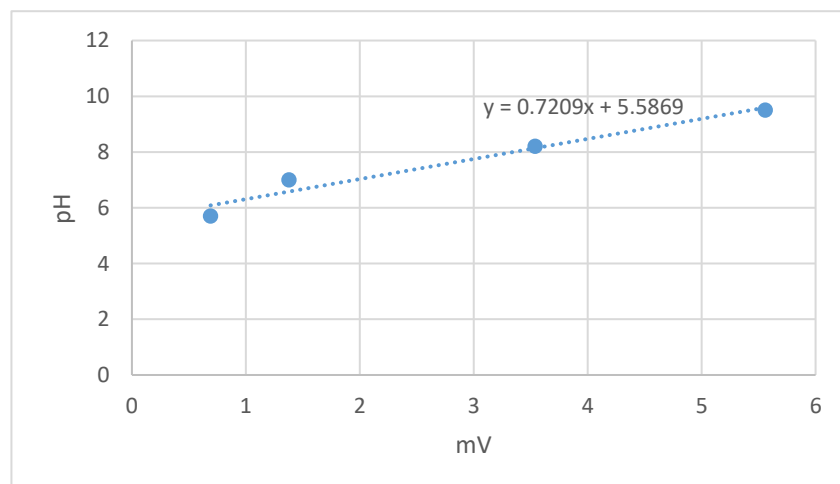


Figura 5-23 - Gráfico de calibração do módulo de pH

Após a obtenção da reta, esta é aplicada ao valor lido pela sonda, obtendo assim uma leitura correta do valor atual do pH na solução em que a sonda se encontra inserida.

Tal como o módulo de pH, o módulo de distância necessita de ser calibrado. Inicialmente, este necessita de ser instalado. Para tal, necessitam de ser indicadas as portas digitais às quais os pinos *Trig* e *echo* se encontram ligados. Define-se o pino *Trig* como *OUTPUT* e o pino *echo* como *INPUT*, para que sejam enviadas e detetadas as ondas devidamente. Seguidamente, é utilizada a função *digitalWrite()* do Arduino para colocar o pino *Trig* a *LOW* durante 5 milissegundos, seguidamente a *HIGH* e após 10 milissegundos novamente a *LOW*. Desta forma o sensor é acionado devidamente.

Após o seu acionamento, é usada a função *pulseIn()* para detetar o tempo que a onda leva a refletir e a chegar novamente ao sensor. Utilizando este valor, junto com o já conhecido valor da velocidade do som e a equação da velocidade mencionada no capítulo 2.3.5, é calculada a distância pretendida.

Para o cálculo da distância é utilizada a velocidade do som no ar. Na calibração do sensor é usada a temperatura lida pelo termístor instalado no controlador. No entanto, para ser fornecida uma explicação do seu cálculo é usada a temperatura de 25°C.

Usando a temperatura referida acima e aplicando a fórmula $c \left(\frac{m}{s} \right) = 331,3 * \sqrt{\frac{T(^{\circ}C)}{273,15}}$, obtém-se uma velocidade do som no ar de $c \left(\frac{m}{s} \right) = 346,13$, que por sua vez equivale a $0,034613 \left(\frac{cm}{\mu s} \right)$, dado que o tempo utilizado se encontra em microssegundos e a distância pretendida em *cm*. Desta forma, obtém-se a equação $d (cm) = t * \frac{t * 0.034613}{2}$, na qual *t* é o tempo em microssegundos que a onda leva a refletir do objeto em causa, devolvido pela função *pulseIn()* referida acima, sendo por isso necessário de ser dividido por dois, de forma a obter apenas o tempo de ida e não o tempo completo de ida mais volta. Após a aplicação da função o sensor encontra-se calibrado.

5.3 Controlador Secundário-Slave_Relês

O Controlador Secundário-Slave_Relês é responsável pelo controlo dos relês do sistema e pelo envio dos seus estados atuais para o Controlador Principal-Master, através de uma comunicação por radiofrequência.

5.3.1 Componentes do Controlador Secundário-*Slave_Relês*

5.3.1.1 *Microcontrolador*

Tendo em conta o número de elevador de relês que são necessários utilizar para controlar o sistema, um Arduino Uno ou Micro não possuiriam portas digitais suficientes para os instalar. Principalmente, quando também é necessário instalar o módulo de radi-frequência e este utiliza 7 pinos digitais. Como tal, é utilizado um Arduino Mega 2560, ilustrado na Figura 3-11, como microprocessador para o Controlador Secundário-*Slave_Relês*. Tal como os Arduinos Uno e Micro utilizados nos outros controladores, o Arduino Mega possui as funções *setup()* e *loop()* explicadas anteriormente como métodos de funcionamento.

5.3.1.2 *Módulo relês*

De forma a que seja possível haver um controlo sob os diversos elementos do sistema, têm de ser utilizados dois módulos de relês, como o ilustrado na Figura 5-24. Cada módulo possui 8 relês e a cada relê é ligado um diferente elemento de controlo do sistema.



Figura 5-24 - Módulo relês

No total são utilizados 2 módulos de relês, fazendo um total de 16 relês de controlo. No entanto, nem todos são utilizados devido ao número de elementos a conectar ser inferior a 16. No total apenas são utilizados 12 relês.

Cada módulo possui ligações para alimentação, VCC e GND, e ligações para o controlo de cada relê, como se pode observar na Figura 5-25.

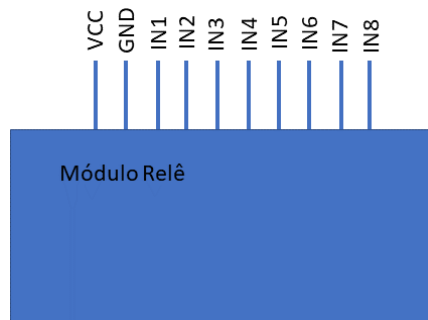


Figura 5-25 - Ligações módulo relê

Para que os módulos controlem os elementos do sistema, estes necessitam de ser ligados ao relês. No entanto, os relês não se encontram preparados para se ligarem a uma ficha típica de 230V. Para tal, a cada relê é efetuada uma ligação igual à da Figura 5-26. Desta forma, a cada relê está associada uma ficha elétrica fêmea, facilitando a ligação dos elementos do sistema aos diversos relês. É importante notar que, para que tenha energia cada relê tem de se ligar também a uma fonte de alimentação de 230V, através da ficha macho ilustrada da Figura 5-26.

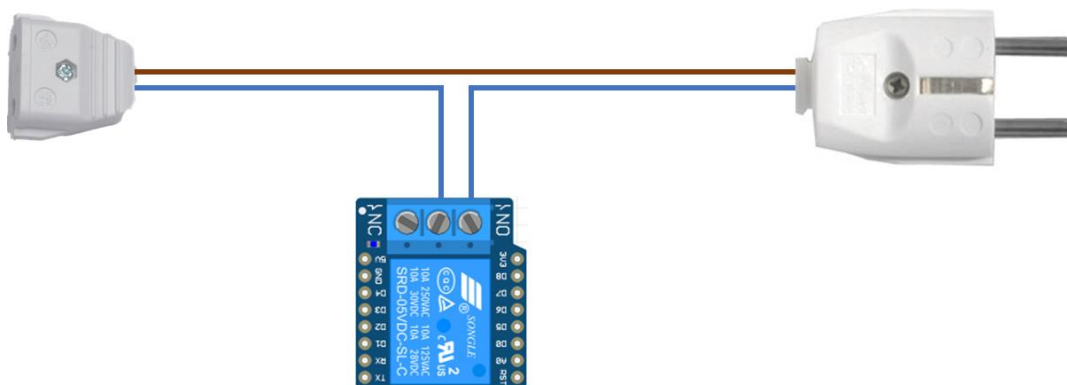


Figura 5-26 - Montagem ficha fêmea ao relê

5.3.1.3 Módulo de radiofrequência

O Controlador Secundário-*Slave_Relês* utiliza o mesmo módulo NRF24L01, ilustrado na Figura 3-20, que é utilizado no Controlador Principal-*Master* e no Controlador Secundário-*Slave_Sensores*, e segue a mesma lógica de funcionamento. Este possui as ligações ilustradas na Figura 5-20.

5.3.2 Ligações dos componentes do controlador Secundário-*Slave_Relês*

Visto que a principal função do Controlador Secundário-*Slave_Relês* é atualizar o estado dos diversos relês de controlo do sistema de acordo com as ordens que este recebe do Controlador Principal-*Master* os seus componentes são mais limitados em número que os do Controlador Secundário-*Slave_Sensores*. A Figura 5-27 mostra as ligações necessárias para a montagem do Controlador Secundário-*Slave_Relês*.

As duas placas de relês são alimentadas por 5V e GND. Ambas são controladas por pinos digitais do Arduino Mega 2560, sendo que a primeira placa liga os relês de 1 a 8 respetivamente aos pinos 22, 24, 26, 28, 30, 32, 34 e 36 e segunda placa liga os relês de 1 a 4 respetivamente aos pinos 38, 40, 42 e 44.

Tal como no caso dos outros controladores, o Controlador Secundário-*Slave_Relês* utiliza o módulo NRF24L01 para receber e enviar informação. No caso do Arduino Mega este é alimentado por 3.3V e GND e possui os pinos CE e CSN ligados respetivamente aos pinos 7 e 8.

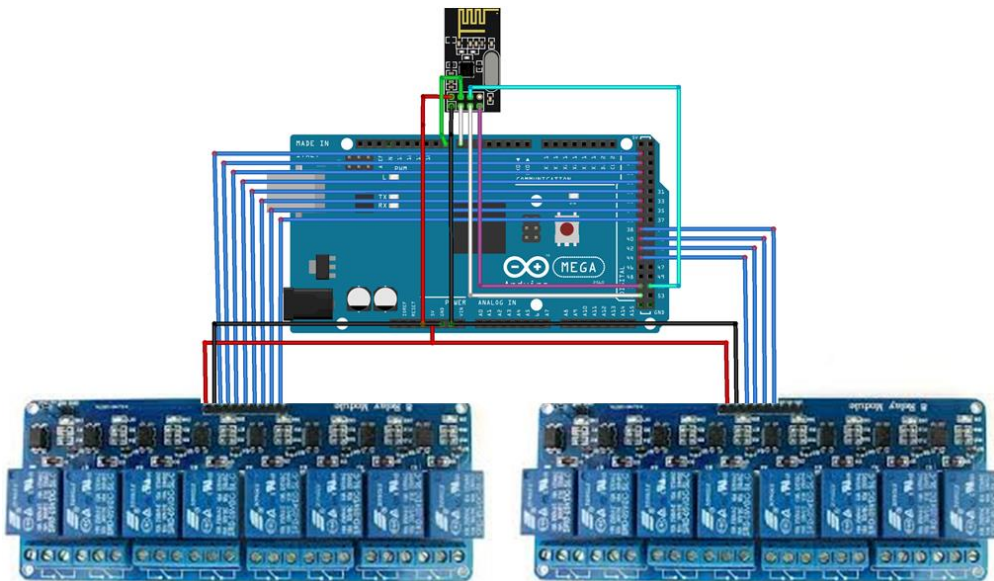


Figura 5-27 - Montagem Controlador Secundário-*Slave_Relês*

5.3.3 Implementação

As funcionalidades do Controlador Secundário-*Slave_Relês* consistem em receber ordens do Controlador Principal-*Master* sobre que relês de controlo do sistema ativar ou desativar e enviar, também para o Controlador Principal-*Master*, informações relativas aos estados atuais dos diversos relês de controlo do sistema.

5.3.3.1 Estruturas

De forma a que o Controlador Secundário-*Slave_Relês* seja capaz de satisfazer as suas funcionalidades este necessita de utilizar estruturas, tal como o Controlador Secundário-*Slave_Sensores*. No entanto, este necessita de utilizar duas estruturas, uma correspondente às ordens recebidas do Controlador Principal-*Master* e outra relativa aos estados atualizados que este vai enviar para o Controlador Principal-*Master*. Ambas as estruturas possuem exatamente as mesmas variáveis, visto que ambas contêm informações relativas aos mesmos elementos, os relês de controlo do sistema. A Figura 5-28 ilustra o conteúdo das estruturas usadas.

```
struct rele_package_recev
{
    int rele1_state; //led blue
    int rele2_state; //led white
    int rele3_state; //T4's
    int rele4_state; //sump return pump //liga inicio
    int rele5_state; //skimmer //liga inicio
    int rele6_state; //movement engine UP //liga inicio
    int rele7_state; //movement engine DOWN //liga inicio
    int rele8_state; //thermostat //liga inicio
    int rele9_state; //Cooling fans
    int rele10_state; //Auto top up sist em //liga inicio
    int rele11_state; //Sump lights
    int rele12_state;
    int rele13_state;
    int rele14_state;
    int rele15_state;
    int rele16_state;
};
typedef struct rele_package_recev r_package_recev;
r_package_recev rele_recev;
```

Figura 5-28 - Estrutura usada pelo Controlador Secundário-*Slave_Relês*

5.3.3.2 Envio e recepção de valores via radiofrequência e controle dos relês

Os valores enviados pelo Controlador Secundário-*Slave_Relês* são relativos aos estados atuais dos relês de controlo do sistema e, tal como o Controlador Secundário-*Slave_Sensores*, este utiliza o módulo NRF24L01 para os enviar via radiofrequência. Desta forma a biblioteca RF24 é também utilizada e os métodos de utilização são os mesmos dos outros controladores.

Inicialmente têm de ser definidas, utilizando a biblioteca RF24, as portas correspondentes aos pinos CE e CSN e o vetor de encriptação. Seguidamente, na função *setup()* do Arduino, tem de ser inicializada a comunicação via radiofrequência e definido o modo de funcionamento como modo de leitura, pois inicialmente o Controlador Secundário-*Slave_Relês* recebe os valores enviados pelo Controlador Principal-*Master*. Seguidamente, no decorrer da função *loop()* do Arduino, são aplicadas as alterações contidas na estrutura recebida, via radiofrequência, aos relês de controlo. Após a aplicação das alterações, são lidos os estados dos vários relês de controlo e é atualizada a estrutura correspondente a estes. Posteriormente, é alterado o modo de funcionamento do NRF24L01 para modo de escrita e a estrutura contendo os estados atuais dos relês de controlo do sistema é enviada. O fluxograma da Figura 5-29 ilustra este funcionamento.

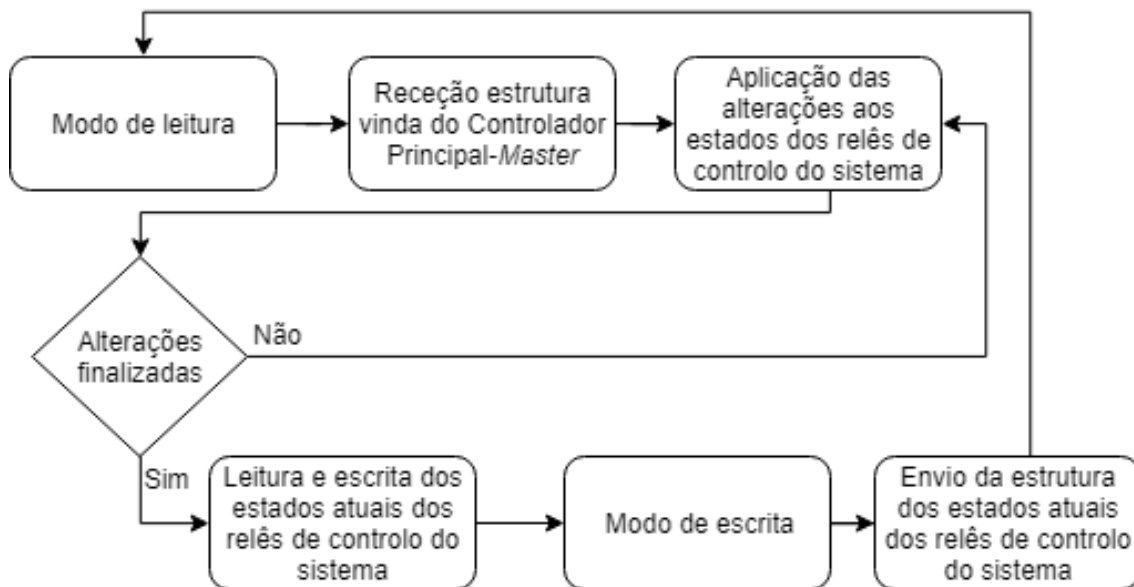


Figura 5-29 - Fluxograma do funcionamento do envio e recepção de informações do Controlador Secundário-*Slave_Relês*

5.4 Servidor de Internet

O Servidor de Internet possui um comportamento e funcionalidades diferentes de todos os outros controladores apresentados anteriormente, pois este não tem como principais funcionalidades contribuir para o correto funcionamento do sistema.

As suas principais funcionalidades são guardar todos os valores recebidos e posteriormente enviados para o Servidor de Internet, pelo Controlador Principal-*Master*, numa base de dados. Este funciona também como servidor para uma interface web na qual podem ser visionados todos os valores mais recentes recebidos, podem ser ativadas ou desativadas funcionalidades do sistema, pode ser visionado um vídeo em direto do aquário e pode ser reiniciado todo o sistema remotamente.

5.4.1 Componentes do Servidor de Internet

5.4.1.1 *Microcontrolador*

O microcontrolador utilizado no Servidor de Internet necessita de ser capaz de conter e gerir uma base de dados SQL e funcionar como servidor para uma interface web. Ambos estes serviços necessitam de ser disponibilizados numa rede sem fios local, logo o microcontrolador necessita de possuir uma antena sem fios integrada para que se consiga ligar à rede Wi-Fi usada.

Uma outra necessidade do microcontrolador usado é ser capaz de receber informações via ligações USB e possuir uma boa capacidade de processamento. Desta forma, o microcontrolador utilizado para o Servidor de Internet é o RaspberryPi, ilustrado na Figura 3-12.

5.4.1.2 *Câmara*

Uma das principais funcionalidades do Servidor de Internet é possibilitar o visionamento de um vídeo em direto através da sua interface web. Para que tal seja possível é necessário este utilizar uma câmara de vídeo.

A câmara usada é uma webcam da HP HD-2200, ilustrada na Figura 3-4, com conexão USB. Esta é ligada diretamente ao RaspberryPi através de uma das entradas USB disponíveis e, através do uso de uma biblioteca, envia as suas imagens captadas para o RaspberryPi, onde são posteriormente encaminhadas para a interface web.

5.4.2 Ligações dos componentes do Servidor de Internet

Como visto anteriormente, o Servidor de Internet recebe as suas informações relativas ao sistema através do Controlador Principal-*Master*. Estas são enviadas via uma ligação série entre a porta de saída do Arduino Uno do Controlador Principal-*Master* e uma das portas USB do RaspberryPi do Servidor de Internet, como se pode observar na Figura 5-30.

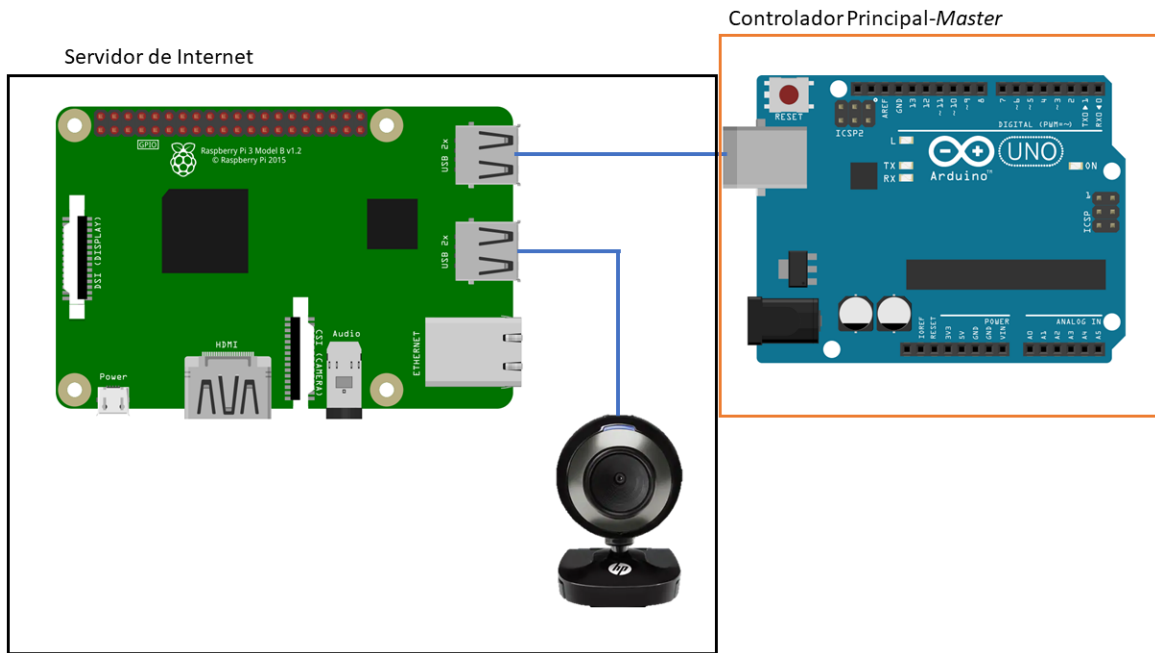


Figura 5-30 - Montagem do Servidor de Internet

O RaspberryPi do Servidor de Internet não possui muitos sensores ou módulos ligados a ele mesmo. Tal é devido ao facto das suas funcionalidades serem realizadas maioritariamente via software. Para além da ligação série ao Arduino Uno do Controlador Principal-*Master* previamente referida, o RaspberryPi do Servidor de Internet apenas possui uma ligação a uma câmara de vídeo, via uma das suas portas USB.

É importante notar que os constituintes do Servidor de Internet se restringem ao RaspberryPi e à câmara HP HD-2200. A representação do Arduino Uno na Figura 5-30 apenas é usada para ilustrar a ligação que o Raspberry Pi do Servidor de Internet realiza com o Arduino Uno do Controlador Principal-*Master*.

5.4.3 Implementação

O Servidor de Internet trabalha maioritariamente com todos os valores lidos pelo sistema, tanto do ponto de vista de sensores e módulos, como de estados atuais dos relés de controlo. Todas estas informações são recebidas pelo Servidor de Internet via uma ligação série utilizando o protocolo UART.

5.4.3.1 Método de receção de informação

Dado que todas as informações recebidas pelo Servidor de Internet são enviadas de uma só vez pelo Controlador Principal-*Master*, métodos de identificação dos diversos campos contidos na mensagem recebida necessitam de ser implementados.

Para tal, o Controlador Principal-*Master* utiliza um espaçamento predefinido entre cada campo da sua mensagem, isto é, utiliza um espaçamento entre, por exemplo, o campo do valor do pH e o campo do valor da temperatura ambiente. Desta forma, ao receber a mensagem, o Servidor de Internet procura pelos espaçamentos para saber onde acaba e onde começa cada campo. No final da mensagem, isto é, após o valor do último campo, é enviado o carácter correspondente a um parágrafo. Desta forma, o Servidor de Internet consegue identificar o fim da mensagem. A ordem dos campos enviados na mensagem está representada na Tabela 5-6.

Tabela 5-6 - Estrutura da mensagem recebida pelo Servidor de Internet

| Mensagem | | | |
|----------------|---------------|------------------|---------------|
| Campo | Campo | Campo | Campo |
| Temperatura | pH | Temperatura água | Hora |
| Campo | Campo | Campo | Campo |
| Nível da água | Nível Água | Data | Estado rele 1 |
| Campo | Campo | Campo | Campo |
| Estado rele 2 | Estado rele 3 | Estado rele 4 | Estado rele 5 |
| Campo | Campo | Campo | Campo |
| Estado rele 6 | Estado rele 7 | Estado rele 8 | Estado rele 9 |
| Campo | | Campo | |
| Estado rele 10 | | Estado rele 11 | |
| Campo | | | |
| Estado rele 12 | | | |

O Servidor de Internet lê uma mensagem com os campos presentes na Tabela 5-6 e com a estrutura referida acima a cada minuto de funcionamento, de forma a possuir sempre os valores mais recentes na sua interface web e ter um registo mais detalhado do comportamento do sistema.

5.4.3.2 Base de dados

Como referido no capítulo 4.4.3.1, as informações recebidas pelo Servidor de Internet são colocadas numa base de dados. Para tal, uma base de dados necessita de ser configurada no RaspberryPi.

De forma a que possa ser configurada uma base de dados, primeiro têm de ser instalados os pacotes que esta necessita para existir. Para tal, é necessário instalar pelo menos a versão 7.0 do *PHP*, a partir da qual pode então começar a ser instalada a base de dados a partir do comando *php-MySQL*.

Tendo o *php-MySQL* instalado, é necessário instalar o *My-SQL-Server* e o *My-SQL-Client*. Por fim, falta apenas instalar um método de criar e interagir com diversas bases de dados. Para tal, é instalado o *phpmyadmin*. Após este ser configurado, o meio a partir do qual diversas bases de dados podem ser criadas está instalado. Na Figura 5-31 está disposta a interface de início de sessão no *phpmyadmin*.



Figura 5-31 - Interface de início de sessão na ferramenta *phpmyadmin*

Após o login na ferramenta *phpmyadmin* é então possível criar e interagir com várias bases de dados. Para o caso do controlador desenvolvido é usada uma base de dados intitulada “Controlador_Aquário”, na qual existe uma tabela intitulada “Sensores”, como

é possível visualizar na Figura 5-32. Abrindo a tabela “Sensores” é então possível visualizar todos os campos constituintes da tabela e todas as entradas existentes em cada um deles, como é possível visualizar na Figura 5-33.



Figura 5-32 - Base de dados e tabela para o controlador

| TemperaturaAmbiente | pH | TemperaturaAgua | Hora | NivelAgua | NivelAguaRetorno | Data | rele1_state | rele2_state |
|---------------------|------|-----------------|----------|-----------|------------------|------------|-------------|-------------|
| 29 | 8.24 | 28.25 | 20:20:00 | OK | Medio Baixo | 2019-08-07 | 1 | 1 |
| 29 | 8.24 | 28.25 | 20:19:00 | OK | Medio Baixo | 2019-08-07 | 1 | 1 |
| 29 | 8.23 | 28.25 | 20:18:00 | OK | Medio Baixo | 2019-08-07 | 1 | 1 |
| 29 | 8.23 | 28.25 | 20:16:00 | OK | Medio Baixo | 2019-08-07 | 1 | 1 |
| 29 | 8.25 | 28.25 | 20:15:00 | OK | Medio Baixo | 2019-08-07 | 1 | 1 |
| 28 | 8.25 | 28.25 | 20:14:00 | OK | Medio Baixo | 2019-08-07 | 1 | 1 |

Figura 5-33 - Parte da tabela "Sensores" da base de dados "Controlador Aquário"

5.4.3.3 Interface web

A interface web desenvolvida permite um acesso remoto a todas as funcionalidades do Controlador Principal-Master, às entradas mais recentes na base de dados e a um vídeo em direto do aquário. Para tal, a interface web utiliza um servidor *Apache* junto de um servidor *Flask* como base do seu funcionamento.

O servidor *Apache* é utilizado na página principal da interface web, ilustrada na Figura 5-34. Nesta, são apresentadas duas tabelas, uma correspondente aos valores mais recentes lidos pelos diversos módulos e sensores instalados no sistema e outra correspondente ao estado mais recente dos vários relês de controlo do sistema. É necessário o uso do servidor *Apache* sobre o uso do servidor *Flask* para esta página devido ao facto de que para que seja possível para esta página conter os valores mais recentes de sensores, módulos e relês, esta necessita de aceder à base de dados “Controlador_Aquário” e retirar

todos os campos da tabela “Sensores” com o “id” mais recente. Para tal, a página principal necessita de ser do tipo *PHP*. Ora, o servidor *Flask* não é capaz de correr páginas do tipo *PHP*, daí o uso do servidor *Apache*.

Controlador Aquário

| Camara | Funcionalidades | Base de Dados | Atualizar Parametros | Reiniciar Sistema | | |
|---|----------------------------------|-----------------------|-------------------------|--------------------|----------|------------|
| Valores dos Parâmetros do Aquário | | | | | | |
| pH | Temperatura Ambiente | Temperatura Água | Nível Água | Nível Água Retorno | Hora | Data |
| 8.24 | 28 | 28.12 | OK | Medio Baixo | 02:15:00 | 2019-08-08 |
| Estado Relês de Contrlo do Sistema | | | | | | |
| Led Azul | Led Branco | Luzes T4 | Motor Retorno | | | |
| 0 | 0 | 0 | 1 | | | |
| Escumador | Motor Movimento Cima | Motor Movimento Baixo | Termostato | | | |
| 1 | 1 | 1 | 0 | | | |
| Ventoinhas de Arrefecimento | Sistema Reposição Água Evaporada | Luzes Filtros | Termostato de Segurança | | | |
| 0 | 0 | 0 | 0 | | | |

Figura 5-34 - Página Principal da interface web

As restantes páginas da interface web correm no servidor *Flask*. O servidor *Flask* é baseado na linguagem de programação Python e é utilizado na interface web do controlador desenvolvido devido, principalmente, à sua capacidade de executar scripts de Python junto de pedidos *Ajax*. Tal funcionalidade é necessária no controlador desenvolvido, pois este necessita de enviar comandos via série para o Arduino Uno do Controlador Principal-*Master* a pedido do utilizador, via um clique num botão na interface web.

O servidor *Flask* funciona de uma forma diferente do servidor *Apache*. A principal diferença consiste no facto de que todos os pedidos, sejam para trocar de página web ou executar um script de *Python*, necessitam de passar por um programa *Python* intitulado de “app.py”. O programa “app.py” funciona como o cérebro do servidor *Flask*. Este está sempre em execução e é responsável pelo lançamento da página inicial, ilustrada na Figura 5-35, no lançamento do servidor. É também responsável pela troca de páginas web a pedido do utilizador via o uso de botões na interface, pela execução de ações e pela execução de scripts, com o objetivo de realizar funções no Controlador Principal-*Master*. Este funcionamento está ilustrado na Figura 5-36.

Controlador Aquário

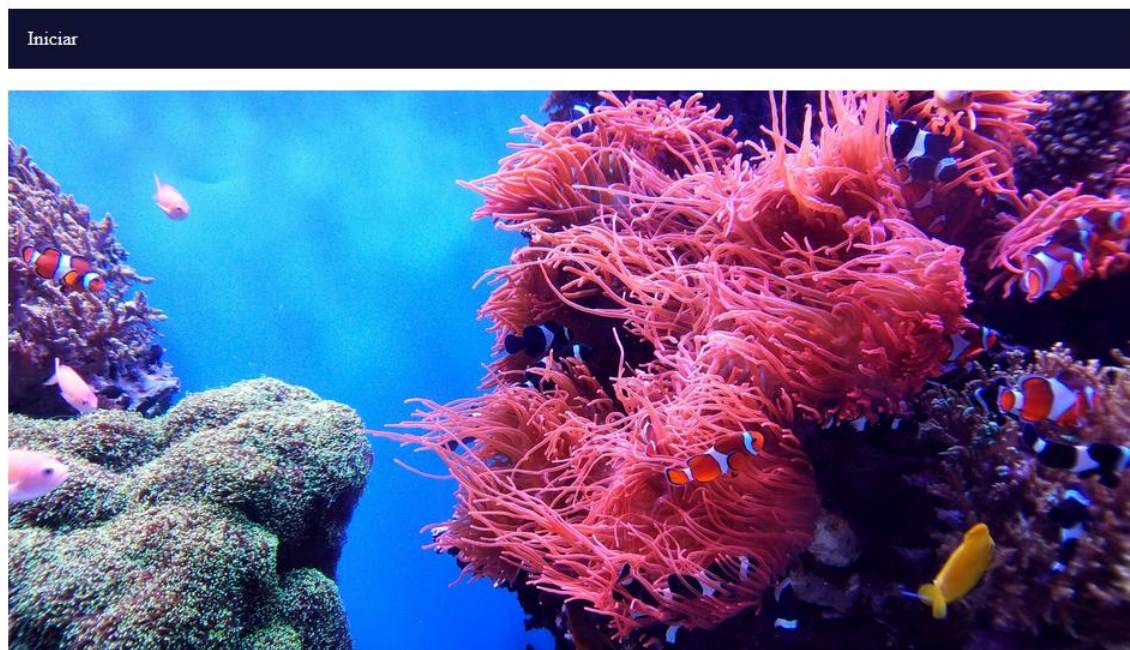


Figura 5-35 - Página inicial da interface web

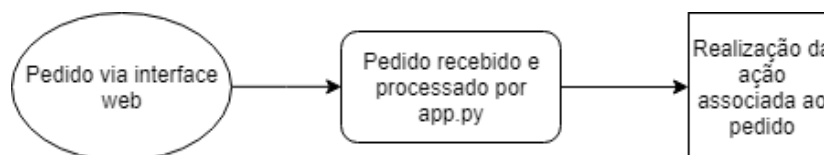


Figura 5-36 - Modo de funcionamento do servidor *Flask*

A página inicial da interface web tem como principal objetivo realizar uma ponte entre os dois servidores. Sendo inicialmente lançada uma página no servidor *Flask*, caso não haja uma troca para o servidor *Apache*, não será possível de executar uma página do tipo *PHP* com o objetivo de aceder à base de dados e ler os campos mais recentes da tabela em causa. Logo, tem de ser efetuada uma troca de servidores. Para tal, ao ser usado o botão “Iniciar” na página inicial, em vez de ser lançado um pedido ao programa “app.py”, como explicado anteriormente, é antes utilizado um comando, “*href*”, para que o utilizador seja reencaminhado para a página principal, funcionando esta no servidor *Apache*.

No entanto, estando na página principal qualquer pedido via o uso de um botão na interface web que o utilizador realize irá reencaminhá-lo automaticamente para uma página a funcionar no servidor *Flask*. Desta forma, ao ser utilizado o botão “Funcionalidades” na página principal, o utilizador é reencaminhado para uma página web baseado no servidor *Flask*. A página resultante está representada na Figura 5-37.

Controlador Aquário

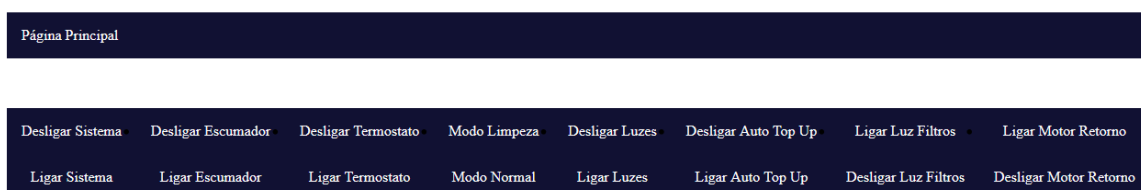


Figura 5-37 - Página Funcionalidades da interface web

Nesta página o utilizador pode realizar qualquer uma das tarefas dispostas na Tabela 5-2, sendo que as ações são realizadas segundo a lógica do fluxograma da Figura 5-38.

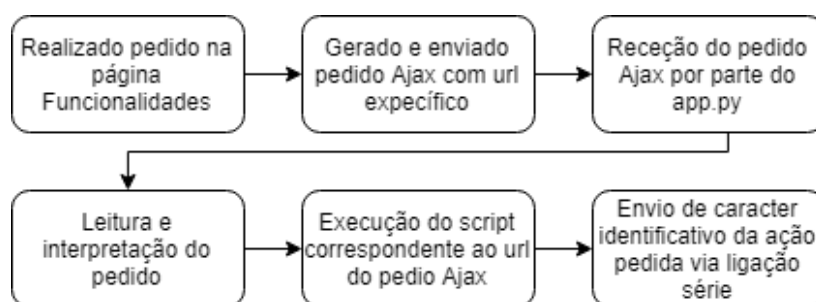


Figura 5-38 - Fluxograma do funcionamento de um pedido via a página web Funcionalidades

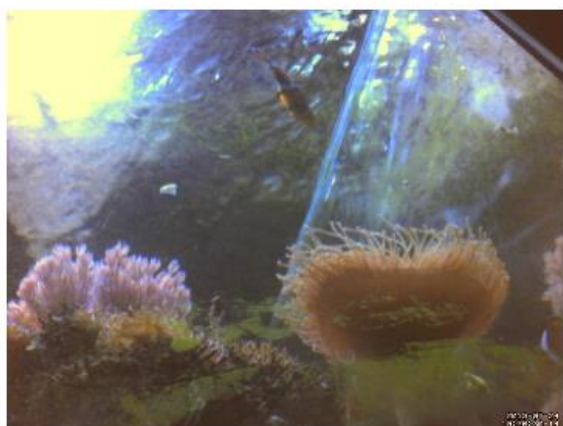
Caso o utilizador, estando na página principal, escolha o botão “Câmara” este é reencaminhado, segundo as regras do servidor *Flask*, para uma página, ilustrada na Figura 5-39, onde pode visionar um vídeo em direto pela câmara instalada no RaspberryPi do Servidor de Internet. Nesta página o utilizador encontra também um link, situado abaixo do vídeo em direto, com a função de o reencaminhar para uma página contento apenas o vídeo em direto.

A emissão de vídeo em direto na página câmara é possível devido à instalação e utilização da biblioteca *Motion* no RaspberryPi. Esta biblioteca possibilita a emissão de vídeo em direto de uma câmara instalada no RaspberryPi, após devida configuração,

numa porta web à escolha pelo *developer*. A porta usada neste sistema é a 8081. Desta forma, se o utilizador pretender visualizar apenas o vídeo em direto da câmara instalada no sistema, este apenas necessita de aceder à porta 8081. Esta ação é a realizada pelo link existente na página câmara mencionado acima.

Controlador Aquário

Página Principal



[Camara Live Stream Safety Link](#)

Figura 5-39 - Página Câmara da interface web

As configurações necessárias de realizar para a utilização da biblioteca *Motion* encontram-se expostas na Tabela 5-7.

Tabela 5-7 - Configurações necessárias de realizar na biblioteca *Motion*

| | |
|----------------------|-----------|
| Framerate | 100 |
| Stream_localhost | Desligado |
| Webcontrol_localhost | Desligado |
| Qualidade | 100 |
| Largura | 640 |
| Altura | 480 |
| Post_capture | 5 |
| Output_pictures | Desligado |
| ffmeg_output_movies | Desligado |

O *framerate*, a qualidade, o *post_capture* e as dimensões da janela de vídeo foram escolhidas de forma a proporcionar um equilíbrio entre a qualidade da imagem e o peso que o vídeo gera na transmissão. O campo *Stream_localhost* e o *Webcontrol_localhost* encontram-se desativados de forma a que seja possível o visionamento do vídeo transmitido através de outro computador para além do computador em que a biblioteca se encontra instalada. Os campos *Output_pictures* e *ffmpeg_output_movies* encontram-se desativados para que não seja feita uma cópia do vídeo transmitido no Raspberry Pi. O primeiro campo impede que seja gravado o vídeo sob a forma de imagens e o segundo impede que seja gravado o vídeo sob a forma de pequenos vídeos.

Caso o utilizador deseje ver a base de dados através da ferramenta *phpmyadmin* basta clicar no botão “Base de Dados” quando na página principal. Nesta mesma página o utilizador pode realizar ações de reinício do sistema e atualização de parâmetros.



Validação do funcionamento

Neste capítulo são apresentados e validados os diferentes modos de funcionamento do controlador desenvolvido e as diversas ações autónomas que o controlador toma, de forma a minimizar danos no ecossistema, consoante alterações em parâmetros do próprio ecossistema e também do meio em que este se encontra, através da instalação do controlador num aquário próprio. De forma a que todos estes sejam validados, é usada a base de dados do controlador.

6.1 Validação

O normal funcionamento do controlador desenvolvido gera dados relativos tanto aos sensores e módulos usados, como ao funcionamento dos relês de controlo do sistema. Estes valores são guardados numa base de dados, segundo a ordem presente na Tabela 5-2, para os relês de controlo do sistema.

Neste ponto, são usados esses valores para validar o funcionamento do controlador. Serão validadas situações em funcionamento normal, como o ligar e desligar dos sistemas de iluminação, e situações em funcionamento de correção de anomalias, como a reação do controlador perante um aumento da temperatura ambiente.

Serão também testadas as diversas funcionalidades do sistema, perante comandos gerados de diferentes formas.

6.1.1 Validação do correto funcionamento em situações normais

Por situação normal entende-se uma situação em que não existem alterações no meio em que o sistema se encontra que possam causar uma mudança no seu comportamento, nem alterações no próprio sistema, tanto do ponto de vista de mau funcionamento de elementos, como problemas gerados por má qualidade da água, que possam causar desajustes. Nestas situações o controlador deve lidar com todos os seus elementos sem efetuar nenhum ajuste.

6.1.1.1 Sistema de iluminação

O sistema possui três diferentes sistemas de iluminação e cada um deles realiza uma função distinta. Deste modo, é fundamental que estes funcionem nos horários devidos. Os seus horários de funcionamento encontram-se na Tabela 5-3.

Avaliando uma situação em que a hora atual é 13:15H, o único sistema de iluminação que deve estar ligado é o LED azul, pois este entra em funcionamento às 13:00H. Analisando a base de dados existente na Figura 6-1 é possível observar este mesmo funcionamento. É também possível verificar que todos os outros sistemas de iluminação se encontram desligados.

| Hora | NivelAgua | Data | rele1_state | rele2_state | rele3_state |
|----------|-----------|------------|-------------|-------------|-------------|
| 13:17:00 | OK | 2019-07-17 | 0 | 1 | 0 |
| 13:16:00 | OK | 2019-07-17 | 0 | 1 | 0 |
| 13:15:00 | OK | 2019-07-17 | 0 | 1 | 0 |
| 13:14:00 | OK | 2019-07-17 | 0 | 1 | 0 |
| 13:13:00 | OK | 2019-07-17 | 0 | 1 | 0 |

Figura 6-1 Validação dos sistemas de iluminação às 13:15H

Verificando uma outra situação em que a hora atual é 13:59H, os dois sistema de iluminação que devem estar ligados são os LED azul e branco, pois o sistema correspondentes às luzes mais fortes apenas se liga às 14:00H. Tal comportamento é verificado na base de dados, como é possível verificar na Figura 6-2.

| Hora | NivelAgua | Data | rele1_state | rele2_state | rele3_state |
|----------|-----------|------------|-------------|-------------|-------------|
| 14:02:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 14:01:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 14:00:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 13:59:00 | OK | 2019-07-17 | 1 | 1 | 0 |
| 13:58:00 | OK | 2019-07-17 | 1 | 1 | 0 |

Figura 6-2 - Validação dos sistemas de iluminação às 13:59H

Para a hora atual de 16:48H todos os sistemas de iluminação devem estar operacionais, como se pode verificar na Figura 6-3.

| Hora | NivelAgua | Data | rele1_state | rele2_state | rele3_state |
|----------|-----------|------------|-------------|-------------|-------------|
| 16:52:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 16:51:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 16:49:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 16:48:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 16:47:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 16:46:00 | OK | 2019-07-17 | 1 | 1 | 1 |

Figura 6-3 - Validação dos sistemas de iluminação às 16:48H

O comportamento dos sistemas de iluminação na altura de se desativarem também cumpre as regras estabelecidas. Como se pode observar na Figura 6-4, para uma situação em que a hora atual seja 21:48H todos os sistemas de iluminação se encontram operacionais. No entanto, para uma situação em que a hora atual seja 22:48H, apenas se encontra ativo o LED azul, como se pode observar na Figura 6-5.

| Hora | NivelAgua | Data | rele1_state | rele2_state | rele3_state |
|----------|-----------|------------|-------------|-------------|-------------|
| 21:52:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 21:51:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 21:49:00 | OK | 2019-07-17 | 1 | 1 | 1 |
| 21:48:00 | OK | 2019-07-17 | 1 | 1 | 1 |

Figura 6-4 - Validação dos sistemas de iluminação às 21:48H

| Hora | NivelAgua | Data | rele1_state | rele2_state | rele3_state |
|----------|-----------|------------|-------------|-------------|-------------|
| 22:51:00 | OK | 2019-07-17 | 1 | 0 | 0 |
| 22:50:00 | OK | 2019-07-17 | 1 | 0 | 0 |
| 22:49:00 | OK | 2019-07-17 | 1 | 0 | 0 |
| 22:48:00 | OK | 2019-07-17 | 1 | 0 | 0 |
| 22:47:00 | OK | 2019-07-17 | 1 | 0 | 0 |

Figura 6-5 - Validação dos sistemas de iluminação às 22:48H

6.1.1.2 Modo limpeza e Modo normal

O modo de limpeza é usado pelo utilizador quando este necessita de efetuar manutenção no sistema. Por esta razão, este desativa todos os elementos do aquário de filtração, sendo estes o escumador, o sistema de reposição de água evaporada e o motor de retorno de água do aquário de filtração. Devido à desativação deste último, o nível de água no aquário principal irá baixar. Desta forma, é também desativado o motor de movimento de cima, para impedir que este funcione fora de água durante o período de limpeza, o que o poderia danificar. É também desativado o termostato, pois devido à sua localização no aquário principal este pode sobreaquecer se permanecer ativo fora de água por algum tempo.

A Figura 6-6 e Figura 6-7 ilustram os estados dos relês de controlo do sistema antes e depois da ativação do modo de limpeza, sendo que este foi ativo às 14:22H.

| Hora | NivelAgua | Data | rele1_state | rele2_state | rele3_state | rele4_state | rele5_state |
|----------|-----------|------------|-------------|-------------|-------------|-------------|-------------|
| 14:23:00 | OK | 2019-07-22 | 1 | 1 | 1 | 0 | 0 |
| 14:22:00 | OK | 2019-07-22 | 1 | 1 | 1 | 1 | 1 |

Figura 6-6 - Estados dos relês de controlo no modo limpeza (1)

| rele6_state | rele7_state | rele8_state | rele9_state | rele10_state | rele11_state | rele12_state |
|-------------|-------------|-------------|-------------|--------------|--------------|--------------|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Figura 6-7 - Estados dos relês de controlo no modo limpeza (2)

Como é possível observar os relês 4, 5, 6 e 8, que correspondem, respetivamente, ao motor de retorno de água do aquário de filtração, ao escumador, ao motor de movimento de cima e ao termostato, se desativaram após a ativação do modo limpeza.

O modo normal consiste no cancelamento do modo limpeza. Por esta razão, a sua ativação gera a reativação dos relês desativados pelo modo limpeza, como é possível verificar na Figura 6-8 e na Figura 6-9.

| Hora | NivelAgua | Data | rele1_state | rele2_state | rele3_state | rele4_state | rele5_state |
|----------|-----------|------------|-------------|-------------|-------------|-------------|-------------|
| 14:26:00 | OK | 2019-07-22 | 1 | 1 | 1 | 1 | 1 |
| 14:25:00 | OK | 2019-07-22 | 1 | 1 | 1 | 0 | 0 |

Figura 6-8 - Estado dos reles de controlo apos a ativação do modo normal (1)

| rele6_state | rele7_state | rele8_state | rele9_state | rele10_state | rele11_state | rele12_state |
|-------------|-------------|-------------|-------------|--------------|--------------|--------------|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |

Figura 6-9 - Estado dos reles de controlo após ativação do modo normal (2)

6.1.1.3 Termostato

O termostato é responsável pelo aquecimento e estabilização da temperatura da água do sistema. O utilizador escolhe a temperatura desejada no termostato e este ativa-se e desativa-se consoante a temperatura atual, de forma a manter a temperatura o mais constante possível.

No entanto, uma outra forma de alterar o estado de ativação do termostato é através de qualquer uma das interfaces de controlo que o utilizador tem ao seu dispor. Caso o

utilizador escolha a funcionalidade de desativar o termostato, este permanece desativado até que ordem em contrário seja dada, independentemente de todas as outras ações do sistema, como é possível verificar na Figura 6-10. Nesta é possível verificar que às 14:56H foi dada a ordem pelo utilizador de desligar o termostato. Logo, este encontra-se desligado independentemente de todos os fatores, como a temperatura da água, que ao ser inferior a 25°C gera uma ordem de ativação do termostato. Desta forma, caso o utilizador não tivesse gerado uma ordem de desativação, o termostato deveria estar ativo.

| TemperaturaAgua | Hora | rele8_state |
|-----------------|----------|-------------|
| 24.5 | 14:57:00 | 0 |
| 24.44 | 14:56:00 | 1 |

Figura 6-10 - Desativação manual do termostato

6.1.1.4 Luzes filtros

O sistema possui um aquário de filtração normalmente situado dentro de um móvel. Por esta razão, é necessária a utilização de uma luz neste, de forma a facilitar a manutenção. Desta forma, o utilizador consegue ativar e desativar esta luz através de qualquer uma das interfaces disponibilizadas pelo controlador, como é possível visualizar na Figura 6-11.

| Hora | NivelAgua | Data | rele11_state |
|----------|-----------|------------|--------------|
| 03:44:00 | OK | 2019-07-18 | 0 |
| 03:44:00 | OK | 2019-07-18 | 1 |
| 03:43:00 | OK | 2019-07-18 | 1 |
| 03:43:00 | OK | 2019-07-18 | 0 |

Figura 6-11 – Ativação e desativação do rele correspondente à luz do aquário de filtração

6.1.1.5 Escumador

O escumador é um elemento de filtração essencial para qualquer sistema de água salgada. No entanto, existem alturas em que este necessita de ser desligado. Para tal, o controlador disponibiliza em qualquer uma das suas interfaces a opção de desativação deste mesmo a qualquer altura de funcionamento do sistema, como é possível observar na Figura 6-12.

| Hora | rele5_state |
|----------|-------------|
| 15:12:00 | 0 |
| 15:11:00 | 1 |

Figura 6-12 - Desativação do rele correspondente ao escumador sob ordem do utilizador

6.1.1.6 Motor retorno

O motor de retorno envia água do aquário de filtração de volta para o aquário principal. Este, tal como as luzes do aquário de filtração, pode ser controlado por ambas as interfaces do controlador. No entanto, ao ser desligado o motor de retorno é necessário também ser desligado o escumador, por razões de segurança. Como tal, o controlador força o estado de ativação do rele correspondente ao escumador a desligado. A Figura 6-13 ilustra a mudança de estado de ativação do rele correspondente ao motor retorno e a consequente alteração de estado do rele do escumador.

| Hora | rele4_state | rele5_state |
|----------|-------------|-------------|
| 16:22:00 | 0 | 0 |
| 16:21:00 | 1 | 1 |

Figura 6-13 - Alteração dos estados de ativação dos reles correspondentes ao motor retorno e escumador

6.1.1.7 Bomba retorno de água evaporada

Com o tempo o aquário perde alguma água devido ao fenómeno de evaporação. Com o intuito de resolver tal problema, o controlador utiliza um sensor que deteta o nível de água no sistema a todo o momento. Quando este desce abaixo de um certo ponto, uma bomba é ligada. Esta bomba envia água de volta para o aquário, normalizando assim o seu nível. Após a sua normalização a bomba é desligada. A Figura 6-14 mostra este fenómeno.

| NivelAgua | Data | rele10_state |
|-----------|------------|--------------|
| OK | 2019-07-24 | 0 |
| Baixo | 2019-07-24 | 1 |

Figura 6-14 - Comportamento do sistema perante uma diminuição e posterior aumento do nível de água

6.1.1.8 Desligar e ligar sistema

Ambas as interfaces do sistema possibilitam ao utilizador desligar por completo o sistema, isto é, desligar todos os reles de controlo do sistema e impedir o controlador de os ativar independentemente dos valores lidos pelos vários sensores e módulos instalados no sistema. A Figura 6-15 ilustra os estados de todos os reles a desativo devido a uma ordem dada às 20:49H.

| TemperaturaAmbiente | pH | TemperaturaAgua | Hora | NivelAgua | NivelAguaRetorno | Data |
|---------------------|------|-----------------|----------|-----------|------------------|------------|
| 29 | 8.24 | 28.31 | 20:49:00 | OK | Medio Baixo | 2019-08-07 |

| rele1_state | rele2_state | rele3_state | rele4_state | rele5_state | rele6_state | rele7_state |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| rele8_state | rele9_state | rele10_state | rele11_state | rele12_state |
|-------------|-------------|--------------|--------------|--------------|
| 0 | 0 | 0 | 0 | 0 |

Figura 6-15 - Estados dos reles de controlo do sistema todos desativados devido a ordem do utilizador

É possível visualizar nesta que certos elementos deviam estar ativos, como por exemplo os reles correspondentes ao sistema de iluminação ou às ventoinhas de arrefecimento do sistema. No entanto, como foi dada uma ordem pelo utilizador de desativação do sistema nenhum rele se encontra ativo.

O controlador mantém o estado de desativação dos reles durante 12 horas, caso o utilizador não o desative através de uma das duas interfaces do sistema. Esta desativação é gerada automaticamente com o intuito de impedir danos no sistema devido a uma desativação total accidental por parte do utilizador. Ao ser ligado o sistema volta ao estado de funcionamento em que se encontrava antes de ter sido desligado.

6.1.2 Validação do correto funcionamento em situações extraordinárias

Todos os comportamentos descritos anteriormente são relativos a situações de funcionamento normal do sistema, isto é, situações em que não existem fatores externos, ou relacionados ao sistema a afetar o seu funcionamento. No entanto, estas situações existem. Como tal, o controlador é capaz de as detetar e avaliar de forma a realizar ações com intuito de as corrigir.

6.1.2.1 Termostato

Como referido no capítulo 5.1.1.3 o termostato é responsável pelo aquecimento e estabilização da temperatura da água do sistema. No entanto, existem vários casos no mundo da aquariofilia de termostatos que ficaram presos no seu modo de aquecimento, o que causa a temperatura da água do aquário aumentar consideravelmente, o que pode causar danos irreversíveis ao ecossistema.

Para impedir tal acontecimento, o controlador monitoriza a temperatura da água e apenas enquanto esta é inferior a 25°C é que o rele correspondente ao termostato se encontra ativo, como é possível verificar na Figura 6-16. Desta forma, nunca existem casos problemáticos como o descrito acima.

| TemperaturaAgua | Hora | NivelAgua | Data | rele8_state |
|-----------------|----------|-----------|------------|-------------|
| 27.5 | 03:29:00 | OK | 2019-07-18 | 0 |
| 27.37 | 03:28:00 | OK | 2019-07-18 | 0 |
| 24.87 | 03:27:00 | OK | 2019-07-18 | 1 |
| 24.81 | 03:26:00 | OK | 2019-07-18 | 1 |

Figura 6-16 – Alteração estado rele correspondente ao termostato devido ao aumento da temperatura da água

O controlador avalia também a temperatura da água e caso esta seja maior que 28°C ou inferior a 23°C, não só força o estado de ativação do termostato a ligado, ou desligado conforme a situação em causa, como envia também um email para o utilizador a informar de tal acontecimento. Na Figura 6-17 é possível de verificar que a temperatura da água é maior que 28°C e que, como tal, o termostato se encontra desligado. Na Figura 6-18 é possível de verificar que um email foi enviado para o utilizador na mesma hora a informar de tal acontecimento.

| TemperaturaAgua | Hora |
|-----------------|----------|
| 30.75 | 17:01:00 |

Figura 6-17 - Indicação na base de dados da temperatura da água às 17:01H

ALERTA!



aquariumcontroller26@gmail.com <aquariumcontroller26@gmail.com>

17:02

Temperatura Agua Elevada

Figura 6-18 - Email recebido pelo utilizador às 17:02H a informar da temperatura da água elevada

O mesmo comportamento é realizado para situações em que a temperatura da água seja inferior a 23°C.

6.1.2.2 *Motor retorno*

Como referido no capítulo 5.1.1.6 o motor retorno é usado para enviar a água que desce do aquário principal para o aquário de filtração, de volta para o aquário principal. Por esta razão, caso a bomba falhe certos elementos situados no aquário de filtração, como o escumador e o sistema de reposição de água evaporada, não podem continuar em funcionamento sem colocar o sistema em risco.

Desta forma, o controlador monitoriza o estado de ativação do rele correspondente ao motor retorno. Quando este se encontra ativo, o controlador não altera o estado dos elementos referidos acima, mas quando este se encontra desativo, seja por que razão for, o sistema força o estado de funcionamento dos elementos referidos acima a desligado. A Figura 6-19 ilustra este comportamento.

| Hora | NivelAgua | Data | rele4_state | rele5_state |
|----------|-----------|------------|-------------|-------------|
| 17:28:00 | OK | 2019-07-24 | 0 | 0 |
| 17:27:00 | OK | 2019-07-24 | 0 | 0 |
| 17:26:00 | OK | 2019-07-24 | 1 | 1 |

Figura 6-19 - Detecção de falha no motor retorno às 17:26H e consequente desativação do rele correspondente ao escumador

6.1.2.3 Ventoinhas

O aumento da temperatura ambiente do local em que o sistema se encontra pode gerar um aumento considerável da temperatura da água do aquário, o que como já referido anteriormente pode causar danos irreversíveis ao ecossistema. Como tal, para ajudar a minimizar este tipo de situações o controlador monitoriza a temperatura ambiente. Quando esta é superior a 27°C o relê correspondente ao controlo de duas ventoinhas de arrefecimento é ativado, como é comprovado pela Figura 6-20.

| TemperaturaAmbiente | rele9_state |
|---------------------|-------------|
| 28 | 1 |
| 26 | 0 |

Figura 6-20 - Ativação do rele correspondente às ventoinhas de arrefecimento decido ao aumento de temperatura ambiente

Para situações em que a temperatura ambiente é mais elevada que 30°C, é também enviado um email ao utilizador com o objetivo de o alertar de tal acontecimento. Na Figura 6-21 é possível verificar o registo na base de dados da temperatura ambiente às 18:15H. Devido a esta ser superior a 30°C o utilizador recebeu um email a avisar de tal situação, como é possível verificar na Figura 6-22.

| TemperaturaAmbiente | pH | TemperaturaAgua | Hora |
|---------------------|------|-----------------|----------|
| 31 | 9.17 | 27.37 | 18:15:00 |

Figura 6-21 - Registo de temperatura ambiente elevada às 18:15H

ALERTA!



aquariumcontroller26@gmail.com <aquariumcontroller26@gmail.com>

18:15

Temperatura Ambiente Elevada

Figura 6-22 - Email recebido pelo utilizador às 18:15H a avisar da temperatura ambiente elevada

6.1.2.4 pH e escumador

Um dos vários parâmetros lidos no aquário é o pH. Este valor representa o bom, ou mau, estado de saúde do ecossistema. Um sistema com má manutenção e sistema de filtração, irá ter consequentemente um aumento gradual do pH. Desta forma, existe uma parceria entre o pH de um aquário e a qualidade do sistema de filtração deste, ou seja, o escumador.

O controlador recebe e analisa constantemente o valor mais atual do pH do sistema e, caso este seja inferior a 8, que é o valor mínimo aceitável para o sistema em causa, ou superior a 9.5, que é o valor máximo aceitável para o sistema em causa, o controlador força a ativo o estado de ativação do relê correspondente ao escumador, pois poderia por alguma razão estar desligado. É importante notar que caso o sistema se encontre em modo limpeza o controlador não ativa o escumador, pois nessa situação o valor de pH lido pode ser incorreto. Tal comportamento é possível de ser verificado pela Figura 6-23.

| pH | TemperaturaAgua | Hora | rele5_state |
|------|-----------------|----------|-------------|
| 8.52 | 27.31 | 18:57:00 | 1 |
| 7.88 | 27.31 | 18:56:00 | 1 |
| 8.61 | 27.31 | 18:56:00 | 0 |

Figura 6-23 – Alteração do estado do rele do escumador devido à diminuição do valor de pH

Para além da verificação do funcionamento do escumador o controlador envia um email ao utilizador com o objetivo de o alertar para a situação em causa, seja ela de pH elevado ou de pH baixo, como é possível verificar na Figura 6-25 relativo à informação presente na Figura 6-24.

| pH | TemperaturaAgua | Hora |
|------|-----------------|----------|
| 7.72 | 27.37 | 18:37:00 |

Figura 6-24 - Registo na base de dados do valor de pH às 18:37H

ALERTA!



aquariumcontroller26@gmail.com <aquariumcontroller26@gmail.com>

18:38

pH muito baixo

Figura 6-25 - Email recebido pelo utilizador às 18:38H relativo ao baixo valor de pH

6.1.2.5 Bomba retorno água evaporada

Como referido no capítulo 5.1.1.7 o controlador possui um sistema de reposição de água evaporada que utiliza um reservatório à parte. No entanto, existe um problema caso

este reservatório fique sem água, pois não só o motor de retorno de água evaporada acabaria por se queimar, como o sistema ficaria com um défice de água, o que colocaria todos os seus elementos em perigo.

De forma a impedir tal acontecimento, o controlador verifica a quantidade de água existente no reservatório e controla a bomba de retorno com base na sua quantidade. Desta forma, se o reservatório ficar sem água, o controlador impede a bomba de retorno de água evaporada de se ativar e envia um email ao utilizador para o avisar de que necessita de repor água no reservatório. Apenas quando a quantidade de água no reservatório for superior a um limite mínimo estabelecido é que o controlador volta a permitir que a bomba de retorno de água evaporada funcione. Tal acontecimento é possível de se verificar através da Figura 6-26.

| NivelAgua | NivelAguaRetorno | rele10_state |
|-----------|------------------|--------------|
| Baixo | Baixo | 0 |
| OK | Baixo | 0 |

Figura 6-26 - Registo da não ativação do rele correspondente à bomba de retorno de água evapora perante uma diminuição no nível de água do aquário

Como se pode verificar o rele correspondente à bomba de retorno de água evaporada não se ativou perante uma diminuição do nível da água, como visto no capítulo 5.1.1.7. Neste caso, este mantém-se desativado devido ao nível de água no reservatório ser baixo. Apenas quando este aumentar é que a bomba de retorno de água evaporada voltará a se comportar como visto no capítulo 5.1.1.7.

ALERTA!



aquariumcontroller26@gmail.com <aquariumcontroller26@gmail.com>

20:51

Falta de agua no reservatorio

Figura 6-27 - Email recebido pelo utilizador quando o nível de água no reservatório é baixo

Na Figura 6-27 está ilustrado o email recebido pelo utilizador quando o nível de água no reservatório se encontra abaixo do mínimo permitido.

6.1.2.6 Termostato de emergência

Como visto anteriormente o termostato é responsável por aquecer a água do sistema. Caso este falhe o controlador detecta esta falha e avisa o utilizador de tal acontecimento. No entanto, até que este possa resolver a falha a água do aquário começa a arrefecer, o que coloca todos os elementos do aquário em perigo.

Perante esta situação, o controlador possui a capacidade de lidar com um termostato de emergência, isto é, um termostato que se ativa quando a temperatura do sistema é demasiado baixa, minimizando assim a probabilidade da água do aquário arrefecer. A Figura 6-28 ilustra a sua ativação, em conjunto com o termostato principal, perante um valor baixo da temperatura da água.



Figura 6-28 - Registo da ativação dos reles correspondentes ao termostato principal e termostato de segurança perante um valor de temperatura da água muito baixo



Conclusões e Trabalho Futuro

Neste capítulo são apresentadas as conclusões relativas aos métodos de implementação utilizados e aos resultados obtidos relativamente ao controlador desenvolvido. São também apresentados trabalhos futuros de que o controlador poderia beneficiar.

7.1 Conclusões

Ao longo desta dissertação foi apresentado um controlador para aquários de água salgada de baixo custo e fácil instalação que possui dois principais objetivos, manter o correto estado de funcionamento do aquário e possibilitar ao utilizador um acesso rápido e fácil a todas as suas funcionalidades e parâmetros lidos. O controlador teve de obedecer a diversas regras de funcionamento dos aquários de água salgada, como um horário fixo de ativação e desativação de diversos tipos de iluminação.

De forma a que todos os requisitos fossem cumpridos, foram utilizadas quatro diferentes placas programáveis, possuindo cada uma delas uma funcionalidade diferente e interagindo cada uma delas com diferentes elementos. O mantimento do correto estado de funcionamento do sistema foi possível devido à constante comunicação entre três das quatro placas programáveis que constituem o controlador. A quarta placa foi utilizada para possibilitar o rápido e fácil acesso às diversas funcionalidades e parâmetros lidos do sistema.

Numa fase inicial foi desenvolvido um sistema simples, através do uso de apenas uma placa programável, que fosse capaz de receber informações lidas por diversos sensores e processá-las de forma a identificar se existiam algumas anomalias.

Em seguida, foi desenvolvido um sistema com a mesma finalidade, mas com uma maior complexidade, pois em vez de utilizar apenas uma placa programável, utilizava três placas programáveis. Uma delas funcionava como o *master* e as outras duas funcionavam como *slaves*. Um dos *slaves* era responsável por ler os valores enviados pelos sensores usados e o outro era responsável por ativar e desativar relês de controlo.

Por fim, foi utilizada uma quarta placa programável ligada diretamente ao *master* do sistema referido no passo anterior. Através desta foi criada uma base de dados para onde os valores lidos pelos sensores e as ações tomadas sob os relês de controlo fossem enviados para que ficassem registados. Posteriormente, foi desenvolvida uma interface web capaz de ilustrar os valores mais recentes existentes na base de dados, de transmitir um vídeo em direto do aquário e de realizar ações sobre o *master*. A última tarefa realizada foi a criação de uma lógica de análise de parâmetros para que o utilizador fosse informado, via email, sempre que algum parâmetro se encontrasse fora de um intervalo aceitável.

Após validado o controlador, foi verificado que este cumpriu os requisitos necessários, ao ser capaz de responder a alterações, tanto no meio em que o aquário se encontra, como em alterações de parâmetros do próprio ecossistema, devidamente e a tempo de minimizar possíveis danos causados no ecossistema. Foi também verificado que o controlador é capaz de realizar ações de alteração de modos de funcionamento do aquário e de estados de ativação de relês de controlo de diversos elementos devidamente e em tempo reduzido.

Uma outra verificação realizada no desenrolar do desenvolvimento do controlador foi que os sensores de salinidade com capacidade suficiente para detetar alterações no pequeno intervalo de salinidade necessário para manter o desejado funcionamento do ecossistema, de 1.021 a 1.025 SG (*Specific Gravity*), possuem um custo muito elevado. Desta forma, o controlador desenvolvido não possui um sensor de salinidade.

Todos os testes realizados ao sistema com o intuito de o validar mostram que o controlador, apesar de ser capaz de responder aos pedidos sempre dentro de um prazo máximo de 1 minuto, pode ser otimizado através da utilização de um método de comunicação mais eficiente e com menos falhas, como o Wi-Fi, em vez dos métodos utilizados de série e radiofrequência.

O controlador desenvolvido conta com vários elementos, desde sensores a plataformas de desenvolvimento. A Tabela 8 - Custos dos diversos elementos do controlador desenvolvido apresenta os custos dos vários elementos na altura do desenvolvimento desta dissertação.

Tabela 8 - Custos dos diversos elementos do controlador desenvolvido

| Elemento | Custo (€) |
|------------------------|-----------|
| Arduino Uno | 27,62 |
| Arduino Mega | 43,72 |
| Arduino Micro | 35,06 |
| Raspberry Pi 3+ | 43,4 |
| Sensor de distância | 3,05 |
| Sensor de temperatura | 2,53 |
| Sensor de pH | 10,83 |
| Sensor nivel fludo | 3,03 |
| Módulo RTC | 1,8 |
| Módulo Relês | 10,45 |
| Módulo Relês | 10,45 |
| Módulo Radiofrequência | 1,64 |
| Camara USB | 10 |
| Total | 203,58 |

7.2 Trabalho Futuro

O controlador desenvolvido é capaz de lidar com alterações de parâmetros inesperadas, no entanto não possui um sistema de alimentação dos animais existentes no aquário. Tal sistema pode ser realizado através do uso de uma arca frigorífica de pequenas dimensões na qual se encontra um reservatório com comida dissolvida. Ligado a este reservatório usar-se-ia um tubo que, através do uso de uma bomba de água, seria capaz de enviar comida diretamente para o aquário. Visto que a bomba de água seria controlada pela placa programável *master*, seria possível de implementar um horário de alimentação fixo ou variável, consoante necessidades.

Um outro trabalho futuro é a alteração dos métodos de comunicação usados por Wi-Fi, de forma a unificar e facilitar a instalação e utilização do dispositivo.

Como trabalho futuro é também sugerido o desenvolvimento de um método de machine learning que seja capaz de prever problemas ou falhas no sistema.

Referências

- [1] Mello, Silvia, "*Aquarofilia como ferramenta de educação ambiental no ensino*", vol. 9, Janeiro 2015.
- [2] Henrique Miguel Pereira, Tiago Domingos, Luís Vicente, Vânia Proença, "*Ecosistemas e Bem-Estar Humano*", Lisboa: Escolar Editora, 2009.
- [3] Baychenko, Nikolay, "Implementing a Master/Slave Architecture for a data synchronization service", TURKU UNIVERSITY OF APPLIED SCIENCES, Finlândia, 2018.
- [4] Fang Yi-yuan, CHEN Xue-jun, "International Workshop on Intelligent Systems and Applications", *Design and Simulation of UART Serial Communication Module Based on VHDL*, Maio 2011.
- [5] Inkbird, "manualslib," 2016. [Online]. Available: <https://www.manualslib.com/manual/1405269/Inkbird-Itc-308.html#manual>. [Acedido em Janeiro 2019].
- [6] Inkbird, "manualslib image", 2016. [Online]. Available: <https://www.manualslib.com/manual/1405269/Inkbird-Itc-308.html?page=5#manual>. [Acedido em Janeiro 2019].
- [7] Digital Aquatics, "Amazonaws", 2013. [Online]. Available: https://s3.amazonaws.com/brsinstructions/brsDigitalAquatics/rkl/RKL+User+Guide+2_0+-+Web.pdf. [Acedido em Janeiro 2019].

- [8] Digital Aquatics, "Reefing made fun & easy", 2013. [Online]. Available: <https://www.bulkreefsupply.com/reefkeeper-lite-basic-open-box-digital-aquatics.html>. [Acedido em Janeiro 2019].
- [9] Peacock, Raymond, "sensoronline", 1 Janeiro 2006. [Online]. Available: <https://www.sensorsmag.com/components/temperature-sensors-contact-or-noncontact>. [Acedido em Janeiro 2019].
- [10] Aspencore, "electronics-tutorials", AspenCore, 2018. [Online]. Available: <https://www.electronics-tutorials.ws/>. [Acedido em Janeiro 2019].
- [11] Xiaoqiu Yang, Weiren Lin, "Encyclopedia of Engineering Geology", *Thermistor*, Agosto 2018.
- [12] Stella Girousi, Sotiris Sotiropoulos, "*pH: Principles and Measurement*", Dezembro 2016.
- [13] Woodford, Chris, "Explainthatstuff!", 23 Maio 2018. [Online]. Available: <https://www.explainthatstuff.com/how-ph-meters-work.html>. [Acedido em Janeiro 2019].
- [14] Sensorland, "Understanding pH measurement", [Online]. Available: <http://www.sensorland.com/HowPage037.html>. [Acedido em Janeiro 2019].
- [15] TC Fluid Control, "How Liquid Level Sensors Work", 2013. [Online]. Available: <http://www.tc-fluidcontrol.com/about-us/blog-posts/how-liquid-level-sensors-work/>. [Acedido em Janeiro 2019].
- [16] BoxEletronica, "Liquid Water Level Sensor Vertical", [Online]. Available: <https://www.boxelectronica.com/pt/sensores-arduino/154-liquid-water-level-sensor-vertical.html>. [Acedido em Janeiro 2019].
- [17] Foundation, RaspberryPi, "raspberrypi", [Online]. Available: <https://www.raspberrypi.org/products/camera-module-v2/>. [Acedido em Julho 2019].
- [18] "RandomNerdTutorials.com", [Online]. Available: <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>. [Acedido em Julho 2019].

- [19] Centre for microcomputer applications, "Salinity sensors", *SALINITY SENSOR BT78i*, 2014.
- [20] Copco, "Sensor de Salinidade", 2019. [Online]. Available: <http://www.copcoinc.com/salinity-sensor/>. [Acedido em Janeiro 2019].
- [21] Zlatanov, Nikola, "*Arduino and Open Source Computer Hardware and Software*", Novembro 2015.
- [22] Arduino, "Arduino Uno", 2019. [Online]. Available: <https://store.arduino.cc/arduino-uno-rev3>. [Acedido em Janeiro 2019].
- [23] Arduino, "Arduino Micro", 2019. [Online]. Available: <https://store.arduino.cc/arduino-micro>. [Acedido em Janeiro 2019].
- [24] "Arduino Mega", 2019. [Online]. Available: <https://store.arduino.cc/mega-2560-r3>. [Acedido em Julho 2019].
- [25] opensource, "What is Raspberry Pi", [Online]. Available: <https://opensource.com/resources/raspberry-pi>. [Acedido em Janeiro 2019].
- [26] Raspberry Pi Foundation, "Raspberry Pi 3 Model B+", [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. [Acedido em Janeiro 2019].
- [27] Raspberry Pi Foundation, "Raspberry Pi 3 Model B+", [Online]. Available: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>. [Acedido em Janeiro 2019].
- [28] S. Kesatoshi Takeuchi, "Serial communication system using". Estados Unidos Patente US 7,802,036 B2, 21 Setembro 2010.
- [29] "robomart", [Online]. Available: <https://www.robomart.com/433mhz-rf-transmitter-and-receiver>. [Acedido em Junho 2019].
- [30] Agarwal, Tarun, "EDGEFX", 2017. [Online]. Available: <https://www.edgefxkits.com/blog/rf-transceiver-module-with-block-diagram-explanation/>. [Acedido em Junho 2019].
- [31] Ugweje, Okechukwu, "*Radio Frequency and Wierless Communications*", Abril 2004.

- [32] Deavall, Stuart, "toner giant", 15 Agosto 2015. [Online]. Available: <https://www.tonergiant.co.uk/blog/2013/08/wireless-printing-everything-you-wanted-to-know-but-were-afraid-to-ask/>. [Acedido em Agosto 2019].
- [33] Tp-link, "tp-link", [Online]. Available: <https://www.tp-link.com/pt/home-networking/wifi-router/archer-c50/>. [Acedido em Agosto 2019].
- [34] Telele, Daniel, "CCM", [Online]. Available: <https://ccm.net/faq/298-what-is-wifi-and-how-does-it-work>. [Acedido em Agosto 2019].
- [35] Banggood.com, "Geekcreit", [Online]. Available: https://pt.banggood.com/433Mhz-RF-Transmitter-With-Receiver-Kit-For-Arduino-ARM-MCU-Wireless-p-74102.html?cur_warehouse=CN. [Acedido em Junho 2019].
- [36] Aspencore, "electronics-tutorials", 2018. [Online]. Available: https://www.electronics-tutorials.ws/io/io_4.html. [Acedido em Janeiro 2019].
- [37] José Jair Alves Mendes Júnior, Sergio L. Stevan Jr, "*LDR e sensores de luz ambiente: funcionamento e aplicações*", Junho 2013.
- [38] Raspberry Pi Foundation, "raspberrypi", [Online]. Available: <https://www.raspberrypi.org/products/camera-module-v2/>. [Acedido em Fevereiro 2019].