

Sistema de monitorização para equipamentos de refrigeração biomédicos

Tiago Filipe Duarte Resende

Dissertação para obtenção de Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Orientadores Prof. Carlos Manuel Ribeiro Almeida
Prof. Rui Manuel Rodrigues Rocha

Júri

Presidente: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Orientador: Prof. Carlos Manuel Ribeiro Almeida
Vogal: Prof. Alberto Manuel Ramos da Cunha

Outubro 2020

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Agradecimentos

Uma tese de mestrado é uma jornada de superação, onde inúmeros fatores contribuem para o seu sucesso. O apoio de familiares e amigos, bem como do corpo docente e suas instituições, são os alicerces que suportam a caminhada. Deixo aqui o meu obrigado ao Professor Carlos Manuel Ribeiro Almeida e ao Professor Rui Manuel Rodrigues Rocha que me orientaram e guiaram para alcançar os objetivos propostos. Desde da análise à crítica construtiva, sem a vossa ajuda esta tese não tinha tido o mesmo rumo e estarei para sempre grato. Estendendo também este agradecimento ao Instituto Superior Técnico que providenciou o conhecimento, ferramentas de desenvolvimento e espaço laboratorial para o desenvolvimento da tese.

Aos três melhores amigos, cujos nomes colocarei por ordem alfabética e não por ordem de importância, Guilherme Cid, Mariana Silva e Tomás Costa. O vosso apoio não se resume à tese e o meu agradecimento por escrito não faz justiça ao sustento que trazem.

À minha namorada, Catarina Gomes, que foi a viga emocional pelo apoio que me deu ao longo de todo este caminho. Muitas foram as vezes que me ajudou a encontrar forças para continuar a superar os desafios que se iam apresentando. O desfecho desta tese não seria o mesmo sem ti. Obrigado.

E por fim à minha família. Helena Resende, Vítor Resende, Tânia Resende e Tiago Almeida. Foram vocês que lá estiveram sempre nos dias de frustração e nos dias de sucesso. Sem a vossa compreensão aos sacrifícios necessários e sem os vossos próprios sacrifícios, não tinha tido capacidade de envergar neste objetivo. O agradecimento é eterno.

Abstract

The temperature monitoring of medical products for human use is an internationally debated issue. Given the importance of ensuring the safety in distribution and storage, there are regulations that mandate the control of these substances during their life cycle, that is, from manufacture to administration. By means of electronic equipment, the control of these conditions can be done without human intervention, allowing the triggering of alerting mechanisms on abnormal situations. The economic viability of this electronic equipment has a great impact on the facilities that need it. The current master's thesis intends to study a solution that is economically viable for small and large facilities, enabling a mechanism that offers scalability at a low monetary cost with alarm mechanisms and system configuration. As well as the use of smartphones and the integration of a web platform to view and control the system. Using a mobile app for the Android operating system, one can view the current settings of the sensors that are in the system, in addition to configure each sensor in order to reach the desired criteria. Through SMS exchanges, it is possible to carry out these configuration operations without the need to be present on the site or the latest generation mobile network. The use of a web server, and in the presence of Internet access, it is possible to view the data collected and the system configuration in any part of the world, and it is also within reach to create reports if requested by the competent entities. The solution proposed in this master's thesis shows that it is feasible to develop a scalable system in an economical way with the intrinsically necessary functionalities.

Keywords: Temperature Monitoring, Alert Mechanisms, Scalability, Configuration, Mobile App, Sensor, SMS, Web Server

Resumo

A monitorização de temperaturas de produtos médicos para uso humano é um tema debatido internacionalmente. Dada a importância em garantir a segurança na distribuição e armazenamento existem regulamentações que obrigam ao controlo destas substâncias durante o seu ciclo de vida, isto é, desde a fabricação à administração. Por meio de equipamentos eletrónicos, o controlo destas condições pode ser feito sem intervenção humana, permitindo ainda desencadear mecanismos de alerta em situações que fogem às esperadas. A viabilidade económica destes equipamentos eletrónicos tem um grande impacto no seu emprego nas instalações que os necessitam. A presente tese de mestrado pretende estudar uma solução que seja economicamente viável para pequenas e grandes instalações, possibilitando um mecanismo que ofereça escalabilidade a um baixo custo monetário com mecanismos de alarme e configuração de sistema. A utilização de *smartphones* e a integração com uma plataforma web para a visualização e controlo do mecanismo presente na instalação. Com recurso a uma aplicação móvel para o sistema operativo Android, consegue-se visualizar as configurações dos sensores que se encontram no sistema, bem como configurar cada sensor de forma a atingir os critérios pretendidos. Por meio de trocas de SMS consegue-se realizar estas operações de configuração sem haver a necessidade de presença no local ou rede móvel de última geração. A utilização de um servidor *web*, e na presença de acesso à Internet, pode-se visualizar os dados recolhidos e a configuração do sistema em qualquer parte do mundo, sendo ainda possível a criação de relatórios caso solicitado pelas entidades competentes. A solução proposta nesta tese de mestrado, mostra que é possível desenvolver um sistema escalável de uma forma económica e com as funcionalidades intrinsecamente necessárias.

Palavras-chave: Monitorização de temperaturas, Mecanismos de alerta, Escalabilidade, Configuração, Aplicação Móvel, Sensor, SMS, Servidor *Web*

Índice

1	Introdução	1
1.1	Motivação e Objetivos	1
1.2	Desafios	2
1.3	Estrutura do Documento	2
2	Estado da Arte	4
2.1	Produtos Existentes	4
2.1.1	Accsense	4
2.1.2	Monnit	4
2.1.3	Phase IV Engineering	6
2.1.4	Vaisala	7
2.1.5	Sensaphone	8
2.1.6	Resumo - produtos existentes	9
2.2	Tecnologia Existente	9
2.2.1	Bluetooth®	10
2.2.2	ANT	10
2.2.3	Zigbee®	10
2.2.4	Wi-Fi®	10
2.2.5	Resumo - Comunicação sem fios	11
2.2.6	Sensores de temperatura	11
3	Arquitectura do Sistema	13
3.1	Visão global do Sistema	13
3.2	Requisitos do Sistema	13
3.3	Especificação do Sistema	15
3.3.1	Unidade de Sensor	15
3.3.2	Hub	16
3.3.3	Interface de Utilizador	16
3.4	Arquitectura	17
3.4.1	Hardware	17
3.4.2	Software	19
4	Implementação	22
4.1	Unidade de Sensor	22
4.1.1	Considerações	28
4.2	Hub	29
4.2.1	Repeater	32
4.2.2	Celular	35
4.2.3	Wi-Fi e Wi-Fi & Celular	36
4.3	Interface de Utilizador	39

4.3.1	Webpage	39
4.3.2	Aplicação Móvel	45
5	Teste do Sistema e Resultados Experimentais	52
5.1	Unidade de Sensor	52
5.2	Hub	59
5.3	Interface de Utilizador	66
6	Hardware Final	70
6.1	Unidade de Sensor	70
6.2	Hub	72
7	Conclusões e Trabalhos Futuros	74
8	Bibliografia	76
A	Anexos	80
A.1	Código C implementado na Unidade de Sensor para a Equação Steinhart-Hart	80
A.2	Código C implementado na Unidade de Sensor para o reenvio de dados e receção do <i>acknowledge</i>	80
A.3	Código Python implementado no RaspberryPi que interpreta as comunicações via UART	81
A.4	Código PHP implementado no Webserver no ato de registo de novo utilizador	82
A.5	Código PHP implementado no Webserver na geração do relatório de temperaturas registadas.	83
A.6	Código LUA implementado no NodeMCU no modo de <i>Access Point</i>	84
A.7	Código LUA implementado no NodeMCU - Envio e receção de dados via 802.11	85
A.8	Código JAVA implementado na aplicação móvel - Envio e receção de SMS	86
A.9	Características dos produtos comercializados no mercado	88

Lista de Equações

1	Divisor de tensão	24
2	Valor de conversão do ADC	24
3	Valor de conversão do ADC substituição pelo Divisor de Tensão	24
4	Relação entre valor binário e divisor de tensão	24
5	Equação de Stein-Hart	25
6	Equação simplificada de Stein-Hart	25
7	Corrente média da Unidade de Sensor	71
8	Corrente média da Unidade de Sensor (continuação)	71
9	Capacidade da bateria da Unidade de Sensor	71

Lista de Tabelas

I	Apanhado das tecnologias de comunicação wireless.	11
II	Características dos diferentes tipos de sensores de temperatura.	12
III	Requisitos funcionais do Sensor.	14
IV	Requisitos não funcionais do Sensor.	14
V	Requisitos funcionais da Hub.	14
VI	Requisitos não funcionais da Hub.	15
VII	Requisitos funcionais da Interface de Utilizador.	15
VIII	Requisitos não funcionais da Interface de Utilizador.	15
IX	Comunicação entre MCU e "Módulo Celular"	36
X	Comunicação entre MCU e "Módulo Webserver"	37
XI	<i>Software</i> utilizado na implementação LAMP do <i>webserver</i>	39
XII	Caracterização dos campos associados ao registo de um novo utilizador.	41
XIII	Corrente medida na Unidade de Sensor em diferentes contextos.	58

Lista de Figuras

1	Dispositivos de recolha de dados da Accsense.	4
2	Linha de sensores da Monnit.	5
3	Linha de <i>gateways</i> da Monnit.	6
4	Dispositivo de alerta local da Monnit. ^[1]	6
5	Linha de produtos, sensor e <i>gateways</i> , da Phase IV Engineering.	7
6	”Logger” de temperatura da Vaisala ^[2]	8
7	Sistema de monitorização da Sensaphone.	9
8	Diagrama de blocos - Unidade de Sensor.	17
9	Diagrama de blocos - Hub.	18
10	Via de comunicação entre cliente e servidor.	19
11	Via de comunicação entre aplicação móvel e Hub.	19
12	Diagrama funcional do <i>software</i> da Unidade de Sensor.	20
13	Diagrama funcional do <i>software</i> da Hub.	20
14	Diagrama funcional da Interface de Utilizador no modo Hub Wi-Fi.	21
15	Diagrama funcional da Interface de Utilizador no modo Hub Celular.	21
16	Diagrama de blocos - Unidade de Sensor.	22
17	<i>Breakout</i> , desenhado pela ”Adafruit Industries”, para o SoC RFM69HCW.	23
18	Esquema elétrico da Unidade de Sensor	25
19	Algoritmo implementado no microcontrolador na Unidade de Sensor	27
20	Algoritmo implementado no microcontrolador na Hub	31
21	Página de configuração do NodeMCU.	34
22	Exemplo de comunicação entre duas Hub’s através dos NodeMCU	35
23	Caminho de dados desde o MCU até à base de dados.	38
24	Diagrama funcional da Interface de Utilizador no modo Hub Wi-Fi.	39
25	Interface de Utilizador (Web) - formulário de autenticação e registo.	40
26	Interface de Utilizador (Web) - gestão de perfil.	42
27	Interface de Utilizador (Web) - gestão de utilizadores.	42
28	Interface de Utilizador (Web) - gestão de sensores.	43
29	Interface de Utilizador (Web) - configuração de sensores.	43
30	Interface de Utilizador (Web) - relatórios de temperatura.	44
31	Interface de Utilizador (Web) - registo de alarmes anteriores.	44
32	Interface de Utilizador (Web) - Registo de alterações de temperaturas	45
33	Diagrama funcional da Interface de Utilizador no modo Hub Celular.	45
34	Fluxo de ações na aplicação móvel	46
35	Interface de Utilizador (App) - introdução do número de telefone da Hub.	47
36	Interface de Utilizador (App) - verificar ou alterar dados de configuração.	48
37	Interface de Utilizador (App) - verificar ou atualizar dados de configuração.	48
38	Interface de Utilizador (App) - visualização de dados de configuração.	49
39	Interface de Utilizador (App) - modificar dados de configuração.	50

40	Interface de Utilizador (App) - modificar Periodicidade ou Temperatura e Contacto.	51
41	Comparação da temperatura ambiente entre exposição ao ar e ao glicol.	53
42	Temperatura registada dentro um frigorífico comparando exposição ao ar e ao glicol.	55
43	Temperatura registada dentro um congelador comparando exposição ao ar e ao glicol.	56
44	Temperatura do ar sentida dentro um congelador.	57
45	Evolução da temperatura perante uma abertura de porta do congelador.	58
46	Deteção do Módulo Celular e do Raspberry.	60
47	Configuração de um novo sensor por parte do MCU e envio dessa informação para o Raspberry. .	61
48	Modificação da temperatura de alarme.	62
49	Periodicidade alterada no Website e, por fim, enviada para Unidade de Sensor.	63
50	Configuração do sensor realizada via aplicação Android.	63
51	Periodicidade alterada via aplicação Android, SMS captado pelo MCU.	64
52	Receção do SMS que solicita o envio das configurações atuais dos sensores.	64
53	Envio de alarme	65
54	Reenvio de SMS de alarme.	66
55	Gráfico com as temperaturas recolhidas para um dado sensor para um intervalo de tempo. . . .	67
56	Página de visualização e configuração de sensores.	67
57	Página com o registo de alterações das temperaturas máximas dos sensores.	68
58	Geração do relatório com o registo de temperaturas.	68
59	Validação da aplicação móvel e notificações de alarme.	69
60	Esquema elétrico da Unidade de Sensor com o hardware final proposto.	71
61	Esquema elétrico da Hub com o hardware final proposto.	73
62	Código implementado - Equação de Steinhart-Hart.	80
63	Código implementado - Reenvio de dados e receção de <i>Acknowledge</i>	80
64	Código implementado - Comunicação UART entre RaspberryPi e MCU.	81
65	Código implementado - Registo de novo utilizador.	82
66	Código implementado - Geração de relatório.	83
67	Código implementado - Modo <i>Access Point</i>	84
68	Código implementado - Envio de dados de temperatura via 802.11.	85
69	Código implementado - Receção de dados de temperatura via 802.11.	85
70	Código implementado - Envio de SMS de pedido de dados de configuração	86
71	Código implementado - Receção de SMS com atuais dados de configuração	86
72	Código implementado - Envio de SMS com os novos dados de configuração	87
73	Características dos produtos - <i>Gateways</i> e sensores	88
74	Características dos produtos - Notificações, política e interface de utilizador.	89

Glossário, Acrónimos e Siglas

A	Ampere, unidade de corrente elétrica
ADC	<i>Analog to Digital Converter</i>
CDC	Centro de Controle e Prevenção de Doença
dB	decibel, unidade de medida de energia ou intensidade numa escala logaritmica
<i>Gateway</i>	Dispositivo central de gestão de dados entre diversos sistema
Hz	Hertz, unidade de medida de frequência
ID	Identificador
<i>Logger</i>	Dispositivo de registo de dados
Microcontrolador	Micro computador num único circuito integrado
PC	”Personal Computer” ou usualmente referido como Computador
V	Volt, unidade de diferença de potencial elétrico
FIFO	<i>First In First Out</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
I/O	<i>Input/Output</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
ISM	<i>Industrial Scientific and Medical</i>
LCD	<i>Liquid Cristal Display</i>
LED	<i>Light Emitting Diode</i>
MCU	<i>Microcontroller Unit</i>
PHP	PHP: <i>Hypertext Preprocessor</i>
RF	Rádio Frequência
RTD	<i>Resistive Temperature Dectector</i>
SMS	<i>Short Message System</i>
SO	Sistema Operativo
SoC	<i>System on Chip</i>
SP	<i>Serial Peripheral Interface</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
USART	<i>Universal Synchronous/Asynchronous Receiver-Transmitter</i>
USB	<i>Universal Serial Bus</i>
Wi-Fi	Dispositivos que utilizam o protocolo de comunicação IEEE 802.11

1 Introdução

Parafraseando o site governamental do Centro de Controle e Prevenção de Doenças (CDC), o armazenamento adequado de vacinas desempenha um papel importante na proteção de indivíduos e comunidades contra doenças evitáveis por vacinação. O armazenamento adequado de vacinas e a durabilidade da sua qualidade é uma responsabilidade compartilhada por todos, desde o momento da sua fabricação até à sua administração^[3].

Uma vacina é uma substância biológica que perde as suas propriedades quando exposta a temperaturas inadequadas. Uma vez perdidas estas propriedades, as mesmas não poderão ser restauradas. De forma a combater a administração de vacinas deterioradas, estas devem ser distribuídas, armazenadas e administradas nas temperaturas recomendadas^[4]. O sangue e os produtos sanguíneos são outro exemplo crítico de uma substância biológica que deve ser armazenada em condições adequadas. Um desvio das faixas de temperatura especificadas durante o armazenamento pode seriamente afetar a viabilidade dos constituintes do sangue^[5].

Um equipamento de armazenamento de bioprodutos é usualmente conhecido por um frigorífico biomédico, sendo que ao longo deste documento refere-se a frigorífico para aplicações de temperatura acima dos 0 °C e congelador abaixo dos 0 °C. Esses frigoríficos diferem na faixa de temperatura de acordo com a aplicação desejada. Por exemplo, um frigorífico de um banco de sangue pode estar definido para 4 °C em casos gerais, mas, no caso de armazenamento de plasma, pode chegar aos -65 °C ^[6] Em alguns países, como nos Estados Unidos da América, é obrigatório o uso de equipamento de registo de temperatura e/ou *logs* para documentar o armazenamento adequado de medicamentos ou outros produtos^[7]. Em Portugal, tem-se o INFARMED como a entidade do Ministério da Saúde que, entre várias competências, é responsável pela segurança da utilização de medicamentos de uso humano^[8]. Este produziu um documento denominado "Boas Práticas de Distribuição"^[9], onde é salientado a importância e necessidade de "existir sistemas de alarme adequados para darem o alerta sempre que se registem desvios em relação às condições de armazenamento predefinidas".

Como mencionado pelo CDC, erros de armazenamento custam milhares de dólares em vacinas desperdiçadas e revacinação^[10]. Um trabalhador da "Helmer Scientific", uma empresa que fabrica equipamentos de refrigeração biomédicos afirmou que um único frigorífico pode comportar entre 2 000 € e 50 000 €^[11] em vacinas e outros produtos de uso hospitalar.

1.1 Motivação e Objetivos

Reconhecendo o problema chega-se à conclusão de que deve haver dispositivos que monitorizem regularmente a temperatura do equipamento de refrigeração. Os dispositivos que se encontram no mercado lidam com este problema com diferentes abordagens, no entanto, existem algumas fragilidades nesses sistemas que podem ser exploradas, levando a uma abordagem nova que pode ultrapassar essas lacunas.

Em primeiro lugar, existe o problema da escalabilidade. Ter um produto que não é adequado para um dimensionamento escalável significa que este pode ser aceitável para uma pequena clínica comunitária mas, num ambiente de maior escala como num centro hospitalar, os custos aumentam rapidamente, inviabilizando a solução. Num centro hospitalar, podem haver dezenas de equipamentos de refrigeração que, dado ao custo financeiro para monitorizar todas essas unidades, seria um investimento insuportável.

O segundo obstáculo é bastante simples, embora como um grande impacto no desempenho geral do sistema de monitorização - a interface de utilizador. Havendo a necessidade de realizar alguma modificação ao produto, é necessário que haja um ecossistema *user friendly* nas interações com o dispositivo físico ou com o *software* que

o suporta.

Finalmente chegamos à economia. Alguns desses dispositivos de monitorização são apenas leitores de temperatura que comunicam, de uma forma sem fios, com um *gateway*. É um processo simples, mas o leitor pode ter um preço acima de 200 € e a *gateway* chegar aos 590 € [12]. Para uma pequena empresa, isso pode não ser uma solução economicamente viável e, para uma instalação grande, tem-se o inerente ao problema da escalabilidade.

O trabalho proposto tem como objetivo principal a conceção e desenvolvimento de um sistema de monitorização de equipamentos de refrigeração biomédicos. O sistema de monitorização consiste num sensor de temperatura que envia informações específicas para uma *hub*. Essa *hub* comunica com um servidor que contém um registo de leituras de cada sensor. Por meio de um *website* e uma *app* para sistema "Android", os utilizadores poderão configurar o seu sistema e visualizar os dados recolhidos.

A forma mais fácil de entender o que se está a propor é passando para um caso de operação: Um sensor ligado a um frigorífico biomédico que faz leituras periódicas ao longo do dia. Essas leituras são enviadas para uma *hub* que determina se os dados de temperatura recebidos estão fora do intervalo de temperatura especificado. No caso da temperatura exceder o limite máximo, haverá um sistema de notificação que notifica o utilizador com a informação de que um determinado frigorífico encontra-se nessas condições de alarme. O objetivo do sistema de notificação é enviar informações críticas para um utilizador quando as condições são verificadas. O sistema global é construído para evitar uma situação em que milhares de euros em substâncias biomédicas são perdidos.

Como mencionado anteriormente, um dos objetivos fundamentais deste trabalho é ter um produto que possa ser escalável, a um preço razoável, e ofereça diferentes opções para diferentes clientes. Tem de ser capaz de trabalhar com pelo menos um sensor, capacidade de notificação, um *website* e/ou aplicativo móvel para visualização de dados e configuração do sistema.

Todos os componentes do sistema devem ser projetados, tendo em mente que, diariamente, não há interação entre o utilizador e o sistema, portanto, existe também o objetivo de ter uma interface de utilizador simples que minimize o tempo quando uma intervenção humana é necessária.

1.2 Desafios

Os desafios para este projeto são diversos, olhando para os desafios relacionados com os sensores em si, estes irão comunicar de uma maneira *wireless* com uma *hub* e a tecnologia escolhida deverá ser a apropriada com a intenção de garantir a comunicação entre os dois dispositivos; Visando uma necessidade de manutenção baixa, a bateria desses sensores tem de aguentar por período de tempo relativamente longo; O sistema tem que ser economicamente viável para que possa ser apelativo no mercado; Um dos objetivos é a notificação do utilizador mediante uma situação de alarme, ou seja, o sistema deve sempre garantir algum tipo de *feedback* neste tipo de situações; O projeto e construção de um sistema que possui um processo simples de configuração e configuração com a menor intervenção humana é possível.

1.3 Estrutura do Documento

Para possibilitar um mais fácil acesso e leitura do presente documento, referente a todo o planeamento e desenvolvimento do sistema de monitorização de temperaturas, o mesmo encontra-se segmentado em sete capítulos, havendo diversos subcapítulos em cada um destes.

Neste primeiro capítulo, todas as introduções são colocadas e esclarecidas, passando por uma fase inicial de

determinar o âmbito da tese. Assim, é exposto algumas considerações em relação à necessidade de monitorizar equipamentos de refrigeração médicos, mencionando a motivação e os objetivos propostos bem como os desafios espetáveis.

Passando de seguida para o segundo capítulo que se refere ao Estado da Arte, aqui são abordados os produtos já existentes no mercado, de um ponto de vista a explorar as características e fragilidades que apresentam, seguido de um resumo referente à tecnologia existente que permitirá alcançar os objetivos propostos da tese de mestrado.

No terceiro capítulo tem-se a Arquitetura do Sistema. Aqui aborda-se a visão global que se pretende para o sistema, passando para uma breve explicação dos diversos constituintes que caracterizam o mesmo. Este encontra-se segmentado em "Unidade de Sensor", "Hub" e "Interface de Utilizador", tendo cada um destes os seus requisitos funcionais e não funcionais descritos a seguir. Introduzido cada um deles, é feita a Especificação do Sistema onde, para cada um dos constituintes do sistema, é descortinado requisito a requisito a implementação proposta. Seguindo com a Arquitetura, dividida em *Hardware* e *Software*, é aprofundado um pouco mais a relação entre blocos individuais de componentes dentro de cada constituinte do sistema, isto é, pegando na Unidade de Sensor como exemplo, parte-se esta num diagrama de blocos que ajudará a segmentar as diversas fases de implementação. O *hardware* selecionado neste capítulo visa uma implementação de "prova de conceito", uma vez que existe a necessidade de avaliar e validar o que se selecionou, originando eventuais modificações de determinados componentes que só serão justificadas com uma análise ao desempenho e caracterização real do *hardware* durante a fase de testes.

Passando para o quarto capítulo, vem a Implementação, aqui são justificadas as decisões tomadas ao longo de todo o processo de desenvolvimento, sendo uma vez mais segmentada a secção pelos três constituintes do sistema global. Caracterizando o *hardware* selecionado e, por meio de diagramas de blocos, o *software* que permite alcançar os requisitos propostos.

Seguindo para o quinto capítulo, Teste do Sistema e Resultados Experimentais, é feita uma análise ao que foi desenvolvido, tanto do ponto de vista de performance como funcional. Neste ponto analisa-se o registo de temperaturas dentro de um frigorífico e congelador doméstico, no contexto da Unidade de Sensor, o funcionamento da Hub nos diversos modos de operação e, por fim, uma análise à Interface de Utilizador com a distinção entre a interface via *web browser* e aplicação móvel.

No sexto capítulo tem-se o Hardware Final que é destinado a evoluir o *hardware* selecionado na Arquitetura do Sistema em função dos resultados obtidos no Teste do Sistema e Resultados Experimentais. Havendo a inerente necessidade de determinar os pontos fortes e fracos do sistema proposto, esta secção permite determinar qual o *hardware* que funcionou daquele que ficou abaixo das expectativas.

Terminado então com o capítulo das Conclusões e Trabalhos Futuros, aqui encontra-se exposto todo o trabalho realizado e críticas às decisões de projeto tomadas. Nesta secção é também discutida os desenvolvimentos futuros.

2 Estado da Arte

Como referido anteriormente, há países com monitorização obrigatória e, por essa razão, existem empresas que comercializam equipamentos nesse sentido. Algumas empresas têm como alvo a indústria de alimentos, instalações industriais ou salas de servidores, apenas uma minoria tem como alvo a indústria biomédica. A secção seguinte foca-se no estudo dos dispositivos existentes no mercado com a identificação dos seus pontos fortes e fracos, aprofundando o contexto da tese desta forma. No final, tem-se uma subcapítulo dedicado à tecnologia de comunicação sem fios disponível atualmente e aos sensores de temperatura.

2.1 Produtos Existentes

2.1.1 Accsense

A "Accsense" oferece o seu "Monitor Vaccine Temperature in Hospitals & Clinics" ^[13] que se resume a um dispositivo que possui um sensor de temperatura e ligação à Internet via Ethernet ou "IEEE 802.11" (Wi-Fi), visto na Figura 1a e Figura 1b, respectivamente. Cada dispositivo pode conectar-se até três sensores de temperatura com fio. Existem dois tipos de sensores de temperatura, um RTD que tem uma gama entre -100 e 100 °C^[14] e um termopar com faixa de temperatura de -200 e 200 °C^[15]. A interface de utilizador só está disponível por meio de um navegador *web*, não havendo nenhum aplicativo móvel. Possuem notificações por falta de energia, onde no caso de ter uma queda de energia na instalação, o utilizador será notificado de que o dispositivo está limitado ao uso de baterias; a notificação derivada de indisponibilidade no acesso à Internet também é algo que é oferecido, sempre que os dispositivos ficam sem esse acesso, os utilizadores recebem uma notificação vinda do servidor. Analisando as características do sistema, o que a "Accsense" fica em falta é em termos de escalabilidade derivado do limite de três sensores de temperatura por dispositivo.



(a) Accsense Ethernet Temperature Data Logger^[15].

(b) Accsense Wireless Temperature Data Logger^[14].

Figura 1: Dispositivos de recolha de dados da Accsense.

A Accsense tem ainda no seu catálogo um dispositivo que regista dados de temperatura, de uma forma *wireless*, que pode ser equipado com um RTD externo para a monitorização de congeladores e frigoríficos. Ao contrário do dispositivo já mencionado, o alvo deste não cinge aplicações biomédicas.

2.1.2 Monnit

Com uma ampla coleção de produtos, a "Monnit" destaca-se da concorrência com uma grande variedade de dispositivos de monitorização à disposição do consumidor. O sistema que oferecem consiste em ter sensores de temperatura que comunicam via rádio (sub 1 GHz) para um *gateway*. Existe um *gateway* celular e um de

Ethernet. O celular é usado para notificações baseadas em telefones móveis, por exemplo, sob certas condições, um SMS é enviado para um utilizador. O *gateway* Ethernet recolhe dados dos sensores de temperatura e envia-os para um servidor onde os utilizadores podem aceder via aplicativo móvel, PC ou navegador *web*. No âmbito das notificações, onde estas diferem das outras empresas é a implementação de um dispositivo que exhibe algumas informações sobre o estado atual dos sensores e, no caso de uma situação alarmante, fornece um alarme sonoro local, antes de qualquer outro circuito de notificação.

Começando com as unidades de deteção, há duas opções disponíveis, transmissão rádio de 1 GHz e Wi-Fi. De acordo com a faixa de temperatura de operação, diferentes sensores podem ser seleccionados, como visto na Figura 2a e 2d respetivamente. Os *gateways*, vistos nas Figura 3a e 3b, podem conter até 50 000 mensagens de sensor para o *gateway* celular e 16 000 para o *gateway* Ethernet. O *gateway* de celular tem uma fonte de alimentação independente "que permite que o dispositivo continue em operação por um período de 24 horas em caso de falta de energia"^[16]. Há outro tipo de *gateway* que comunica diretamente com um PC, no SO Microsoft Windows. Se o PC tiver conectividade com a Internet, as informações recolhidas pelos sensores são sincronizadas com um servidor dedicado.



(a) Sensor de temperatura (-40 °C) - pilha AA^[17].



(b) Sensor de temperatura (-40 °C) - pilha de célula^[18].



(c) Sensor de temperatura (-200°)^[19].



(d) Sensor de temperatura Wi-Fi^[20].

Figura 2: Linha de sensores da Monnit.



(a) Gateway celular^[16].



(b) Gateway Ethernet^[21].

Figura 3: Linha de gateways da Monnit.

Como mencionado anteriormente, a Monnit possui um dispositivo de alerta local que trabalha diretamente com os sensores. Este dispositivo "oferece uma forma adicional de se receber notificações críticas e leituras dos sensores". Quando uma notificação de alarme é recebida, existe um LED que acende intermitentemente, é feito soar um alarme sonoro e exibe as informações atuais no LCD incorporado^[22], Figura 4.



Figura 4: Dispositivo de alerta local da Monnit.^[1].

2.1.3 Phase IV Engineering

A Phase IV Engineering desenvolve sensores personalizados para uma ampla variedade de aplicações, sendo uma delas, a monitorização de congeladores e frigoríficos^[23], não visando os seus produtos para um ambiente biomédico. O sistema é bastante simples, possuindo apenas um sensor, um *gateway* celular ou Ethernet e uma interface de utilizador *web*, onde o utilizador pode definir alertas das mensagens de texto dos SMS, configurar sensores e visualizar dados ^[24]. A Figura 5a representa o *gateway* de Ethernet, a Figura 5b o *gateway* celular e a Figura 5c o sensor de temperatura quando se encontra montado num frigorífico.



(a) Gateway Ethernet^[25].



(b) Gateway celular^[26].



(c) Sensor de temperatura^[27].

Figura 5: Linha de produtos, sensor e gateways, da Phase IV Engineering.

2.1.4 Vaisala

A Vaisala afirma ser a líder mundial neste tipo de monitorizações, concentrando o seu negócio em medições industriais, bem como no clima e meio ambiente^[28]. Na área de medições industriais, têm especificamente uma categoria para aplicações de "Ciências da Vida", com ênfase na "Monitorização de Temperatura em Frigoríficos e Congeladores"^[29]. Pelo que foi encontrado, o sistema Vaisala pode ser bastante complexo, oferecendo diversas opções de escolha. Um dos principais produtos da Vaisala é o "logger" de temperatura, que possui uma bateria interna que pode alimentar o dispositivo até dez anos^[2]. Cada dispositivo pode suportar até quatro sensores de temperatura com fio, dependendo do modelo. Na Figura 6 é mostrado esse dispositivo de recolha de temperaturas com 4 sensores conectados.

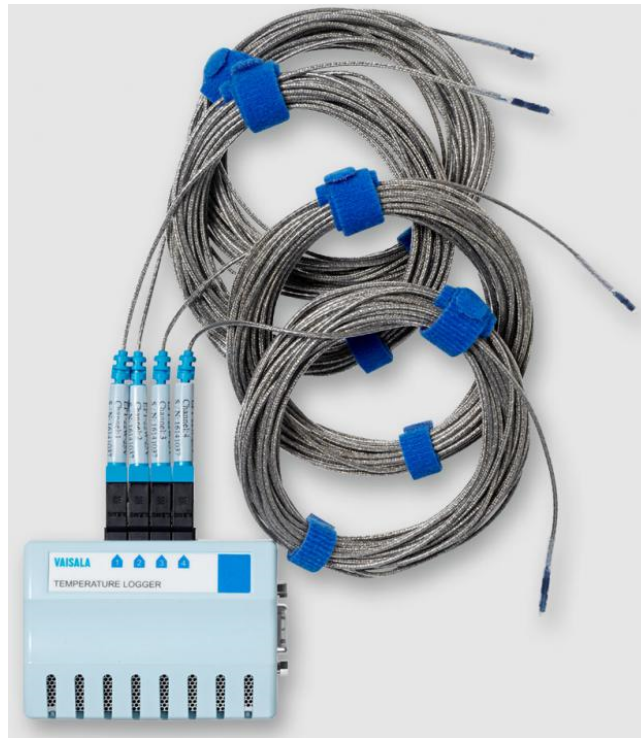


Figura 6: "Logger" de temperatura da Vaisala^[2].

O sistema global é mutável em diferentes opções, pode ser configurado para um único sensor com uma conexão Wi-Fi; um "logger" de dados de temperatura com um *gateway* Wi-Fi ou Ethernet (também há a opção de ter um *gateway* Ethernet com múltiplas portas que pode ser conectado a vários "loggers") e finalmente ter um "logger" de dados de temperatura único com uma conexão direta ao PC via USB^[30]. O maior revés da Vaisala é o preço, tendo apenas o preço para o custo de "Calibração Credenciada", onde o consumidor pode gastar cerca de 460 €^[31].

2.1.5 Sensaphone

Como tantas outras empresas, a Sensaphone permite a monitorização de diversos parâmetros, tais como falha de energia, humidade, água e temperatura. Dando foco à monitorização de temperatura, a Sensaphone tem à disposição o produto "Sensaphone WSG30", Figure 7a, que permite ligar até 30 sensores *wireless*. Este dispositivo permite visualizar dados e gerir configurações com recurso a um serviço *web*. Disponibiliza alertas de voz e notificações de alarme via ligação Ethernet, notificações por e-mail, SMS, Simple Network Management Protocol ou Modbus. O "Sensaphone WSG30" guarda 67 000 amostras que incluem dados, data e tempo. O WSG30 deteta de uma forma autónoma os sensores *wireless*, na Figura 7b tem-se o "WSG Wireless Ultra Low Temperature Sensor" quer permite monitorizar apenas um equipamento de refrigeração^[32]. O pedido de quotação à empresa foi disponibilizado e o preço individual de cada um destes equipamentos resume-se a 640 € pelo WSG30 e 175 € por cada sensor de temperatura.



(a) Estação wireless WSG30 [33].



(b) Sensor de temperatura wireless [33].

Figura 7: Sistema de monitorização da Sensaphone.

2.1.6 Resumo - produtos existentes

De forma a analisar os dados expostos nestas últimas leituras, tem-se no capítulo de Anexos o anexo A.9 com as Figuras 73 e 74 que são um resumo das características dos produtos aqui mencionados. Analisando as mesmas, podemos concluir que todas as empresas têm uma política de subscrição das suas plataformas de interface de utilizador. A Vaisala é a única que oferece algum tipo de solução escalável, mas com a desvantagem dos seus produtos serem desenhados a aplicações industriais e de não possuir uma solução integrada para utilização com uma aplicação móvel. A Monnit oferece algum nível de escalabilidade, mas prende o utilizador a um plano de subscrição e a custos de sensores relativamente elevados. A Phase IV Engineering possui uma autonomia de 5 anos de funcionamento nos seus sensores mas com a revés de ter os custos da unidade de sensor e das *gateways* mais elevados no mercado com ainda a desvantagem de não possuir uma aplicação móvel. A Accsense não possui soluções de sensores sem que estes sejam via Internet, *wireless* ou por cabo de rede. Sendo ainda de mencionar que na sua solução *wireless* oferece uma reduzida autonomia de apenas 4 horas (com uma periodicidade de 1 minuto). A Sensaphone permite conectar até 30 sensores de temperatura à sua *gateway* e visualizar e configurar o seu sistema através de uma interface *web*. Esta tem a desvantagem do sensor de temperatura ter dimensões que dificultem a introdução do mesmo dentro de uma unidade de refrigeração, tendo ainda custos por sensor elevados. Termina-se este resumo com a nota de que os pedidos de cotação efetuados, a Accsense e a VAISALA recusaram-se a providenciar algum tipo de informação, não sendo possível avaliá-las nesta métrica.

2.2 Tecnologia Existente

Abordando então as tecnologias existentes, estas serão divididas em tecnologias de comunicação *wireless* e sensores de temperatura. Debate-se sobre as tecnologias mais conhecidas no meio, explorando as suas características, seguido de um resumo com as vantagens e desvantagens que cada uma oferece.

2.2.1 Bluetooth®

Bluetooth® é uma especificação de curta distância para comunicações baseadas em radiofrequências^[34]. O Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) definiu o *standard* 802.15.1 que serve como base para a tecnologia de comunicação Bluetooth®. De acordo com o "Artigo 5 - Alocações de frequências" da edição de 2012 do Regulamento das Radiocomunicações da União Internacional das Telecomunicações (UIT), Portugal, entre outros países europeus, pode utilizar a gama de frequência 433,05 - 434,79 MHz, bem como 2,4 GHz - 2,5 GHz, que é a frequência de operação da tecnologia Bluetooth®, para usos em aplicações industriais, científicas e médicas (ISM)^[35]. Uma utilização comum desta tecnologia é perante a existência de um dispositivo que desempenha o papel de *master* e outro o papel de *slave*, podendo o *master* comunicar com sete *slaves* no máximo - esta estrutura denomina-se de *piconet*. Por desenho, a tecnologia restringe o performance, desta forma a sua aplicabilidade limita-se a redes de sensores sem fios^[34].

O Bluetooth Low Energy® (BLE) é uma versão de baixa potência da tecnologia Bluetooth® clássica^[36]. A diferença entre estas duas versões é que a clássica é "orientada à conexão" com um intervalo de conexão fixo, ideal para *links* de alta atividade, o BLE, por outro lado, para reduzir o consumo de energia, emprega um intervalo de conexão variável que pode ser definido de alguns milissegundos a vários segundos^[36]. O baixo consumo energético é garantido ao maximizar o tempo de *standby*, um curto tempo de conexão entre dispositivos e um pico de potência baixo na transmissão/receção de dados^[36].

2.2.2 ANT

Em "Comparisons Between Low Power Wireless Technologie", de Phill Smith (2011), afirma que a ANT é uma tecnologia sem fios proprietária de baixa potência que opera na gama de frequência de 2,4 GHz. Normalmente, o dispositivo transceptor ANT é tratado como uma *black box* que não deve exigir grande esforço do ponto de vista de *design* para implementação numa rede. Refere ainda que, à semelhança com o BLE, os dispositivos ANT podem operar durante largos anos com uma simples bateria de célula^[37].

2.2.3 Zigbee®

O Zigbee® é a tecnologia mais popular em implementações de rede do tipo *mesh* na conexão de sensores, instrumentação, e sistemas de controlo^[38]. O Zigbee® é baseado na especificação IEEE 802.15.4 que define redes sem fio de baixo consumo de energia, baixo *duty cycle* e uma taxa de transmissão de dados reduzida. A camada física IEEE 802.15.4 adotada pelo Zigbee® foi projetada para a banda de 868 MHz na Europa; 915 MHz na América do Norte e 2,4 GHz como a banda global aceita em quase todos os países ^[38].

2.2.4 Wi-Fi®

Wi-Fi é uma tecnologia que permite que vários dispositivos comuniquem sem fio à Internet usando ondas de rádio ^[39]. A "Wi-Fi Alliance" diz que dispositivos Wi-Fi são quaisquer produtos de rede local sem fio (WLAN) baseados na especificação IEEE 802.11 ^[39]. O padrão 802.11 define um controle de acesso ao meio e especificações da camada física para implementações de rede WLAN.

2.2.5 Resumo - Comunicação sem fios

A Tabela I é um apanhado das características de maior importância das diferentes tecnologias discutidas. A tecnologia de Bluetooth e BLE tem a limitação de apenas conseguirem comunicar com um número reduzido de dispositivos, com a agravante do Bluetooth tradicional ter uma corrente de transmissão de 60 mA^[38]. Em relação ao Wi-Fi, esta tem a corrente de transmissão mais elevada, chegando aos 400 mA^[38], bem como uma corrente em *standby* igualmente elevada. As tecnologias expostas com as características mais apelativas são a Zigbee, com o seu grande alcance de comunicação e relativa baixa corrente de transmissão e a ANT com a corrente de transmissão mais baixa mas ao custo de uma menor distância de comunicação.

Tabela I: Apanhado das tecnologias de comunicação wireless.

Parâmetro	Bluetooth	BLE	Zigbee	ANT	Wi-Fi
Gama de frequência [GHz]	2.4	2.4	2.4	2.4	2.4 to 5
Especificação	IEEE 802.15.1	IEEE 802.15.1	IEEE 802.15.4	ANT	IEEE 802.11
Alcance [metros]	10 ^[40]	60 ^[41]	70 to 300 ^[40]	30 ^[40]	50 to 100 ^[40]
Corrente (transmissão - <i>standby</i>)	60 mA - NA ^[38]	4.5 mA - 0.78 μ A ^[42]	9.3 mA - 4.18 μ A ^[42]	2.9 mA - 3.1 μ A ^[42]	400 mA - 20 mA ^[38]
Número máximo de <i>nodes</i>	7 ^[38]	3 ^[43]	64 000 ^[38]	-	32 ^[38]

2.2.6 Sensores de temperatura

Citando o *website* da empresa de renome National Instruments, existe uma "variedade de sensores que convertem fenómenos de temperatura em sinais mensuráveis. Os três sensores mais comuns são o termopar, RTD e o termistor"^[44]. É ainda providenciado um resumo das características destes sensores, Tabela II. Nesta pode-se concluir que termopar tem uma grande necessidade de acondicionamento de sinal e uma sensibilidade reduzida, mas oferece as vantagens de não possuir o fenómeno de auto aquecimento e o custo mais baixo. O RTD tem a desvantagem de necessitar de um ajuste à sua resistência primária bem como deter o custo mais elevado dos três tipos de sensor. Por fim, o termistor, tem uma grande sensibilidade às diferenças de temperatura com uma precisão média, um custo baixo e poucos requisitos de acondicionamento de sinal, mas um auto aquecimento bastante elevado.

Tabela II: Características dos diferentes tipos de sensores de temperatura.

Característica	Termopar	RTD	Termistor
Gama de Temperatura	-210 °C to 1760 °C	-240 °C to 650 °C	-40 °C to 250 °C
Linearidade	Mau	Bom	Mau
Sensibilidade	Baixo	Médio	Muito Alto
Tempo de Resposta	Médio a Rápido	Médio	Médio a Rápido
Estabilidade	Mau	Bom	Mau
Precisão	Médio	Alto	Médio
Auto Aquecimento	Não	Sim, mínimo	Sim, altíssimo
Durabilidade	Excelente	Bom	Mau
Custo	Mais baixo	Alto	Baixo
Requisitos de Acondicionamento de Sinal	Compensação de <i>Cold-Junction</i> ; Amplificação; Detecção de termopar aberto; Escalabilidade	Excitação; Correção de resistência primária; Escalabilidade	Excitação; Escalabilidade

3 Arquitectura do Sistema

3.1 Visão global do Sistema

Chegado a este ponto, quem lê o presente documento pode ainda não ter a noção global do sistema, posto isto, segue-se um breve sumário. O sistema visa a monitorização de equipamentos de refrigeração médicos com o objetivo de alertar um utilizador quando a temperatura dentro do mesmo excede determinados valores. Para este efeito é necessário ter um sensor para recolher a temperatura e uma unidade de processamento que permite enviar notificações de alarme. Posteriormente vem a necessidade de ter uma interface de utilizador que permita configurar o sistema quando necessário. O sistema encontra-se então dividido em três secções, a "Unidade de Sensor", "Hub" e "Interface de Utilizador".

Unidade de Sensor

A Unidade de Sensor tem um mecanismo de funcionamento simples. Tem como função a leitura periódica da temperatura do meio onde se encontra e estabelecer comunicação com a Hub afim de enviar esses dados.

Hub

A Hub é o cérebro das operações uma vez que é quem controla os sensores e é quem determina como vai ser a "Interface de Utilizador". A principal característica da Hub é a sua modularidade, possuindo uma arquitetura modular com modos de operação distintos. Nomeando esses modos de operações, temos "Hub Repeater", "Hub Cellular", "Hub Wi-Fi" e "Hub Cellular & Wi-Fi". A Hub estabelece uma rota de comunicação com os sensores e, consoante os módulos associados, trata de enviar essas informações para a Interface de Utilizador.

Interface de Utilizador

Resume-se a uma página *web* onde é permitido visualizar dados referentes às leituras de temperatura e alarmes ocorridos, bem como a configuração dos sensores conforme desejado. Esta interface está inteiramente ligada à "Hub Wi-Fi". Existe ainda outro tipo de interface mais simples que é alcançado através de uma aplicação móvel para Android. Esta aplicação permite ver o estado de configuração dos sensores e reconfigurar os mesmos quando necessário. Em paralelo, a aplicação de SMS que vem por omissão com o sistema operativo Android, servirá para receber e ler os SMS de alarme.

3.2 Requisitos do Sistema

Abaixo, as tabelas III a VIII mostram os requisitos funcionais e não funcionais das três secções mencionadas. Cada requisito é numerado e tem uma prioridade associada, sendo que a prioridade mais elevada corresponde ao valor "1".

Unidade de Sensor

Tabela III: *Requisitos funcionais do Sensor.*

Número	Prioridade	Descrição
1	1	Leitura de temperaturas entre -30 °C e 20 °C
2	1	Comunicação externa é assegurada de forma <i>wireless</i> .
3	1	Precisão de temperatura menor que ± 1 °C.
4	1	Leituras periódicas programáveis entre 15 minutos e 24 horas.
5	2	Bateria com capacidade para funcionamento ininterrupto durante dois anos.

Tabela IV: *Requisitos não funcionais do Sensor.*

Número	Prioridade	Descrição
1	1	Fonte de alimentação fornecida por uma bateria DC não recarregável
2	1	<i>Socket</i> para antena.
3	1	Frequência de comunicação na faixa de frequência ISM.
4	2	Temperatura de operação entre -40 °C a 40 °C.
5	2	Encapsulamento resistente a produtos de limpeza.

Hub

Tabela V: *Requisitos funcionais da Hub.*

Número	Prioridade	Descrição
1	1	Implementação modular que resulta em diferentes finalidades para Hub.
2	1	Conecção a uma rede Wi-Fi.
3	1	Conecção a uma rede celular.
4	1	<i>Host</i> de um servidor <i>web</i> .
5	1	Comunicação <i>wireless</i> com os sensores.
6	1	Servidor <i>Web</i> permite que os utilizadores efetuem login, gerenciem sensores e visualizem históricos.
7	1	Comunicação indireta com o Sensor por meio de uma rede celular ou Wi-Fi.
8	2	Fonte de alimentação fornecida a partir da rede e bateria de reserva.

Tabela VI: *Requisitos não funcionais da Hub.*

Número	Prioridade	Descrição
1	1	Comunicação com sensores dentro da faixa de frequência ISM.
2	1	Protocolo 802.11n nas comunicações Wi-Fi.
3	1	Rede móvel com tecnologia 3G.
4	2	Capacidade de hospedar pelo menos 30 sensores.
5	3	Armazenamento mínimo de 5 760 mensagens provenientes de sensores.

Interface de Utilizador

Tabela VII: *Requisitos funcionais da Interface de Utilizador.*

Número	Prioridade	Descrição
1	1	Servidor Web: Login de utilizador.
2	1	Servidor Web/Aplicação móvel: Definir temperatura de alarme do sensor.
3	1	Servidor Web/Aplicação móvel: Definir que utilizador recebe o alarme.
4	1	Servidor Web: Visualização de alarmes de temperatura.
5	1	Servidor Web: Visualização de alarmes anteriores.
6	1	Servidor Web/Aplicação móvel: Definir a periodicidade das leituras de temperatura de cada sensor.
7	2	Servidor Web: Registo das alterações de alarmes de temperatura.
8	2	Servidor Web: Geração de relatório com as leituras de temperatura.
9	2	Servidor Web: Utilizadores com privilégios de administrador.
10	2	Servidor Web: Permitir que os administradores ativem/desativem contas de utilizadores.
11	3	Servidor Web: Gerenciamento do perfil de utilizador.

Tabela VIII: *Requisitos não funcionais da Interface de Utilizador.*

Número	Prioridade	Descrição
1	1	Acesso à Interface de Utilizador por meio de um navegador da web/aplicativo móvel.
2	1	Aplicação para dispositivos móveis com Android OS.
3	2	Interface de Utilizador simples e intuitiva.

3.3 Especificação do Sistema

3.3.1 Unidade de Sensor

Olhando para os requisitos funcionais e não funcionais da Unidade de Sensor, Tabela III e IV, é de notar que não existe nenhum requisito no que toca às dimensões do mesmo. Havendo apenas o cuidado de não ultrapassar

um volume máximo de 0,5 litros, uma vez que é pretendido a introdução deste subsistema dentro das unidades de refrigeração. É ainda importante referir o ambiente hostil onde se encontrará o dispositivo. Com uma gama de operação que atinge um valor mínimo de $-40\text{ }^{\circ}\text{C}$ é de extrema importância assegurar materiais que sejam capazes de resistir aos possíveis efeitos degradantes inerentes a estas condições. Atendendo ao requisito número 3 da tabela de requisitos não funcionais, "Frequência de comunicação na faixa de frequência ISM", salienta-se o referido no documento providenciado pelo *European Communications Office* em que é explicitado que a gama de frequência para aplicações ISM (*Industrial, Scientific and Medical*) compreende as frequências 433.05 a 434.79 MHz^[45].

3.3.2 Hub

A Hub, apesar de ser o "cérebro" da tese proposta, não possui muitas especificações técnicas. Da mesma forma que a Unidade de Sensor referida anteriormente, a Hub comunica com os sensores na faixa de frequência ISM, entre 433.05 a 434.79 MHz. Quando esta se encontrar no modo Hub Wi-Fi, o protocolo de comunicação será o estabelecido pela IEEE (*Institute of Electrical and Electronics Engineers*), nomeadamente o 802.11n ou compatível. Estando no modo Hub Celular, a tecnologia para a rede móvel será a de terceira geração (3G). Menciona-se ainda o requisito de armazenamento 5 760 mensagens de sensores que vem do cálculo de 30 sensores com uma periodicidade de 15 minutos a comunicar com a Hub durante 48 horas.

3.3.3 Interface de Utilizador

Este último subcapítulo da caracterização do sistema diz respeito ao subsistema com o qual o utilizador tem mais contacto, a "Interface de Utilizador". Esta encontra-se dividida em dois paradigmas, interface no navegador *web* e interface no aplicativo móvel. Na interface via *web browser*, é necessário realizar uma autenticação (*login*) onde haja a diferenciação entre utilizadores comuns e utilizadores com privilégios de administrador. O utilizador pode alterar a configuração atual do sistema, visualizar gráficos de temperatura, alarmes anteriores, gerar um relatório com a informação das temperaturas, visualizar as informações de todos os sensores integrados no sistema e as suas configurações atuais. Um utilizador que possua privilégios de administrador é responsável por ativar ou desativar outras contas de utilizador. Referenciando a Tabela VII, tecem-se algumas considerações abaixo.

- Requisito Nr. 1: Para efetuar *login* é necessário um registo prévio no sistema. Para registar o utilizador deverá preencher os campos "Username", "First Name", "Last Name", "Department", "Contact" e "Password". A *password* de acesso será encriptada por razões de segurança.
- Requisito Nr. 2: A temperatura de alarme não possui casa decimal.
- Requisito Nr. 3: Com a Hub em modo "Hub Celular", pela aplicação móvel não existirão utilizadores, será apenas necessário introduzir o número de telefone de quem fica responsável por receber SMS de alarme.
- Requisito Nr. 7: Será de registar as alterações aos alarmes de temperatura, guardando a informação apenas em base de dados. Neste registo de alterações deverá ser possível verificar qual era a temperatura de alarme antes da alteração, a temperatura de alarme depois da alteração e que utilizador efetuou a alteração.

- Requisito Nr. 8: O relatório com as leituras de temperatura tem de ter o formato ".xls" ou ".xlsx" onde deverá conter um *timestamp*, a temperatura e a identificação do sensor.
- Requisito Nr. 11: O utilizador poderá apenas alterar os campos referentes ao "Departamento", "Contacto" e "Password". Sendo que para a última seja necessário introduzir a *password* atual para que se possa proceder a alteração para uma nova *password*.

3.4 Arquitectura

3.4.1 Hardware

Unidade de Sensor

A arquitetura da Unidade de Sensor, a mais simples encontrada no sistema, baseia-se num transdutor de temperatura, um microcontrolador, um módulo de RF e uma fonte de alimentação, Figura 8. O transdutor e o microcontrolador estabelecem uma comunicação unidirecional, dado que o transdutor é um componente passivo. A unidade de sensoriamento pode ser reconfigurada no que toca ao intervalo entre medições de temperatura, logo é necessária uma comunicação bidirecional entre o microcontrolador e o módulo RF. Os requisitos exigem um dispositivo alimentado por uma bateria, com capacidade suficiente para durar dois anos. Este requisito será melhor aprofundando na secção 4.1 que se refere à implementação do hardware escolhido.

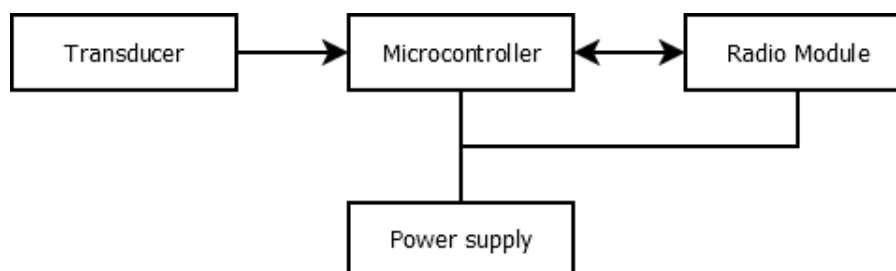


Figura 8: Diagrama de blocos - Unidade de Sensor.

Hub

O primeiro requisito de máxima prioridade da Hub é a implementação de uma arquitetura modular que origine diferentes funcionalidades para a Hub, Figura 9.

A "Hub Repeater" tem o objetivo de impedir a situação em que dois equipamentos de refrigeração encontram-se distantes um do outro e a comunicação por RF não é possível, dada a longa distância. A "Hub Repeater" permite que as Unidades de Sensor, que se encontram nessa situação, comuniquem com a Hub principal por meio de uma rede Wi-Fi (protocolo 802.11n). A Hub principal nesta situação seria uma "Hub Celular" ou "Hub Wi-Fi".

Em relação à "Hub Celular", existem duas abordagens que permitem interpretar a sua funcionalidade. O "Módulo Celular" fornece uma conectividade celular com uma tecnologia 3G. Um "Módulo Wi-Fi" pode ser adicionado a esta Hub, complementando o cenário em que o utilizador requer um "Hub Repeater" dada a disposição das Unidades de Sensor. O utilizador comunica indiretamente com o sensor, por meio da rede celular, enviando SMS's com uma nova configuração ou recebendo notificações em situações de alarme.

A "Hub Wi-Fi" funciona como um servidor *web* no qual um *website* é hospedado. Este site pode ser acessado pelo utilizador através de um navegador *web* convencional. Sempre que for necessário efetuar alguma configuração ao sistema, o utilizador acede ao site, efetua *login* e procede às ações de configuração ou visualização dos parâmetros do sistema. Aquando da presença do "Módulo Wi-Fi" na Hub, sempre que chegar novos dados de temperatura à Hub serão enviados para o "Módulo *Webserver*" onde serão guardados numa base de dados. Quando é feita uma nova configuração através do *website*, essa configuração é guardada na memória interna da Hub e na base de dados do "Módulo *Webserver*". Só quando a Hub confirmar que recebeu as novas configurações é que se gravar efetivamente na base de dados.

O modo mais complexo em que a Hub pode operar é quando existe um "Módulo *Webserver*" e um "Módulo Celular" acoplado. Quando existe a intenção de configurar sensores neste modo, o utilizador deverá recorrer ao *website* para o efeito. O "Módulo Celular" só servirá para o envio de alarmes de temperatura. Quando for recebido um alarme de temperatura, poderá ser enviado um SMS de resposta para que se desabilite, temporariamente, os alarmes do referido sensor.

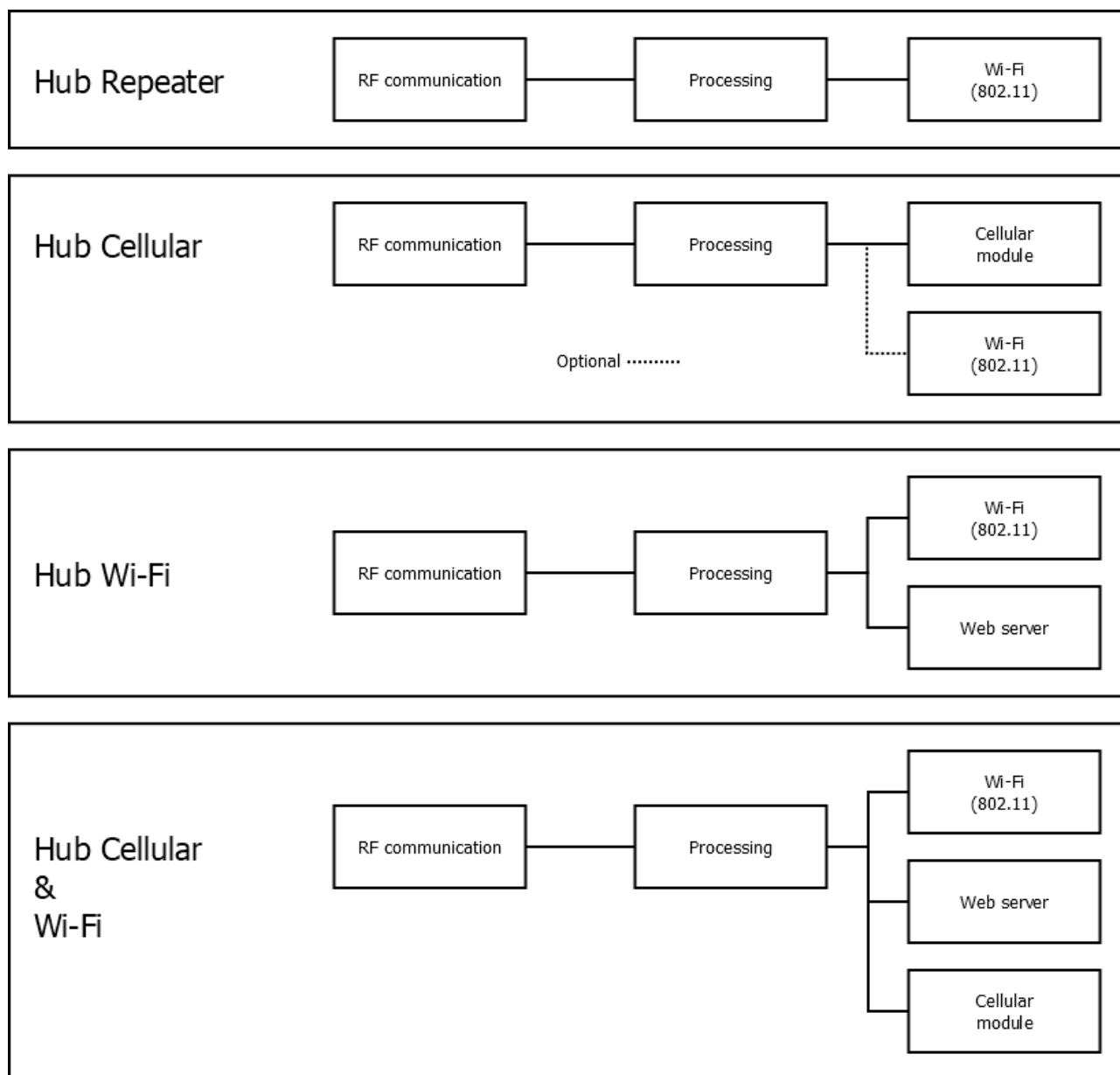


Figura 9: Diagrama de blocos - Hub.

Interface de Utilizador

A Figura 10 fornece uma representação visual da arquitetura do sistema no modo "Hub WiFi". O cliente e o servidor comunicam por meio de pedidos e respostas HTTP (*Hypertext Transfer Protocol*). No lado do servidor, existe uma base de dados que tem todos os dados que o utilizador visualiza. No que toca à interface de utilizador aquando de uma "Hub Celular", esta não irá funcionar "online", isto é, a aplicação e a Hub comunicam por meio de trocas de SMS, Figura 11. Este mecanismo é assíncrono logo, quando a Hub procede a um envio de SMS na eventualidade de uma situação de alarme, este é enviado para o número de telefone configurado. No que toca à visualização de alarmes, esta será levada a cabo pela aplicação móvel de SMS que vem com o sistema operativo do dispositivo. Uma aplicação móvel dedicada é indispensável para uma melhor visualização da configuração do sistema e, quando se pretender reconfigurá-lo, o envio de um SMS, via aplicação desenvolvida, com os dados dessa nova configuração.

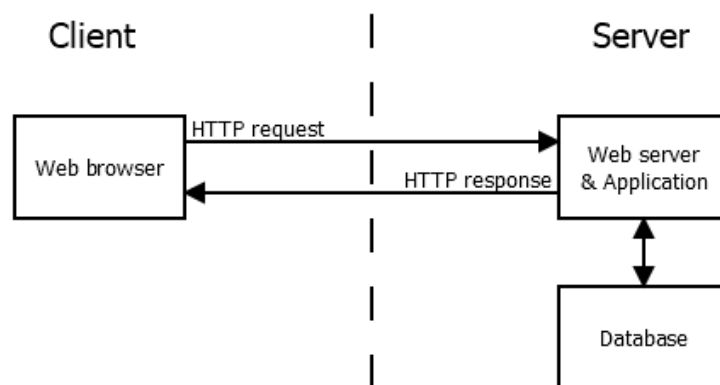


Figura 10: Via de comunicação entre cliente e servidor.

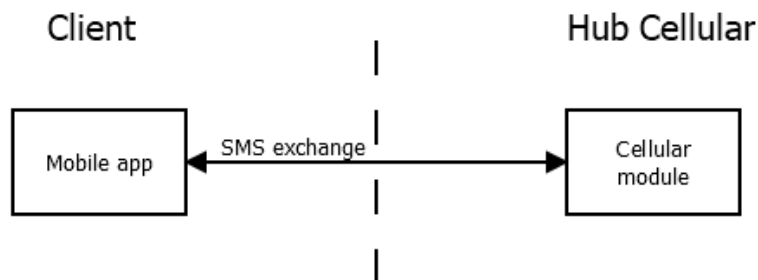


Figura 11: Via de comunicação entre aplicação móvel e Hub.

3.4.2 Software

Unidade de Sensor

O fluxo lógico da Unidade de Sensor não carece de grande complexidade, Figura 12. Este terá um processo de inicialização onde irá, por exemplo, configurar o "Módulo Rádio" com determinados parâmetros. Seguidamente, é iniciado um ciclo periódico em que é feita uma leitura da temperatura e enviada para a Hub. De forma a assegurar que a Hub recebeu de facto a informação, esta aguardará por *feedback*.

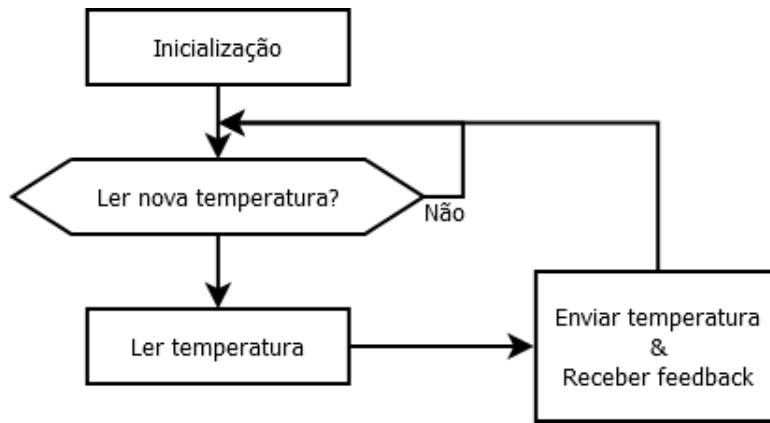


Figura 12: Diagrama funcional do software da Unidade de Sensor.

Hub

A Hub realiza uma inicialização em si semelhante àquela por parte da Unidade de Sensor, seguido de uma ação que a leva a determinar o seu tipo, ou seja, existe a necessidade de determinar que módulos se encontram associados à mesma, uma vez que isso condiciona o seu comportamento. Concluído este processo, a Hub fica "à escuta" de dois acontecimentos - receção de dados provenientes do "Módulo Rádio" ou dos módulos que lhe possam estar acoplados. Consoante os dados recebidos via RF, será enviada a apropriada resposta. Uma vez que os módulos podem comunicar com a Hub assincronamente, é necessário realizar constantes ações de verificação da receção de dados provenientes dos mesmos, Figura 13.

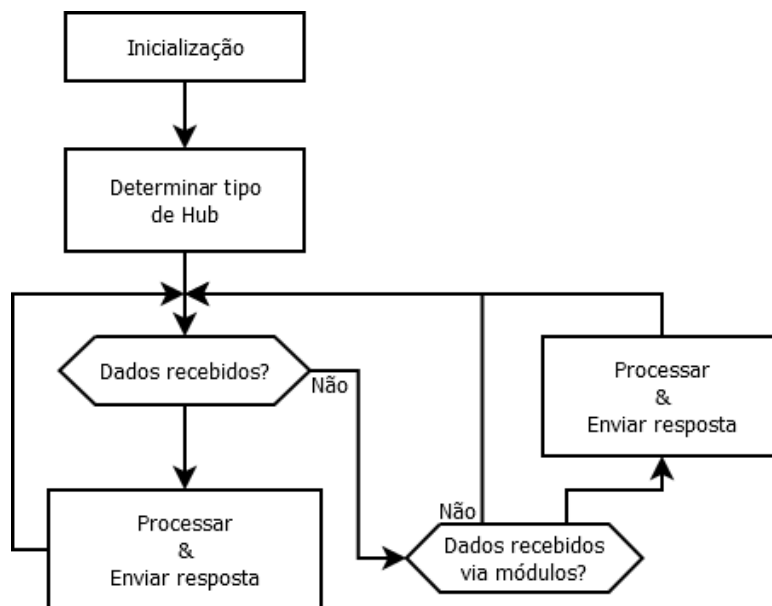


Figura 13: Diagrama funcional do software da Hub.

Interface de Utilizador

A interface de utilizador vem em dois modos de utilização como já explicado anteriormente. Quando a Hub é uma Hub Wi-Fi, a interface de utilizador é ao nível de um *browser*, Figura 14. Por motivos de segurança, é fundamental existir um método de autenticação antes de se poder efetuar alterações à configuração do sistema. Após esta etapa, o utilizador poderá realizar diversas ações que vão desde a gestão do seu perfil à configuração

dos sensores. Aquando de uma Hub Celular, o utilizador estará limitado a três ações, nomeadamente, configurar sensores, visualizar configurações de sensores ou alterar o número de telefone da Hub (Figura 15). Se pretender configurar o sensor, terá à sua disposição a alteração da periodicidade de leituras de temperatura, alterar o número de telefone associado ao sensor em configuração ou a temperatura máxima permitida.

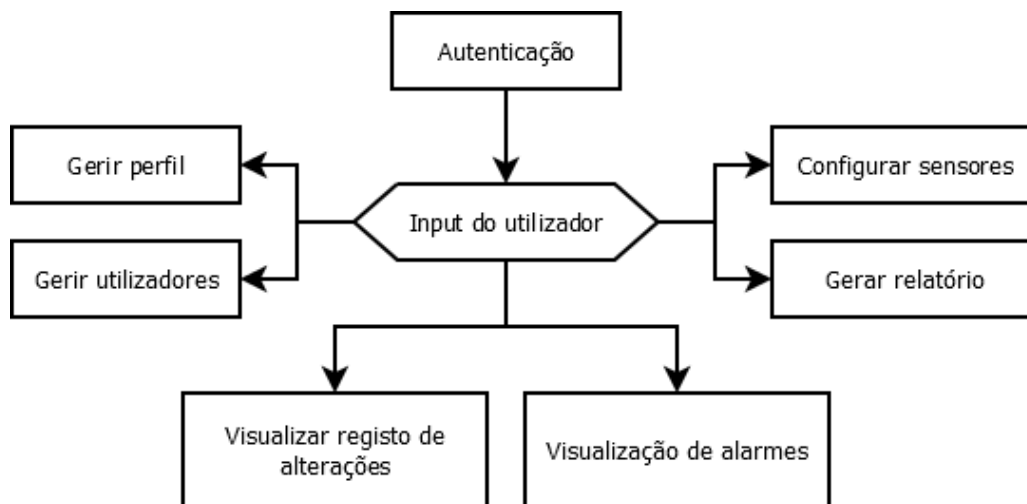


Figura 14: Diagrama funcional da Interface de Utilizador no modo Hub Wi-Fi.

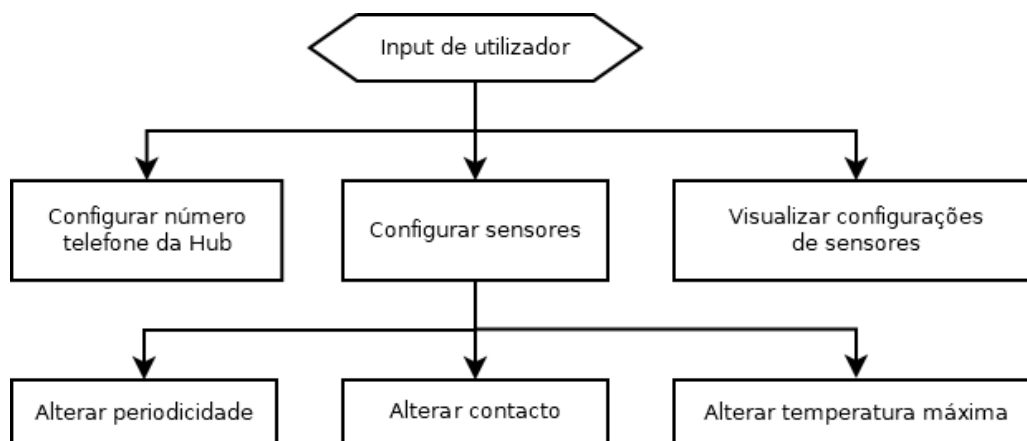


Figura 15: Diagrama funcional da Interface de Utilizador no modo Hub Celular.

4 Implementação

A presente secção visa documentar e justificar as decisões de implementação realizadas ao longo da fase de desenvolvimento. Havendo sempre a separação entre Unidade de Sensor, Hub e Interface de Utilizador, esta secção estará segmentada de acordo com essas unidades.

4.1 Unidade de Sensor

A Unidade de Sensor é o componente crítico deste projeto, terá que trabalhar em condições não ideais de temperatura, humidade e comunicação RF, sendo então de extrema importância testar a viabilidade do mesmo. Pensando na temperatura que se pretende expor a Unidade de Sensor, é crucial ter em conta a gama de operação de temperatura dos diversos componentes. Tendo em mente que existe uma temperatura ideal de operação para uma bateria, nos casos de temperaturas baixas, pode não ter uma diferença de potencial suficiente nos seus terminais para alimentar os diversos circuitos.

No que toca ao diagrama de blocos deste subsistema, este foi previamente discutido na secção 3.4.1 mas, para conveniência do leitor, é novamente apresentado, Figura 16. O hardware selecionado para o desenvolvimento desta unidade encontra-se abaixo e seguidamente tem-se algumas considerações e características dos componentes.

- Microcontrolador: "MSP430G2452" da Texas Instruments Inc.
- Módulo Rádio: "RFM69HCW" fabricado pela "HopeRF"[®] com um "breakout" da Adafruit Industries.
- Transdutor: Termístor de 1 k Ω a 25 °C.
- Fonte de alimentação: Quatro pilhas AA alcalinas, MN1500 da Duracell[®].

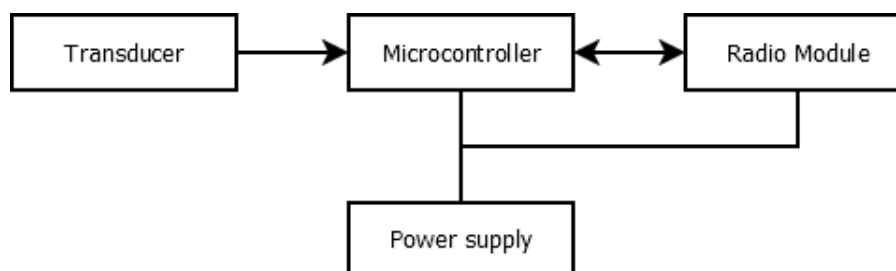


Figura 16: Diagrama de blocos - Unidade de Sensor.

Hardware selecionado e algoritmo

O microcontrolador escolhido possui modos de funcionamento de baixo consumo, diversos portos de I/O (Input/Output) e interfaces de comunicação, nomeadamente USART (*Universal Synchronous/Asynchronous Receiver-Transmitter*). Uma vez que possuía todas características descritas e dada a disponibilidade em laboratório, a escolha deste microcontrolador tornou-se viável e justificada.

Para a comunicação RF, foi identificado o SoC (System on Chip) RFM69HCW da "HopeRF", com um *breakout* desenhado pela "Adafruit Industries". As características principais deste SoC referem-se à utilização da banda de frequência de 433.05 a 434.79 MHz, uma sensibilidade de -148 dBm e uma temperatura de operação

compreendida entre -55 e $+115$ °C. Havendo a necessidade de garantir que o *breakout* escolhido possui as características necessárias, estudou-se os componentes que o compõem. Na Figura 17^[46], tem-se o esquema do *breakout*, e nesta identifica-se o SoC mencionado, alguns conectores, componentes passivos, um conversor de tensão não inversor (74HC4050D) e um regulador de tensão (AP2112K). O conversor de tensões atua sobre os pinos de *input* e limita a tensão de entrada dos *inputs* do SoC. Referenciando o *datasheet* da "NXP Semiconductors"^[47], a temperatura de operação compreende os -40 °C e $+125$ °C e a tensão de alimentação entre 2 V e 6 V. No que toca ao regulador de tensão, retira-se do seu *datasheet*^[48] uma temperatura de operação compreendida entre -40 e 85 °C e, para uma tensão de alimentação de 2,8 V, tem-se uma tensão regulada de 1,8 V. Uma vez que o conversor de tensões necessita de 2 V, é necessário fornecer uma tensão superior ao regulador de tensão. Posto isto, o regulador de tensão teria que ser alimentando com uma tensão acima de 2,8 V de forma a fornecer pelo menos 2 V à saída. Realizaram-se testes funcionais simples, para aferir o correto funcionamento do *breakout*, e constatou-se que alimentando o regulador de tensão apenas com 3 V, o *breakout* comporta-se como esperado, contudo, no par "Módulo Rádio" e microcontrolador o comportamento diferia. Na presença de duas baterias em série de 1,5 V cada, o circuito tinha um comportamento instável. Quando o "Módulo Rádio" iniciava o processo de transmissão, havia consumos elevados - picos de corrente que levavam a uma queda de tensão na alimentação do circuito e isto originava que o microcontrolador desliga-se momentaneamente, "reiniciando" o mesmo. Testou-se a mesma implementação mas com dois conjuntos em paralelo, com duas baterias de 1,5 V em série cada um. Assim, a corrente fornecida por série reduziu para metade e o problema identificado foi mitigado.

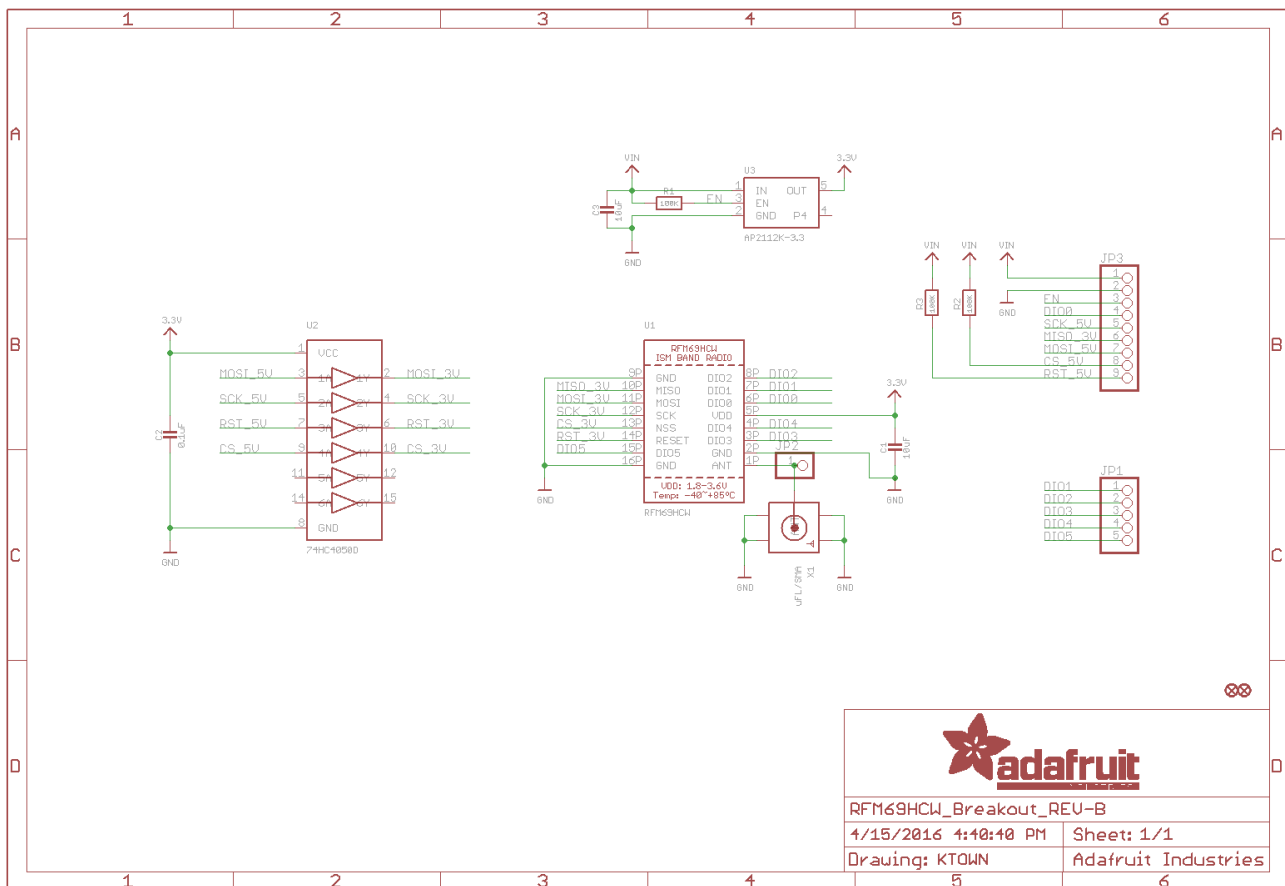


Figura 17: *Breakout*, desenhado pela "Adafruit Industries", para o SoC RFM69HCW.

Na figura 18, vê-se que se encontra um filtro passa-baixo entre o pino 16 e a alimentação do microcontrolador.

Este filtro permite evitar que ocorra o fenômeno de *bouncing* no pino de *reset* quando se liga o dispositivo, prevenindo comportamentos não desejados. A frequência de corte é dada por $f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 100 \times 10^3 \times 1 \times 10^{-6}} \cong 1.6 \text{ Hz}$.

Das diversas escolhas possíveis para o transdutor de temperatura, optou-se por um termistor (B57164K) produzido pela "TDK Epcos"^[49], que apresenta uma resistividade de 1 kΩ a 25 °C. Este tipo de transdutor varia a sua resistividade linearmente com a temperatura. Havendo a relação entre a resistividade e a temperatura, é possível calcular a temperatura com base na equação de *Steinhart-Hart*(5). Para ser calculada a temperatura é necessário que o microcontrolador consiga determinar a resistividade. Para isto utiliza-se o conversor analógico-digital do microcontrolador que, com meio a um divisor de tensão, dá-nos a relação entre temperatura e tensão. Sendo V_{Ref+} a tensão que alimenta o divisor de tensão, a saída do divisor de tensão, V_O , é dada pela equação (1). O divisor de tensão é composto por duas resistências, o termistor e uma outra resistência de valor fixo e conhecido. Foi então selecionada uma resistência de 1 kΩ de forma a que V_O seja metade de V_{Ref+} quando a temperatura for 25 °C.

$$V_O = \frac{R}{R + 1K} \times V_{Ref+} \quad (1)$$

Com V_{ARef} a ser a referência positiva do ADC, o valor derivado da conversão, ADC_{Value} , tendo o mesmo 10-*bits*, é visto em (2).

$$ADC_{Value} = V_{ADCInput} * (2^{10} - 1) / V_{ARef} \quad (2)$$

Substituindo V_O da equação (1) por $V_{ADCInput}$ da equação (2) ficamos com a relação entre o valor binário da conversão com a tensão no divisor de tensão, (3).

$$ADC_{Value} = \frac{R}{R + 1K} \times V_{Ref+} \times \frac{1023}{V_{ARef}} \quad (3)$$

Se o valor de referência analógica positiva, V_{Ref} , for igual à tensão que alimenta o divisor, V_{ARef} , ficamos com um valor de conversão que é independente da tensão de alimentação do divisor de tensão, (4).

$$ADC_{Value} = \frac{R}{R + 1K} \times 1023 \quad (4)$$

Para converter o valor binário em temperatura, recorre-se à equação de *Steinhart-Hart* (5). Contudo, esta é complexa e requer parâmetros do termistor que não se encontram documentados no seu *datasheet*^[49], como tal, recorre-se à equação do parâmetro β (6), que se trata de uma simplificação da equação de *Steinhart-Hart*. Nessa equação (6) a temperatura, T , é a que se pretende determinar. T_0 é a temperatura nominal da resistência, ou seja, 25 °C. O valor de R é resistividade do termistor no momento de leitura e R_0 é o valor da resistência à temperatura nominal. O parâmetro β é específico de cada termistor, como tal, recorrendo ao *datasheet*^[49] do mesmo retirou-se que $\beta = 3730 \pm 3\%$.

$$\frac{1}{T} = a + b \times \ln(R) + c \times (\ln(R))^3 \quad (5)$$

Para $a = \frac{1}{T_0} - \frac{\ln(R_0)}{\beta}$, $b = \frac{1}{\beta}$ e $c = 0$ tem-se:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} \times \ln\left(\frac{R}{R_0}\right) \quad (6)$$

No capítulo de Anexos, tem-se o anexo A.1 com a Figura 62 com o excerto do código implementado da equação (6).

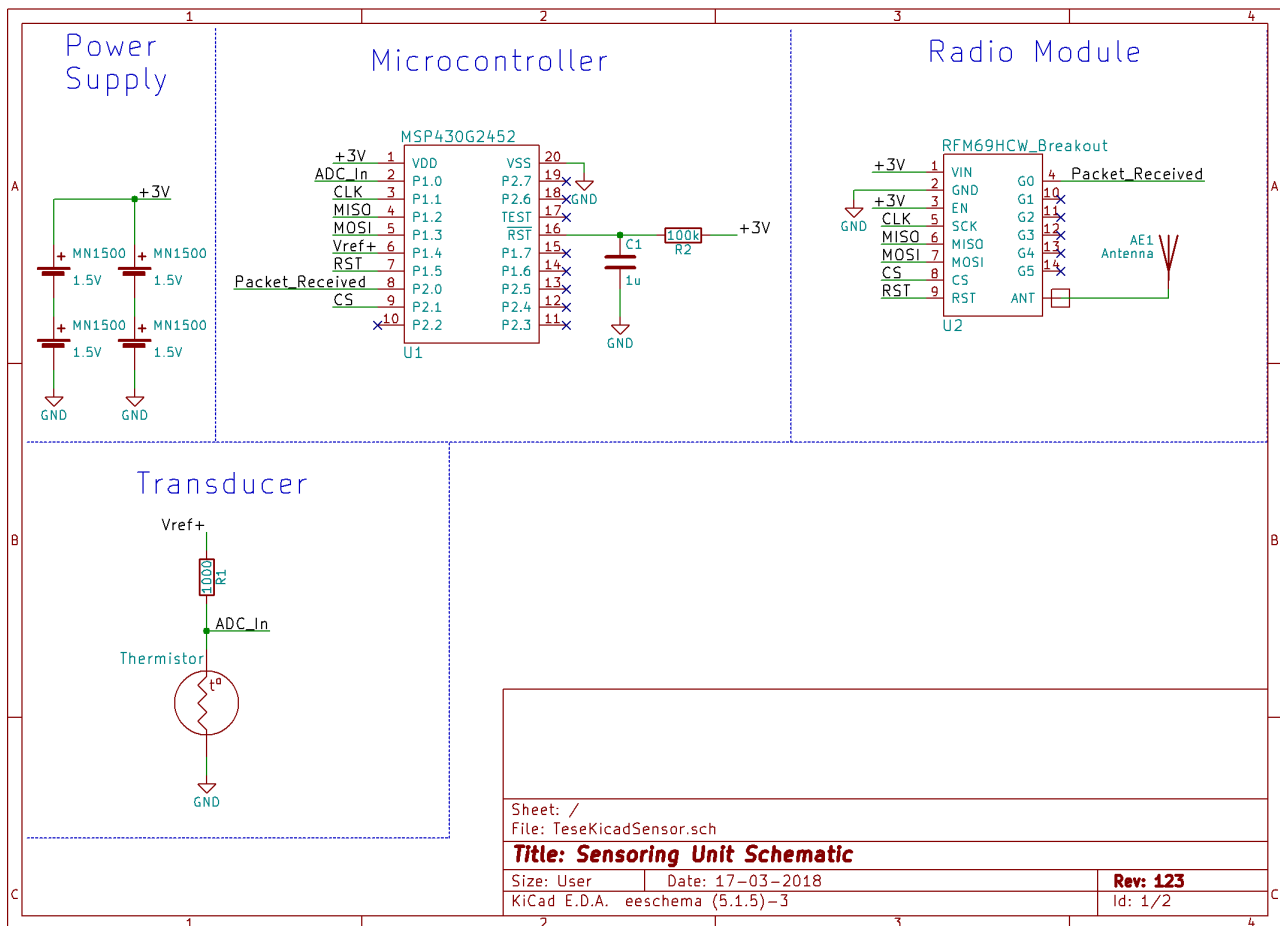


Figura 18: Esquema elétrico da Unidade de Sensor

O microcontrolador selecionado possui uma USART que permite alternar o modo de funcionamento entre UART e SPI (Serial Peripheral Interface). Dado que o "Módulo Rádio" possui uma interface e protocolo SPI, o microcontrolador teria de ser programado para funcionar neste modo. Realizaram-se testes para aferir o correto funcionamento desta comunicação. Estes levaram à conclusão de que o modo SPI do equipamento não estava a ter a resposta prevista. Fornecendo toda a configuração necessária para se estabelecer comunicação, ao efetuar uma leitura num dos registos de teste do "RFM69HCW", o mesmo não retornava qualquer valor. O modo de funcionamento SPI foi exaustivamente testado e concluiu-se que existia algum problema com o funcionamento do mesmo. Posto isto, foi desenvolvido o código necessário para estabelecer uma comunicação SPI através dos pinos I/O do microcontrolador, ignorando na totalidade a *hardware* dedicado a este efeito. O protocolo SPI não possui grande complexidade e é facilmente implementado por código. No que toca à sua interface requer apenas um pino para fornecer o *clock*, outros dois para entrada e saída de dados e um último para realizar o

chip select.

Na Figura 19 tem-se o algoritmo desenvolvido e implementando no microcontrolador da Unidade de Sensor. Este encontra-se sucinto e visa oferecer o fluxo de trabalho que é dado ao MCU quando em funcionamento. O ponto importante de referir refere-se ao primeiro ponto de decisão onde é aferido se se trata do primeiro *boot* após um *factory reset*. Cada Unidade de Sensor tem um ID associado e quem gere os ID é a Hub. No primeiro arranque a Unidade de Sensor não tem em memória nenhum ID, como tal, é feito um pedido de ID à Hub. Seguidamente é feito um pedido da periodicidade das leituras de temperatura. Através da Interface de Utilizador, é possível alterar o período com que ocorrem leituras de temperatura. De forma a simplificar o sistema, a solução que se arranjou passa por requisitar o período à Hub sempre que este arranca, desta forma, se forem feitas alterações à periodicidade a maneira de fazer chegar essa alteração à Unidade de Sensor é reiniciando o mesmo. Esta implementação implica presença física no local por parte do utilizador. Uma alternativa a esta implementação seria requisitar a periodicidade à Hub sempre que fosse enviado uma temperatura à Hub. Não se optou por esta implementação pela razão de que no caso de uma periodicidade de 5 horas, por exemplo, o tempo de espera pela alteração poderia ser demasiado longo para o utilizador.

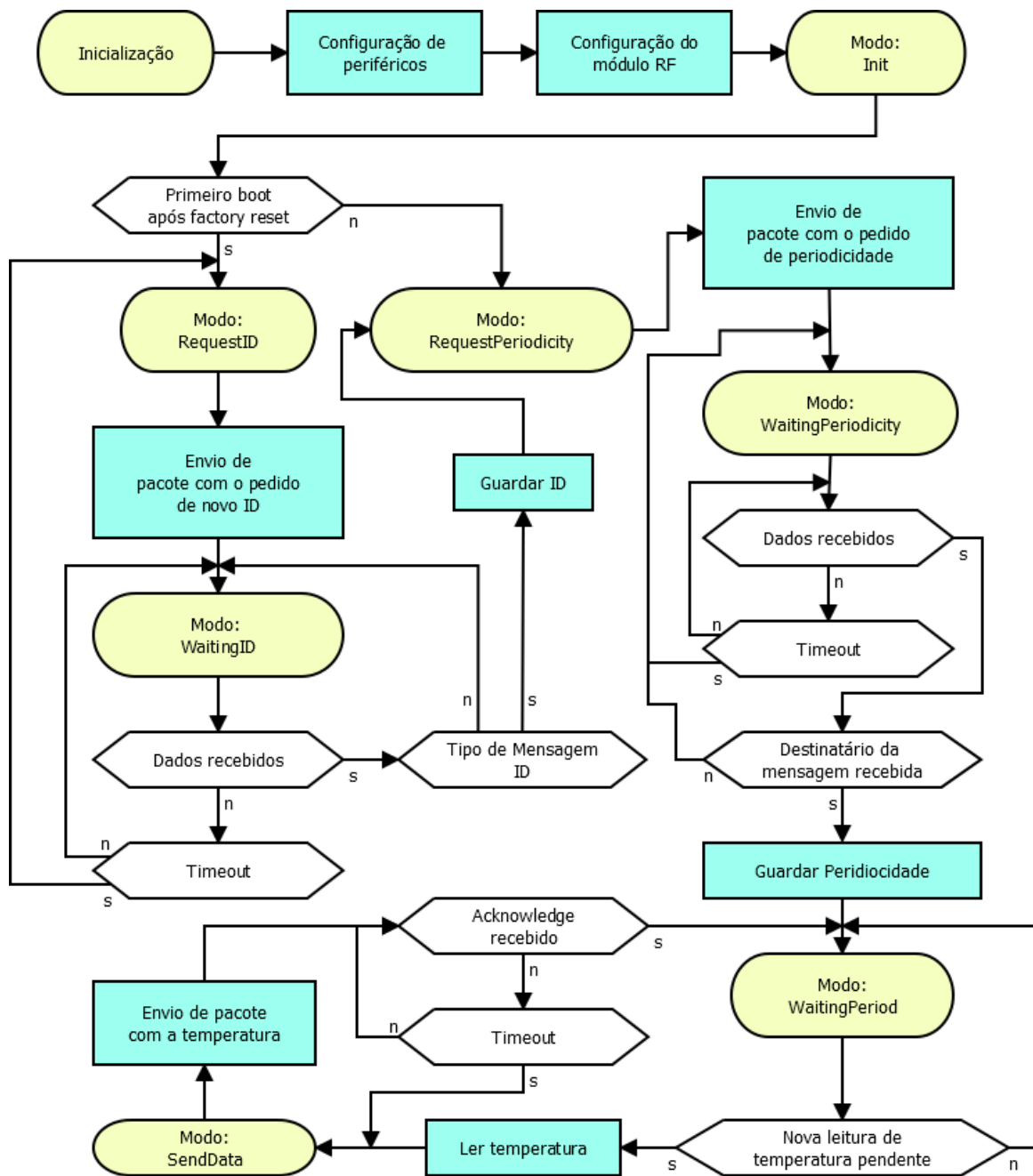


Figura 19: Algoritmo implementado no microcontrolador na Unidade de Sensor

4.1.1 Considerações

Olhando para os requisitos da Unidade de Sensor e para o hardware escolhido para a prova de conceito, é de notar que se teve de sacrificar determinados requisitos, funcionais e não-funcionais, de modo a testar a comunicação *wireless* nas temperaturas alvo. Começando pela temperatura, esta teve que ser limitada devido às baterias utilizadas. Pelo *datasheet* do fabricante, é referido que a gama de temperatura de operação é entre os -20 e os 54 °C, levando a que o teste tenha de ser limitado aos -20 °C.

É visto pelo esquemático do *breakout* do "Módulo Rádio" que tem um regulador de tensão, "AP211". Quando se testou o *breakout*, identificou-se que para a tensão de alimentação estipulada, 3 V, o regulador fornece 1,8 V ao "RFM69HCW", levando a que o ganho de transmissão máximo seja de +17 dBm. Com isto o que se pretende explicar é que para uma tensão de alimentação de 3 V, não há a possibilidade de utilizar o ganho máximo de transmissão permitido pelo dispositivo, +20 dBm. Para que o mesmo fosse possível seria necessário fornecer, pelo menos, 3,5 V ao *breakout*.

O facto de a temperatura ter um impacto negativo na performance da bateria, segundo o "Alkaline Handbook" da Energizer^[50], é de esperar que, em baterias alcalinas, a diferença de tensão aos terminais da bateria, decaia cerca de quatro vezes mais rápido para as temperaturas alvo. A introdução de um segundo conjunto de baterias em paralelo leva a que o consumo energético por série caia para metade, assim consegue-se contrariar de certa forma o problema, uma vez que a tensão no conjunto de baterias irá decair apenas duas vezes mais rápido.

Existem pontos importantes a referir de modo a diminuir o tempo de prototipagem para projetos futuros com o "RFM69HC". A maneira como se transmite informação neste módulo é alcançada de diversas formas. Quando se escrevem dados para o rádio, via SPI, estes podem ser destinados para registos de configuração ou para a FIFO. A FIFO possuiu 64 *bytes* e as escritas são feitas ao *byte*. O método utilizado para que o módulo inicie a transmissão de dados é ao definir quantos *bytes* têm de ser escritos na FIFO, de modo a que esta inicie a transmissão, ou seja, existe um *threshold* e só quando se ultrapassa esse *threshold* é que se dá início ao processo que leva à transmissão. O problema recai no facto de que é necessário fazer três escritas *dummy* adicionais para que o "Módulo Rádio" ative a sua *flag* de *threshold* da FIFO (*First In First Out*). Não se arranjou explicação para este fenómeno uma vez que não estava documentado no *datasheet* do "RFM69HCW". De notar ainda que essas três escritas *dummy* não são transmitidas. Um outro ponto importante é no que toca à receção de dados. É explicado que existe um mecanismo que acorda periodicamente o módulo e este fica "à escuta" de dados - denominado de modo "Listen". Na eventualidade de detetar que existem dados no meio estes são desmodulados, caso contrário o dispositivo entra, uma vez mais, em modo de poupança de energia. De modo a sair do modo de "Listen" é imperativo cumprir com uma determinada sucessão de escritas nos registos de configuração, contudo o mesmo não se verificou. Fazendo o procedimento referido, o dispositivo saía de modo "Listen" e entrava no seu modo de poupança de energia mais eficiente, "Sleep". Estando então no modo "Sleep", não eram detetados dados no meio e nunca se alcançava a comunicação. A alternativa que se arranjou para a receção de dados é no modo manual, tendo o microcontrolador que colocar o módulo em "Receive" periodicamente. Não se encontraram quaisquer explicações para estes fenómenos, mas foram consistentes em ambos os módulos de RF que se adquiriram.

4.2 Hub

A presente secção compreende o maior esforço no desenvolvimento do projeto. Visando uma Hub com uma implementação modular, a secção encontra-se seccionada nas partes que constituem tais módulos.

Não deixando de realizar a justificação do tronco comum a todas as configurações de Hub, o ponto de partida recaí no *hardware* selecionado para a comunicação RF e processamento. À luz do que se fez na secção anterior, segue-se uma listagem do *hardware* selecionado:

- Microcontrolador: "MSP430G2553" da Texas Instruments Inc.
- Módulo Rádio: "RFM69HCW" fabricado pela "HopeRF"[®] com um "breakout" da Adafruit Industries.
- Módulo Wi-Fi: NodeMCU v1.0 (SoC ESP8266)
- Módulo Celular: Fona 3G/GSM+GPS da Adafruit Industries
- Módulo *Webserver*: Raspberry Pi 1 B+

Nesta fase inicial, iremos descrever o algoritmo base por trás da Hub. Na Figura 20 tem-se o diagrama de blocos do algoritmo implementado. Dá-se uma fase inicial de configuração interna do MCU, seguida da configuração do "Módulo Rádio", à semelhança do que se fez na Unidade de Sensor. Após esta fase, é feita uma deteção do *hardware* que se encontra disponível uma vez que isto determina o comportamento futuro no código. Olhando para a figura, em cada um dos módulos dentro da "Deteção de hardware", levam a um processo de inicialização e configuração dos mesmo. Uma vez que este processo é transparente à funcionalidade do algoritmo, omitiu-se nesta fase. O "loop" principal inicia-se e termina no modo "WaitingData", onde o MCU se encontra à espera de dados por parte do "Módulo Rádio" ou "Módulo Wi-Fi". Os dados podem ser pedidos de informação ou envios de informação:

- Os pedidos de informação resumem-se a pedidos de "Periodicidade" ou de "ID". O MCU é quem gere os ID e os respetivos períodos associados a esses ID's. Quando é chegado o pedido de um novo ID, é registado uma periodicidade *default* que será posteriormente pedida pela Unidade de Sensor, como visto anteriormente no seu algoritmo. Se o módulo que fornece a característica de *webservice* se encontra disponível, será enviado para o mesmo o novo ID com o intuito de se guardar essa informação numa base de dados. De forma a garantir que a Unidade de Sensor recebeu a informação, a Hub fica à espera de um novo pedido de envio, derivado de uma falha na comunicação, por um curto espaço de tempo. Por outras palavras, estipulou-se um período de tempo mínimo para o qual a Unidade de Sensor já espera ter recebido as informações provenientes da Hub. Se foi enviado um pedido de informação e esta não chegou num período de doze segundos, é feito um pedido de "Retry" para reforçar o pedido original. Quando a Hub envia a resposta do pedido de informação, fica à escuta de uma mensagem de "Retry". Esta mensagem de "Retry" poderá ser fortuita de uma perda de pacote ou uma falha no *checksum*.
- Os envios de informação estão relacionados com os envios da temperatura registada. Uma vez que a periodicidade de leituras da Unidade de Sensor pode atingir ciclos de 24 horas, é de extrema importância garantir que o envio de temperatura é chegado e confirmado. Dada a importância da informação que se está a enviar, o sistema não podia estar sujeito ao caso em que uma temperatura, que originaria um alarme, não chegasse à Hub. Assim sendo, quando é chegada uma nova temperatura, a Hub envia um

acknowledge à Unidade de Sensor. Se o envio original de temperatura falhou, a Unidade de Sensor envia um "Retry" de dados e aguarda novamente pelo seu *acknowledge*. No capítulo de Anexos, tem-se o anexo A.2 com a Figura 63 com o excerto do código implementado do comportamento descrito. Quem recebe informação envia um *acknowledge*, na eventualidade do *acknowledge* não chegar ao destino, aguarda-se por um *timeout* desse *acknowledge*. Durante esse período, quem enviou a mensagem originalmente, poderá reenvia-lá novamente. Um exemplo seria se a Hub recebeu de facto a temperatura, mas a transmissão do *acknowledge* para a Unidade de Sensor falhou, seria reenviado uma informação que já havia sido registada. Na eventualidade de o mesmo se suceder, esta redundância não prejudicará a funcionalidade do sistema. Enviado o *acknowledge* e já passado o tempo mínimo para se considerar o reenvio, passa-se para o modo de processamento de dados, "ProcessData". Neste é feita uma comparação da temperatura recebida com a temperatura máxima permitida para essa Unidade de Sensor. Se o que se recebeu foi maior ou igual à temperatura máxima, são lidos os dados de configuração da Unidade de Sensor que originou o alarme. Os dados de configuração encontram-se na memória não volátil do MCU e são acedidos com base no ID da Unidade de Sensor. É enviado para o "Módulo Celular", se existir no sistema, os dados necessários para o envio do SMS, ou seja, o número de telefone associado àquele ID, a temperatura máxima permitida e a temperatura corrente. Seguidamente é enviado para o "Módulo Webserver", se existir no sistema, a temperatura recebida. Neste envio o MCU fica à espera por um *acknowledge* à semelhança do que se fez anteriormente.

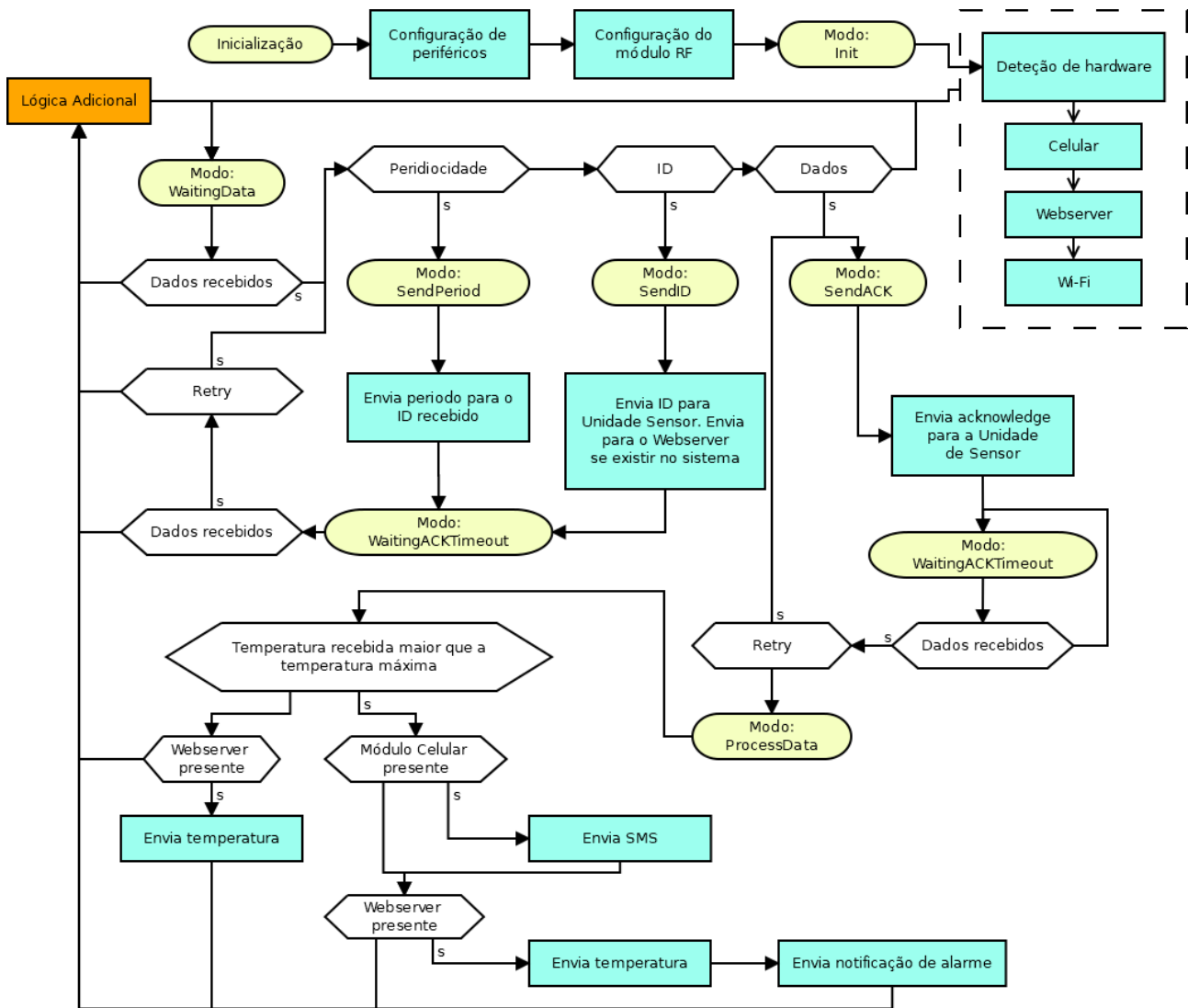


Figura 20: Algoritmo implementado no microcontrolador na Hub

Envio de SMS

Quando existe o objetivo de enviar um SMS ao utilizador derivado de temperaturas excessivas é enviado um SMS com essa informação. Se pegarmos num exemplo de haver um sensor com ID "2", uma temperatura máxima de "-7" e uma temperatura atual de "-5", o SMS recebido pelo utilizador terá a seguinte estrutura:

Temperature alarm!
 Sensor ID: 2
 Max temperature: -7 C
 Current temperature: -5 C
 Reply "Stop alarm sensor 2" to disable this alarm.

Pegando na parte da mensagem que diz "Reply "Stop alarm sensor 2" to disable this alarm.", abaixo será explicado como se faz o controlo desta desabilitação do alarme.

Lógica adicional

Na análise efetuada ao algoritmo implementado, não se mencionou ainda o bloco identificado pela cor laranja que se vê na Figura 20. Este refere uma "Lógica Adicional", lógica essa que tem de ser processada paralelamente ao ciclo principal do algoritmo.

Neste bloco de código são feitas validações no que toca a processamentos assíncronos. É verificado se o utilizador final do sistema enviou algum SMS de inibição do alarme de temperatura e é verificado se a UART tem no *buffer* de entrada uma das seguintes *strings* provenientes do "Módulo Webservice" ou "Módulo Wi-Fi:

- WBCONTACT (Webservice): atualiza o contacto telefónico de um determinado sensor. Estrutura: WBCONTACT@SensorID@AlarmTemperature@UserContact
- WBPERIOD (Webservice): atualiza a periodicidade das leituras de um determinado sensor. Estrutura: WBPERIOD@SensorID@Periodicity
- WIFITMP (Wi-Fi): temperatura proveniente de outro sensor. Estrutura:WIFITMP@Temperature@SensorID

Se o utilizador pretender atualizar o contacto telefónico e/ou a temperatura de alarme, será enviada toda a informação, isto é, se na interface apenas alterar o contacto telefónico será também enviada a temperatura de alarme que se encontra na base de dados. O MCU reconfigura a sua memória interna de configurações de sensores independentemente da intenção do tipo de atualização do utilizador. Exemplificando que é recebido uma ação de atualização do contacto telefónico para o sensor "02" e a temperatura de alarme associado a esse sensor é "-12", a mensagem recebida seria "WBCONTACT@02@-12@9XXXXXXXXX". Uma funcionalidade deste sistema é garantir o *feedback* de que o utilizador recebeu a informação de que dado sensor encontra-se em situação de alarme.

Se a mensagem recebida inicia em "WIFITMP", então sabe-se *à priori* de que se trata de dados de temperatura proveniente de um outro sensor no sistema. Como tal, será iniciado o habitual processamento de dados. Referenciando a Figura 20, o processo avançava para o modo "ProcessData".

Quando é recebido uma temperatura maior ou igual à temperatura de alarme, é ativada uma *flag* que indica que determinado sensor encontra-se em alarme. Neste bloco de controlos assíncronos, é validado periodicamente se foi recebido algum SMS que contenha a informação "Stop alarm sensor X", em que "X" seria o ID do sensor. Se foi recebida essa ação, será desabilitada a *flag* que permite o envio de SMS para dado sensor. Caso contrário será reenviado um SMS com as mesmas informações.

4.2.1 Repeater

Na explicação da Figura 20 na secção anterior, foi referido que se havia omitido a fase de "Detecção de Hardware" vista na referida figura. A lógica por detrás da determinação da Hub *Repeater* é simples, na presença do "Módulo *Wi-Fi*" e se o "Módulo Celular" não tiver sido encontrado, então esta Hub só poderá ser uma Hub *Repeater*. É de notar que a existência do "Módulo *Wi-Fi*" apenas confere a possibilidade de comunicação via protocolo 802.11n, havendo então duas formas de comportamento que dependem da existência do "Módulo Celular" estar presente ou não. Se estiver presente então a Hub comportar-se-á como uma Hub Celular. Se não estiver presente então será uma pura Hub *Repeater* que funcionará como um transmissor dos dados para a Hub primária do sistema. Após esta determinação comportamental, será enviado, via UART, a informação para o "Módulo *Wi-Fi*" de que se trata de uma Hub com a funcionalidade de *Repeater* ou não.

Basicamente a Hub *Repeater* quando entra no seu loop principal, já sabe qual o hardware que se encontra na mesma. As funcionalidades únicas para a "Hub Celular" e "Hub *Repeater*" encontram-se isoladas no código. Uma vez que para se ativar a *flag* que indica que um determinado *hardware* encontra-se disponível, é necessário realizar um *handshake* para esse específico *hardware*. Encontra-se também isolada na perspetiva da arquitetura do sistema porque a linha de dados do "Módulo Celular" difere da linha de dados do "Módulo *Websserver*" e "Módulo Wi-Fi". Estes últimos partilham a mesma linha de dados, mas são mutuamente exclusivos, quando um se encontra presente o outro não poderá fazer parte do sistema. Tudo isto leva a que a implementação da "Hub *Repeater*" seja fácil de implementar do ponto de vista do código da Hub. Todo o processo decorre normalmente até ao ponto de processamento de dados (Ponto referente a "Modo: *ProcessData*" na Figura 20) em que neste, é enviado a temperatura para a UART e, o "Módulo Wi-Fi" estará pronto a receber dados. Esse "Módulo Wi-Fi", para o qual se está a enviar dados, está associado à Hub principal. O algoritmo implementado nesse módulo leva a que aquando da receção de dados de temperatura via HTTP, estes são reencaminhados para a UART do módulo. No lado do microcontrolador da Hub principal, quando for recebida uma mensagem, por meio da UART, com um determinado formato, é interpretado que foi recebida temperatura e que é suposto processar a mesma.

Caracterização funcional

O NodeMCU necessita de saber em que contexto se encontra. Para isto, configura a sua UART para que consiga estabelecer uma via de comunicação com o microcontrolador. Quando o microcontrolador passar pelo seu troço de código que valida o *hardware* presente no sistema, enviará para o NodeMCU o contexto do mesmo, ou seja, se é um *repeater* ou não. Parte da sequência de arranque do NodeMCU consiste em procurar na sua memória interna por um ficheiro de configuração. Ficheiro esse que têm a informação do *access point* que é suposto ligar. Caso não haja nenhuma configuração, o NodeMCU entra em modo de *Access Point*, fornecendo uma rede Wi-Fi com o IP estático "192.168.111.1". Um computador ou um telemóvel com capacidade para ligar a redes Wi-Fi é requerido para uma primeira configuração. Um *workflow* exemplo seria estabelecer uma conexão com NodeMCU através da rede Wi-Fi com um *smartphone* e no seu *browser* digitar na barra de endereços o link "http://192.168.111.1" e aceder ao mesmo. Nesse momento será feito um pedido HTTP, via porto 80 (*default* em ligações HTTP), o NodeMCU terá um *socket* aberto para esse porto e quando lhe é chegado o pedido, o mesmo responde com um HTML que o *browser* interpretará. Na Figura 21 tem-se a página *web* apresentada ao utilizador.

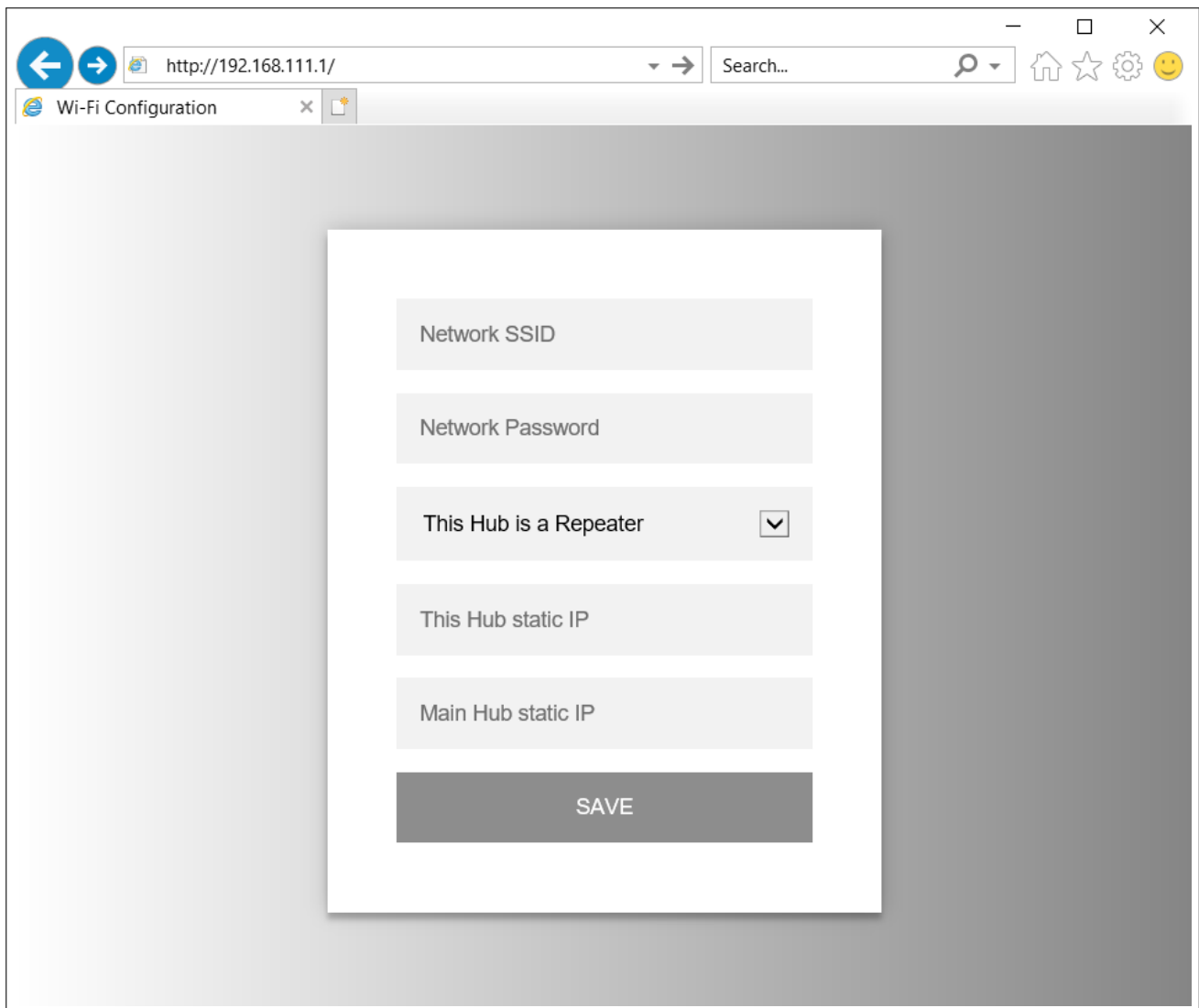


Figura 21: Página de configuração do NodeMCU.

Aqui, é da responsabilidade do utilizador, introduzir a nome e a *password* da rede Wi-Fi que pretende que o NodeMCU se conecte. Por defeito, é apresentado que o NodeMCU comportar-se-á como um "repeater" mas se não for o caso, ao clicar na seta ao lado do texto "This Hub is a Repeater", aparecerá a opção "This Hub is Cellular/Wi-Fi". Independentemente do caso, é necessário que o utilizador identifique o IP do outro NodeMCU que existe no sistema. Este preenchimento é imperativo para que aquando da receção de uma temperatura, se saiba para que IP é suposto estabelecer comunicação. No acto de submissão, o *browser* comunica com o NodeMCU uma vez mais, enviando-lhe os dados fornecidos pelo utilizador que serão, em última instância, guardados no ficheiro de configuração. A partir deste ponto, o NodeMCU sai do modo *Access Point* e inicia uma tentativa de ligação à rede Wi-Fi que existe agora num ficheiro de configuração na sua memória interna. No anexo A.6 tem-se Figura 67 com o código implementado neste modo de "Access Point". Se ao fim de um minuto de tentativas falhadas ainda não tiver sido estabelecido comunicação, é apagado o ficheiro de configuração e reinicia-se o processo. Desta forma, se for introduzido algo de forma equívoca será possível reverter a situação. Estabelecida a conexão à rede pretendida, o microcontrolador já teve tempo para detetar o *hardware* presente no sistema e, durante este processo de configuração, já foi dado o contexto ao NodeMCU. Posto isto, quando o NodeMCU termina de configurar, já estará pronto a ser utilizado.

Partindo do caso de um *repeater*, o microcontrolador envia para a UART os dados referentes à temperatura e ao ID do sensor, o NodeMCU envia esses dados para o IP estático especificado pelo utilizador aquando da configuração da rede Wi-Fi. Existirá no sistema um outro NodeMCU que vai receber esses dados e enviar para o seu microcontrolador, via UART, uma mensagem do tipo: WIFITMP@-12@01. Na Figura 22 tem-se o esquema de um exemplo da comunicação entre duas Hub's e os seus NodeMCU's.

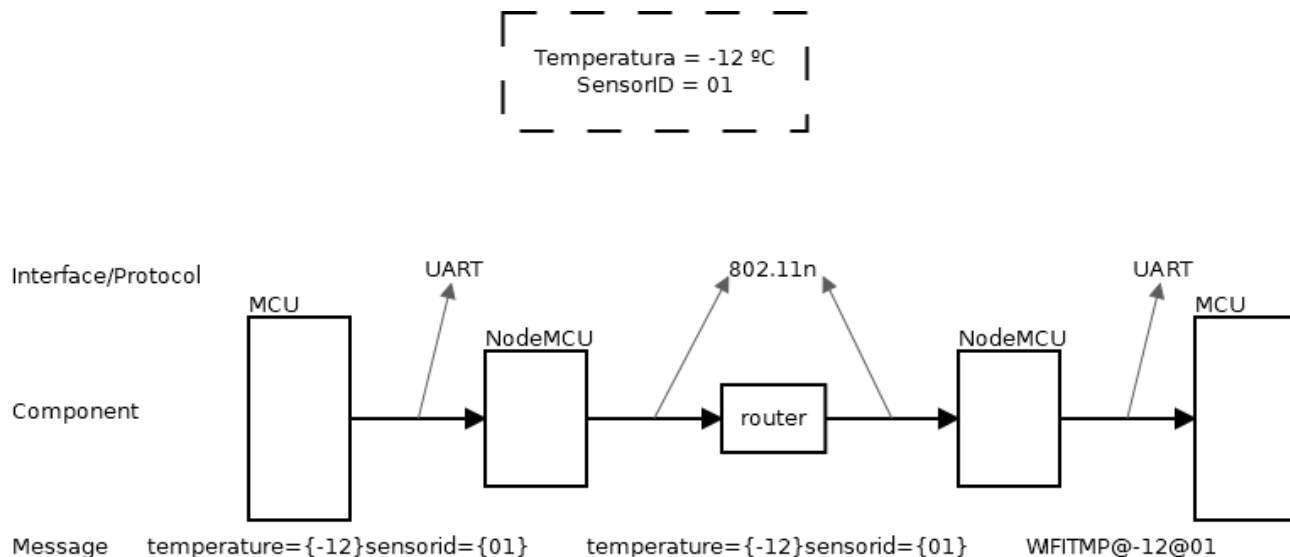


Figura 22: Exemplo de comunicação entre duas Hub's através dos NodeMCU

No capítulo de Anexos encontra-se o anexo A.7 com as Figuras 68 e 69 que dizem respeito às conexões estabelecidas entre a Hub Repeater e a Hub principal presente no sistema, por meio dos seus "Módulos Wi-Fi".

4.2.2 Celular

Já mencionado na secção anterior, a fase de deteção de *hardware* indicará o tipo de Hub. Para uma Hub se tornar do tipo "Celular", é necessário que seja apenas detetado o "Módulo Celular", podendo ser também detetado o "Módulo Wi-Fi" posteriormente. No processo de inicialização, são enviados comandos "AT" via UART e é de esperar que a resposta seja "OK". Nas circunstâncias descritas, a Hub enviará SMS's em situações de alarme e, havendo apenas este módulo na Hub, receberá SMS's com novas configurações de sistema.

Hardware necessário

O "Módulo Celular" é composto pelo *breakout* e outros componentes necessários e indispensáveis para o correto funcionamento. A Adafruit Industries recomenda a utilização de uma antena, que opera 433 MHz, e uma bateria de 900 mHA para assegurar que os picos de corrente, consumida nos instantes iniciais de comunicação, não leve a um *reboot* do circuito. Toda esta comunicação não seria possível se não houvesse também um cartão SIM instalado com capacidade de envio e receção de SMS's. Esta comunicação não é gratuita e está associada a uma operadora de redes móveis, havendo a necessidade de escolher um plano que se adequa às necessidades do utilizador.

Caracterização funcional

Pegando uma vez mais na Figura 20 e estando presente um "Módulo Wi-Fi", a Hub comportar-se-á como recetora de dados provenientes de outras Hub Repeater que existam em sistema e do próprio "Módulo Rádio" que a integra. Posto isto, existe uma transparência às funcionalidades dadas pela "Módulo Wi-Fi" em relação ao "Módulo Celular", isto é, o momento em que este último módulo tem impacto é independente da forma como chegaram os dados. O envio em si de um SMS necessita apenas do número de telefone e do conteúdo da mensagem. Cada sensor tem a sua parametrização guardada na Hub primária do sistema, Hub essa que será a que contém o "Módulo Celular". Cada sensor é parametrizado com base na temperatura máxima, no número de telefone e no seu ID. Os sensores presentes no sistema, têm os ID's guardados na memória interna do microcontrolador e encontram-se numa outra estrutura de dados que tem esse único objetivo. Com base no ID, endereça-se a posição de memória a que lhe corresponde uma configuração e daí retira-se a temperatura máxima permitida e o número de telefone. A partir deste ponto já se detém toda a informação necessária para o envio de SMS. Na Tabela IX em baixo, tem-se a troca de mensagens que ocorrem entre o MCU e o "Módulo Celular".

Tabela IX: Comunicação entre MCU e "Módulo Celular"

Formato de Mensagem	Exemplo	Descrição
MASENCFG@'SensorID' 'Temperatura' 'Contacto'	MASENCFG@01+25+910000000	Envio das configurações dos sensores (MCU para módulo)
MAUPDPER@'SensorID'@ 'Periodicidade'	MAUPDPER@01@02	Alteração da periodicidade via app (módulo para MCU)
MAUPDSEN@ 'SensorId'@'TemperaturaMax'@ 'Contacto'	MAUPDSEN@01@+25 @910000000	Alteração de temperatura de alarme e contacto telefónico via Interface (módulo para MCU)

4.2.3 Wi-Fi e Wi-Fi & Celular

A Hub Wi-Fi é caracterizada pela presença do "Módulo *Webserver*". O acréscimo desde módulo abre toda uma janela de oportunidades em termos de visualização de dados e configuração via interface que a Hub Celular não pode oferecer. Esta Hub comunica com o módulo via UART e para este são enviados os dados após a receção de temperatura por parte do sensor.

Caracterização funcional

Com referido, a comunicação é por meio da comunicação UART com um simples *handshake* para confirmar a receção dos dados no lado do "Módulo *Webserver*". Foi desenvolvido um *software*, na linguagem de programação "Python", que inicializa a UART do RaspberryPi e monitoriza a receção de dados por parte da mesma. No *buffer* de entrada da UART, procura-se por determinadas mensagens e trata-se da informação de acordo. Estas mensagens separam-se entre dados de presença *hardware*, temperatura/alarmes e configuração de sensores. Segue-se a Tabela X com estruturas das mensagens bem como exemplos de utilização.

Tabela X: Comunicação entre MCU e "Módulo Webserver"

Formato de Mensagem	Exemplo	Descrição
WebServerModuleCheck	WebServerModuleCheck	Determinação da presença do módulo (MCU para módulo)
WebServerDt@'SensorID'@ 'Temperatura'@	WebServerDt@01@+25.3@	Envio de um registo de temperatura (MCU para módulo)
WebServerAl@'SensorID'@ 'Temperatura'@	WebServerAl@01@+25.3@	Envio de um registo de alarme (MCU para módulo)
WebServerSensorConfig@ 'SensorId'@'TemperaturaMax'@ 'Contacto'	WebServerSensorConfig@01@+25 @000000000	Registo de um novo sensor (MCU para módulo)
WBCONTACT@'SensorId'@ 'TemperaturaMax'@'Contacto'	WBCONTACT@01@-15@ 910000000	Alteração de temperatura de alarme e contacto telefónico via Interface (módulo para MCU)
WBPERIOD@'SensorId'@ 'Periodicidade'	WBPERIOD@01'@03	Alteração da periodicidade via interface (módulo para MCU)

Como referido, para interpretar a tabela acima, desenvolveu-se um pequeno *script* em "Python" onde se utiliza as bibliotecas que permitem a utilização da UART e do protocolo HTTP. No capítulo de Anexos, tem-se o anexo A.3 com a Figura 64 com um excerto do *script*. Com um *method* do tipo "GET" do protocolo HTTP, consegue-se definir os parâmetros a interpretar pelo servidor. O cliente e o servidor encontram-se na mesma máquina, isto é, o próprio RaspberryPi, portanto para se endereçar o ficheiro no *webserver* responsável pelo tratamento dos parâmetros, recorre-se ao próprio IP local do *webserver*, o *localhost*. O ficheiro mencionado foi escrito em PHP uma vez que permite capturar os parâmetros no método "GET" e possui ferramentas de comunicação com bases de dados. Segue-se a Figura 23 com um esquema simplificado desta comunicação.

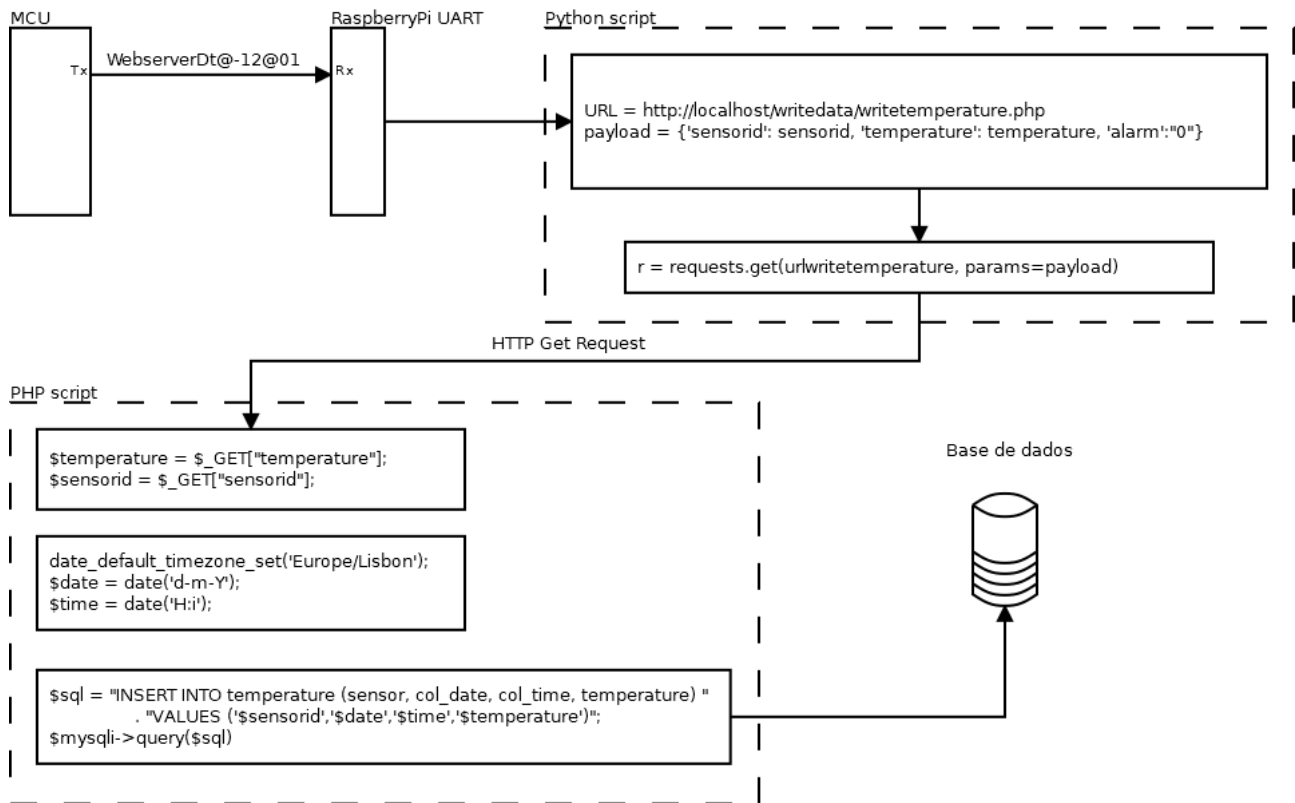


Figura 23: Caminho de dados desde o MCU até à base de dados.

O fluxo de dados apresentando na figura também é válido para alarmes e para o envio de novos sensores para a base de dados. No caso do envio de alarme, por parte do MCU, será feita uma segunda chamada apenas destinada ao envio da mensagem a que corresponde um alarme. Posteriormente, o *script* "Python" enviará a temperatura/ID e uma *flag* que indica a situação de alarme para o *script* PHP. Este último vai inserir uma nova entrada na tabela destinada exclusivamente às situações de alarme.

O envio de novos ID's de sensores necessita apenas dos parâmetros ID, temperatura máxima e contacto telefónico. Ficando desde já a nota que na Hub Wi-Fi ou Wi-Fi & Celular a configuração dos sensores é ao nível da página *web* desenvolvida. Os dados em si enviados no ato de inserção de novos ID's, passam por uma temperatura máxima de "25" graus e um contacto telefónico "000000000". A informação na memória do MCU, necessita de valores *default* e serão esses mesmos valores que se guarda na base de dados.

Invertendo a direção na comunicação, temos o *webserver* a comunicar com o MCU. O foco principal deste acontecimento é quando o utilizador configura um sensor. Por meio de um outro *script* "Phyton", que é invocado por um *script* PHP, será enviada uma das seguintes mensagens, já antes discutida, "WBCONTACT" ou "WBPERIOD".

Falando um pouco sobre o *webserver* a correr no RaspberryPi, este corre aquilo que se denomina na indústria por LAMP (Linux Apache MySQL PHP). Segue-se a Tabela XI com o *software* utilizado.

Tabela XI: *Software utilizado na implementação LAMP do webservice*

<i>Software</i>	Versão
Linux	Raspbian GNU/Linux 9
Apache	2.4.25 (Raspbian)
MySQL	10.1.23-MariaDB-9+deb9u1 Raspbian 9.0
PHP	7.0.30-0+deb9u1

Salienta-se ainda o facto de que foi necessário configurar a *router*, que fornece a rede Wi-Fi no local, de forma a que o *website* seja acessível fora da rede onde se encontra. Foi então definido o IP estático "192.168.2.6" no RaspberryPi e configurada a *router* para que permita chamadas a esse dispositivo no porto 80.

4.3 Interface de Utilizador

A interface de utilizador é um ponto crítico no desenvolvimento do projeto uma vez que é aquele que o utilizador interage. Será nesta secção que o utilizador poderá decidir a configuração de cada sensor e a visualização dos dados recolhidos.

4.3.1 Webpage

Na secção 3.4.2 deste documento apresentou-se uma imagem com o diagrama funcional da Interface de Utilizador numa Hub Wi-Fi. Por conveniência do leitor, a Figura 24 reflete o que já antes foi discutido. Focando nos pontos vistos na imagem tais como "Autenticação" ou "Gerir perfil", iremos abordar as diferentes páginas *web* que os representa. Desde da lógica à funcionalidade da mesma, a presente secção será dividida de acordo com os pontos na Figura 24. O texto presente no *website* encontra-se em inglês com a justificação de este dispositivo poder ser utilizado por utilizadores em que a língua nativa não é o português. Desta forma generaliza-se a utilização do mesmo.

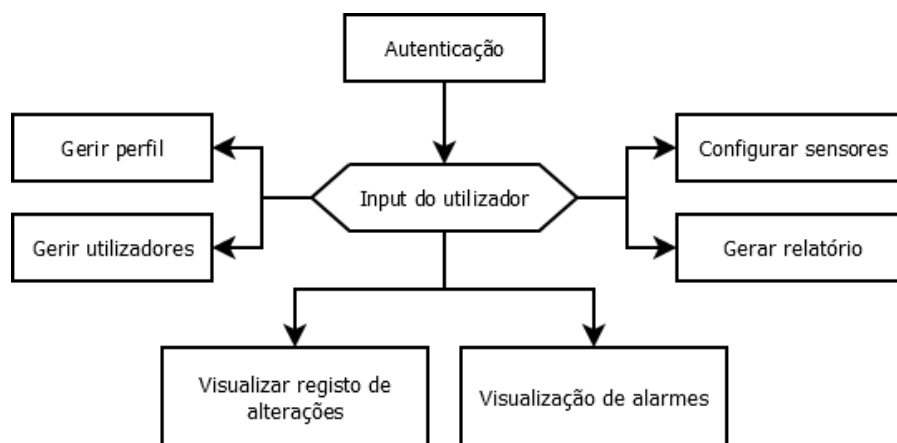
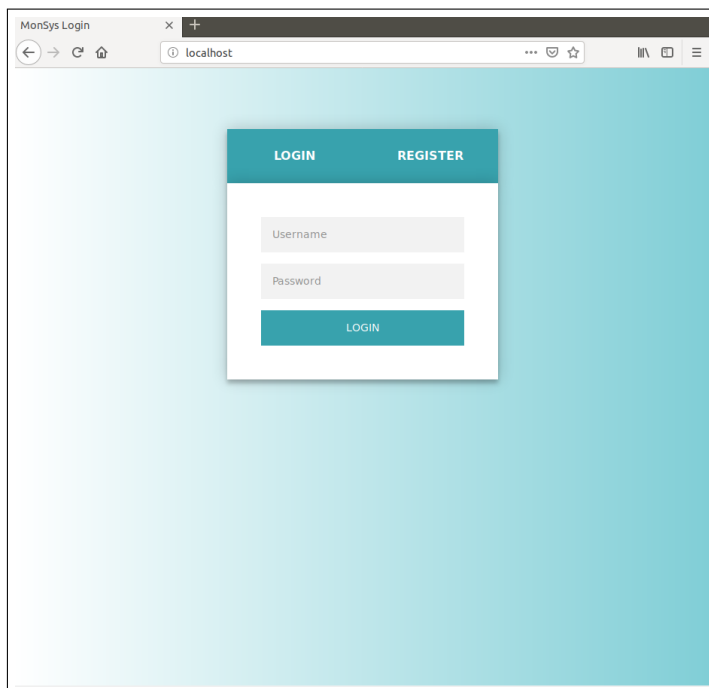


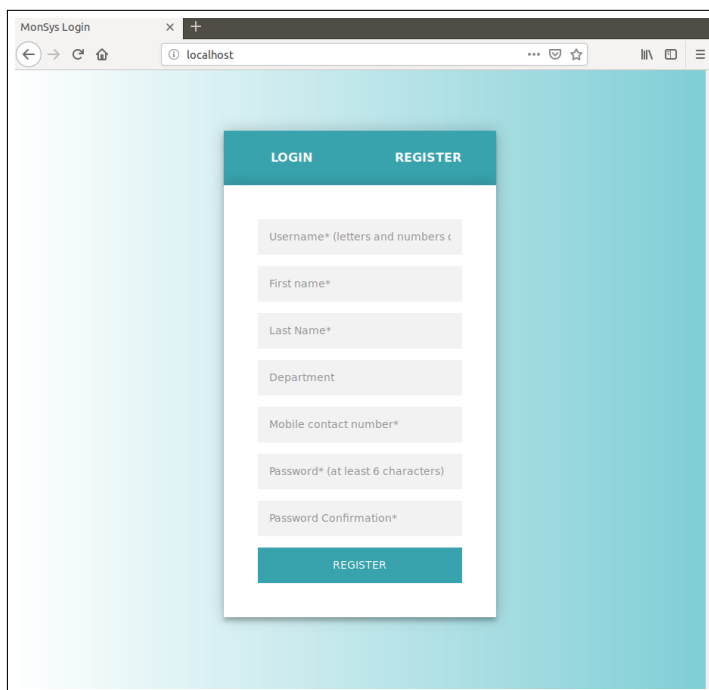
Figura 24: *Diagrama funcional da Interface de Utilizador no modo Hub Wi-Fi.*

Autenticação

Quando se acede ao *website*, é apresentado o formulário de *login* ao utilizador onde se poderá introduzir o "username" e "password". Se o utilizador ainda não tiver efetuado um registo prévio na plataforma, terá de carregar no botão "Register", onde encontrará o formulário de registo.

A screenshot of a web browser window titled "MonSys Login" with the address bar showing "localhost". The page has a light blue background. In the center, there is a white rectangular form with a teal header bar containing the words "LOGIN" and "REGISTER". Below the header, there are two input fields: "Username" and "Password". At the bottom of the form is a teal button labeled "LOGIN".

(a) Autenticação.

A screenshot of a web browser window titled "MonSys Login" with the address bar showing "localhost". The page has a light blue background. In the center, there is a white rectangular form with a teal header bar containing the words "LOGIN" and "REGISTER". Below the header, there are several input fields: "Username* (letters and numbers c", "First name*", "Last Name*", "Department", "Mobile contact number*", "Password* (at least 6 characters)", and "Password Confirmation*". At the bottom of the form is a teal button labeled "REGISTER".

(b) Registo

Figura 25: Interface de Utilizador (Web) - formulário de autenticação e registo.

Abaixo segue a Tabela XII com os dados necessários a introduzir bem como as limitações impostas aos campos.

Tabela XII: Caracterização dos campos associados ao registo de um novo utilizador.

Campo	Descrição	Limitação
<i>Username</i>	Campo obrigatório do nome de utilizador	Apenas letras e números com um mínimo de 4 e um máximo de 30 caracteres.
<i>First name</i>	Campo obrigatório do primeiro nome do utilizador	Apenas letras com um máximo de 30 caracteres
<i>Last name</i>	Campo obrigatório do último nome do utilizador	Apenas letras com um máximo de 30 caracteres
<i>Department</i>	Campo opcional que indica o departamento do utilizador	-
<i>Password</i>	Campo obrigatório correspondente à <i>password</i> de acesso	Apenas letras e números com um mínimo de 6 e um máximo de 30 caracteres.
<i>Password Confirmation</i>	Campo obrigatório com a confirmação da <i>password</i> introduzida	Este campo e o anterior necessitam de ser iguais.

Quando é efetuado a intenção de registo, ou seja, carregar no botão de "Register", as validações dos campos são ao nível dos códigos PHP. Sempre que um dos campos não for devidamente introduzido, irá aparecer uma mensagem de erro que diz o que foi mal introduzido. Se todas as validações forem corretamente verificadas, será então introduzido na base de dados de utilizadores, um novo registo. Se não existir nenhum outro utilizador na base de dados, ou seja, se este for o primeiro registo a ser efetuado no sistema, então será mostrada uma mensagem ao utilizador de que se trata do primeiro registo e que a sua conta terá privilégios de administrador, sendo imediatamente possível efetuar *login*. Todos os outros novos utilizadores terão de ver a sua conta ativada por um administrador. Caso um utilizador efetue autenticação sem ter a conta ativa, irá aparecer uma mensagem que indica que a conta necessita de ativação prévia. Por motivos de segurança, todos os outros novos utilizadores terão que ter a sua conta ativada por um utilizador com privilégios de administrador. É ainda de mencionar que a *password* será encriptada quando guardada na base de dados. A encriptação é ao nível do código PHP e é utilizado o método de encriptação "BCRYPT"^[51], tem-se o anexo A.4 no capítulo de Anexos com a Figura 65 com o excerto do código PHP implementado no ato de registo.

Gerir perfil e utilizadores

Após a autenticação, é apresentado ao *user* por defeito a "tab" de gestão de perfil, Figura 26. Os únicos campos que permitem alterações referem-se ao departamento, contacto telefónico e alteração de *password*. Os campos têm as mesmas restrições impostas que no formulário de registo. A alteração da *password* requer a introdução da *password* atual por razões de segurança.

Profile				Sensor charts	Sensor configuration	Alarms & Reports	Log	Exit
Username	tiagoresende							
First name	Tiago							
Last name	Resende							
Department	IST		Change department			APPLY CHANGES		
Contact	910000000		Change contact			APPLY CHANGES		
User type	admin							
Change password	Current password		New password				APPLY CHANGES	
			Confirm new password					
Check if there are users to be activated				OK				

Figura 26: Interface de Utilizador (Web) - gestão de perfil.

Os utilizadores com privilégios de administrador, têm uma opção adicional nesta "tab". Podem verificar se existem utilizadores pendentes de ativação. Carregando no botão "OK", Figura 26, passa-se para uma janela à parte com uma listagem dos utilizadores presentes no sistema e o seu estado de ativação, dando a possibilidade de ativar ou desativar utilizadores, Figura 27.

User	First name	Last name	Department	User type	Status	
tiagoresende2	Tiago	Resende	IST2	user	Active	DEACTIVATE
tiagoresende3	Tiago	Resende	IST3	user	Not active	ACTIVATE
RETURN						

Figura 27: Interface de Utilizador (Web) - gestão de utilizadores.

A abordagem à implementação deste conjunto de informações passou por colocar o código PHP a realizar *queries* às tabelas "users" da base de dados "accounts", e fazer o *output* dessa informação para uma estrutura de tabela em HTML. Desta forma consegue-se popular a informação toda no *request* HTTP efetuado. A desvantagem deste tipo de implementação passa por não haver uma informação atualizada periodicamente enquanto dentro da página. Sendo sempre necessário refrescar a página de forma a determinar se existe novo conteúdo.

Configurar sensores

Na "tab" denominada "Sensor Configuration", tem-se a opção de verificar ou alterar as configurações de um determinado sensor. Para isso, seleciona-se o sensor pretendido e carrega-se em "OK", Figura 28. Se não existir dados de configuração para o sensor selecionado, aparecerá uma mensagem de erro com essa mesma informação.

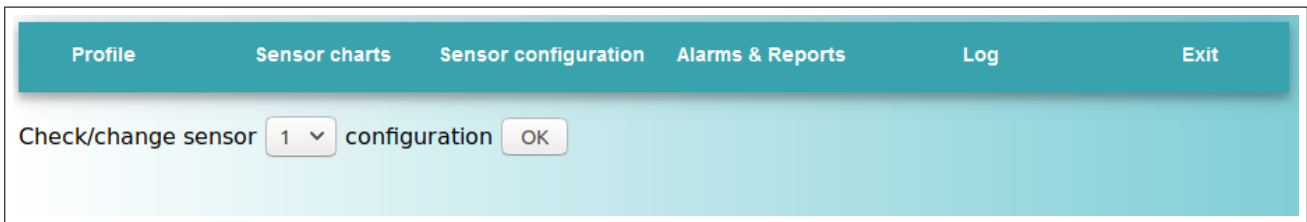


Figura 28: Interface de Utilizador (Web) - gestão de sensores.

Na nova janela, Figura 29 o utilizador visualiza o estado atual de configuração do sensor. Esses dados estão guardados na tabela "sensorconfig" da base de dados "sensors" onde foram indiretamente inseridos pela Hub. Neste momento consegue-se atribuir uma temperatura máxima, um utilizador responsável e a periodicidade entre leituras. Ao definir o utilizador responsável, será enviado para a Hub o número de telefone associado a esse utilizador, estando essa informação na tabela de "users". Ao aplicar as alterações por meio do botão "Apply Changes", será invocado um "script" em Python que irá enviar esta informação, via UART, para a Hub. É feita uma validação, via PHP, de forma a não realizar ações desnecessárias às bases de dados ou à Hub. Esta passa por verificar que na intenção de alteração, existe um utilizador associado àquele sensor que se pretende alterar. Esta validação justifica-se pelo objetivo de associar um sensor a um utilizador, tornando-o responsável pelo mesmo numa situação de alarme.



Figura 29: Interface de Utilizador (Web) - configuração de sensores.

Gerar relatório

Na secção de "Alarms & Reports", tem-se a funcionalidade de gerar um relatório com as temperaturas recolhidas para um dado sensor num mês a escolher pelo utilizador. É ainda possível visualizar os alarmes que foram gerados num sensor, Figura 30.

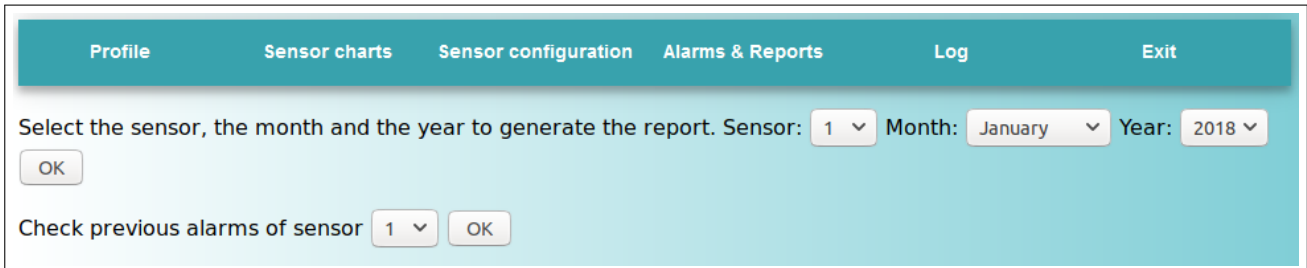


Figura 30: Interface de Utilizador (Web) - relatórios de temperatura.

Visualização de alarmes

Ainda na secção "Alarms & Reports", o utilizador pode visualizar todos os alarmes ocorridos, para um dado sensor, desde do início da monitorização. Assim tem toda a informação necessária para uma rápida visualização de eventos ocorridos, Figura 31.

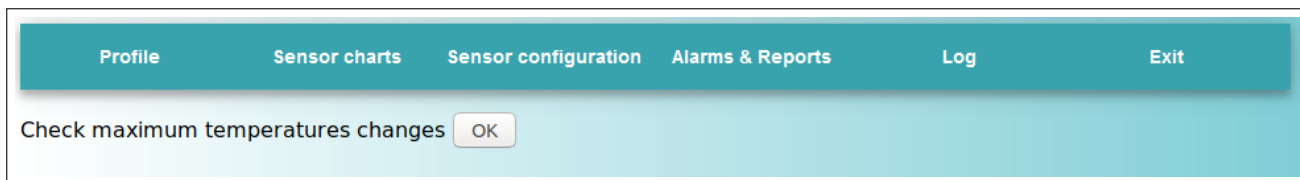


Sensor ID	Date	Time	Recorded temperature
-----------	------	------	----------------------

Figura 31: Interface de Utilizador (Web) - registo de alarmes anteriores.

Visualizar registo de alterações

Como mecanismos de segurança, foi implementada a funcionalidade de registar as alterações às temperaturas máximas permitidas. Sempre que um utilizador alterar uma temperatura de alarme de um sensor, será guardada essa informação em base de dados, Figuras 32a e 32a .



(a) Menu de acesso.

User who made the change	Sensor	Old maximum temperature	New maximum temperature	Date	Time

RETURN

(b) Tabela de registo.

Figura 32: Interface de Utilizador (Web) - Registo de alterações de temperaturas

4.3.2 Aplicação Móvel

Com a Hub a funcionar no modo de Hub Celular, a configuração de sensores e visualização de dados é controlada por meio de uma aplicação móvel. Na especificação de requisitos, secção 3.2 - Requisitos do Sistema, é colocado como prioridade máxima a implementação de uma aplicação móvel que permita definir temperaturas de alarmes, que utilizadores recebem que alarmes e definir a periodicidade entre leituras. Posto isto, utilizou-se o *software* "Android Studio" que oferece um meio de desenvolvimento de aplicações móveis para sistemas Android. Com base na Figura 33, apresentada na secção 3.4.2 - Arquitectura Interface de Utilizador, estipulou-se que cada "caixa" vista na Figura 34, corresponderia a uma janela com informação. Em cada janela tem-se decisões por parte do utilizador que correspondem às ramificações entre "caixas". Desta forma conseguia-se desenvolver páginas estáticas que avançavam para outras páginas consoante o *input* do utilizador, simplificando desta forma a curva de desenvolvimento desta componente gráfica.

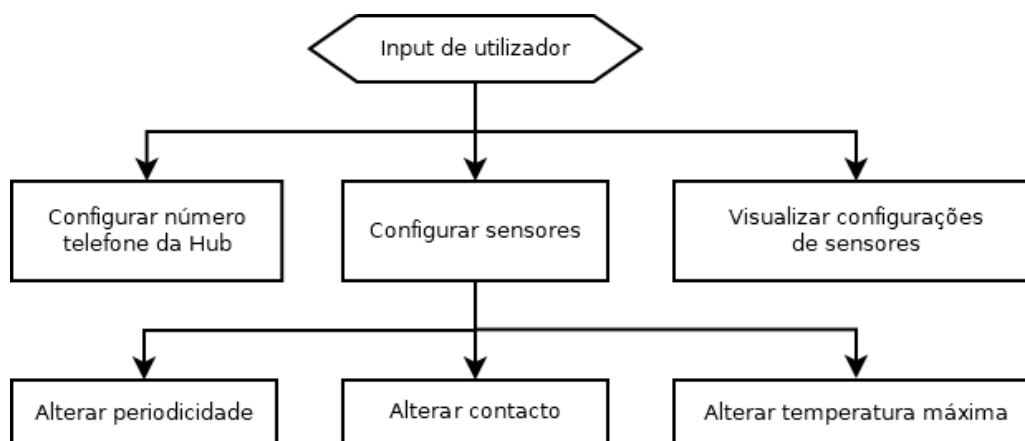


Figura 33: Diagrama funcional da Interface de Utilizador no modo Hub Celular.

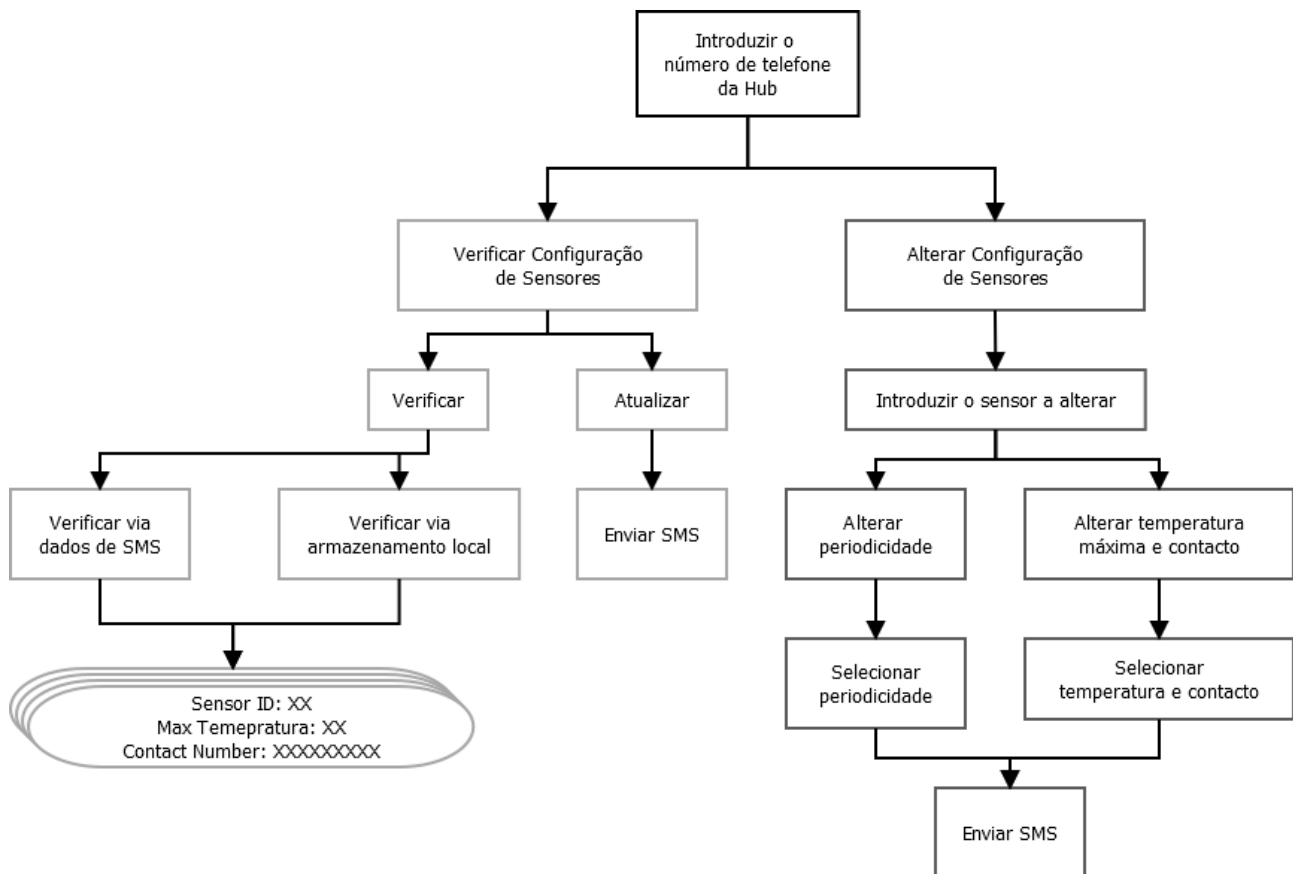
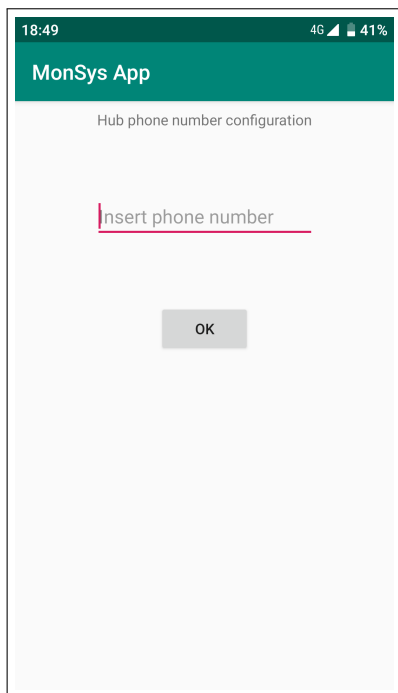
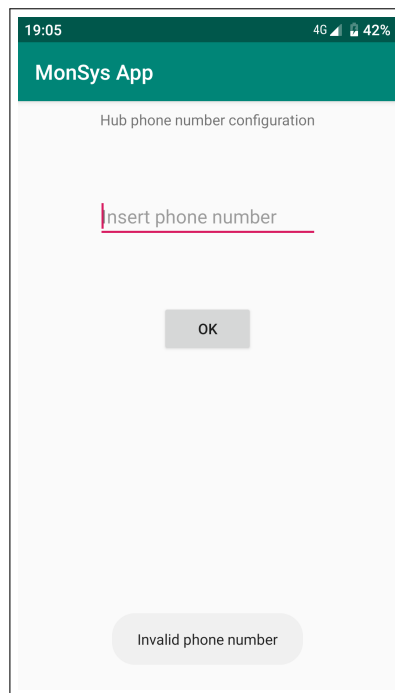


Figura 34: Fluxo de ações na aplicação móvel

Na Figura 35a, o primeiro passo refere-se à introdução do número de telefone da Hub, existindo um "placeholder" com essa mesma informação. É imposto a introdução de apenas caracteres numéricos e que este campo esteja restringido a nove caracteres. Caso contrário, aparecerá uma mensagem que informa que o número introduzido é incorreto, Figura 35b. Validado o número, este é guardado em memória para que na próxima vez que seja aberta a aplicação, não seja necessário introduzir novamente o mesmo.



(a) "Placeholder".



(b) Mensagem de erro.

Figura 35: Interface de Utilizador (App) - introdução do número de telefone da Hub.

Passada a fase de introdução do número, é apresentado o menu que permite ao utilizador escolher se pretende verificar ou alterar dados de configuração, Figura 36.

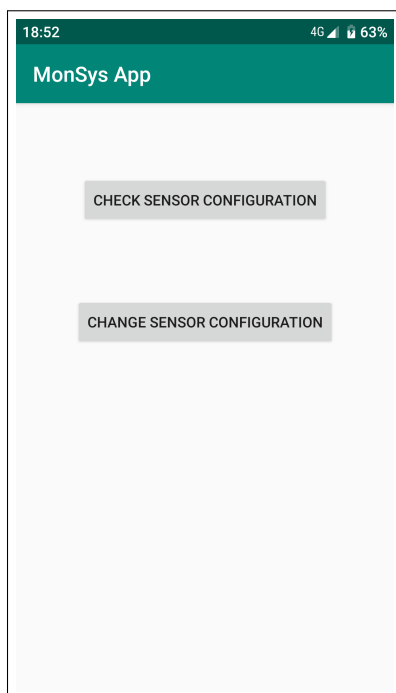
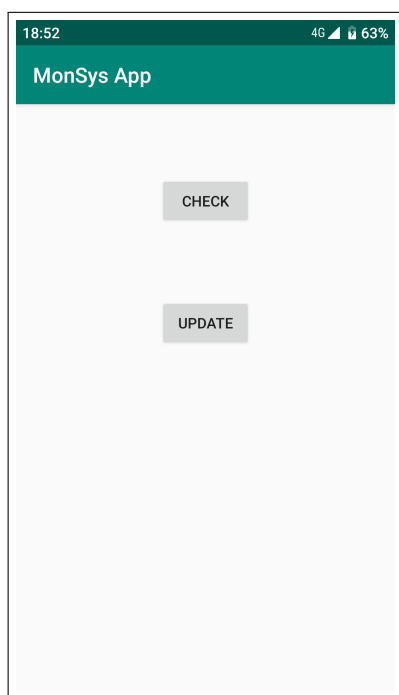
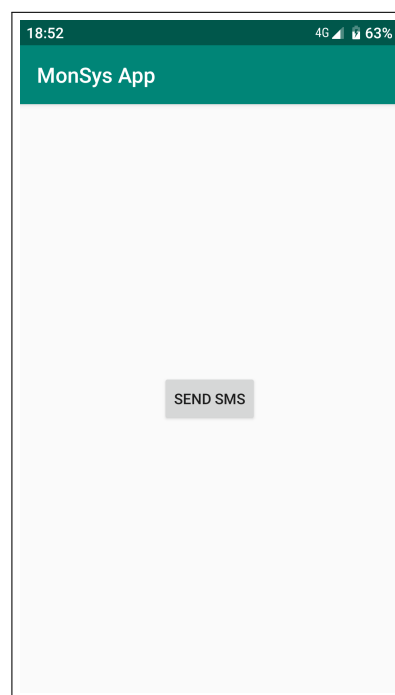


Figura 36: Interface de Utilizador (App) - verificar ou alterar dados de configuração.

Tendo o utilizador a intenção de visualizar as configurações atuais dos sensores, terá ainda de decidir se pretende verificar a informação ou enviar um pedido à Hub, Figura 37a, para que lhe seja chegado dados atualizados ao dispositivo móvel. O pedido à Hub por novos dados é feito, naturalmente, por meio de um SMS que é enviado pela própria aplicação para o número introduzido no passo anterior, Figura 37b. No capítulo de Anexos, tem-se o anexo A.8 com Figura 70 que apresenta o código JAVA implementado neste envio de SMS.



(a) Verificar ou atualizar.



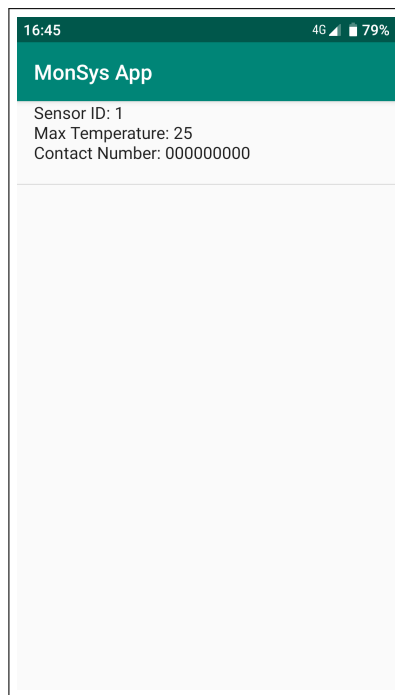
(b) Ponto de envio de SMS para atualizar os dados

Figura 37: Interface de Utilizador (App) - verificar ou atualizar dados de configuração.

Mediante a solicitação de dados atualizados, o utilizador terá de seleccionar a opção de verificar a caixa de entrada de SMS's, Figura 38a. A aplicação irá procurar por SMS's vindos do número de telefone da Hub (configurada previamente), Figura 71 no capítulo de Anexos. Caso não seja encontrado nenhum SMS, aparecerá uma mensagem informativa de erro, à semelhança das anteriores, com essa mesma informação. Escolhida a opção de método de visualização, o utilizador entra para o ecrã final onde encontra uma lista dos sensores em sistema com o seu ID, temperatura máxima e contacto telefónico, Figura 38b.



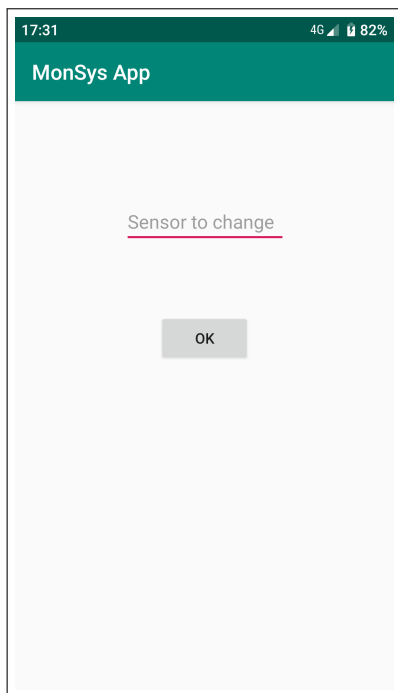
(a) Verificar via caixa de SMS's ou armazenamento local.



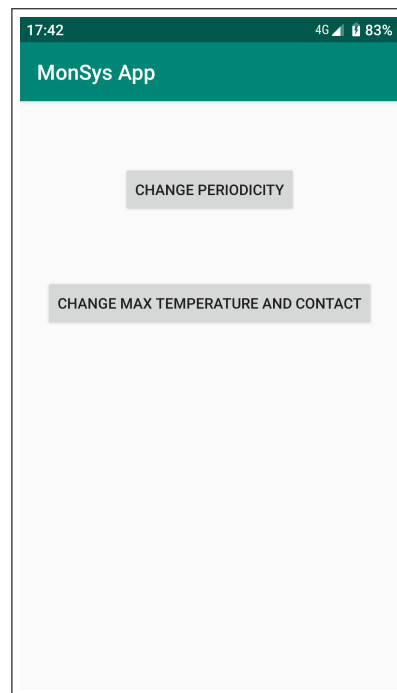
(b) Visualização de configuração.

Figura 38: Interface de Utilizador (App) - visualização de dados de configuração.

No menu principal, Figura 36, mediante a opção de alterar a configuração de um sensor, é apresentado o ecrã da Figura 39a. Neste o utilizador terá de identificar qual o sensor que pretende alterar. O ID introduzido é validado na medida em que terá de existir nos dados internos da aplicação, levando a que numa primeira instância, não seja possível alterar sensores sem antes solicitar dados atualizados à Hub. Passando de seguida para a escolha da alteração pretendida, alteração da periodicidade ou temperatura máxima e contacto telefónico.



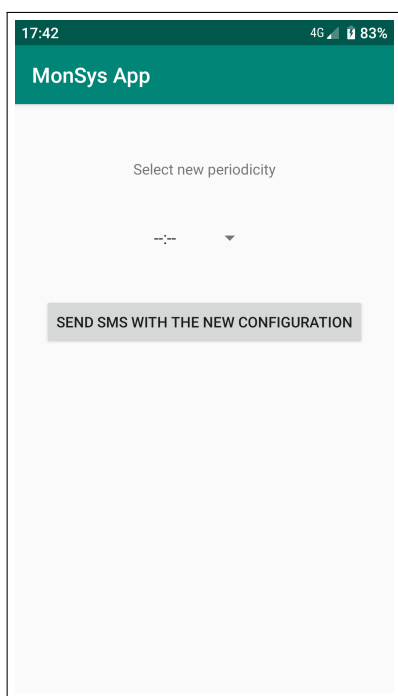
(a) Seleção do sensor a modificar.



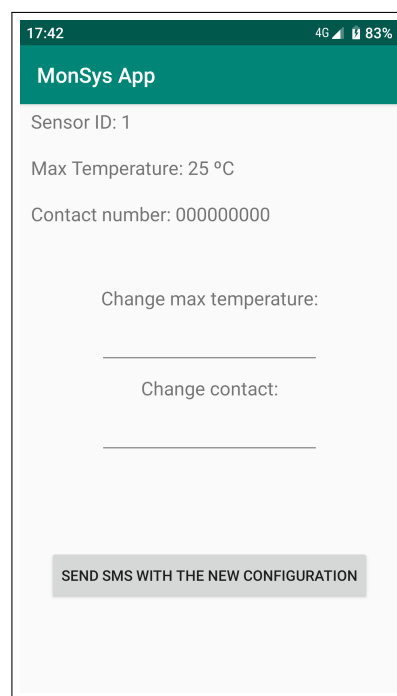
(b) Modificar Periodicidade ou Temperatura e Contacto.

Figura 39: Interface de Utilizador (App) - modificar dados de configuração.

Na alteração da periodicidade, Figura 40a, é apresentado uma lista ao estilo "dropdown list", onde as periodicidades possíveis de seleção encontram-se separadas em períodos de quinze minutos. A opção restante, alterar temperatura e contacto, leva à visualização do estado atual de configuração do sensor selecionado, havendo assim uma maior comodidade na alteração do mesmo, Figura 40b. Em ambos os casos, o passo final será o ato de carregar no botão que envia um SMS à Hub com os novos dados de configuração. No capítulo de Anexos, tem-se o anexo A.8 com a Figura 72 com o código JAVA que envia o SMS de configuração.



(a) *Modificar Periodicidade.*



(b) *Modificar Temperatura e Contacto.*

Figura 40: *Interface de Utilizador (App) - modificar Periodicidade ou Temperatura e Contacto.*

5 Teste do Sistema e Resultados Experimentais

Caracterizado todo o sistema e justificadas as decisões de projeto, chega-se agora ao ponto em que se testa efetivamente o sistema e se retiram ilações ao que foi dimensionado. Começando pela Unidade de Sensor, seguido da Hub e por fim a Interface de Utilizador. Nesta última, os resultados apresentados limitam-se à visualização de dados que foram realizados ao longo dos testes nas fases anteriores. Nos testes realizados à Hub, tem-se configurações de sensores que foram realizadas via a aplicação móvel e a página *web*. Configurações essas que se resumem a mensagens que o MCU da Hub interpreta e decide sobre as mesmas.

Os testes realizados servirão de base que fundamentam a secção que sucede esta, Hardware Final. Com base nas conclusões aqui retiradas, justifica-se as alterações necessárias ao sistema a fim de determinar o *hardware* final do sistema.

5.1 Unidade de Sensor

Esta subsecção encontra-se segmentada em testes de medição e transmissão de temperaturas dentro de equipamentos de refrigeração, passando pela distância de comunicação entre Unidade de Sensor e Hub e os consumos energéticos desta unidade. Em cada um destes pontos, é feita uma introdução às condições de testes e conclusões dos mesmos.

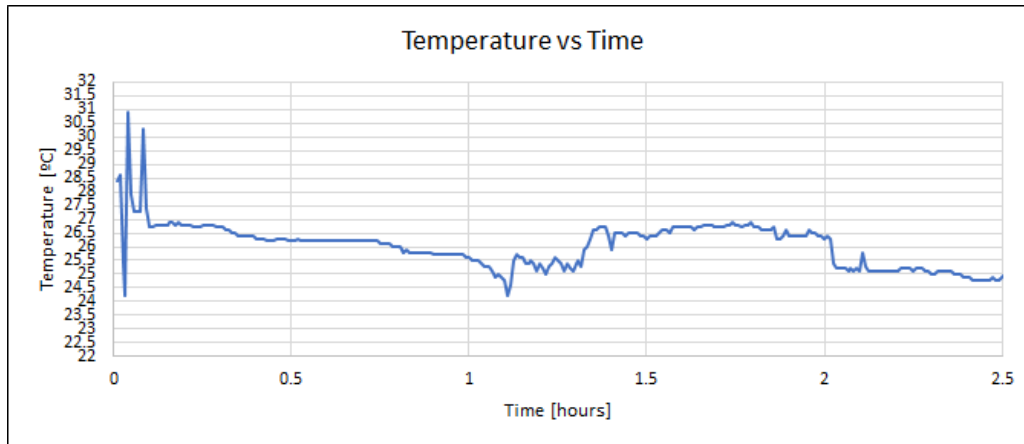
Leituras de Temperatura

Na presente secção tem-se o teste realizado referente a leituras remotas de temperatura numa abordagem convergente com um caso real. Foram realizados três testes em gamas de temperatura distintas e, em cada uma delas, explorou-se as diferenças entre expor o termístor ao ar e embebido em glicol, visando a justificação deste último. Com esta bateria de testes consegue-se colocar o sistema em ambientes agressivos de operação, nomeadamente quando exposto a temperaturas baixas. Estipulou-se que um teste para ser considerado válido, teria de ter uma duração mínima de duas horas ininterruptas, com leituras de temperatura distanciadas em 10 segundos. O sistema é suposto funcionar com periodicidades de 15 minutos no mínimo, contudo isto levaria a que os dados recolhidos ficassem limitados a apenas 8 temperaturas recolhidas. Outro ponto negativo seria uma menor sensibilidade na análise em relação às flutuações entre leituras, isto é, espera-se que o glicol sirva de "buffer" às pequenas variações de temperatura e, para isso, é necessária uma periodicidade entre leituras bastante reduzida de forma a se conseguir observar essas variações. Por outro lado, com o termístor exposto ao ar, as variações deverão de ser mais acentuadas.

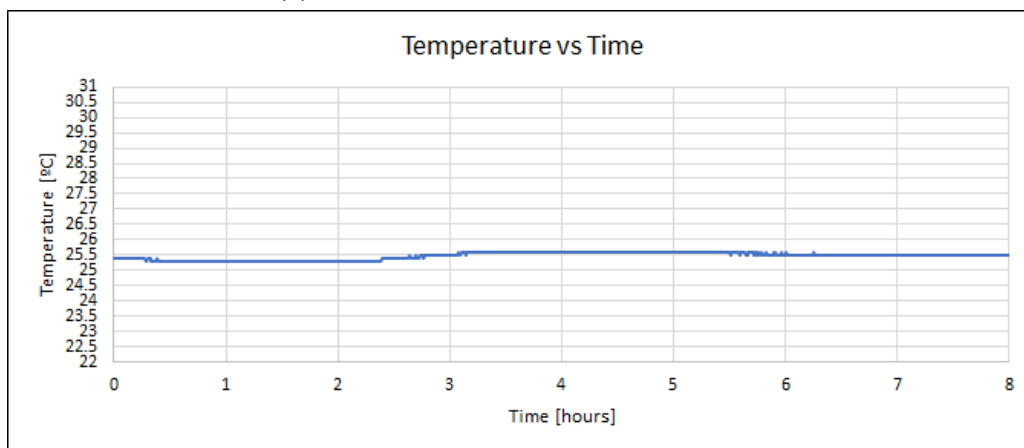
Temperatura ambiente

Este primeiro teste resume-se a efetuar leituras de temperatura, com termístor exposto ao ar e embebido em glicol. O sistema foi colocado numa sala fechada, virada para o Sol, levando a que a temperatura sentida nessa divisão fosse relativamente alta. Na Figura 41a tem-se os resultados do teste relativo ao termístor exposto ao ar, sendo de notar que a duração máxima foi de 2 horas e 30 minutos, sensivelmente. A flutuação que se vê nos primeiros minutos de teste vão de acordo com o sucedido no momento, onde a movimentação do defensor de tase nas imediações do termístor deverá ter levado a pequenas turbulências no ar, correspondendo a picos de temperatura entre os 25 e 30 °C. No restante tempo do teste, o sistema permaneceu imóvel nessa sala sem ninguém dentro da mesma e com a porta fechada, levando a uma maior estabilidade nas leituras seguintes. De

notar que entre os 60 e 80 primeiros minutos de teste, identificam-se flutuações na temperatura na ordem dos 2,5 °C, não sendo possível arranjar justificação para as mesmas. Estas variações levam a que o grau de confiança nestas leituras seja baixo. Na Figura 41b, tem-se o resultado em relação ao termístor embebido em glicol. Ao fim de 2 horas e 30 minutos, os registos de temperatura situavam-se todos na ordem dos 25,5 °C. De forma a ter uma melhor leitura na flutuações de temperatura, prolongou-se o teste para as 8 horas totais.



(a) Temperatura ambiente - Exposição ao ar.



(b) Temperatura ambiente - Embebido em glicol.

Figura 41: Comparação da temperatura ambiente entre exposição ao ar e ao glicol.

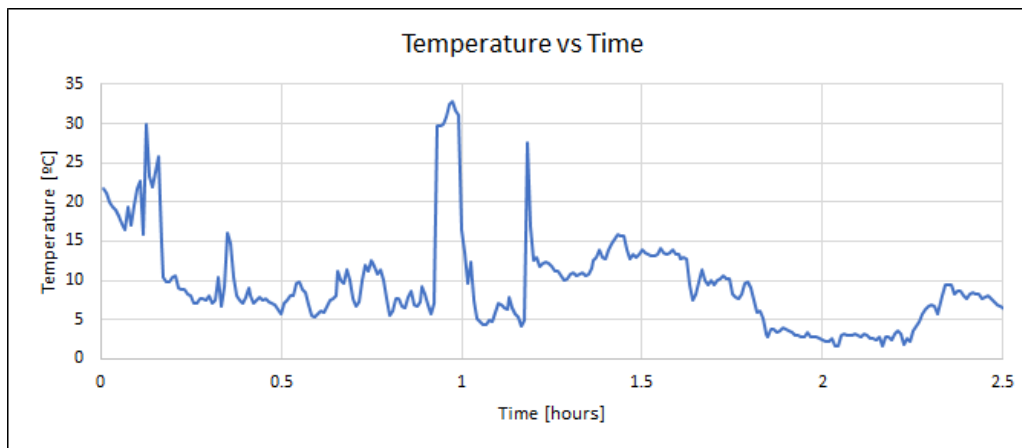
De notar que ambas as figuras têm a escala de temperatura igual, situada entre os 22 e 31 °C, desta forma torna-se mais fácil avaliar os resultados obtidos. O aumento do tempo de teste com o glicol não se considera uma variável introduzida uma vez que o que se encontra em avaliação é a variação da temperatura ao longo do tempo. A conclusão que se retira é que o glicol teve o resultado esperado, permitiu que as flutuações de temperatura fossem controladas e levou a uma representação da temperatura sentida mais fiável. Com o termístor exposto ao ar, os resultados falam por si mesmo, onde se vê grandes variações nas temperaturas recolhidas que diminuem a fiabilidade que cada leitura traz.

Frigorífico

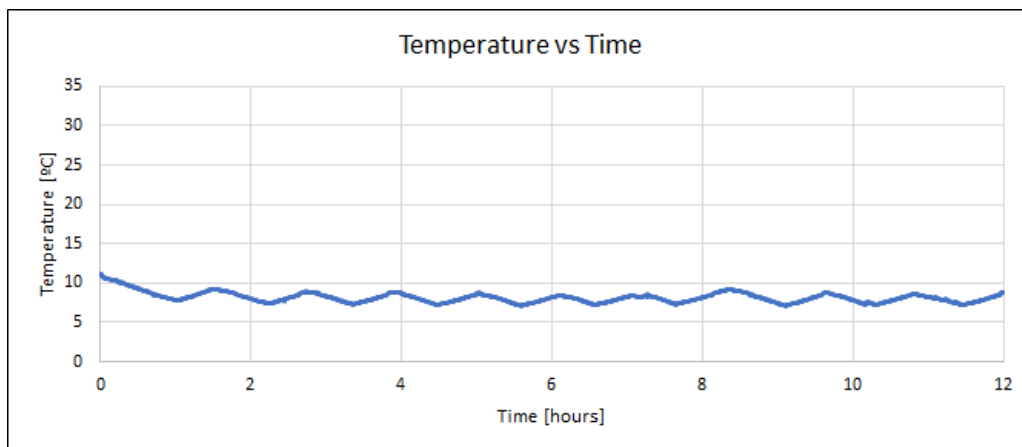
O segundo teste realizado foi dentro de um frigorífico. Aqui adotou-se a mesma metodologia, colocar o termístor em exposição ao ar e de seguida, embebido em glicol. Uma vez mais, com um tempo de teste mínimo de 2 horas, colocou-se o sistema dentro do frigorífico, estando este regulado para os 5 °C. Os dados referentes

à exposição ao ar encontram-se na Figura 42a. Nos primeiros momentos do teste, visualizam-se temperaturas na ordem dos 20 e 25 °C, apresentado inúmeras oscilações entre leituras. Estas oscilações iniciais justificam-se pela movimentação do ar circulante dentro do frigorífico. Durante a preparação do teste, a porta do frigorífico esteve aberta, levando a que tivesse entrado ar à temperatura ambiente. Fechada a porta e iniciado o teste, o ar ambiente origina turbulências que são visíveis nas temperaturas recolhidas. De forma a testar o impacto de uma abertura de porta, algo que aconteceria naturalmente, perto da primeira hora de teste, foi aberta a porta do frigorífico durante 1 minuto, indo de acordo com um abrupto aumento da temperatura regista. Esta decisão tornou-se uma mais valia para o teste uma vez que é notório o impacto real no aumento da temperatura do ar circulante dentro do frigorífico. Por lapso, foi aberta a porta do frigorífico uma segunda vez por poucos instantes, e esse registo ocorreu ao fim de 10 minutos após a primeira hora de teste, sendo caracterizado por um novo aumento repentino na temperatura registada. As oscilações que se identificam ao longo de todo o teste (após a primeira hora de teste) podem ser justificadas pelo sistema de anti-congelamento característico do frigorífico. Onde é injetado ar exterior para evitar o congelamento da zona onde se encontra o condensador. Esta justificação pode ser incorreta, uma vez que a informação encontrada é reduzida e varia entre fabricantes, ainda assim, dá uma ideia ao porquê de oscilações que por vezes rondam os 10 °C acima da temperatura expectável.

Na segunda parte do teste no frigorífico, com o termístor embebido em glicol, a temperatura do glicol encontrava-se à temperatura ambiente, que rondava os 25 °C, e o teste arrancou sendo essa a temperatura inicial. Ao fim de 2 horas de teste sensivelmente, notou-se que o sistema se encontrava a reiniciar constantemente. Mediu-se a tensão das baterias não recarregáveis e identificou-se que as mesmas já não tinham a tensão necessária para garantir estabilidade. Trocaram-se as mesmas e iniciou-se novo teste. Neste momento, a temperatura do glicol já não se encontrava à temperatura ambiente, e nos momentos iniciais do teste, a ver pela Figura 42b, a temperatura estava pouco acima do 10 °C. Deixou-se o mesmo a funcionar durante 12 horas seguidas e verificou-se que a temperatura oscilava entre os 7,5 °C e os 9,5 °C. Uma vez mais, o aumento do tempo de teste não trás impacto ao comportamento que se pretende observar, flutuações nas temperaturas registadas. O frigorífico tem um termostato que permite o ligar e desligar do compressor, levando à períodos de circulação do líquido refrigerante no sistema de refrigeração (correspondendo aos momentos em que a temperatura decresce) e períodos onde não existe circulação (correspondendo ao momentos de aumento de temperatura). Como dito, o frigorífico estava programado para os 5 °C contudo, nunca foi possível alcançar essa temperatura, sendo a média das temperaturas registadas na ordem dos 8 °C, levantando à hipótese de que a condutividade térmica do termostato é superior à do glicol.



(a) *Frigorífico - Exposição ao ar.*



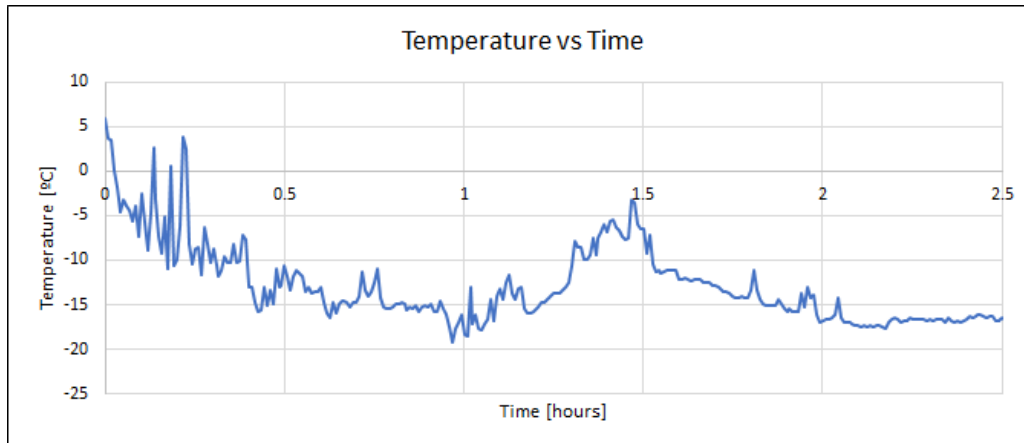
(b) *Frigorífico - Embebido em glicol.*

Figura 42: *Temperatura registada dentro um frigorífico comparando exposição ao ar e ao glicol.*

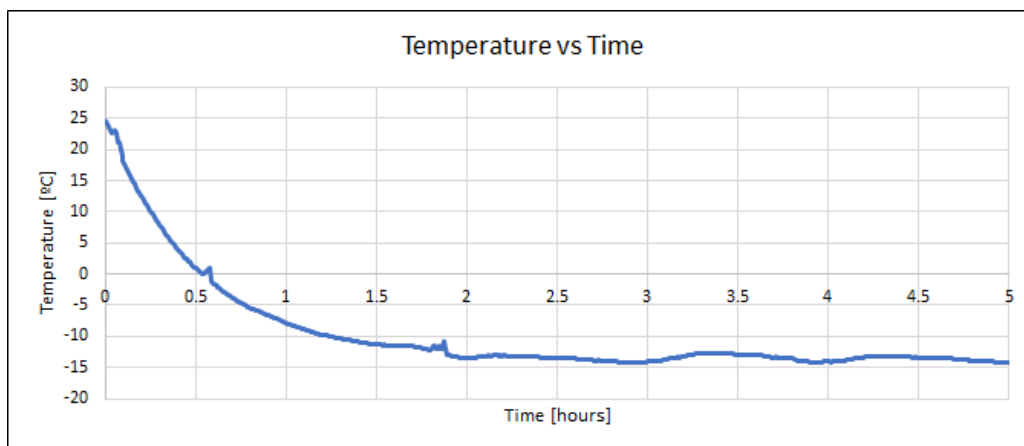
Congelador

Por fim tem-se o teste realizado com o sistema dentro do congelador configurado para operar nos $-17\text{ }^{\circ}\text{C}$. Numa primeira instância, colocou-se o termistor exposto ao ar circulante do congelador. Na Figura 43a vê-se que a temperatura registada encontra-se a decrescer até ao minuto 60 de teste. Registam-se oscilações entre os -10 e os $3\text{ }^{\circ}\text{C}$ no pior dos casos. Uma explicação para estes picos passará pelo tempo que foi necessário ter a porta do congelador aberta de forma a preparar o teste. No continuar do teste, houve um segundo aumento na temperatura ao fim da primeira hora, este foi resultante da abertura total da porta durante 1 minuto. De seguida, a porta não foi totalmente fechada, ficando com pequena saliência por onde o ar exterior poderia entrar. Deixou-se assim durante 30 minutos sensivelmente e esse fenómeno verifica-se no gráfico da temperatura. O restante período do teste a porta permaneceu fechada e nota-se um novo decréscimo da temperatura registada. A segunda parte do teste passou por colocar o termistor dentro do glicol. O líquido do glicol encontrava-se à temperatura ambiente, $25\text{ }^{\circ}\text{C}$ sensivelmente, e viu a sua temperatura diminuir ao longo de 2 horas, Figura 43b. Ao fim de meia hora de teste, abriu-se a porta do congelador na sua totalidade durante 1 minuto, sendo isso mesmo observado por um ligeiro pico na temperatura registada. Perto da segunda hora de teste, foi aberta uma segunda vez, durante 5 minutos, e uma vez mais notou-se um ligeiro aumento na temperatura registada. Como visto na experiência realizada no frigorífico, após estabilizada a temperatura, ao fim da segunda hora de

teste, a temperatura sobe e desce na ordem dos 2,5 a 3 °C, indo de acordo com a regulação da temperatura por parte do congelador via o termóstato que detém. Estando regulado o mesmo para -17 °C, regista-se que a temperatura média alcançada foi de -14 °C, sensivelmente, forçando a hipótese anteriormente levantada de que existe uma diferença considerável na condutividade térmica entre o termóstato e o termístor embebido em glicol - a diferença média de temperatura situa-se nos 3 °C, tanto neste teste como no anterior. Se o termóstato atinge a temperatura alvo mais rápido que o termístor embebido em glicol, retira-se a conclusão de que existe um *offset* que teria de ser ajustado.



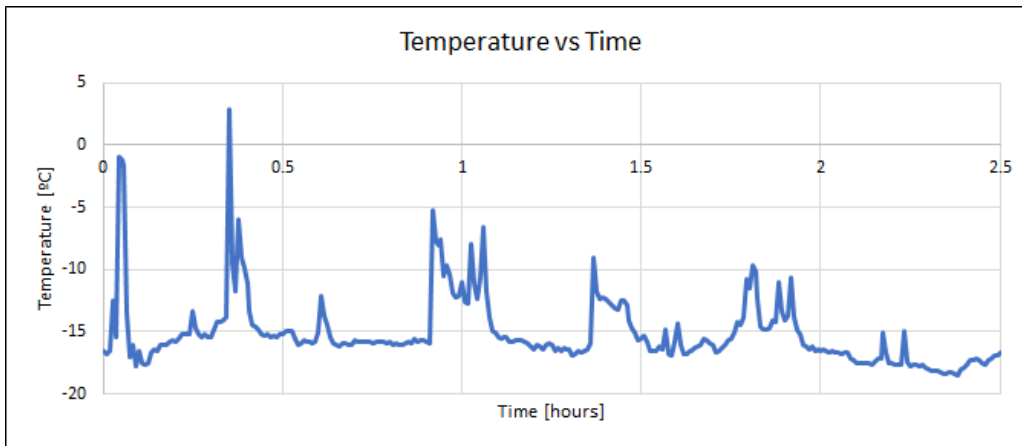
(a) Congelador - Exposição ao ar.



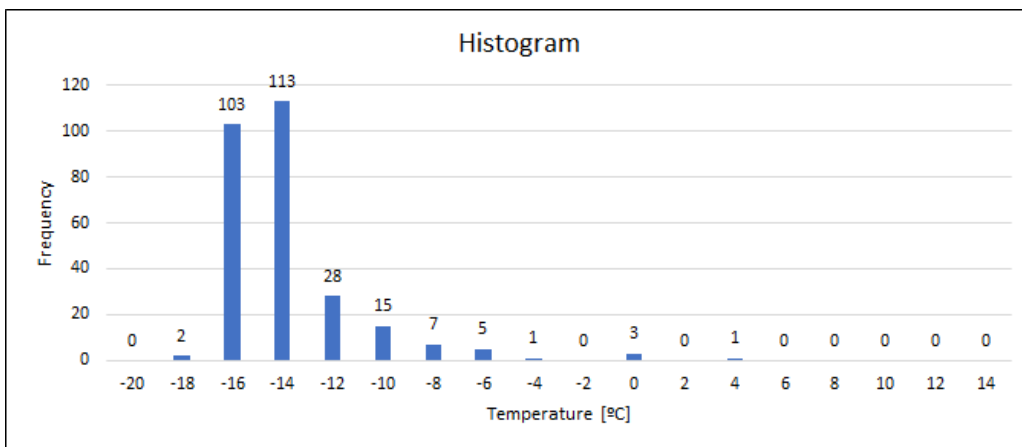
(b) Congelador - Embebido em glicol.

Figura 43: Temperatura registada dentro um congelador comparando exposição ao ar e ao glicol.

Foi tomada a decisão de realizar uma vez mais esta experiência com o objetivo de aproximar a casos reais. Foi colocado o termístor exposto ao ar dentro do congelador e ao fim de 1 hora ligou-se o sistema, sendo pretendido que o termístor alcançasse o equilíbrio térmico. Durante duas horas e meia de teste, o congelador permaneceu com a porta fechada e, observando o resultado na Figura 44a, vê-se uma grande instabilidade nas leituras registadas. Uma explicação para este fenómeno passa pelo sistema de anti-congelamento do congelador, que poderá injetar ar não refrigerado de forma a descongelar as superfícies que vão, naturalmente, acumulando gelo derivado da humidade que entra a cada abrir e fechar de porta. Na Figura 44b tem-se um histograma da experiência realizada. Sabendo que o congelador se encontrava regulado para os -17 °C, e contando com um *offset* de temperatura de 3 °C, verifica-se que a temperatura de maior evidência é de -14 °C, tendo sido registado temperaturas acima dos -8°C, havendo um registo que atingiu os 4 °C.



(a) Congelador - Exposição ao ar, segundo teste.



(b) Congelador - Exposição ao ar, histograma das temperaturas registadas

Figura 44: Temperatura do ar sentida dentro um congelador.

Por fim, realizou-se um teste de abertura de porta por um período de tempo prolongado. Decidiu-se então deixar a porta com uma abertura de 2-3 centímetros durante um período de 45 minutos, e o resultado, visto na Figura 45, é o empiricamente esperado. Pouco antes da primeira hora de teste, iniciou-se um aumento gradual da temperatura durante os 45 minutos em que a porta permaneceu aberta e, após o fecho, um decrescer de temperatura ao longo de um período de 2 horas. Uma vez mais, não se atingiu a temperatura alvo de $-17\text{ }^{\circ}\text{C}$, tendo esta estabilizado entre os $-13\text{ }^{\circ}\text{C}$ e $-15\text{ }^{\circ}\text{C}$, havendo uma vez mais um *offset* de $3\text{ }^{\circ}\text{C}$ em relação à temperatura alvo.

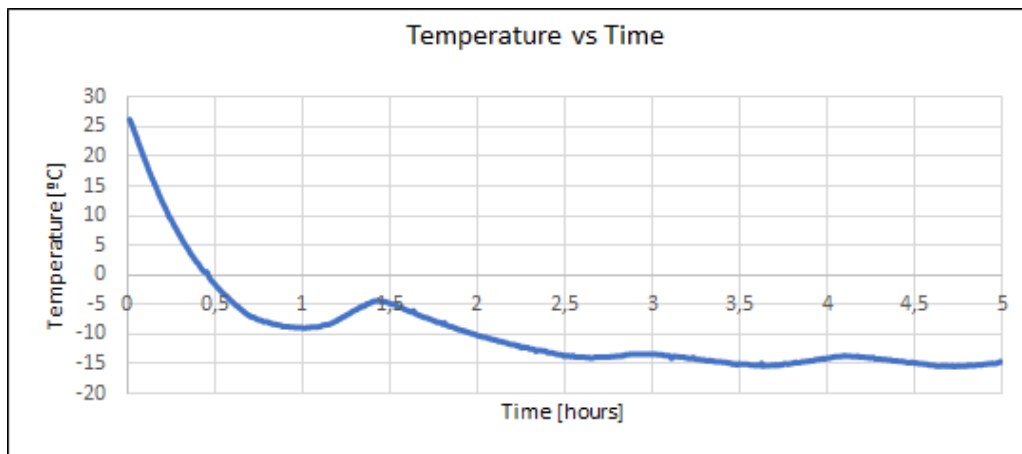


Figura 45: Evolução da temperatura perante uma abertura de porta do congelador.

Consumo energético

Por fim mediui-se os consumos energéticos nas diferentes fases de atividade. Na medição número 1, Tabela XIII , mediui-se a corrente do MCU no seu modo de operação *Standby* e *Active*; medição número 2 a 4, isolou-se o Módulo Rádio e mediui-se a corrente nos modos de operação *Sleep*, *Standby* e *Reset*.

Durante os testes realizados notou-se correntes bastante elevadas ao nível do módulo de rádio com este em *sleep*, sendo registado 17,3 mA. Modificou-se o código de forma a colocá-lo em modo *standby* e registou-se 17,6 mA. Não havendo nenhuma melhoria, decidiu-se colocar em modo *reset* sempre que possível e, registou-se desta forma, uma corrente de 1,2 mA. Com uma tensão de alimentação de 3,27 V, reduziu-se o consumo energético de 56,6 mW para 3,9 mW durante o período de inatividade. Se assumirmos a periodicidade mínima de 15 minutos, tem-se que o *dutty cycle* é bastante reduzido tendo em conta que, o tempo necessário entre envio de dados e receção do *acknowledge* teve uma duração de 2 segundos, sensivelmente. Posto isto em média, no pior caso de periodicidade, tem-se um consumo médio de 5,17 mW.

Tabela XIII: Corrente medida na Unidade de Sensor em diferentes contextos.

Medição	Descrição	Consumo
1	MCU	<i>Standby</i> - 0,1 mA; <i>Activo</i> - 1,9 mA
2	Módulo rádio	<i>Sleep</i> 17,3 mA
3	Módulo rádio	<i>Standby</i> 17,6 mA
4	Módulo rádio	<i>Reset</i> 1,2 mA

Comunicação rádio

Durante a fase inicial de desenvolvimento, a antena colocada no módulo de rádio resumia-se a um fio de cobre com 20 cm de comprimento que correspondia a 1/4 de comprimento de onda. Isto levou a uma distância de comunicação entre Unidade de Sensor e Hub de poucos centímetros, sendo registado cerca de 50 cm de distância de comunicação. Sendo pretendido comunicação rádio, é de esperar que seja possível colocar uma Unidade de Sensor numa divisão e que na divisão seguinte tivesse uma outra e uma Hub, por exemplo. Embora que o requisito funcional para a Unidade de Sensor seja "comunicação externa é assegurada de forma

wireless” e se tenha conseguido comunicar de uma forma *wireless*, 50 centímetros de distância de comunicação não se considera suficiente e portanto não cumpre com este requisito não funcional do projeto. Desta forma resolveu-se adquirir uma antena calibrada, para uma frequência de operação de 433 MHz e com ligação à massa, por fim de melhorar esta distância. Com esta alteração, registou-se ainda assim apenas 1 metro de distância de comunicação. Experimentou-se diversos modos de comunicação de forma a tentar perceber o que estava a degradar a distância de comunicação, contudo, não se encontrou uma justificação para estas diminutas distâncias. Não sendo possível trocar o módulo por outro de forma a testar a possibilidade de um destes encontrar-se defeituoso, conclui-se que a utilização do mesmo não é fiável. Esta conclusão é abordada na secção ”Hardware Final” onde se explora uma nova implementação para a comunicação rádio.

5.2 Hub

Nesta subsecção tem-se várias figuras que referem o comportamento registado na Hub e dos diversos módulos que a constituem. De forma a avaliar as diferentes fases do algoritmo implementado, visto no capítulo 4 - Implementação, colocou-se o MCU a enviar informação relevante para a sua UART. Com um adaptador USB para UART, visualizou-se num PC essas mensagens provenientes do MCU.

Funcionamento nos diversos modos de operação

Após a configuração do módulo de rádio, a Hub passa pela deteção do *hardware* presente no sistema. No exemplo da Figura 46a tem-se o mecanismo de deteção de uma Hub Celular/Wi-Fi. Foi enviado o comando ”AT” para o ”Módulo Celular” e foi recebida a resposta ”OK” por parte do mesmo, levando à mensagem exposta pelo MCU ”Cellular Module - OK”. Algo semelhante se sucedeu com o ”Módulo *Webserver*”, visto na Figura 46b, onde foi enviada a mensagem ”WebserverModuleCheck’ para a UART do Raspberry Pi e a resposta específica do mesmo permitiu decidir que o módulo também se encontrava presente.

```
--Checking available hardware...
AT
Cellular Module - OK
WebServerModuleCheck
WebServerModuleCheck
WebServer Module - OK
WiFi Module - NOT FOUND
--Waiting DT...
```

(a) Detecção de hardware - RX e TX da UART no MCU.

```
pi@raspberrypi:/media/pi/PEN $ sudo python UART.py
running...
t0: 1583766680.02
PID: 2669

WebServerModuleCheck

Module Check!
WebServerModuleCheck
```

(b) Detecção de hardware - RX e TX da UART ao nível do Raspberry Pi.

Figura 46: Detecção do Módulo Celular e do Raspberry.

Quando a Unidade de Sensor se inicializa pela primeira vez, não tem em contexto o seu ID, como tal, envia um pedido de novo ID à Hub. Foi então recebido esse pedido e enviado para a Unidade de Sensor e, estando presente o "Módulo *Webserver*", enviou-se a configuração *default*, delineado a vermelho na Figura 47a, para um novo sensor. O código em *python* que corre no Raspberry Pi, faz uma chamada HTTP com o método "GET" ao *localhost*. Essa chamada levará a uma inserção na base de dados através do código PHP desenvolvido. A resposta da chamada HTTP vê-se na Figura 47b com o texto "Update - Query success!".


```
--Waiting DT...
-----Message Acquired-----ID
WebServerSensorConfig@01@+25@0000000000@
----ID sent: 0001
--Waiting DT...
-----Message Acquired-----PD----Sensor ID: 0001
----Period sent: 0001
--Waiting DT...
```

(a) Sensor novo (MCU) - Configuração automática na presença de um novo sensor.

```
pi@raspberrypi:/media/pi/PEN $ sudo python UART.py
running..
t0: 1583766680.02
PID: 2669

WebServerModuleCheck

Module Check!
WebServerModuleCheck

Sensor Config!
{'usercontact': '000000000', 'maxtemperature': '+25', 'sensor': '01'}
Update - Query success!
200
WebServerSensorConfig@01@+25@0000000000@
```

(b) Sensor novo (Raspberry) - Receção de um novo sensor e registo em base de dados.

Figura 47: Configuração de um novo sensor por parte do MCU e envio dessa informação para o Raspberry.

Na Figura 48a tem-se um caso onde a Unidade de Sensor de ID "1", enviou uma temperatura de 26 °C, estando o mesmo configurado para uma temperatura de alarme de 25° C, levando então a uma situação de alarme. Estando o *webservice* presente em sistema, é enviado para o mesmo a informação para se registar uma nova temperatura e também o registo de um alarme, Figura 48b. Neste ponto o sensor pode ser configurado por meio do *website* construído. Definiu-se uma temperatura de alarme de 28 °C, o utilizador responsável pelo mesmo e enviado para a Hub essa nova configuração. O envio de temperatura seguinte, novamente de 26 °C, já não gerou um alarme, provando uma correta configuração do sensor via interface.

```

--Waiting DT...
-----Message Acquired-----DT----Sensor ID: 0001
--ACK success
--Processing DT
26.0
SensorCurrentTemperature: +26.0
SensorAlarmTemperature: 25
-Alarm situation
MCU: Alarm!
WebServerDt@01@+26.0@
WebServerAl@01@+26.0@
--Waiting DT...
WBCONTACT@01@+28@91 3
WBCONTACTSuccess
-----Message Acquired-----DT----Sensor ID: 0001
--ACK success
--Processing DT
26.0
SensorCurrentTemperature: +26.0
SensorAlarmTemperature: 28
WebServerDt@01@+26.0@
--Waiting DT...

```

(a) Receção de temperaturas com alarme e, pós configuração, sem alarme (MCU).

```

Data received!
{'sensorid': '01', 'alarm': '0', 'temperature': '+26.0'}
2020-03-09
16:24
+26.0
temperatureQuery success!
200
WebServerDt@01@+26.0@

Data received!
{'sensorid': '01', 'alarm': '1', 'temperature': '+26.0'}
2020-03-09
16:24
+26.0
alarmsQuery success!
200
WebServerAl@01@+26.0@

WBCONTACTSuccess

Data received!
{'sensorid': '01', 'alarm': '0', 'temperature': '+26.0'}
2020-03-09
16:25
+26.0
temperatureQuery success!
200
WebServerDt@01@+26.0@

```

(b) Receção de dados de temperatura e alarme (Raspberry).

Figura 48: Modificação da temperatura de alarme.

Continuando com a configuração do sensor por meio do *website*, tem-se a alteração da periodicidade. Quando se pretende a atualização da periodicidade da Unidade de Sensor, decidiu-se que a forma mais económica, do ponto de vista energético, seria reiniciar a mesma. Desta forma, mantém-se a Unidade de Sensor em modo de baixo consumo de energia durante o maior período de tempo possível. Como tal, quando esta inicializa, se já tiver configurado em memória o seu ID, realiza um pedido de periodicidade à Hub. Configurou-se então na interface para uma periodicidade de 30 em 30 minutos e enviou-se para a Hub. Esta recebeu a mensagem "WBPERIOD@01@02" que é interpretada como alteração da periodicidade para "02" no sensor "01", Figura 49. A Unidade de Sensor pediu esse valor à Hub quando foi reiniciada e, vê-se pela figura mencionada, que foi enviado a periodicidade "02" (o formato na figura refere "0002" mas é meramente a forma como se está a fazer a conversão do valor decimal em "ASCII" no envio para a UART"). Uma vez que a periodicidade tem uma granularidade de 15 minutos, a Unidade de Sensor pega no valor recebido e multiplica-o por 15 minutos.

```
WBPERIOD RECEIVED
WBPERIOD@01@02
WBPERIODSuccess
-----Message Acquired-----PD----Sensor ID: 0001
----Period sent: 0002
--Waiting DT...
```

Figura 49: Periodicidade alterada no Website e, por fim, enviada para Unidade de Sensor.

Via aplicação móvel, foi enviado um SMS com a configuração de temperatura de alarme e contacto telefónico para o sensor de ID "01". A Hub lê periodicamente os SMS que se encontram no "Módulo Celular" e, neste caso, foi detetada uma nova configuração de um sensor por meio da mensagem "MAUPDSEN@01@+24@91*****3", Figura 50. Vê-se também pela figura que na temperatura recebida que se sucedeu a configuração, a temperatura de alarme já se encontrava devidamente configurada para 24 ° C.

```
-----Message Acquired-----DT----Sensor ID: 0001
AT+CMGR=0

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
T+CMGR: "REC UNREAD", "+35191[REDACTED]3", "", "20/03/12, 12:03:54+00"
MAUPDSEN@01@+24@91[REDACTED]3

OK

@@@@@@@@@@@@@@@@@@@@
SMS "MAUPDSEN" found!
MAUPDSENSuccess
AT+CMGD=0
-----Message Acquired-----DT----Sensor ID: 0001
--ACK success
--Processing DT
23.1
SensorCurrentTemperature: +23.1
SensorAlarmTemperature: 24
```

Figura 50: Configuração do sensor realizada via aplicação Android.

O mesmo se sucedeu em relação à alteração de periodicidade via aplicação móvel. Foi enviado um SMS com o texto "MAUPDPER@01@02" e, à semelhança do que se viu quando a periodicidade foi alterada via *website*,

se a Unidade de Sensor solicita-se sua periodicidade, seria enviado o valor "02", Figura 51.

```
--Waiting DT...
-----Message Acquired-----DT----Sensor ID: 0001
--ACK success
--Processing DT
23.6
SensorCurrentTemperature: +23.6
SensorAlarmTemperature: 25
AT+CMGR=0

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
T+CMGR=0
+CMGR: "REC UNREAD", "+35191[REDACTED]3", "", "20/03/12,14:21:55+00"
MAUPDPER@01@02

OK

@@@@@@@@@@@@@@@@@@@@
SMS "MAUPDPER" found!
MAUPDPERSuccess
AT+CMGD=0
```

Figura 51: Periodicidade alterada via aplicação Android, SMS captado pelo MCU.

Através da aplicação móvel foi enviado um SMS com o pedido das configurações atuais dos sensores que se encontram no sistema. Pela Figura 52 tem-se que foi rececionado uma temperatura de 22.8 °C onde se vê que a temperatura de alarme é de 30 °C. Seguidamente, foi encontrado o SMS com o texto "MASENCFG" e foi enviado então o SMS de resposta, para o mesmo número de telefone, com o texto "MASENCFG@01+3091*****3" que, do lado da aplicação móvel, será visto como um único sensor configurado uma vez que as configurações dos sensores estão separadas por cada "@".

```
WebServerDt@01@+22.8@
-----Message Acquired-----DT----Sensor ID: 0001
--ACK success
--Processing DT
22.9
SensorCurrentTemperature: +22.8
SensorAlarmTemperature: 30
-----Message Acquired-----DT----Sensor ID: 0001
AT+CMGR=0

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
T+CMGR=0
+CMGR: "REC UNREAD", "+35191[REDACTED]3", "", "20/03/12,10:09:18+00"
MASENCFG

OK

@@@@@@@@@@@@@@@@@@@@
SMS "MASENCFG" found!
AT+CMGF=1
AT+CMGS="+35191[REDACTED]3"
MASENCFG@01+3091[REDACTED]3
```

Figura 52: Receção do SMS que solicita o envio das configurações atuais dos sensores.

Recebeu-se uma temperatura de 22.7 °C onde a temperatura máxima é de -16 °C, posto isto, e estando o Módulo Celular no sistema como visto anteriormente na Figura 46a, foi enviado um SMS para o número de telefone configurado.

```
-----Message Acquired-----DT----Sensor ID: 0001
--ACK success
--Processing DT
22.7
SensorCurrentTemperture: +22.7
SensorAlarmTemperture: -16
SensorCurrentTemperture >= SensorAlarmTempertureMCU: Alarm!
WebServerDt@01@+22.7@
WebServerAl@01@+22.7@
Sending SMS...
AT
AT+CPIN?
AT+CPIN=1294
AT+CMGF=1
AT+CMGS="91[redacted]3"
Temperature alarm!
Sensor ID: 1
Max temperature: -16 C
Current temperature: +22.7 C
Reply "Stop alarm sensor 1" to disable this alarm.
--Waiting DT...
```

Figura 53: Envio de alarme

Quando ocorre um alarme para um determinado sensor, a Hub fica com essa informação em memória. É guardado o ID do sensor, a temperatura registada no momento do alarme e a temperatura de alarme, bem como uma indicação de envio de SMS prévio. Pela Figura 54 confirma-se isso mesmo. Foi recebida uma temperatura de 22,7 °C onde a temperatura de alarme é de -16 °C. Foi escrito para a UART, para contextualizar o momento no código, "Sending SMS..." contudo, a função responsável pelo envio do SMS identificou que já havia sido enviado um SMS para este ID de sensor e, por isso, não foi enviado. O processo que aguarda pelo *feedback* do utilizador foi ativado passados 5 minutos e, iniciou-se a procura pelo SMS de resposta, visto na Figura como "Check for 'Stop alarm' SMS". A primeira posição de memória do cartão SMS não se encontrava povoada com nenhum SMS, concluindo-se então que é necessário o reenvio do SMS. Indo à memória de configuração de sensores, retirou-se o número de telefone e, indo à ao armazenamento volátil de registos de alarmes, retirou-se a temperatura inicial que originou alarme, temperatura essa de 22,0 °C.

```
-----Message Acquired-----DT----Sensor ID: 0001
--ACK success
--Processing DT
22.7
SensorCurrentTemperture: +22.7
SensorAlarmTemperture: -16
SensorCurrentTemperture >= SensorAlarmTempertureMCU: Alarm!
WebServerDt@01@+22.7@
WebServerAl@01@+22.7@
Sending SMS...
--Waiting DT...
Check for "Stop alarm" SMS
AT+CMGR=0

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
T+CMGR=0
+CMS ERROR: Invalid memory index

@@@@@@@@@@@@@@@@
Send new alarm SMS!
Sending SMS...
AT
AT+CPIN?
CellularModule_PIN_Status= 1
AT+CMGF=1
AT+CMGS="91[redacted]3"
Temperature alarm!
Sensor ID: 1
Max temperature: -16 C
Current temperature: +22.0 C
Reply "Stop alarm sensor 1" to disable this alarm.
```

Figura 54: Reenvio de SMS de alarme.

5.3 Interface de Utilizador

Do ponto de vista da validação da interface gráfica, existe sempre a relação dos testes realizados anteriormente, onde são realizadas comunicações entre a interface e o MCU da Hub, esta não existe isoladamente, desta forma os testes estão relacionados. Contudo nem todos os pontos de validação assim são, nomeadamente o que se refere na Figura 55 onde foi registado na base de dados os dados referentes à leitura de temperaturas por parte da Unidade de Sensor. Nesta figura tem-se o gráfico dessas mesmas leituras realizadas num determinado dia. Durante a fase de testes da Hub, foi definido uma periodicidade inferior àquela descrita nos requisitos do projeto. Colocou-se no código da Unidade de Sensor que o fator de multiplicação da periodicidade recebida seria de "10", correspondendo a 60 segundos entre leituras, em vez de um fator de "150" que corresponde a 900 segundos, ou seja, 15 minutos. Posto isto, durante os testes foram recebidos mais dados que o normal e isso é notório na Figura 55, colocando de parte a quantidade de dados recolhidos, valida-se que o gráfico foi gerado com sucesso. Vê-se também nesta figura que no eixo horizontal que representa "Date & Time", não existem valores nesse eixo uma vez que este corresponde ao *timestamp* composto por data e hora. Isto colocaria muita informação nesse eixo e dificultava bastante a leitura do mesmo. Para visualizar o *timestamp*, clica-se no gráfico de forma a selecionar um dos registos. Isto leva a que apareça o detalhe composto pelo *timestamp* e temperatura recolhida.



Figura 55: Gráfico com as temperaturas recolhidas para um dado sensor para um intervalo de tempo.

Como visto nos testes realizados na Hub, foi mostrado que tinha sido feita uma alteração à configuração do Sensor. Na Figura 56 tem-se o ecrã que mostra a configuração atual do sensor e que permite uma nova configuração do mesmo. Atribuindo um valor de temperatura máxima (coluna "Change maximum temperature"), utilizador responsável (coluna "Change responsible user") e/ou periodicidade (coluna "Change period of readings") e carregando em "Apply Changes", é executado um comando, via PHP, que executará o *script* em Python que envia efetivamente a mensagem à Hub via UART. Essas comunicações são vistas na subcapítulo anterior nas Figuras 48a e 49.

Sensor ID	Maximum Temperature	Periodicity	Responsible User	Contact	Change maximum temperature	Change responsible user	Change period of readings	
1	-16	00:15	tiagoresende2	91	-	-	-	APPLY CHANGES
RETURN								

Figura 56: Página de visualização e configuração de sensores.

Achou-se importante registar alterações de temperatura máxima dos sensores, por razões de segurança, como tal foi definido nos requisitos isso mesmo. Na Figura 57 tem esse mesmo registo, contudo havendo inúmeras alterações ao valor de temperatura máxima durante os testes efetuados, é apenas mostrado um excerto ilustrativo. Dos testes realizados destaca-se um pequeno lapso na temperatura máxima registada, no momento

de configuração de um novo sensor, apresenta o sinal "+". Vê-se pela Figura 48b da subsecção anterior que quando se regista um novo sensor em base de dados, é enviado esse sinal "+". Nas configurações sucessivas de temperatura máxima, o sinal só é enviado para temperaturas negativas.

User who made the change	Sensor	Old maximum temperature	New maximum temperature	Date	Time
tiagoresende	1	+25	30	15-02-2019	15:15
tiagoresende	1	30	25	09-03-2020	13:25
tiagoresende	1	28	30	09-03-2020	16:10
tiagoresende	1	30	30	10-03-2020	12:26
tiagoresende	1	30	30	10-03-2020	12:26
tiagoresende	1	30	20	11-03-2020	18:07

Figura 57: Página com o registo de alterações das temperaturas máximas dos sensores.

Passando por fim ao teste da geração do relatório do registo de temperaturas, tem-se a Figura 58 onde foi definido o sensor de ID "1" para o mês de Março de 2020. Carregando no botão "OK", foi lida da base de dados os registos de temperatura e criado um ficheiro Excel para realização de *download*. Durante a fase de desenvolvimento, não se identificou com os testes unitários que a extensão do ficheiro estava a ser gerada como "xlsx_" em vez de "xlsx". No capítulo de anexos tem-se o anexo A.5 com a Figura 66 com o excerto do código PHP que gera o ficheiro Excel, neste excerto vê-se que o formato de output do ficheiro é "xlsx". Verificou-se que em determinados *browsers* o formato era, ainda assim, reconhecido como um ficheiro Excel. A correção deste *bug* passará por utilizar uma outra biblioteca PHP que permita a geração de ficheiros Excel, caso a biblioteca utilizada não permita contornar esta situação.

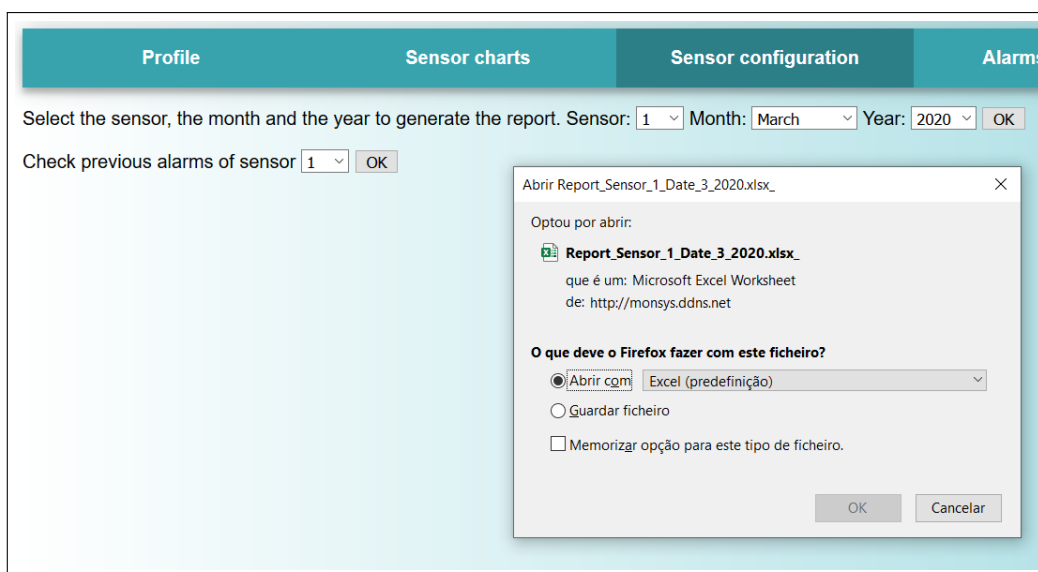
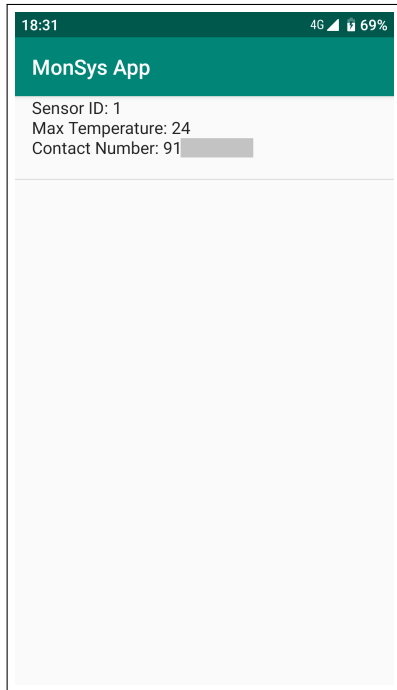
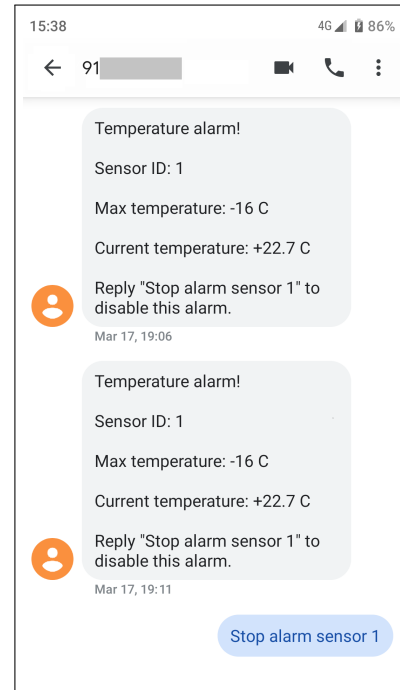


Figura 58: Geração do relatório com o registo de temperaturas.

Para terminar a fase de testes, menciona-se a aplicação móvel onde, uma vez mais, foi parcialmente testada na subcapítulo anterior com a leitura dos SMS recebidos via aplicação móvel bem como o envio de SMS com os dados de configuração ou de alarme. Relativamente às diferenças visuais em relação às figuras da subsecção 4.3.2 Aplicação Móvel, onde se caracterizou ecrã a ecrã a aplicação, estas resumem-se à visualização da configuração do sensor no sistema, Figura 59a e ao SMS recebido da notificação de alarme, bem como, o SMS enviado de *feedback*, Figura 59b. É visto também nesta última que foi reenviado o SMS de alarme uma vez que não tinha sido dado o *feedback* necessário.



(a) Estado de configurações dos sensor via aplicação móvel.



(b) SMS de alarme recebido.

Figura 59: Validação da aplicação móvel e notificações de alarme.

6 Hardware Final

No capítulo anterior identificou-se alguns pontos ao nível *hardware* proposto que precisam de ser abordados no presente capítulo, formalizando assim uma proposta de *hardware* final.

6.1 Unidade de Sensor

Uma solução encontrada para a escolha do microcontrolador foi o "CC1310 SimpleLink™" da Texas Instruments. A natureza deste System-on-Chip (SoC) passa por ter todos os periféricos necessários à implementação da Unidade de Sensor, já debatidas em pontos anteriores, bem como a integração de um módulo de rádio que compreende uma gama de frequências abaixo dos 1 GHz. Reduzido os componentes discretos, optando por esta solução integrada, reduz-se os pontos críticos de falha de sistema e permite uma redução do consumo energético. O módulo de rádio utilizado até então, consumia cerca de 20 mA durante a fase de transmissão a 10 dBm, o SoC proposto reduz esse valor, teoricamente, em 60%, consumindo apenas 13,4 mA. Esta implementação também reduziria o consumo energético despendido na comunicação entre o MCU e o módulo de RF, isto é, haveria uma menor utilização de portos I/O. A arquitetura da Unidade de Sensor sofre uma outra alteração que passa pela introdução de um MOSFET, Figura 60. Esta é justificada pela redução do consumo energético enquanto o MCU não realiza leituras da tensão no divisor. Tendo em conta que com uma periodicidade de quinze minutos, o tempo efetivo de medição da tensão é bastante reduzido, logo, a introdução do MOSFET no divisor de tensão, quando este no estado "off", permite colocar o divisor de tensão em alta impedância.

O algoritmo implementado é transparente ao *hardware*, a utilização em si do *hardware* é que necessita de modificações, isto é, a utilização dos portos de I/O, endereçamento às posições de memória e utilização de periféricos, *timers* por exemplo. O SoC em causa tem um sensor de temperatura incorporado, contudo, como visto pelos testes práticos realizados, as medições de temperatura recolhidas com o termístor embebido em glicol provaram-se bastante sólidas e coloca-se de parte a utilização deste sensor para a realização deste registo crítico.

O maior ponto de falha na Unidade de Sensor são as baterias. Se olharmos para o caso extremo de temperatura de operação a -30 °C, têm de ser selecionadas baterias que permitam esta gama de operação. As baterias utilizadas na prova de conceito foram as "MN1500" da "Duracell®", que têm por base zinco e dióxido de manganês, possuindo uma gama de operação que começa nos -20 °C^[52], não sendo esta suficiente para os -30 °C desejados. Um segundo ponto a ter em consideração na escolha será a capacidade. A bateria tem de ter uma capacidade suficientemente grande que permita dois anos de funcionamento com uma periodicidade de 15 minutos. Assumindo um caso extremo de tempos de operação, considera-se que o MCU CC1310, por cada período de 15 minutos, estaria dentro das situações:

- Activo^[53] - Modo de operação "activo" durante 3 segundos
- ADC - Tempo de leitura do ADC de 1 segundo (consumo no divisor de tensão)
- TX^[53] - Tempo de transmissão via RF de 1 segundo
- RX^[53] - Tempo de receção via RF de 1 segundo
- Standby^[53] - Modo de operação "standby" durante 895 segundos

A corrente média, em miliamperes, é dada pela equação 7.

$$Corrente_{Média} = \frac{(Activo * 3 + ADC + TX + RX + Standby * 10^{-3})}{15 * 60} [mA] \quad (7)$$

$$Corrente_{Média} = \frac{(1.9 * 2 + \frac{3}{1k+1k} + 23.4 + 5.5 + 0.8 * 10^{-3})}{15 * 60} = 0.0409 [mA] \quad (8)$$

Da equação 7 é deduzido que a capacidade de uma bateria de forma a garantir uma operação durante 2 anos é dada pela equação 9.

$$Capacidade_{Bateria} = 2 * 365 * 24 * Corrente_{Média} = 716,57 [mAH] \quad (9)$$

No subcapítulo 5.1, Teste do Sistema e Resultados Experimentais - Unidade de Sensor, mediu-se a corrente média para uma periodicidade de 15 minutos e registou-se 5,17 mW. Para uma tensão registada de 3,27 V, tem-se uma corrente de 1,58 mAH. Para um período de funcionamento de dois anos, é então necessária uma bateria, ou conjunto de baterias, com uma capacidade combinada de 27 682 mAH. Comparando este valor com o resultado da equação 9, conclui-se que o *hardware* final proposto reduz a capacidade bateria por um fator de 38,6. O impacto desta redução influencia diretamente o custo final do produto e o volume físico ocupado pela Unidade de Sensor.

Posto isto, decide-se que uma opção válida para a bateria é por parte do fabricante "ANSMANN" com a referência "5020022"^[54]. Existem diversas opções válidas no mercado, optado por esta tem-se uma capacidade de 850 mAH e uma tensão nominal de 3 V. Oferece também uma gama de operação compreendida entre -40 e 60 °C e uma corrente máxima de descarga de 800 mA.

Tomada estas decisões para o "hardware" final, o esquema final para a Unidade de Sensor encontra-se apresentado na Figura 60.

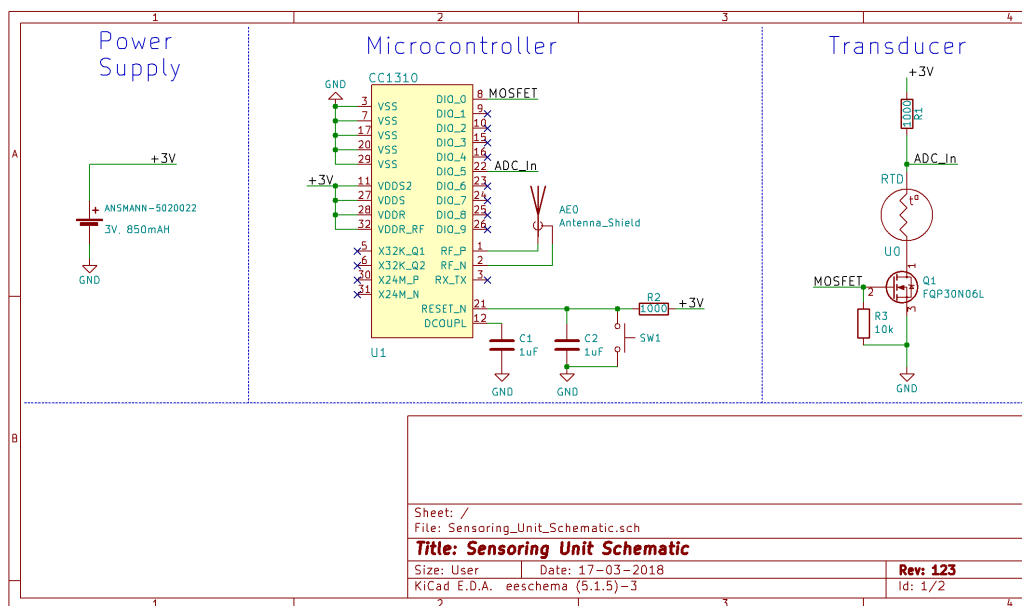


Figura 60: Esquema elétrico da Unidade de Sensor com o hardware final proposto.

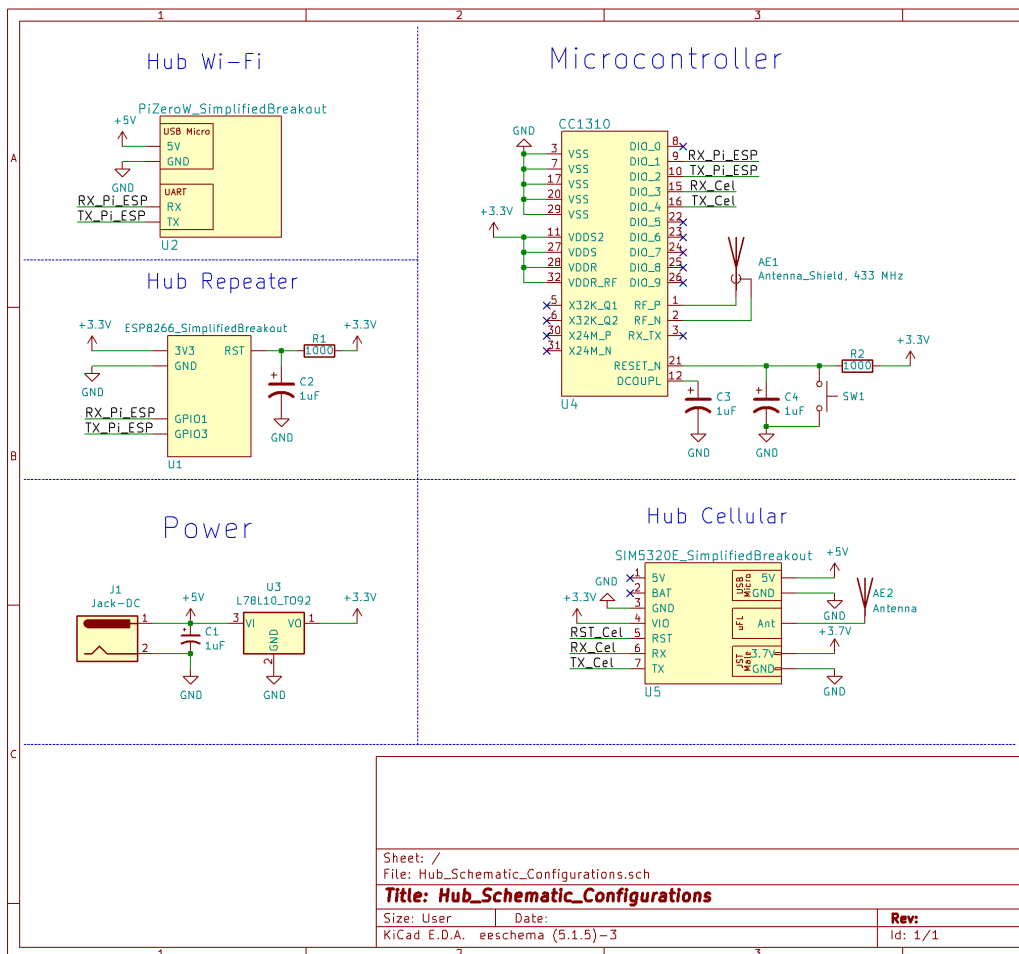
6.2 Hub

Um dos pontos de melhoria na Hub seria no modo de funcionamento em que a Hub hospeda um servidor *web*. Optou-se originalmente pela utilização de um "Raspberry Pi 1 B" com o sistema operativo "Raspbian" e uma ligação Wi-Fi via um adaptador de USB para 802.11n. A solução proposta para o *hardware* final passa pela utilização do "Raspberry Pi Zero W", onde oferece capacidade de Wi-Fi nativamente. Do ponto de vista de performance, ambos oferecem 512 GB de memória RAM e o mesmo CPU (BCM2835) mas com a diferença do "Raspberry Pi Zero W" operar a uma frequência superior, 1 GHz em vez de 700 MHz^[55]. Do ponto de vista económico, consegue-se uma grande redução de custos optando pelo "Raspberry Pi Zero W". Existe uma grande flutuação no mercado e analisando o preço dos vários vendedores, o que se pode aferir é que existe uma redução de preço de 40% quando comparando ambas as opções. Indo ao modelo mais recente que a "Raspberry Pi Foundation" oferece, "Raspberry Pi 4 Model B", o modelo "Raspberry Pi Zero W" reduz os custos em 65%. A relevância deste ponto é importante uma vez que reduzirá o custo do produto final.

Tendo em conta que a "Hub Repeater" que se resolveu com um "NodeMCU", não se encontra grande vantagem em substituir este pelo "Raspberry" mencionado. Uma vez que o "NodeMCU" apresenta um custo inferior e atinge os mesmos objetivos que o "Raspberry", a única vantagem passaria pela redução de componentes distintos e permitiria ainda uniformizar o sistema. Desta forma, a avaliação do *hardware* presente por parte do MCU mantém-se inalterada, sendo que o porto de comunicação é o mesmo no caso de uma "Hub Wi-Fi" e "Hub Repeater".

Em termos de custo monetário em *hardware*, o constituinte do sistema da prova de conceito mais caro é o "breakout" para as comunicações celulares. Este é um ponto crítico por onde se pode investir na escolha de um fornecedor que ofereça melhores relações de preço. O desempenho do "SIM5320A" da "SIMCom" revelou-se bastante sólido nos testes práticos realizados, não tendo sido encontrada nenhuma dificuldade na receção e envio de SMS's. Neste aspeto, considera-se que um "breakout" alternativo que utilize o modelo "SIM5320" resolveria este ponto.

A Figura 61 apresenta o esquema elétrico com os diferentes modos de operação da Hub. Os "breakouts" encontram-se simplificados, estando apenas apresentados os pinos indispensáveis à utilização dos mesmos, isto é, todos os que não tem uma conexão foram omitidos. Salienta-se a conexão denominada "RX_Pi_ESP" e "TX_Pi_ESP" que são comuns ao "breakout" do "Raspberry Pi Zero W" e ao "NodeMCU", uma vez que ambos os componentes não se encontram em simultâneo no sistema.



Sheet: /		
File: Hub_Schematic_Configurations.sch		
Title: Hub_Schematic_Configurations		
Size: User	Date:	Rev:
KiCad E.D.A.	eeschema (5.1.5)-3	Id: 1/1

Figura 61: Esquema elétrico da Hub com o hardware final proposto.

7 Conclusões e Trabalhos Futuros

Chegando a este ponto da dissertação, está na altura de se olhar para todo o trabalho desenvolvido até então e não só relembrar o que foi feito, mas também analisar, de forma crítica, a solução realizada. Dessa forma, o primeiro aspeto a referir trata-se do problema que foi imposto na realização desta tese de mestrado, que consiste no desenvolvimento de um sistema cuja função é monitorizar a temperatura dentro de equipamentos de refrigeração médicos, de uma forma mais económica e escalável em comparação às soluções hoje encontradas no mercado. Oferecendo ainda a possibilidade de se optar por diferentes configurações de sistema, monitorizar remotamente equipamentos, receber alarmes em situações específicas e configurar o sistema por meio de uma aplicação móvel ou página *web*.

De modo a possibilitar uma mais fácil compreensão de toda a crítica final ao trabalho realizado, a mesma foi dividida em três instâncias, onde na primeira serão abordados todos os objetivos propostos, na segunda tem-se a discussão de todas as funcionalidades desenvolvidas e na terceira, e última categoria, encontram-se os trabalhos futuros a serem desenvolvidos no contexto da tese.

A presente tese de mestrado foi introduzida ao defensor de tese durante o estágio profissional, ao abrigo do IEFP, realizado após graduação em Licenciatura em Engenharia Biomédica, no Serviço de Utilização Comum dos Hospitais ou, resumidamente, SUCH. É uma "associação privada sem fins lucrativos, tutelada pelos Ministérios da Saúde e das Finanças", com um foco na "prestação de serviços na área hospitalar"^[56]. Sendo um dos seus valores o "desenvolvimento sustentável", foi apresentado o problema discutido no início deste documento. Problema esse que, parafraseando o engenheiro tutor do estágio profissional, resume-se a um custo demasiado elevado na implementação generalizada de sensores medidores de temperatura nos hospitais em Portugal. Havendo uma oportunidade no mercado de apresentar um produto capaz de competir com a concorrência, não só na solução final em si, mas também no impacto económico da solução. Posto isto, definiram-se requisitos funcionais e não funcionais para um sistema modular que iria recolher medições de temperatura de equipamentos de refrigeração médicos onde seria possível visualizar dados e realizar configurações no sistema por meio de uma interface num *browser* e, em alternativa, numa aplicação móvel para o sistema operativo "Android". Em cima disto, seria necessário montar mecanismos de alerta que iria notificar um utilizador que um determinado sensor se encontrava fora dos parâmetros de temperatura definidos. A tese proposta e desenvolvida desafiou o defensor de tese na medida em que havia uma inexperiência no desenvolvimento em linguagens de programação que não fossem o tradicional "C", passando pelas linguagens "LUA", "PHP", "HTML", "MySQL", "Python", e "Java". A projeção de um sistema modular e a forma como diversos componentes iriam ser integrados no ecossistema proposto, provou-se um desafio interessante e aliciante ao desenvolvimento.

Foi então projetado e desenvolvido um sistema que cumpria com esses requisitos balizado numa implementação de prova de conceito que daria profundidade a uma implementação final necessária à comercialização do produto. Referenciando as Tabelas IV a VIII do capítulo 3.2 Requisitos do Sistema, retira-se que alguns requisitos de menor prioridade que não foram atingidos. O requisito número 5 da Tabela IV que menciona "Encapsulamento resistente a produtos de limpeza" não foi alcançado, o número 8 da Tabela V refere "Fonte de alimentação fornecida a partir da rede e bateria de reserva" não houve desenvolvimento no que toca à "bateria de reserva" e, por fim, o requisito 5 da Tabela VI "Armazenamento de 5 760 mensagens provenientes de sensores" não foi contemplado. Os restantes requisitos, encontrados nas Tabelas IV a VIII da secção 3.2 Requisitos do Sistema, foram alcançados com sucesso, tendo sido possível testar do ponto de vista prático o sistema num caso

com semelhanças a um funcionamento real. Os maiores fatores que levaram a uma implementação demorada resumiram-se ao dispositivo de transmissão rádio selecionado. O "RFM69HCW" foi selecionado tendo por base a documentação técnica do fabricante, "HopeRF"[®], que se provou pouco fiável pelas incoerências encontradas na funcionalidade do módulo de rádio. Com uma medíocre distância de comunicação de 1 metro, testado em condições de laboratório, revelou-se a inutilidade do mesmo. Colocando a performance do módulo de rádio de parte e focando nos objetivos alcançados, conseguiu-se construir um sistema modular que permite albergar trinta Unidades de Sensor que se encontraram configurados ao nível de uma unidade central de processamento, a Hub. A Hub foi testada nos seus diversos modos de funcionamento, tendo sido possível receber um SMS com a informação de que determinado sensor encontrava-se em alarme, receber o SMS de reforço quando não se forneceu o *feedback* necessário para cessar o alarme e, por fim, realizou-se a configuração dos sensores presente em sistema via aplicação móvel e via interface *web*. O sistema torna-se escalável pelo fato de que a Unidade de Sensor é construída com componentes de baixo custo, sendo ela apenas constituída, tendo por base o *hardware* final, um microcontrolador CC1313 da Texas Instruments Inc., uma antena de 433 MHz, um transdutor do tipo termoresistência e uma bateria de 850 mAH, sensivelmente. Tendo a possibilidade de enviar dados dentro de uma rede Wi-Fi local, torna-se possível a implementação desta solução em grandes instalações, não obstante de que para instalações de menor dimensão, a modularidade inerente torna-o uma opção viável. Tocando um pouco na interface gráfica, esta provou-se bastante funcional uma vez que permite monitorizar e configurar remotamente o sistema, havendo apenas a necessidade de o utilizador realizar trocas de SMS com a Hub onde, esses SMS, teriam os custos associados à operadora da rede móvel. O *design* gráfico, tanto da aplicação móvel como a página *web*, considera-se básica uma vez que são compostos por páginas estática e não dinâmicas. A justificação passa por durante a fase de desenvolvimento, a ênfase caiu na funcionalidade das páginas e não no seu aspeto gráfico. Como já mencionado no presente capítulo, houve objetivos que não foram alcançados. De momento não é possível guardar informação de dados de temperatura registados localmente, isto é, a Hub em si não guarda informação de dados de temperatura. Teria de ser implementado um mecanismo de sincronização entre a Hub e a base de dados, onde sempre que não fosse possível estabelecer ligação à base de dados, os dados teriam que ser guardados em memória não volátil para posterior sincronismo. O consumo energético da Hub não foi abordado durante o desenvolvimento, contudo, para garantir o requisito desta possuir uma bateria de reserva, garantindo um funcionamento de 48 horas, a capacidade da bateria não seria inconcebível ainda assim, seria de desenvolver um mecanismo de alerta que notifica-se um utilizador de que a Hub estaria a funcionar apenas com a bateria.

Após o término de todo o projeto, podemos olhar para o mesmo como uma boa plataforma para futuros desenvolvimentos neste âmbito. Posta toda a análise crítica, o ponto fulcral seria na conceção de um protótipo funcional do *hardware* final caracterizado, passando por uma implementação inicial de um protótipo da Unidade de Sensor que teria todas as condições de comunicar com a Hub conceptual, seguindo da implementação de um protótipo da Hub com os diversos módulos que podem ser acoplados. Sendo sempre necessário garantir que todos os cenários de comunicação entre dispositivos conseguem ser assegurados, testado uma vez mais a comunicação rádio com o novo *hardware* e garantir que a implementação do sistema numa rede Wi-Fi consegue ser realizada sem grandes obstáculos. Esta sistema poderia evoluir para uma Unidade de Sensor modular, sendo possível associar novos transdutores e visualizar novas métricas na interface gráfica. Um objetivo claro seria a possibilidade de registar níveis de humidade, com a necessidade de aprimorar o código de forma a que este novo parâmetro pudesse ser propagado entre os diferentes pontos de comunicação.

8 Bibliografia

Referências

- [1] Monnit, “Monnit Local Alert.” <https://www.monnit.com/Products/Local-Alert/Local-Alert/Wireless-Local-Alert>, 2016. Acedido a Dezembro 2017.
- [2] Vaisala, “DL1016/1416 Temperature Logger For demanding temperature monitoring.” <https://www.vaisala.com/en/products/instruments-sensors-and-other-measurement-devices/instruments-industrial-measurements/dl1016-1416>, 2017. Acedido a Dezembro 2017.
- [3] Centers for Disease Control and Prevention, “Immunization recommendations and guidelines.” <https://www.cdc.gov/vaccines/hcp/admin/storage/index.html>, Dezembro 2014. Acedido a Dezembro 2017.
- [4] “The vaccine cold chain: Maintaining cool links.” ICN - International Council of Nurses, 2019.
- [5] World Health Organization, “Blood cold chain.” <http://www.who.int/bloodsafety>, Dezembro 2012. Acedido a Dezembro 2017.
- [6] “29th standards for blood banks and transfusion services.” Advancing Transfusion and Cellular Therapies Worldwide, 2019.
- [7] Centers for Disease Control and Prevention, “CFR - Code of Federal Regulations Title 21.” <https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=205.50>, November 2016. Acedido a Dezembro 2017.
- [8] INFARMED, “Sobre o Infarmed - INFARMED, I.P.” <https://www.infarmed.pt/web/infarmed/perguntas-frequentes-area-transversal/sobre-o-infarmed>, . Acedido a Agosto 2020.
- [9] “Regulamento relativo às Boas Práticas de Distribuição.” INFARMED, Março 2015.
- [10] U.S. Food and Drug Administration, “Pinkbook | Vaccine Storage and Handling | Epidemiology of VPDs.” <https://www.cdc.gov/vaccines/pubs/pinkbook/vac-storage.html>, Abril 2017. Acedido a Dezembro 2017.
- [11] Miranda Allen, “The True Cost of Improper Vaccine Storage.” <https://www.linkedin.com/pulse/true-cost-improper-vaccine-storage-miranda-allen>, Janeiro 2017. Acedido a Dezembro 2017.
- [12] Phase IV Engineering Inc., “Refrigerator / Freezer Wireless Temperature Sensor Network Bundles Archives.” <https://www.phaseivengr.com/product-category/battery-powered-wireless-sensors/wireless-sensor-network-wsn/starter-kits-wsn-all-types/refrigerator-freezer-wireless-temperature-sensor-network-bundles>, 2014. Acedido a Dezembro 2017.
- [13] Accsense Monitoring Systems and Data Loggers Inc., “Monitor Vaccine Temperature in Hospitals & Clinics.” <http://accsense.com/news/monitor-vaccine-temperature-in-hospitals-clinics>, 2016. Acedido a Dezembro 2017.
- [14] “A2-05W 3 Input WiFi Connected Temperature Pod - Specification Sheet.” Accsense, 2016.
- [15] “A2-05 3 Input LAN Wired Temperature Pod - Specification Sheet.” Accsense, 2016.

- [16] Monnit, “Monnit International 3G Cellular Gatewayst.” <https://www.monnit.com/Products/Gateways/Cellular-Gateway/International-3G-Cellular-Gateways>, 2016. Acedido a Dezembro 2017.
- [17] Monnit, “Monnit Wireless Temperature Sensor - Commercial AA Battery Powered.” <https://www.monnit.com/Product/MNS-8-W2-TS-ST-L03>, 2016. Acedido a Dezembro 2017.
- [18] Monnit, “Monnit Wireless Temperature Sensor - Commercial Coin Cell Powered.” <https://www.monnit.com/Product/MNS-8-W1-TS-ST-L03>, 2016. Acedido a Dezembro 2017.
- [19] Monnit, “Monnit Wireless Low Temperature Sensor - Commercial AA Battery Powered.” <https://www.monnit.com/Product/MNS-8-W1-TS-LT>, 2016. Acedido a Dezembro 2017.
- [20] Monnit, “Monnit Wi-Fi Temperature Sensors.” <https://www.monnit.com/Product/MNS-2-WF-TS-ST-L03>, 2016. Acedido a Dezembro 2017.
- [21] Monnit, “Monnit Ethernet Gateway.” <https://www.monnit.com/Products/Gateways/Ethernet-Gateway/Wireless-Ethernet-Gateways>, 2016. Acedido a Dezembro 2017.
- [22] Monnit, “Monnit Local Alert.” <https://www.monnit.com/Products/Local-Alert/Local-Alert/Wireless-Local-Alert>, 2016. Acedido a Dezembro 2017.
- [23] Phase IV Engineering, “Wireless Sensor Product Index.” <https://www.phaseivengr.com/wireless-sensor-index-page>, 2016. Acedido a Dezembro 2017.
- [24] Phase IV Engineering, “Internet User Interface.” <https://www.phaseivengr.com/product/internet-user-interface-sensormanager-com-annual-subscription-wireless-sensor-network-wsn-starter-kit>, 2016. Acedido a Dezembro 2017.
- [25] Phase IV Engineering, “Ethernet Gateway – Wireless Sensor Network (WSN) Active Sensor System.” <https://www.phaseivengr.com/product/refrigerator-freezer-wireless-temperature-sensor-network-bundle-ethernet-gateway>, 2016. Acedido a Dezembro 2017.
- [26] Phase IV Engineering, “Cellular Gateway with 24 Hour Battery Back-Up – Wireless Sensor Network (WSN).” <https://www.phaseivengr.com/product/refrigerator-freezer-wireless-temperature-sensor-network-battery-cellular-gateway>, 2016. Acedido a Dezembro 2017.
- [27] Phase IV Engineering, “Refrigerator and Freezer Monitor – Wireless Temperature Sensor with Tethered Probe and Thermal Buffer.” <https://www.phaseivengr.com/product/refrigerator-freezer-monitor-wireless-temperature-sensor-tethered-probe-compact>, 2016. Acedido a Dezembro 2017.
- [28] Vaisala, “Business Areas and Customers.” <https://www.vaisala.com/en/about-vaisala/business-areas-and-customers>, 2017. Acedido a Dezembro 2017.
- [29] Vaisala, “Refrigerator and Freezer Temperature Monitoring.” <https://www.vaisala.com/en/industries-innovation/life-science/refrigerator-and-freezer-temperature-monitoring>, 2017. Acedido a Dezembro 2017.
- [30] “Vaisala Continuous Monitoring System Common Configurations - Technology Description.” Vaisala, 2012.

- [31] Vaisala Online Store, “Fixed Price Repair and Accredited Calibration for DL1016 Data Logger.” <https://store.vaisala.com/eu/fixed-price-repair-and-accredited-calibration-for-dl1016-data-logger/DL1016SERV/dp?nosto=right-block>, 2015. Acedido a Dezembro 2017.
- [32] Sensaphone, “WSG Wireless Ultra Low Temperature Sensor with External Probe.” <https://www.sensaphone.com/products/wsg-wireless-ultra-low-temperature-sensor-w-external-probe.php>, 2018. Acedido a Março 2018.
- [33] Sensaphone, “Sensaphone WSG30 Monitoring System.” <https://www.sensaphone.com/industries/medical-cold-storage/product/sensaphone-wsg30.php>, 2018. Acedido a Março 2018.
- [34] S. K. M. D. Z. T., “Counterspeculation, auctions and sealed tenders,” *Wiley-Interscience*, vol. , pp. 129–130, 2007.
- [35] “Radio Regulations.” International Telecommunication Union, Edição de 2012.
- [36] “Bluetooth low energy wireless technology backgrounder.” Nordic Semiconductor, Versão 4: Atualizado a 22 Março 2011.
- [37] P. Smith, “Comparisons between Low Power Wireless Technologies.” Cambridge Silicon Radio, Edição 1: 22 Maio 2011.
- [38] A. Tomar, “Introduction to ZigBee.” Element 14, Volume 1: Julho 2011.
- [39] “Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements.” Tektronix, 2013.
- [40] A. S., “An overview of wireless communication technologies used in wireless sensor networks,” *Conference Paper*, Setembro 2013.
- [41] “Bluetooth Low Energy.” LitePoint, Junho 2012.
- [42] S. J. Dementyev A.; Hodges S.; Taylor S., “Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario,” *Proceedings of IEEE International Wireless Symposium (IWS)*, Abril 2013.
- [43] Texas Instruments, “BLE - FAQ - Texas Instruments Wiki.” <http://processors.wiki.ti.com/index.php/BLE>, Julho 2015. Acedido a Dezembro 2017.
- [44] National Instruments, “Overview of Temperature Sensors.” <http://www.ni.com/white-paper/4218/en>, Agosto 2016. Acedido a Dezembro 2017.
- [45] “The European Table of Frequency Allocation and Applications in the Frequency Range 8.3 kHz to 3000 GHz.” European Communications Office, Outubro 2017.
- [46] Adafruit Industries, “Adafruit Learning System.” <https://learn.adafruit.com/assets/31773>, . Acedido a Abril 2019.
- [47] “74HC4050 - Hex non-inverting HIGH-to-LOW level shifter.” NXP Semiconductors, 5 Fevereiro 2016. Rev. 4.
- [48] “AP2112 - 600mA CMOS LDO Regulador with Enable.” Diodes Incorporated, Junho 2017. Rev. 2.

- [49] “NTC thermistors for temperature measurement.” EPCOS, Fevereiro 2019.
- [50] “Alkaline Manganese Dioxide - Handbook and Application Manual.” Energizer, 2018.
- [51] The PHP Group, “PHP: password_hash - Manual.” <https://www.php.net/manual/en/function.password-hash.php>, . Acedido a Abril 2020.
- [52] “Alkaline-Manganese Dioxide Battery MN1500.” Duracell, 2016.
- [53] “CC1310 SimpleLink™ Ultra-Low-Power Sub-1 GHz Wireless MCU.” Texas Instruments, Setembro 2015 – Revisto Outubro 2016.
- [54] “CR-2 Lithium Battery.” ANSMANN, 2018.
- [55] Raspberry Pi Foundation, “FAQs - Raspberry Pi Documentation.” <https://www.raspberrypi.org/documentation/faqs/>, . Acedido a Fevereiro 2020.
- [56] SUCH, “SUCH - Serviço de Utilização Comum dos Hospitais Feed.” <https://www.such.pt/pt/apresentacao/>, . Acedido a Abril 2020.

A Anexos

A.1 Código C implementado na Unidade de Sensor para a Equação Steinhart-Hart

```
ADCAverage = 0;
for(i=0; i < NumSamples; i++){
    ADCAverage += ADCSamples[i];
}

ADCAverage /= NumSamples; //Average the ADC result
ADCAverage = 1023/ADCAverage - 1;
ADCAverage = SeriesResistor/ADCAverage;
Steinhart = ADCAverage/ThermistorNominal;
Steinhart = logf(Steinhart);
Steinhart /= BCoefficient;
Steinhart += 1.0/(TemperatureNominal + 273.15);
Steinhart = 1.0/Steinhart;
Steinhart -= 273.15;
TemperatureMSB = 0;
TemperatureLSB = 0;
TemperatureMSB = (signed char)Steinhart;
if(Steinhart<0)TemperatureLSB = (Steinhart - TemperatureMSB)*-10;
else TemperatureLSB = (Steinhart - TemperatureMSB) * 10;
Mode = SendDataMode;
```

Figura 62: Código implementado - Equação de Steinhart-Hart.

A.2 Código C implementado na Unidade de Sensor para o reenvio de dados e recepção do *acknowledge*

```
case WaitingACKMode:
    Radio_RxMode();
    if(P2IN & BIT0){
        PayloadReady = 0;
        AcquireMessage(data_buffer[0]);
        if(MessageAcquired == 1){
            if(message_Rx[3] == ACK){
                P1OUT |= BIT7;
                Mode = WaitingPeriodMode;
                for(i=0;i<13;i++)buffer_Tx[i] = 0;
                counter = 0;
                counter_period = 0;
            }
            MessageAcquired = 0;
        }
    }
    if(counter > 2){ //Timeout, send again
        buffer_Tx[11] = Retry; //timeout flag
        Mode = SendDataMode;
        counter = 0;
        counter_period = 0;
    }
    break;
```

Figura 63: Código implementado - Reenvio de dados e recepção de *Acknowledge*.

A.3 Código Python implementado no RaspberryPi que interpreta as comunicações via UART

```
urlwritetemperature = "http://localhost/writedata/writetemperature.php"
urlsensorconfig = "http://localhost/writedata/writesensorconfig.php"
ser = serial.Serial("/dev/ttyAMA0", 9600, timeout=1)
Rx = ""

def main():
    global t0
    try:
        while True:
            time.sleep(1)
            try:
                Rx = ser.readline()
            except:
                pass
            if(Rx == "WebServerModuleCheck\r\n"):
                print("Module Check!")
                ser.write("OK")
            elif("WebServerDt" in Rx or "WebServerAl" in Rx):
                print("Data received!")
                ser.write("OK")
                x = [pos for pos, char in enumerate(Rx) if char == '@']
                sensorid = Rx[x[0]+1:x[1]]
                temperature = Rx[x[1]+1:x[2]]
                if("WebServerDt" in Rx):
                    payload = {'sensorid': sensorid, 'temperature': temperature, 'alarm':"0"}
                else:
                    payload = {'sensorid': sensorid, 'temperature': temperature, 'alarm':"1"}

                r = requests.get(urlwritetemperature, params=payload)
```

Figura 64: Código implementado - Comunicação UART entre RaspberryPi e MCU.

A.4 Código PHP implementado no Webserver no ato de registo de novo utilizador

```
$username = $mysqli->escape_string($_POST['username']);
$first_name = $mysqli->escape_string($_POST['first_name']);
$last_name = $mysqli->escape_string($_POST['last_name']);
$department = $mysqli->escape_string($_POST['department']);
$contact = $mysqli->escape_string($_POST['contact']);
$password = $mysqli->escape_string(password_hash($_POST['password'],
    PASSWORD_BCRYPT));
$hash = $mysqli->escape_string(md5(rand(0,1000)));
$result = $mysqli->query("SELECT * FROM users WHERE username='$username'");
if($result->num_rows > 0){
    $_SESSION['message'] = "The username already exists!";
    header("location:login/error.php");
}
else{
    $mysqli->query("SET character_set_results = 'utf8', character_set_client =
        'utf8', character_set_connection = 'utf8', character_set_database =
        'utf8', character_set_server = 'utf8'");
    $result = $mysqli->query("SELECT * FROM `users` WHERE 1 ");
    if($result->num_rows < 1){
        $sql = "INSERT INTO users (username, first_name, last_name, department,
            contact, usertype,password, hash, active) "
            . "VALUES ('$username','$first_name','$last_name', '$department','$
            contact', 'admin','$password','$hash', '1')";

        if($mysqli->query($sql): $_SESSION['message'] = "This is the
            first account ever created, therefore you have administrator
            privilege. You can now login.";
        else: $_SESSION['message'] = "Something went
            wrong, try again later.";
        endif;
    }
    else{
        $sql = "INSERT INTO users (username, first_name, last_name, department,
            contact, password, hash) "
            . "VALUES ('$username','$first_name','$last_name', '$department','$
            contact','$password','$hash')";

        if($mysqli->query($sql): $_SESSION['message'] = "And
            administrator must now active this account.";
        else: $_SESSION['message'] = "Something went
            wrong, try again later.";
        endif;
    }
}
header("location:login/error.php");
```

Figura 65: Código implementado - Registo de novo utilizador.

A.5 Código PHP implementado no Webserver na geração do relatório de temperaturas registradas.

```
require_once dirname(__FILE__) . '/../Classes/PHPExcel.php';
$objPHPExcel = new PHPExcel();
$objPHPExcel->getActiveSheet()->setCellValue('A1', 'Date');
$objPHPExcel->getActiveSheet()->setCellValue('B1', 'Time');
$objPHPExcel->getActiveSheet()->setCellValue('C1', 'Temperature');
$Row = 2;
while($r = $result->fetch_assoc()){
    $StrRow = 'A' . $Row;
    $objPHPExcel->getActiveSheet()->setCellValue($StrRow, $r['col_date']);
    $StrRow = 'B' . $Row;
    $objPHPExcel->getActiveSheet()->setCellValue($StrRow, substr($r['col_time'], 0, -3));
    $StrRow = 'C' . $Row;
    $objPHPExcel->getActiveSheet()->setCellValue($StrRow, $r['temperature']);
    $Row++;
}
$objPHPExcel->getActiveSheet()->setTitle('Sensor ID - '.$SensorToShow);
header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header('Content-Disposition: attachment;filename=Report_Sensor"'.$SensorToShow.'_'Date_'.'$
    MonthToShow.'_'.'$YearToShow.'.xlsx"');
header('Cache-Control: max-age=0');
$objWriter = PHPExcel_IOFactory::createWriter($objPHPExcel, 'Excel2007');
$objWriter->save('php://output');
```

Figura 66: Código implementado - Geração de relatório.

A.6 Código LUA implementado no NodeMCU no modo de *Access Point*.

```
wifi.ap.config({ssid="ESP-"..node.chipid(),pwd="ESP-"..node.chipid()})
wifi.ap.setip({ip="192.168.111.1",netmask="255.255.255.0",gateway="192.168.111.1"})
wifi.setmode(wifi.SOFTAP, true)
print("Server created on "..wifi.ap.getip())
tmr.alarm(1, 1000, 1, function()
  if(wifi.ap.getip() ~= desiredIP) then
    print("Desired IP is wrong: ".. wifi.ap.getip())
    node.restart()
  else tmr.stop(1) end
end)
server=net.createServer(net.TCP)
payload = nil
server:listen(80, function(conn)
  conn:on("receive", function(conn, data)
    payload = data
    conn:send("HTTP/1.1 200 OK\r\n\r\n")
  end)
  conn:on("sent", function(conn)
    print(payload)
    local wifiConfig = {}
    if file.open("AP.html", "r") then
      while true do
        tmr.delay(10000)
        line = file.readline()
        if (line == nil) then break
        end
        conn:send(string.sub(line, 1, -2))
        tmr.wdclr()
      end
      file.close()
    else
      print("file not found!")
    end
    conn:close()
    local ap=nil
    local psw=nil
    local hubType=nil
    local thisHubIP=nil
    local mainHubIP=nil
    if(payload ~= nil) then
      for ap in string.gmatch(payload,"ap=(.)&") do table.insert(wifiConfig, ap) end
      for psw in string.gmatch(payload,"psw=(.)&") do table.insert(wifiConfig, psw) end
      for hubType in string.gmatch(payload,"hubType=(.)&") do table.insert(wifiConfig, hubType) end
      for thisHubIP in string.gmatch(payload,"thisHubIP=(.)&") do table.insert(wifiConfig, thisHubIP) end
      for mainHubIP in string.gmatch(payload,"mainHubIP=(.)&") do table.insert(wifiConfig, mainHubIP) end
      if(wifiConfig[1]~=nil and wifiConfig[2] ~=nil and wifiConfig[3]~=nil
        and wifiConfig[4] ~=nil and wifiConfig[5] ~=nil)then
        tmr.alarm(1, 6000, 1, function()
          --print("after")
          wifiConfig[1]=string.gsub(wifiConfig[1], "+", " ")--in case of SSID with spaces
          wifiConfig[2]=string.gsub(wifiConfig[2], "+", " ")--in case of PSW with space
          print("ap: ".. wifiConfig[1])
          print("psw: ".. wifiConfig[2])
          print("hubType: ".. wifiConfig[3])
          print("thisHubIP: ".. wifiConfig[4])
          print("mainHubIP: ".. wifiConfig[5])
          file.open("configwifi.lua", "w")
          file.writeline('station_cfg={}')
          file.writeline('station_cfg.ssid="'..wifiConfig[1] .. "'')
          file.writeline('station_cfg.pwd="'..wifiConfig[2] .. "'')
          file.writeline('hubType="'..wifiConfig[3] .. "'')
          file.writeline('thisHubIP="'..wifiConfig[4] .. "'')
          file.writeline('mainHubIP="'..wifiConfig[5] .. "'')
          file.writeline('IPConfigured=0')
          file.close()
          node.restart()
        end)
      end
    end
  end)
end)
end)
end)
```

Figura 67: Código implementado - Modo *Access Point*.

A.7 Código LUA implementado no NodeMCU - Envio e recepção de dados via 802.11

```
uart.on("data", "\r",
function(data)
  receiveddata = {}
  local temperature = nil
  local sensorid = nil
  if(gpio.read(1) == 1) then
    for temperature in string.gmatch(data,"temperature={{(.-)}}")
      do table.insert(receiveddata, temperature) end
    for sensorid in string.gmatch(data,"sensorid={{(.-)}}")
      do table.insert(receiveddata, sensorid) end
    if(receiveddata[1] ~= nil and receiveddata[2] ~= nil) then
      do_get(receiveddata[1],receiveddata[2])
    end
  end
end, 0)

function do_get(data1,data2)
  srv = net.createConnection(net.TCP)
  srv:connect(server.port,server.domain)
  srv:on("receive", function(conn, payload)
    if (SERIAL_PRINT) then
      print(payload)
    end
  end)
  srv:on("connection", function(conn,payload)
    conn:send("temperature={" .. data1 .. "}sensorid={".. data2.."}")
  end)
end
```

Figura 68: Código implementado - Envio de dados de temperatura via 802.11.

```
function receiver(sck, payload)
  print(payload)
  local receiveddata = {}
  local temperature = nil
  local sensorid = nil
  if payload ~=nil then
    for temperature in string.gmatch(payload,"temperature={{(.-)}}")
      do table.insert(receiveddata, temperature) end
    for sensorid in string.gmatch(payload,"sensorid={{(.-)}}")
      do table.insert(receiveddata, sensorid) end
    if(receiveddata[1] ~= nil and receiveddata[2] ~= nil) then
      tmr.alarm(1, 1000, 1, function()
        uart.write(0, "WIFITMP@" .. receiveddata[1] .. "@" .. receiveddata[2])
        timeout = timeout + 1
        if receivedOK == 1 then tmr.stop(1)
        elseif timeout >= 60 then tmr.stop(1) end
      end)
    end
  end
  sck:close()
  collectgarbage()
end
collectgarbage()
server = net.createServer(net.TCP,30)
server:listen(80, function(conn)
  conn:on("receive", receiver)
end)
```

Figura 69: Código implementado - Recepção de dados de temperatura via 802.11.

A.8 Código JAVA implementado na aplicação móvel - Envio e receção de SMS

```
public void sendSMSMessage(String phoneNo){
    message = "MASENCFG";
    toastMessage = "SMS update request sent. Do \"Check from SMS inbox\" when the SMS response arrives."
    if(checkPermission(Manifest.permission.SEND_SMS)){
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(phoneNo, null, message, null, null);
        Toast.makeText(getApplicationContext(),toastMessage,Toast.LENGTH_LONG).show();
    }
    else{
        Toast.makeText(getApplicationContext(),"No permissions", Toast.LENGTH_LONG).show();
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.SEND_SMS},SEND_SMS_PERMISSION_REQUEST_CODE);
    }
}
```

Figura 70: Código implementado - Envio de SMS de pedido de dados de configuração

```
public void checkSMS(String hubPhoneNumber){
    int permissionCheck = ContextCompat.checkSelfPermission(this, Manifest.permission.READ_SMS);
    if (permissionCheck == PackageManager.PERMISSION_GRANTED){
        Uri inboxUri = Uri.parse("content://sms/inbox");
        smsList = new ArrayList<>();
        ContentResolver contentResolver = getContentResolver();

        Cursor cursor = contentResolver.query(inboxUri, null, null, null, null);
        int flag = 0;
        while (cursor.moveToNext()) {
            String number = cursor.getString(cursor.getColumnIndexOrThrow("address")).toString();
            String body = cursor.getString(cursor.getColumnIndexOrThrow("body")).toString();
            if (number.contains(hubPhoneNumber) && body.contains("MASENCFG@")) {
                for (int i = -1; (i = body.indexOf("@", i + 1)) != -1; i++) {
                    try{
                        sensor sensorCfg = new sensor();
                        sensorCfg.id = Integer.parseInt(body.substring(i + 1, i + 3));
                        sensorCfg.maxTemperature = Integer.parseInt(body.substring(i + 3, i + 6));
                        sensorCfg.contactNumber = body.substring(i + 6, i + 15);
                        //i = i + 16;
                        smsList.add("Sensor ID: " + sensorCfg.id + "\n" + "Max Temperature: " + sensorCfg.maxTemperature +
                            "\n" + "Contact Number: " + sensorCfg.contactNumber + "\n");
                        saveData(smsList);
                        flag = 1;
                        Log.d(TAG, "showContacts: number: " + number + "body: " + body);
                    } catch (Exception e) {
                        break;
                    }
                }
            }
        }
        cursor.close();
        if(flag == 1){
            Toast.makeText(getApplicationContext(),"Sensor config SMS found, check via storage now.",Toast.LENGTH_LONG).show();
        }
        else{
            Toast.makeText(getApplicationContext(),"No SMS found.",Toast.LENGTH_LONG).show();
        }
    }
    else{
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.READ_SMS}, PERMISSIONS_REQUEST_READ_CONTACTS);
    }
}
```

Figura 71: Código implementado - Receção de SMS com atuais dados de configuração

```

public void sendSMSMessage(){
    String phoneNo = hubPhoneNumber;
    String message = "MAUPDSEN";
    int messageCharsASCII;

    message = message+"@"+String.format("%02d", sensorID)+"@"+plusMinusNF.format(changeTemp)+"@"+changeContact;
    char [] messageChars = message.toCharArray();

    if(checkPermission(Manifest.permission.SEND_SMS)){
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(phoneNo, null, message, null, null);
        Toast.makeText(getApplicationContext(),"SMS sent!",Toast.LENGTH_LONG).show();
    }
    else{
        Toast.makeText(getApplicationContext(),"No permissions", Toast.LENGTH_LONG).show();
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.SEND_SMS},SEND_SMS_PERMISSION_REQUEST_CODE);
    }
}

```

Figura 72: Código implementado - Envio de SMS com os novos dados de configuração

A.9 Características dos produtos comercializados no mercado

Gateways									
Company	Cellular	Wi-Fi (802.11)	Ethernet	USB	Datalogger	Sensors per unit/gateway	Power Supply	Battery	
Description	Communicates to sensors, via RF (433, 868, 900, 915 or 920 MHz), and transmits information with GSM/UMTS technology		Communicates to sensors, via RF (433, 868, 900, 915 or 920 MHz), and transmits information over the internet through a Ethernet connection	Communicates to sensors, via RF (433, 868, 900 or 920 MHz), and transmits information to a PC with a USB connection	In the event of not having connection to the server, sensor messages will be stored in the gateway				
Accense	N	N	N	N	255 Samples	3	NA	NA	
Monnit	2G: GSM/GPRS, 850/900/1800/1900 MHz (100€) 3G: UMTS/GSM/GPRS, 800/850/900/2100 MHz (322€)	N	IEEE Standard: 802.3-2002 Protocols: UDP/DHCP/TCP/SNMP/MODBUS (194€)	Compatibility: Windows 10, 8.1, 8, 7 and Vista (92€)	N. of sensor messages: Cellular - 50 000 Ethernet - 16 000 USB - 14,336 up (to gateway) 2,048 down (to sensors)	Cellular and Ethernet: 500 USB: 100 (The real value depends on the subscription plan)	Cellular: 5.5 DC - 2.5 A Ethernet: 5.5V or Power-Over-Ethernet USB: Power through USB or external power supply	N	
Phase IV	(503€)	N	(503€)	N	20 000 Sensor messages	100	Y	Cellular: 24h back-up battery	
VAISALA	N	N	Y	N	45000	4	NA	NA	

Sensor									
Communication protocol									
Company	Wireless	Wi-Fi (802.11)	Ethernet	Power supply	Battery	Autonomy	Temperature range		
Description	RF < 1GHz					Battery autonomy for a given sampling rate			
Accense	N	802.11a/b/g/n Security: WEP, WPA, WPA2 Supports: 802.11i & 802.1x Supplicant	Y	100-240V AC 50-60Hz	Lithium Battery (4 hours at 1 minute sampling)	4 hours at 1 minute sampling	RTD: -100 to +100° C Thermocouple: -200 to +200° C		
Monnit	433, 868, 900 & 920 MHz (pricing depends on temperature range)	802.11 b/g (2.4GHz) Security: WEP, WPA & WPA2 (126€)	N	Optional: 3.0 - 3.6V DC with batteries for backup (39€)	Wireless: 2 x AA 1.5 V or 3V Coin cell WIFI(802.11): 2 x AA 1.5 V	Coin cell: 1 to 2 years at 1 hours sampling AA: 3 to 4 years at 1 hours sampling	Coin cell: -40°C (42€) AA: -50°C (109€) -200°C (160€)		
Phase IV	915 MHz (183€)	N	N	N	Y	5 years at 15 minutes sampling	-40°C to +125 °C		
VAISALA	N	Y	N	N	Y	Unknown	Y		

Figura 73: Características dos produtos - Gateways e sensores

Notifications						
PSTN						
Company	Local alert	Power outage	Internet outage	Mobile phone	Telephone Line	SMS
	Receives critical notifications from sensors. Notifications may be flashing of a LED, sound a audible alarm and/or display critical data.	In the event of device running only on batteries, the user will be notified	In the event of not having internet connection, the user will be notified	A call is made to a specific set of users	A call is made to a specific set of users	
Accsense	N	Y	Y	Y	Y	Y
Monnit	Y (296€)	N	N	Y	Y	Y
Phase IV	N	N	N	Y	N	Y
VAISALA	Y	N	N	Y	N	Y

Policy		User Interface			
Company	Scalability	Subscription	Cloud	WEB	PC
Accsense	N	Year	Y	Y	N
Monnit	Y/N	Year	Y	12 month subscription 6 sensors - 33€ 25 sensors - 84€ 50 sensors - 127€ 100 sensors - 170€ 500 sensors - 509€	
Phase IV	N	Month	Y	Y	N
VAISALA	Y	6-12 Months	Y	N	Y

Overall system/user configuration. Management of sensor data.

Figura 74: Características dos produtos - Notificações, política e interface de utilizador.

