



Integração de Redes de Sensores sem fio com Sistemas de Automção e Controlo

NELSON FERNANDO SILVA NOVAIS

Novembro de 2016

ISEP

Instituto Superior de Engenharia do Porto

Integração de Redes de Sensores sem fio com Sistemas de Automação e Controlo

Tese de Mestrado

Para obter o diploma de mestre no
Instituto Superior de Engenharia do Porto,
defendido publicamente por

Nelson Novais

Mestrado em Engenharia Eletrotécnica e de Computadores
Porto, Portugal.

Supervisor:

Prof. Dr. Mário Alves

Composição do Comité de Supervisão:

Prof. Dr.

Prof. Dr.

Prof. Dr.

Copyright © 2016 by EE

Todos os direitos reservados.

ISBN ++-++++-++++-

Email do autor: 1081724@isep.ipp.pt; nelsonfsnovais@gmail.com

Agradecimentos

Agradeço:

Aos meus Pais por todo o carinho e dedicação dado ao longo do meu percurso académico.

À minha irmã Sandra, pela ajuda e dedicação que me deu durante toda a minha vida.

Ao Eng. Francisco Sales, pelos conselhos dados ao longo do desenvolvimento da minha tese.

Ao Eng. Hélder Fernandes, pelas condições fornecidas dentro da empresa que permitiu o desenvolvimento da tese.

Ao Eng. Mario Alves por toda a sua paciência e conselhos dados ao longo do desenvolvimento da tese.

Por fim dedico esta tese à minha afilhada, Maria Inês, que tem o mundo todo pela frente.

Resumo

O presente documento vem apresentar o trabalho realizado no decorrer do desenvolvimento da tese de Mestrado de Engenharia Electrotécnica de Computadores com o tema ”*Integração de Redes de Sensores Sem fio com Sistemas de Automação e Controlo*”.

O objectivo principal prende-se com a demonstração da integração da tecnologia das redes de sensores sem fio com a automação. De forma a entender melhor os dois conceitos, ao longo da tese são abordados vários protocolos de comunicação, com e sem fios, sendo dada principal relevância ao protocolo MODBUS, ao qual se dedica integralmente um capítulo e onde se esclarece sua arquitectura.

O protocolo em questão foi determinante para o desenvolvimento do projecto que permitiu criar uma interface HMI (*Human Machine Interface*) que, suportado através de várias ferramentas que são abordadas no desenrolar da tese de mestrado, permitiu a integração entre as redes sem fio com a automação.

Numa fase inicial deste documento é abordado o contexto tecnológico da aplicabilidade deste tema, contexto este que despoletou a motivação para a sua criação, suportado por casos reais em que o projecto desenvolvido foi aplicado.

Dedica-se um capítulo ao estado da arte, referenciando-se várias tecnologias, algumas das quais integradas no presente documento, apresentando-se ainda um caso de estudo, devidamente referenciado, que serviu de exemplo.

Seguidamente, é abordada a arquitectura escolhida para dar corpo a todo o sistema pensado, explicando-se passo a passo todos os trâmites para a sua implementação.

Abstract

The following document hereby presents the work done during the development of the Master Thesis in Computer Electronic Engineering on the subject of "Integration of Wireless Sensor Networks with Automation and Control Systems".

The main objective focuses on demonstrating the integration of wireless sensor network technology with automation technology.

In order to better understand both concepts throughout the thesis several communication protocols are tackled, both wired and wireless. The main focus is given to the MODBUS protocol, to which a whole chapter is dedicated explaining its architecture.

The aforementioned protocol was key to the project's development as it allowed to create an HMI (Human Machine Interface) which, while being supported by the various tools mentioned during the thesis, enabled the integration between wireless networks and automation systems.

Initially, the subject's technological applicability is discussed. This was the main motivation for its creation while being supported by real use cases the project was later applied to.

A chapter is dedicated to the state of the art, mentioning several technologies, some of which are integrated in the present document. A properly referenced case study is also presented which was used as example.

Furthermore, the architecture chosen to shape the devised system is discussed, explaining step-by-step all the procedures for its implementation.

Conteúdo

1	Introdução	1
1.1	Motivação	2
1.1.1	Meio Hospitalar	3
1.1.2	Indústria	3
1.1.3	Benefícios	4
1.1.4	Integração com Automação	4
1.1.5	Plataforma integrada	4
1.2	Objectivo	5
1.3	Contribuições	6
1.3.1	Hospital Garcia da Orta	6
1.3.2	Hotel Sheraton	7
1.3.3	Validair Group	9
1.3.4	Resultados	10
1.4	Calendarização	11
1.5	Estrutura	11
2	Requisitos e Aplicação	13
2.1	Requisitos	13
2.2	Caderno de Encargos	14
2.3	Visão Geral da Arquitectura	15
3	Análise das Tecnologias	19
3.1	Redes de Sensores sem Fio	19
3.1.1	Protocolos de Comunicação sem Fio	21
3.1.2	Áreas de Aplicação de WSN	25
3.2	Automação Industrial	27
3.2.1	Controladores Lógicos Programáveis	28

3.2.2	Sistemas de Supervisão	29
3.2.3	Sistemas de Controlo de Edifícios	29
3.2.4	Redes de Campo	30
3.3	Plataformas Integradas	32
3.3.1	Computação na <i>Cloud</i>	33
3.3.2	<i>Software</i> como Serviço	34
3.3.3	Centralização de informação	35
3.4	Integração da Automatização de Edifícios com WSN	36
3.4.1	Automatização de Edifícios	36
3.4.2	Integração com WSN	36
3.4.3	Resultados	38
4	Descrição da Rede/Protocolo de Comunicação Base	39
4.1	História	39
4.2	Modelo de Comunicação	40
4.2.1	Descrição do Protocolo	41
4.2.2	Estrutura ADU	41
4.3	Registos MODBUS	42
4.4	MODBUS TCP/IP	44
4.5	Rede sem fios	45
4.6	Motivação	47
5	Arquitetura do Sistema	49
5.1	Arquitetura Proposta	49
5.2	Estrutura WSN	52
5.3	Estrutura Servidor - Cloud	54
5.4	Estrutura PLC	55
5.5	Raspberry Pi	57
6	Implementação e Validação	59
6.1	Fluxograma	59
6.2	Plataforma Web	62
6.3	Interface Raspberry - Fieldbuses	63
6.4	Validação	68
6.4.1	Testes - Comunicação	68

6.4.2	Testes - Automato	70
6.5	Apreciação	71
7	Conclusões	73
	Appendices	79
A	Anexo	81
A.1	Requisitos de Sistema	81
A.2	Instalação SO	82
A.3	Módulos Necessários	83
A.4	Ficheiros Necessários	84
A.5	Configuração V3-Shield Module - RTU	84

Lista de Figuras

1.1	Remote IO Moxa E1214, leitor de entradas Analógicas [1]	7
1.2	Analisador de Energia SmartRail[2]	8
1.3	Modulo de comunicação MODBUS RTU[3]	9
1.4	Sensores e Receptor Tektronix[4]	10
1.5	Calendarização de tarefas, de Março de 2015 a Junho de 2016	11
2.1	Visão geral da arquitectura a projectar	16
3.1	Gama de bandas e frequências da norma 802.15.4 [5]	21
3.2	Topologia mesh. [5]	23
3.3	Aplicação das WSN em florestas [6]	25
3.4	Aplicação das WSN na área da saúde [7]	26
3.5	Aplicação das WSN na área militar [8]	27
3.6	Imagem de vários PLC's [9]	28
3.7	Interface gráfica de um SCADA[10]	29
3.8	Pirâmide da Automação[11]	30
3.9	Arquitectura Cloud[12]	33
3.10	Modelo de Software como um Serviço[13]	34
4.1	Modelo OSI[14]	40
4.2	Camada de Aplicação MODBUS[15]	41
4.3	Estrutura da ADU[15]	41
4.4	Trama MODBUS TCP/IP[16]	44
4.5	Arquitetura da Comunicação MODBUS TCP/IP[16]	44
4.6	Comunicação entre Cliente e Servidor [16]	45
5.1	Arquitetura de Sistema	50
5.2	Receptor Carel [17]	52

LISTA DE FIGURAS

5.3	Sensor de Humidade, Temperatura e Luz Carel. [17]	53
5.4	Receptor Eltek [18]	53
5.5	Receptor Eltek [18]	54
5.6	Gateway - Noxa Nport 5110, [19]	54
5.7	Plataforma Vigie [21].	55
5.8	Modulo de Controlo.	55
5.9	PLC Carel.	56
5.10	Conversor MODBUS RTU / TCP.	56
5.11	Aspecto do Raspberry Pi II Modelo B.	57
6.1	Fluxograma representativo do funcionamento da arquitectura.	60
6.2	Cadeia de Eventos	61
6.3	Plataforma Adaptável	62
6.4	Aspecto da plataforma num computador/tablet	63
6.5	Parte do código de teste de comunicação em Python	63
6.6	Modelo da Base de Dados	65
6.7	Análise da comunicação através do MGate Manager	69
6.8	Resposta do Script Python	70
6.9	Resposta do PLC	70
6.10	Valores apresentados pela plataforma	71

Lista de Tabelas

3.1	Protocolos de comunicação FieldBus populares, comunicação série. [22][23]	31
3.2	Comparação de Fieldbus, Performance [24][25]	32
4.1	Protocolos de comunicação Ethernet/IP, FieldBus [15]	42
4.2	Tipo de Registos[15]	42
4.3	Funções MODBUS[15]	43
4.4	Especificações Técnicas [18] [26]	46
6.1	Tabela de Registos MODBUS	69

Acrónimos

ADU	Aplication Data Unit
API	Aplication Programming Interface
BAS	Building Automation Systems
CA	Collision Avoidance
CSMA	Carrier Sense Multiple Access
CO2	Dióxido de Carbono
Ethernet IP	Ethernet Industrial Protocol
FDA	Federal Drugs Administration
GHz	Giga Hertz
HMI	Human Machine Interface
IT	Information technology
I/O	Input/Output
IP	Internet Protocol
LSB	Least Significant Byte
MSB	Most Significant Byte
MAC	Mediun Access Control
WSN	Wireless Sensor Networks
OSI	Open Systems Interconnection
PLC	Programable Logic Controler
PDU	Protocol Data Unit
Profibus	Process Field Bus
QGBT	Quadro Geral de Baixa Tensão

LISTA DE TABELAS

SaaS	Software as a Service
SCADA	Supervisory Control and Data Aquisition
SO	Sistema Operativo
TCP	Transmission Control Protocol
RAM	Random Acess Memory
UPS	Uninterruptible Power Supply
XML	Extensible Markup Language
ZC	ZigBee Coordinator
ZED	ZigBee End Device
ZR	ZigBee Router

1

Introdução

Neste capítulo apresenta o tema da presente Tese bem como o seu enquadramento com o Mestrado de Engenharia Electrotécnica de Computadores. Assim, são identificados: os objectivos do tema, a sua motivação, contribuições, planeamento e, por último, uma breve abordagem à estrutura da Tese.

A utilização de redes de sensores sem fios (WSN, Wireless Sensor Networks) tem, nos últimos anos, tido uma grande ascensão. Estas consistem na distribuição de vários sensores por forma a criar uma vasta rede sem fios com o objectivo de monitorizar máquinas, sistemas e variáveis do meio ambiente (ex: temperatura, humidade, CO₂).

As WSN são utilizadas em vários contextos aplicativos, tais como hospitais, indústrias alimentares ou de processo, entre outros. Tipicamente, estas são utilizadas para monitorizar o meio através dos sensores/transmissores, que enviam a informação recolhida para uma base de dados, alojada num servidor local ou *cloud*. A informação recolhida pelos sensores é utilizada para gerar alertas, caso se detectem valores anómalos e também para apoio à decisão.

Através dos registos é possível comparar valores registados em períodos homólogos bem como perceber o efeito de medidas tomadas, correlacionando causa/efeito.

Esta tecnologia tem vindo a ser gradualmente utilizada nos vários sectores industriais, na medida em que, todos eles necessitam da introdução de novas tecnologias no seu meio, o que, na maior parte dos casos, pode provocar paragens em alguns serviços, traduzindo-se em avultadas perdas económicas, algo que com as WSN dificilmente acontece.

Como as WSN utilizam sensores sem fios, e sendo estes normalmente alimentados através de baterias internas, não são necessários cabos para ligação à rede de comunicação nem à rede eléctrica, o que faz com que a sua integração no meio industrial seja rápida e sem provocar grandes paragens no processo fabril, conseguindo-se assim uma redução custos de instalação associada à supressão da cablagem.

Com a potencialidade das WSN em mente surgiu a ideia para este trabalho de Tese: integrar as WSN em sistemas de automação e controlo já presente nos meios assinalados e, desta forma, conseguir aumentar a eficiência da gestão das variáveis do meio e/ou dos sistemas de afectos aos serviços/sectores de uma empresa/organização.

1.1 Motivação

Este tema surge no âmbito da actual função que o aluno ocupa, nomeadamente como Engenheiro de Suporte na empresa "ViGIE Solutions"[21], sediada no polo tecnológico da Maia.

Esta empresa dedica-se exclusivamente à monitorização e gestão de serviços industriais e hospitalares. Esta oferece os seus serviços através de plataformas integradas.

Durante a sua actividade, o aluno constatou, junto de vários clientes da empresa, da necessidade de integração na sua plataforma de um módulo de controlo e, assim, estender a capacidade da sua rede de monitorização ao controlo automatizado, interagindo directamente com controladores lógicos programáveis (PLC).

1.1.1 Meio Hospitalar

As redes de monitorização em ambiente hospitalar são utilizadas, essencialmente, para monitorizar temperaturas da rede de frio hospitalar (frigoríficos, arcas congeladoras com medicamentos, amostras e reagentes sensíveis a desvios de temperatura), consumos energéticos, centrais de tratamento e distribuição de água, de forma a monitorizar concentração de cloro na rede de abastecimento de água fria e a temperatura dos circuitos de água quente sanitária (AQS). Esta monitorização consegue, ao mesmo tempo, salvaguardar material de valor económico avultado, que se encontra na rede de frio hospitalar bem como, da mesma forma, detectar fugas e/ou uso indevido de electricidade e água (e tomar eventuais medidas correctivas). Gere ainda todo o tratamento de água, de forma a garantir que os ciclos de desinfectação, quer sejam químicos ou por elevação de temperatura, são correctamente efectuados, permitindo ainda correlacionar a eficiência desses mesmos ciclos com (eventuais) medidas tomadas.

1.1.2 Indústria

As redes de monitorização são fortemente utilizadas na indústria alimentar, de forma a garantir que os alimentos sejam conservados da melhor forma, com o intuito de assim atestar a sua qualidade. Tal como no ambiente hospitalar, sempre que existe desvio de temperaturas nas câmaras ou arcas frigoríficas, são gerados alarmes para que os responsáveis fiquem ocorrentes da situação e possam ser tomadas medidas para salvaguardar o produto.

Ainda no seio industrial, estas redes são também utilizadas para monitorizar e gerir consumos eléctricos, desde uma secção inteira de uma empresa até uma só máquina em específico, com o intuito de comparar consumos entre os diversos serviços, assim como detectar eventuais usos indevidos e/ou avarias nas máquinas eléctricas que despoletem consumos anormais para as mesmas. O facto de se conseguir aceder a um grande volume de informação no que respeita a consumos, permite que os gestores consigam também determinar quais os tarifários (luz, água, etc...) que mais se adequam às suas necessidades e assim conseguir diminuir a despesa eléctrica.

1.1.3 Benefícios

A utilização das WSN permite também aos seus utilizadores obterem dados, não só em tempo real, que lhes permite obterem previsões e, assim, aumentar o rendimento de certos sectores ou, em outros casos, melhorar a sua eficiência, ao mesmo tempo que salvaguardam a sua mesma organização.

Controlar a qualidade de ar dentro de um escritório pode parecer um mero exercício trivial mas que, com um olhar atento, pode levar a alterações profundas no *layout*, de forma a obter uma melhor qualidade interior (controlo de temperatura ambiente), o que possibilita um ambiente laboral mais confortável e que trará, como consequência, um aumento de rendimento dos trabalhadores.

1.1.4 Integração com Automação

A integração da automação com as WSN é um passo óbvio. Adicionar a flexibilidade de uma vasta gama de sensores *wireless* ao controlo permite, não só uma redução de custos em cabo (e na montagem), como também alcançar locais onde as soluções com fios não chegam, bem como acrescentar mais dados aos já recolhidos pelo próprio autómato.

Desta forma, aglomerando toda informação recolhida por ambos os sistemas, torna-se mais fácil correlacionar dados, que permitem averiguar desvios de leituras nos sensores similares, bem como alterar *set points* em função dos valores médios entre os sistemas, de modo corrigir eventuais erros de medição.

Um aspecto importante é o facto de o sensor *wireless* (ou os sensores) providenciarem leituras com maior granularidade, (como se comprova pelo estudo do Capítulo 3) e, assim, "obrigar" o autómato a tomar acções que, por ventura, não tomaria com os dados recolhidos pelas sondas.

Em suma a integração das WSN aos sistemas de automação, alargando o seu alcance, pode aumentar a sua eficiência e adicionar múltiplas funcionalidades.

1.1.5 Plataforma integrada

Uma plataforma integrada é geralmente um *software* de utilização simples, assente num portal web que agrega vários tipos de *hardware* com funcionalida-

des distintas, com o objectivo de facilitar o acesso à informação dos diferentes sectores de uma organização.

Um dos grandes problemas ao nível das organizações é a diversidade de aplicações, isto é, para cada sector/área existe uma aplicação que só funciona em certos dispositivos, dificultando o acesso a estes.

Na medida em que estas plataformas podem ser acedidas através de qualquer dispositivo que possua um *web browser*, torna-se mais fácil centralizar a informação, sendo que qualquer utilizador pode aceder à mesma e em qualquer lugar, seja em casa ou no trabalho. Munir estas plataformas de um módulo de controlo torna-se imperativo.

Veja-se o exemplo de um hospital que possui a monitorização da temperatura na sala de espera de urgências: a recepcionista, com acesso à plataforma, consegue vigiar a temperatura a que a sala se encontra, mas não consegue aceder à parte de controlo de ar condicionado e aquecimento (HVAC). No caso de sala se apresentar anormalmente fria ou quente, esta não consegue alterar essa situação, podendo apenas reportar o mesmo aos chefes de manutenção, que por sua vez, não podem, em alguns casos, alterar a programação do autómato que gere o HVAC, dado que este possui um algoritmo que gere a temperatura de forma sazonal.

Com a integração deste controlo nas plataformas integradas, torna-se possível que a recepcionista, desde que com permissões para tal, seja capaz de controlar a temperatura da sala.

1.2 Objectivo

O principal objectivo desta Tese é a integração de uma WSN com um sistema de automação já instalado. Esta integração é feita recorrendo à comunicação entre o servidor que gere a WSN com um PLC.

De forma a alcançar este objectivo, subdividiu-se o trabalho em múltiplas tarefas, tais como:

- Recolha de especificações funcionais;
- Pesquisa de tecnologias;
- Estudo de protocolos de comunicação, nomeadamente o protocolo MODBUS, para que seja possível comunicar com o sistema de automação;

- Desenho das especificações;
- Desenvolvimento de um protótipo;
- Desenvolvimento de uma interface gráfica e de um algoritmo para comunicar com o sistema de automação;
- Testes globais.

1.3 Contribuições

O projecto desenvolvido, dada a sua versatilidade, foi implementado num contexto comercial e encontra-se, neste momento, a ser utilizado por duas instituições: o Hospital Garcia da Orta e o Hotel Sheraton de Lisboa; e em fase de testes, por um parceiro tecnológico no Reino Unido, a *Validair Group* [27] tendo sido, em qualquer um dos casos, utilizado para funções diferentes. Os seguintes subpontos apresentam em maior detalhe as suas aplicações nos casos referidos.

1.3.1 Hospital Garcia da Orta

O Hospital da Garcia [28] da Orta já se encontrava a utilizar o sistema de monitorização contínua da "ViGIE", mais concretamente na monitorização da rede de frio, na qual a plataforma de modo contínuo analisa as condições em que os medicamentos armazenados estão sujeitos; e na emissão de alertas em caso de disparidades.

Neste contexto, foi pedido à empresa ViGIE que integrasse a central de alarmes, que contém informação sobre alarmes de incêndio, inundação e estado das UPS (*Uninterruptible Power Supply*).

Problema

Para se proceder à integração das centrais de alarmes do Hospital Garcia da Orta foi necessário criar um interface com a central de alares, uma vez que esta se encontrava bastante ultrapassada e não possuía qualquer modo de comunicação, somente através de saídas analógicas.

Solução

O primeiro passo foi, então, colocar um dispositivo com entradas analógicas, ver Figura 1.1, com capacidade de comunicação por MODBUS TCP. A este dispositivo foram ligadas todas as saídas da central de alarme.



Figura 1.1: *Remote IO Moxa E1214, leitor de entradas Analógicas [1]*

O segundo passo consistiu em dotar o servidor local, com o projecto desenvolvido durante a Tese a "varrer" todos os *Remote IO's* instalados e colocar os seus estados na plataforma ViGIE, conseguindo desta forma centralizar a informação das centrais de alarmes do Hospital numa única plataforma.

1.3.2 Hotel Sheraton

O Hotel Sheraton [29], localizado na baixa de Lisboa, recorreu à empresa ViGIE para que esta instalasse, nas suas instalações, analisadores de energia, ver Figura 1.2, de forma a ser possível controlar/analisar os seus consumos/gastos diários de forma continua.

Problema

Dado que a tecnologia utilizada pela ViGIE se centra na comunicação sem fios, foi detectado um problema com o alcance dos transmissores, uma vez que o sinal era bloqueado pelas paredes onde estavam instalados os QGBT (Quadro Geral de Baixa Tensão) e também, possivelmente, pelos campos magnéticos gerados por estes.



Figura 1.2: *Analizador de Energia SmartRail[2]*

Solução

A solução adoptada neste caso foi, assim, colocar pontos de rede comunicação Ethernet pelos QGBT's, e colocar, em cada, um Raspberry com um módulo V3-Shield, ver Figura 1.3, que permite a comunicação com dispositivos que comuniquem por MODBUS RTU, como é o caso dos analisadores de energia em cima representados na Figura 1.2.

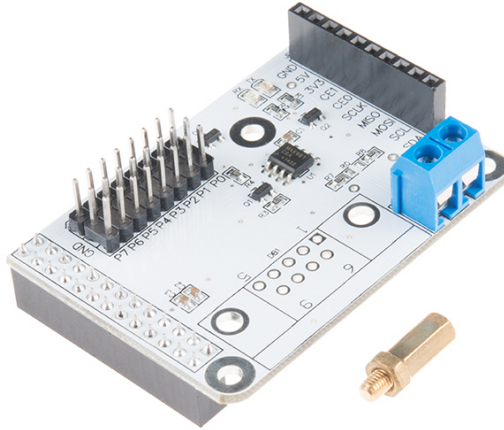


Figura 1.3: *Modulo de comunicação MODBUS RTU[3]*

Desta forma, foi possível aceder a todos os consumos em tempo real, através do Raspberry, e colocá-los na plataforma *web* da ViGIE.

1.3.3 Validair Group

A Validair [30] é um parceiro da ViGIE no Reino Unido que utiliza a plataforma ViGIE através de um *rebranding* da marca. A Validair destaca-se no mercado por monitorizar salas brancas, cumprindo de forma integral todos os requisitos obrigatórios pela FDA (*Federal Drugs Administration*) relativos à sua área de negócio.

No decorrer do ano de 2016 foi solicitado à ViGIE que integrasse no portfólio da mesma gama de sensores utilizados pela Validair, de forma a que esta pudesse tirar partido de toda a potencialidade que a plataforma ViGIE fornece, com o *hardware* já reconhecido e testado pelos seus clientes. Os sensores (ver Figura 1.4) em questão são fabricado pela Tektronix [31], companhia também ela sediada no Reino Unido.

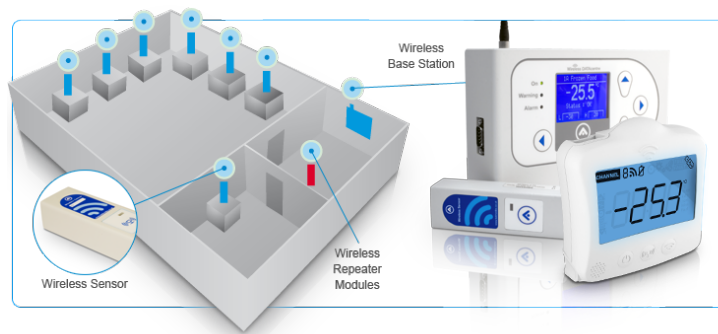


Figura 1.4: *Sensores e Receptor Tektronix*[4]

Problema

A Figura 1.4 representa uma solução completa de monitorização através de uma WSN, em que os sensores comunicam por um protocolo proprietário com uma base, no qual ficam armazenados todos os valores lidos pelos sensores. Esta base possui um servidor *web* próprio no qual se pode aceder às leituras e programar alarmes. No entanto, é bastante limitado em termos de tratamento de dados de forma estatística e ao nível de interface gráfico. Para resolver esta situação, foi então solicitado à ViGIE que integrasse esta gama na sua plataforma de monitorização.

Solução

Apesar da gama de sensores referida comunicar com a base através de um protocolo proprietário ao qual não se teve acesso, o fabricante, para efeitos legais, incorporou na mesma um módulo de comunicação MODBUS TCP, o qual permite aceder aos valores dos sensores em tempo real e aos valores previamente registados pela base. Mais uma vez, para se efectuar a integração, foi utilizado o trabalho desenvolvido durante a presente Tese, que permitiu a integração de mais uma solução no portfólio da empresa.

1.3.4 Resultados

Os resultados das implementações referidas foram bastante satisfatórios, apesar de não estarem a tirar total partido do projecto desenvolvido, uma vez que apenas se está a ler valores dos dispositivos sem nestes ter qualquer tipo de

actuação. Contudo, o sistema funciona 24/7 de forma contínua e cumprindo os seus pressupostos.

1.4 Calendarização

Sendo a comunicação com os sistema de automação e controlo a motivação para o tema deste trabalho, a sua prossecução levou à criação do gráfico de *Gantt*, Figura 1.5, que apresenta a calendarização das tarefas enunciadas.

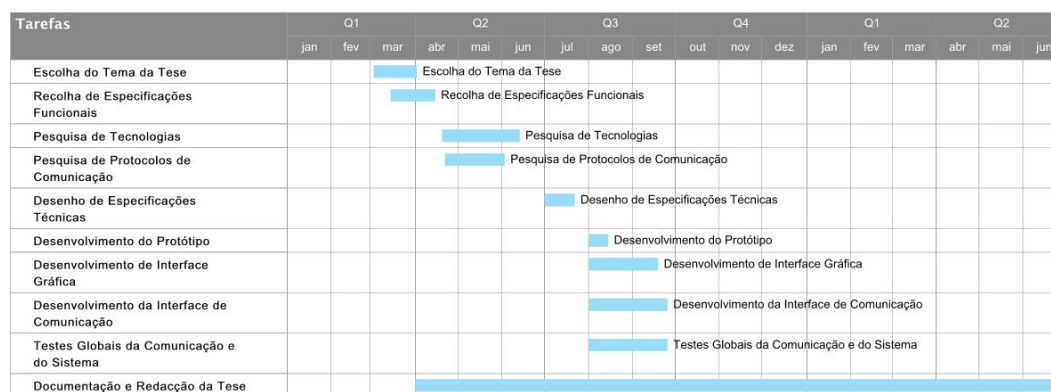


Figura 1.5: Calendarização de tarefas, de Março de 2015 a Junho de 2016

1.5 Estrutura

A presente secção visa a apresentação da estrutura da presente tese.

O relatório encontra-se organizado por capítulos, sendo que em cada capítulo serão abordados os temas que visam à compreensão da temática presente. Desta forma o Capítulo 1 enquadra o tema da Tese com o Mestrado em Engenharia Electrotécnica de Computadores. O Capítulo 2 apresenta a visão geral da Tese e da sua arquitectura bem como o caderno de encargos necessário à sua implementação nos contextos abordados no Capítulo 1. No Capítulo 3 analisa-se e contextualiza-se as tecnologias abordadas durante o desenvolvimento da Tese. O Capítulo 4 aborda em maior detalhe protocolo de comunicação escolhido para se realizar a interface com o sistema de automação. O Capítulo 5 apresenta arquitectura construída para se interligar o sistema de automação com uma WSN. O Capítulo 6 apresenta a sua implementação. Por fim, o

CAPÍTULO 1. INTRODUÇÃO

ultimo Capítulo engloba as principais conclusões retiradas pelo o aluno no término da Tese de mestrado.

2

Requisitos e Aplicação

O seguinte capítulo tem o propósito de explicar os requisitos em termos de *hardware* e *software* que aplicação desenvolvida requer para poder ser implementada nas áreas referidas no Capítulo 1.

Apresenta-se o caderno de encargos referente ao processo de desenvolvimento da aplicação.

2.1 Requisitos

O objectivo principal desta aplicação é realizar a integração de uma rede de sensores com um PLC, apresentando esta integração numa plataforma *web*. Como se pretende que aplicação em causa seja aplicada em ambientes industriais e hospitalares, onde o controlo climático é necessário, é importante que esta consiga alterar os *setpoints* do controlador HVAC, mediante dos resultados das leituras da rede de sensores, tornando o controlo climático mais eficiente. Contudo, dada a natureza do meio onde se vai aplicar o sistema, é necessário satisfazer os seguintes pontos:

Longevidade Como a rede sensores sem fio não possui alimentação externa,

ou não é expectável que o tenha, e de forma a que os custos de manutenção sejam baixos, é necessário que o tempo de vida das baterias dos sensores tenham a duração mínima de 5 anos.

Configuração Flexível A arquitetura do sistema deverá ser o mais flexível possível por forma acomodar todo o tipo de cenários. Deverá permitir vários tipos de topologias, vários tipos de sensores e actuadores, bem como ser compatível com vários protocolos de comunicação para assegurar a interoperabilidade entre sistemas.

Comunicação em tempo Real Como objectivo é controlar os actuadores de um automato, é necessário que a comunicação entre a plataforma e o automato seja efectuada em tempo real, para que esta realize o pretendido em tempo real. Perante este objectivo é necessário dotar aplicação de um modulo capaz de comunicar com os autómatos através de um protocolo *standart* de comunicação.

Amostragem Os tempos de recolha de dados não deverão ser inferiores a 5 minutos.

Interoperabilidade A plataforma deverá ser capaz de englobar a informação de redes de sensores sem fio já implementadas, ou a ser implementadas, com a toda a informação proveniente do(ou dos) automato(s).

Segurança De forma a evitar ataques maliciosos aos sistema de controlo, é necessário que aplicação possua um modo de autenticação de utilizador, de forma evitar utilizadores mal intencionados.

2.2 Caderno de Encargos

O presente caderno de encargos compreende, apenas, os requisitos técnicos necessários à implementação de uma plataforma *web* para a monitorização e controlo.

Requisitos técnicos

A solução a fornecer deverá consistir nos seguintes componentes:

Software

1. Licença de software sem limite de utilizadores;
2. Possibilidade de controlo remoto da aplicação;
3. Acesso aos parâmetros detalhados de cada parâmetro incluindo valores actuais, médias diárias e gráficos em tempo real;
4. Definição e parametrização de aletas por parâmetro/setor;
5. Tratamento estatístico de dados;
6. Gestão de utilizadores com níveis de acesso diferenciados e definição do tipo de acções permitidas; a cada utilizador;
7. Possibilidade definir tempo de amostragem (*Duty Cycle*);
8. Controlo de actuadores;
9. Controlo de Sistemas de Automação e Controlo.

Hardware

1. Servidor aplicacional necessário para a instalação e execução do software referido no ponto anterior;
2. O Hardware deverá ser dimensionado para as futura necessidades de modo a comportar novos pontos de monitorização que possam ser integrados futuramente;
3. Modulo de comunicação que permita a interoperabilidade entre sistemas.

2.3 Visão Geral da Arquitectura

A estrutura geral da arquitectura proposta no tema da Tese deve respeitar as tecnologias e estruturas existentes, isto é, deverá integrar não só os sistemas de automação e controlo existentes bem como utilizar a tecnologia ao nível das WSN já instaladas nos locais.

O sistema projectado deverá permitir aceder às leituras dos vários periféricos que formam os sistemas de automação e controlo, e integrá-los com a WSN, e ser apresentados numa plataforma móvel, tal como apresentado na Figura 2.1.

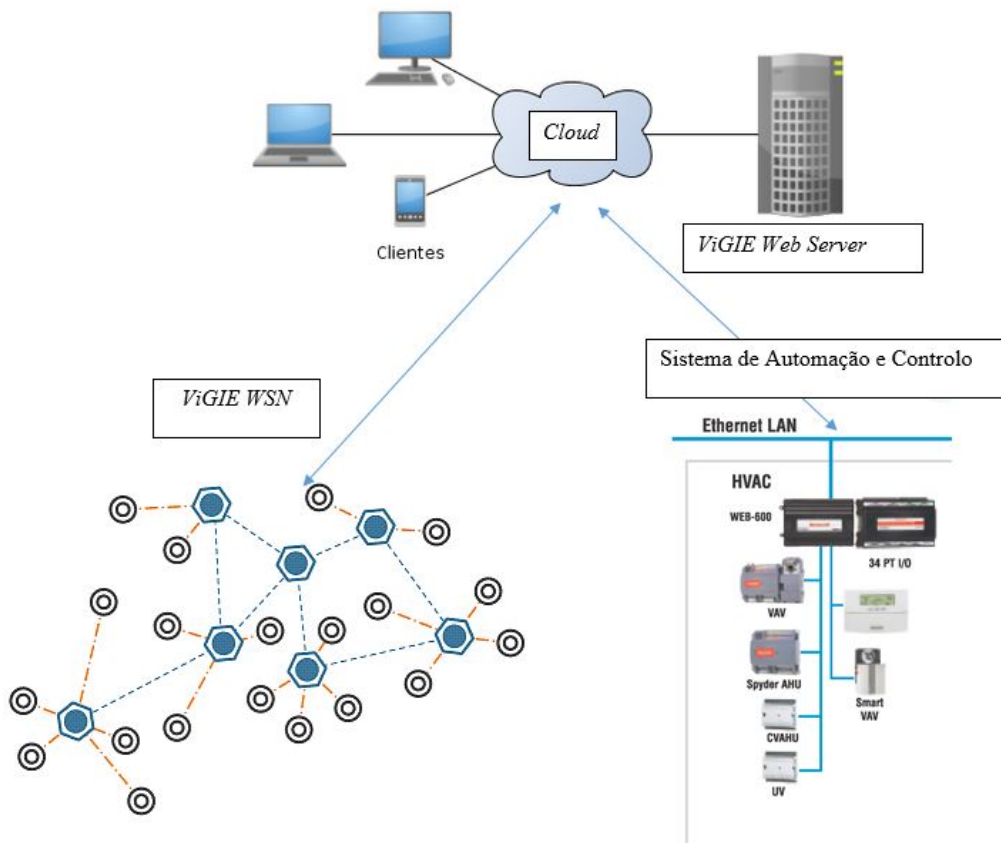


Figura 2.1: Visão geral da arquitectura a projectar

Como se depreende pela Figura 2.1, o projecto deverá permitir ao utilizador consultar os dados da sua rede WSN (fornecida pela empresa ViGIE) e na mesma plataforma integrar o seu sistema de automação, tirando partido de uma centralização de informação, evitando a dispersão de informação. Desta forma, a arquitectura que se projecta deverá permitir a introdução de novos equipamentos e ao mesmo tempo ser retrocompatível com as estruturas já existentes. Desta forma deverá ter em conta as seguintes referencias ao nível do

Hardware:

1. Sensores Eltek;
2. Conversores RS232-Ethernet Moxa;
3. Conversores RS485-MODBUS Moxa;
4. PLC Carel;
5. Sensores Carel;
6. Servidor ViGIE;

Cumprindo-se assim o pressuposto no caderno de encargos para as áreas em que se pretende aplicar a Tese.

3

Análise das Tecnologias

Este capítulo tem como intuito apresentar em detalhe as tecnologias que compõem o tema da tese.

A primeira secção é dedicada às WSN, onde se aborda a sua arquitectura e modo de funcionamento.

A secção seguinte faz uma pequena apresentação sobre a automação e uma descrição do meio de comunicação utilizado na área da automação. Por fim, e de forma a ligar a área da automação e da WSN, surge o tema das plataformas integradas, que vem no seguimento do projecto da tese e que tem como objectivo, não só integrar as WSN com a automação, mas também centralizar toda informação oriunda de autómatos numa só plataforma.

3.1 Redes de Sensores sem Fio

Tradicionalmente, os sensores existentes no mercado utilizavam a comunicação por fio. Contudo, o aparecimento de sensores que utilizam a comunicação sem fio, veio revolucionar a forma de comunicação e o tipo de redes possíveis de criar, abrindo caminho para novas aplicações[32].

Com um tamanho reduzido, baixa capacidade de processamento, baixo consumo, alcance reduzido e baterias de dimensão reduzidas, estes sensores conseguem medir diversas variáveis, desde: distância, direcção, velocidade, temperatura, humidade, CO₂, luz, vibrações, pressão diferencial, informação sísmográfica, entre muitas outras variáveis[32].

Estes sensores, numa WSN, são vistos como nós independentes com poder de medição, processamento e com capacidade de comunicação. De forma geral, as WSN são redes com uma grande densidade de nós, dispostos de forma aleatória ou específica, numa área (local) de estudo.

Uma vez colocados, espera-se que estes sensores comuniquem de forma autónoma e que formem uma rede operacional com vários tipos de topologias. Como estes sensores possuem baterias de duração limitada, e para assegurar a longevidade das mesmas, utilizam-se protocolos de comunicação desenhados para maximizar a vida das baterias, implementado mecanismos que reduzem o consumo energético[33].

Sensor e Actuador

Os sensores-actuadores podem surgir como complemento aos sensores tradicionais, fornecendo a capacidade de actuação para além da capacidade de medição das variáveis da área em que estão inseridos. Com este tipo de sensores é possível formar uma rede de sensores actuadores (WSNA, *Wireless Sensor Network Actuator*), que se espera que actue autonomamente e que ao mesmo tempo esteja ligada directamente ao utilizador por meio de uma interface que possibilite ao utilizador final actuar.

Devido às suas capacidades de actuação, os sensores actuadores dispõem de um poder de processamento maior que os tradicionais, dada a necessidade das tarefas dos sensores-actuadores ocorrerem em tempo real, o que implica uma maior durabilidade das baterias utilizadas. Com baterias mais potentes e com maior durabilidade, torna-se também possível aumentar o alcance da transmissão, o que implica também uma menor densidade da rede criada destes sensores relativamente a uma rede de sensores tradicionais.

3.1.1 Protocolos de Comunicação sem Fio

A pensar nas limitações ao nível da bateria dos sensores foram desenvolvidos protocolos de comunicação com o objectivo principal de reduzir o consumo de energia.

Assim, neste ponto apresentam-se os protocolos de comunicação de baixo consumo, tais como o *Bluetooth Low Energy*, ANT+, ISA100.11a e outros que correm sob a norma IEEE 802.15.4[33], como: ZigBee, *WirelessHART* e 6LoWPAN. Existem outros protocolos para além dos enunciados, no entanto, escolheram-se aqueles que foram abordados em ambiente académico e profissional pelo aluno.

IEEE 802.15.4

A norma IEEE 802.15.4 é o protocolo *standard* proposto para redes *wireless* com uma baixa transmissão de dados e oferece três bandas de frequência operacional de: 2.4 GHz, 915 MHz e 868 MHz [5] [33], ver figura 3.1. Existe um único canal entre os 868 e 868.6 MHz (20 kbit/s), 10 canais entre os 902 e os 928 MHz (40 kbit/s), e 16 canais entre os 2.4 e 2.4835 GHz (250 kbits/s) [5]. Esta norma foca-se, principalmente, no baixo consumo de energia e na baixa complexidade de comunicação. Foi desenhada a pensar nas aplicações de redes de sensores sem fios, que não requerem um grande alcance por parte do transmissor por forma a maximizar a vida da bateria. Este protocolo permite a "criação" de redes de sensores do tipo estrela, em que todos os sensores comunicam com um dispositivo central e redes de sensor *peer-to-peer*, em que os sensores funcionam de forma *ad hoc* e/ou de forma auto configurável [33].

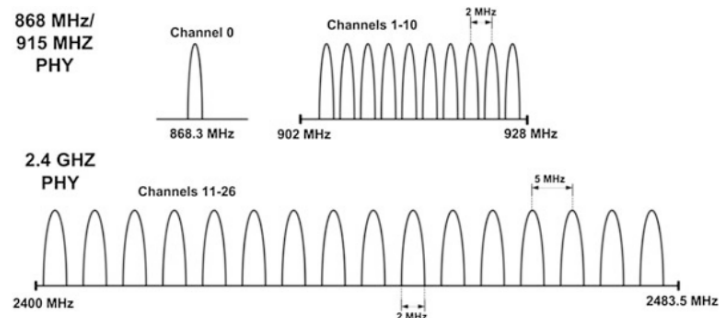


Figura 3.1: Gama de bandas e frequências da norma 802.15.4 [5]

Controlo de Acesso ao Meio

A norma IEEE 802.15.4 possui um protocolo controlo de acesso ao meio (MAC), que suporta 2 modos de operação:

Modo sem "sinalização" Quando selecionado este modo, o coordenador ZigBee selecciona o modo de operação "*non-beacon*" e, desta forma, não há envios de *beacons*¹, nem de *superframes*².

O acesso ao meio é feito através de um mecanismo de detecção da onda portadora com prevenção de colisão, *Carrier sense multiple access with collision avoidance*, CSMA/CA, sem intervalo de tempo definido para a transmissão[5].

Modo com "sinalização" Neste modo, são enviados periodicamente *beacons* pelo coordenador ZigBee, de forma a sincronizar os nós que lhe estão associados ao mesmo tempo que identifica a sua rede. O acesso ao meio é também ele feito através de um mecanismo CSMA/CA, mas com intervalos de transmissão definidos [5].

ZigBee

O ZigBee é assente na norma IEEE 802.15.4, está disponível publicamente desde 5 de Junho de 2005[5] e é definido como uma camada superior de comunicação [33]. É utilizado em redes de baixa transmissão e permite a utilização de tecnologia *wireless* simples, barata e de baixo consumo. Os dispositivos que utilizem este tipo de protocolo de comunicação conseguem formar redes de topologia *mesh*³, Figura 3.2, que permite a ligação de centenas de dispositivos entre si. Como referido anteriormente, os dispositivos que utilizam o protocolo de comunicação ZigBee consomem pouca energia e conseguem operar durante anos dependendo apenas de uma bateria de dimensões reduzidas. Existem três tipos de dispositivos ZigBee: ZigBee *coordinator* (ZC), ZigBee *router* (ZR) e

¹São frames que contêm informação acerca da rede e são enviados de forma periódica para "anunciar" a presença de uma rede *wireless*

²Uma *superframe* é definida entre dois frames de *beacons* e tem um período ativo e outro inativo.

³Rede em que cada nó pode operar como transmissor e router, isto é, cada nó pode criar uma sub-rede que, no computo geral, forma uma única rede, sendo possível enviar informação de um nó para outro por caminhos diferentes. [5]

o ZigBee *end device* (ZED). O ZC, actua como um coordenador de rede, configura a topologia da rede, guarda informação e produz as ligações entre redes diferentes; o ZR liga os dispositivos entre eles formando uma rede; o ZED consiste nos sensores, actuadores e controladores que recolhem a informação e que a comunicam com o ZC ou ZR.

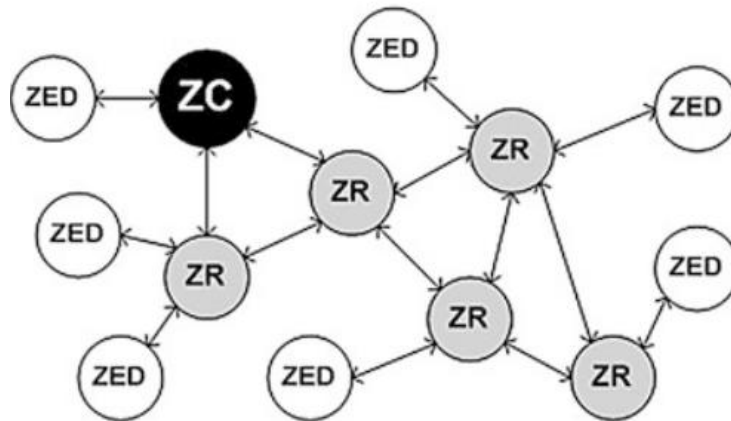


Figura 3.2: *Topologia mesh.* [5]

WirelessHART

O protocolo de comunicação WirelessHART [34] possibilita a criação de redes *wireless* para aplicações de controlo e de medição. Tal como o ZigBee, corre sobre a norma IEEE 802.15.4 [33]. Este protocolo é confiável, seguro e eficiente em termos energéticos, foi desenhado para suportar redes de topologia *mesh*, estrela (Figura 3.2), repetição do sinal e envio de mensagens de forma sincronizada. A segurança e a robustez deste protocolo é assegurada pela encriptação da comunicação, gestão de chaves de rede, pela validação da informação recebida bem como a autenticação da mesma.

Este protocolo permite ainda a gestão de consumo energético dos dispositivos presentes na rede, o que faz com que aumente a sua eficiência energética. As redes criadas por este protocolo são composta por *gateways*, sensores, actuadores e controladores, aplicações de supervisão e ferramentas de gestão de rede.

ISA100.11a

O protocolo que se segue é utilizado em redes de baixa transmissão de dados e foi desenhado para operar em aplicações de monitorização em automação [33]. Foca-se no baixo consumo, escalabilidade, robustez e na interoperabilidade com outros dispositivos de rede. A comunicação é feita através da frequência 2.4 GHz e utiliza repetidores de sinal para aumentar o alcance e diminuir a interferência.

6LoWPAN

Protocolo bastante inovador, uma vez que permite comunicar directamente com dispositivos IP, na medida em que utiliza um protocolo baseado no IPv6⁴ [35], permitindo, assim, fornecer aos dispositivos de baixo consumo as vantagens de uma comunicação por IP, tornando-o bastante flexível.

Bluetooth Low Energy

As primeiras versões do *Bluetooth* não foram criadas a pensar em redes de comunicação de baixo consumo e, portanto, impossibilitavam a utilização desta tecnologia em WSN. No entanto, nos últimos anos foram desenvolvidas novas versões deste protocolo, que culminaram na versão *Bluetooth LE* [36].

Esta versão permite alterar o seu *Duty Cycle*⁵ e, em paralelo com outras estratégias para a redução de consumo, levou a que esta tecnologia possa ser agora implementada em WSN [37].

ANT+

O protocolo ANT é um protocolo proprietário e, tal como os protocolos já enunciados, foi desenvolvido a pensar na redução de consumo de energia. Trabalha na banda dos 2.4 GHz e, do mesmo modo que o ZigBee, suporta diferentes tipos de topologia de rede, como *mesh*, estrela e em árvore.

Este protocolo foi desenvolvido a pensar em aplicações desportivas e de monitorização de sinais vitais (ex: pulsação) [38][39].

⁴Versão mais actual do Internet Protocol

⁵Período em que está activo

3.1.2 Áreas de Aplicação de WSN

As aplicações possíveis de se realizar com as WSN são tão vastas que é possível introduzi-las virtualmente em qualquer área de estudo. Estas estão presentes em áreas como o controlo ambiental, saúde, agricultura, industria e até militar.

Controlo Ambiental

Ao nível do controlo ambiental, as WSN vieram flexibilizar a monitorização de variáveis como temperatura, humidade, CO₂, velocidade do ar, fumo, pressão atmosférica, pressão diferencial, entre outras. Como os sensores em causa são, assim como referido, de pequenas dimensões, o meio de comunicação é sem fio, tornando-os bastante uteis na monitorização de determinados ambientes, como florestas (Figura 3.3), locais de quarentena, criopreservadores, pelo facto de não serem tão evasivos como a presença humana e também pelas áreas/locais que conseguem alcançar, uma vez que não estão limitados pela cablagem.

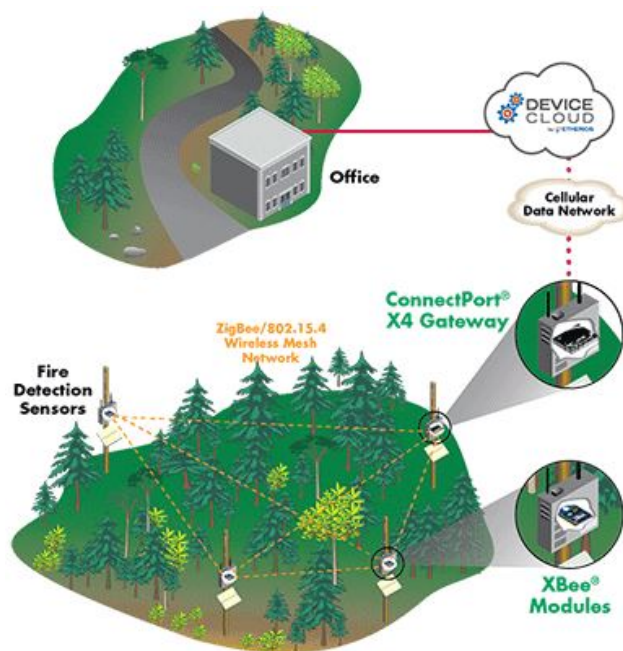


Figura 3.3: Aplicação das WSN em florestas [6]

Agricultura

Tal como no controlo ambiental, as WSN são uma mais-valia na agricultura, já que, recorrendo a informações que os sensores recolhem, os agricultores conseguem detectar se estão a ser criadas todas as condições para o aparecimento de pragas e, em tempo útil, actuar de forma preventiva contra as mesmas.

A aplicação das WSN, na agricultura, não se resume apenas à detecção de condições favoráveis a pragas, permite também aos agricultores monitorizar as condições do solo, tais como a temperatura, podendo garantir as melhores condições às plantações e, assim, aumentar a produção e a qualidade do produto final.

Saúde

Na saúde, as WSN são utilizadas em várias áreas, desde a monitorização de sinais vitais, ver Figura 3.4 dos pacientes à monitorização de temperatura de criopreservadores, nos quais, pequenos desvio na temperatura do azoto poderão significar uma redução na taxa de fertilização. Esta tecnologia é também utilizada nas salas brancas⁶, na monitorização das partículas em suspensão e no diferencial de pressão entre salas limpas.

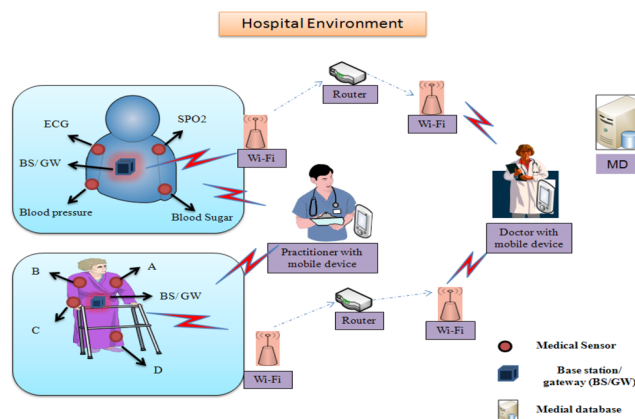


Figura 3.4: Aplicação das WSN na área da saúde [7]

⁶Salas em que o ambiente é controlado de forma a garantir que a interferência/contaminação por partículas em suspensão seja mínima.

Industria

Tal como nos sectores anteriores, as WSN na indústria são utilizadas para medir condições de temperatura de produtos, de forma a assegurar a sua qualidade perante identidades estatais, monitorizar o desgaste de maquinaria, qualidade de ar interior, consumo de recursos como energia eléctrica e água.

Em suma, as WSN, através da sua capacidade de medição e flexibilidade, podem ser utilizadas em vários meios com fins diferentes.

Aplicações Militares

As WSN na área militar podem ser utilizadas para vários fins: detecção de ataques químicos, intrusos [8], Figura 3.5, delineação de perímetros de segurança, monitorização do aprovisionamento do material militar (munições, bombas, etc).

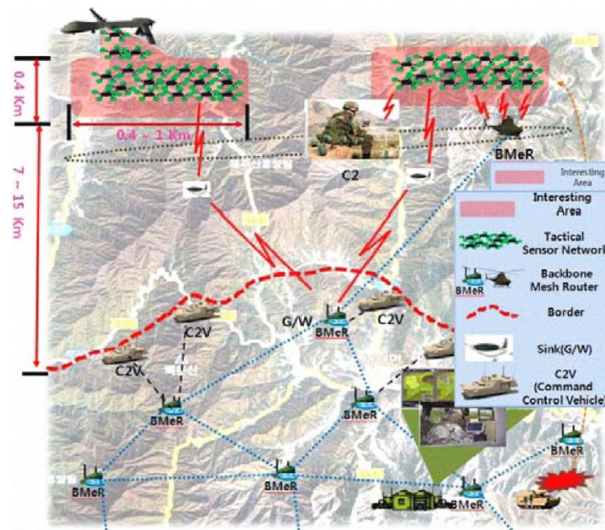


Figura 3.5: Aplicação das WSN na área militar [8]

3.2 Automação Industrial

Controlo automático ou automação é o termo utilizado para descrever uma variedade de sistemas que controlam vários tipo de dispositivos, desde maquinaria pesada na industria de processo até ao controlo de climatização em edifícios, sem intervenção humana directa.

Um dos mais simples mecanismos de controlo é o vulgarmente conhecido "ON/OFF", que consiste num comando digital ou analógico que permite ligar ou desligar qualquer dispositivo. No entanto, o controlo automatizado permite muito mais que um simples ligar/desligar, podendo, por exemplo, controlar toda uma instalação industrial, assim como o fabrico por completo de um carro, a regulação de luz e temperatura ou o controlo de acessos do pessoal afecto às instalações. Mas, para que este controlo seja possível, é necessário que os dispositivos de campo, como sensores, actuadores, válvulas, máquinas, comuniquem com um PLC (pode ser um ou vários), para que este, mediante da parametrização previamente programada, actue consoante os *inputs* recebidos.

3.2.1 Controladores Lógicos Programáveis

Os controladores lógicos programáveis, vulgarmente conhecidos como PLC, Figura 3.6, consistem num processador lógico que, mediante uma programação prévia, executa funções perante os valores fornecidos pelos actuadores e sensores que compõem o sistema de controlo. Estes sistemas de controlo podem ser simples, tendo apenas um PLC a cargo de toda a operação de controlo, ou sistemas de complexos de vários PLC's interligados entre si, geridos por um PLC de maior capacidade de processamento.

Os PLC's podem ainda executar funções perante acções humanas, através de uma interface, *Human Machine Interface* (HMI), que pode ser um painel físico ou um software de controlo presente num computador nas instalações.



Figura 3.6: Imagem de vários PLC's [9]

3.2.2 Sistemas de Supervisão

Para os sistemas mais complexos de controlo, de forma geral, utiliza-se sistemas de supervisão e aquisição de dados SCADA (*Supervisory Control and Data Acquisition*). Estes sistemas são utilizados para monitorizar e controlar processos e encontram-se fortemente implantados em industriais de processo, na distribuição de água, gás e de energia e até em hospitais.

Através dos sistema SCADA, ver Figura 3.7, é possível aceder a informação dos valores lidos pelos sensores e pelo estado dos actuadores, em tempo real, bem como efectuar acções de controlo, abrir e fechar uma válvula, ou aumentar/diminuir a cadência de uma determinada linha de produção.

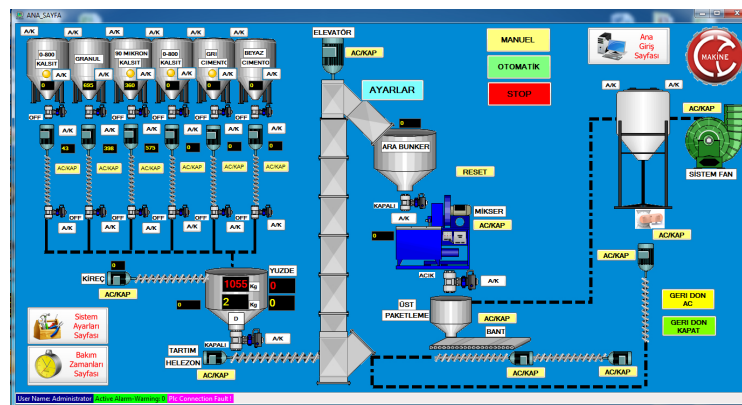


Figura 3.7: Interface gráfica de um SCADA[10]

3.2.3 Sistemas de Controlo de Edifícios

Os sistemas de controlo *Heating Ventilating Air Conditioning* (HVAC), que controlam o ar condicionado, calor e ventilação de um edifício, são um exemplo de sistema de controlo distribuído, também conhecido por *Building Automation Systems* (BAS).

O controlo climático de um edifício tem-se tornado fundamental para as empresas, não só em termos de conforto, mas também em eficiência energética. Automatizando os edifícios, é possível controlar a temperatura de varias áreas afectas ao edifício, bem como intensidade da luz e controlo de acessos. Tradicionalmente, o BAS, tal como na automação em geral, é um sistema que requer grande cablagem para interligar todos os componentes e, num edifício de vários

andares, isso representa grandes custos a nível de cablagem mas também na reconfiguração do próprio sistema quando é necessário efectuar mudanças[40].

3.2.4 Redes de Campo

As redes de campo (*Fieldbus*) são o meio de comunicação que interliga vários dispositivos de diferentes níveis, Figura 3.8, que compõe a pirâmide da automação[22].

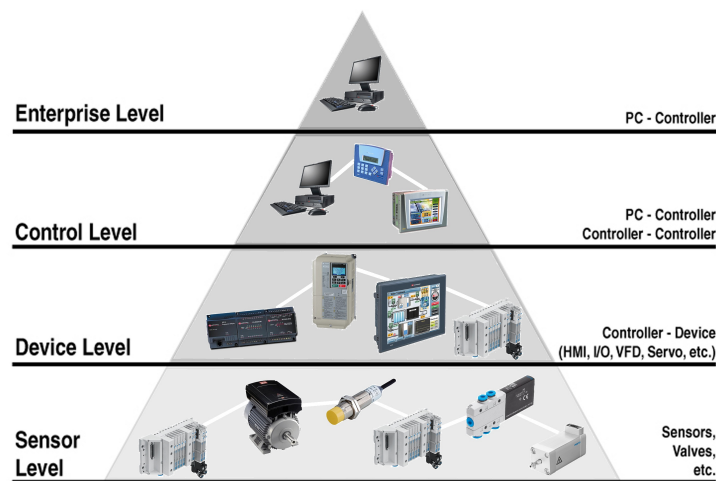


Figura 3.8: Pirâmide da Automação[11]

Este tipo de rede começou a ser projectado em 1982, em França [22], com o apoio do Ministério da Indústria e levou ao aparecimento, ao longo dos anos, de vários protocolos de comunicação, como o *Process Field Bus* (Profibus) para a indústria de processo [41] e o *Controller Area Network* (CAN)[42]. A tabela 3.1 apresenta alguns dos protocolos de comunicação em uso.

Modelo Normalizado

Com o desenvolvimento de vários protocolos *Fieldbus*, criou-se uma necessidade de criar um protocolo standard, de forma a criar uma maior interoperabilidade entre sistemas. Durante vários anos, assistiu-se a guerras entre fabricantes até que, em 1999, se chegou a um acordo e criou-se a norma IEC 61158[43]. Esta baseia-se numa simples estrutura de todos os protocolos utilizados que resulta numa única estrutura.

Tabela 3.1: *Protocolos de comunicação FieldBus populares, comunicação série. [22][23]*

Protocolo Fieldbus	Ano de Introdução
MODBUS RTU	1979
Profibus	1982
LonWorks	1991
DeviceNet	1994
CANOpen	1995
P-NET	1996
WorldFIP	1996
Foundation Fieldbus	2000

Ethernet na Automação

As redes de comunicação *Ethernet* já estão estabelecidas no meio empresarial há bastantes anos, ligando todo tipo de dispositivos presentes ao nível de gestão, tais como: computadores, impressoras, telefones e faxes.

Colocar esta tecnologia ao serviço da automação afigura-se de bastante interesse, uma vez que esta se encontra implementada numa grande parte da estrutura de qualquer empresa e transportá-la para os níveis mais "baixos" da pirâmide da automação é relativamente barato.

Porém, a necessidade de trazer a *Ethernet* para a indústria de processo não se resume apenas ao seu preço de implementação, mas sim à necessidade crescente de ter redes de comunicação com maior largura de banda para suportar o crescente aumento de informação que circula neste tipo de redes.

A utilização da *Ethernet* na indústria culminou com o aparecimento de um novo modelo de comunicação, *Ethernet/IP* [22] (*Ethernet Industrial Protocol*) e com este surgindo novas adaptações de protocolos não proprietários para utilizar as potencialidades da *Ethernet*.

Comparação entre Fieldbus

Os protocolos em questão possuem vantagens e desvantagens que importa ter em mente quando se pretende criar uma rede de comunicação. A Tabela 3.2, ilustra algumas das vantagens/desvantagens a ter em conta.

Em suma, todos os protocolos presentes na Tabela 3.2 apresentam-se como

soluções bastante viáveis, dependendo do intuito da aplicação e do hardware disponível, dado que cada fabricante escolhe a comunicação que o seu PLC suporta.

Tabela 3.2: *Comparação de Fieldbus, Performance [24][25]*

Performace	MODBUS/TCP	Ethernet/IP	Profinet IRT	EtherCAT
Vel. Comunicação	9.6 - 115.2 kBit/s	10 - 100 MBit/s	100-1000MBit/s	+100MBit/s
Complexidade	Simples	Avançada	Avançada	Avançada
Hardware Adicional	Não	Não	Sim	Não
Custos	Baixos	Baixos	Altos	Baixos
Flexível	Sim	Sim	Sim	Sim
Topologias	E A L	B E AN	E A L	E A AN L
Segurança	Baixa	Boa	Boa	Boa

E - Estrela; L - Linha; A - Árvore; AN - Anel

De notar que este protocolo este se apresenta como uma solução bastante barata, uma vez que não necessita de *hardware* adicional e, muitos fabricantes, apesar de terem os seus próprios protocolos de comunicação, como é o caso da Siemens (Profibus), acrescentam o protocolo MODBUS aos seus produtos, o que faz com que este protocolo providencie uma interoperabilidade maior que os demais.

Outra vantagem da adoção do protocolo MODBUS é a sua baixa complexidade, que facilita o desenvolvimento de aplicações e a sua própria implementação.

Um senão deste protocolo é a sua insegurança ao nível de ataques informáticos, uma vez que não tem meios de autenticação implementados, tornando-o bastante vulnerável.

3.3 Plataformas Integradas

Uma plataforma integrada é um *software* que é adquirido como um serviço (SaaS, *Software as a Service*) que é acessível a partir da Internet e permite agregar e consultar vários tipos de informação, tais como:

1. Monitorização;

2. Segurança;
3. Informação Administrativa;
4. Gestão.

Com a generalização dos serviços *cloud*, este tipo de plataformas têm cada vez mais procura e, segundo apontam os mercados [44], é uma área de negocio com perspectiva de crescimento nos próximos anos.

3.3.1 Computação na *Cloud*

O termo computação na *cloud* surge como definição para uma infraestrutura composta por vários computadores interligados entre si [45].

A *cloud* funciona de forma muito simples. Nos seus servidores aloja informação e aplicações que são acedidas pelos seus clientes/utilizadores através de múltiplas plataformas via *web*, ver Figura 3.9.

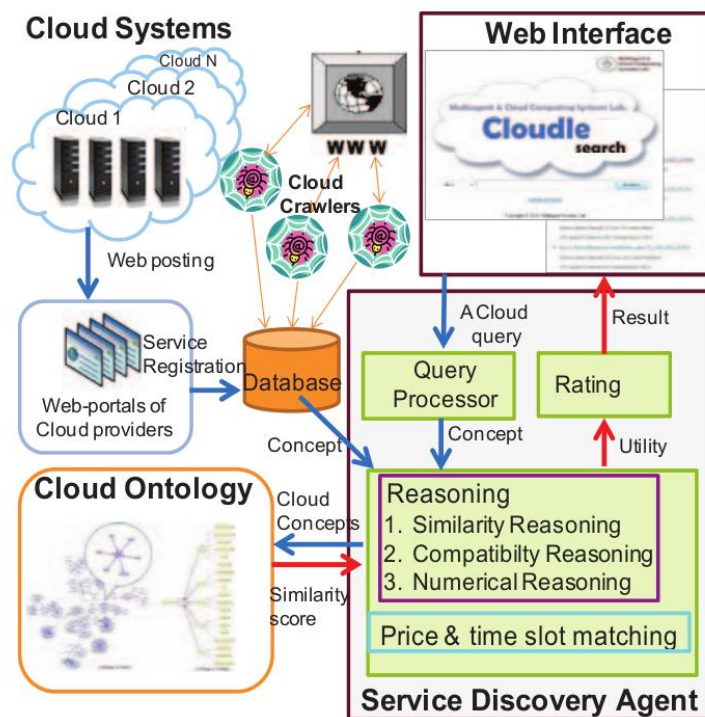


Figura 3.9: Arquitetura Cloud[12]

A computação na *cloud* consiste numa virtualização dinâmica de *hardware* como um só computador ou como vários. Isto permite o ajuste dos requisitos

de hardware, consoante a necessidade do cliente, isto é, caso o cliente necessite maior poder de processamento para atender as suas necessidade, pode negociar o contrato previamente feito e aumentar a capacidade de processamento [12].

3.3.2 *Software como Serviço*

O SaaS é um modelo de negócio, Figura 3.10, criado pelas empresas que desenvolvem *software*, que assenta numa arquitectura *web* orientada ao serviço [13].

este modelo apenas requer que os seus cliente possuam um computador capaz de correr um web browser e que este esteja devidamente actualizado.

Ao contrário do modelo tradicional, em que se fornece uma aplicação (e licença) por computador, o SaaS, com recurso à computação *cloud*, oferece a aplicação a vários utilizadores, independentemente da sua localização ou do dispositivo que utilizam para aceder.

A essência deste serviço é que o cliente não pague o *software*, mas sim o serviço prestado. O cliente ganha desta forma, em várias frentes. Utilizando os recursos *cloud* do fornecedor, permite-lhe reduzir custos em infraestruturas como *Data Centers*, ao mesmo tempo que a introdução do *software* na empresa é muito mais rápida.

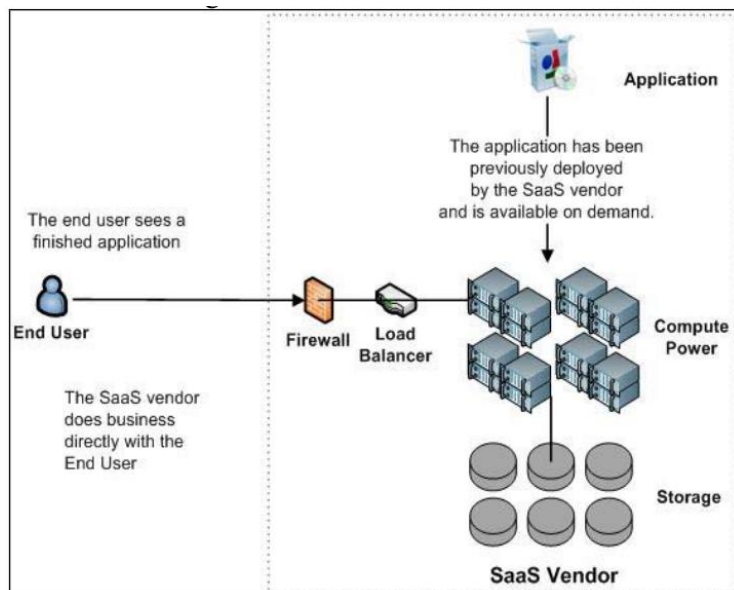


Figura 3.10: Modelo de Software como um Serviço[13]

Modelo de Negócio

O modelo de negócio do SaaS é assente numa taxa periódica (mensal, anual, etc . . .), que pode reflectir, ou não, o número de utilizadores que acedem a esse serviço ou até mesmo o nível de utilização desse serviço, dado que o consumo de recursos por parte de um cliente poderá também ser taxado.

Outra das vantagens deste serviço é o aspecto modular do mesmo. De forma geral, as empresas fornecem um serviço *standard* que fornece aos clientes serviços mínimos, ao qual o cliente pode adicionar módulos, que podem ser gratuitos ou não e que lhe podem permitir aumentar rendimento da aplicação.

Independência de Recursos

Uma das características fundamentais do SaaS é o seu acesso via *web browser*, que permite que o cliente não necessite de upgrades em termos de *hardware* constantes para acompanhar a evolução do *software*, dado que a computação é feita do lado do servidor, ao mesmo tempo que poupa em serviços de informação & Tecnologia (IT).

Escalabilidade

A instalação tradicional de *software* obriga a ter uma licença por utilizador, o que obriga a, sempre que é necessário adicionar um novo utilizador à estrutura, designar uma equipa IT para que proceda a instalação do software e à orientação tutorial do mesmo.

Com o SaaS, o cliente não precisa de se preocupar com esse tipo de tarefas, uma vez que estas ficam a cargo do fornecedor, realizando todas as acções necessárias para ajustar o serviço que fornece ao cliente, seja na adição de novos utilizadores, seja no ajuste dos recursos de *hardware* atribuídos ao cliente para que forneça a melhor performance possível.

3.3.3 Centralização de informação

Um dos grandes problemas nas organizações é a proliferação de sistemas de informação, o que faz com que os utilizadores tenham de consultar vários sistemas para aceder ou encontrar a informação pretendida [46].

As plataformas integradas vêm de certo modo ajudar à centralização da informação. Apesar de serem mais um sistema de informação, permitem a integração de sistemas já existentes e, desta forma, providenciar o acesso mais rápido à informação, o que, no caso dos departamentos/serviços técnicos das organizações, é uma mais valia, abordada em maior detalhe no Capítulo ??.

3.4 Integração da Automatização de Edifícios com WSN

Esta secção tem como fim apresentar um resumo de um caso de estudo realizado para integração de WSN com a automação. Apesar de, neste caso de estudo, se escolher um protocolo de comunicação diferente do adoptado, fica a ideia geral dos autores do artigo[47].

3.4.1 Automatização de Edifícios

A automatização de edifícios é utilizada para aumentar a qualidade da climatização dos edifícios e em paralelo reduzir os custos operacionais.

Para diminuir estes custos, é necessário adicionar ao controlo HVAC a capacidade de monitorização de variáveis, como luz e segurança, aliadas a um sistema de supervisão. De forma geral, este tipo de controlo é implementado em edifícios como escolas, hospitais e escritórios, no qual a fatura energética ao final do ano representa um valor considerável nos custos operacionais.

3.4.2 Integração com WSN

Integrar as WSN com automatização de edifícios possui varias vantagens.

principal, relaciona-se com os custos de instalação de uma WSN serem bastante inferiores ao de uma tradicional instalação HVAC, já que a cablagem é evitada.

Desta forma, os novos edifícios podem ser equipados com controlo HVAC baseado na tecnologia das WSN. Esta integração com as WSN proporciona ainda a extensão e a atualização de sistemas HVAC já existentes, aumentando assim as áreas cobertas pelo sistema em funcionamento. Com a redução de

custos de instalação, cria-se a possibilidade de expandir o número de sensores e assim aumentar a resolução espacial, o que levará a que a qualidade das medições aumente e, bem assim, influenciando diretamente a qualidade do controlo HVAC.

Outra das vantagens dos sensores sem fio é a possibilidade de estes serem descolocados e realizarem medições em locais temporários, o que permite verificar e avaliar se certas acções realizadas (como por exemplo o aumento da temperatura de uma sala) está a ter os efeitos desejados na temperatura global dessa mesma sala.

Compatibilidade

Para que seja possível integrar as WSN com a automação é necessário escolher com cuidado os protocolos de comunicação sendo preferível que estes sejam *standard*.

Utilizando protocolos *standard* a compatibilidade com soluções de controlo HVAC é assegurada.

Neste artigo os seus autores escolheram o BACnet[48] para a integração.

Implementação

Os autores do artigo realizaram a integração com um sistema HVAC através de uma *board*, com um sensor da FU Berlin [49] a correr o Contiki⁷. A comunicação com o controlo HVAC foi obtida através de um aplicação a correr sobre o Contiki que realizava a ligação com o controlo através do protocolo BACnet.

Validação

Os autores do artigo, para validar a comunicação, colocaram o sensor numa sala de estar durante 96 horas e configuraram o sistema de supervisão HVAC para alertar o utilizador sempre que existisse um desvio de um 1 grau Celsius desde o envio da última leitura. Como a temperatura da sala de estar era relativamente estável, receberam apenas 9 alarmes de alteração de temperatura. No entanto, os autores afirmam que se tivessem colocado o tempo de amostragem da temperatura em 10 minutos, teriam recebido 576 alarmes.

⁷Contiki é um sistema operativo *Open Source* desenvolvido para o *Internet of Things*.

3.4.3 Resultados

Os autores do estudo em causa concluíram que, através da integração das WSN, é possível reduzir os custos de instalação e manutenção na automatização de edifícios. Conclui-se, ainda, que soluções que originalmente não foram desenvolvidas a pensar na integração com as WSN, como o BACnet, podem ser aplicadas nesse domínio com relativo sucesso.

4

Descrição da Rede/Protocolo de Comunicação Base

O presente capítulo tem como intuito abordar em detalhe o protocolo MODBUS, que está na base da arquitectura proposta. Na primeira secção, realiza-se uma pequena introdução à história do protocolo, enquanto nas secções seguintes apresenta-se a estrutura que compõe o protocolo e como este funciona. Em seguida, abordam-se os requisitos para a implementação da comunicação MODBUS TCP utilizada no projecto. Por fim, realiza-se uma breve explicação da comunicação sem fios utilizada na WSN e os motivos que levaram a optar pelo protocolo MODBUS.

4.1 História

O protocolo de comunicação MODBUS foi originalmente desenvolvido pela Modicon, hoje conhecida como Shneider Electric e é um protocolo *de-facto*¹ *standard* para comunicação série na automação industrial desde 1979. Em

¹Palavra de origem latina, que atesta afirmação como um facto.

Abril de 2014, a Schneider decidiu transferir os seus direitos de autor para a MODBUS-IDA, uma organização sem fins lucrativos.

4.2 Modelo de Comunicação

O protocolo MODBUS está assente num modelo de comunicação cliente/servidor e funciona na óptica pedido/resposta [15]. É um modelo bastante robusto e encontra-se posicionado no nível 7 do modelo OSI, Figura 4.1.

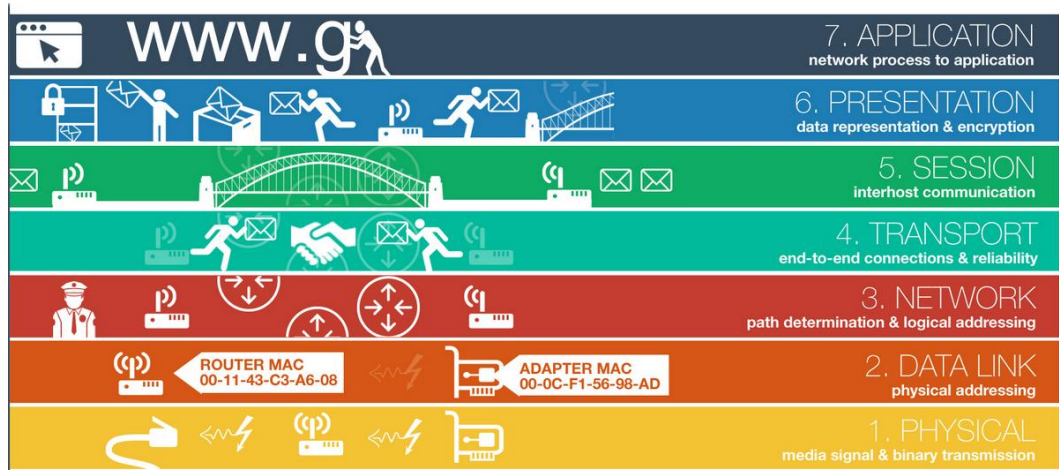


Figura 4.1: Modelo OSI[14]

A comunicação MODBUS é utilizada para comunicar com diferentes tipos de dispositivos e redes (ver Figura 4.2), e pode ser implementada através:

1. Da camada TCP/IP (utilizando a Ethernet);
2. Comunicação série (RS232; RS485, fibra óptica, radio, fio, etc);
3. MODBUS+, rede de passagem rápida de "testemunho" (*token*).

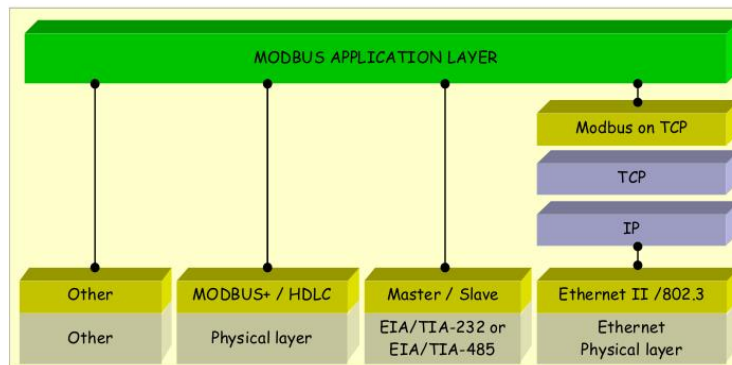


Figura 4.2: Camada de Aplicação MODBUS[15]

4.2.1 Descrição do Protocolo

O protocolo MODBUS define uma unidade de dados protocolar, (PDU, *Protocol Data Unit*), independente das sub-camadas inferiores de comunicação. O mapeamento do protocolo MODBUS, em determinados dispositivos de campo ou determinadas redes, pode obrigar a adicionar campos extras à unidade de dados do aplicativo, (ADU, *Application Data Unit*).

4.2.2 Estrutura ADU

A ADU é construída quando um cliente MODBUS inicia uma transacção MODBUS. Dentro da estrutura da ADU, Figura 4.3, é incluída trama do PDU, que estabelece o formato do pedido feito pelo cliente e acção a realizar.

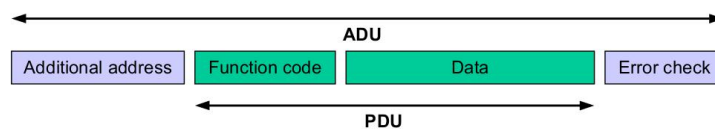


Figura 4.3: Estrutura da ADU[15]

Pela Figura 4.3 sabemos que a ADU possui quatro campos:

Additional Address Contem informação do endereço do servidor MODBUS que o cliente pretende consultar;

Funciton Code Quando é feito um pedido do cliente ao servidor, esta função especifica o tipo de acção/resposta que o servidor deve realizar;

Data Este campo contém informação utilizada pelo servidor, poderá conter informação acerca de registos a consultar e a quantidade de informação requerida desses mesmos registos;

Error Check No caso de transacção cliente-servidor conter campos inválidos este campo irá conter uma resposta com os códigos de excepção (*exception codes*).

Codificação MODBUS

O protocolo MODBUS utiliza a representação *'big-Endian'* para blocos de dados e para os endereços. Isto significa que quando a informação numérica transmitida é superior a um *byte*, é enviado primeiro o *byte* mais significativo, ver Tabela 4.1.

Tabela 4.1: *Protocolos de comunicação Ethernet/IP, FieldBus [15]*

Tamanho do Registo	Valor	Exemplo
16 - Bits	0x1234	O primeiro <i>byte</i> a ser enviado 0x12 depois 0x34

4.3 Registos MODBUS

O protocolo MODBUS baseia o seu modelo de dados numa série de tabelas que possuem características únicas, como se pode ver na Tabela 4.2.

Tabela 4.2: *Tipo de Registos[15]*

Tabelas principais	Tipo de Objecto	Leitura/Escrita
Discretes Input	Single Bit	Apenas Leitura
Coils	Single Bit	Leitura/Escrita
Input Registers	16-Bit Word	Apenas Leitura
Holding Registers	16-Bit Word	Leitura/Escrita

Apesar do protocolo MODBUS efectuar uma distinção entre registos, é possível um objecto MODBUS possuir as quatro camadas e, assim, dependendo do pedido recebido, responder consoante o tipo.

Para cada tabela enunciada, o protocolo permite selecionar até 65546 registros e a operação daí resultante, leitura ou escrita, está desenhada para abranger múltiplos dados até ao limite da memória da tabela em questão, que depende da função executada.

Funções MODBUS

Como referido em pontos anteriores, este modelo de comunicação funciona à base de códigos, isto é, mediante um código recebido, o servidor executa uma determinada tarefa, tal como abrir uma válvula ou enviar para o cliente o valor da temperatura de um sensor. Estas funções encontram-se distribuídas pelas três categorias seguintes:

Funções Públicas Funções validadas pela MODBUS-IDA, documentadas e testadas;

Funções de Utilizador Funções desenvolvidas pelo utilizador para um fim específico, que não são de domínio público;

Funções Reservadas Funções utilizadas por empresas e em determinados produtos que legalmente não se encontram disponíveis para utilização pública.

A Tabela 4.3 apresenta os códigos e funções mais comuns no protocolo MODBUS.

Tabela 4.3: *Funções MODBUS[15]*

Código	Função	Classe
Read Coil Status	01	1
Read Input Discretes	02	1
Read Holding Registers	03	1
Read Input Registers	04	1
Write Coil	05	1
Write Single Register	06	1
Read Exception Status	07	1
Force multiple coils	15	2
Read/write registers	23	2

4.4 MODBUS TCP/IP

A comunicação MODBUS sobre TCP/IP provoca algumas alterações na trama geral do MODBUS, (Figura 4.3) retira dois campos, o *Additional Address* e *Error check*, que passam a ser contidos no cabeçalho da trama, Figura 4.4, denominado MODBUS *Application Unit Data* (MBAP).

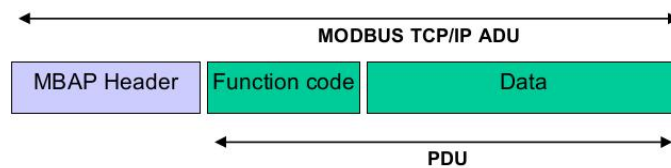


Figura 4.4: Trama MODBUS TCP/IP[16]

Esta modificação na trama MODBUS gera algumas diferenças na comunicação, quando comparada com o MODBUS RTU utilizado na comunicação série:

1. O endereço do servidor MODBUS (*slave address*) utilizado na comunicação serie é substituído por um identificar unitário (*Unit ID*) dentro do MBAP. Esta identificação é utilizada para conseguir diferenciar dispositivos de campo (ex: PLC) que estejam associados a único IP, Figura 4.5.

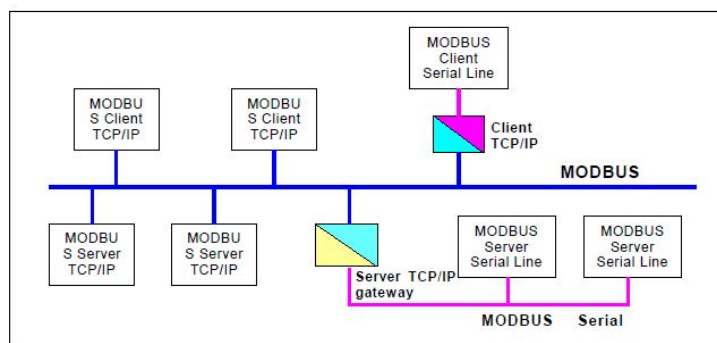


Figura 4.5: Arquitetura da Comunicação MODBUS TCP/IP[16]

2. Todos os pedidos MODBUS são desenhados de forma que o receptor possa verificar que o envio da mensagem foi finalizado.

3. Quando o MODBUS corre em cima da comunicação TCP, é colocada informação adicional no cabeçalho do MBAP para que o receptor consiga perceber os limites da mensagem, mesmo que a mensagem vá dividida em vários pacotes.

Regras de Implementação

O meio de comunicação MODBUS requer que seja estabelecido uma ligação TCP entre o cliente e servidor, como representado na Figura 4.6.

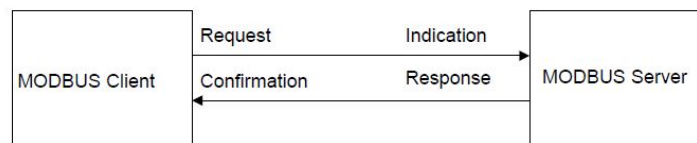


Figura 4.6: *Comunicação entre Cliente e Servidor [16]*

Para que seja estabelecida uma comunicação estável, o protocolo de comunicação requer as seguintes regras:

1. Implementar uma gestão de ligações TCP;
2. Deixar a ligação TCP aberta com o dispositivo remoto, e não abrir e fechar sempre que se efectua uma transacção MODBUS TCP;
3. Permitir varias ligações TCP em simultâneo;
4. Em caso de comunicação bidireccional entre servidor e cliente é necessário criar ligações separadas para o fluxo de informação tanto do cliente como do servidor;
5. A trama TCP deverá apenas transportar informação relativa ao ADU MODBUS.

4.5 Rede sem fios

Esta secção tem como intuito abordar as normas de comunicação utilizados pelas WSN que se pretende integrar com o sistema de automação e controlo no

projecto da Tese, de acordo com o caderno de encargos e arquitectura proposta no Capítulo 2.

ELTEK - Data Loggers

A empresa ELTEK, sediada no Reino Unido, produz sensores que utilizam a comunicação sem fios para enviarem os dados. Estes sensores são de grande fiabilidade utilizando a norma *standard* EN300-220 [26] para a comunicação sem fios.

Comunicação

Como já referido, a rede criada pelos sensores sem fios ELTEK utiliza um protocolo de comunicação proprietário que comunica sobre a norma EN300-220, possuindo grande alcance de comunicação *indoor* e *outdoor*. O protocolo da ELTEK apresenta as seguintes características na Tabela 4.4:

Tabela 4.4: *Especificações Técnicas [18] [26]*

Especificação	Descrição
Meio de Comunicação	Radio Frequência
Frequência	Banda dos 438 MHZ
Potencia de Transmissão	10 mW
Intervalo de Transmissão	De 1 s a 4 horas
Error Check	Sem mecanismos de retransmissão
Receptor	Recebe dados de/até 160 transmissores em simultâneo
Topologia	Estrela
Duração da Bateria	Até 5 anos

Carel - Sensores *Wireless*

Os sensores Carel, também utilizados para efeitos de teste relativamente aos sensores actuadores, comunicam entre si através do protocolo ZigBee, abordado no Capítulo 3 e permitem ao utilizador final comunicar com eles através do protocolo MODBUS RTU. Para comunicar com estes, utiliza-se um ponto de acesso que, mediante a identificação definida para cada sensor, comunica com estes, pedindo a informação que vem na trama MODBUS.

4.6 Motivação

A escolha do protocolo MODBUS para realização deste projecto/tese deveu-se ao facto de o protocolo em questão ser um protocolo aberto e somente algumas funções serem privadas, o que facilita a comunicação com os dispositivos de campo que o utilizem.

Outra vantagem deste protocolo é o facto de a maioria dos fabricantes optarem por o incluírem nos seus dispositivos, tornando-se assim um protocolo universal e de rápida implementação

Uma vez que a tese também se foca na centralização de informação, a inclusão de um protocolo de comunicação que permite comunicar com diferentes tipos de dispositivos de diferentes fabricantes, faz todo o sentido em termos práticos.

5

Arquitetura do Sistema

O presente capítulo tem como intuito apresentar o diagrama do sistema a implementar para alcançar o objectivo da integração entre a rede de sensores e o autómato, seguindo-se apresentação de forma detalhada a estrutura de cada componente da arquitetura projetada para aplicação.

5.1 Arquitectura Proposta

O sistema proposto para aplicação, representado na Figura 5.1, é composto por três áreas distintas, compostas pelo servidor *cloud*, WSN e pelo controlador Carel UltraCella, interligados por dispositivos de rede.

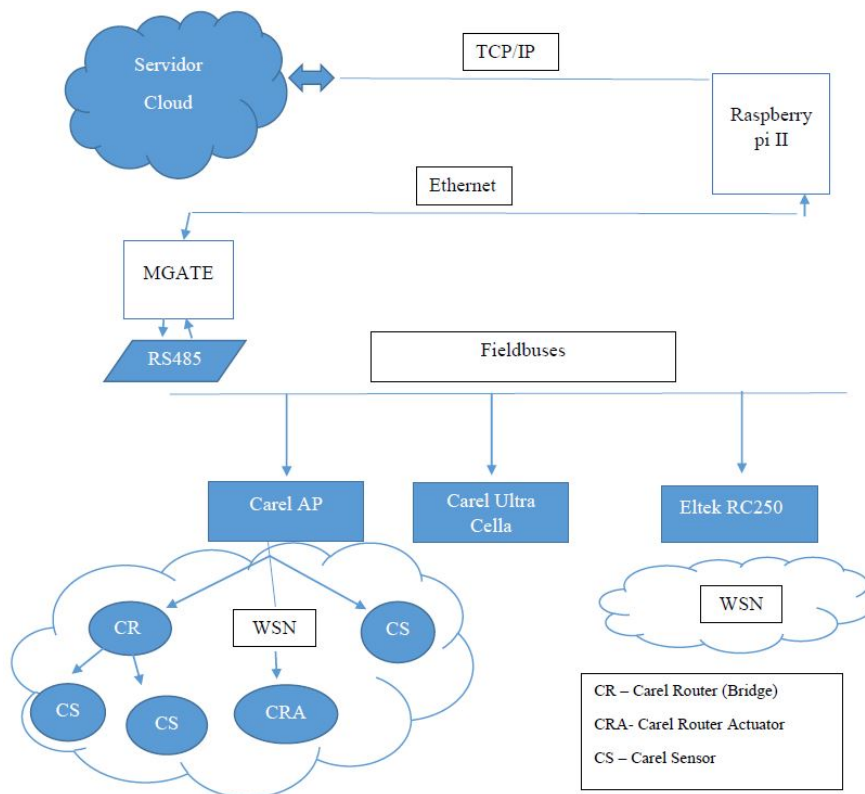


Figura 5.1: *Arquitetura de Sistema*

Dispositivos Integrantes

O sistema apresentado na Figura 5.1, é composto pelos seguintes dispositivos:

1. Moxa Mgate 3180;
2. Moxa Nport 5110;
3. Carel Wireless Access Point;
4. Carel Router Actuator;
5. Carel Wireless Sensor (Temperatura; Humidade; Luz);
6. Carel UltraCella;
7. RaspBerry Pi II;
8. Eltek RC250;

9. Sensor Humidade, Temperatura Eltek GD10.

De forma a justificar e, ir de encontro aos requisitos definidos no Capítulo 2, utilizou-se os seguintes componentes:

Moxa

A utilização dos dispositivos Moxa, 5110 e 3180 surge na experiência acumulada pela empresa com este tipo de dispositivos de rede, que permitem a comunicação com o servidor (Nport 5110), enviando tramas de dados no formato XML(Extensible Markup Language) e a conversão MODBUS RTU para MODBUS TCP, que permite comunicar com os dispositivos que o façam por MODBUS RTU ou ASCII.

Carel

Os dispositivos sem fio Carel foram escolhidos de forma a que cumprissem os requisitos deste projecto e, ao mesmo tempo, aumentar o portfólio da empresa, que se dedica, como já referido, à monitorização de diversas variáveis, recorrendo a WSN. Como os dispositivos encontrados no mercado eram passíveis de se comunicar pelo meio de comunicação MODBUS e comunicavam entre eles através do protocolo ZigBee, entendeu-se que seriam uma boa escolha, uma vez que abordavam as temáticas discutidas ao longo da tese. O controlador de frio Care UltraCella foi escolhido, dada possibilidade de integrar com os dispositivos sem fio escolhidos.

Raspberry Pi

O Raspberry foi escolhido dada a sua robustez, já comprovada pelo modelo anterior e pela sua versatilidade. Uma vez que este pequeno "computador" corre em si uma distro Linux, permite criar diversos algoritmos para conseguir comunicar com vários dispositivos através de múltiplos protocolos e, assim, aumentar a versatilidade e flexibilidade da solução.

Eltek

Os sensores Eltek já estão presentes no portfólio da empresa há vários anos, pelo que, integraram o projecto, apenas, para complementar a estrutura apre-

sentada, integrando a parte de controlo com a sua capacidade de aquisição de dados no meio, através da comunicação sem fios.

De acrescentar que nos teste de implementaçom a comunicação com estes não é apresentada uma vez que as medições realizadas por estes sensores serão obtidas através de "queries" à base de dados existente.

5.2 Estrutura WSN

A rede de sensores apresentada na Figura 5.1, é composta pelos seguintes dispositivos:

1. Receptor Carel, Figura 5.2, permite o acesso às leituras enviadas pelos sensores, através de uma ligação MODBUS RTU. O receptor, como referido no ponto 3.1.1, comporta-se como um ZC, criando a rede *wireless* ao qual os sensores se irão associar. É explicado em detalhe a forma como se associa os sensores ao receptor carel em [17];



Figura 5.2: Receptor Carel [17]

2. Sensor de temperatura, humidade e luz Carel, Figura 5.3. Este sensor estabelece comunicação com a rede através do protocolo Zigbee. Enviando amostras para a base de dados de 5 em 5 minutos.



Figura 5.3: *Sensor de Humidade, Temperatura e Luz Carel. [17]*

3. Receptor Eltek, Figura 5.4. Este receptor permite receber valores de até 160 sensores. O protocolo de comunicação criado por este receptor é assente na norma EN300-220. Apenas suporta a topologia estrela e não permite comunicação bidirecional, pelo que são os sensores a tomar a iniciativa de comunicação.



Figura 5.4: *Receptor Eltek [18]*

4. Sensor de temperatura e humidade Eltek 5.5, Figura, estabelece comunicação com a rede através de um protocolo proprietário que corre sobre a norma EN300-220. O tempo de amostragem destes sensores é de 5 minutos.



Figura 5.5: *Sensor de Temperatura e Humidade G10 [18]*

5. Moxa Nport 5110, fabricado pela Moxa [1], este dispositivo actua como um *gateway*, providenciando a ligação entre os receptores da WSN e o servidor. O NPort 5110 possui uma entrada RS232, que o permite ligar à saída dos receptores.



Figura 5.6: *Gateway - Moxa Nport 5110, [19]*

5.3 Estrutura Servidor - Cloud

O servidor *cloud* tem instalado o sistema operativo (SO) Linux baseado na distribuição Ubuntu Pangolin Server 12.04 *Long term Service* (LTS), a correr o servidor Apache2 para responder a pedidos *http*. Os dados recebidos são armazenados numa base de dados relacional *open source*, PostgreSQL.

Devido a políticas comerciais não é possível enunciar as características do *hardware* do servidor. No servidor em questão está alojada a plataforma Vi-

GIE, Figura 5.7, e o módulo de controlo desenvolvido para a Tese, Figura 5.8.

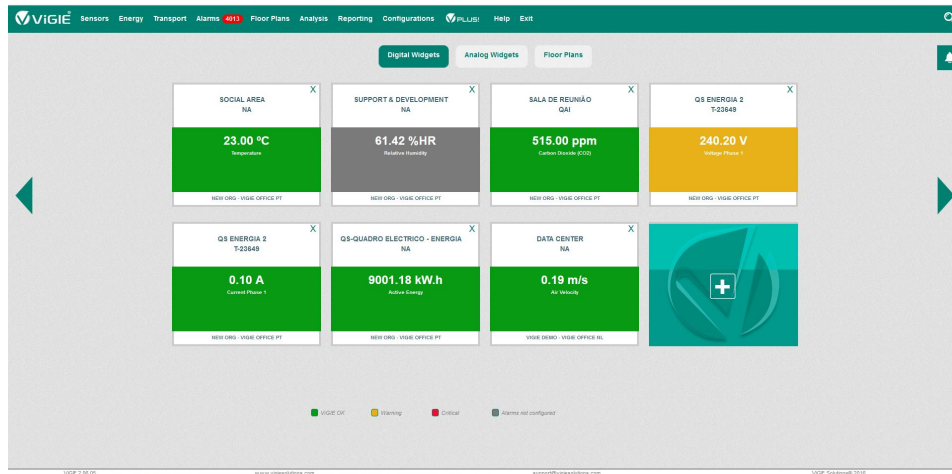


Figura 5.7: Plataforma Vigie [21].

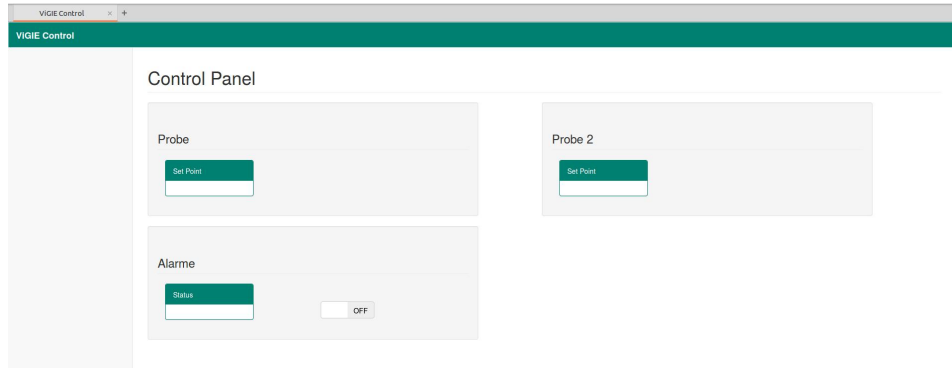


Figura 5.8: Módulo de Controlo.

5.4 Estrutura PLC

O PLC, Figura 5.9, utilizado para realizar os testes de comunicação, é um controlador fornecido pela Carel, empresa especializada em controlo HVAC.

As especificações ao nível de *hardware*, como processador e memória virtual, deste controlador não são fornecidas pela marca, pelo que impossibilita dessa forma de serem apresentadas.

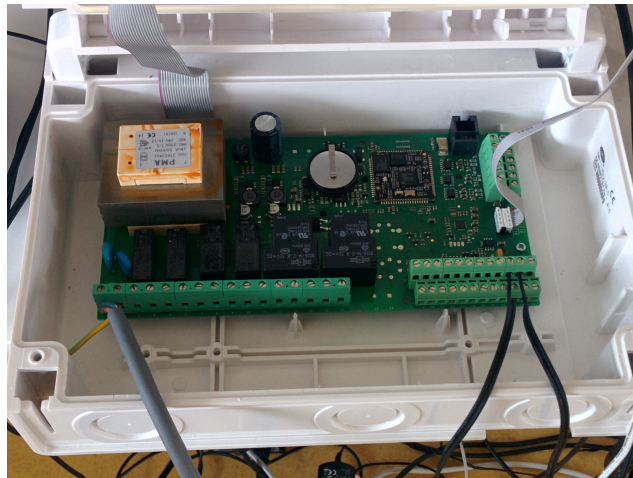


Figura 5.9: *PLC Carel.*

O PLC em questão vem programado para a realizar o controlo de temperatura de câmaras congeladoras industriais. Possui entradas para 6 sondas de temperatura, 4 relés e varia portas I/O. A carel disponibiliza o protocolo MODBUS RTU (comunicação série) para aceder aos registos MODBUS dos I/O do PLC. No entanto, para se aceder ao PLC via *web*, foi necessário associar ao dispositivo um conversor MODBUS RTU / TCP, representado na Figura 5.10.



Figura 5.10: *Conversor MODBUS RTU / TCP.*

Este conversor, fabricado pela Moxa, permite correr o protocolo TCP em

cima do protocolo MODBUS e, assim, conseguindo-se o acesso ao PLC via TCP/IP.

5.5 Raspberry Pi

O Raspberry Pi II, Figura 5.11, foi adicionado à estrutura de forma a aumentar a segurança para que não se realizasse um acesso directo ao PLC via *web*. Desta forma, realiza-se uma nova autenticação ao pedido de acesso ao PLC sempre que é estabelecida uma nova ligação.

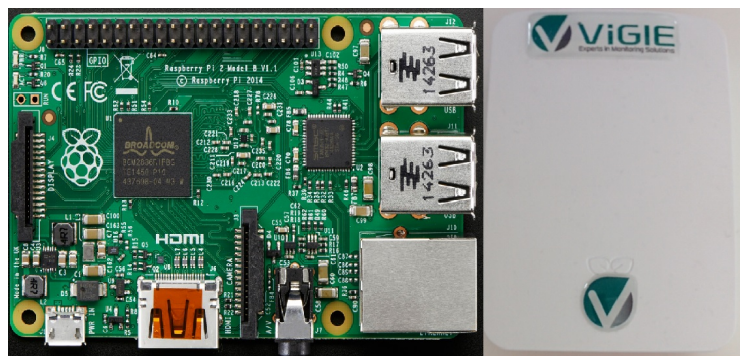


Figura 5.11: *Aspecto do Raspberry Pi II Modelo B.*

O Raspberry Pi II possui as seguintes especificações técnicas:

1. Processador Quad Core, com uma velocidade de relógio de 900 MHz;
2. 1GB RAM;
3. Quatro portas USB;
4. Saída HDMI;
5. Entrada de alimentação USB;
6. Leitor de cartão MicroSD.

No Raspberry foi instalado o SO Raspbian, que é baseada na versão Debian Linux, mas otimizada para correr no Raspberry Pi.

6

Implementação e Validação

O presente capítulo tem o intuito de apresentar todos passos realizados para desenvolver aplicação. Na primeira secção é apresentado o fluxograma que detalha o funcionamento da aplicação. Nas secções seguinte, são apresentados os passos efetuados para chegar à comunicação com o PLC e com o servidor. De referir que, todos os passos efectuados na configuração do Raspberry, se encontram no Anexo A.

6.1 Fluxograma

Neste subcapítulo pretende-se dar uma imagem sobre como funciona a arquitectura abordada no capítulo 5.

fluxograma que se segue na Figura 6.1, é ilustrativo de toda cadeia que está por de trás do módulo de controlo.

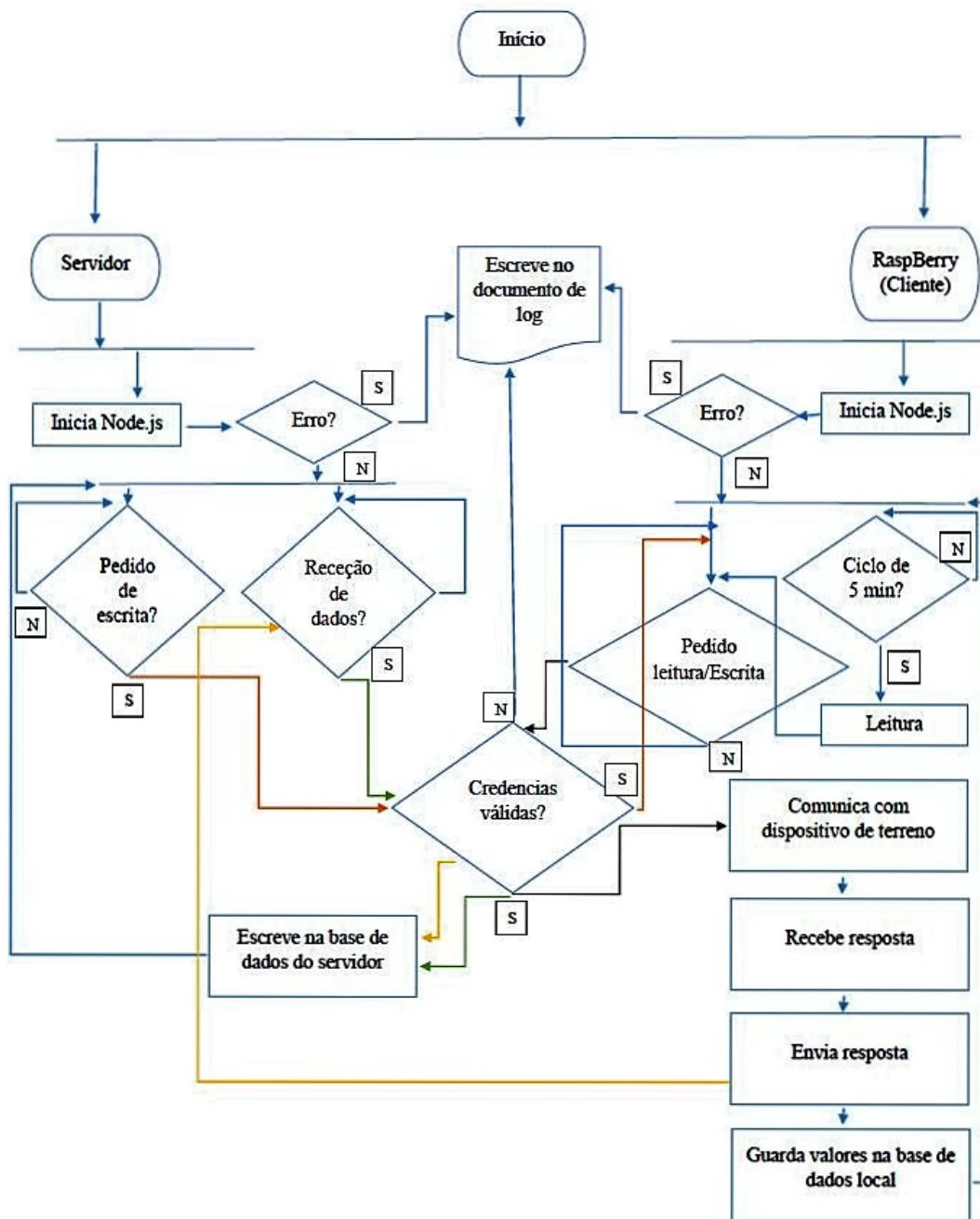


Figura 6.1: Fluxograma representativo do funcionamento da arquitectura.

Tal como se vê no fluxograma, Figura 6.1, a cadeia de funcionamento do sistema está dependente *Application Programming Interface* (API), que é assente em Node JavaScript. Esta API lida com todos os pedidos da parte do cliente e servidor e procede à autenticação e validação dos mesmos, impedindo

utilização indevida por parte de utilizadores mal intencionados.

API-NodeJS

Tal como já referido, a API presente na arquitectura foi desenvolvida através de um "motor" de execução de eventos em JavaScript[50], denominado NodeJS.

O NodeJS[51] é um sistema de gestão de eventos, i.e. A cada evento, pedido/resposta, desencadeia um processo de resposta, de forma assíncrona e escalável, permitindo uma gestão de recursos mais eficiente por parte do servidor e evitando, assim, que num período de maior "procura", não se crie *bottlenecks*. O utilizador consegue, então, uma experiência de utilização fluída.

Através desta API é feita a gestão dos pedidos por parte do servidor e/ou cliente, bem como a autenticação dos mesmos.

Autenticação

A autenticação utilizada pelo sistema, denomina-se *basic-auth*[52], que é um módulo do JavaScript que permite gerir de forma simples a autenticação. Este módulo fica à escuta num determinado porto, e ao receber um pedido de autenticação procede verificação das credenciais do pedido, e no caso de este possuir as credencias válidas, dá passagem ao evento que irá desencadear uma cadeia de um ou vários eventos, Figura 6.2, dependendo do tipo de pedido.

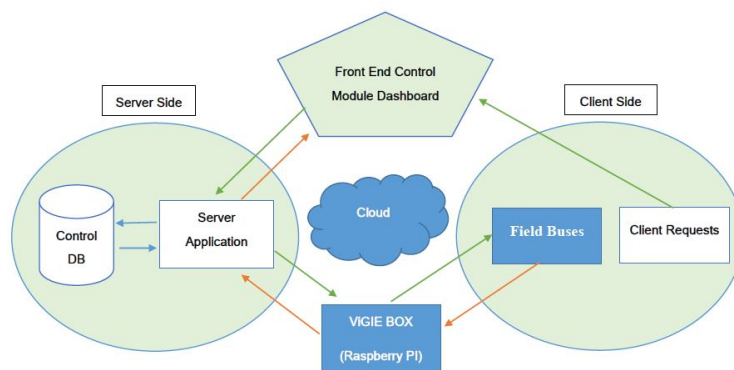


Figura 6.2: Cadeia de Eventos

6.2 Plataforma Web

A plataforma que vai servir de interface gráfica com o utilizador, ou seja, o HMI de controlo, foi desenvolvida através de uma *framework*, denominada por Bootstrap[53], que permite criar páginas responsivas, i.e, a página em si, adapta-se de forma automática ao dispositivo que a consulta, ver Figura 6.3. Isto para que, no caso de ser um telemóvel, o formato da página se adapte melhorando a utilização por parte do utilizador.

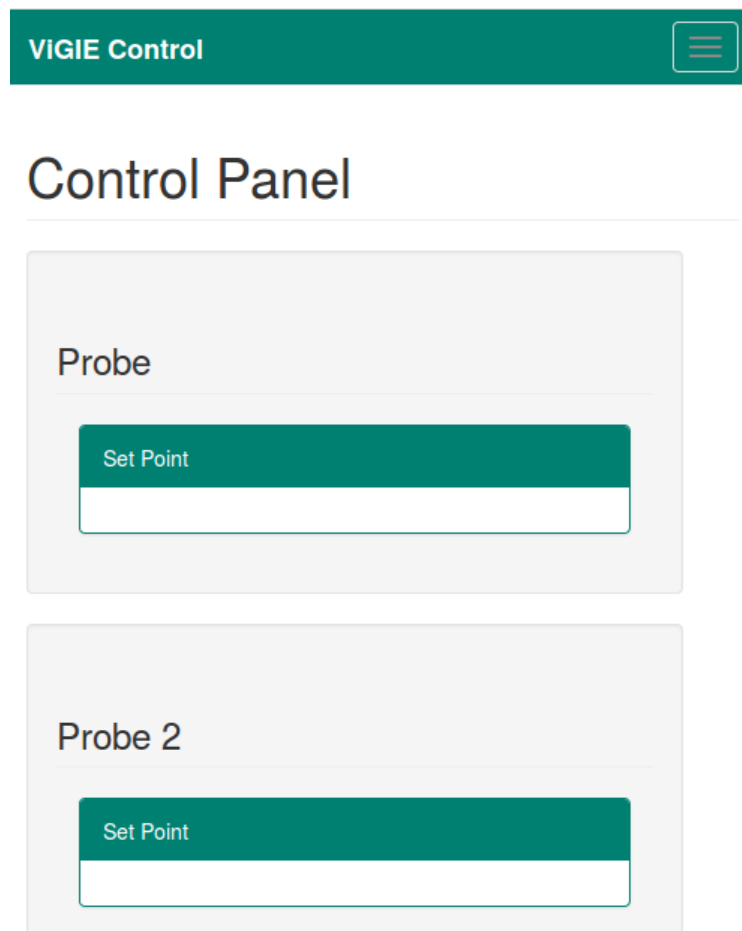


Figura 6.3: *Plataforma Adaptável*

No caso de ser um dispositivo com grande resolução, como um computador ou *tablet*, a plataforma tem o aspecto apresentado na Figura 6.4.

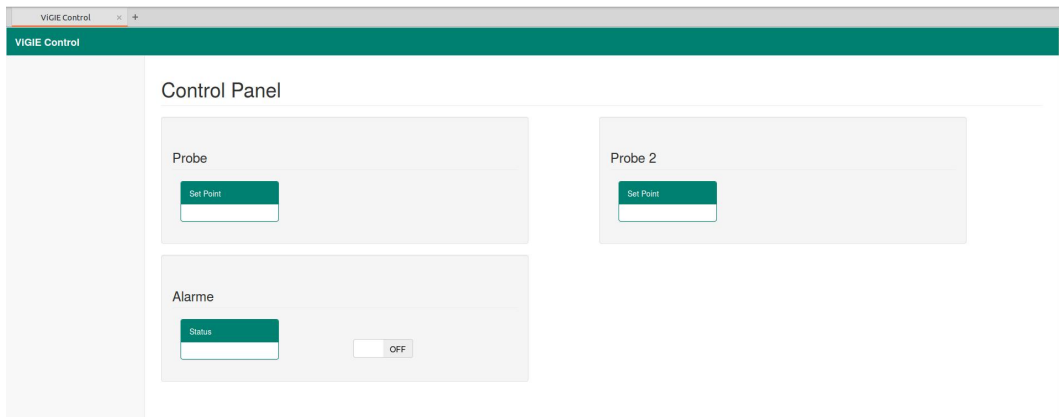


Figura 6.4: *Aspecto da plataforma num computador/tablet*

A plataforma *web* foi desenhada para que se auto ajuste também conforme o tipo de dispositivos que a ela estejam associados. No caso de o utilizador apenas possuir sensores, aparecer-lhe-á uma caixa com o valor que o sensor está a medir, bem como a identificação do mesmo. Se este possuir dispositivos que permitam atuação como um alarme sonoro, é-lhe criada uma caixa com o estado do alarme e um botão *ON/OFF*, que lhe permite accionar ou desligar o alarme.

6.3 Interface Raspberry - Fieldbuses

O interface com os dispositivos de campo (*Fieldbuses*) foi realizado através de um *script* em Python[54] que, através de uma biblioteca de MODBUS, fornecida por[55], permitiu criar um *script* bastante flexível e fluído.

Os mecanismos para comunicação com os dispositivos de terreno são bastante simples, utilizando funções específicas para cada tipo de registo MODBUS, ver Figura 6.5.

```
if mod == 'teste':
    print "Test Mode"
    client = ModbusClient('10.192.10.163', port=5200)
    #if connection_refused == 0:
    rc = client.read_holding_registers(0, 1, unit=4)
    response = float(rc.registers[0])*0.0061036709-200
    print response
```

Figura 6.5: *Parte do código de teste de comunicação em Python*

Na Figura 6.5, podemos ver a utilização das funções disponibilizadas pela biblioteca referida.

Nesta, é realizada uma ligação a dispositivo MODBUS, por TCP, e é-lhe pedido o valor do registo do tipo "Holding Register", no seu endereço 0, ao seu "Slave ID" definido com 4. A resposta é depois tratada, uma vez que, regra geral, os valores que vêm como resposta estão num formato que carece de ser convertido para uma resolução que seja possível de ser compreendida, i.e, para um sensor de temperatura, que retorne um valor de 20° C, é recebido um valor 20000, ao qual é necessário aplicar uma resolução de 0,001. Estas conversões são fornecidas pelos fabricantes dos dispositivos uma vez que variam de fabricante para fabricante.

Estruturação do Script

O *script*, desenvolvido para a comunicação com dos dispositivos de campo, necessitou de ter em conta vários aspectos para que fosse possível a integração de vários modelos de equipamentos e de forma a atender à especificidade de cada fabricante.

No sentido a tornar o *script* flexível, foi adicionado um módulo de comunicação, *psycopg2* [56], com uma base de dados PostgreSQL [57] que conterá toda a informação e detalhe de cada registo MODBUS e para que o script seja capaz de comunicar com este.

Desta forma, é possível criar "*drivers*" de comunicação para cada fabricante sem ser necessário alterar o script sempre que se pretenda adicionar um equipamento que não esteja no portfólio, excepto nos casos pontuais em que a estrutura do *script* não consiga contemplar esses mesmos casos.

Modelo da Base de Dados

O modelo adotado da base de dados segue a estrutura apresentada na Figura 6.6. Este modelo foi pensado de forma a que o *script* conseguisse dar resposta a "quase todas" as variantes e especificidades da comunicação do protocolo MODBUS. Estas "variantes" referem-se aos detalhes de cada registo que se pretenda consultar ou alterar, tal como referido na secção 4. Os registos MODBUS possuem um endereço e um tipo e este modelo contempla essas variáveis, por

forma a criar uma interface de comunicação para cada dispositivo de que se pretenda comunicar.

A especificidade da comunicação vem com a forma de como os dados são enviados, em registos de 16 ou 32 *bits*, e o formato, *Big or Small Endian*. No caso em que os registos são de 32 *Bits* é necessário consultar dois registos e concatená-los num só valor. O formato *Big or Small Endian* vem com o modo de como é enviado o byte mais significativo MSB (*Most Significant Byte*) ou LSB (*Least Significant Byte*), i.e, quando é utilizado o formato MSB, por exemplo num envio de valor 1940, enviado em dois pacotes de dados 19 e 40, que depois é recebido e agrupado nessa ordem. No formato LSB, o valor 1940 é enviado em dois pacotes de dados mas de forma inversa, 40 e 19, por isso, quando o valor está formatado em *Small Endian*, é necessário agrupá-lo de forma inversa.

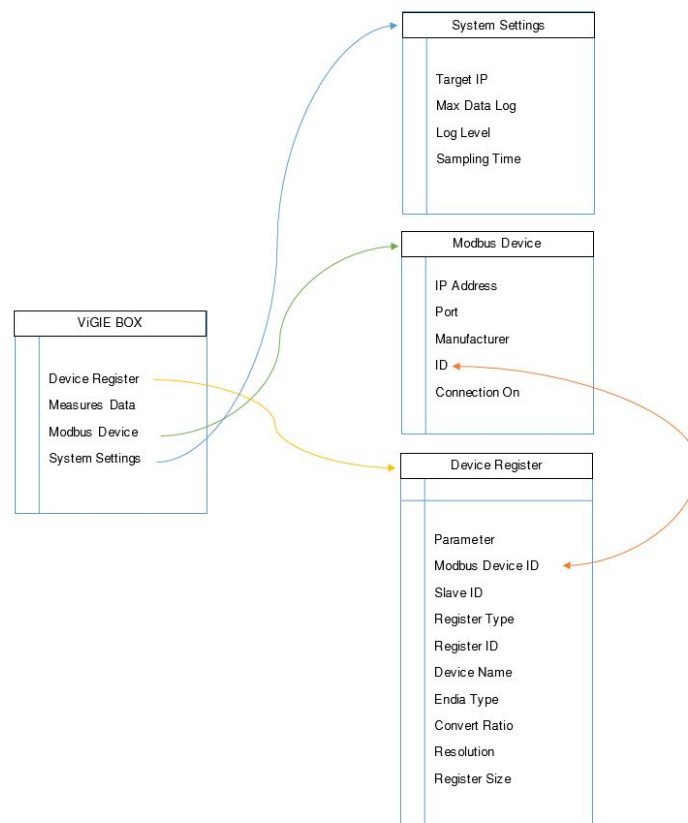


Figura 6.6: Modelo da Base de Dados

Como se pode ver na Figura 6.6, e como já explicado, este modelo foi pen-

sado para responder a todas as especificidades. Para que melhor se compreenda, aqui fica uma breve explicação das tabelas da base de dados apresentada:

Esquema - ViGIE BOX

Este esquema contém quatro tabelas, no entanto, só se explicam três delas, uma vez que a tabela "Measures Data" contém informação comercial que não pode ser apresentada.

Tabela - System Settings

Esta tabela contém a informação sobre o IP para o qual será enviada a informação dos dispositivos de campo que será recolhida (*Target IP*). Pode ser para um servidor (plataforma) local ou remoto.

O tempo (em dias) de *logging* que se manterá antes de ser apagado. Este *logging* refere-se a informação recolhida sobre os dispositivos de campo que, no caso de quebra de comunicação com o servidor, será mantida.

O nível de *logging* referente ao resultado do processamento de todos os processos que correm no Raspberry, de forma a que, na eventualidade de um erro durante o processo de execução, seja possível verificar em que momento do código esse erro se procedeu e também qual a resposta a esse mesmo erro por parte do SO.

Por último, temos o tempo de amostragem ou de contacto com os dispositivos de campo, em que estes respondem aos mais variados pedidos de informação por parte dos *master*.

Tabela - Modbus Device

Nesta tabela temos todos os dispositivos de MODBUS que são (ou podem) ser contactados pelo Raspberry. Estes, são diferenciados pelo seu endereço IP e pela porta de comunicação, que, por defeito, é a 502, ver Capítulo 4.

Nesta tabela é ainda permitido "Desligar" a comunicação com um dispositivo que, por vezes, é necessária quando, por um qualquer motivo, este se encontra desligado, poupando se assim tempo de execução do *script*.

Tabela - Device Register

Esta tabela contém toda a informação necessária para que se possa aceder aos registos MODBUS e está associada através de uma chave "estrangeira" à tabela "MODBUS Device". Esta associação é feita de forma a evitar dados replicados e para que seja possível contactar-se vários registos em cada dispositivo MODBUS.

Esta tabela contém os seguintes campos:

Parâmetro Nesta tabela consegue-se especificar o parâmetro do qual se pede informação ao dispositivo de campo, que pode ser desde o estado de um sensor de contacto a uma bomba de água (entre muitos outros).

Slave ID Esta tabela contém também informação que possibilita diferenciar vários dispositivos "escravos" (*Slave ID*) que estejam associados a um único IP. Assim, torna-se muito útil nos casos em que, vários dispositivos de campo comunicam apenas por MODBUS RTU e, ainda nos casos em que, por meio de um dispositivo que faz a conversão MODBUS RTU para MODBUS TCP, passam a comunicar por MODBUS TCP, mas ficando agrupados num único endereço IP (em alguns casos). No entanto, através da identificação do seu "Slave ID" é, assim possível produzir uma distinção no código aquando do estabelecimento da comunicação com os diferentes dispositivos MODBUS e, assim aceder aos seus registos de forma diferenciada.

Register Type Concluído o estabelecimento da comunicação, é necessário definir o tipo de registo que se irá consultar uma vez que, para cada tipo, existe uma função diferente para o aceder. Nesta linha "Register Type" vem a informação do registo.

Device Name Contém informação acerca do registo, o que permite, em caso de erro, perceber a que componente do dispositivo se refere. Por exemplo, num caso de um autómato que possui 2 sensores de temperatura, este campo permite especificar a qual dos sensores este registo se refere (ex: Sensor 1A - Temperatura).

Endian Type Formato em que se apresentam os valores, *Small ou Big Endian*.

Convert Ratio Este campo é utilizador para inserir fórmulas matemáticas para que seja possível converter o valor recebido. Por vezes, o valor que é enviado pelo dispositivo de campo vem num valor que precisa de ser tratado, como já explicado anteriormente.

Resolution Resolução da medida obtida.

Register Size Tamanho do Registo, pode ser 16 ou 32 *bits*.

6.4 Validação

Concluído o desenvolvimento do interface entre o Raspberry e os dispositivos MODBUS, passou-se à validação do mesmo através dos respectivos testes.

6.4.1 Testes - Comunicação

De forma a testar a comunicação entre o Raspberry e os dispositivos de campo, utilizou-se um analisador de comunicação MODBUS, denominado *MGate Manager*, fornecido pela Moxa [1].

Através deste *software* é possível verificar em tempo real os pedidos efectuados por um dispositivo *Master* a um dispositivo *Slave* MODBUS.

Configurações MODBUS

No teste que se segue, utilizou-se o seguinte material:

1. Mgate 3180, conversor RS485- Ethernet;
2. AP Carel;
3. Sensor de Temperatura, Humidade e Luz Carel;
4. Sensor Actuador Carel;
5. Mgate Manager;
6. Raspberry.

Ao nível das configurações MODBUS, e seguindo o mapa de registos fornecido pela Carel [17], foram as seguintes apresentadas na Tabela 6.1.

Tabela 6.1: Tabela de Registos MODBUS

Dispositivo	Tipo de Registo	Endereço	Slave ID
Sensor de Temperatura	Input Register	6	16
Sensor de Humidade	Input Register	8	16
Sensor de Contacto	Coil Status	4	24

Resultados

Após aplicadas as configurações referidas no ponto anterior, correu-se o *script* de comunicação, obtendo-se o seguinte *output* no software de análise de tráfego, representado pela Figura 6.7:

Traffic Monitor (MGate MB3180 10.192.10.67)

All
 Exception only
 Slave ID only
 Source only
 Function code only

Filter info: Start Stop Clear

No.	Time	Src. & Dst.	Type	Slave ID	Function Code	Data	Comment
1	0.000	10.192.10.83:44200<-	TCP Req.	16	0x04	00 01 00 00 00 06 10 04 00 06 00 01	Read input register
2	0.005	Port1->	RTU Req.	16	0x04	10 04 00 06 00 01 D2 8A	Read input register
3	0.030	Port1<-	RTU Resp.	16	0x04	10 04 02 00 E4 45 78	Read input register
4	0.030	10.192.10.83:44200->	TCP Resp.	16	0x04	00 01 00 00 00 05 10 04 02 00 E4	Read input register
5	0.030	10.192.10.83:44200<-	TCP Req.	16	0x04	00 02 00 00 00 06 10 04 00 08 00 01	Read input register
6	0.035	Port1->	RTU Req.	16	0x04	10 04 00 08 00 01 B3 49	Read input register
7	0.060	Port1<-	RTU Resp.	16	0x04	10 04 02 00 37 04 E5	Read input register
8	0.060	10.192.10.83:44200->	TCP Resp.	16	0x04	00 02 00 00 00 05 10 04 02 00 37	Read input register
9	0.060	10.192.10.83:44200<-	TCP Req.	24	0x01	00 03 00 00 00 06 18 01 00 04 00 01	Read coils
10	0.065	Port1->	RTU Req.	24	0x01	18 01 00 04 00 01 BE 02	Read coils
11	0.085	Port1<-	RTU Resp.	24	0x01	18 01 01 00 56 D4	Read coils
12	0.085	10.192.10.83:44200->	TCP Resp.	24	0x01	00 03 00 00 00 04 18 01 01 00	Read coils

Previous page Next page

Figura 6.7: Análise da comunicação através do MGate Manager

Podemos verificar pela Figura 6.7, que foi efectuado um pedido MODBUS TCP sendo esse mesmo reconvertido para MODBUS RTU. No pedido em questão que foram pedidos ao *Slave ID* 16, dois registos com a função 0x04 (*Input Register*), e ao *Slave ID* 24 um registo com a função 0x01 (*Coil Status*).

Se desconstruirmos, por exemplo, a mensagem de resposta (TCP) do primeiro pedido (Sensor de Temperatura), [00 01 00 00 00 05 10 04 02 00 E4], e traduzirmos o ultimo *byte* de um valor hexadecimal (0x00E4) para um valor

decimal, obtemos o valor 228, que se multiplicado por 0,1 (valor da resolução), fornece o valor de 22,8°.

Pela Figura 6.8, podemos verificar que foi o valor obtido pelo *script*:

```
vsuser@raspberrypi ~/vigiesystem/modbuspy $ python modbus_tcp.py
Local Server: False
Slave IP: 10.192.10.67 | Slave ID: 16 | Address: 6 | Response: 22.8 |
Slave IP: 10.192.10.67 | Slave ID: 16 | Address: 8 | Response: 55.0 |
Slave IP: 10.192.10.67 | Slave ID: 24 | Address: 4 | Response: False |
Closing Modbus - Connection
ModBus Connection - Closed
local
Exec time0.128780126572
Sleep
```

Figura 6.8: Resposta do Script Python

Na resposta apresentada pela Figura 6.8 podemos também verificar que o tempo de resposta para os 3 pedidos foi de 128 ms, o que dá um tempo de resposta na ordem dos 42,67 ms por pedido. De referir que é acrescentado um ficheiro de registo *log*, "modbus.log" (no CD da Tese) com mais informação gerada pelo *script* Python a cada iteração.

6.4.2 Testes - Automato

De forma a se realizarem os testes, ligou-se o Raspberry ao PLC da Carel (Figura 5.9). Uma vez que o PLC da Carel comunicava apenas por MODBUS RTU, foi aplicado um conversor MODBUS RTU - MODBUS TCP. Este PLC continha duas sondas de temperatura ligadas e um relé que ligava um alarme. Através do seu manual[19], obteve-se a descrição dos registos que continham os valores medidos pelas referidas sondas e do estado do relé. Dessa forma, colocou-se o *script* a correr, obtendo-se o seguinte "output", produzido pela API e representado na Figura 6.9:

```
Program node ./bin/www exited with code 8
Starting child process with 'node ./bin/www'
gdbd0fe19c9a301c4708287780df41a2
[ 'size:3',
  [{"val": 2493.0, "register_id": 1, "coil_val": null}, {"val": 2544.4, "register_id": 2, "coil_val": null}, {"val": 2544.4, "register_id": 3, "coil_val": "OFF"}] ]
POST /ret 200 1090.962 ms - 160
gdbd0fe19c9a301c4708287780df41a2
[ 'size:3',
  [{"val": 2493.0, "register_id": 1, "coil_val": null}, {"val": 2544.4, "register_id": 2, "coil_val": null}, {"val": 2544.4, "register_id": 3, "coil_val": "OFF"}] ]
gdbd0fe19c9a301c4708287780df41a2
[ 'size:3',
  [{"val": 2493.0, "register_id": 1, "coil_val": null}, {"val": 2544.4, "register_id": 2, "coil_val": null}, {"val": 2544.4, "register_id": 3, "coil_val": "OFF"}] ]
POST /ret 200 1192.741 ms - 160
gdbd0fe19c9a301c4708287780df41a2
[ 'size:3',
  [{"val": 43.3, "register_id": 1, "coil_val": null}, {"val": -979.9, "register_id": 2, "coil_val": null}, {"val": -979.9, "register_id": 3, "coil_val": "OFF"}] ]
gdbd0fe19c9a301c4708287780df41a2
[ 'size:3',
```

Figura 6.9: Resposta do PLC

Tal como se apercebe pela Figura 6.9, nos valores circunscritos pelos círculos coloridos, ao correr o *script* para os registos dos PLC, obtivemos a resposta para estes mesmos registos:

Circulo Amarelo Contém informação sobre o Registo 1, que traz os valores da sonda 1 de temperatura do PLC. Ao aceder ao registo, obteve-se o valor 2493, referente à temperatura lida. No entanto, para se apresentar ao utilizador final, é necessário aplicar resolução de 0,01, o que nos dará o valor de 24,98° Celsius.

Circulo Azul Informação Relativa à sonda 2, de onde se obtém o valor 2544,4, ao qual se aplica de igual forma a resolução de 0,01.

Circulo Vermelho Apresenta o valor do estado do relé que aciona o Alarme, neste caso está "OFF".

Este output é depois enviado para plataforma desenvolvida, como se pode ver na Figura 6.10:

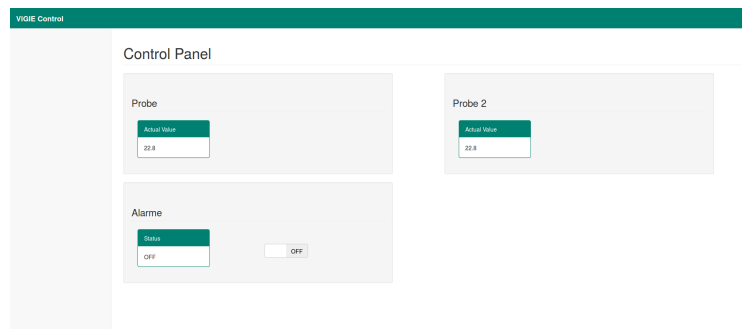


Figura 6.10: Valores apresentados pela plataforma

6.5 Apreciação

Os testes apresentados serviram, apenas e só, como exemplo, não representando todos os testes que foram utilizados para se chegar ao resultado final apresentado nas Figuras 3.6, 6.8, 6.9 e 6.10. Resultado este que demonstra que o objectivo proposto para este projecto foi alcançado.

Desta forma, a integração entre uma plataforma que acede aos valores vindos de uma WSN pode também ela integrar valores vindos de um ou varios PLC's e retirar valor dessa mesma análise contínua do meio.

7

Conclusões

Através do trabalho desenvolvido foi possível retirar várias conclusões. Em primeiro lugar, que as WSN, para além de serem o futuro, são já de uma extrema utilidade nos vários sectores de mercado. Através delas, é possível dotar infraestruturas, relativamente antigas, tais como alguns Hospitais portugueses, de meios de análise e de monitorização, o que lhes confere um valor acrescentado, na medida em que, pode ajudar a colmatar algumas necessidades que vêm já desde a sua concepção.

Integrar as WSN com o poder de automação é um passo necessário e, num mundo em constante mutação, estas podem vir a desempenhar um papel fulcral nas *smart cities*. Aliadas ao poder de actuação, podem mudar completamente o modo como vivemos nas grandes cidades, desde análise e controlo do tráfego rodoviário, à análise e detecção de poluição, até ao controlo das luzes da cidade, podendo-se, desta feita, diminuir a pegada ecológica. Em suma, o poder de análise, dotado de um poder de actuação sobre essa mesma análise, pode originar resultados de valor acrescido.

Constata-se, também, que a centralização de informação nos serviços IT é um bem essencial num mundo empresarial em que, cada vez mais, os índices de produção são obrigados a ser tão altos quanto possível. Torna-se, assim,

essencial, centralizar toda a informação de forma a evitar perdas de tempo desnecessárias. Colocar tudo numa só plataforma, acessível através de qualquer dispositivo, torna-se imprescindível, acabando-se, dessa forma, com a alteração entre softwares e até mesmo com a necessidade de trocar de máquina ou sistema operativo só para poder correr determinada aplicação.

Relativamente à parte técnica deste trabalho, conclui-se que, apesar do protocolo de comunicação MODBUS ser um protocolo antigo e, por isso, com algumas limitações, quando comparado com alguns protocolos mais recentes, a sua simplicidade de implementação, o facto de ser "aberto" e de estar presente em larga escala nos dispositivos de campo, faz dele uma boa escolha quando se pretende integrar equipamentos de diferentes fabricantes a uma larga escala. Aliás, quando complementado com as tecnologias utilizadas neste projecto, como o *NodeJs* e ligações a base de dados, consegue-se produzir ferramentas bastantes úteis e flexíveis.

A satisfação na conclusão deste projecto é ainda maior quando este é implementado na "vida real" e lança as pedras para o desenvolvimento de uma nova plataforma dentro da empresa, plataforma esta que incluirá, também, módulos de controlo/actuação graças ao trabalho desenvolvido. Isto porque, já desde a minha entrada dentro da mesma, se constatava a ausência de uma plataforma capaz de actuar perante os valores adquiridos, pelo que, fazer parte da sua evolução é, obviamente, algo de salutar e motivo de orgulho.

Por fim, de referir também que o trabalho desenvolvido será ainda aplicado no Centro Hospitalar Lisboa Norte, onde fará uso de toda a sua potencialidade, controlando o sistema de bombeamento e aquecimento da água dos circuitos sanitários, encontrando-se neste momento em fase de estudo e implementação e sendo espectável que, até ao final do ano, todo o sistema de bombeamento esteja já a ser controlado pela plataforma ViGIE, que, por sua vez, fará uso de todo o trabalho relatado.

Bibliografia

- [1] MOXA, *RemoteIO* ("www.moxa.com", 2016).
- [2] SmartProcess, *SmartRail* ("www.smartprocess.co.uk/mid-approved-100a-three-phase-multifunction-electricity-meter-pulse-modbus-smartrail-x835-100", 2016).
- [3] Sparkfun, *RS485 Shield V3* ("www.sparkfun.com/products/13706", 2016).
- [4] Tektroniks, *Wireless DataCentre* (www.tek-troniks.com/site/wireless-monitoring-systems/, 2016).
- [5] S. Tennina *et al.*, *IEEE 802.15. 4 and ZigBee as enabling technologies for low-power wireless systems with quality-of-service constraints* (Springer Science & Business Media, 2013).
- [6] L. Jovanovic, "The Anatomy of an M2M Solution: Hardware, the First Step to Connected Machines," <http://www.etherios.com/blog/thoughtleadership/the-anatomy-of-an-m2m-solution-hardware-the-first-step-to-connected-machines/>, 2013.
- [7] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors* **12**, 1625–1647 (2012).
- [8] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications : Remote large-scale environments," In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pp. 1–7 (2009).
- [9] Siemens, "PLC-Siemens," www.siemens.com, 2015.
- [10] Fultek, "SCADA Software," www.fultek.com.tr, 2015.
- [11] Z. Jovic, "Automation Pyramide," <http://www.mikromotor.rs/en/iba-automatika>, 2015.
- [12] K. M. Sim, "Agent-Based Cloud Computing," *Services Computing, IEEE Transactions on* **5**, 564–577 (2012).

- [13] G. Kulkarni, P. Chavan, H. Bankar, K. Koli, and V. Waykule, “A new approach to software as service cloud,” In *Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on*, pp. 196–199 (2012).
- [14] Z. Gargasz, “OSI model how Internet works,” <http://www.gargasz.info/osi-model-how-internet-works/>, 2013.
- [15] M. A. P. Specification, “V1. 1b,” 2006.
- [16] I. Modbus, “Modbus messaging on tcp/ip implementation guide v1. 0b,” North Grafton, Massachusetts (www.modbus.org/specs.php) (2006).
- [17] Carel, *Wireless sensors* (<http://www.carel.com/documents/10191/0/+0300030EN/c70a24cc-f069-40fd-b867-65aa47937a1b?version=1.0>, 2015).
- [18] Eltek, “Products,” <http://www.eltekdataloggers.co.uk/transmitters.html>.
- [19] Carel, *UltraCella* (<http://www.carel.com/documents/>, 2015).
- [20] Ubuntu, “Pangolin Server,” www.ubuntu.com, 2015.
- [21] V. Solutions, “Plataforma Vigie,” www.vigie.pt, 2015.
- [22] J.-P. Thomesse, “Fieldbus technology in industrial automation,” *Proceedings of the IEEE* **93**, 1073–1101 (2005).
- [23] M. Felser, “The fieldbus standards: History and structures,” Technology Leadership Day (2002).
- [24] M. Rostan, “Industrial Ethernet Technologies: Overview,” In , 1 (2008).
- [25] Profibus, “PROFINET The leading Industrial Ethernet Standard,” In , 1 (www.profibus.com/nc/pi.../regional.../13722/, 2015).
- [26] E. ETSI, “300 220-1,” European Standard (Telecommunications Series) 1 (2000).
- [27] Validair, *About Validair* (“www.validair.com/about-validair-group/”, 2016).
- [28] HGO, *Hospital Garcia da Orta* (www.hgo.pt, 2016).
- [29] S. Hotels, *Hotel Lisboa Sheraton* (www.sheratonlisboa.com/, 2016).
- [30] Validair, *Validair Group* (www.validair.com, 2016).
- [31] Tektronix, *Tektronix* (www.tek.com, 2016).

- [32] I. S. Amiya Nayak, in *Wireless Sensor and Actuator Networks*, M. Stojmenovic, ed., (John Wiley and Sons, Inc., Hoboken, New jersey, United States, 2010).
- [33] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer networks* **52**, 2292–2330 (2008).
- [34] D. Chen, M. Nixon, and A. Mok, *Why WirelessHART* (Springer, 2010).
- [35] Z. Shelby and C. Bormann, *6LoWPAN: The wireless embedded Internet* (John Wiley & Sons, 2011), Vol. 43.
- [36] R. Heydon, *Bluetooth low energy: the developer’s handbook* (Prentice Hall, 2012).
- [37] J. Shandle, “Bluetooth Low Energy: Chips, Use Profiles and Development Tools Coming Online Fast,” *Electronic Products* (2011).
- [38] D. I. Inc, “Ant Applications,” www.thisisant.com/consumer/ant-101/what-is-ant (2015).
- [39] ANT, *Antware II users guide* (www.thisisant.com/resources/antware-ii-users-guide, 2009).
- [40] C. Herzog, “Creating value with ZigBee networks,” *Industrial embedded system resource guide, networking: technology* **67**, 31–33 (2005).
- [41] K. Bender, *Profibus: the fieldbus for industrial automation* (Prentice-Hall, Inc., 1993).
- [42] B. C. CAN, “Specification-Version 2.0 Part A, R,” Bosch Gmbh, Germany (1991).
- [43] I. E. Commission *et al.*, “IEC 61158: Digital data communications for measurement and control-Fieldbus for use in industrial control systems,” 2003.
- [44] L. Columbus, “Roundup Of Cloud Computing Forecasts And Market Estimates,” IBM (2015).
- [45] M. Miller, *Cloud computing: Web-based applications that change the way you work and collaborate online* (Que publishing, 2008).
- [46] G. Paltoglou, M. Salampasis, and M. Satratzemi, “A Comparison of Centralized and Distributed Information Retrieval Approaches,” In *Informatics, 2008. PCI '08. Panhellenic Conference on*, pp. 21–25 (2008).

- [47] F. Osterlind, E. Pramsten, D. Roberthson, J. Eriksson, N. Finne, and T. Voigt, “Integrating building automation systems and wireless sensor networks,” In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pp. 1376–1379 (2007).
- [48] S. T. Bushby, “BACnet TM: a standard communication infrastructure for intelligent buildings,” *Automation in Construction* **6**, 529–540 (1997).
- [49] Schiller and Voigt, *Scatterweb low power sensor nodes and energy aware routing* (HICSS, 2005).
- [50] D. Flanagan, *JavaScript: the definitive guide* (“O’Reilly Media, Inc.”, 2006).
- [51] O. Nodejs, *Node.js: Documentation* (“<https://nodejs.org/api/documentation.html>”, 2016).
- [52] O. NPM, *Basic Auth: Documentation* (“<https://docs.npmjs.com/>”, 2016).
- [53] Bootstrap, *Getting started* (“<http://getbootstrap.com/getting-started/>”, 2016).
- [54] G. Van Rossum and F. L. Drake, *Python Tutorial, Release 2.2. 1* (Citeseer, 2002).
- [55] G. Collins, *PyModbus documentation* (“pymodbus.readthedocs.org”, 2009).
- [56] D. V. Federico Di Gregorio, “Psychopg2 Documentation,” (2015).
- [57] T. P. G. D. Group, *PostgreSQL Documentation* (“www.postgresql.org/docs/”, 2016).

Appendices

A

Anexo

Serve o presente anexo para apresentar o guia de configuração utilizado para configurar o Raspberry Pi II model B.

A.1 Requisitos de Sistema

1. RaspBerry
2. Pi II Model B;
3. MicroSD;
4. Lexar 16 GB type: HC;
5. OS;
6. Raspbian;
7. Jessie Lite Linux (link: <https://www.raspberrypi.org/downloads/raspbian/>);
8. SSH;
9. FTP;
10. Postreges;
11. apache2;
12. PHP;

13. Telnet;
14. Htop;
15. phpPGAdmin;
16. Pythondev;
17. Pymodbus;
18. Psycopg2;
19. socketIOclient;
20. Node.js;
21. Weavedconnectd;
22. Crontab.

A.2 Instalação SO

After you get the distro, please follow the next steps:

1. Run `df -h` to see what devices are currently mounted.
2. If your computer has a slot for SD cards, insert the card. If not, insert the card into an SD card reader, then connect the reader to your computer;
3. Run `df h` again. The new device that has appeared is your SD card. The left column gives the device name of your SD card? it will be listed as something like `/dev/mmcblk0p1` or `/dev/sdd1` . The last part (p1 or 1 respectively) is the partition number but you want to write to the whole SD card, not just one partition. You therefore need to remove that part from the name, getting, for example, `/dev/mmcblk0` or `/dev/sdd` as the device name for the whole SD card. Note that the SD card can show up more than once in the output of `df`? it will do this if you have previously written a Raspberry Pi image to this SD card, because the Raspberry Pi SD images have more than one partition;
4. Now that you've noted what the device name is, you need to unmount it so that files can't be read or written to the SD card while you are copying over the SD image;
5. Run `umount /dev/sdd1` , replacing `sdd1` with whatever your SD card's device name is (including the partition number);

6. If your SD card shows up more than once in the output of `df` due to having multiple partitions on the SD card, you should unmount all of these partitions;
7. In the terminal, write the image to the card with the command below, making sure you replace the input file `if=` argument with the path to your `.img` file, and the `/dev/sdd` in the output file `of=` argument with the right device name. This is very important, as you will lose all data on the hard drive if you provide the wrong device name. Make sure the device name is the name of the whole SD card as described above, not just a partition of it? for example, `sdd`, not `sdds1` or `sddp1`, and `mmcblk0`, not `mmcblk0p1`.
8. Example: `umount /dev/sdd1 -f; dd bs=4M if=20160318raspbianjessie.img of=/dev/sdd;`
9. Run `sync`; this will ensure the write cache is flushed and that it is safe to unmount your SD card;

A.3 Módulos Necessários

Passos necessários para a instalação dos módulos de Python:

1. Postgres: `sudo apt-get install -y postgresql, sudo apt-get install -y postgresqlserverdev9.4;`
2. Apache2: `sudo apt-get install -y apache2`
3. PHP: `sudo apt-get install -y php5 php5-pgsql`
4. Python Dev: `sudo apt-get install -y python-dev python-pip`
5. Psycopg2: `sudo pip install psycopg2`
6. Pymodbus: `sudo pip install pymodbus`
7. SocketIO: `sudo pip install SocketUI-Client`
8. Nodejs: `curl -sL https://deb.nodesource.com/setup5x — sudo -E Bash - sudo apt-get install -y nodejs`
9. Weavedconnected: `sudo apt-get install weavedconnected`
10. Crontab: `sudo apt-get install gnome-schedule`

A.4 Ficheiros Necessários

De forma a instalação ficar completa é necessário conter uma pasta com os seguintes ficheiros Python:

1. modbus_tcp.py;
2. sync.py
3. delete_old_measures;
4. postgresdb.py

A.5 Configuração V3-Shield Module - RTU

De forma a prevenir que o Raspberry envie informação pelas portas serie ao iniciar, é necessário colocar a seguinte linha no ficheiro `"/boot/cmdline.txt"`:

1. `console=ttyAMA0,115200 kgdboc=ttyAMA0,115200`