

2015

Instituto Politécnico de Coimbra

INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

Soluções de Comunicação para Controlo e Monitorização de Motores

MESTRADO EM ENGENHARIA ELETROTÉCNICA

AUTOR | David Emanuel Ferreira Morais Apóstolo

ORIENTADORES |

Professor Doutor Victor Daniel Neto dos Santos

Professor Doutor Fernando José Teixeira Estêvão Ferreira

Coimbra, junho 2015

Soluções de Comunicação para Controlo e Monitorização de Motores

Trabalho de Projeto apresentado para a obtenção do grau de Mestre em
Engenharia Eletrotécnica – Área de Especialização em Automação e
Comunicações em Sistemas Industriais

Autor

David Emanuel Ferreira Morais Apóstolo

Orientadores

Doutor Victor Daniel Neto dos Santos

Professor Adjunto ISEC

Doutor Fernando José Teixeira Estêvão Ferreira

Professor Auxiliar DEEC - FCTUC

Coimbra, junho 2015

AGRADECIMENTOS

A todas as pessoas que, ao longo do Mestrado em Engenharia Eletrotécnica (MEE) área de especialização em Automação e Comunicações em Sistemas Industriais, direta ou indiretamente, me ajudaram a cumprir os meus objetivos e a realizar mais esta etapa na minha formação académica, quero agradecer o estímulo e o apoio dado.

Ao orientador Professor Doutor Victor Daniel Neto dos Santos, expresso o meu agradecimento pela orientação, apoio e sentido de responsabilidade que me incutiu em todas as fases do projeto, que em muito me estimularam o desejo e vontade de fazer mais e melhor. Agradeço também a sua disponibilidade e boa disposição com que sempre me presenteou.

Ao Professor Doutor Fernando José Teixeira Estêvão Ferreira pela co-orientação neste projeto e por ter proporcionado as condições necessárias para a elaboração da minha dissertação e por permitir a minha integração num estimulante e prestigiado projeto.

Aos meus colegas da Optisigma, José Guarino, Miguel Oliveira e Ricardo Ferreira, o meu muito obrigado pela vossa amizade, companheirismo e entreaajuda, fatores imprescindíveis na elaboração desta dissertação permitindo que cada novo dia fosse encarado com particular motivação.

Aos meus amigos de Mestrado, muito obrigado pela partilha de bons momentos

À Marlene, um agradecimento muito especial pelo carinho e apoio incondicional, pelas palavras de conforto e pela transmissão de confiança e força, em todos os momentos. Por tudo, o meu enorme obrigado!

À minha família, em especial aos meus pais e irmão, agradeço por depositarem em mim a vossa confiança, por sempre acreditarem em mim e no que faço e por todos os ensinamentos de vida. É com esperança que, nesta etapa que termino, possa de alguma forma retribuir e compensar todo carinho, apoio e dedicação que, consistentemente, me ofereceram. A eles, dedico este trabalho.

RESUMO

O homem na sua busca incessante pela inovação criou um dispositivo, designado por motor elétrico, cuja principal função compreende a conversão de energia elétrica em energia mecânica. O motor elétrico é sem dúvida uma das maiores invenções da história, visto que impulsionou o processo de industrialização e alterou radicalmente o modo de vida das pessoas.

De entre os diversos tipos de motores elétricos existentes, o motor de indução trifásico de rotor em gaiola de esquilo é indiscutivelmente o mais utilizado a nível mundial em aplicações industriais. A popularidade deste motor em particular advém da sua robustez, simplicidade, reduzido custo e excelente rendimento. O motor de indução apresenta contudo algumas desvantagens que se prendem com o decréscimo de rendimento em cargas de baixo valor. Para colmatar as limitações enunciadas surgiram algumas soluções tecnológicas, designadamente os contadores, os *soft-starters* e o *InSwitch*.

O *InSwitch* pretende ser uma alternativa aos demais dispositivos de controlo de motores já existentes no mercado. Para além de controlar o motor de indução, este dispositivo pode disponibilizar informação pertinente relativa ao seu funcionamento. Assim sendo, utilizaram-se as capacidades do *InSwitch* para desenvolver uma ferramenta intuitiva e de fácil acesso de forma a permitir aos técnicos de manutenção a visualização de toda a informação relevante relativa ao seu estado atual de funcionamento.

Nesta dissertação são apresentadas duas soluções tecnológicas distintas, as quais foram desenvolvidas com o objetivo mencionado. Uma das soluções oferece ao utilizador a possibilidade de controlar e analisar os dados em tempo real do motor que pretende monitorizar, sem que para isso necessite de equipamento de medida de motores de indução, o qual é muito dispendioso. Através de qualquer dispositivo móvel equipado com o *Sistema Operativo Android* e *Bluetooth* o utilizador pode aceder a toda a informação do motor sem necessidade de parar todo o sistema de produção. A segunda solução desenvolvida permite que os motores comuniquem, através de uma ligação de rádio frequência, com um dispositivo central. O dispositivo central dispõe de um processador *ARM Cortex-M4*, o qual é responsável pelo processamento de toda a informação recebida dos motores e de a reenviar para uma base de dados. Por fim, toda a informação armazenada na base de dados pode ser consultada e analisada, em tempo real, pelo utilizador através de um *website* desenvolvido para o efeito.

Palavras-chave: Motor de indução, *InSwitch*, *Bluetooth*, rádio frequência, *ARM Cortex-M4*, monitorização, base de dados, *website*, *Android*.

ABSTRACT

Mankind in his constant search for innovation has created a device, called the electric motor, whose main function includes the conversion of electrical energy into mechanical energy. The electric motor is undoubtedly one of the greatest inventions in history, as it boosted the industrialization process and radically changed people's way of life.

Among the various existing types of electric motors, the three-phase squirrel cage induction motor is arguably the most widely used worldwide in industrial applications. The popularity of this specific motor comes from its robustness, simplicity, low cost and excellent performance. The induction motor has, however, some drawbacks that are related to the income decrease in low-value cargo. To overcome these limitations some technological solutions have emerged, including contactors, the soft-starters and InSwitch.

The InSwitch intended to be an alternative to other existing motor control devices on the market. Apart from controlling the induction motor, this device can provide information relevant to its operation. Therefore, we used the InSwitch capabilities to develop an intuitive tool and easy to access in order to allow maintenance technicians to visualize all relevant information relating to its current state of operation.

In this thesis are presented two different technological solutions, which were developed for the purpose mentioned. One solution offers the user the ability to control and analyze data in real-time motor that you want to monitor, without this requires measuring equipment induction motors, which is very expensive. Through any mobile device equipped with Android OS and Bluetooth you can access all the motor information without the need to stop the entire production system. The second solution developed allows motors to communicate, via radio frequency with a central device. The central device has an ARM Cortex-M4 processor, which is responsible for processing all the information received from the motor and to resend a data base. Finally, all the information stored in the database can be queried and analyzed in real time by the user through a website developed for this purpose.

Keywords: Induction motor, InSwitch, Bluetooth, radio frequency, ARM Cortex-M4, monitoring, database, website, Android.

ÍNDICE

AGRADECIMENTOS	i
RESUMO	iii
ABSTRACT	v
ÍNDICE	vii
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABELAS	xiii
ABREVIATURAS	xv
SIMBOLOGIA	xix
Capítulo 1 INTRODUÇÃO	1
1.1 Enquadramento	1
1.2 Motor de Indução Trifásico	3
1.2.1 Constituição do motor de indução trifásico	3
1.2.2 Tipos de motores de indução trifásicos	4
1.2.3 Princípio de Funcionamento	5
1.3 Impacto Energético dos Motores de Indução Trifásicos	5
1.4 Controlo de Motores	7
1.4.1 Contatores	8
1.4.2 Smart Switch	9
1.4.3 SinuMEC	9
1.4.4 Soft-starters	10
1.4.5 InSwitch	11
1.5 Objetivos	12
1.6 Metodologia	12
1.7 Estrutura da Dissertação	13
Capítulo 2 ESTADO DE ARTE DOS SISTEMAS DE COMUNICAÇÃO INDUSTRIAL	15
2.1 Introdução	15
2.2 Comunicações Industriais	15
2.3 Sistemas Cablados	16
2.3.1 Fieldbus	16
2.3.2 ModBus	19
2.3.3 Profibus	21
2.3.4 Ethernet TCP/IP	23
2.3.5 PLC	25
2.3.6 Profinet	28
2.3.7 AS-Interface	29
2.3.8 DeviceNet	31
2.3.9 CANopen	33
2.4 Sistemas Sem Fios	35
2.4.1 Principais tipos de redes sem fios	35
2.4.2 WI-FI	37
2.4.3 BLUETOOTH	38
2.4.4 ZIGBEE	39
2.4.5 WIRELESS HART	41
2.4.6 ISA100	43
2.4.7 WISA	44
2.4.8 Comparativo	45
2.5 Conclusão	45
Capítulo 3 SOLUÇÃO TECNOLÓGICA SELECIONADA	47
3.1 Introdução	47
3.2 Unidades Microcontroladores	47
3.2.1 Arduino	48
3.2.2 Raspberry Pi	50
3.2.3 STM 32-bit ARM	52
3.2.4 BeagleBoard	55

3.2.5	pcDuino	56
3.2.6	TinkerForge	57
3.3	Soluções tecnológicas selecionadas	59
3.3.1	Módulo Bluetooth HC-05	59
3.3.2	Módulo RFM12b	60
3.3.3	Módulo nRF24L01+	61
3.3.4	Placa STM32F4DISCOVERY	62
3.3.5	Placa Base STM32F4DIS-BB	62
3.4	Ferramentas de Software de desenvolvimento	63
3.4.1	CadSoft EAGLE PCB software de design	63
3.4.2	Keil μ Vision Integrated Development Environment	63
3.4.3	MIT App Inventor 2	64
3.4.4	Visual Studio	64
3.5	Tecnologia e Gestão de Informação Digital	64
3.5.1	Tecnologias e ferramentas utilizadas	65
3.6	Conclusão	69
Capítulo 4	IMPLEMENTAÇÃO	71
4.1	InSwitch	71
4.2	Solução de comunicação desenvolvida	72
4.2.1	Protótipo Arduino com o módulo Bluetooth HC-05:	72
4.2.2	Protótipo Arduino com o módulo RFM12b:	74
4.2.3	Protótipo STM32F4 com o módulo HC-05:	76
4.2.4	Protótipo STM32F4 com o módulo nRF24L01+	78
4.2.5	Protótipo STM32F4 com placa base STM32F4DIS-BB:	79
4.3	Protocolo de comunicação desenvolvido	82
4.3.1	Rádio Frequência	82
4.3.2	Bluetooth	84
4.3.3	Comunicação Série	85
4.4	Arquitetura da aplicação <i>web</i>	86
4.5	Base de dados desenvolvida	88
4.5.1	Análise do projeto	88
4.5.2	Diagrama Entidade – Relacionamento	90
4.5.3	Diagrama físico	91
4.5.4	Implementação da base de dados	92
4.6	Website desenvolvido	94
4.6.1	Processo de Desenvolvimento do Website	94
4.6.2	Conceção do Website	96
4.7	Ambiente de testes	103
4.8	Conclusão	105
Capítulo 5	CONCLUSÕES	107
5.1	Desenvolvimentos futuros	108
Capítulo 6	BIBLIOGRAFIA	109

ÍNDICE DE FIGURAS

Figura 1 – O primeiro motor elétrico – Michael Faraday [2].....	1
Figura 2 – Motor de indução trifásico [4].....	3
Figura 3 – Consumo de energia dos motores no setor industrial na EU [6].....	6
Figura 4 – <i>Smart-Switch</i> desenvolvido pelo ISEC [10].	9
Figura 5 – Exemplo de um SinuMEC [11].....	9
Figura 6 – Dispositivo <i>InSwitch</i>	11
Figura 7 – Níveis da pirâmide de automação [14].....	16
Figura 8 – Exemplos de topologia Fieldbus [18].....	19
Figura 9 – Exemplo de uma rede com diferentes vertentes <i>ModBus</i> [22].....	20
Figura 10 – Exemplo de rede PROFIBUS [25].	23
Figura 11 – Topologia em estrela [21].	25
Figura 12 – Topologia barramento [21].....	25
Figura 13 – Topologia de uma rede PLC [29].	27
Figura 14 – Exemplo da topologia <i>Profinet</i> [35].....	29
Figura 15 – Exemplo de uma topologia AS-Interface [22].....	31
Figura 16 – Imagem ilustrativa da topologia <i>DeviceNet</i> [36].	33
Figura 17 – Exemplo de utilização da topologia CAN [37].....	35
Figura 18 – Análise de mercado das tecnologias wireless [38].	36
Figura 19 – Divisão das redes <i>wireless</i> . [76].....	36
Figura 20 – Rede Wi-Fi [41].	37
Figura 21 – Possíveis dispositivos numa rede Bluetooth [44].	39
Figura 22 – Exemplo de uma rede ZigBee [46].....	40
Figura 23 – Exemplos de topologias de uma rede ZigBee [47].	41
Figura 24 – Possíveis topologias de uma rede WirelessHART [50].....	42
Figura 25 – Estrutura de uma rede WirelessHART [49].....	43
Figura 26 – Exemplo de uma rede ISA100 [52].	43
Figura 27 – Visão geral e dispositivos WISA [55].	44
Figura 28 – Arduíno UNO.....	48
Figura 29 – Ambiente de desenvolvimento do Arduino.	49
Figura 30 – Raspberry Pi modelo B+.	50
Figura 31 – SO Raspbian interface [60].	52
Figura 32 – Interface do SO RaspBMC [60].	52

Figura 33 – Comparação dos desempenhos dos diversos microcontroladores [61].	53
Figura 34 – Desempenho dos processadores da família STM32 da STMicroelectronics [62].	54
Figura 35 – Módulo HC-05 e respetivo esquemático.	60
Figura 36 – Módulo RFM12b e respetivo esquemático.	61
Figura 37 – Módulo nRF24L01+ e respetivo esquemático.	61
Figura 38 – Placa <i>STM32F4DISCOVERY</i> [69].	62
Figura 39 – Placa Base <i>STM32F4DIS-BB</i> . [70].	63
Figura 40 – Esquema de ligação do teste piloto Arduino Uno e o módulo HC-05.	73
Figura 41 – Aplicação “.ino” e interface para Android.	74
Figura 42 – Esquema de ligação do teste piloto Arduino Uno e o módulo RFM12b.	74
Figura 43 – <i>Shield</i> desenvolvido para o Arduino Uno.	75
Figura 44 – Esquema de ligação da placa <i>STM32F4DISCOVERY</i> e o módulo HC-05.	76
Figura 45 – Código responsável por enviar informação para o módulo HC-05.	77
Figura 46 – Aplicação <i>Android</i> para controlo e monitorização do motor por <i>Bluetooth</i> .	77
Figura 47 – Esquema de ligação da placa <i>STM32F4DISCOVERY</i> e o módulo nRF24L01+.	79
Figura 48 – Código responsável por formatar a mensagem a enviar por rádio frequência.	79
Figura 49 – DevKit407, conjunto das placas <i>STM32F4DISCOVERY</i> e <i>STM32F4DIS-BB</i> .	80
Figura 50 – Página Home	81
Figura 51 – Exemplo de uma rede local de monitorização de motores.	82
Figura 52 – Parte da rede de monitorização – comunicação rádio frequência.	83
Figura 53 – Fluxograma do processo de comunicação entre o <i>InSwitch</i> e o dispositivo central.	83
Figura 54 – Parte da rede de monitorização – Comunicação <i>Bluetooth</i> .	84
Figura 55 – Parte da rede de monitorização – Comunicação série.	85
Figura 56 – Arquitetura <i>web</i> .	87
Figura 57 – Conjunto de tabelas desenvolvidas no MySQL Workbench.	89
Figura 58 – Diagrama entidades relacionamento.	91
Figura 59 – Diagrama físico criado no MySQL Workbench.	91
Figura 60 – Diagrama do processo de desenvolvimento.	95
Figura 61 – Estrutura de navegação.	97
Figura 62 – Página de autenticação de utilizador.	98
Figura 63 – Página de registo de utilizador.	98
Figura 64 – Página de inicial, com a grelha dinâmica de motores.	99
Figura 65 – Página do motor.	100
Figura 66 – Página que permite ver a chapa característica do motor.	101

Figura 67 – Janela popup – página Tempo Real.....	101
Figura 68 – Janela popup – página Cumulativo.....	102
Figura 69 – Janela popup – página Info Extra.	102
Figura 70 – InSwitch instalado na caixa de terminais do motor e bancada de testes, respetivamente.	103
Figura 71 – Placa STM32F4 equipado com o módulo Bluetooth HC-05.	104
Figura 72 – Módulo HC-05 incorporado na placa de controlo do InSwitch.	104
Figura 73 – Placa STM32F4 equipado com o módulo nRF24L01+.	105

ÍNDICE DE TABELAS

Tabela 1 – Consumo de energia elétrica por parte dos motores elétricos em diversos países.	6
Tabela 2 – Características da camada física da rede Foundation Fieldbus.....	18
Tabela 3 – Comparação de <i>Narrowband</i> PLC e <i>Broadband</i> PLC [28].....	26
Tabela 4 – Classificação da tecnologia PLC com base na velocidade de transmissão [27].	27
Tabela 5 – Taxa de comunicação selecionável [36].	32
Tabela 6 – Relação distância/débito suportadas [37].....	35
Tabela 7 – Características da norma 802.11.	38
Tabela 8 – Tabela comparativa de diversas normas IEEE sem fios.....	45
Tabela 9 – Características físicas do Arduino Uno [56].	49
Tabela 10 – Características físicas dos modelos Raspberry Pi [57] [58].	51
Tabela 11 – Comparação das placas da família STM32 da STMicroelectronics [63].....	54
Tabela 12 – Comparação das placas BeagleBoard [64].....	56
Tabela 13 – Características de <i>software</i> BeagleBoard.....	56
Tabela 14 – Características físicas do pcDuino [65].....	57
Tabela 15 – Características de <i>software</i> do pcDuino.	57
Tabela 16 – Características de <i>software</i> do <i>TinkForge</i>	59
Tabela 17 – Características do módulo Bluetooth HC-05 [67].....	60
Tabela 18 – Especificações do módulo nRF24L01 [68].	62
Tabela 19 – Parâmetros monitorizados pelo <i>InSwitch</i>	71
Tabela 20 – Parâmetros calculados pelo <i>InSwitch</i>	71
Tabela 21 – Comandos de configuração do módulo HC-05 [73].....	73
Tabela 22 – Constituição das mensagens “ <i>Ping</i> ”, <i>done</i> e <i>dados</i>	84
Tabela 23 – Constituição da mensagem enviada para o módulo HC-05.....	85
Tabela 24 – Constituição da mensagem enviada pelo dispositivo central para o PC.	85
Tabela 25 – Tabela com todos os atributos presentes na base de dados.	90
Tabela 26 – Dados inseridos na tabela atributos.....	92

ABREVIATURAS

ASCII – American Standard Code for Information Interchange

BD – Base de Dados

BPSK – Binary Phase Shift Keying

CGI – Common Gateway Interface

CIP – Common Industrial Protocol

CRC – Cyclic Redundancy Check

CSMA/CD – Carrier Sense Multiple Access/Collision Detection

CSS – Cascading Style Sheets

DCS – Distributed Control Systems

DESA – Department of Economic and Social Affairs

DSSS – Direct Sequence Spread Spectrum

ETSI – European Telecommunications Standards Institute

FCC – Federal Communications Commission

FFD – Full Function Device

FH/TDD – Frequency Hopping/Time-Division Duplex

FH-CDMA – Frequency Hopping - Code-Division Multiple Access

FHSS – Frequency-Hopping Spread Spectrum

FIC – Factory Instrumentation Protocol

FSK – Frequency Shift Keying

GCC – GNU Compiler Collection

GDS – Generic Data Slave

GFSK – Gaussian Frequency Shift Keying

HSE – High Speed Ethernet

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

I²C – Inter-Integrated Circuit

IDE – Integrated Development Environment

IEC – International Electrotechnical Commission

ISA – International Society of Automation

ISM – Industrial, Scientific and Medical

MBWA – Mobile Broadband Wireless Access

MI – Motores de Indução

MIT – Motor de Indução Trifásico

MTBF – Mean Time Between Failures

MTTR – Mean Time To Repair

NCSA – National Center of Supercomputing Applications

ODVA – Open DeviceNet Vendor Association

OFDM – Orthogonal Frequency Division Multiplexing

PCB – Printed Circuit Board

PHP – Personal Home Page

PLC – Power Line Communication

QAM – Quadrature Amplitude Modulation

RFD – Reduced Function Device

RTU – Remote Terminal Unit

SCR – Silicon-Controlled Rectifier

S-FSK – Spread- Frequency Shift Keying

SGDB – Sistema de Gestão de Bases de Dados

SIG – Special Interest Group

SinuMEC – Sinusoidal Motor Efficiency Controller

SO – Sistema Operativo

SOC – System on a Chip

SQL – Structured Query Language

SSI – Server Side Includes

TDMA – Time Division Multiple Access

UNSD – United Nations Statistics Division

UWB – Ultra Wide Band

VEV – Variador Eletrónico de Velocidades

WiMAX – WorldWide Interoperability for Microwave Access

WISA – Wireless Interface for Sensor and Actuators

WLAN – Wireless Local Area Network

WMAN – Wireless Metropolitan Area Network

WPAN – Wireless Personal Area Network

WWAN – Wireless Wide Area Network

XAMPP – Operating System + Apache + MySQL + PHP + Perl

SIMBOLOGIA

P	Potência elétrica
n_s	Velocidade síncrona (rpm)
f	Frequência da corrente (Hz)
p	Número de pares de pólos do motor
S	Escorregamento (deslizamento)
n	Velocidade real (rpm)
P_n	Potência nominal

Capítulo 1

INTRODUÇÃO

1.1 Enquadramento

A história dos motores elétricos começa em 1600 com o cientista inglês William Gilbert [1] ao publicar em Londres a obra intitulada de *Magnete*, descrevendo a força de atração magnética. Contudo, o fenómeno da eletricidade estática já tinha sido observado antes pelo grego Tales, em 641 a.c., ao friccionar uma peça de âmbar com um pano, ele verificou que esta adquiria a propriedade de atrair corpos leves, como pelos, penas, cinzas, etc.

Hans Christian Oersted, um físico dinamarquês, ao fazer experiências com correntes elétricas, verificou em 1820 que a agulha magnética de uma bússola era desviada da sua posição quando esta passava perto de um condutor no qual circulava corrente elétrica [1]. Esta observação permitiu a Oersted reconhecer a íntima relação entre o magnetismo e a eletricidade, dando assim, o primeiro passo em direção ao desenvolvimento do motor elétrico. O francês André Marie Ampère, em 1821, completou a experiência de Oersted, criando a conhecida “lei da mão direita”.

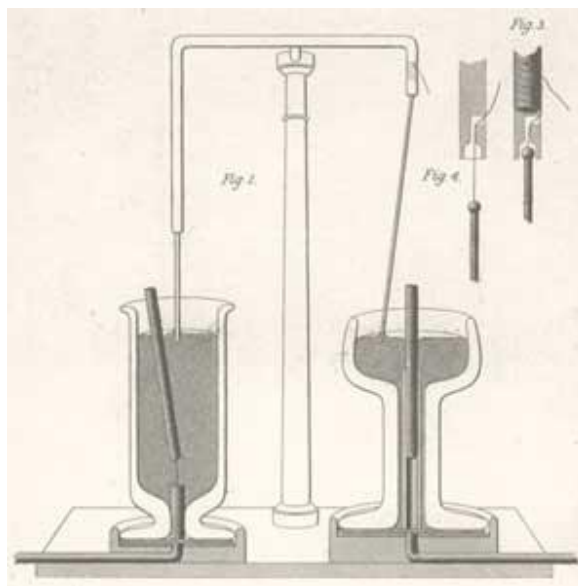


Figura 1 – O primeiro motor elétrico – Michael Faraday [2].

Em 1825, o sapateiro inglês William Sturgeon que estudava eletricidade nas suas horas vagas e o cientista inglês Michael Faraday, inspirados pelas descobertas de Oersted e Ampère foram os responsáveis pelos últimos passos rumo à construção do motor elétrico. Sturgeon constatou que um núcleo de ferro envolto num fio condutor elétrico transformava-se num íman quando se aplicava uma corrente elétrica, observando também que a força do íman terminava assim que a corrente fosse interrompida. Na Figura 1 é demonstrada uma das experiências que levou Michael Faraday a descobrir a indução eletromagnética, provando que Tales há quase dois mil anos atrás estava certo. Assim foi inventado o eletroímã, que seria de fundamental importância na construção de máquinas elétricas girantes. Já no ano de 1833 [2], o inglês W. Ritchie inventou o comutador, construindo um pequeno motor elétrico onde o núcleo de ferro enrolado girava em torno de um íman permanente.

O primeiro motor de indução surge no verão de 1883 em Estrasburgo, França pelo cientista e inventor sérvio-americano *Nikola Tesla* (1856-1946) [1]. Em 1888 regista a patente do seu primeiro motor de indução. Considerada pela *Tesla Memorial Society of New York* o “*Tesla’s AC Induction Motor is one of the 10 greatest discoveries of all time*” [3].

Contudo um dos problemas dos motores era o rendimento da transformação da energia elétrica em mecânica, e portanto grandes esforços foram feitos para alcançar o objetivo de uma máquina que fosse capaz de fazer essa transformação com o menor desperdício de energia possível. Assim chegou-se ao motor de indução com rotor em gaiola de esquilo.

O motor de indução funciona normalmente com uma velocidade constante, que pode variar ligeiramente em função da carga mecânica aplicada ao eixo. Por se tratar de uma máquina simples, robusta e de baixo custo é o motor mais utilizado na indústria.

Desde então foi desenvolvida uma grande variedade de motores, a seleção de um motor para uma aplicação específica determina o seu custo inicial e também o seu custo de operação. A escolha de um motor de potência insuficiente pode induzir a um funcionamento inadequado do sistema a ele acoplado resultando numa baixa produtividade assim como avarias prematuras no motor. Por outro lado, a escolha de um motor de potência acima do necessário, terá como consequência um maior investimento inicial, apresentará um rendimento mais baixo, terá um fator de potência baixo e uma vida útil reduzida. Do ponto de vista económico e técnico todas estas consequências são indesejáveis. Assim podemos concluir que a escolha acertada de motores para uma aplicação específica assume grande importância na atualidade, uma vez que é crescente o custo e da energia elétrica assim como também pela crise económica que se faz sentir nestes últimos anos em diversos países.

1.2 Motor de Indução Trifásico

Ao contrário dos motores convencionais de corrente contínua em que o estator e o rotor necessitam de alimentação, nos motores assíncronos só é necessário alimentar o estator, o rotor recebe energia por indução, daí advém o seu nome, motores de indução.

O motor de indução trifásico, também conhecido por motor assíncrono trifásico, possui as seguintes características: robustez, fiabilidade, económico, elevado desempenho possuindo ainda um binário de arranque que satisfaz a maioria das aplicações praticas.

1.2.1 Constituição do motor de indução trifásico

Um motor de indução trifásico é constituído pelo estator, pelo rotor e por uma carcaça com se pode observar na Figura 2. O estator é a parte imóvel da máquina e consiste num conjunto de chapas ferromagnéticas empilhadas e isoladas eletricamente entre si de forma a reduzir as perdas por histerese e as correntes de *Foucault*. Nas ranhuras das chapas estão alojados os enrolamentos que são alimentados por um sistema trifásico de correntes. Existem diferentes estatores mas na sua composição mais simples, é constituído por três enrolamentos que perfazem entre si 120° , fases A, B e C. Cada fase é responsável por criar um campo magnético, resultando o seu efeito combinado num campo magnético girante no estator do motor.

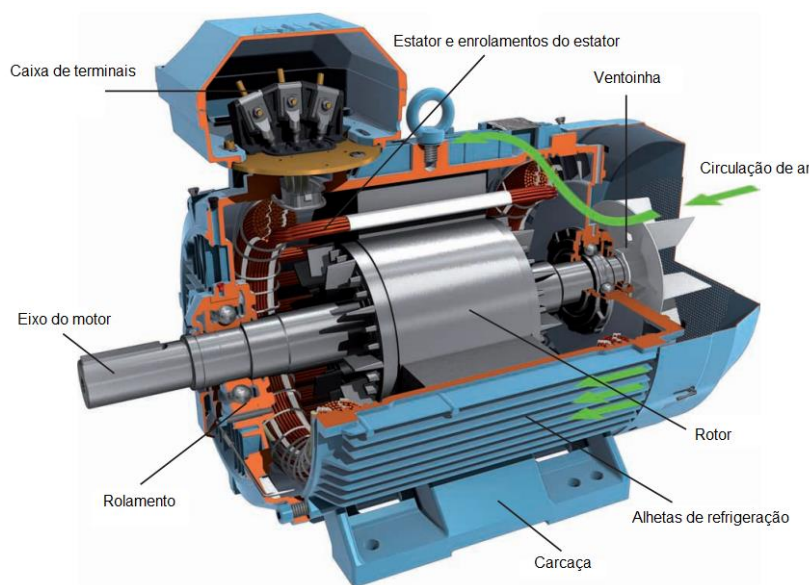


Figura 2 – Motor de indução trifásico [4].

O rotor, parte móvel da máquina, é formado, assim como o estator, por um conjunto de finas chapas isoladas entre si e está apoiado no veio de rotação do motor, que é suportado por um rolamento em cada extremo.

O estator e o rotor possuem entre si uma pequena abertura de ar, denominada por entreferro. Que quanto mais pequena for a sua dimensão menor será a relutância magnética total do circuito e assim sendo maior será a indução e por consequência maior será o fluxo magnético disponível.

A carcaça tem como objetivo alojar o estator e o rotor assim como fornecer proteção aos componentes do motor dos efeitos negativos e prejudiciais do ambiente em que este opera. A carcaça é constituída por ferro fundido, aço ou alumínio. Por norma no eixo é montada uma ventoinha que força a refrigeração do motor [5].

1.2.2 Tipos de motores de indução trifásicos

Existem dois tipos de motores de indução trifásicos, os quais diferem principalmente no rotor: o motor de rotor em gaiola de esquilo e o motor de rotor bobinado.

O motor de rotor em gaiola tem como principal característica a existência de barras condutoras dispostas em paralelo e ligadas entre si, tanto mecanicamente como eletricamente, as quais são inseridas em ranhuras das chapas do rotor. A sua constituição faz lembrar uma gaiola de esquilo, daí advém o nome pelo qual o motor é conhecido [5].

O facto da alimentação deste tipo de motor não necessitar de coletores nem de escovas, as extremidades das ligações elétricas terminam em anéis condutores, é provavelmente a sua principal vantagem tendo em conta que não existem contactos elétricos sujeitos a desgaste. As características apresentadas tornam este motor uma opção robusta para aplicações industriais que solicitem reduzida manutenção.

O motor de rotor bobinado é um motor que possui nas ranhuras do rotor enrolamentos que são ligados a anéis coletores colocados no veio. Estes anéis em contacto com escovas ligam o rotor ao circuito exterior. Trata-se de um motor, normalmente de potência elevada, utilizado em aplicações que necessitem de arranques de carga com elevado binário resistente e grande inércia [5]. Possibilita arranques suaves e progressivos recorrendo a resistências rotóricas, ligadas, através das escovas e dos anéis coletores, em série com o enrolamento trifásico do rotor. Tendo em conta estas vantagens, para uma mesma aplicação, o motor de rotor bobinado é menos económico e menos eficiente que o motor de rotor em gaiola. Por isso, este tipo de motor só é utilizado caso o motor de rotor de gaiola não consiga fornecer o binário de arranque pretendido.

1.2.3 Princípio de Funcionamento

Nos motores assíncronos trifásicos, o estator é usualmente formado por um conjunto de três enrolamentos colocados de forma a formar entre eles um ângulo de 120° . Quando alimentados com um sistema trifásico simétrico de tensões, formam-se três campos magnéticos alternados sinusoidais, um por enrolamento. Tal como as correntes de alimentação, estes campos estão desfasados de 120° entre si. A soma destes três campos resulta num campo magnético girante na máquina que roda a uma velocidade expressa por:

$$n_s = 60 \frac{f}{p} \quad (1)$$

Na equação (1) apresentada anteriormente f representa o valor da frequência da corrente (Hz), p o número par de polos do motor sendo n_s a sua velocidade de sincronismo (r.p.m.) [5].

Este campo girante atua sobre o enrolamento rotórico, assim como sobre o estatórico, induzindo nele uma f.e.m (força eletromotriz) que, uma vez fechado o enrolamento, origina a circulação de corrente rotórica. De acordo com a lei de Lenz, estas correntes induzidas têm sentido tal que, pelas suas ações magnéticas, tendem a opor-se à causa que lhe deu origem. Assim e uma vez que o rotor é móvel em torno do seu eixo (veio), tenderá a rodar no mesmo sentido e à mesma velocidade do campo girante estatórico, por forma a anular a causa do aparecimento dessas correntes. No entanto devido às perdas mecânicas (binário resistente) e outras que ocorrem no motor, a velocidade de rotação alcançada será sempre menor à velocidade de sincronismo n_s . O rotor desloca-se portanto com um certo deslizamento s em relação ao campo girante estatórico, o qual pode ser expresso de acordo com a equação (2).

$$s = \frac{(n_s - n)}{n_s} \quad (2)$$

1.3 Impacto Energético dos Motores de Indução Trifásicos

Os motores elétricos são pelas suas características usados em larga escala nos mais diversos sectores de atividades. Esse facto traduz-se em consumos de energia elétrica muito elevados.

Como é possível observar pela Figura 3 o motor elétrico, no sector industrial, é responsável pelo consumo de grande parte da energia elétrica.

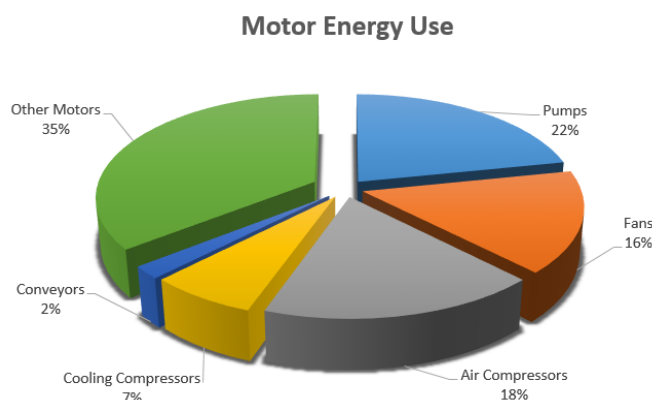


Figura 3 – Consumo de energia dos motores no setor industrial na EU [6].

De acordo com estudos efetuados [7] [8] estima-se que entre 30% a 40% do consumo de energia elétrica, nas diversas aplicações industriais, esteja associado a motores elétricos.

Tabela 1 – Consumo de energia elétrica por parte dos motores elétricos em diversos países.

Pais	Motor, uso de energia (%)
EUA	75
Reino Unido	50
EU	65
Jordânia	31
Malásia	48
Turquia	65
Eslovénia	52
Canadá	80
Índia	70
China	60
Coreia	40
Brasil	49
Austrália	30
África do Sul	60

Na União Europeia, os motores elétricos são os mais utilizados, sendo responsáveis por cerca de 65% [6] do consumo de eletricidade verificado no setor industrial, como se pode observar da análise da Tabela 1.

A larga utilização dos motores elétricos no setor industrial torna pertinentes a implementação de medidas que visem o aumento da sua eficiência. Apesar da vasta diversidade de motores elétricos disponíveis no mercado, os motores de indução trifásicos com rotor em gaiola de esquilo são dominantes, representando a maior quota de mercado.

Estima-se que na indústria cerca de 90% dos motores elétricos são motores de indução trifásicos de gaiola de esquilo na gama dos 0.75 kW aos 750 kW [9]. A sua popularidade deve-se ao facto

de serem máquinas com elevado número de vantagens, de entre elas refira-se a sua versatilidade, a ampla escala de potências nominais disponíveis que abrange praticamente todos os escalões e a sua constituição relativamente simples, o que confere a estas máquinas uma grande robustez e baixos custos de fabrico.

A sua utilização em grande escala, nos principais setores industriais, torna perceptível que a obtenção de um ganho na eficiência do motor em alguns pontos percentuais, resulta numa elevada poupança nos consumos de energia atualmente registados. Assim sendo, assume particular relevância a existência de um dispositivo eletrónico, que permita o aumento da eficiência deste tipo de motores.

Para o efeito, é necessário desenvolver uma aplicação que efetue o controlo e monitorização dos motores, tendo em conta a crescente necessidade de reduzir o consumo elétrico das empresas a partir de formas inovadoras. A monitorização de motores introduz diversas vantagens, pois permite ao utilizador prever com uma maior exatidão o momento em que o motor realmente necessita de manutenção e assim precaver problemas inerentes à manutenção corretiva. É de extrema importância para as indústrias produtoras o conhecimento das características e parâmetros de funcionamento dos motores, a fim de se evitar ao máximo paragens indesejáveis que implicam perdas económicas avultadas no sector da produção.

1.4 Controlo de Motores

O motor constitui a peça central de uma instalação industrial de qualquer empresa produtiva e por isso é necessário conhecer as suas características, proteções e forma de arranque.

Os arrancadores $Y \rightarrow D$ (estrela – triângulo, do inglês star \rightarrow delta) baseados em contadores são os que lideram o mercado tecnológico. Estes representam uma grande fatia do equipamento usado para fazer *soft starts* nos motores de indução. Contudo estes dispositivos têm alguns inconvenientes, os contactos mecânicos e o seu elevado tempo de comutação são dois bons exemplos de inconvenientes que podem danificar tanto o motor como o dispositivo em si.

Por outro lado, *soft-starters* foram desenvolvidos especificamente para permitir um arranque suave no motor modulando a frequência da tensão de alimentação no arranque. No entanto, este tipo de tecnologia provoca harmónicos de rede que contribuem para a degradação da instalação. Esta tecnologia não se conseguiu impor no mercado.

A tecnologia mais recente é o *SinuMec*. Trata-se de um dispositivo capaz de fazer arranques suaves no motor assim como controlar a taxa de fluxo durante o seu tempo de funcionamento. Comparando esta tecnologia com as demais apresentadas, esta é a única que se compromete a

poupar energia nos motores de indução de carga variável e velocidade constante, todavia não se conseguiu impor no mercado, provavelmente pelo seu elevado custo.

Uma vez que estas tecnologias não se conseguiram impor no mercado, outras tecnologias e outros conceitos foram surgindo para tentar combater as lacunas que as demais deixaram. Em seguida são descritas algumas das tecnologias e dispositivos utilizadas que permitem de alguma forma fazer o controlo de motores.

1.4.1 Contatores

Um dos dispositivos mais simples usado no controlo industrial de motores é o contator. Trata-se de uma chave eletromagnética que tem como função controlar (ligar/desligar) cargas num circuito de potência a partir de um circuito de comando, cujos contactos alteram o seu estado no instante em que é ativado. Isto é, os contactos do contador que estavam abertos, fecham e os que estavam fechados, abrem. Assim, quando o operador acionar o contator, este por sua vez vai acionar o equipamento de potência a ele associado. O contator é um dispositivo de baixo consumo de energia e pode ser acionado por corrente alternada ou corrente continua.

Basicamente, existem dois tipos de contatores:

- Contator de potência;
- Contator auxiliar.

Enquanto o contator de potência liga e desliga o motor e outras cargas elétricas, o contator auxiliar liga e desliga circuitos de comando, sinalização, controlo, interface, etc.

Os contatores têm na sua constituição um conjunto de contactos fixos e outro de contactos móveis, cujo movimento de abrir/fechar é comandado pela parte móvel de um núcleo de ferro, que por sua vez está envolvido por uma bobine que ao ser alimentada cria um campo magnético, forçando o movimento da parte móvel desse núcleo.

Tecnicamente os contatores possuem um conjunto de contactos normalmente fechados (NF) que “abrem” e um conjunto de contactos normalmente abertos (NA) que “fecham” quando a bobine é alimentada. A título de exemplo, um contator utilizado para o controlo de um motor trifásico é constituído por:

- 3 contactos principais NA, que pertencem ao circuito principal trifásico;
- 1 contato auxiliar NA, que pertence ao circuito de comando;
- 1 contato auxiliar NF, que pertence ao circuito de sinalização.

1.4.2 Smart Switch

O *Smart Switch*, representado na Figura 4, é um dispositivo eletrónico de baixo custo que efetua a mudança automática de estado dos enrolamentos do estator (estrela ou triângulo), com a finalidade de melhorar a eficiência e o fator de potência em motores de indução de carga variável.

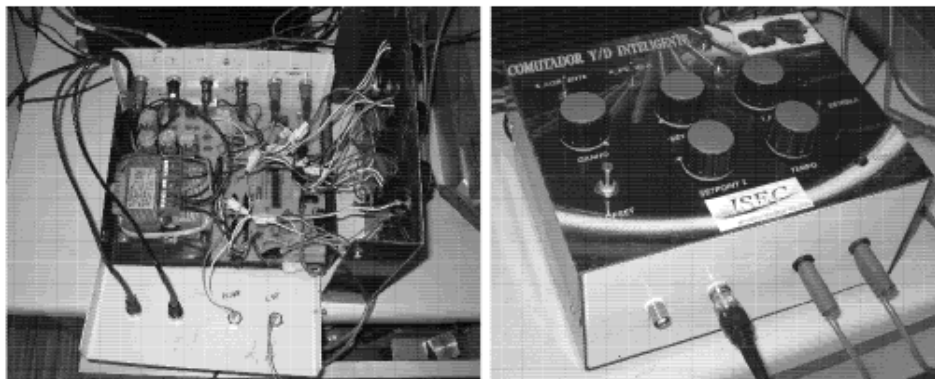


Figura 4 – *Smart-Switch* desenvolvido pelo ISEC [10].

1.4.3 SinuMEC

O *SinuMEC* apresenta-se como uma nova categoria de controladores de motores de corrente alternada que utiliza a tensão correta para melhorar a eficiência dos motores de carga variável com velocidade variável. Construído a volta de uma tecnologia patenteada, o *SinuMEC* fornece uma forma de onda de tensão sinusoidal pura quando o motor é ligado e durante o seu normal funcionamento. O *SinuMEC*, representado pela Figura 5, monitoriza continuamente o consumo de energia do motor e reduz a tensão, quando a carga no motor diminui, permitindo assim um melhor desempenho do motor e um aumento da eficiência energética [11].



Figura 5 – Exemplo de um SinuMEC [11].

1.4.4 *Soft-starters*

Em termos técnicos, um *soft-starter* é qualquer dispositivo capaz de reduzir o binário aplicado ao motor elétrico. Geralmente consiste num dispositivo de estado sólido como tirístores para controlar a aplicação da tensão de alimentação para o motor. O funcionamento do *soft-starter* é baseado no facto do binário ser proporcional ao quadrado da corrente de arranque, que por sua vez é proporcional à tensão aplicada. Assim, o binário e a corrente podem ser ajustados através da redução da tensão no momento do arranque do motor.

O *soft-starter* é, em geral, constituído por alguns componentes principais, tais como tirístores, uma placa de circuitos impresso, dissipador de calor, ventiladores e caixa (plástico ou metal). Os circuitos de controlo podem ser do tipo digital ou analógico ou ainda uma combinação destes. O sinal de saída dos relés pode ser programado para ser de um tipo com função fixa ou como um tipo de programação livre onde o usuário pode decidir sobre a função de saída [12].

Podem existir dois tipos de controlo usando *soft-starters*:

- 1) *Open Control*: A tensão inicial é aplicada ao longo do tempo, independentemente da corrente consumida ou velocidade do motor. Para cada fase são usados dois *Silicon-Controlled Rectifier's* (SCR's) estão conectados "back-to-back". Os SCR's são alimentados inicialmente com um atraso de 180° graus durante os respetivos ciclos de meia onda, este atraso é posteriormente reduzido de uma forma gradual com o tempo até as rampas de tensão aplicada atingirem a tensão de alimentação. Este processo é conhecido como *Time Voltage Ramp System*. O método não é relevante, uma vez que na verdade não controla a aceleração do motor.
- 2) *Closed Loop Control*: Qualquer uma das características de saída do motor, tais como a corrente consumida ou a velocidade, é monitorizada e a tensão inicial é modificada por forma a obter a resposta necessária. A corrente em cada fase é monitorizada e se for superior a um determinado ponto de referência, o tempo de rampa de tensão é interrompido.

Assim, o princípio básico de funcionamento do *softs-starter* afirma que se controlarmos o ângulo de condução dos SCR's, a aplicação da tensão de alimentação também pode ser controlada.

1.4.5 *InSwitch*

O *InSwitch* é um projeto inovador que promete ser uma alternativa a todas as tecnologias anteriormente descritas. O *InSwitch* distingue-se dos demais ao oferecer uma melhoria da eficiência e do fator de potência de motores trifásicos, através da comutação do tipo de ligação dos enrolamentos estatóricos. Apresenta-se como um dispositivo eletrónico plug & play de baixo custo, que permite *soft starts* e controlo sinusoidal nos motores de indução, os quais geram poupanças de energia através da melhoria da eficiência e do fator de potência para regimes de carga baixa.

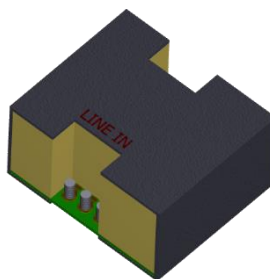


Figura 6 – Dispositivo *InSwitch*.

O dispositivo *InSwitch* representado na Figura 6 tem como principais objetivos: otimizar a eficiência dos motores, suavizar os seus arranques, proporcionar uma forma adicional de proteção, detetar as avarias que possam ocorrer, efetuar monitorização de consumos, reduzir as perdas por condução, aumentar a qualidade da onda de tensão evitando a injeção de harmónicos na rede. O *InSwitch* foi desenvolvido de forma a permitir o controlo, a gestão inteligente e a supervisão remota de motores.

A otimização da eficiência é obtida através da estimativa que é feita da carga aplicada ao motor. O dispositivo comuta de forma autónoma o tipo de ligação dos enrolamentos do motor, optando pelo tipo de ligação mais eficiente para cada regime de carga. O arranque suave do motor é feito pela comutação do tipo de ligação dos enrolamentos do estator, reduzindo assim até três vezes a corrente de arranque. O dispositivo eletrónico *InSwitch* permite detetar problemas nos enrolamentos assim como permite a proteção contra sobreintensidades e alimentações desequilibradas. É possível também a aquisição de grandezas elétricas que possibilitam ao utilizador conhecer os consumos médios e instantâneos do motor. Com o *InSwitch* é ainda possível reduzir a corrente absorvida pelos motores a qual resultam poupanças, quer na máquina quer nos cabos de alimentação.

Os harmónicos provocam mau funcionamento dos equipamentos eléctricos, perdas por efeito de joule e redução de qualidade de rede eléctrica. Com o controlo sinusoidal, o *InSwitch*, permite fazer poupanças de energia sem contaminar a instalação com harmónicos. Para minimizar ainda mais as perdas o *InSwitch* adquire em tempo real os valores eléctricos do motor, possibilitando assim realizar o controlo do tipo de ligação em tempo real, este controlo pode ser remoto, isto é, o do arranque/paragem pode ser realizados à distância.

1.5 Objetivos

Após se ter apresentado o contexto em que este projeto se insere, são agora indicados os objetivos que se pretende atingir.

Numa primeira fase era pretendido realizar um estudo comparativo sobre as diferentes tecnologias de comunicação passíveis de implementação em ambiente industrial. Com base na informação recolhida pretende-se determinar as tecnologias que melhor se adequam ao ambiente industrial e que sirvam os melhores interesses do projeto.

Posteriormente, será necessário definir a informação a ser enviada na comunicação pelos motores eléctricos numa perspectiva de facilitar a interoperabilidade entre a máquina e o homem. A informação recolhida é registada numa base de dados para monitorização, posterior análise e previsão de possíveis avarias.

A elaboração de uma infraestrutura que possibilite a comunicação entre o “motor” e um dispositivo móvel (telemóvel e/ou tablet) assim como entre o motor e um PC, servirá de base à plataforma de monitorização que se pretende desenvolver.

O objetivo final desta dissertação compreende o projeto e desenvolvimento de um módulo de comunicação capaz de se integrar no dispositivo eletrónico a instalar nos motores. Por fim será necessário analisar e testar o referido módulo, numa aplicação prática real.

1.6 Metodologia

O projeto descrito nesta dissertação foi realizado conjuntamente no laboratório de máquinas do ISEC e nas instalações da empresa Optisigma. Para o desenvolvimento do mesmo foi inicialmente realizado uma revisão bibliográfica utilizando diversas fontes de informação, com o intuito de recolher a maior quantidade possível de informação. Posteriormente realizou-se um estudo sobre os diversos temas necessários ao desenvolvimento deste projeto de forma a conhecer as suas características, aplicações e tendências. Este projeto abrange diversos temas tais como: motor de indução trifásico e o seu impacto energético, controlo de motores,

tecnologias de comunicação industrial (cablados e sem fios), microcontroladores, tecnologias de gestão de informação digitais, etc.

1.7 Estrutura da Dissertação

A presente dissertação é estruturada da seguinte forma:

No capítulo 1 é feita uma contextualização histórica dos motores e uma abordagem à sua evolução ao longo dos anos. Apresentam-se os dois tipos de motores de indução trifásica, a sua constituição e funcionamento assim como o seu impacto energético no setor industrial. Segue-se uma análise de vários tipos de tecnologias capazes de controlar um motor. Por fim são apresentados os objetivos, a metodologia adotada e a estrutura da tese.

No capítulo 2, estado de arte dos sistemas de comunicação industrial, depois de uma breve introdução, é abordado o tema da comunicação industrial, analisando alguns dos sistemas cablados e alguns dos sistemas sem fios mais comuns.

No capítulo 3 é feita uma breve introdução à solução tecnológica desenvolvida e em seguida são apresentados algumas das unidades microcontroladas que existem no mercado. São introduzidos os módulos de comunicação sem fios e as unidades microcontroladas selecionadas. Para completar o capítulo é feita uma introdução das ferramentas, tecnologias e *softwares* utilizados ao longo do desenvolvimento do projeto.

No capítulo 4, depois de uma breve introdução é apresentado com mais detalhe algumas das características do *InSwitch*. Em seguida são demonstrados os protótipos desenvolvidos, o protocolo de comunicação proposto, a arquitetura do interface *web* implementada e a base de dados que foi construída. Por fim é apresentada em detalhe a solução final desenvolvida para a rede de monitorização de motores de indução.

Finalmente no capítulo 5 são apresentadas as conclusões mais relevantes obtidas, sendo também expostas algumas sugestões de trabalho futuro.

Capítulo 2

ESTADO DE ARTE DOS SISTEMAS DE COMUNICAÇÃO INDUSTRIAL

2.1 Introdução

No presente capítulo pretende-se realizar uma análise comparativa dos sistemas de comunicação industriais, com o objetivo de perceber a importância das tecnologias de informação que compõem a “pirâmide da automação” e dos protocolos de comunicação que permitem a ligação de todas estas tecnologias em harmonia. Em seguida serão discriminadas algumas das mais utilizadas soluções de comunicação na indústria, descrevendo a sua história, uma breve descrição da tecnologia, assim como o seu modo de comunicação.

2.2 Comunicações Industriais

A necessidade de aperfeiçoar o controlo de dispositivos de campo, aumentar a capacidade /velocidade de tráfego de dados e promover mensagens de diagnósticos e de configuração remotamente entre os diversos elementos, levou ao desenvolvimento de protocolos de comunicação industriais e a uma melhoria na automação industrial. A automação industrial permite economias: de energia, de força de trabalho e de matérias-primas, um melhor controlo de qualidade do produto e uma melhor gestão do plano de manutenção. Todas estas características resultam num aumento de produtividade e segurança operacional. Em suma a automação industrial permite elevar os níveis de produtividade e de controlo dos processos com uma maior eficácia, aproximar ao máximo a produção real ao expectável, ao reduzir ao mínimo as paragens e interrupções, de manutenção corretiva [13].

A tecnologia da informação tem sido determinante no desenvolvimento da automação nos mais diversos ambientes industriais. A capacidade de comunicação entre dispositivos e o uso de mecanismos padronizados, abertos e transparentes são componentes indispensáveis do conceito de automação de hoje. Na Figura 7 onde está representada a pirâmide da automação, é possível distinguir os diversos níveis de automação assim como as diversas tecnologias utilizadas para a comunicação entre os mesmos níveis.

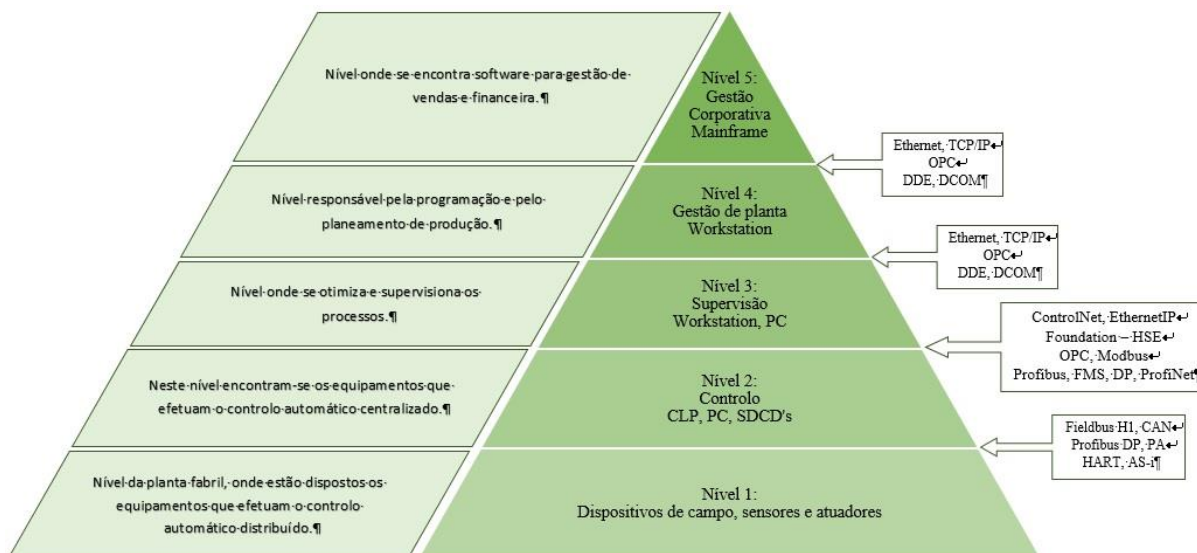


Figura 7 – Níveis da pirâmide de automação [14].

Os sistemas de comunicação, parte constituinte da automação industrial, podem ser divididos em duas grandes áreas, sistemas cablados e sistemas sem fios. De seguida são apresentados detalhadamente alguns dos sistemas de comunicação mais utilizados na indústria.

2.3 Sistemas Cablados

2.3.1 Fieldbus

Histórico

Os primeiros processos industriais utilizavam sensores de pressão na ordem dos 3 a 15 psi para a monitorização de dispositivos de controlo. Mais tarde foi apresentada uma solução usando um padrão com sinal de 4 a 20 mA para instrumentação. Contudo este padrão apresentava muitos níveis de sinais que não atendiam as especificações. Com a migração do sistema pneumático para o sistema elétrico advieram muitas vantagens para os sistemas industriais, tais como: a diminuição de ruído, uma maior simplicidade de implementação e manutenção, aumento de confiabilidade, etc.

O aparecimento de processadores, na década de 70, impulsionou o uso de computadores para a monitorização e controlo de dispositivos a partir de um dispositivo central. Com a implementação e utilização destes novos dispositivos e métodos de controlo, rapidamente se percebeu que era necessário implementar uma padronização. Na década de 80, surgiram os sensores inteligentes os quais foram utilizados em sistemas microcontrolados, pois estes eram de grande confiabilidade, rapidez e de baixo custo.

Com esta tendência nasceu um movimento que incluía a *International Society of Automation* (ISA), *International Electrotechnical Commission* (IEC), Profibus e *Factory Instrumentation Protocol* (FIP) para a formação do comité IEC/ISA SP50 *Fieldbus*, com o intuito de criar e especificar normas e padrões para instrumentação. O padrão a ser desenvolvido deveria integrar diferentes tipos de dispositivos de controlo, oferecendo uma interface para a operação dos diversos dispositivos simultaneamente e também um conjunto de protocolos de comunicação para todos eles. O processo de padronização tornou-se lento devido à grande diversidade de produtos e métodos de implementação, não possibilitando assim a elaboração de uma solução imediata e simples para ser padronizada.

Já em 1992, dois grandes grupos lideravam o mercado para soluções de interligação de instrumentos de campo: *Internet Service Provider* (ISP) e *WorldFIP*. Ambas tinham uma conceção e implementação diferentes de redes *fieldbus* mas ainda assim garantiram que iriam alterar os produtos se a norma SP50 fosse formalizada. Em 1994, a ISA em conjunto com *WorldFIP* criaram a *Fieldbus Foundation*, com o objetivo de acelerar o processo de normalização das redes *fieldbus*. Contudo continuava difícil de se alcançar uma solução de padronização e por isso o comité SP50 decidiu padronizar apenas quatro níveis da rede *Fieldbus*. [15]

Descrição

Fieldbus é um protocolo voltado para a automação de sistemas de produção, desenvolvido pela *Fieldbus Foundation* o qual foi normalizado pela ISA, uma sociedade internacional de medida e controlo. O *Fieldbus* proporciona uma interligação dos dispositivos de campo com os equipamentos informáticos, possibilitando o controlo e monitorização de processos. É geralmente utilizado com programas de supervisão (p. ex. SCADA) que permitem a aquisição e visualização de dados (desde dados de um sensores até ao estado de equipamentos). [15]

Modo de Comunicação

O *Fieldbus* foi desenvolvido com base no padrão OSI, mas não contém todos os seus níveis. O protocolo *Fieldbus* pode ser dividido essencialmente em dois níveis principais: nível físico (baseado na camada física do modelo OSI) e nível *software* (baseado nas camadas 2 e 7 do modelo OSI).

A norma inicial subdividia-se em duas opções para a camada física: H1 e H2. O H1, com uma taxa de 31.25 kbps, foi projetado para equipamentos de campo e pode ser usado em áreas que necessitem de segurança intrínseca (ambientes explosivos). O H2, com uma taxa de 1 a 2.5 Mbps, foi concebido para integrar controladores e equipamentos mais complexos. Mas a

rápida evolução tecnológica determinou a substituição do H2 pelo *High Speed Ethernet* (HSE), que utiliza *Ethernet* a 100 Mbps. [16]

Portanto, para conexões de equipamentos de campo existe o *Foundation Fieldbus H1*, com camada física com base na ISA-S50.02-1992 ou IEC61158-2:2000. Para interligações entre controladores lógicos programáveis, *Gateways* e PCs, existe o *Foundation Fieldbus HSE*, sustentado na *Ethernet* (IEEE 802.3-2000, ISSO/IEC 8802.3-2000). [16]

Características Físicas

Como referido anteriormente existem dois tipos de rede:

- **Rede Foundation Fieldbus H1:**

É uma rede constituída por vários barramentos H1, interligados entre si através de *bridges* ou *Linking Devices Foundation Fieldbus* e que por sua vez conectam as redes H1 ao *backbone* HSE. Cada barramento H1 é capaz de suportar até 32 equipamentos de campo, cada um com um endereço lógico único. Destes 32 equipamentos, 12 são alimentados pelo próprio barramento e os restantes 20 equipamentos não são alimentados pelo barramento. [17]

- **Rede Foundation Fieldbus HSE:**

As características de comunicação e sincronismo entre os equipamentos são praticamente as mesmas do H1, sendo que as principais diferenças residem no determinismo. Através do uso da rede *Ethernet* na rede *Foundation Fieldbus HSE* é exequível a construção de uma rede de controlo industrial com componentes diversos independentemente do fabricante. [17]

Na Tabela 2 é possível visualizar as características da camada física da rede *Fielbus*.

Tabela 2 – Características da camada física da rede Foundation Fieldbus.

	H1	HSE
Número de equipamentos		Até 32
Sinal Alimentação		12 a 32 V DC
Sinal Comunicação		750 a 1000 mV
Taxa de Comunicação	31.25 kbits/s	10 ou 100 Mbit/s
Distância (segmento)	1900 m (cabo tipo A)	100 m
Dois fios	Sim	Não
Multidrop	Sim	Não (UTP)
Bus-power	Sim	Não
Intrinsecamente Segura	Sim	Não
Redundância	Não	Sim
Determinística	Sim	Sim (com switches)

Exemplo de topologia

Como é possível verificar pela Figura 8, os projetos Fieldbus podem assumir diferentes topologias. As topologias de rede Fieldbus mais comuns são as de barramento, ponto a ponto, árvore e mista.

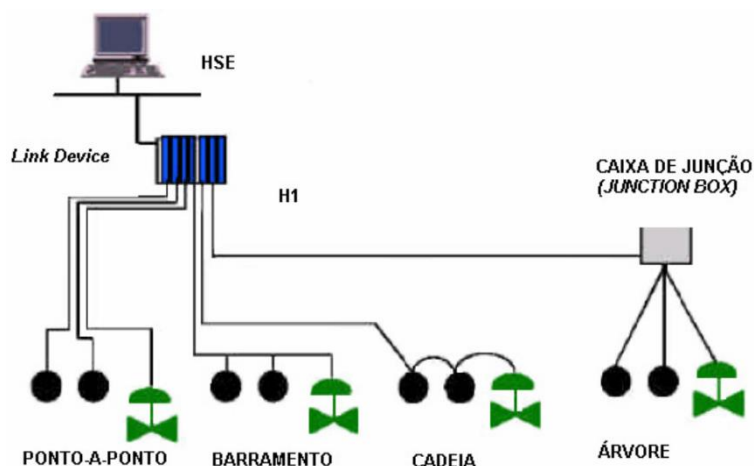


Figura 8 – Exemplos de topologia Fieldbus [18].

2.3.2 ModBus

Histórico

O protocolo *ModBus* foi desenvolvido pela *Modicon*, hoje *Schneider Electric*, e introduzido no mercado em 1979. [19]

O facto de ser um protocolo simples e com mais de 30 anos de mercado, levou a que hoje em dia exista um grande número de fabricantes em todo o mundo que providenciam os seus equipamentos com a capacidade de comunicar em *ModBus*.

Descrição

O *ModBus* é um protocolo aberto e por norma utilizado em comunicação série. Trata-se de um *bus* extremamente divulgado e com grande potencialidade de utilização em, variadores de velocidade, robots, máquinas especiais, autómatos programáveis industriais, etc.

No mercado existem duas opções para ModBus, o *American Standard Code for Information Interchange* (ASCII) e o *Remote Terminal Unit* (RTU), sendo que o último é o mais utilizado e com melhor desempenho:

- ASCII: Caracteres codificados em 7 bits + 1 bit de paridade;
- RTU: Caracteres codificados com 8 bits +1 bit de paridade. [20]

Modo de Comunicação

O ModBus tem semelhanças com o modelo *Open Systems Interconnection* (OSI) ao nível das camadas 1, 2 e 7, respetivamente camada física, ligação e aplicação.

Nível físico: Par trançado, débito máximo de 19200 Bauds, RS232/RS485/Anel de corrente.

Ligação de dados: O controlo de erros utilizado é o CRC16 (Modo RTU) sendo o acesso à rede realizado por mecanismos do tipo mestre/escravo, onde só o mestre pode tomar a iniciativa de enviar mensagens. Se uma resposta for requisitada, os escravos respondem à solicitação do mestre ou então limitam-se a executar as ações pedidas pelo mesmo. O mestre pode dirigir-se individualmente aos escravos ou difundir mensagens em *broadcast*.

Aplicação: Neste nível foram definidas as funções de leitura e escrita de variáveis (bits, Words, E/S), diagnóstico e estatísticas de ocorrência da rede. [21]

Características Físicas

Topologia de rede: Em linha e em estrela.

Máximo de equipamentos (Nós): 64;

Suporte: Par trançado blindado, RS-485 (opção de anel) ou fibra ótica (opção de redundância);

Distância máxima em par trançado (segmento): 450 m;

Com repetidores: 1800 m;

Velocidade de transmissão: 1 Mbps. [21]

Exemplo de topologia

O protocolo Modbus, representado na Figura 9, suporta topologias em linha e em estrela.

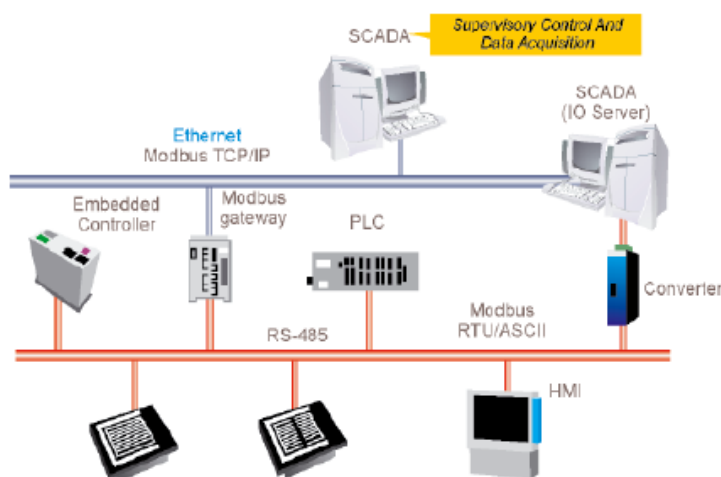


Figura 9 – Exemplo de uma rede com diferentes vertentes *ModBus* [22].

2.3.3 Profibus

Histórico

Tem origem em 1987, pelos fabricantes de automação alemãs, pelo o governo Alemão e por outras empresas líderes da indústria. O seu esforço conjunto possibilitou a conceção de uma solução de automação que ainda hoje é viável e permitiu o aparecimento de novas soluções. A definição de *Profibus Decentralized Peripherals* (Profibus DP) aparece em 1993. Esta foi aceite três anos mais tarde como norma europeia e em 2000 é incluída na norma IEC 61158. [23]

Descrição

A norma *Profibus* foi desenvolvida nos anos 80 por *Johan Sartwisch Wilman* e apresentada pela organização *Profibus Trade Organization*. Esta evoluiu para três implementações no início dos anos 90, sendo hoje um dos protocolos mais difundidos na Europa e também na América. É adequado para o controlo de processos, uma vez que possui elevada velocidade, distância e gestão de dados. *PROFIBUS* é o acrónimo para *PROcess FIeld BUS* e existem 3 tipos de Profibus que podem coexistir numa rede:

- *Profibus FMS* (Sistema de Mensagens de Campo);
- *Profibus DP* (Periféricos Descentralizados);
- *Profibus PA* (Automação de Processos). [23]

A sua implementação é feita nos chips ASIC produzidos por diversos fabricantes, baseado em RS-485 e na especificação europeia EN50170. O *Profibus* é padronizado na norma IEC 61158.

É por norma utilizado em controlo de processos, linhas de montagem e manuseamento de materiais. Esta tecnologia tem bons tempos de resposta, podendo chegar aos 5 ms para uma rede a 12 Mbps.

Modo de Comunicação

O *Profibus FMS* é a versão disponibiliza ao utilizador que maior número de funções possui em comparação com as outras variantes. Por se tratar de um padrão de comunicação universal é por norma considerado uma solução para resolver tarefas complexas de comunicação entre PLC's e *Distributed Control Systems* (DCS). Esta versão do *Profibus* é normalmente utilizada ao nível do controlo e suporta ainda a comunicação entre sistemas de automação, tal como a troca de dados entre equipamentos inteligentes. Com a evolução da tecnologia e por se tratar de um protocolo que tem como função primária a comunicação mestre-mestre, tem sido substituído por aplicações em *Ethernet*.

O *Profibus DP* é o protocolo com melhor desempenho, de alta velocidade, especificamente dedicado a comunicação entre sistemas de automação e equipamentos descentralizados, destinado à comunicação com atuadores e sensores. A versão DP v2 para além de suportar as funcionalidades da versão posterior suporta ainda a comunicação entre escravos e ciclos de *bus* simultâneos. Com uma comunicação direta entre escravos, o ciclo de *bus* pode ser bastante reduzido (o tempo de ciclo de rede máximo é inferior a 5 ms a 12 Mbps). Permite impulsos de curta duração para garantir precisão em aplicações de movimentação, assim como o registo de hora e data na fonte e outras características que facilitam o diagnóstico. Atualmente a utilização de Profibus DP em aplicações que utilizam escravos *Profibus* ronda os 90%. [24]

O *Profibus PA* é a solução *Profibus* que pretende atender todos os requisitos da automação de processos, aplicado em sistemas onde exista uma ligação de sistemas de automação e sistemas de controlo de processos com equipamentos de campo, como por exemplo, sensores de pressão, temperatura, conversores, etc. A ligação destes equipamentos numa rede *Profibus PA* é feita através de um acoplador *DP/PA*. Para a alimentação e comunicação de dados de cada equipamento é utilizado um cabo par trançado de dois fios, o que facilita a instalação e a manutenção, assim como baixos custos do *software* de engenharia.

Todas as versões de *Profibus* têm semelhanças com alguns níveis do modelo de comunicação de redes OSI, mais propriamente os níveis 1, 2 e ainda o nível 7 no caso do *Fieldbus Message Specification* (FMS) [24]. Numa rede Profibus, o mestre, que controla a rede, deve conhecer toda a informação sobre os equipamentos, tais como os endereços, configuração de E/S e parâmetros de operação. Cada equipamento deve ter um ficheiro com a sua descrição única - *Generic Data Slave* (GDS) - para descrever as funcionalidades, características e parâmetros de configuração desse equipamento é usada uma ferramenta de configuração para combinar o ficheiro GSD e a informação do utilizador numa base de dados mestre que é utilizada para posteriormente estabelecer a comunicação e iniciar a troca de dados com o escravo *Profibus*.

Características Físicas

Topologia: Linear, anel ou estrela;

Número máximo de equipamentos: 127 (32 por segmento);

Suporte: A ligação mais comum é feita por RS-485, mas é também possível a utilização da fibra ótica;

Distância máxima: Entre 100 m e 1.900 m para um segmento simples, 6000 m no máximo;

Velocidade de transmissão: 9600 bits/s (1200 m), 500 kbps (400 m), 1.5 Mbps (200 m) e 12 Mbps (40 m), com cabo de fibra ótica;

Distância máxima: 24 km;

Velocidade de transmissão: 9600 bits/s a 12 Mbps. [21]

Exemplo de topologia

O Profibus pode ser implementado tendo em conta a ilustração apresentada na Figura 10. A topologia de uma rede Profibus pode ser em barramento, árvore, estrela e ponto a ponto.

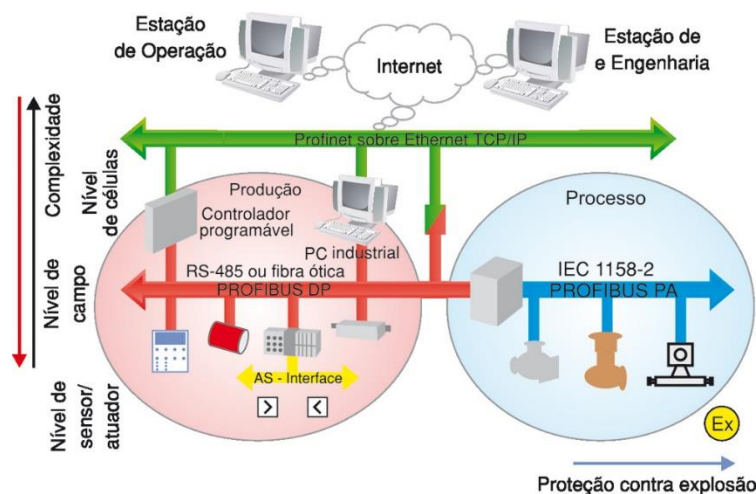


Figura 10 – Exemplo de rede PROFIBUS [25].

2.3.4 Ethernet TCP/IP

Histórico

Tem origem em 1973, desenvolvida pela Xerox no centro de pesquisa *Palo Alto Research Center*. Com o passar do tempo foram introduzindo alguns *upgrades* de onde se pode destacar a especificação da *Ethernet* 10 Mbps (1980), norma *Ethernet* 802.3 (1982), especificações da *Ethernet* 100 Mbps (1993), uso geral da *Ethernet* 100 Mbps (1996), normalização *Gigabit Ethernet* (1998), etc. [21]

Descrição

Ethernet é a rede mundialmente conhecida e utilizada para a implementação de redes de computadores. Transportar a *Ethernet TCP/IP* para a indústria e torna-la numa das redes com maior crescimento neste sector foi um enorme desafio.

O protocolo *Ethernet TCP/IP* pode ser empregue em qualquer aplicação que necessite de:

- Redes entre PLC's e sistemas de supervisão e interligação aos sistemas tecnológicos de informação;
- Redes onde o tempo é extremamente crucial e onde seja necessária uma comunicação em tempo real.

A enorme popularidade, desempenho, baixo custo e a comunicação com os PC's tornou a *Ethernet* atrativa para aplicações industriais.

Modo de Comunicação

A *Ethernet* é uma rede em que o acesso ao meio é feito através do método *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD). Isto é, se dois nós em diferentes localizações tentam enviar dados ao mesmo tempo, o resultado será uma colisão e no caso de ocorrer uma colisão, as mensagens são destruídas e a cada nó é atribuído um período de espera que vão ter de respeitar até que seja possível enviar de novo uma mensagem [26]. Contudo existem regras que ajudam a evitar conflitos na rede e proteger a integridade dos dados. Um nó pode “escutar” a rede para tentar perceber se outro nó está transmitir e assim determinar se deve transmitir nesse instante.

Características Físicas

Na indústria é usado uma versão da *Ethernet* muito semelhante à dita *Ethernet* padrão, no entanto esta foi projetada para ser utilizada em ambiente industrial, ou seja mais robusta em termos de componentes e testes. A ligação entre equipamentos pode ser feita através de diversos meios físicos:

- Ligação em cobre: A ligação física em cobre mais utilizada é o 10Base-T ou 100Base-TX, que utiliza cabo *Unshielded Twisted Pair* (UTP) ou *Shielded Twisted Pair* (STP) com fichas RJ45.
- Ligação fibra ótica: A fibra ótica, pela sua constituição, tem perdas mínimas ao longo do cabo e é imune a interferências eletromagnéticas.

A ligação pode ser feita em fibra monomodo, até 150 km, ou multimodo num máximo de 3 km.

Numa estrutura de rede *Ethernet* são utilizados diversos componentes, tais como, *hubs*, *switches* e *routers*, não podendo os seus constituintes ultrapassar os 1024 equipamentos por rede de área local. [21]

Exemplo de topologia

Uma rede *Ethernet* pode ser configurada em diversas topologias, mas tipicamente as topologias mais utilizadas são a topologia em estrela (Figura 11) e a topologia barramento (Figura 12):

- Topologia em estrela:

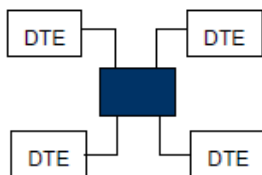


Figura 11 – Topologia em estrela [21].

- Topologia barramento:

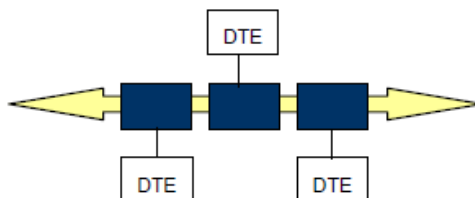


Figura 12 – Topologia barramento [21].

2.3.5 PLC

Histórico

As comunicações *PowerLine* já datam de algum tempo, a sua história é longa, quase tão antiga quanto a própria rede elétrica. Sabe-se que a primeira pesquisa e desenvolvimento foi realizado no fim do século XIX. No início do século seguinte, os sistemas de transporte de frequência começaram a trabalhar em linhas de alta tensão, depois em sistemas de distribuição de média e baixa tensão para fins de telemetria. O termo de transportadores *PowerLine* apareceu na década de 1940. Nos anos 1970 e 1980, foram colocados no mercado os dispositivos PLC para habitações familiares e ao mesmo tempo foram desenvolvidos as primeiras normas. O interesse aumentou durante a década de 1990, com as populares tecnologias como o X-10, CEBus e LonWoks. O objetivo era construir um sistema confiável que fosse barato o suficiente para ser amplamente instalado e capaz de competir com o baixo custo das tecnologias sem fios.

Nos últimos anos, as pesquisas sobre PLC foram-se intensificando, tendo sido propostas novas técnicas de codificação, de modulação e controlo de erro assim como novos padrões produzidos pelas alianças industriais e associações profissionais.

Descrição

A tecnologia de comunicação PLC permite enviar dados através de cabos de energia já existentes. Esta implementação na qual apenas cabos de energia estão ligados aos dispositivos possibilita a sua alimentação e ao mesmo tempo o envio/recepção de dados numa comunicação half-duplex. A tecnologia PLC pode ser dividida em duas classes distintas [27]:

- *Narrowband* PLC;
- *Broadband* PLC.

O *Narrowband* PLC funciona a baixas frequências (até 500 kHz), a baixas taxas de transferências (até 200 kbps) e tem maior alcance (até vários quilómetros) que pode ser estendido usando repetidores. Recentemente, a *narrowband* PLC tem recebido grande atenção devido às suas aplicações em *Smart Grids*. [27]

O *Broadband* PLC, em contraste, funciona em frequências mais elevadas (mais de 2 MHz), a altas taxas de transferências (mais de 1 Mbps) sendo usada em aplicações de curto alcance. Tem encontrado aceitação principalmente como solução que curto alcance para distribuição de *Internet* e redes domésticas. As altas taxas de transferências e dispensa de cabos, tornam o PLC de banda larga uma tecnologia interessante e eficaz para a distribuição de multimédia dentro de habitações familiares. [27]

Na Tabela 3 estão apresentados os principais parâmetros, aplicações e fornecedores dos sistemas *Narrowband* e *Broadband* PLC.

Tabela 3 – Comparação de *Narrowband* PLC e *Broadband* PLC [28].

	Narrowband PLC	Broadband PLC
Velocidade	Até 200 kbps	Mais de 1 Mbps
Frequência	Até 500 kHz	Mais de 2 MHz
Modulação	FSK, S-FSK, BPSK, SS, OFDM	OFDM
Aplicações	Automação predial Energias renováveis Iluminação de rua Veículos elétricos Smart Grid	Internet HDTV Áudio Jogos
Fornecedores	Ariane Controls Cypress Echelon Maxim ST Microelectronics Texas Instruments Yitran	Atheros Broadcom Lantiq Marvell Maxim Sigma

Modo de Comunicação

Num sistema PLC, o emissor modula os dados a serem enviados, injeta-os no meio de transmissão sendo o recetor responsável pela desmodulação dos dados por forma a os conseguir interpretar. A principal diferença é que o PLC não precisa de instalação de cabos extra, pois utiliza os existentes. Quanto à tecnologia de comunicação é por norma referenciado a sua relação com o modelo OSI. Alguns dispositivos PLC conseguem unicamente implementar a camada física do modelo OSI, enquanto outros podem integrar todas as 7 camadas [27].

Características Físicas

Na Tabela 4 são apresentados algumas das características da tecnologia PLC.

Tabela 4 – Classificação da tecnologia PLC com base na velocidade de transmissão [27].

	Low Data Rate	Medium Data Rate	High Data Rate
Velocidade	0-10 kbps	10 kbps-1 Mbps	>1 Mbps
Modulação	BPSK, FSK, SFSK, QAM	PSK-OFDM	PSK+OFDM
Standards	IEC 61334, ANSI/EIA 709.1/.2 UPD	PRIME, G3, P1901.2	G.hn, IEEE 1901
Frequência	até 500 kHz	até 500 kHz	@ MHz
Aplicações	Controlo e Comando	Controlo comando e Voz	Broadband PLC, rede doméstica

Exemplo de topologia

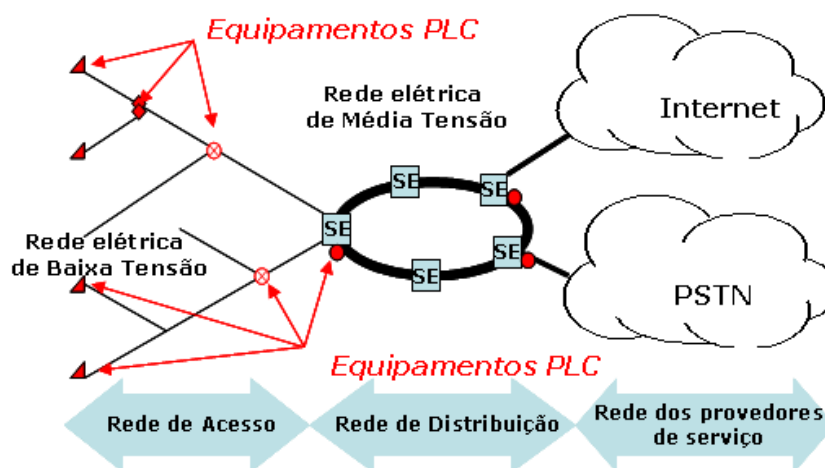


Figura 13 – Topologia de uma rede PLC [29].

Na Figura 13 é apresentado um exemplo de uma topologia de rede PLC, onde se distinguem os diferentes constituintes da mesma.

2.3.6 Profinet

Histórico

O aparecimento da especificação *Profibus* deu-se em 1989 pelo consórcio de empresas e organizações alemãs mais propriamente conhecido como *Profibus International*. Mas devido ao seu sucesso no mercado industrial e aliado à rápida evolução tecnológica e ao desejo por soluções de baixo custo, nasce em 2003 o conceito *Profinet*. Este novo protocolo é elaborado com base no seu antecessor, *Profibus*, mas desenvolvido para aplicações *Ethernet*. [30]

Descrição

O *PROFINET* é um padrão aberto, inovador e completo para *Ethernet* industrial, pois satisfaz a grande maioria dos requisitos da tecnologia de automação. O seu uso permite, aos fabricantes, minimizar os custos de instalação, investimento inicial e de engenharia. O utilizador final irá beneficiar com a facilidade de expansão e com uma automação rápida e eficiente. As redes Profinet podem ser divididas em dois grupos [31] [32]:

- *Profinet IO*: é a versão do *Profinet* mais utilizada para aplicações que necessitem de soluções em tempo real.
- *Profinet CBA*: é uma versão do *Profibus* desenvolvida para aplicações onde o tempo não é um fator crítico.

Modo de Comunicação

A comunicação padrão utiliza os protocolos TCP/UDP. Para estabelecer a comunicação entre o controlador e um equipamento de campo, devem ser estabelecidos os caminhos de comunicação. Esses caminhos de comunicação são definidos pelo controlador durante a inicialização do sistema com base nos dados de configuração do sistema de engenharia. Isto especifica explicitamente a troca de dados. Múltiplos controladores podem ser usados num sistema *Profinet*. Se for necessário os controladores podem ter acesso aos dados presentes nos equipamentos de campo, sendo que esta opção deve ser especificada durante a configuração. [31] [33]

Características Físicas

Topologia: Linear, anel, estrela e árvore;

Número máximo de equipamentos: 127 (32 por segmento);

Suporte: Soluções por cabo ou fibra ótica;

Distância máxima: 100 m entre segmentos;

Velocidade de transmissão: até 1000 Mbps. [34]

Exemplo de topologia

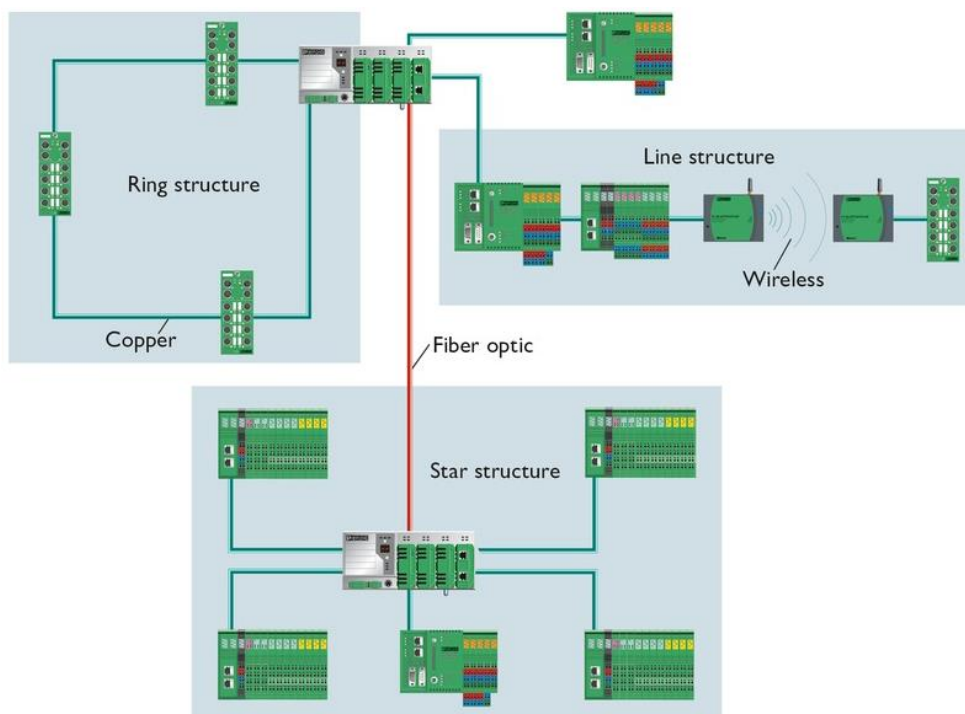


Figura 14 – Exemplo da topologia *Profinet* [35].

Na Figura 14 é representada a flexibilidade que caracteriza as redes Profinet, onde é possível verificar três distintas possíveis topologias (anel, linha e estrela) interligadas por cabo de fibra ótica.

2.3.7 AS-Interface

Histórico

Em 1991 através de uma parceria de 11 empresas europeias, nasce a organização *AS-International*, que apoia o *AS-Interface*. Hoje em dia a associação está aberta a qualquer fornecedor ou utilizador desta tecnologia, envolve 100 empresas em todo o mundo e oferece

mais de 600 produtos e serviços. Em 1997 iniciou-se o desenvolvimento do A2S-Interface, o qual permite o dobro dos escravos da versão antecessora, estando disponível desde 1998. [21]

Descrição

O *AS-Interface* é conhecido pelo seu característico cabo amarelo, é uma das mais inovadoras soluções de rede ao nível de sensores/atuadores. Foi desenvolvido como alternativa de baixo custo da estrutura de cabos e provou ser extremamente fiável, após vários anos de utilização em diversos setores industriais.

O objetivo principal é transmitir dados e alimentação simultaneamente entre sensores e atuadores de diversos fabricantes com um único cabo.

O *AS-Interface* foi criado com o intuito de se transformar numa solução para a complexa cablagem dos sistemas de automação ao nível do campo atual, prometendo assim substituir a cablagem tradicional nos meios industriais. Outra grande vantagem da tecnologia *AS-Interface* é a sua flexibilidade e interoperabilidade com qualquer outro *bus* de campo ou rede. [21]

Modo de Comunicação

Para a comunicação o sistema *AS-Interface* utiliza um só cabo comum que permite ligar todos os elementos periféricos da rede. Numa rede *AS-Interface* o elemento mais básico é o escravo, através do qual é possível ligar sensores e atuadores ao cabo *AS-Interface*.

Por ciclo, 4 bits de informação são transferidos em série do mestre para todos os escravos e outros 4 bits são devolvidos dos escravos para o mestre com entradas/saídas.

Com esta tecnologia é possível criar uma rede onde os componentes *AS-Interface* e os seus associados podem estar num módulo de interface onde sensores ou atuadores comuns se podem ligar. É também possível integra-los nos próprios sensores ou atuadores. Tal flexibilidade permite ao utilizador escolher entre reutilizar os seus equipamentos já existentes ou renovar a sua rede com equipamentos específicos *AS-Interface*.

Este sistema utiliza apenas um mestre por rede para controlar a troca de dados. O mestre interroga cada escravo sequencialmente e aguarda pela sua resposta. Uma vez que utiliza um formato fixo para a transmissão de dados, o *AS-Interface* elimina a necessidade de processos complicados de controlo de transmissão. Assim, o mestre consegue interrogar os 31 escravos e atualizar as entradas e saídas em menos de 5 ms. [21]

Características Físicas

Topologia: Estrutura em árvore;

Meio: Cabo de 2 condutores para dados e alimentação (24 VDC nominal, 2 A tipicamente);

Comprimento do bus: 100 m por cada mestre *AS-Interface* (300 m com repetidores);

Número de escravos: Máximo de 31;

Número de E/S: Até 4 sensores e 4 atuadores por escravo (máximo 248 E/S digitais);

Endereçamento: O endereço é definido pelo mestre ou equipamento específico;

Tempo de ciclo: Com 31 escravos, 5 ms. O tempo diminui ao reduzir o número de escravos;

Módulo Interface: 4 portas configuráveis (entradas, saídas ou bidirecional) e 4 parâmetros.

[21]

Exemplo de topologia

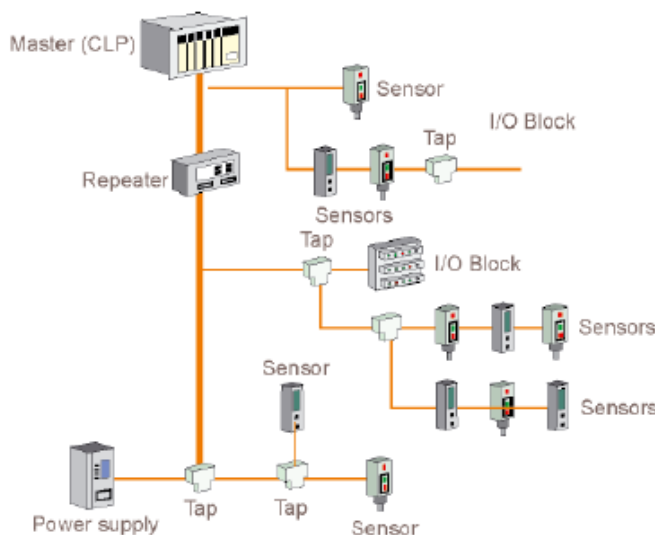


Figura 15 – Exemplo de uma topologia AS-Interface [22].

A Figura 15 apresenta um exemplo de uma topologia de uma rede AS-Interface, assim como alguns dos dispositivos possíveis de se ligar a este tipo de rede.

2.3.8 *DeviceNet*

Histórico

O protocolo *DeviceNet* foi apresentado em 1994 pela *Allen-Brandley*, um ano depois a sua tecnologia foi transferida para a *Open DeviceNet Vendor Association* (ODVA).

Fundada em 1995, a ODVA criou todas as especificações para a rede e equipamentos do sistema *DeviceNet*, baseado em tecnologia CAN. [36]

Descrição

A tecnologia *DeviceNet* é um protocolo de comunicação em automação utilizado para a troca de dados entre dispositivos de controlo, em ambiente industrial. O padrão aberto de comunicação baseia-se na camada 2 do modelo OSI e na técnica de transporte CAN. Apresenta como vantagem a possibilidade de manutenção sem ser necessário desligar a rede elétrica e assim remover e substituir equipamentos em redes sob tensão, sem um aparelho de programação, e/ou ainda a possibilidade de fornecer a alimentação aos equipamentos através do próprio cabo de rede. A sua aplicação surge ao nível operacional. É uma rede direcionada para equipamentos de baixa tensão, que proporciona comunicações fiáveis e possibilita a troca de informações entre sistemas de diferentes fabricantes. [36]

Modo de Comunicação

O protocolo *DeviceNet* disponibiliza dois tipos básicos de mensagens, *Cyclic I/O* e *Explicit Message*. Cada um deles é adequado a um determinado tipo de dados:

- *Cyclic I/O*: tipo de telegrama síncrono dedicado à movimentação de dados prioritários entre um produtor e um ou mais consumidores. Dividem-se de acordo com o método de troca de dados.
- *Explicit Message*: tipo de telegrama de uso geral e não prioritário. Utilizado principalmente em tarefas assíncronas tais como parametrização e configuração do equipamento.

Características Físicas

Topologia: Linear e com derivações;

Número máximo de equipamentos: 64 máximo;

Meio: Cabo de duplo par trançado blindado;

Distância máxima: 100 m por segmento;

Taxa de comunicação selecionável: valores apresentados na Tabela 5.

Tabela 5 – Taxa de comunicação selecionável [36].

TAXA DE DADOS (kbps)	125	250	500
Comprimento máximo de tronco da rede (m)	500	250	100
Comprimento máximo das derivações (m)	156	78	39

Exemplo de topologia

Na Figura 16 é representado um exemplo de uma topologia de rede *DeviceNet*. Esta tecnologia usa uma topologia de rede do tipo derivação que permite que tanto a ligação de sinal quanto a de alimentação estejam presentes no mesmo cabo.

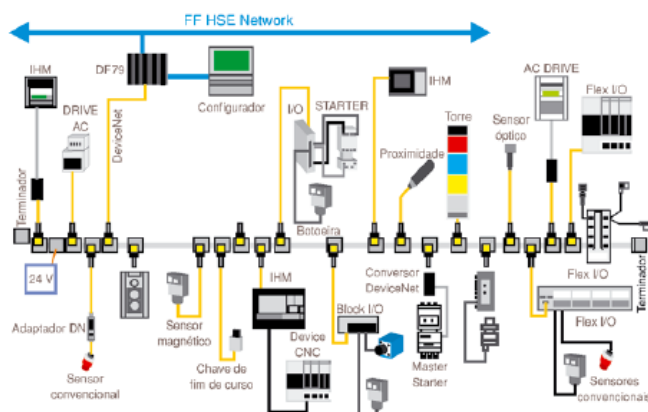


Figura 16 – Imagem ilustrativa da topologia *DeviceNet* [36].

2.3.9 CANopen

Histórico

A rede CAN teve a origem em 1983, foi inicialmente desenvolvida pela Bosh para a indústria automóvel, mas a sua introdução oficial ocorreu em 1986. Em 1991 são publicadas as suas especificações.

O protocolo CANopen foi publicado em 1995 pela CiA, associação criada em 1991 que elaborou e definiu as ferramentas a utilizar, assegurando a interoperacionalidade dos equipamentos. [21]

Descrição

O CAN é um *bus* de comunicação que funciona baseado no conceito multi-mestre, possibilitando a todos os módulos alternarem entre mestres ou escravos num determinado momento. Trata-se de um protocolo bastante utilizado na indústria, devido à sua versatilidade e baixo custo. Suporta uma topologia livre, em *bus* ou com derivações. Esta tecnologia não identifica os equipamentos mas sim as mensagens sendo que cada equipamento pode enviar mensagens sempre que o *bus* estiver disponível.

Este sistema apenas define os níveis 1 e 2 do modelo OSI, sendo a comunicação realizada por eventos, reduzindo assim o tráfego na rede. Possíveis conflitos gerados pelo tráfego na rede são evitados através da definição de níveis de prioridade.

O CANopen tem como aplicação típica todo e qualquer cenário em que se torne vantajoso a utilização de um único cabo para a ligação a um conjunto de sensores e atuadores, leitores de códigos de barras, variadores de velocidade, interfaces para o operador, etc.

De entre as possíveis aplicações refira-se o controlo de movimento, a robótica, os aparelhos médicos, os transportes e as linhas de produção, apesar da sua área de aplicação geral ser o controlo de máquinas descentralizadas.

O CAN é uma rede confiável e apresenta um índice muito baixo de erros de transmissão devido à implementação de diversos mecanismos de controlo de erros. [21]

Modo de Comunicação

A rede CANopen é uma rede baseada em CAN, o que significa que esta utiliza as tramas de dados CAN para a troca de informação na rede.

Os dados na rede CAN são transmitidos através de uma trama de dados. Este tipo de trama é composto principalmente por um campo identificador de 11 bits (*arbitration field*), e um campo de dados (*data field*), que pode conter até 8 bytes de dados.

Numa rede CAN, qualquer elemento da rede pode tentar transmitir num determinado instante. Numa possível situação em que dois elementos tentam transmitir ao mesmo tempo, só conseguirá transmitir aquele que enviar a mensagem mais prioritária. A prioridade da mensagem é definida pelo identificador da trama CAN. [37]

Características Físicas

Topologia: Linear, com um sinal diferencial em dois condutores e um comum;

Número máximo de equipamentos: 126 máximo;

Meio: Cabo trançado com dois pares; CAN-H, CAN-L e CAN-GND;

Distância máxima: 1000 m; [21]

Taxa de comunicação selecionável: É possível visualizar os valores na

Tabela 6.

Tabela 6 – Relação distância/débito suportadas [37].

Distância (m)	1000	600	500	250	100	50	25
Velocidade (Kb/s)	50	100	125	250	500	800	1000

Exemplo de topologia

A Figura 17 apresenta um exemplo de uma topologia de rede CAN, esta pode assumir topologias em linha.

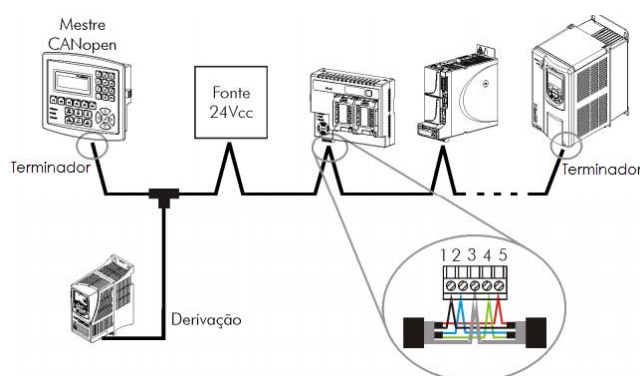
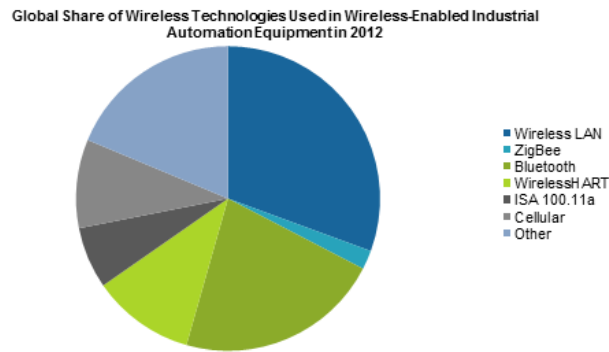


Figura 17 – Exemplo de utilização da topologia CAN [37].

2.4 Sistemas Sem Fios

2.4.1 Principais tipos de redes sem fios

Um estudo de mercado das tecnologias *wireless*, efetuado em 2013, determinou quais as tecnologias mais utilizadas e a Figura 18 demonstra o resultado obtido.



Source: IHS Inc. July 2013

Figura 18 – Análise de mercado das tecnologias wireless [38].

As redes wireless são normalmente divididas em quatro grupos específicos e a sua distinção é baseada na sua área de abrangência: rede pessoal ou de curta distância tem a designação de WPAN, rede local é designada de WLAN, rede metropolitana têm a designação de WMAN e por fim a rede geograficamente distribuída ou de maior distância é designada de WWAN. Em seguida é feito uma classificação detalhada dos quatro grupos de redes sem fios.

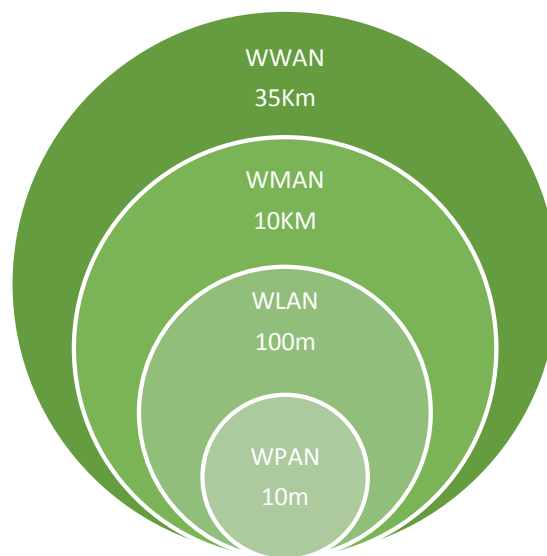


Figura 19 – Divisão das redes *wireless*. [76]

- *WPAN* é uma tecnologia de pequeno alcance (até 10m) e de baixa taxa de transmissão. É um padrão para redes que interligam dispositivos pessoais ou redes de sensores sem fios, definido pelo padrão IEEE 802.15. Entre esses padrões, destacam-se o IEEE

802.15.1 (*Bluetooth*), IEEE 802.15.3 (*Ultra Wide Band - UWB*) e o IEEE 802.15.4 (*ZigBee, Wireless Hart e ISA100*).

- *WLAN* é a tecnologia destinada a interligação de redes locais com alcance de 100 metros, também conhecidas por *Wi-Fi (Wireless Fidelity)*. Trata-se de padrão existente como extensão ou alternativa para as redes com cabeamento convencional *Ethernet* (Par metálico ou fibra ótica), definido pelo IEEE 802.11x (onde o x equivale ao tipo de rede: a, b, g, i ou n).
- *WMAN*, tecnologia que trata dos acessos de banda larga para a última malha em redes metropolitanas, com alcance em torno de 10 km, definida pelo padrão IEEE 802.16 (*WIMAX*).
- *WWAN* é uma tecnologia voltada para redes de longa distância em telecomunicações, atendendo aos serviços de voz e a alguns serviços de dados, definidas pelo padrão IEEE 802.20 *Mobile Broadband Wireless Access (MBWA)*. [39]

2.4.2 *WI-FI*

Um consórcio internacional de especialistas em Engenharia, de diversas empresas de tecnologia trabalharam em conjunto com a IEEE, por volta da década de 1990, a fim de desenvolverem os padrões sem fios para a indústria por forma a determinar como é que os novos produtos sem fios iriam interagir uns com os demais. Paralelamente nascia a *Wi-Fi Alliance* que acabaria por utilizar essas normas para em todo o mundo testar e certificar que os produtos cumprem os padrões de interoperabilidade e segurança definidos. O protocolo *Wi-Fi* (Figura 20), nome com que ficou conhecido o sistema *Ethernet Wireless*, é a tecnologia que permite interligar redes locais e dispositivos. [40]

Com esta tecnologia é possível criar redes locais sem fios com elevado débito desde que não se verifiquem distâncias demasiado grandes entre o dispositivo *wireless* e o ponto de acesso.



Figura 20 – Rede Wi-Fi [41].

Esta tecnologia não precisa de licença de instalação nem de operação e permite operar nas faixas de frequência de 2.4 GHz ou 5 GHz (Tabela 7), proporcionando uma velocidade de transmissão de dados igual ou superior a 11 Mbps para um comprimento máximo de 30 m. Essa velocidade é limitada pela largura de banda disponível [40]. [42]

Sabe-se que as redes sem fios são mais lentas do que as cabladas, por isso os padrões dessas redes devem validar cuidadosamente os dados recebidos de forma a evitar as perdas de dados existentes devido à não confiabilidade no meio ambiente em que se encontram. Para obtermos uma comunicação confiável tem de ser levado em conta o ambiente hostil onde os equipamentos se encontram a operar. Assim, deve ser considerado que os equipamentos estão sujeitos a pó, poeira, calor e humidade para que o valor de *Mean Time Between Failures* (MTBF) possa ser alto e o valor de *Mean Time to Repair* (MTTR) seja o menor possível. [40]

Tabela 7 – Características da norma 802.11.

Padrão	Banda de frequência	Débito	Alcance
Wi-Fi A (802.11a)	5 GHz	54 Mbit/s	10 m
Wi-Fi B (802.11b)	2.4 GHz	11 Mbit/s	100 m
Wi-Fi G (802.11g)	2.4 GHz	54 Mbit/s	100 m

2.4.3 BLUETOOTH

A empresa Ericsson começou em 1994, a estudar a viabilidade de desenvolvimento de uma tecnologia para a comunicação entre telemóveis e os seus acessórios de baixo custo utilizando sinais de rádio. Do estudo efetuado resultou um sistema de rádio de curto alcance que recebeu o nome MCLink. Por volta de 1998, algumas empresas de renome (Ericsson, Intel, IBM, Toshiba e Nokia) uniram-se e criaram a associação Bluetooth SIG, o que permitiu o desenvolvimento de padrões que garantem o uso e a interoperabilidade da tecnologia em diversos dispositivos. [43]

O protocolo Bluetooth foi desenvolvido na banda de frequências sem licença Industrial, *Scientific and Medical* (ISM). A comunicação é implementada usando a técnica de modulação *Frequency Hopping Code Division Multiple Access* (FH-CDMA). Assim, o dispositivo que estabelece a conexão salta de um canal para outro de forma muito rápida, evitando interferência com outros protocolos, sendo possível utilizar até 79 frequências, dentro da faixa ISM, cada uma espaçada da outra de 1 MHz. [43]

Um dispositivo conectado por Bluetooth recebe e transmite dados de forma alternada entre *slots* para transmissão e *slots* para receção. O Bluetooth é capaz de transmitir dados e voz.

O dispositivo possui dois tipos de conexão: *Synchronous Connection Oriented Link* (SCO) e *Asynchronous Connection-Less* (ACL), o primeiro estabelece um *link* sincronizado entre o dispositivo mestre e o dispositivo escravo, onde é feita uma reserva de *slots* para cada um, e por isso acaba por ser mais utilizado em aplicações de envio contínuo de dados, como por exemplo, voz. O padrão ACL estabelece um *link* entre um dispositivo mestre e os dispositivos escravos existentes na rede, sendo esse *link* assíncrono, já que utiliza os *slots* previamente livres. Ao contrário do SCO, o ACL permite o reenvio de pacotes de dados perdidos, garantindo a integridade das informações trocadas entre dispositivos. A Figura 21 representa a diversidade de dispositivos a que é possível se conectar via Bluetooth. [43]

A tecnologia Bluetooth está em constante evolução, fazendo com que as suas especificações sofram alterações e que novas versões surjam com o passar do tempo. A versão mais atual da tecnologia Bluetooth é a v4.2.



Figura 21 – Possíveis dispositivos numa rede Bluetooth [44].

2.4.4 ZIGBEE

Homologado em maio de 2003, o protocolo *ZigBee* (IEEE 802.15.4) foi desenvolvido pela *ZigBee Alliance*. É uma tecnologia relativamente simples, que utiliza um protocolo de pacotes de dados com características específicas, sendo projetado para oferecer flexibilidade perante os tipos de dispositivos que pode controlar. [45]

O *ZigBee* permite comunicações robustas operando nas bandas de frequência não licenciadas ISM, a 868 MHz (1 canal), a 915 MHz (10 canais) e a 2.4 GHz (16 canais). Este protocolo oferece uma boa imunidade contra interferências, assim como a capacidade de hospedar milhares de dispositivos numa rede (sensivelmente 65.000), com taxas de transferências de dados que podem variar entre 20 kbps e os 250 kbps. Presentemente é possível adquirir diversos módulos, os quais foram desenvolvidos para economizar ao máximo a energia consumida. Quando aplicados em sensores remotos, são alimentados a pilhas ou a baterias comuns, a

durabilidade esperada é de meses ou até mesmo anos sem que seja necessário proceder a sua substituição. Isto é possível pois os módulos *ZigBee*, quando não estão em operação, estão em modo “*sleep*”, consumindo o mínimo de energia possível. [45] [46]

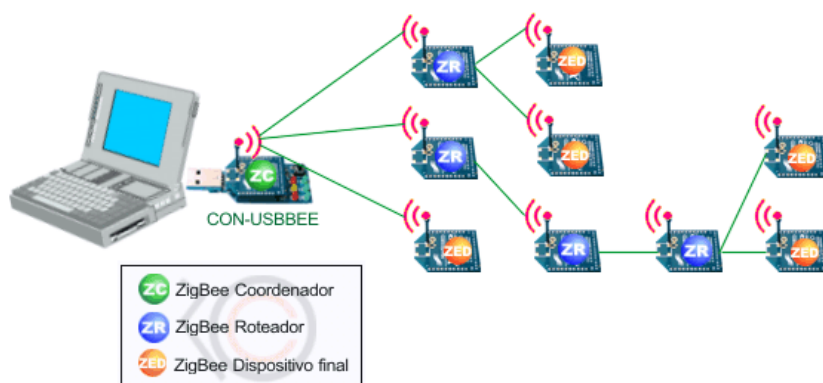


Figura 22 – Exemplo de uma rede ZigBee [46].

A Figura 22 ilustra um exemplo clássico de uma rede ZigBee, onde diversos dispositivos são responsáveis, por exemplo, por monitorizar a temperatura, controlar válvulas, sistemas de ventilação e iluminação.

Para garantir a interoperabilidade e a padronização entre fabricantes, a IEEE definiu que uma rede ZigBee é constituída por dois tipos de dispositivos:

- Full Function Device (FFD), que pode ser um coordenador ou um roteador da rede, ambos possuem as mesmas características físicas e as mesmas funções;
- Reduced Function Device (RFD), dispositivo mais económicos e de construção mais simples e que não atua como coordenador ou roteador da rede, uma vez que só comunica com um coordenador ou roteador ao qual esta associado, sendo conhecido como escravo.

As redes ZigBee podem ser classificadas quanto a topologia: estrela; árvore e malha. A Figura 23 apresenta exemplos das tecnologias mencionadas.

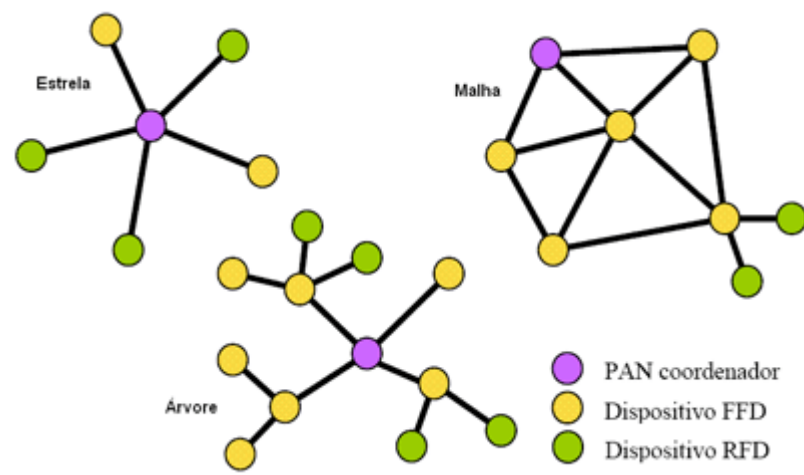


Figura 23 – Exemplos de topologias de uma rede ZigBee [47].

2.4.5 WIRELESS HART

O protocolo WirelessHART é baseado no HART 7, o primeiro padrão aberto de comunicação sem fio desenvolvido com o intuito de atender especificamente às necessidades da indústria de processo, fornecendo uma comunicação sem fios robusta. Com base no protocolo de comunicação *Highway Addressable Remote Transducer* (HART), a WirelessHART garante de aos seus utilizadores os benefícios da tecnologia sem fios rápida, mantendo a compatibilidade com dispositivos já existentes.

Desenvolvida pela *HART Communication Foundation*, a tecnologia WirelessHART trabalha na frequência dos 2.4 GHz ISM usando a técnica de acesso múltiplo TDMA [48] a fim de sincronizar a comunicação entre os diversos equipamentos na rede. A comunicação numa rede HART é feita dentro de uma janela temporal de 10 ms. Esta tecnologia suporta a comutação de canais com o intuito de evitar interferências e reduzir os efeitos de desvanecimento multi-percurso. O protocolo HART foi elaborado tendo como base o protocolo OSI. [49]

A rede de comunicação WirelessHART é por norma estruturada em malhas, onde cada sensor funciona como um *router* ou como um repetidor, assim o alcance de uma rede não fica dependente de um *gateway* central, o que permite a configuração de uma vasta estrutura de rede distribuída. Desta forma é garantida a integridade da rede WirelessHART, pois numa situação de falha ou obstrução que possa causar a interrupção de um caminho de comunicação, o sistema pode estabelecer uma nova rota alternativa, aumentando e garantindo assim a disponibilidade da rede. A rede WirelessHART pode tomar a forma de estrela (*star*), malha (*mesh*) e estrela-malha (*star-mesh*), como se pode observar na Figura 24.

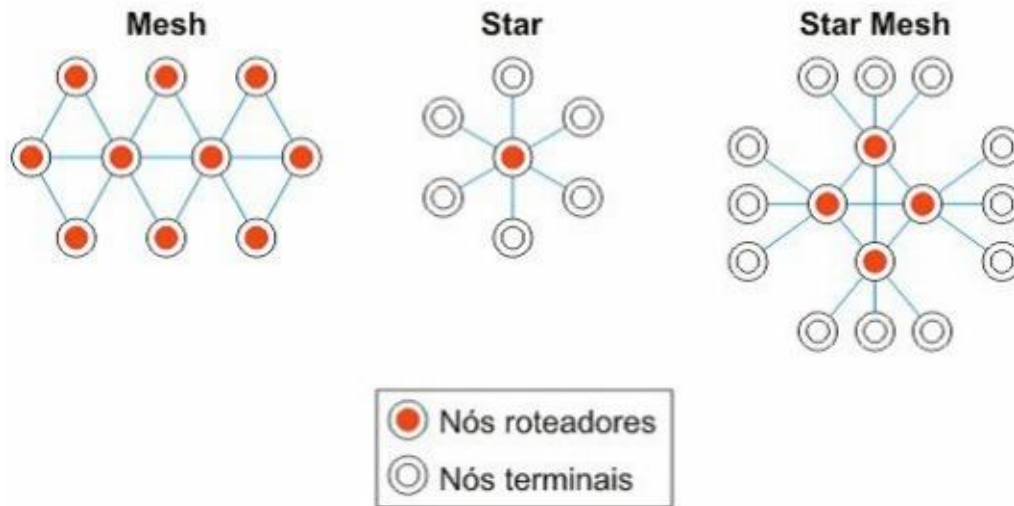


Figura 24 – Possíveis topologias de uma rede WirelessHART [50].

O dispositivo WirelessHART é autónomo, eliminando as ligações analógicas para o sistema de controlo, podendo ser instalado em qualquer local sem necessidade de fios. Uma rede WirelessHART (Figura 25) é constituída por três dispositivos principais [50]:

- **Wireless Field devices:** Equipamentos de campo, sem fios conectados ao processo ou outros equipamentos já instalados. Podem ser dispositivos WirelessHART, ou qualquer outro dispositivo de campo HART existente que possua um adaptador WirelessHART acoplado.
- **Gateways:** Os gateways permitem a comunicação entre os equipamentos de campo e as aplicações de controlo.
- **Network Manager:** Responsável pela configuração da rede, gerindo a comunicação entre os diversos dispositivos, rotas de comunicação e monitorizando o estado da rede. Este pode ser integrado no *gateway*, numa aplicação *host* ou num controlador de processos.

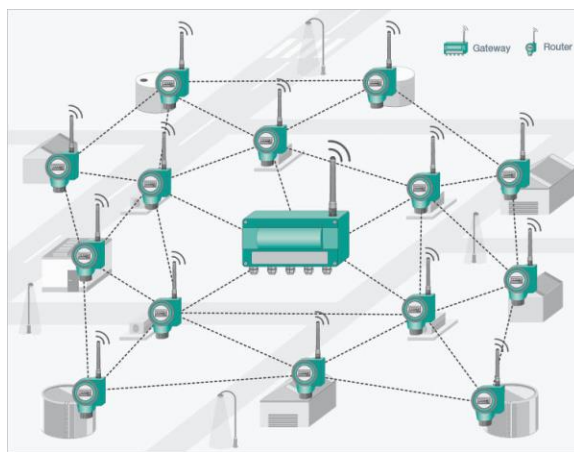


Figura 25 – Estrutura de uma rede WirelessHART [49].

2.4.6 ISA100

Desenvolvido pela comissão SP100 da ISA, ISA100 é um padrão de rede sem fios industrial. O padrão foi desenvolvido fundamentalmente para atender às exigências dos utilizadores finais e para funcionar num ambiente industrial hostil, suportando os principais protocolos utilizados, tais como, HART, Profibus, CIP e Foundation Fieldbus no topo da sua camada de aplicação. Uma rede ISA100 típica é apresentada na Figura 26. Os dispositivos dispostos na rede são caracterizados por regras bem definidas que controlam o seu funcionamento. No geral os dispositivos podem assumir as seguintes funções: gerente do sistema, gerente de segurança, *gateway*, roteador *backbone*, roteador, dispositivo I/O, dispositivo portátil [51].

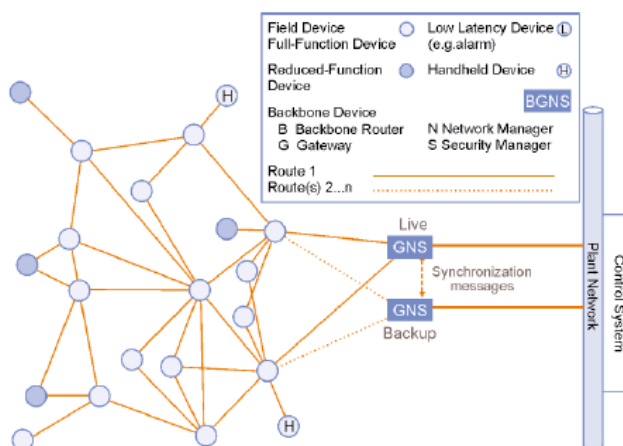


Figura 26 – Exemplo de uma rede ISA100 [52].

O padrão ISA100 baseia-se numa arquitetura de uma versão simplificada do modelo OSI, o qual contempla apenas 5 camadas.

Semelhante ao padrão *WirelessHART*, a camada física do padrão ISA100 é baseada no IEEE 802.15.4. As camadas físicas de ambos os padrões têm semelhanças, no entanto, o ISA100 adiciona algumas particularidades. O ISA100 permite a utilização de 16 canais (11-26) apesar do canal 26 ser opcional. Em relação ao alcance, uma vez que o padrão não especifica nenhum parâmetro e devido à sua compatibilidade com o IEEE 802.15.4, era de esperar que tivesse um alcance de até 100 m.

2.4.7 WISA

Lançado pela *ABB* e apresentado na *IEEE-Industrial Electronics Magazine* [53], *WISA* é uma interface sem fios projetada para sensores e atuadores que pretende cumprir os requisitos das aplicações de automação, em tempo real, particularmente na robótica. A comunicação é baseada no padrão IEEE 802.15.1, na banda livre de licença à frequência de 2.4 GHz. Os dispositivos de campo que fazem parte desta rede comunicam utilizando tecnologia rádio com módulos de entrada/saída que transmitem/recebem os sinais. Cada dispositivo de entrada poderá processar informação num limite máximo de 120 sensores [54], decorrendo um máximo de 20 ms por trama de comunicação. Esta tecnologia *wireless* permite reconhecer falhas através de sinais de diagnóstico que são periodicamente enviados de todos os dispositivos de campo conectados à rede. A ligação entre dispositivos de entrada/saída e o controlador é efetuada via barramento de campo. A figura 26 apresenta os dispositivos que podem fazer parte de uma aplicação *WISA* com a alimentação, dispositivos de entrada/saída, sensores e atuadores.



Figura 27 – Visão geral e dispositivos WISA [55].

2.4.8 Comparativo

A Tabela 8 apresenta uma comparação entre as diversas redes sem fios estudadas no presente capítulo apresentando os valores característicos de diversos parâmetros.

Tabela 8 – Tabela comparativa de diversas normas IEEE sem fios.

Norma IEEE (Nome de Mercado)	802.15.1 (Bluetooth)	802.11 (Wi-Fi)	802.15.4 (ZigBee)	802.15.4 (Wireless Hart)	802.15.1 (WISA)
Aplicações principais	Suprimir excesso de cabos	<i>Ethernet</i> industrial	Controlo e Monitorização	Medições e processos de controlo	Interface sensores atuadores
Frequência	2.4GHz				
Taxa de comunicação	1-3 Mbps	11 Mbps	20-250 kbps	11 Mbps	1 Mbps
Distancias máximas	30 m (Classe2) 100m+ (Classe1)	100 m+ (Antenas direcionais)	30 m-70 m, 100 m+ (com amplificador externo)	100 m, até 2000 m com repetidor	5m (ambiente industrial; tipicamente 10 m)
Número de dispositivos	7	32	2*64	250	120
Autonomia da bateria	1 a 7	0.5 a 5	100 – 1000+	3.5 Anos: taxa de trans. De 30 min.	3 a 4 anos
Consumo na transmissão	45 mA (Classe2) <150 mA (Classe1)	300 mA	30 mA	150-300 mA	200 mA
Adequação para, controlo e supervisão.	Baixa (Boa média, mas conexão inicial lenta)	Baixa (Taxa alta, mas conexão inicial lenta)	Baixa (Bom compromisso entre taxa e custo de conexão)	Alto (Bom compromisso entre taxa e custo de conexão)	Alto (Boa taxa e custo de conexão)
Método de transmissão	FHSS	DSSS	DSSS	FHSS/DSSS	FHSS/DSSS
Vantagens	Custo e flexibilidade	Velocidade e flexibilidade	Potência e custo	Flexibilidade, Potência e velocidade	Velocidade e custo

2.5 Conclusão

Em virtude da informação mencionada neste capítulo é possível afirmar que não existe uma tecnologia de comunicação que seja superior em todos os aspetos em relação a outra tecnologia, cada uma tem as suas vantagens e os seus pontos fracos. Por exemplo o ModBus é uma tecnologia versátil pois foi desenvolvida com base no protocolo OSI e apresenta um potencial de utilização enorme, contudo acresce ao seu custo de instalação o custo de cabeamento de toda a rede. Ainda dentro dos sistemas cablados, o CANopen é uma tecnologia voltada para a indústria, capaz de suportar uma topologia livre e de custo reduzido, porém também acrescenta custos de cabeamento. Já nas tecnologias sem fios, por exemplo, o 802.15.1, mais conhecido no mercado por Bluetooth é uma tecnologia de comunicação rádio de baixo custo e desenvolvida para transmissão de dados e voz, sendo a sua maior desvantagem o curto alcance. Já a tecnologia Wi-Fi apresenta como vantagem a possibilidade de se interligar redes locais

(livres de licença de instalação e operação) e dispositivos sem a necessidade de utilizar qualquer fio.

Tendo em conta estes factos as tecnologias de comunicação eleitas para a elaboração deste projeto foram o Bluetooth para uma monitorização local e rápida dos motores e a comunicação rádio frequência que ficará responsável por transportar os dados dos motores para um dispositivo central que por sua vez ficara encarregue de armazenar toda a informação numa base de dados. Ambas as tecnologias apresentaram a melhor relação custo benefício.

Capítulo 3

SOLUÇÃO TECNOLÓGICA SELECIONADA

3.1 Introdução

Nos capítulos anteriores foi referida a necessidade de desenvolver de um dispositivo que ofereça simplicidade e rapidez no controlo e monitorização de motores. Foram apresentadas também as opções disponíveis no mercado para o transporte de dados num ambiente industrial, que possibilitam a interligação entre dispositivos de campo e equipamentos de controlo e monitorização. São agora apresentados, no capítulo 3, as plataformas eletrónicas *open-source* mais conhecidas e empregadas atualmente. Tendo em conta a vulgarização dos chips *ARM* e o seu reduzido custo, apareceram no mercado uma série de dispositivos que, de uma forma em geral, se podem dividir em dois grupos: os mini-PC's e as placas de desenvolvimento. Os primeiros são mais orientados para o público em geral, apresentando como principal função o entretenimento. Já o segundo grupo está mais direcionado para aplicações didáticas, pois vêm equipados com uma série de pinos de entrada/saída permitindo, por exemplo, a realização de projetos na área da robótica.

Neste capítulo é feito também a seleção da tecnologia a utilizar no desenvolvimento do projeto. As escolhas recaíram sobre as placas da STMicroelectronics (STM32DISCOVERY e STM32DIS-BB) assim como pelo módulo de comunicação Bluetooth HC-05 e pelo módulo RF nRF24L01+ (que numa fase mais avançada substitui o módulo RFM12b).

Por fim são apresentados os *softwares*, tecnologias e ferramentas utilizadas no desenvolvimento da aplicação *web*, tais como: *XAMPP*, *MySQL Workbench 6.2 CE*, *Notepad++*, *Apache*, *PHP*, *HTML*, *SQL*, *CSS* e *JavaScript*.

3.2 Unidades Microcontroladores

De entre as diversas plataformas disponíveis no mercado, são analisadas em pormenor as três principais plataformas eletrónicas de desenvolvimento e apresentadas três outras suas concorrentes.

3.2.1 Arduíno

O Arduino, representado na Figura 28, é uma plataforma de *hardware* e *software* livre, criado em 2005 por Massimo Banzi e David Cuartielles em Itália com o objetivo de permitir o desenvolvimento de controlo de sistemas interativos, de baixo custo e acessível a todos. A placa foi equipada com microcontroladores Atmel AVR 8 bits ou Atmel ARM 32 bits permitindo a amadores, artistas, designers, entre outros, a criação de aplicações, objetos e ambientes interativos. Através desta placa simples é possível integrar facilmente sensores como por exemplo acelerómetros, sensores de ultra-sons, sensores de pressão assim como é possível também controlar diversos dispositivos: luzes, motores, etc. Com o Arduino é possível enviar e receber informação de praticamente qualquer outro sistema eletrónico.



Figura 28 – Arduíno UNO.

Outra característica importante deste tipo de sistemas é que todo o material disponível incluindo o software e bibliotecas (software, bibliotecas, *hardware*) é open-source, isto é, pode ser reproduzido e usado por todos sem a necessidade de pagamento de royalties ou direitos de autor. A plataforma é constituída essencialmente por duas partes: o *hardware* e o *software*.

- **O Hardware**

Baseado no microcontrolador ATmega328, a placa Arduino UNO possui 14 entradas/saídas digitais (das quais 6 podem ser usadas como saídas PWM), 6 entradas analógicas, um oscilador baseado num ressonador cerâmico de 16 MHz, uma porta USB, um *power jack*, um *header In Circuit Serial Programming* (ICSP) e um botão de *reset*. Contém tudo o que é necessário para suportar o microcontrolador, basta conectá-lo a um computador com o cabo USB ou ligá-lo com um adaptador AC para DC ou até uma bateria. O Arduino UNO já foi alvo de três revisões. A Tabela 9 apresenta as principais características do Arduino UNO.

Tabela 9 – Características físicas do Arduino Uno [56].

Parâmetros	Características
Microcontroller	ATmega328
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

- **O Software**

O Arduino possui um compilador *GNU Compiler Collection* (GCC) (linguagem C e C++) baseado em *Wiring* e usa uma interface gráfica construída em Java baseado no projeto *Processing*. Tudo isto resulta num (IDE) muito simples de se utilizar, como se pode observar na Figura 29 para além disso possui diversas bibliotecas as quais são disponibilizadas gratuitamente.

O código é compilado usando o Integrated Development Environment (IDE), sendo de seguida enviado para a placa onde é gravado no *chip* controlador. O *software* de microcontrolador que é descarregado na placa é denominado de *firmware*.

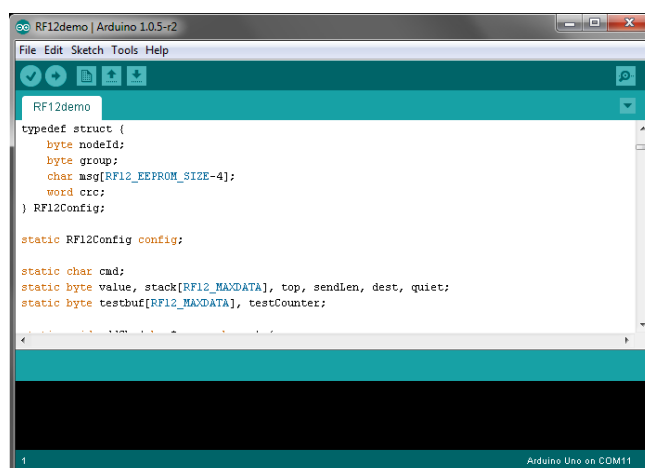


Figura 29 – Ambiente de desenvolvimento do Arduino.

Para além das diversas placas com diferentes microcontroladores e *pinouts* foram também lançadas, e desenvolvidos diversos *shields*, acessórios e um *kit starter*.

De seguida enumera-se as diversas placas, *shields* e acessórios presentes no mercado.

- **Placas:** Arduino Uno, Arduino Leonardo, Arduino Due, Arduino Yún, Arduino Tre, Arduino Micro, Arduino Robot, Arduino Esplora, Arduino Mega ADK, Arduino *Ethernet*, Arduino Mega 2560, Arduino Mini, LilyPad Arduino USB, LilyPad Arduino Simple, LilyPad Arduino SimpleSnap, LilyPad Arduino, Arduino Nano, Arduino Pro Mini, Arduino Pro e Arduino Fio;
- **Shields:** Arduino *GDM Shield*, Arduino *Ethernet Shield*, Arduino *Wifi Shield*, Arduino *Wireless SD Shield*, Arduino *USB Host Shield*, Arduino *Motor Shield*, Arduino *Wireless Proto Shield* e Arduino *Proto Shield*.
- **Acessórios:** TFT LCD screen, USB/Serial Light Adapter, Arduino ISP, Mini USB/Serial Adapter.

3.2.2 Raspberry Pi

O Raspberry Pi, representado na Figura 30, é essencialmente um computador do tamanho de um cartão de crédito desenvolvido no Reino Unido pela Fundação Raspberry Pi. Constituído por uma única placa foi desenvolvido com o intuito de promover o ensino de ciências da computação ao nível escolar. É uma plataforma relativamente fácil de se utilizar e tem uma infinidade de utilizações, nas mais diversas áreas. Para interagir com um Raspberry Pi é necessário uma ligação entre o mesmo e um monitor, um rato e um teclado. De uma forma similar ao Arduino também o Raspberry Pi é constituído essencialmente por duas partes: o *hardware* e o *software*.



Figura 30 – Raspberry Pi modelo B+.

- **Hardware**

No mercado estão disponíveis dois modelos, A e B, os quais possuem as características descritas na Tabela 10. O modelo A tem 256 MB de RAM, uma porta USB e não suporta ligação *Ethernet*. O modelo B tem 512 MB de RAM, duas portas USB e uma ligação *Ethernet*. Ambos os modelos não possuem uma memória não-volátil (por exemplo um disco rígido) mas possuem uma entrada de cartão SD para armazenamento de dados. Na Tabela 10 é apresentado um resumo das características físicas dos modelos Raspberry Pi A, B e C.

Tabela 10 – Características físicas dos modelos Raspberry Pi [57] [58].

Parâmetros	Modelo A	Modelo B	Modelo B+
SoC	Broadcom BCM2835 (CPU, GPU, DSP e SDRAM)		
CPU	700 MHz ARM1176JZF-S core		
GPU	Open GL ES 2.0, OpenVG e 1080p30 H.264 high-profile decode		
Memória (SDRAM)	256MB	512MB	
Portas USB 2.0	1	2	4
Saídas de Vídeo	RCA Composto (PAL & NTSC), HDMI e painéis LCD via DSI.		
Saídas Áudio	Conector de 3.5mm, HDMI		
Armazenamento	SDcard		MicroSD
Rede onboard	Nenhuma	10/100 BaseT <i>Ethernet</i> (RJ45)	
Periféricos de baixo nível	GPIO, UART, I ² C e SPI		
Power Draw / voltage	600mA a 1.2A @ 5V	750mA a 1.2A @ 5V	600mA a 1.8A @ 5V
GPIO	26		40
Tamanho	85.6 x 53.98 x 17mm		85 x 56 x 17mm

- **Software**

A Raspberry Pi Foundation disponibiliza no *site* do RaspBerry Pi, diversos SO's que podem ser descarregados e utilizados por qualquer um, dependendo da aplicação que se pretende dar ao Raspberry. A forma mais comum de fazer “boot” do Raspberry Pi é através de um sistema operativo instalado num cartão SD.

Um sistema operativo indicado para iniciantes é o Raspbian. A interface do sistema operativo Raspbian é apresentada na Figura 31. Este SO é baseado no Linux (Debian) e otimizado para o Hardware do Raspberry Pi. É uma opção muito comum para principiantes pois tem um conjunto de programas e utensílios básicos que facilitam a utilização do mesmo. Beneficia dele quem pretende programar, navegar na internet, gerir/editar ficheiros entre outras funcionalidades. Na vasta lista de SO's baseados em Linux estão presentes também o Pidora e o Archlinux [59].

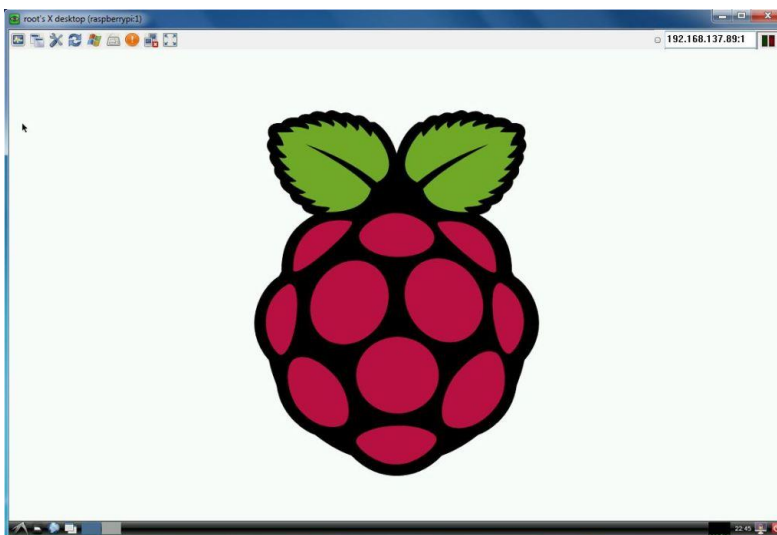


Figura 31 – SO Raspbian interface [60].

Sistemas operativos como o RaspBMC e OpenELEC foram desenvolvidos com o intuito de servirem como um centro de média bastante completo, sendo possível aceder a ficheiros de imagem, musica e vídeo, assim como ver/gravar *stream* de vídeos, as suas características assemelham-se a uma espécie de box de televisão. Ambos são baseados no programa XBMC. A Figura 32 apresenta a interface do SO RaspBMC.



Figura 32 – Interface do SO RaspBMC [60].

3.2.3 STM 32-bit ARM

A *STMicroelectronics* é uma empresa mundialmente conhecida pelo seu amplo portfólio de soluções inovadoras de semicondutores. Na sua vasta lista de produtos está a conceituadíssima gama de microcontroladores que vai desde o robusto e de baixo custo 8-bit MCUs até ao 32-bit ARM-based Cortex-M0 e M0+, Cortex-M3 e o microcontrolador Flash Cortex-M4 com uma

grande variedade de periféricos. A Figura 33 apresenta uma análise comparativa dos microcontroladores da STM em função de um conjunto de parâmetros significativos.



Figura 33 – Comparação dos desempenhos dos diversos microcontroladores [61].

A família *STM32* é composta pelos microcontroladores *flash* de 32 bits baseados no processador *ARM Cortex-M* projetado e desenvolvido para oferecer novos níveis de liberdade para os utilizadores de *MCUs*. Esta gama de produtos de 32 bits alia o baixo consumo de energia com alto desempenho, à capacidade de realizar tarefas em tempo real com o processamento de sinais digitais a baixas tensões. Por se tratar de dispositivos baseados nos núcleos padrão da indústria e por se fazerem acompanhar por uma variedade de escolhas entre ferramentas e *software*, os dispositivos *STM32* são uma família de produtos que representam a escolha ideal, tanto para pequenos projetos como para aplicações que necessitem de plataformas de decisões mais complexas.

Na Figura 34, são apresentadas as diferentes versões da família *STM32 32-bit ARM Cortex MCU's*. Está dividida pelas seguintes séries, do processador de menor desempenho para o maior: *STM32L0*; *STM32L1*, *STM32F0*; *STM32F1*; *STM32F2*; *STM32F3*; *STM32F4*.



Figura 34 – Desempenho dos processadores da família STM32 da STMicroelectronics [62].

• **Hardware**

A *STMicroelectronic* disponibiliza para os seus clientes um portfólio de microcontroladores diversificado e de qualidade. Como foi referido anteriormente, na família *STM32* existe uma grande variedade de opções, sendo estas comparadas na Tabela 11. Para além das diversas series das placas *STM32* disponibilizadas pela *ST*, também existe todo um ecossistema de *kits* e expansões que podem acrescentar ainda mais funções às placas principais.

Tabela 11 – Comparação das placas da família STM32 da STMicroelectronics [63].

STM32F4	STM32F3	STM32F2	STM32F1	STM32F0	STM32L1	STM32L0
Até 180MHz Cortex-M4 DSP/FPU	72MHz Cortex-M4 DSP/FPU	120MHz Cortex-M3 CPU	Até 72 MHz Cortex-M3 CPU	48MHz Cortex-M0 CPU	32MHz Cortex-M3 CPU	32 MHz Cortex-M0+ CPU
Até 2-Mbyte Flash	Até 512-Kbyte Flash	Até 1-Mbyte Flash	Até 1-Mbyte Flash	Até 128-Kbyte Flash	Até 512-Kbyte Flash	Até 64-Kbyte Flash
Até 256-Kbyte SRAM	Até 64-Kbyte SRAM	Até 128-Kbyte SRAM	Até 96-Kbyte SRAM	Até 16-Kbyte SRAM 20-byte backup	Até 80-Kbyte SRAM	Até 8-Kbyte SRAM
2x USB 2.0 OTG FS/HS	USB FS/HS	2x USB 2.0 OTG FS/HS	USB FS/HS	USB Clock free	Até 16-Kbyte EEPROM	Até 2-Kbyte EEPROM
1x 12-bit AMC timer	3x 16-bit AMC timer (144MHz)	1x 12-bit AMC timer	1x 12-bit AMC timer	USB 2.0 FS Crystal less	USB2.0 FS device	USB 2.0 FS Crystal less
2x CAN 2.0B	CAN 2.0B	2x CAN 2.0B	Até 2x CAN 2.0B	CAN 2.0B	LCD 8x40 4x44	LCD 8x28 4x32

SDIO 2x FS Áudio Câmara IF	Até 7x comparador 4x 12-bit DAC 4x PGA	SDIO 2x FS Áudio Câmara IF	SDIO 2x FS Áudio	DAC comparator	OP-amps comparator	True RNG
<i>Ethernet</i> IEEE 1588	HDMI CEC	<i>Ethernet</i> IEEE 1588	<i>Ethernet</i> IEEE 1588	CEC	BOR MSI VScal AES 128-bit	BOR MSI VScal AES 128-bit
LCD-TFT SDRAM I/F	3x 16-bit $\Sigma\Delta$ ADC	Crypto				

USART,SPI,I2C, Timers, Reset, Watchdogs, Osciladores, Low voltage (3.6v) e sensor de temperatura

- **Software**

Os microcontroladores *STM32* podem ser programados usando a clássica linguagem C/C++, mas também é possível programar utilizando outros meios, tais como, *Java* ou *Matlab/Simulink*. Os parceiros da *ST* disponibilizam uma vasta oferta nesta família de microcontroladores, desde ferramentas de *software* de desenvolvimento (como por exemplo, *IAR Embedded Workbench EWARM IDE*, *Keil MDK-ARM uVision IDE* e o *GCC.based IDEs*) até soluções de *software* embutido (como por exemplo, *Alpwise*, com a sua solução para *Bluetooth*; *FreeRTOS*, o sistema operativo open-source em tempo real; *Micrium uC collection*, com o protocolo de internet *uC/TVP-IP*; *Express Logic*, com o sistema operativo em tempo real *TheadX* e por fim o *HCC*, com as bibliotecas para USB). A *ST* completa ainda as ofertas dos seus parceiros com um *software* dedicado para o desenvolvimento em *STM32*, o *software* livre *STMCube*.

3.2.4 *BeagleBoard*

Lançada nos Estados Unidos em 2008, *BeagleBoard.org* é uma comunidade open-source que fornece aos entusiastas de tecnologia os recursos de que precisam para desenvolver rapidamente novos produtos para o mercado e ao mesmo tempo reduzir os seus riscos. Devido ao sucesso das ferramentas de desenvolvimento *BeagleBoard* e *BeagleBoard-XM*, a *BeagleBoard.org* cria a ferramenta *BeagleBone*, uma plataforma de desenvolvimento aberta de *hardware* flexível e extensível, que facilita e potencia a transição de uma equipa desenvolvimento para o mundo open-source. A existência de uma vasta gama de interfaces e pilhas de comunicação como TCP/IP, USB e muitas outras funcionalidades foram também fatores que contribuíram para a sua popularidade. Outra vantagem existente é o suporte ao desenvolvimento em Linux, pois este fornece uma camada de abstração subjacente acima do *hardware*. Isto é, se ocorrer uma mudança para uma nova família de microcontroladores ou para uma nova geração de MCU's, não implicará ter de desenvolver novamente o *firmware*, a nível do sistema ou aplicativo de *software*.

A Beagleboard.org disponibiliza quatro dispositivos principais na sua lista de produtos *BeagleBoard*, sendo estes resumidamente apresentados na Tabela 12.

Tabela 12 – Comparação das placas BeagleBoard [64].

	BeagleBone Black	BeagleBone	BeagleBoard-XM	BeagleBoard
Processador	AM3358 ARM Cortex-A8	AM3348 ARM Cortex-A8	DM3730 ARM Cortex-A8	OMAP3530 ARM Cortex-A8
Velocidade Máxima do Processador	1GHz	720MHz	1GHz	720MHz
Pins Analógicos	7	7	0	0
Pins Digitais	65 (3.3V)	65 (3.3V)	53 (1.8V)	24 (1.8V)
Memoria	512MB DDR3, 2GB eMMC	256MB DDR2, SD	512MB LPDDR, SD	256MB LPDDR, SD
USB	USB 2.0 host port, USB 2.0 device port	USB 2.0 host port, USB 2.0 device port	4 USB 2.0 port	USB hostport, USB 2.0 OTG port
Vídeo	microHDMI, cape add-ons	cape add-ons	DVI-D e S-Video	DVI-D e S-Video
Áudio	microHDMI, cape add-ons	cape add-ons	3.5mm stereo jack	3.5mm stereo jack
Interface	4xUART, 8xPWM, LCD, GPMC, MMC1, 2xSPI, 2xI2C, A/D Converter, 2xCAN bus e 4 Timers	4xUART, 8xPWM, LCD, GPMC, MMC1, 2xSPI, 2xI2C, A/D Converter, 2xCAN bus, 4 Timers e FTDI USB to Serial JTAG via USB	McBSP, DSS, I2C, UART, LCD, McSPI, PWM, JTAG e Camera Interface	McBSP, DSS, I2C, UART, McSPI, PWM e JTAG

- **Software**

Tabela 13 – Características de *software* BeagleBoard

Parâmetros	Características
OS	Linux, Android, Windows CE
Language	English
Programming language	C, C++, Python, Perl, Ruby, Java, or shell script

Na Tabela 13 são apresentados algumas das características do software BeagleBoard.

3.2.5 *pcDuino*

No início do ano 2013 a *LinkSprite* lançou uma poderosa plataforma mini-PC que combina os benefícios de um ARM com o ecossistema Arduino: *pcDuino*, que representa PC + Arduino. *pcDuino* é pin-to-pin compatível com Arduino. Assim sendo os *Arduino Shields* podem ser instalados no *pcDuino* com uma simples placa de conexão (T-Board). Sendo uma placa

compatível com Arduino, pcDuino é uma plataforma que alia o poder do *software* livre Linux com o poder do *hardware* livre. Nas tabelas seguintes são apresentadas as especificações de *hardware* (Tabela 14) e *software* (Tabela 15) dessa placa.

- **Hardware**

Tabela 14 – Características físicas do pcDuino [65].

Parâmetros	Características
CPU	1 GHz ARM Cortex A8
GPU	OpenGL ES2.0, OpenVG 1.1 Mali 400 core
DRAM	1 GB
Onboard Storage	2 GB Flash, microSD card (TF) slot for up to 32 GB
Video Output	HDMI
Extension Interface	2.54 mm Headers
Network interface	10/100 Mbps RJ45 and USB WiFi extension (not included)
Power	5 V, 2000 mA
Overall Size	125 mm X 52 mm

- **Software**

Tabela 15 – Características de *software* do pcDuino.

Parâmetros	Características
Sistema Operativo	Ubuntu 12.04 e Android ICS 4.0
Linguagem	Inglês
API	Todos os Arduino Shield's são compatíveis, com acesso aos seguintes interfaces: UART, ADC, PWM, GPIO, I ² C e SPI
Linguagem de Programação	C, C++ with GNU tool chain, Java with standard Android SDK e Python

3.2.6 TinkerForge

O *TinkForge* introduziu o princípio dos Legos ao mundo da eletrónica. É uma plataforma *hardware* open-source de blocos microcontroladores empilháveis (*Bricks*) que podem controlar diferentes módulos (*Bricklets*). A interface de comunicação principal dos blocos de construção pode ser estendida usando uma extensão mestre. Esta plataforma apresenta como vantagem a possibilidade de controlar o *hardware* através de programas escritos em C, C++, C#, Object Pascal, Java, Perl, PHP, Python, Ruby, Shell and VB.NET sobre uma conexão USB, *Wi-Fi* ou *Ethernet* nos sistemas operativos *Windows*, *Linux* e *Mac OS X*. Esta abordagem de programação não embebida elimina requisitos típicos e limitações, como por exemplo, as ferramentas de desenvolvimento, disponibilidade limitada de memória RAM e poder de processamento.

O *hardware* e *software* de *TinkForge* são ambos open-source sendo que todos os arquivos estão hospedados no *GitHub*.

- **Hardware**

Os diversos produtos da *TinkForge* estão divididos nas seguintes categorias:

- *Os Bricks* são módulos 4x4 cm que podem ser controlados por dispositivos, tais como PC's, através da sua porta Mini-USB. Cada brick executa basicamente uma tarefa. Existem brick que podem executar tarefas de sensores (IMU Brick), comunicar (Master Brick) e controlar motores (DC Brick).
- *Os Bricklets* podem ser usados para estender as funções dos *Bricks*. Existem *Bricklets* que possibilitam a medição de um conjunto de grandeza tais como tensão, corrente, intensidade de luz ambiente e rotação bem como *Bricklets* para controle de reles de comutação, entradas/saídas digitais e saída alfanumérica em LCD's. Ao contrário dos *Bricks*, *Bricklets* não possuem um tamanho fixo.
- *Master Extension*se um *Master Brick* é utilizado sozinho ou numa pilha, a comunicação é feita através do seu interface USB. O *Master Extension* expande a interface de comunicação dos *Master Bricks*, através das opções por cabo (RS-485 e *Ethernet*) e através da opção sem fios (WiFi).
- *Power Supplies*: Os *bricks* e os *bricklets* podem ser alimentados pelo conector USB disponível no *brick*. As pilhas de blocos também podem ser alimentadas pelo USB contido no *Master*, contudo para casos que necessitem de correntes mais elevadas existe um módulo de alimentação específico.
- *Acessórios*: Até ao momento o único acessório que está disponível é um adaptador entre um jack de 5 mm e um conector de 2 fios.

- **Software**

A Tabela 16 apresenta as características de *software* do dispositivo *TinkForge*.

Tabela 16 – Características de *software* do *TinkForge*.

Parâmetros	Características
Sistema Operativo	Linux, Windows, Mac OS X, Android, iOS
Linguagem	Inglês
Linguagem de Programação	C/C++, C#, Delphi/Lazarus, Java, JavaScript, LabVIEW, Mathematica, MATLAB/Octave, Perl, PHP, Python, Ruby, Shell, Visual Basic .NET

3.3 Soluções tecnológicas selecionadas

Tendo em conta a variedade de possibilidades de módulos de comunicação, a escolha recaiu para os módulos sem fios, uma vez que não acrescem custos acrescidos aquando da sua instalação. Os módulos foram selecionados tendo em conta o seu baixo custo, o seu bom desempenho em ambiente industrial e o ótimo suporte técnico oferecido pela enorme comunidade que os utiliza.

Em seguida é apresentada a placa *STM32F4DISCOVERY* que servirá de base e suporte para os módulos de comunicação. É também apresentada a placa base *STM32F4DIS-BB* que tem como principal função adicionar à placa *STM32F4DISCOVERY* a possibilidade de se conectar à *Ethernet*. A escolha destas placas de desenvolvimento prende-se com o facto de estas oferecerem alto poder de processamento a um baixo custo comparativamente com as demais placas de desenvolvimento de baixo nível. Outra vantagem apresentada pela *STMicroelectronics* é o ótimo desempenho demonstrado por esta gama de placas nas importantes áreas da indústria e medicina [66], mais propriamente nos painéis de controlo de ecrãs LCD's para sistemas de alarmes, medidores de ponta, automação industrial, servomotores AC industriais, inversores, inversores solares e robótica. Por fim, outra vantagem que serve como base para a escolha desta placa é a existência de uma grande comunidade de utilizadores que oferecem atualizações com regularidade e bibliotecas para as mais diversas aplicações. Esta disponibilização de *software* e bibliotecas open-source oferecem segurança e fiabilidade ao utilizador final.

3.3.1 Módulo Bluetooth HC-05

O HC-05, é um módulo conversor de Bluetooth para série, que pode ser facilmente encontrando na internet a um preço muito reduzido, apresentando-se como uma opção económica para se

adicionar uma conexão Bluetooth a um projeto. As características do módulo são apresentadas na Tabela 17.

Tabela 17 – Características do módulo Bluetooth HC-05 [67].

Característica	Valor
Versão	Bluetooth v2.0+EDR
Frequência	2.4 GHz ISM band
Modulação	GFSK
Emissão	≤ 4 dBm, Class 2
Sensibilidade	≤ -84 dBm at 0.1 % BER
Velocidade Assíncrona	2.1 Mbps (Max) / 160 kps
Velocidade Síncrona	1 Mbps / 1 Mbps
Segurança	Autenticação e Encriptação
Conectividade	Porta série (RS-232)
Alimentação	+3.3 VDC 50 mA
Temperatura	-20 °C até 75 °C

Contudo, os valores que vêm configurados por defeito podem ser alterados e adaptados dependendo do projeto a desenvolver. Para tal é necessário entrar no modo de comando AT do módulo HC-05 e beneficiar de um vasto conjunto de parâmetros configuráveis que o dispositivo oferece. O processo de configuração é muito simples, consiste essencialmente em manter o pin “key” do módulo HC-05, representado na Figura 35, a *high* enquanto este estiver a ser alimentado. O dispositivo entra então no modo de comando AT com uma velocidade por defeito de 38400 bps e espera por qualquer comando AT que lhe seja enviado.

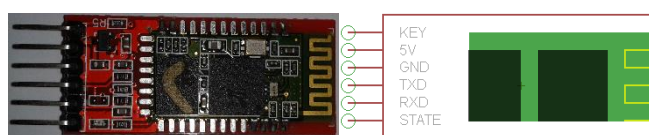


Figura 35 – Módulo HC-05 e respetivo esquemático.

3.3.2 Módulo RFM12b

O RFM12b é um módulo rádio emissor/recetor que faz parte da linha de produtos da *Hoperf EZRadio*, utiliza modulação FSK e tem integrado todas as funções de RF num único circuito integrado. Foi projetado para funcionar em aplicações que necessitem de conformidade com a FCC (Federal Communications Commission) ou ETSI (*European Telecommunications Standards Institute*) nas bandas não licenciadas a 433 MHz, 868 MHz e 915 MHz.

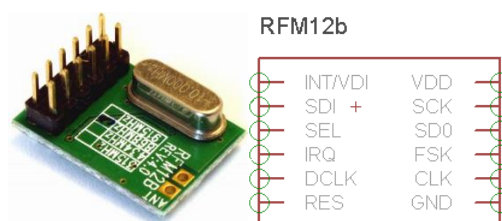


Figura 36 – Módulo RFM12b e respetivo esquemático.

O módulo RFM12B, representado na Figura 36, suporta comandos para configurar vários parâmetros como sejam, a frequência, o desvio de frequência, a potência de saída entre outros. A configuração dos parâmetros pode ser realizada utilizando a biblioteca, *RF12 library* da Jeelabs para o Arduino. Este módulo suporta até 256 grupos de 127 nós cada. O tamanho máximo de cada trama enviada entre módulos é de 128 bits. O módulo pode ser alimentado de 2.2 a 3.8 VDC, sendo que este possui uma interface SPI (nSEL, SCK, MISO e MOSI) responsável pelo envio e receção de dados assim como pela configuração do mesmo.

As suas principais aplicações são o controlo remoto, sensor remoto, aquisição *wireless* de dados, sistemas de segurança, brinquedos e monitorização de sistemas.

3.3.3 Módulo nRF24L01+

O nRF24L01+ é um módulo *transceiver* de custo muito reduzido fornecido com um protocolo de banda base embebida (ShockBurst), adequado para aplicações *wireless* de muito baixo consumo. Produzido pela Nordic, este módulo vem equipado com uma antena embutida que opera na frequência de 2.4 GHz com velocidade de operação de 2 Mbps, com um alcance que pode chegar aos 10 metros em ambiente interiores e 50 metros em ambientes exteriores. Utiliza modulação GFSK, habilidade de anti-interferência, verificação de erros por CRC, comunicação multiponto de 125 canais e controlo de fluxo. O módulo vem equipado com 8 conectores como demonstrado na Figura 37.

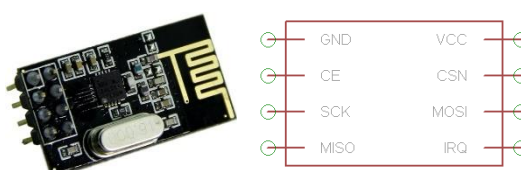


Figura 37 – Módulo nRF24L01+ e respetivo esquemático.

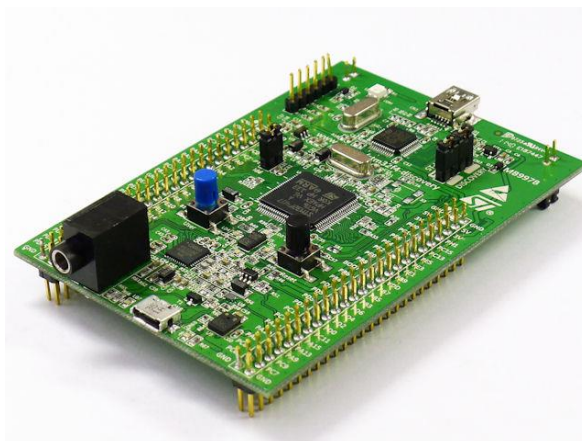
Em seguida apresenta-se a Tabela 18 com as principais características do módulo nRF24L01.

Tabela 18 – Especificações do módulo nRF24L01 [68].

Especificação	Valor
Tensão de Alimentação	1.9 a 3.6 V
Corrente de Alimentação	1 uA
Frequência	2.4 Ghz
Velocidade de Operação	2 Mbps
Modulação	GFSK
Comunicação	Multiponto 125 canais
Dimensões	2.9 x 1.5 x 0.5 cm
Taxa de Transmissão	+7dB
Sensibilidade de Recepção	≤ -90dB

3.3.4 Placa STM32F4DISCOVERY

A placa *STM32F4DISCOVERY*, representada na Figura 38, ajuda o utilizador a descobrir as funcionalidades e recursos da linha STM32F407/417 e a desenvolver de uma forma cómoda as aplicações desejadas. A placa oferece tudo o que é necessário tanto para utilizadores iniciantes como para utilizadores experientes para que a adaptação seja a mais breve possível. Baseada no microcontrolador STM32F407VGT6 ARM Cortex-M4, inclui a ferramenta de *debug* ST-Link/V2, dois sensores, um acelerómetro e um microfone digital, um áudio DAC com *class D speaker driver* integrados, LEDs, botões e um conector USB OTG micro-AB.

Figura 38 – Placa *STM32F4DISCOVERY* [69].

3.3.5 Placa Base STM32F4DIS-BB

A placa base de desenvolvimento de baixo custo *STM32F4DIS-BB* representada na Figura 39, especialmente desenvolvida para a placa *STM32F4DISCOVERY*. Esta placa base oferece ao utilizador da *STM32F4DISCOVERY* ainda mais recursos do que a *STM32F4*. Esta placa inclui

funcionalidades adicionais para a *STM32F4DISCOVERY*, incluindo portas série, *USB*, *Ethernet*, *CAN*, *SPI*, *I2C*, *GPIO*, Câmara, *MicroSD*, *LCD* e interface para *touch screen*. A placa base *STM32F4DIS-BB* e a *STM32F4DISCOVERY* formam em conjunto o kit de desenvolvimento denominado por *DevKit407*.



Figura 39 – Placa Base *STM32F4DIS-BB*. [70]

3.4 Ferramentas de Software de desenvolvimento

No decorrer deste projeto foram utilizadas várias ferramentas que auxiliaram nas diversas etapas de desenvolvimento deste projeto. As ferramentas de *softwares* apresentadas têm como função principal simplificar o processo de desenvolvimento, proporcionando uma melhoria na interatividade computador-máquina tal como na interatividade homem-máquina. Em seguida são apresentados os mais significativos.

3.4.1 CadSoft EAGLE PCB software de design

Mais conhecido simplesmente por *Eagle*, é um *software* que fornece soluções simples e poderosas para projetos de placas de circuitos impresso. É um programa intuitivo, fácil de aprender e de se utilizar, oferece funcionalidade interessante, tais como, *layout* da placa e roteador automático. Tem uma grande comunidade, que através de fóruns, disponibilizam os seus projetos e oferecem ajuda.

3.4.2 Keil μ Vision Integrated Development Environment

O Keil μ Vision MDK-ARM é um *software* completo de desenvolvimento *CortexTM-M*, *Cortex-R4*, *ARM7TM* vocacionado para dispositivos com processadores do tipo *ARM9TM*. O MDK-ARM

foi projetado para aplicações com microcontroladores, é de fácil aprendizagem, mas poderoso o suficiente para as mais exigentes aplicações.

3.4.3 MIT App Inventor 2

O *MIT App Inventor* é uma ferramenta de programação baseada em blocos que corre num *web browser*, possibilitando a qualquer utilizador iniciar-se na programação e desenvolvimento de aplicações totalmente funcionais para dispositivos *Android*. Esta ferramenta suporta um modo *debug* em tempo real inovador, assim como a possibilidade de em poucos segundos descarregar a aplicação desenvolvida para um dispositivo móvel *Android*, a fim de a experimentar. O *MIT App Inventor* suporta uma comunidade de quase 2 milhões de utilizadores, representando um total de 195 países em todo mundo. É uma aplicação de código aberto para *Android* que visa tornar tanto a programação como a criação de aplicações acessíveis a um público amplo [71].

3.4.4 Visual Studio

É um conjunto de programas da Microsoft utilizado para o desenvolvimento de *software* especialmente dedicado ao .NET Framework e a diversas linguagens, tais como, o Visual Basic, C, C++, C# e J#. Com este *software* é possível também trabalhar na área da *web*, através da plataforma ASP.NET.

3.5 Tecnologia e Gestão de Informação Digital

Não fugindo à regra de qualquer desenvolvimento de projeto foi necessário fazer a mobilização de recursos, tanto humanos como materiais. Para projetos relacionados com desenvolvimento *web* é necessário ter em especial atenção a preparação do ambiente de desenvolvimento. Projetos com esta natureza podem ser divididos em dois ambientes distintos:

- Ambiente de desenvolvimento;
- Ambiente de produção.

Enquanto que o ambiente de desenvolvimento é o local onde se desenvolve o site, que por exemplo pode ser a nossa máquina ou servidor, o ambiente de produção é o servidor onde o *website* ficará alojado.

Ambos os ambientes devem ser similares, visto que esse facto facilita a passagem de um ambiente para o outro, evitando assim algum constrangimento no instante final do desenvolvimento.

O sistema operativo que serviu de base para o ambiente de desenvolvimento foi o *Windows 7*. Previamente foram instalados o XAMPP (simulador dos serviços *Apache* e *MySQL*), o *MySQL Workbench 6.2* (ferramenta visual para tratamento de bases de dados) e o *Notepad++* (*software* open-source para edição de código). Estes foram os requisitos necessários para se iniciar o desenvolvimento do projeto.

Ainda na fase de preparação do ambiente de desenvolvimento e tendo em conta que é importante ter todos os elementos bem organizados, foi organizado um esquema de pastas para simplificar e organizar o projeto. Assim foram criadas as seguintes pastas, “*style*”, “*js*”, “*include*” e “*images*”, sendo que em cada uma das pastas anteriormente definidas deve ficar um conjunto específico de ficheiros.

Na pasta “*style*” foram guardados todos os ficheiros relacionados com a formatação das páginas. A pasta “*js*” contém todos os ficheiros *JavaScript*, na pasta “*include*” ficaram guardados os ficheiros *php* que contêm as funções necessárias ao funcionamento das páginas *web*. Por fim a pasta “*images*” servirá para armazenar toda e qualquer imagem utilizada no *website*.

3.5.1 Tecnologias e ferramentas utilizadas

Para o desenvolvimento de uma aplicação *web* deste tipo, existem diversos *softwares*, tecnologias e ferramentas que podem ser utilizadas, no caso particular deste projeto foram utilizadas soluções livres de licença.

Software:

XAMPP

O XAMPP é um pacote constituído pelos principais servidores de código aberto do mercado, incluindo FTP, bases de dados *MySQL* e *Apache* com suporte para linguagens *PHP* e *Perl*. Desenvolvido pela *Apache Friends*, está disponível sob a licença GNU e opera como um servidor livre, fácil de usar e capaz de interpretar páginas dinâmicas. Está também equipado com outros módulos como por exemplo o *OpensSSL* e *phpMyAdmin*. Não necessita de instalação, uma vez que basta descompactar o arquivo e executar o aplicativo que inicia os servidores.

MySQL Workbench 6.2 CE

O *MySQL WorkBench* é uma ferramenta gráfica para gestão de bases de dados que integra tanto a criação como o *design*. É uma ferramenta de fácil utilização que permite a modelagem de

dados, administração de sistemas e uma variedade de outras tarefas necessárias para manter uma aplicação orientada a dados em funcionamento. O *software* é gratuito, bastando simplesmente fazer o registo antes de o descarregar.

Notepad++

O *Notepad* é um editor de texto de código aberto, para Windows, que permite trabalhar com arquivos de texto simples e códigos-fonte de diversas linguagens de programação. Para o utilizador, o *Notepad++* tem suporte à diferenciação de comandos através de um vasto sistema de cores programáveis, um recurso muito utilizado em ambiente de programação. Neste programa já vem embutido o reconhecimento de diversas linguagens como sejam o C, o C++, o Java, o HTML, o XML, o PHP, o JavaScript, entre outras.

Servidor:

APACHE

O *APACHE* é um servidor *web* desenvolvido por *Rob McCool* e tem como base o servidor *web* *NCSA*. É o servidor *web* mais utilizado na internet e funciona em diversas plataformas como Windows, Unix, Linux e também Mac OS X.

O servidor *Apache* é extremamente configurável, robusto e apresenta um elevado desempenho. Foi desenvolvido com o intuito de criar um servidor *web* com diversos recursos e com código fonte disponível gratuitamente na internet. Este programa é responsável, não só por disponibilizar páginas, imagens, vídeos ou qualquer outro tipo de objeto no *browser* do cliente, assim como receber dados do cliente, processar e enviar dados permitindo ao cliente tomar a ação desejada.

Neste projeto o servidor apache foi simulado com o programa XAMMP.

MySQL

O *MySQL* é um Sistema de Gestão de Bases de Dados (SGDB) relacional, que utiliza a linguagem SQL. Este SGDB foi inicialmente projetado para projetos de pequeno e medio porte, com capacidade de suportar por volta de cem milhões de registos em cada tabela, podendo alcançar um tamanho médio de aproximadamente cem megabytes por tabela, no entanto, a necessidade e a crescente procura de quantidade de informação necessária fizeram com que o

MySQL fosse evoluindo e hoje em dia já ultrapassa em muito esses limites e capacidades anteriormente definidos para as versões anteriores.

O *MySQL* é uma ferramenta reconhecida mundialmente pela sua fácil utilização e por isso amplamente usado em empresas que trabalham com grandes volumes de dados, tais como *Facebook*, *Google*, *Cisco* entre outras grandes empresas de renome [72]. É dotado de um interface extremamente simples e é compatível com grande maioria dos sistemas operativos.

A principal vantagem do *MySQL* é ser *open source*, desenvolvido e distribuído sob a licença *GPL*, que determina o tipo de utilização a dar às ferramentas e demais recursos. Possui ainda como principais características a portabilidade, a diversidade de armazenamento, a velocidade e o alto poder de execução e armazenamento.

PHP

O *PHP*, acrónimo recursivo para *Hypertext Preprocessor* é uma linguagem de *scrip open source*, bastante utilizada e especialmente desenhada para o desenvolvimento de aplicações *web* embutível no *HTML* através da tag inicial e final `<?php` e `?>`, respetivamente. A principal característica do *PHP* é que o código é executado no lado do servidor, gerando *HTML* que é posteriormente enviado para o cliente. Outra característica importante do *PHP* é que o cliente recebe os resultados da execução desse script, mas nunca irá ter acesso ao código fonte.

O *PHP* é a linguagem que permite a ligação entre a base de dados *MySQL* e as páginas *web*. A combinação entre o *Apache MySQL* e o *PHP* é basicamente o conjunto de *software* que é necessário para a construção e desenvolvimento de *websites* e aplicações sofisticadas de uma forma simplificada. As aplicações *open source* apresentam vantagens ao nível do desenvolvimento, segurança e confiabilidade das soluções finais.

Linguagem

HTML

HTML, sigla representativa para “*HyperText Markup Language*”, é uma linguagem de marcação utilizada para desenvolvimento das páginas *web*, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela internet. A linguagem *HTML* permite definir as páginas *web*, basicamente trata-se de um conjunto de etiquetas (*tags*) que servem para definir o aspeto de apresentação do texto e de

outros elementos das páginas *web*. O *HTML*, neste caso, foi utilizado apenas para a estruturação das páginas.

CSS

CSS é um simples mecanismo que permite a formatação e personalização do estilo dos documentos *web*. Entre muitas formatações é possível controlar efeitos de cor (*color*), tipo de letra (*style*), cor de fundo (*background*), imagens (*image*), alinhamento (*align*), espaçamentos (*padding*), entre outros.

É possível utilizar esta tecnologia de duas formas, uma é embutida em *HTML* ou *XML* e outra, de uma forma mais avançada é possível guardar as configurações num ficheiro à parte e fazer a ligação com o ficheiro *HTML* principal.

Neste projeto foi utilizada a opção mais avançada, onde uma pasta foi criada especificamente para guardar todos os ficheiros *css*.

SQL

SQL, sigla inglesa para “*Structured Query Language*” é uma linguagem padrão de gestão de dados que interage com as principais bases de dados baseadas no modelo relacional. É caracterizada por ser uma poderosa linguagem de comunicação com bases de dados, própria para a realização de operações relacionais. Tem como principais funções a recuperação, atualização e eliminação de dados da base de dados relacional, assim como, criação e modificação da estrutura da mesma.

JavaScript

O *JavaScript* é uma linguagem de programação poderosa do lado cliente, isto é, o processamento é feito pelo próprio navegador. Com o *JavaScript* é possível criar programas de estrutura pequena que ficaram encarregues por produzir efeitos especiais nas páginas *web* dinâmicas e interativas, para além de ser possível proporcionar aos utilizadores uma maior interatividade. É uma linguagem orientada a objetos, ou seja, trata todos os elementos da página *web* como objetos distintos, facilitando a sua programação.

3.6 Conclusão

Neste capítulo foram abordadas as principais plataformas eletrónicas de desenvolvimento disponíveis no mercado. Uma vez mais é de se salientar que cada tecnologia tem as suas vantagens/desvantagens dependendo da sua aplicação. A escolha da plataforma eletrónica recaiu sobre a placa STM32F4DISCOVERY da STMicroelectronics, que vem equipada com um processador de alto poder de processamento a um reduzido custo económico. Esta apresenta a melhor relação de entre as seguintes características: preço, desempenho, qualidade, robustez e confiabilidade. Ainda neste capítulo foi realizada uma breve introdução aos módulos de comunicação e *softwares* de desenvolvimento utilizados na elaboração do projeto.

Capítulo 4

IMPLEMENTAÇÃO

4.1 InSwitch

Como foi apresentado anteriormente, o *InSwitch* é um dispositivo eletrónico capaz de realizar o controlo do motor e a monitorização de alguns dos parâmetros do motor, tais como, a corrente, a tensão e o valor da temperatura em cada um dos enrolamentos, como se pode observar da análise da Tabela 19.

Tabela 19 – Parâmetros monitorizados pelo *InSwitch*.

Corrente	Por Enrolamento (três)	0 a 30 A
Tensão	Por Enrolamento (três)	0 a 440 V
Temperatura	Por Enrolamento (três)	0 a 100 °C

Estes valores, depois de adquiridos através de sensores, são tratados pelo microcontrolador do *InSwitch* e posteriormente utilizados para calcular diversa informação que mais tarde será disponibilizada ao utilizador via *interface web*. A Tabela 20 apresenta os parâmetros calculados pelo *InSwitch* a partir dos valores anteriormente adquiridos. Apresenta também o *setpoint* e o estado das ligações (estrela/triângulo) do motor baseado na informação disponível.

Tabela 20 – Parâmetros calculados pelo *InSwitch*.

Corrente RMS	Por Fase	0 a 30 A	$X_{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n x(k)^2}$
Tensão RMS	Por Fase	0 a 440 V	
Potência	Por Fase	0 a 40 kW	$P_{med} = \frac{1}{n} \sum_{i=1}^n u(k) i(k)$
Cos φ	Por Fase	0 a 1	$\cos \varphi = \frac{P_{med}}{U_{RMS} I_{RMS}}$
Temperatura	Determinada	Máx. 100 °C	Determinada através de um sensor
Setpoint	Estimado	2 kW a 10 kW	$SetPoint = \frac{P_N}{\sqrt{3}} \times \cos \varphi_N$
Estado	Determinado	Estrela/Triângulo	Determinado tendo em conta o SetPoint

Este dispositivo em desenvolvimento por alunos do ISEC, que permite o controlo de motores, foi o selecionado para ser a base de testes para o sistema de controlo e monitorização em desenvolvimento neste projeto. Pois como foi referido anteriormente este dispositivo já disponibiliza digitalmente informação necessária para a monitorização de um motor.

4.2 Solução de comunicação desenvolvida

Numa primeira fase do desenvolvimento do projeto foram realizados testes aos módulos de comunicação selecionados. Utilizando a plataforma de programação Arduino, mais propriamente a versão Arduino Uno, foram elaborados testes aos módulos de comunicação com o intuito de se obter uma melhor perceção do seu funcionamento.

Após se ter testado os módulos de comunicação com Arduino, foi feita a migração do sistema para a plataforma de desenvolvimento da STMicroelectronics, a qual está dotada do poderosíssimo processador ARM® Cortex®-M4. Ao longo do projeto foram desenvolvidos três protótipos: a *STM32F4DISCOVERY* equipada com o módulo Bluetooth; a *STM32F4DISCOVERY* equipada com o módulo nRF24L01+ e a *STM32F4DISCOVERY* equipada com a placa de expansão STM32F4DIS-BB.

4.2.1 Protótipo Arduino com o módulo Bluetooth HC-05:

Neste teste foram conectados ao Arduino Uno, um módulo HC-05 e um conjunto de quatro leds, como demonstrado na Figura 40. No entanto, foi necessário previamente configurar o módulo HC-05 utilizando código de configuração. Depois de carregado o código no Arduino, conectamos o módulo HC-05 e o Arduino segundo as seguintes especificações:

- HC-05 GND → Arduino GND Pin
- HC-05 VCC → Arduino 5 V
- HC-05 TX → Arduino Pin 10 (RX)
- HC-05 RX → Arduino Pin 11 (TX)
- HC-05 Key → Arduino Pin 9

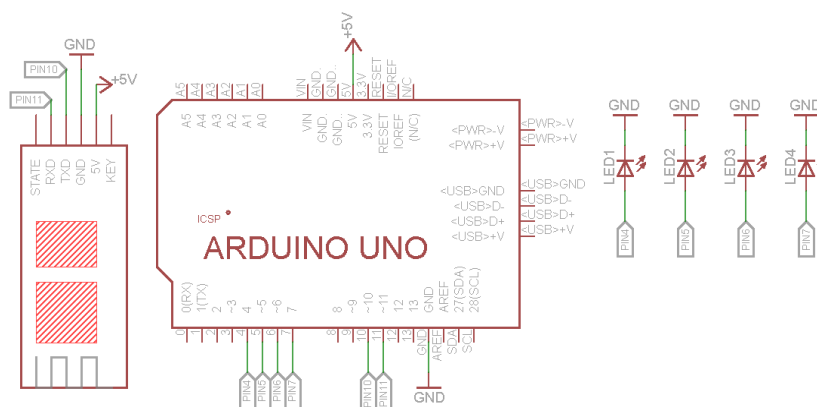


Figura 40 – Esquema de ligação do teste piloto Arduino Uno e o módulo HC-05.

Seguindo os passos de configuração descritos no *datasheet* do módulo HC-05, entramos no modo AT. Uma vez na linha de comando AT do módulo HC-05 é possível utilizar os seguintes comandos de verificação e configuração, presente na Tabela 21:

Tabela 21 – Comandos de configuração do módulo HC-05 [73].

Comando	Função
AT	Teste ao modo AT, a resposta deverá ser OK
AT+ROLE=1	Modo <i>master</i>
AT+ROLE=0	Modo <i>slave</i>
AT+VERSION?	Devolve a versão <i>Firmware</i>
AT+UART=115200,1,0	Configura a baud rate para 115200, 1 stop bit e 0 parity
AT+PIO=10,1	Configura o PIO10 a 1
AT+NAME?	Devolve o nome do módulo
AT+NAME=xxxx	Configuração do nome (substituir x por um nome)
AT+PSWD?	Devolve a <i>password</i> do módulo
AT+PSWD=xxxx	Configuração da <i>password</i> (substituir x por 4 números)

Foram também desenvolvidas duas aplicações (Figura 41), uma para o Arduino Uno e outra para um dispositivo móvel equipado com Android.

Empregando o *software* disponibilizado pelo Arduino foi desenvolvido uma aplicação “.ino” que permite o controlo de um conjunto de quatro leds independentemente ou simultaneamente através do módulo Bluetooth HC-05. Para possibilitar o controlo dos leds pelo utilizador, foi desenvolvido na aplicação *web* “MIT App Inventor 2” uma interface para o dispositivo móvel Android.

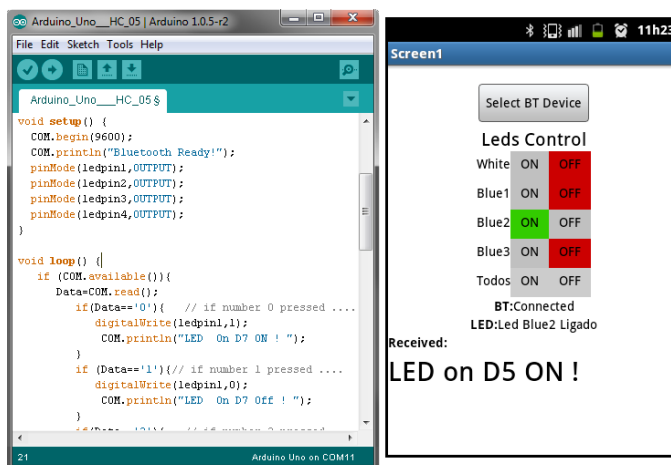


Figura 41 – Aplicação “.ino” e interface para Android.

Após se ter concluído o desenvolvimento da aplicação foi possível controlar (ligar/desligar) independentemente ou conjuntamente cada um dos quatro leds e visualizar qual dos led’s estava ligado através de uma mensagem de controlo que o arduino enviava para o dispositivo móvel.

4.2.2 Protótipo Arduino com o módulo RFM12b:

Com este protótipo, constituído por um Arduino Uno e um módulo RFM12b ligados conforme a Figura 42, pretendia-se estabelecer uma comunicação *wireless* ponto a ponto, onde um dos dispositivos transmitia e o outro recebia os dados. A programação dos dispositivos foi realizada usando o *software* de programação disponibilizado pelo Arduino, tendo sido também utilizada a biblioteca Jeelib, desenvolvida para os módulos *wireless* RFM12b e disponibilizada pela Jeelabs.

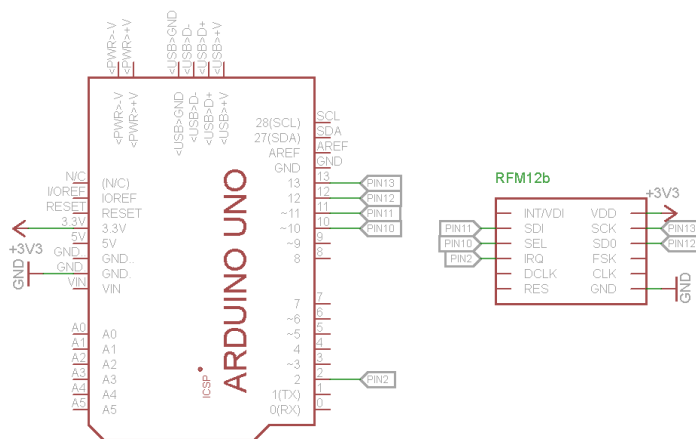


Figura 42 – Esquema de ligação do teste piloto Arduino Uno e o módulo RFM12b.

Numa primeira fase o Arduino Uno foi conectado ao PC e depois de compilado, carregou-se o código “RF12demo” para o Arduino. A taxa de transmissão série é por defeito 57600 kbps, mas pode ser alterada no código fonte, sendo para isso necessário instalar a biblioteca do módulo para o arduino “RF12” disponibilizada na *web* [74].

Em seguida o módulo RFM12b e o arduino Uno foram ligados como representado na Figura 42 e respeitando o esquema de ligação em baixo apresentado.

▪ RFM12b GND	→	Arduino GND
▪ RFM12b 3 V	→	Arduino 3 V
▪ RFM12b IRQ	→	Arduino Pin 2
▪ RFM12b NSS	→	Arduino Pin 10
▪ RFM12b SDI	→	Arduino Pin 11
▪ RFM12b SDO	→	Arduino Pin 12
▪ RFM12b SCK	→	Arduino Pin 13

Para configurar as definições da EEPROM do módulo RFM12b, foi carregado o demo “RF12” no Arduino. O referido demo utiliza as definições de configuração da EEPROM para configurar o módulo RFM12b. A configuração por defeito é “A i1 g212 @ 433 MHz”. Para alterar o ID do dispositivo bastou digitar “<N>i”, onde <N> é um numero de 1 a 26. É igualmente possível alterar a banda de frequência de funcionamento com o comando “<N>b”, onde <N> pode tomar um valor entre 4, 8 ou 9. Estes valores correspondem às bandas de frequências de 433 MHz, 868 MHz e 915 MHz, respetivamente. Por fim a configuração do grupo de network é feita com o comando “<N>g”, sendo <N> qualquer valor compreendido entre 1 e 250.

Com o intuito de facilitar as ligações entre o Arduino e o módulo RFM12b foi desenvolvido um protótipo arduino *shield* (Figura 43) com a ajuda do *software* de prototipagem de PCB’s *EAGLE 6.4.0*.

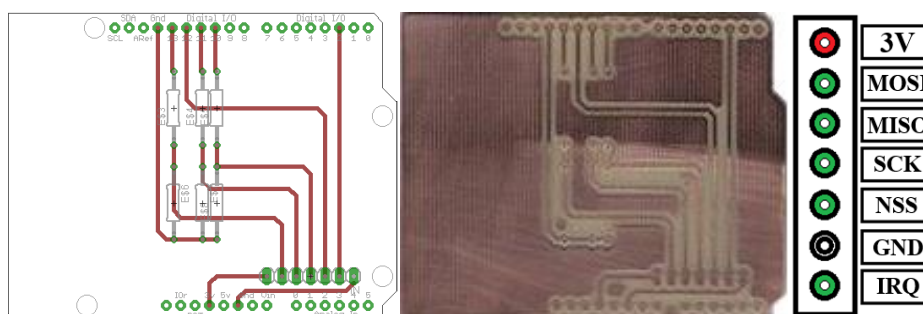


Figura 43 – *Shield* desenvolvido para o Arduino Uno.

4.2.3 Protótipo STM32F4 com o módulo HC-05:

O desenvolvimento deste protótipo pode ser dividido em duas partes distintas, sendo a primeira a montagem e programação da placa *STM32F4DISCOVERY* e do módulo Bluetooth HC-05 e a segunda, o desenvolvimento de uma aplicação *Android*. A aplicação *Android* não é mais do que um interface que proporciona ao utilizador uma ferramenta apelativa e de fácil utilização para efetuar a análise do motor.

Montagem e Programação:

Uma vez que o módulo Bluetooth HC-05 adquirido vem equipado com comunicação série, foi necessário conectá-lo uma das portas UART disponíveis na placa *STM32F4DISCOVERY*, conforme demonstrado na Figura 44:

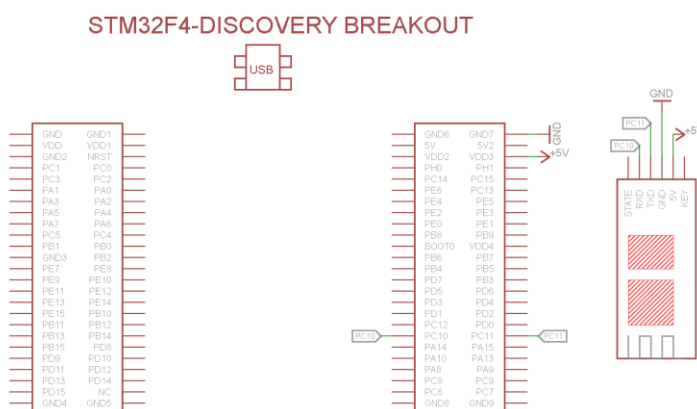


Figura 44 – Esquema de ligação da placa *STM32F4DISCOVERY* e o módulo HC-05.

A placa *STM32F4DISCOVERY* foi programada com o *software Keil μ Vision*, para que quando houvesse informação disponível, esta fosse enviada via porta série para o dispositivo *Bluetooth* que por sua vez encaminhava a informação recebida para o dispositivo móvel equipado com *Android* e *Bluetooth*. Na Figura 45 podemos ver algumas linhas de código responsáveis pela construção das mensagens a enviar para o módulo Bluetooth HC-05.

```
void Bluetooth_Task( void *pvParameters ){
    while(1){
        vTaskDelay(150 / portTICK_RATE_MS);
        xSemaphoreTake(USART1_Mutex, portMAX_DELAY);

        printf("M,%d,I,%d,%d,%d,", Motor_ID, Integrado.Iarms, Integrado.Ibrms, Integrado.Icrms);
        vTaskDelay(150 / portTICK_RATE_MS);
        printf("M,%d,V,%d,%d,%d,", Motor_ID, Integrado.Varms, Integrado.Vbrms, Integrado.Vcrms);
        vTaskDelay(150 / portTICK_RATE_MS);
        printf("M,%d,P,%d,%d,%d,", Motor_ID, Integrado.Pa, Integrado.Pb, Integrado.Pc);
        vTaskDelay(150 / portTICK_RATE_MS);
        printf("M,%d,T,%d,%d,%d,", Motor_ID, Instantaneo.Ta, Instantaneo.Tb, Instantaneo.Tc);
        vTaskDelay(150 / portTICK_RATE_MS);
        printf("M,%d,R,%d,%c,%d,", Motor_ID, Setpoint, Estado, Instantaneo.Vmicro);

        xSemaphoreGive(USART1_Mutex);
    }
}
```

Figura 45 – Código responsável por enviar informação para o módulo HC-05.

Como podemos observar pela Figura 45 a mensagem pode, por exemplo, assumir a seguinte estrutura:

Motor	ID	Grandeza	Valor A	Valor B	Valor C
M	0001	I	3,1	3,2	2,9

Desenvolvimento da aplicação Android:

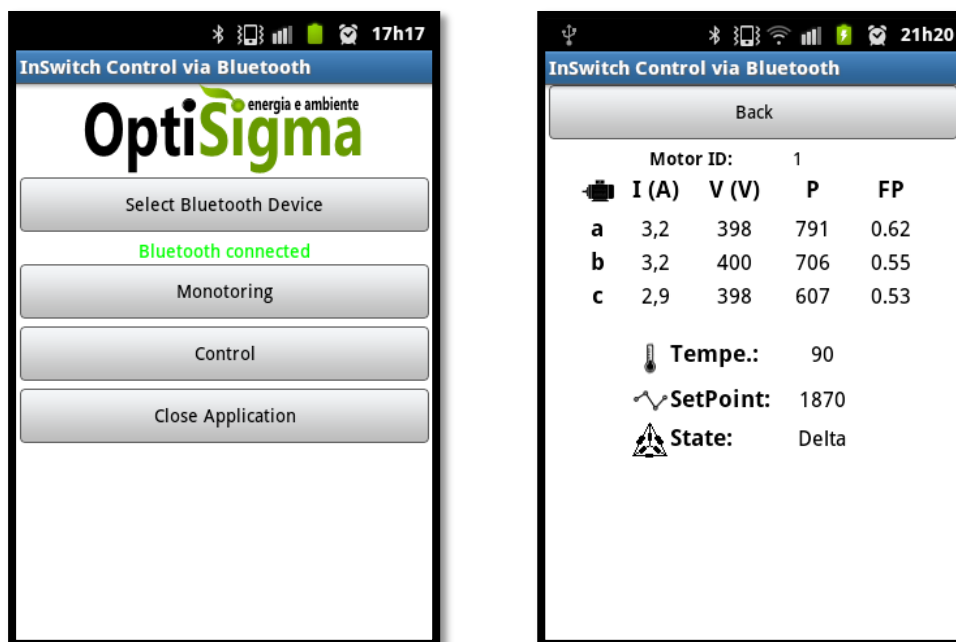


Figura 46 – Aplicação Android para controlo e monitorização do motor por Bluetooth.

Uma vez mais a aplicação “APP Inventor 2” foi utilizada para o desenvolvimento de um interface em Android. Esta aplicação permite ao utilizador não só monitorizar em tempo real valores importantes para a análise do motor de indução trifásica, mas também controlar a comutação estrela/triângulo e ainda configurar um valor de *setpoint*.

Depois de conectado ao módulo Bluetooth HC-05 equipado no motor, o utilizador tem ao seu dispor uma tabela com diversos valores do motor, como por exemplo: corrente de fases, tensão de fases, potência e fator de potência, temperatura do motor, estado do motor (estrela ou triângulo) assim como o seu *setpoint*. Para poder assumir o controlo de alguns parâmetros do motor, visto ser uma tarefa com algum risco, será necessária a introdução de uma *password* por parte do utilizador, como podemos comprovar pelo ecrã nº3 da Figura 46.

4.2.4 Protótipo STM32F4 com o módulo nRF24L01+

Inicialmente foram realizados alguns testes com a placa *STM32F4DISCOVERY* e o módulo RFM12b. Mas devido a dificuldades encontradas com incompatibilidades e falta de bibliotecas respeitantes ao módulo RFM12b para a placa STM32F4 DISCOVERY, conclui-se que a melhor opção seria apostar noutro módulo rádio.

A seleção de outro módulo recaiu no módulo apresentado anteriormente no Capítulo 3, o nRF24L01+.

Mais uma vez, para testar este módulo foram utilizados dois conjuntos formados por um módulo nRF24L01+ e uma placa *STM32F4DISCOVERY*. Um dos conjuntos foi programado por forma a se comportar como “*master*” e o outro conjunto programado para se comportar como “*slave*”.

Montagem e Programação:

Os conjuntos dos módulos rádio foram conectados tendo em conta as indicações da Figura 47, não fazendo diferenciação entre ambos.

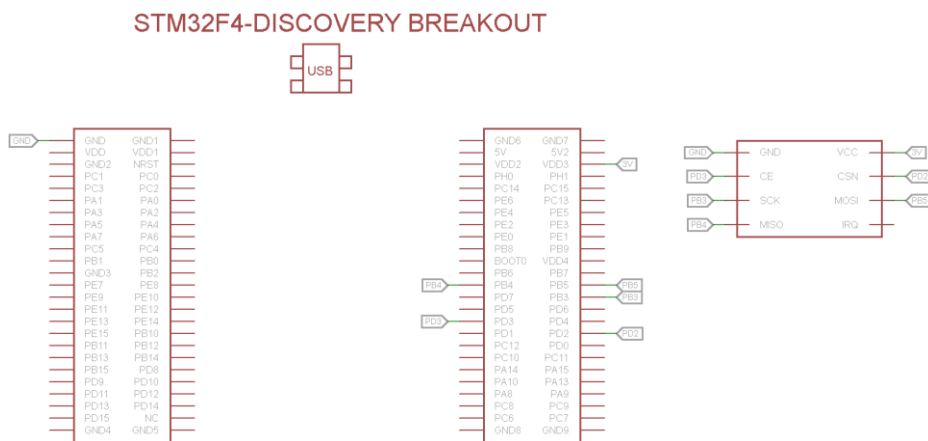


Figura 47 – Esquema de ligação da placa *STM32F4DISCOVERY* e o módulo *nRF24L01+*.

Uma vez mais a programação foi feita com a ajuda do *software Keil μ Vision*. Onde o *master* foi programado para que, quando acusasse a receção de um *ping* proveniente do *slave* e contendo o *id* do motor, o enviasse de volta em forma de *broadcast*. O *slave* ao receber o *ping* verifica se corresponde ao seu *id* e, em caso afirmativo, envia mensagens com toda a informação relativa ao motor. Quando termina de enviar as mensagens envia um código de confirmação de conclusão da comunicação. O *master* sabendo que o *slave* já terminou de comunicar, coloca-se novamente em modo de espera. Na Figura 48 é apresentado um exemplo de como a construção da mensagem de dados a enviar é realizada.

```
memset(&dataOut[0], 0, sizeof(dataOut));
sprintf((char*)dataOut, "M%c%c%c%cIA%BdC%d/n", InSwitchCode[0], InSwitchCode[1], InSwitchCode[2], InSwitchCode[3], Ia, Ib, Ic);
TM_NRF24L01_Transmit(dataOut);
do {
    transmissionStatus = TM_NRF24L01_GetTransmissionStatus();
}
while (transmissionStatus == TM_NRF24L01_Transmit_Status_Sending);
//TIM2 Enable to transmit data
TIM_Cmd(TIM2, ENABLE);
//Wait for data to be sent
while(!WaitingTime == 1);
WaitingTime = 0;
```

Figura 48 – Código responsável por formatar a mensagem a enviar por rádio frequência.

4.2.5 Protótipo *STM32F4* com placa base *STM32F4DIS-BB*:

Com o intuito de disponibilizar ao utilizador de uma forma simples toda a informação recolhida dos motores foi adicionado à placa principal (a *STM32F4DISCOVERY*), a placa base *STM32F4DIS-BB*, a qual possibilita a ligação da rede de monitorização de motores à internet. A placa base *Ethernet* vai tornar os dispositivos de controlo dos motores ainda mais acessíveis, disponibilizando ao utilizador na internet, os dados em tempo real. É pretendido também a criação de uma base de dados com toda a informação recebida dos motores. Com esta informação pretendemos criar um histórico de cada motor por forma a proporcionar ao

utilizador uma análise detalhada, do mesmo que servira para realizar a prevenção de avarias e a predição da data mais adequada para a realização de operações de manutenção preventiva.



Figura 49 – DevKit407, conjunto das placas *STM32F4DISCOVERY* e *STM32F4DIS-BB*.

A comunidade *STMicroelectronics* vem disponibilizando diversos exemplos e aplicações de demonstração para as referidas placas. Na elaboração deste protótipo foi utilizado o exemplo *STM32F4xx_Ethernet_Example*, mais especificamente o demo *Standalone/httpserver*, o qual foi modificado por forma a se adaptar às necessidades do projeto. O demo *HTTPserver* apresenta uma implementação de um servidor *web* com as seguintes características:

- URL;
- Suporte de *Common Gateway Interface (CGI)*;
- Suporte de *Server Side Includes (SSI)*;
- Suporte de pedido de *HTTP Post*.

Ligação à internet:

Por forma a colocar este serviço acessível ao utilizador através de um computador remoto, é necessário conectar o conjunto *DevKit407* (Figura 49) ao computador e abrir o projeto disponibilizado em anexo utilizando o *Keil MDK-ARM*. No ficheiro *main.h*, dependendo das necessidades, é possível comentar/não comentar diversas opções tais como, “*SERILA_DEBUG*”, “*USE_LCD*” ou “*USE_DHCP*”. Ainda no ficheiro *main.h*, é possível configurar o endereço *IP* e *gateway* consoante as necessidades. A última etapa contempla a compilação do projeto e o seu *upload* para a memória *flash* do *STM32F407*. De seguida deve-se conectar a placa *DevKit407* ao *router* com um cabo *Ethernet crossover* através da porta *RJ45* e no computador remoto, abrir um cliente *web* (*Mozilla Firefox* ou *Chrome*) e digitar o endereço *IP* definido para a placa *DevKit407* (por defeito o endereço é 192.168.0.10).

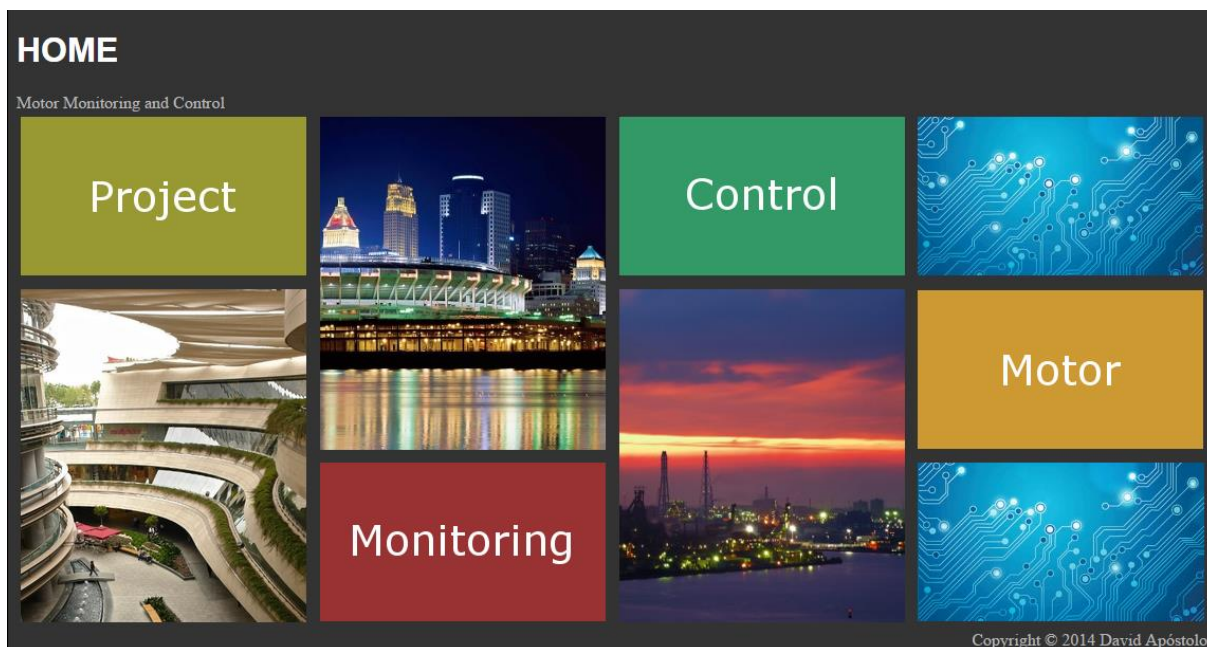
Interface web browser:

Figura 50 – Página Home

Na elaboração do interface (Figura 50) foram desenvolvidas 4 páginas intuitivas e de fácil acesso para o utilizador: “*Home*”, “*Project*”, “*Monitoring*” e “*Control*”.

A página “*Home*” é a página inicial e dá acesso a todas as outras. Na página “*Project*” é possível ler alguma informação sobre o projeto. É a página “*Control*” que vai permitir ao utilizador ter o controlo sobre alguns parâmetros do motor quando este se conectar à rede. Na página “*Control*”, depois de se introduzir uma *password* de segurança, o utilizador tem acesso a um painel de controlo para alguns parâmetros do motor e a introdução de um *SetPoint*. Para possibilitar o controlo do motor através do interface *web* é utilizada uma técnica designada por *CGI*. O *CGI* é uma técnica padrão da *web* usada para executar um pedido vindo de um cliente no lado do servidor e, em seguida, retornar uma resposta para o cliente. Neste caso o *CGI* só funciona com pedidos do método *GET* e pode suportar até 16 parâmetros codificados no *URL*. A função que manipula o *CGI* executado no lado do servidor retorna um arquivo *HTML* que o servidor *HTTP* envia para o cliente. Na página “*Control*”, este método é utilizado para comutar os estados (triângulo/estrela) e para enviar um *SetPoint*.

Por último a página “*Monitoring*” possibilita ao utilizador a visualização de um conjunto de valores indispensáveis para a análise e monitorização do motor. Nesta página é utilizado um método para incluir dinamicamente dados dinâmicos em código *HTML*, denominado *SSI*. Isto é feito através de uma *tag* específica colocada no código *HTML* da página *web*. A *tag* deve ter

o seguinte formato: <!--#tag-->. No caso da página “*Monitoring*” são utilizadas diversas *tag*’s, tais como, <!--#ia--> (corrente fase a), <!--#vb--> (tensão na fase b), <!--#pc--> (potência por fase c), <!--#t--> (temperatura), etc. Quando existe um pedido para que a página “*Monitoring*” seja carregada, o servidor irá analisar a página e quando uma *tag* for encontrada, ela será substituída pelo valor requisitado.

Este protótipo não foi incluído na rede de monitorização para motores de indução uma vez que apresentava falhas na atualização da leitura e disponibilização dos dados na interface *web* desenvolvida. Por forma a colmatar a falha deste protótipo foi desenvolvido de raiz uma base de dados e uma aplicação *web* com o intuito de disponibilizar os dados ao utilizador. O seu desenvolvimento é descrito nos subcapítulos 4.5 e 4.6.

4.3 Protocolo de comunicação desenvolvido

A rede de monitorização desenvolvida (Figura 51) possui diferentes tipos de comunicação: *Bluetooth*, rádio frequência e porta série. Para facilitar a sua apresentação esta secção encontra-se subdividida em três subcapítulos.

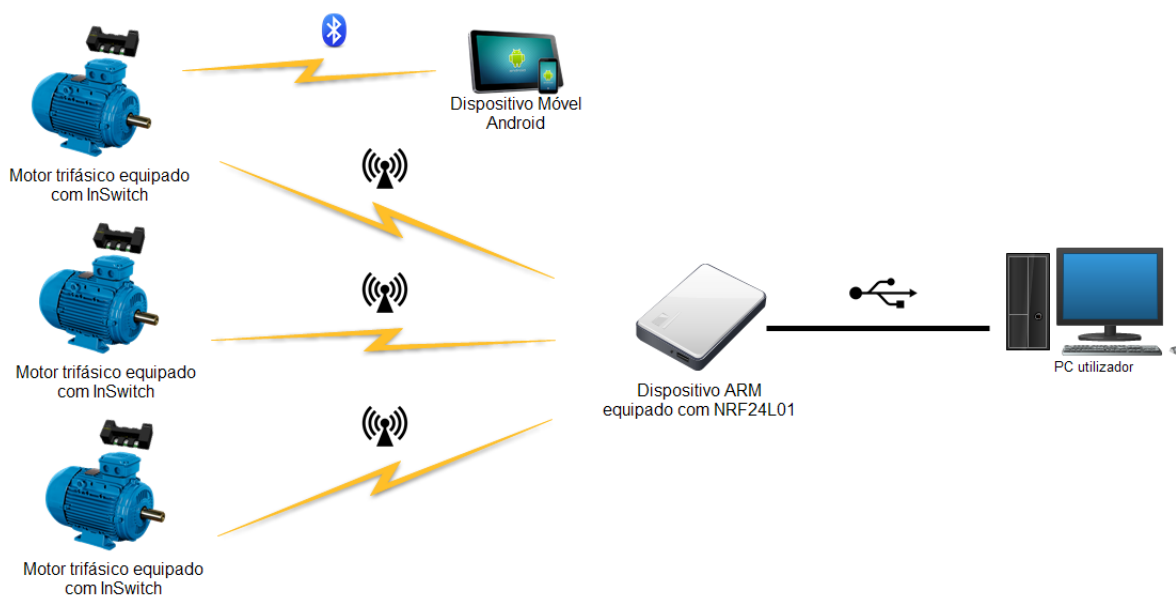


Figura 51 – Exemplo de uma rede local de monitorização de motores.

4.3.1 Rádio Frequência

A comunicação rádio frequência é realizada entre o dispositivo *InSwitch* (*slave*) e o dispositivo central (*master*) como representado na Figura 52. Para que ambos pudessem comunicar entre si, foi necessário, não só programá-los para o efeito, mas também adicionar o módulo nRF24L01+, como descrito no subcapítulo 4.2.4, a ambos os dispositivos.



Figura 52 – Parte da rede de monitorização – comunicação rádio frequência.

O *master* foi programado para enviar uma mensagem de confirmação aquando da receção de um *ping* enviado pelo *slave*. Depois de enviada a confirmação, o *master* recebe os dados do *slave* e envia por porta série os dados recebidos. Se os dados forem enviados, o *master* coloca-se em espera para receber novo *ping*, se não forem enviados e o temporizador for falso ele volta a enviar uma mensagem de confirmação, se não foram enviados e o temporizador for verdadeiro ele fica a espera de novo ping. O *slave* por sua vez foi programado para inicialmente esperar um tempo aleatório até poder enviar um *ping* para o *master*. Depois de receber a confirmação de que pode comunicar com o *master*, este envia todos os dados que necessita enviar, e no fim envia uma mensagem de confirmação de que terminou a sua comunicação. Para finalizar, depois de enviar a confirmação volta a esperar um tempo aleatório até poder comunicar novamente. Para uma melhor percepção da comunicação mestre – escravo que é realizada, apresenta-se na Figura 53 um diagrama do seu funcionamento.

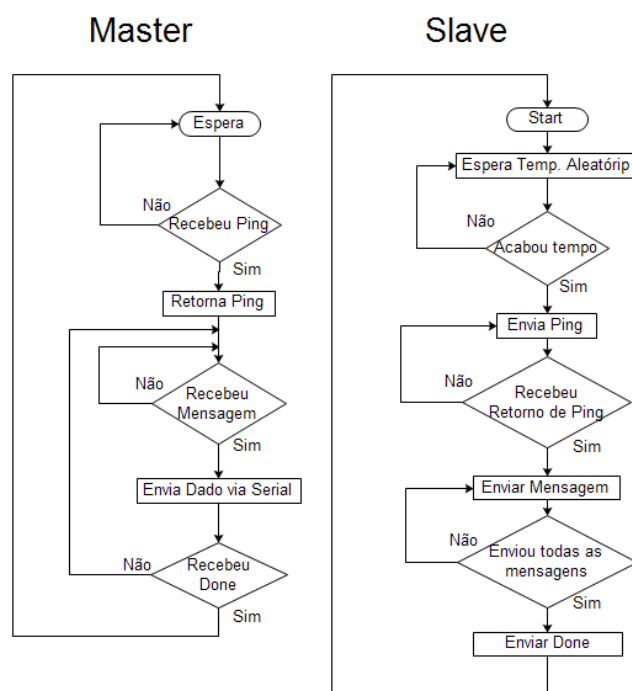


Figura 53 – Fluxograma do processo de comunicação entre o *InSwitch* e o dispositivo central.

Para o efeito foram desenvolvidas três mensagens diferentes, as quais são trocadas entre ambos os dispositivos por forma a controlar a comunicação entre dispositivos. A sua constituição é apresentada na Tabela 22:

Tabela 22 – Constituição das mensagens “Ping”, done e dados.

Mensagem <i>Ping</i>	Motor ID		Indicador			
	M0000	M	PING			

Mensagem <i>Done</i>	Motor ID		Indicador			
	M0000	M	DONE			

Mensagem de dados	Motor	ID	Grandeza	Fase A	Fase B	Fase C
	M	0001	I	AValor	BValor	CValor

No caso da mensagem de dados, uma vez que o tamanho máximo permitido é 32 bytes, e que os dados a enviar excedem esse limite, foi necessário reparti-la em várias mensagens de menor tamanho. Para distinguir cada uma das mensagens de tamanho reduzido foi utilizado um campo relativo às “Grandezas”, podendo este campo assumir os seguintes caracteres: I (Corrente), V (Tensão), P (Potência), F (Fator de Potência) e T (Temperatura).

4.3.2 Bluetooth

A comunicação *Bluetooth* é realizada entre o dispositivo *InSwitch* e o dispositivo móvel equipado com *Android* como representado na Figura 54. Por forma a permitir que o *InSwitch* comunicasse via *Bluetooth*, foi adicionado ao mesmo um módulo HC-05, sendo este posteriormente programado para o efeito.



Figura 54 – Parte da rede de monitorização – Comunicação Bluetooth.

São enviadas cinco mensagens com a informação do motor que o *InSwitch* disponibiliza. As mensagens têm, a título de exemplo, a constituição representada na Tabela 23, onde no campo “Motor” a letra M indica que se trata de uma mensagem vinda do motor, no campo “ID” é enviado o número identificador do motor, no campo “Grandeza” pode ser enviado uma das seguintes letras: I (Corrente), V (Tensão), P (Potência), T (Temperatura) ou R (SetPoint e Estado), por fim nos campos Fase A, Fase B e Fase C são enviados os valores relativos à respetiva grandeza por fase.

Tabela 23 – Constituição da mensagem enviada para o módulo HC-05.

Motor	ID	Grandeza	Fase A	Fase B	Fase C
M	0001	I	Valor a	Valor b	Valor c

4.3.3 Comunicação Série

A comunicação série é realizada entre o dispositivo central (*master*) e um computador (PC) como demonstrado na Figura 55. Esta comunicação é necessária para concretizar o registo dos dados oriundos dos motores na base de dados.

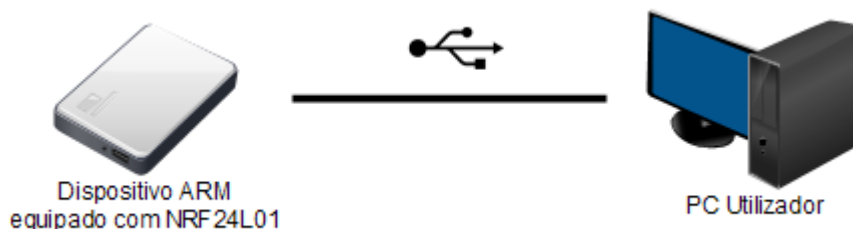


Figura 55 – Parte da rede de monitorização – Comunicação série.

O dispositivo central ao receber as mensagens de dados vai reencaminhá-las para o PC via comunicação série. A estrutura das mensagens (Tabela 24) enviadas e recebidas é idêntica.

Tabela 24 – Constituição da mensagem enviada pelo dispositivo central para o PC.

Motor	ID	Grandeza	Fase A	Fase B	Fase C
M	0003	V	AValor	BValor	CValor

Já do lado do PC, foi desenvolvido uma aplicação, utilizando o *software Visual Studio*, capaz de armazenar todas as mensagens recebidas. Depois de recebidas todas as mensagens de um motor, a aplicação utiliza um comando *INSERT MySQL* com toda a informação do motor e insere-a de uma só vez na base de dados criada para o efeito.

4.4 Arquitetura da aplicação web

De entre as arquiteturas padrões mais reconhecidas da internet estão as arquiteturas *LAMP*, *WAMP* e *MAMP*, populares pelo seu desempenho, robustez e qualidade da documentação disponível:

- ✓ *LAMP* – *Linux*, *Apache*, *MySQL* e *PHP*;
- ✓ *WAMP* – *Windows*, *Apache*, *MySQL* e *PHP*;
- ✓ *MAMP* – *Mac OS*, *Apache*, *MySQL* e *PHP*.

Estas arquiteturas conjugam diferentes tecnologias para prover ao programador as ferramentas necessárias ao desenvolvimento de um *website* dinâmico que se modifica à medida que interage com o utilizador. As arquiteturas analisadas são muito semelhantes, a única diferença entre *LAMP*, *WAMP* e *MAMP* é o sistema operativo base em que funcionam as restantes aplicações, no caso de *LAMP* – L deriva de *Linux*, no caso de *WAMP* – W deriva de *Windows* e no caso de *MAMP* – M deriva de *Mac OS*, existindo pequenas diferenças para quem desenvolve a aplicação *web*.

A Internet funciona com base no modelo cliente-servidor, através de um protocolo conhecido como *HTTP*. O utilizador requer um ficheiro que se encontra num computador remoto utilizando um endereço *URL*, sendo o pedido por norma efetuado através de um *web browser*, como por exemplo o *Chrome* ou *Mozilla*.

O servidor remoto recebe o pedido, processa-o e retorna o ficheiro requisitado pelo utilizador. Neste tipo de arquitetura, a tecnologia encarregue de atender os pedidos num servidor *web* é o servidor *Apache HTTP*, uma aplicação gratuita e bastante utilizada. Segundo dados provenientes da *Netcraft*, em Outubro de 2014, o *Apache* era líder do mercado com aproximadamente 37% de cota de mercado [75].

O ficheiro requisitado é normalmente enviado codificado em *HTML*, uma linguagem de marcação (“*Markup Language*”), que tem como função definir o conteúdo e o aspeto do ficheiro em questão e que o *web browser* interpreta para o apresentar ao utilizador final.

O modo simples como este serviço funcionava tornou a troca de informação um processo acessível, fiável e rápido, características que são apontadas como principais razões para o sucesso inicial na *web*. Contudo, com a evolução tecnológica e com a procura cada vez maior de informação, foram surgindo limitações à forma de comunicação adotada e por consequência começou-se a exigir mais capacidades de interação sobre a plataforma *web*.

Algumas limitações iniciais foram ultrapassadas, por exemplo, não era permitido ao cliente enviar dados para o servidor e por isso foi desenvolvido um mecanismo que permitia enviar informação através do endereço de recurso pretendido (método *GET*), ou anexando o conteúdo ao pedido do cliente (método *POST*). Os dados a serem enviados podem resultar, por exemplo, de um formulário previamente preenchido pelo utilizador. Outro exemplo de uma limitação que foi corrigida era o facto de o servidor não conseguir reconhecer e compreender que diversos pedidos eram feitos pelo mesmo cliente, com as vantagens de ai resultam, como restringir o acesso a zonas exclusivas através de autenticação ou adaptar o aspeto do interface gráfico ao gosto do utilizador. Uma das soluções encontradas foi a tecnologia *HTTP Cookies*, que permite guardar alguma informação do lado do cliente.

Assim que foi necessário processar os pedidos dos clientes e uma vez que esses pedidos eram muito mais do que um simples requisito do envio de um ficheiro estático, começaram a aparecer novas tecnologias com o intuito de facilitar a programação de aplicações *web*. A linguagem de programação *PHP* foi uma dessas tecnologias. Na verdade o *PHP* está integrado com o servidor Apache, processando todos os pedidos a determinados tipos de ficheiros (especialmente os que terminam em *.php*) e apresenta um conjunto de funcionalidade que permite o acesso facilitado a diversas bases de dados, tais como por exemplo *MySQL*, *PostgreSQL* e *Oracle*.

Os critérios subjacentes à escolha de um sistema de gestão de base de dados são variados e muito importantes, tais como: custo de licença, experiência da equipa de desenvolvimento, desempenho, quantidade de documentação, etc. Neste projeto em particular optou-se pelo *MySQL*, pois a sua licença é gratuita para projetos não comerciais.

Para aceder à base de dados são utilizadas funções do *PHP* que foram criadas especificamente para aceder ao *MySQL* e que estão devidamente documentadas em [76].

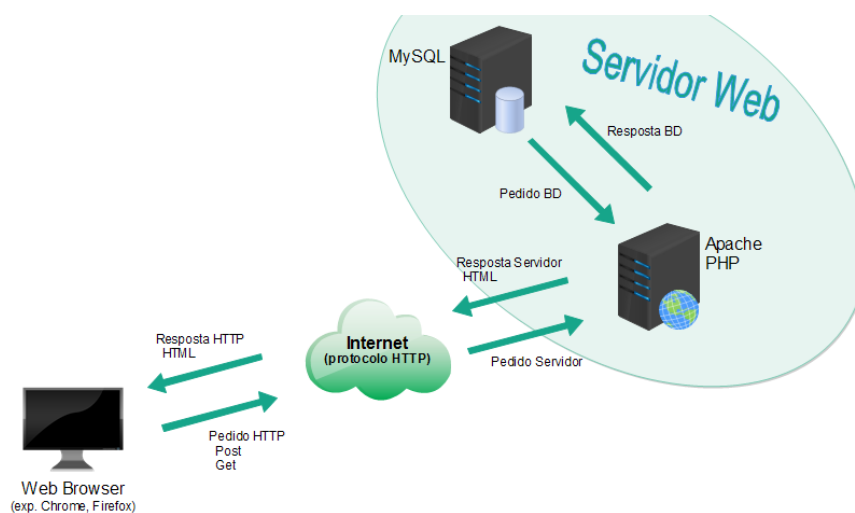


Figura 56 – Arquitetura *web*.

A arquitetura apresentada na Figura 56 é a arquitetura pretendida para este projeto, onde através de um *web browser* o utilizador tem acesso à informação armazenada na base de dados que por sua vez está alojada num servidor *web* remoto composto por um servidor Apache e pela base de dados MySQL.

4.5 Base de dados desenvolvida

No cenário industrial atual, em que cada vez mais os Sistemas de Informação e de armazenamento de dados são uma necessidade chave para o sucesso de uma empresa e em que os custos com licenciamento de *software* muitas vezes inviabilizam a modernização das empresas, o *software* aclamado *open source*, sem custos de licenciamento e/ou com custos reduzidos, poderá ser um importante contributo para diversas soluções.

Pretendendo oferecer ao utilizador um leque de informações em tempo real que o auxilie na análise e manutenção dos motores, onde seja possível a monitorização de diversos motores, assim como a visualização de um histórico de informação do motor, foi desenvolvida uma base de dados simples e flexível.

4.5.1 Análise do projeto

Depois de analisado o conjunto de informação necessária a ser enviada para a base de dados e determinado o tipo de informação vital a guardar na base de dados, foram criadas um conjunto de tabelas com diversos atributos.

As tabelas foram criadas com a ajuda do *software MySQL Workbench*, onde se configurou cada atributo dependendo da sua importância dentro da tabela e do seu tipo, como se pode verificar pela Figura 57.

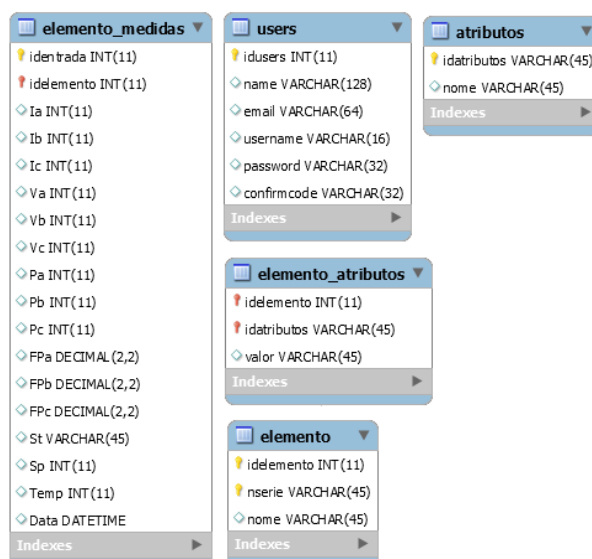


Figura 57 – Conjunto de tabelas desenvolvidas no MySQL Workbench.

Como se pode observar na figura anterior, foram criadas cinco tabelas com atributos bastante diversificados, de entre os quais é possível destacar as chaves primárias, as chaves estrangeiras e os atributos que estão dependentes da chave primária.

A chave primária é unívoca, isto é, o atributo ou atributos da chave primária têm um valor unívoco para qualquer registo da tabela, não podendo existir registos na tabela, que tenham o atributo da chave primária não preenchidos.

A chave estrangeira é um atributo ou um conjunto de atributos de uma tabela que aparece como chave primária numa outra tabela, permitindo estabelecer o relacionamento entre registos dessas tabelas. Uma tabela pode conter tantas chaves estrangeiras quantas necessárias, mas uma chave estrangeira tem de se relacionar sempre com uma chave primária na outra tabela, para que o relacionamento entre os registos seja correto.

Relativamente aos atributos, é necessário referir que estes possuem um conjunto de valores. O conjunto de todos os valores para um dado atributo constitui o domínio do atributo, logo, dependendo do domínio, o atributo pode ser representado por números, texto, datas, entre outros. A Tabela 25 apresenta todos os atributos presentes na base de dados.

Tabela 25 – Tabela com todos os atributos presentes na base de dados.

Tabela	Chave primária	Chave Estrangeira	Atributo
users	idusers	---	name, email, username, password, confirmcode
elemento	idelemento, nserie	---	nome
atributos	idatributos	---	nome
elemento_medidas	identrada	idelemento	Ia,Ib,Ic,Va,Vb;Vc, Pa;Pb,Pc,Fpa,FPb,FPc, St;Sp;Tep, data
elemento_atributos	---	idelemento, idatributos	valor

4.5.2 Diagrama Entidade – Relacionamento

O Diagrama Entidade-Relacionamento (DER) descreve toda a estrutura lógica de uma base de dados. Para a sua construção é necessário saber quais são os objetos e os seus relacionamentos.

O DER tem como ênfase os dados e os relacionamentos e a sua representação é feita da seguinte forma:

Retângulos: Representam entidades;

Círculos: Representam os atributos;

Losango: Representam os relacionamentos entre entidades;

Linhas: Unem os atributos aos conjuntos de entidades e os conjuntos de entidades aos conjuntos de relacionamentos.

No caso da base de dados desenvolvida a representação do DER é a apresentada na Figura 58.

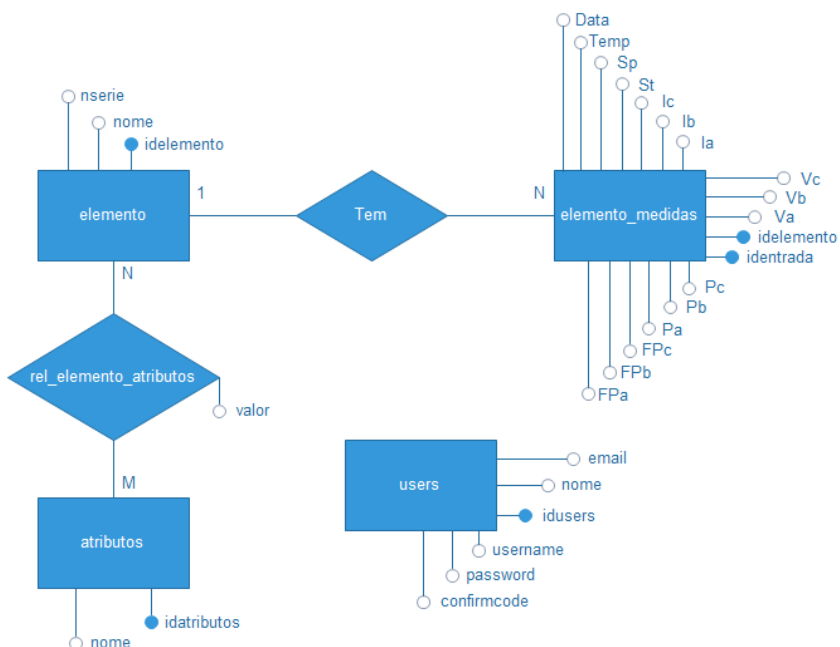


Figura 58 – Diagrama entidades relacionamento.

4.5.3 Diagrama físico

Utilizando ainda o *software MySQL Workbench*, todas as tabelas foram devidamente interligadas respeitando a 3ª Forma Normal, Figura 59.

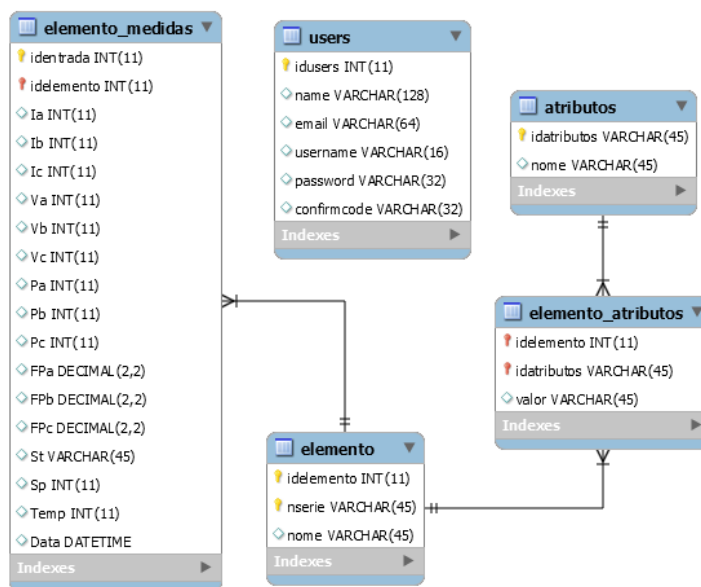


Figura 59 – Diagrama físico criado no MySQL Workbench.

Observando o diagrama acima apresentado, facilmente verificamos que a tabela “elemento” (onde ficará guardada o id, o número de série e o nome do motor) tem um relacionamento de

N:M com a tabela “atributos” (onde será guardado o id do atributo e o respetivo nome), por consequência surge a tabela “elemento_atributos” (onde ficará guardado o valor respeitante a um determinado motor e a um determinado atributo) que proporciona a relação entre as duas tabelas anteriormente apresentadas. A tabela “elemento_medidas” fica responsável por armazenar o id de entrada, o id do motor e os respetivos valores medidos e enviados (Ia, Ib, Ic, Va, Vb, Vc, Pa, Pb, Pc, FPa, FPb, FPc, St, Sp, Temp e Data).

Por fim foi criada a tabela “users” para possibilitar o registo dos utilizadores e para oferecer segurança na utilização da aplicação *web* da base de dados.

4.5.4 Implementação da base de dados

Depois de criadas todas as tabelas e determinadas todas as relações entre elas, no ambiente gráfico do *software* referido anteriormente, foram exportados todos os *scripts* da base de dados.

Nesta fase, com o auxílio do *MySQL Workbench*, foram exportados dois *scripts* da base de dados, um somente com as linhas de código necessárias para a criação de todas as tabelas presentes na base de dados e outro com alguns dados padrão e/ou representativos, inseridos na base de dados através do separador *Insert* de cada tabela.

Utilizando o *software XAMPP* e os *scripts* guardados, foi criada uma base de dados local para facilitar não só a sua manipulação como também para ajudar na elaboração de alguns testes.

Na tabela “atributos” (Tabela 26) foram introduzidos dados padrão que mais tarde estarão disponíveis para o utilizador os poder selecionar.

Tabela 26 – Dados inseridos na tabela atributos.

Tabela atributos	
idatributos	nome
anofab	Ano de Fabrico
cefi	Classe de Eficiência
ciso	Classe de Isolamento
cnomi	Corrente Nominal
fab	Fabricante
fpnomi	Fator de Potência Nominal
freq	Frequência (Hz)
graup	Grau Proteção
mod	Modelo
nserie	Número de Série
pnomi	Potência Nominal
tnomi	Tensão Nominal

Para manipular a informação gravada na base de dados foram usados os comandos de *SELECT* (selecção), *INSERT* (inserção), *UPDATE* (alteração), *DELETE* (apagar) e *SHOW* (mostrar).

Estes quatro comandos são a base da linguagem SQL, manipulando-os consegue-se obter uma grande quantidade de comandos específicos, que permitem extrair da base de dados qualquer tipo informação contida na mesma.

De entre eles o comando **SELECT** permite seleccionar informação contida na base de dados. O comando pode ser empregado usando a sintaxe e exemplo que a seguir se apresenta:

Sintaxe:

```
SELECT coluna1, coluna2,...  
FROM tabela  
WHERE condição  
ORDER BY coluna ASC;
```

Exemplo:

```
mysql > SELECT * FROM elemento  
WHERE  
nome='Motor Azul';
```

Já o comando **INSERT**, é utilizado para introduzir informação na base de dados. O comando pode ser utilizado como se demonstra no exemplo seguinte:

Sintaxe:

```
INSERT INTO tabela (coluna1,  
coluna2,...)  
VALUES (valor1, valor2,...)  
WHERE condição  
ORDER BY coluna ASC;
```

Exemplo:

```
mysql > INSERT INTO elemento  
(idelemento, nserie, nome)  
VALUES  
(null, '1234-GT', 'Motor Verde');
```

O comando **UPDATE** é utilizado para executar alterações de dados na base de dados. Em seguida é apresentado um exemplo de construção do comando

Sintaxe:

```
UPDATE tabela  
SET coluna1=valor1,coluna2=valor2,...  
WHERE condição;
```

Exemplo:

```
mysql > UPDATE elemento  
SET nome='Motor Vermelho'  
WHERE nserie='1234-GT';
```

Com o comando **DELETE** é possível eliminar dados da base de dados se se utilizar a sintaxe apresentada em seguida.

Sintaxe:

```
DELETE FROM tabela
```

WHERE condição;

Exemplo: `WHERE nserie='1234-GT';`

`mysql > DELETE FROM elemento`

Por fim, o comando **SHOW** permite listar as tabelas não temporárias da base de dados. Em seguida é demonstrado a sintaxe e um exemplo de como se pode utilizar o comando.

Sintaxe:

`SHOW COLUMNS`

`FROM tabela;`

Exemplo:

`mysql > SHOW COLUMNS`

`FROM tabela;`

4.6 Website desenvolvido

Em Portugal a percentagem de indivíduos que usam a internet tem vindo a crescer de ano para ano. Em 2013, segundo a UNSD e o DESA [77], a percentagem de utilização da internet era aproximadamente de 62 em cada 100 pessoas. Estes números são indicadores de que a população está a acompanhar a evolução tecnológica e que facilmente se adapta às novas tecnologias. Para aceder à internet os utilizadores utilizam aplicações de acesso à Internet, vulgarmente designadas por *web browsers*. De entre os mais conhecidos e utilizados estão o Chrome da Google, o Internet Explorer da Microsoft e o Firefox da Mozilla Foundation.

Tendo em conta que os *browser's* são utilizados mundialmente e em larga escala, esta plataforma foi a escolhida para se desenvolver o interface e proporcionar ao utilizador uma curva de aprendizagem mínima assim como uma maior facilidade de acesso aos dados armazenados na base de dados.

4.6.1 Processo de Desenvolvimento do Website

Para iniciar a construção de *websites* é importante elaborar um planeamento do projeto e dividi-lo em etapas de menor dimensão para proporcionar uma melhor organização na execução das diversas das tarefas de acordo com o diagrama apresentado na Figura 60.

Apesar do *website* pretendido ser um pouco diferente dos demais sites disponíveis na internet que qualquer pessoa pode aceder, também este, por uma questão organizacional, foi elaborado tendo em conta as etapas de desenvolvimento usuais de um *website*.



Figura 60 – Diagrama do processo de desenvolvimento.

O desenvolvimento de um *site* pode passar por seis fases, contudo a sua divisão não significa a existência exata de uma fronteira entre elas. Na verdade, no desenvolvimento do projeto as etapas interagem enquanto são redefinidos alguns detalhes do projeto.

O processo de desenvolvimento de um *site* pode ser dividido nas etapas seguintes:

Etapa 1 - Planeamento e definição;

Nesta etapa inicial, definem-se as metas e os objetivos para o *website* e recolhe-se as informações necessárias para o desenvolvimento do mesmo. Esta é também uma etapa na qual se define a extensão do conteúdo, as funcionalidades e as tecnologias necessárias para o desenvolvimento do *website*, pois os sites não são todos iguais e nem têm o mesmo público-alvo. Este projeto é um exemplo perfeito de como os *websites* podem ser diferentes, pois ao contrário dos demais, este destina-se especificamente a um número reduzido de utilizadores e pretende ser uma aplicação de controlo e monitorização de motores em ambiente industrial.

Por fim, nesta etapa é preciso detalhar o conteúdo e fazer uma organização do *website*, inventariar todo conteúdo existente e descrever, no caso de existir, novo conteúdo.

Etapa 2 - Especificação e análise de requisitos;

Após o planeamento estar concluído, segue-se a especificação e análise de requisitos. Esta etapa consiste em analisar o que é requerido para o desenvolvimento do projeto e determinar se existe ou não a necessidade de adquirir mais recursos para se dar início ao mesmo.

Etapa 3 - Design;

É na etapa 3 que se planeia e se desenvolve o *layout* para o *website*. O *layout* do *website* é concebido tendo em conta o propósito do *site* e os elementos recolhidos anteriormente. São desenvolvidos alguns protótipos para efeito de testes.

Etapa 4 - Desenvolvimento;

Na etapa de desenvolvimento é feita a ligação com a base de dados, implementação de sistemas de segurança e com os devidos comandos que manipulam as bases de dados são preenchidas as páginas com os respetivos conteúdos. Nesta etapa a estrutura do *website* deve estar praticamente definida.

Etapa 5 - Teste e Publicação;

Esta etapa é vital, pois a realização de testes é de extrema importância para o bom funcionamento do *site* aquando da sua publicação. Estes testes devem ser realizados por alguém externo ao projeto, pois geralmente quem está ligado ao projeto não vê pequenos detalhes que podem fazer toda a diferença na solução final. Um utilizador externo irá notar, em detalhes e poderá tecer críticas construtivas no sentido de melhorar a qualidade e funcionalidades do referido *website*.

Etapa 6 - Atualização e Manutenção.

Por fim a atualização e manutenção do *website* em termos de conteúdos, não é da responsabilidade do utilizador nem de quem desenvolveu o *website*, contudo quem o desenvolveu tem a responsabilidade de garantir o seu bom funcionamento em termos técnicos.

4.6.2 Conceção do Website

Na conceção do *website* foram tidas em conta as etapas de desenvolvimento apresentadas no subcapítulo anterior. Pretende-se com este subcapítulo revelar o que foi desenvolvido em cada uma das etapas apresentadas, demonstrando a sua implementação através das linhas de código e resultados obtidos utilizando algumas imagens representativas.

As primeiras fases do processo de desenvolvimento de um *website* compreendem, como vimos anteriormente, a recolha de informação e desenho do *layout*, mas parte da informação necessária já tinha sido recolhida aquando do desenvolvimento da bases de dados, logo só foi necessário perceber que parte da informação obtida seria fundamental disponibilizar via *website* ao utilizador.

O *layout* pretendido para o *website* era uma grelha dinâmica que apresentasse um botão por cada motor que tivesse conectado à rede e que desse acesso a informação detalhada de cada um. Foi igualmente incluída uma página de *login* para adicionar alguma segurança à aplicação. Em seguida foram desenvolvidas todas as páginas necessárias com as tecnologias html e css e contruída toda a estrutura do *website*. Terminada essa atividade, foi incorporado nas páginas necessárias o código php que permitisse o acesso e integração de toda a estrutura de navegação (Figura 61) com a base de dados desenvolvida anteriormente.

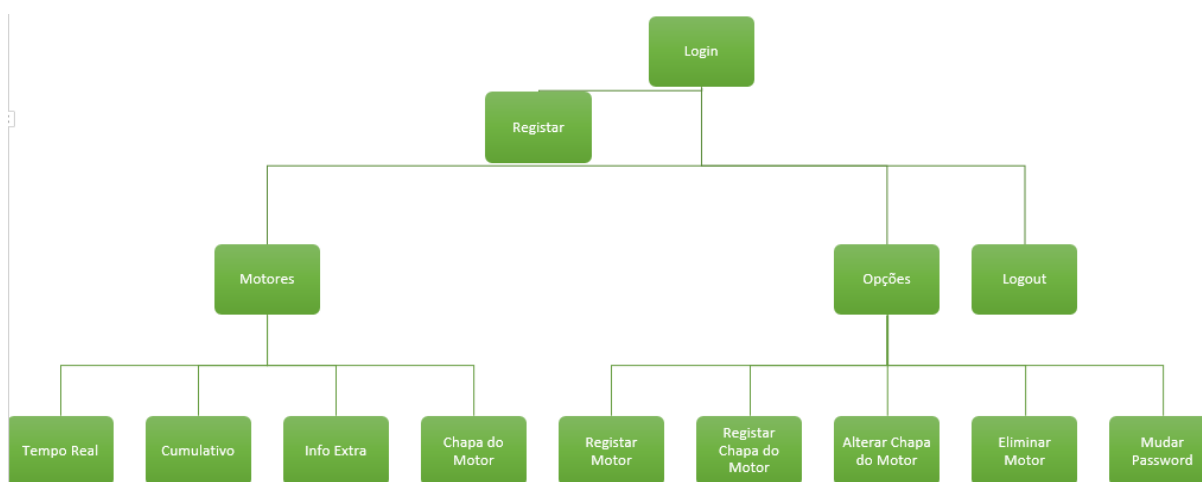


Figura 61 – Estrutura de navegação.

No desenvolvimento do *website* foram concebidas várias páginas em php e html, sendo elas: a página de login, a página de registo de utilizadores, a página inicial/motores, a página para registar motores, a página para registar a chapa do motor, a página para alterar a chapa do motor, a página para eliminar um motor, a página para alterar de *password*, a página de menu do motor e por fim quatro páginas popup: página Tempo Real, página Cumulativo, página Chapa do motor e página Info Extra. Em seguida são descritos em detalhe os registos de cada uma das páginas apresentadas anteriormente, particularmente o conteúdo e funcionalidades de cada uma delas.

Página Login

A página “login” presente na Figura 62, contém o formulário de *login* (campos *username* e *password*). Este formulário impede que o utilizador submeta dados inválidos ao servidor, isto é, o formulário impede que o utilizador submeta os campos *username* e *password* vazios e exhibe um *feedback* informativo relativo à falha que ocorrer.

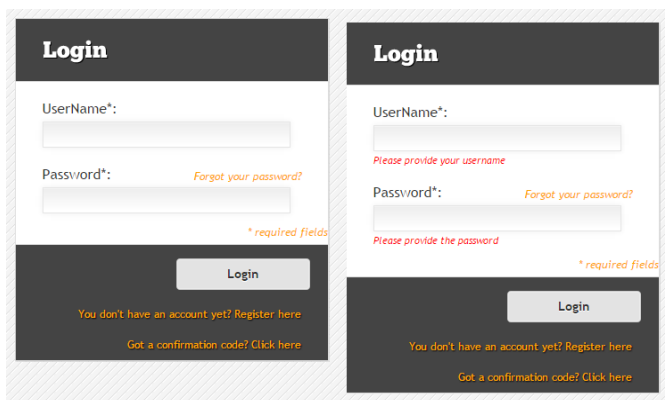


Figura 62 – Página de autenticação de utilizador.

Página Registrar

A página “registar”, apresentada na Figura 63, contém um formulário que permite ao utilizador não registado criar uma conta que lhe dará acesso à página inicial. Para se registar o utilizador terá de introduzir o seu nome completo, o *email*, o *username* e uma *password* (o formulário oferece um gerenciador de *passwords*). Este formulário tem proteção *antibot spam* que impede os *bots* de tentarem criar múltiplas contas assim como também impede que o utilizador introduza dados inválidos. Quando o utilizador faz um registo, o acesso à página inicial não é imediato, o utilizador terá primeiro de introduzir um código de confirmação de conta o qual irá receber por correio eletrónico.

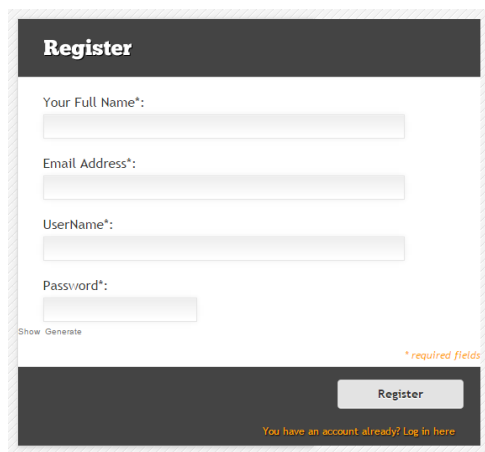


Figura 63 – Página de registo de utilizador.

Página Inicial/Motores

A página “inicial/motores”, apresentada na Figura 64, está dividida em três partes. A primeira é o cabeçalho que se encontra disposto horizontalmente no topo da página e que contém um

menu com *link* para outras páginas. A segunda é uma coluna que se encontra à esquerda da página e que contém o logotipo da aplicação. A última parte é uma coluna que contém uma grelha dinâmica e interativa por meio de botões com os motores que estão conectados e registados na base de dados. Se o motor estiver registado na base de dados é possível seleccionar o seu símbolo e abrir a página “motor” respetiva.

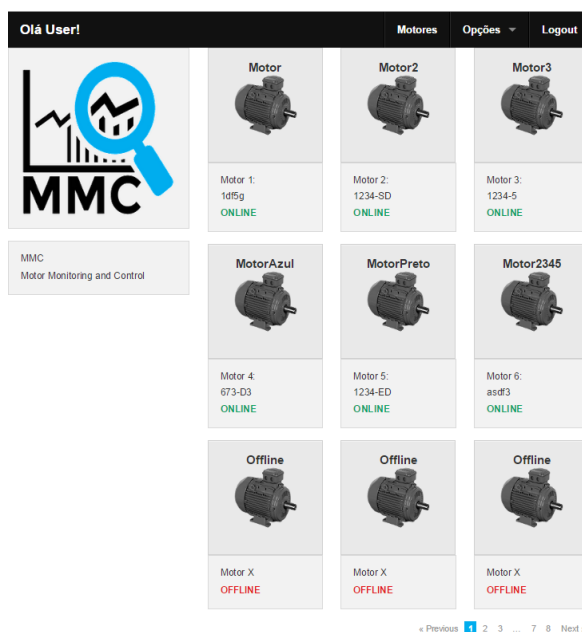


Figura 64 – Página de inicial, com a grelha dinâmica de motores.

Página Registrar Motor

Esta página contém um formulário que permite ao utilizador registar um motor na base de dados. O formulário é composto pelos campos nome e número de série do motor. Depois de registar o motor o utilizador é encaminhado para a página “registar chapa do motor”.

Página Registrar Chapa do Motor

A página “Registrar Chapa do Motor” como o próprio nome indica permite ao utilizador registar a chapa característica do motor. Esta página é composta por um formulário onde o utilizador terá de seleccionar o motor para o qual se pretende registar a chapa do motor e posteriormente preencher os campos fornecidos com os dados mais importantes presentes numa chapa característica de um motor, tais como: fabricante, ano de fabrico, modelo, grau de protecção, frequência, potência, tensões, ligações, fator de potência, entre outros.

Página Alterar Chapa do Motor

A página “Alterar Chapa do Motor” possibilita ao utilizador alterar os dados da chapa característica do motor anteriormente inserido na base de dados. O formulário apresentado contém os mesmos campos que a página “registar chapa característica”.

Página Eliminar Motor

A página “eliminar motor” permite que o utilizador apague qualquer registo de um motor anteriormente registado na base de dados. Para eliminar o motor basta o utilizador seleccionar o identificador do motor que pretende eliminar.

Página Mudar Password

Com a página “Mudar Password” o utilizador poderá atualizar a sua *password*, para isso terá de introduzir no formulário apresentado a sua *password* antiga e a nova *password* (o formulário oferece um gerenciador de *passwords*).

Página do Motor

A página “Motor”, representada na Figura 65 está dividida em três partes assim como a página “inicial/motores” e surge quando o utilizador selecciona um dos motores da grelha dinâmica apresentada na página “inicial/Motores”. Depois de seleccionado o motor o utilizador tem acesso a quatro botões com *links* para as respetivas páginas: Chapa do Motor, Tempo Real, Cumulativo e Info Extra.



Figura 65 – Página do motor.

Página Visualiza Chapa do Motor

A página “Visualiza Chapa do Motor”, representada na Figura 66 disponibiliza ao utilizador os dados presentes na chapa do motor com um *layout* idêntico ao de uma chapa característica verdadeira.

Schneider		2000	
Motor	3 ~	50Hz	-----
Modelo: W21 - BP1-2R		N° 1df5g	
P.N.: 15kW		-----	
-----		C.I.: F	Cos φ = 0.8
Y		---V ---A	
-----		230V	50A
-----		C. E.: EFF1	-----
-----		G.P.: IP 54	Kg

Figura 66 – Página que permite ver a chapa característica do motor.

Página Tempo Real

A página “Tempo Real”, apresentada na Figura 67 é uma janela *popup* que apresenta ao utilizador, dependendo do motor anteriormente selecionado, os valores por fase totais relativos à potência ativa, potência reativa e potência aparente.

	A	B	C	T
Tempo e data atual:				
It is Fri on Dec 12, 2014, 05:02:42				
P (W)	2599	2421	2243	7263
Q1 (VAR)	4880	4546	4211	13637
S (VA)	4130	3848	3564	11542

Figura 67 – Janela popup – página Tempo Real.

Página Cumulativo

Por seu lado a página “Cumulativo”, apresentada na Figura 68, providencia um gráfico com os valores do motor relativos à opção pretendida pelo utilizador, a qual foi previamente seleccionada. O utilizador pode seleccionar uma das seguintes opções: Corrente, Tensão, Fator de Potência e Potência. Para além disso é possível, ao utilizador, ver toda esta informação fase a fase, ou as três fases em simultâneo ou ainda visualizar o total.

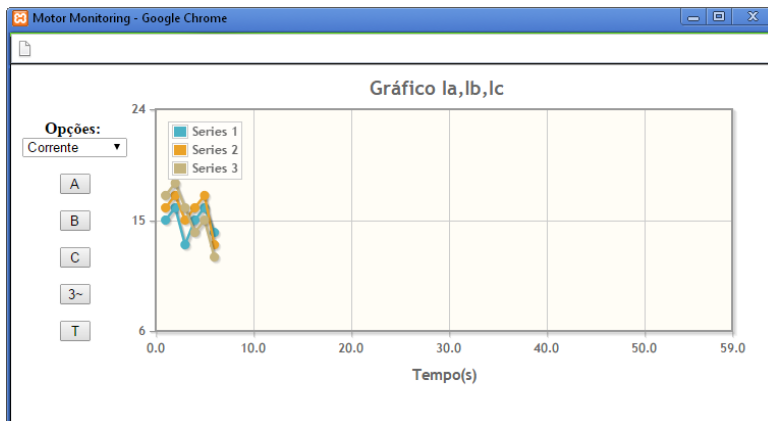


Figura 68 – Janela popup – página Cumulativo.

Página Info Extra

A página “Info Extra”, apresentada na Figura 69, permite ao utilizador visualizar uma tabela com todos os registos do motor seleccionado, os quais se encontram armazenados na base de dados. Toda a informação pode ser ordenada consoante a necessidade e vontade do utilizador.

Ia	Ib	Ic	Va	Vb	Vc	Pa	Pb	FPa	Pc	FPb	FPc	St	Sp	Temp	Dia-Hora
15	16	17	280	290	300	2928	3199	0.80	3479	0.81	0.82	1	1500	78	2014-09-10 10:10:01
16	17	18	281	291	301	2987	3227	0.85	4143	0.86	0.70	1	1600	59	2014-09-10 10:10:02
13	15	16	278	287	289	2491	2936	0.81	3254	0.82	0.79	1	1450	55	2014-09-10 10:10:03
15	16	14	280	290	300	2958	3298	0.79	3015	0.78	0.77	1	1450	95	2014-09-10 10:10:04
16	17	15	300	301	302	2983	3140	0.90	2744	0.91	0.92	1	1800	60	2014-09-10 10:10:05
14	13	12	295	295	297	2599	2421	0.89	2243	0.89	0.89	1	2000	65	2014-09-10 10:10:06

Figura 69 – Janela popup – página Info Extra.

4.7 Ambiente de testes

Os protótipos apresentados durante a descrição do projeto foram desenvolvidos e testados nas instalações da empresa Optisigma sediada no Inopol - Academia de Empreendedorismo situada no campus da Escola Superior Agrária de Coimbra.

O laboratório da Optisigma dispõe de um vasto conjunto de equipamentos de medida e dispositivos eletrónicos os quais são usados na análise do comportamento de motores. O referido laboratório dispõe de uma bancada de testes (Figura 70, imagem à direita) equipada com um motor indução trifásico (4 KW de potência), um servomotor da SEW Eurodrive, um MOVIDRIVE (variador eletrónico de velocidades) do mesmo fabricante e o respetivo quadro elétrico de proteção. Do ponto de vista mecânico o motor elétrico está fixado numa estrutura metálica estando o seu veio acoplado ao servomotor (carga variável) sendo este controlado pelo VEV.

Os testes efetuados ao InSwitch foram realizados na bancada anteriormente descrita. Na imagem Figura 70 a) é possível visualizar o InSwitch instalado na caixa de terminais do motor.



Figura 70 – InSwitch instalado na caixa de terminais do motor e bancada de testes, respetivamente.

Após se ter validado o correto funcionamento do InSwitch foi possível incluir outros módulos ao protótipo desenvolvido neste projeto, ao microcontrolador STM32F4 foi adicionado o módulo HC-05 (Figura 71) descrito no tópico 4.2.3.

A inclusão de funcionalidades de comunicação no protótipo disponibiliza aos técnicos de manutenção, uma ferramenta de análise flexível e fiável para a monitorização e controlo do motor de indução trifásico em tempo real.



Figura 71 – Placa STM32F4 equipado com o módulo Bluetooth HC-05.

Após validação do protótipo anterior, a equipa de desenvolvimento da Optisigma projetou uma placa de circuito impresso proprietária (Figura 72) que inclui o microcontrolador ARM Cortex-M4 da STMicroelectronics, o módulo HC-05 Bluetooth, uma ligação micro USB e um barramento de interface para a placa de potência.

Uma vez equipado o InSwitch com o módulo HC-05, este foi programado de forma a enviar e receber dados via Bluetooth. Como dispositivo terminal foi utilizado um *tablet* com o sistema operativo android no qual foi instalada uma aplicação android desenvolvida para o efeito, a qual foi apresentada no tópico 4.2.3). Esta aplicação pretende controlar e monitorizar o motor de indução trifásico com recurso a uma ligação rádio sem fios.

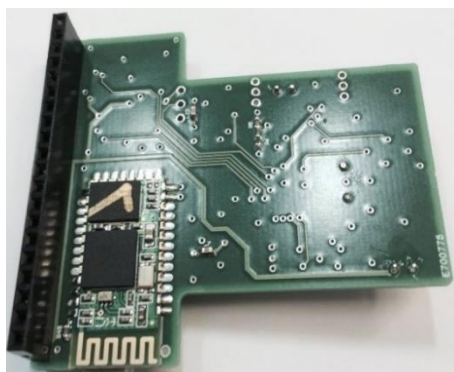


Figura 72 – Módulo HC-05 incorporado na placa de controlo do InSwitch.

Verificou-se que a aplicação android funcionava corretamente. No entanto o raio de alcance do Bluetooth ficou reduzido para 1-2 metros, contudo esse aspeto não é crítico visto que o principal objetivo é o técnico estar perto do motor que pretende analisar.

Foi desenvolvido um segundo protótipo o qual inclui a placa de desenvolvimento STM32F4 e o módulo de comunicação rádio nRF24L01+ representado na Figura 73. O objetivo deste protótipo visava o estabelecimento de funcionalidades de comunicação entre os diversos motores de uma instalação industrial (*slaves*) com um nó central designado por master.

O microcontrolador deste nodo central comunica através de uma porta série com um servidor e respetiva base de dados. Posteriormente foi desenvolvido um *website* de forma a disponibilizar a informação dos motores a um utilizador, em tempo real, com acesso a Internet. Tendo-se verificado a operacionalidade da solução desenvolvida foram validadas as comunicações entre quaisquer dois protótipos STM32F4 através dos módulos nRF24L01+ incorporados. Ficou também patente a validade comunicação entre o master e o servidor que contém a base de dados.

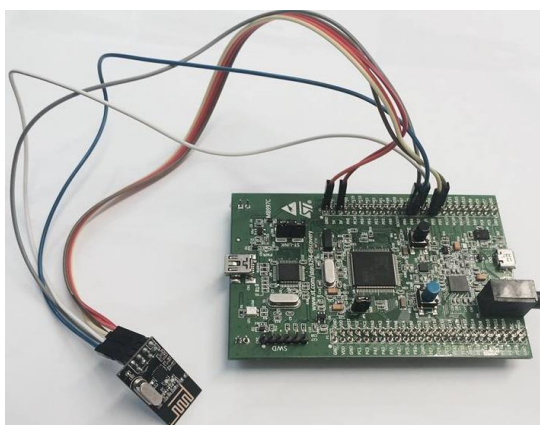


Figura 73 – Placa STM32F4 equipado com o módulo nRF24L01+.

4.8 Conclusão

Tendo em conta os protótipos elaborados neste capítulo podemos concluir que o módulo *RFM12b*, módulo previamente selecionado responsável pela comunicação entre o motor e o dispositivo mestre, não foi a melhor escolha. O módulo *RFM12b* apresentou alguns obstáculos aquando a configuração do mesmo por forma a se conectar e comunicar com a placa *STM32F4DISCOVERY*, devido à falta de bibliotecas e pouca informação disponibilizada. Assim, o módulo rádio *RFM12b* foi substituído pelo módulo nRF24L01+, módulo mais económico e de maior potência. Também o protótipo com a placa base *STM32F4DIS-BB*, sistema que seria responsável por alojar uma base de dados, um *site* e por fazer a comunicação *Ethernet* entre o dispositivo mestre e o computador, não foi satisfatório. Portanto este sistema foi substituído por comunicação série possibilitando o alojamento de uma base de dados e de um *site* num computador.

Capítulo 5

CONCLUSÕES

O projeto realizado no presente trabalho resulta do somatório de diversas fases, que compreenderam a:

- ✓ Recolha de material didático, artigos e informação relacionada com o projeto a desenvolver;
- ✓ Realização de alguns testes às ferramentas, tecnologias e *softwares* selecionados, com o objetivo de se aferir o funcionamento do material das mesmas.
- ✓ Elaboração de protótipos com o material selecionado;
- ✓ Aplicação dos resultados obtidos com os protótipos, para o desenvolvimento de uma rede de monitorização de motores de indução;
- ✓ Desenvolvimento da base de dados e do interface *web* com o intuito de oferecer ao utilizador, uma forma rápida e simples, de aceder a toda a informação útil para análise do funcionamento dos motores.

A primeira fase relativa à recolha de material didático é de elevada importância para o sucesso do projeto, pois é nesta fase que se consolidam os conhecimentos e se interiorizam novas ideias. A evolução dos motores elétricos, até ao presente, foi de tal forma que pode dar a sensação de que nada mais é possível desenvolver nesta matéria.

Podemos concluir ainda que com o passar dos anos, o tráfego de informações num barramento industrial tem vindo a aumentar, desta forma, houve a necessidade da conceção de novos sistemas de comunicação na procura de minimizar a complexidade das interligações entre dispositivos e/ou equipamentos nos processos industriais. Assim sendo tem-se vindo a destacar a utilização dos sistemas sem fios, com a finalidade de aperfeiçoar a ligação física entre os diversos elementos, para além de apresentarem menor custo de investimento aquando a sua instalação. A utilização dos sistemas sem fios pode representar uma poupança económica pois este tipo de comunicação não utiliza cabos de transporte de dados. É importante contudo, perante diversas tecnologias sem fios, saber qual se adequa melhor a cada aplicação específica.

Na segunda fase, é de salientar que alguns protótipos não tiveram o resultado esperado, como foi o caso do módulo RFM12b e da placa base *STM32F4DIS-BB*. No caso do módulo RFM12b,

as incompatibilidades, falta de bibliotecas funcionais e falhas na comunicação verificadas levaram a que este fosse substituído por outro módulo RF de custo mais reduzido, de maior potência, mas com alcance menor, o nRF24L01+. A placa base *STM32F4DIS-BB*, que tinha como função adicionar ligação *Ethernet* à placa *STM32F4DISCOVERY*, também foi descartada por não atingir os resultados esperados. A placa base *STM32F4DIS-BB*, mesmo tendo em conta o bom desempenho que apresentava, não possuía memória suficiente para alojar o *site* e a base de dados, apesar de possuir suporte para cartão MicroSD, a inclusão deste iria acrescentar custos significativos ao projeto, tendo em conta o elevado número de motores presentes numa instalação industrial.

Em relação à terceira fase é de se referir que a comunicação com múltiplos master slave está funcional através de com o algoritmo de comunicação desenvolvido. Os motores vão enviar de segundo em segundo toda a informação relevante através de mensagens previamente programadas. Depois de enviadas as mensagens são armazenadas na base de dados.

Em relação à quarta fase, é importante sublinhar que foi realizado um estudo para melhor se perceber a conceção da arquitetura de uma aplicação *web*. Uma vez que numa fase inicial tudo seria desenvolvido localmente, era necessário garantir a portabilidade da aplicação para outros serviços *online*, isto é, garantir que em caso de necessidade a migração de toda a aplicação para um alojamento *online*, ocorresse sem grandes alterações. Apesar de não ter sido referido ao longo do desenvolvimento da tese, a base de dados e aplicação *web* foram alojadas num servidor *online* e testados via *internet*, através do serviço de alojamento gratuito oferecido pela *Microsoft* – o *Microsoft Azure*. Com este serviço é possível alojar gratuitamente durante 30 dias, tanto o *site* como a base de dados, num servidor da *Microsoft*, sendo disponibilizado um *link* pelo serviço *Azure* para o acesso ao *site*.

5.1 Desenvolvimentos futuros

De acordo com o trabalho realizado, acha-se oportuno referir o seguinte:

Por forma a melhorar ainda mais o serviço que se pretende oferecer com este projeto (auxiliar na manutenção dos motores de indução), acredita-se que seria interessante o desenvolvimento de um algoritmo capaz de prever o momento aproximado em que um motor realmente necessita de manutenção, assim como a implementação de código que possibilitasse a comunicação entre os diversos motores criando uma rede inteligente de motores.

Também os algoritmos que foram desenvolvidos e que são responsáveis pelo controlo da comunicação entre os diversos dispositivos podem ser melhorados relativamente à segurança.

Capítulo 6

BIBLIOGRAFIA

- [1] “MAGCRAFT,” [Online]. Available: <http://www.rare-earth-magnets.com/history-of-magnetism-and-electricity/>.
- [2] “Sparkmuseum,” [Online]. Available: <http://www.sparkmuseum.com/MOTORS.HTM>. [Acedido em 2014].
- [3] “Tesla Society,” [Online]. Available: http://www.teslasociety.com/hall_of_fame.htm. [Acedido em 2014].
- [4] “ABB - MIT,” [Online]. Available: [http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/5e6a1c128ae4fab1c1257b490033f301/\\$file/1sdc007106g0201.pdf](http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/5e6a1c128ae4fab1c1257b490033f301/$file/1sdc007106g0201.pdf). [Acedido em 2014].
- [5] A. M. S. Francisco, “Motores de Indução Trifasicos,” [Online]. Available: http://www.estgv.ipv.pt/PaginasPessoais/lpestanda/maquinas%20el%C3%A9ctricas%202/aulas%20te%C3%B3ricas/Motores_inducao_tri.pdf. [Acedido em 2014].
- [6] S. Rahman, “Renewable and Sustainable Energy Reviews,” *Department of Mechanical Enginneering, University of Malaya*, 2010.
- [7] A. T. Almeida, F. J. T. E. Ferreira e A. Quintino, “Technical and Economical Considerations on SuperHigh-Efficiency Three-Phase Motors,” 2012.
- [8] B. Cassoret, S. Lopez, J.-F. Brudny e T. Belgrand, “Non-Segmented Grain Oriented Steel in Induction Machines,” *Progress In Electromagnetics Research C, Vol. 47, 1-10.*, 2014.
- [9] R. Hanitsch, “Energy Efficient Electric Motors,” *University of Technology Berlin, Germany*, 2002.
- [10] A. T. Almeida, F. J. T. E. Ferreira e A. Quintino, “Automatic Change of the Stator-Winding Connection of Variable-Load Three-Phase Induction Motors to Improve the Efficiency and Power Factor,” 2005.

-
- [11] “SinuMEC,” [Online]. Available: <http://www.powersines.com/Electric-Motor-Efficiency/SinuMEC>. [Acedido em 2014].
- [12] Softstarters, [Online]. Available: [http://library.abb.com/GLOBAL/SCOT/SCOT209.nsf/VerityDisplay/2985284834BCFF7FC1256F3A00274038/\\$File/1SFC132002M0201.pdf](http://library.abb.com/GLOBAL/SCOT/SCOT209.nsf/VerityDisplay/2985284834BCFF7FC1256F3A00274038/$File/1SFC132002M0201.pdf). [Acedido em 2014].
- [13] smar, “automação empresarial,” [Online]. Available: <http://www.smar.com/PDFs/Catalogues/SYSTEM32CP.pdf>. [Acedido em 2014].
- [14] “Pirâmide Da Automação Industrial,” [Online]. Available: <http://www.automacaoindustrial.info/a-piramide-da-automacao-industrial/>. [Acedido em 2014].
- [15] R. Zurawski, “Industrial Communication Technology and Systems,” em *The Industrial Communication Technology Handbook*, 2005.
- [16] smar, “Foundation Fieldbus,” [Online]. Available: www.smar.com/PDFs/Manuals/GERAL-FFMP.pdf. [Acedido em 2014].
- [17] mecatronicaatual, “Tecnologia Foundation Fieldbus Parte 1: Fundamentos e Principais Características,” [Online]. Available: <http://www.mecatronicaatual.com.br/educacao/1410-descobrimdo-a-tecnologia-foundation-fieldbus-parte-1-fundamentos-e-principais-caractersticas?showall=&limitstart=0>. [Acedido em 2014].
- [18] “Fieldbus,” [Online]. Available: <http://archive.sensorsmag.com/articles/0200/63/opc.shtml>. [Acedido em 2014].
- [19] “Modbus,” Modbus, [Online]. Available: <http://www.modbus.org/faq.php>. [Acedido em 2014].
- [20] I. MODICON, “Modicon Modbus Protocol Reference Guide,” [Online]. Available: http://modbus.org/docs/PI_MBUS_300.pdf. [Acedido em 2014].
- [21] S. Erelectric, “Redes de Comunicação Industrial,” [Online]. Available: http://www.schneiderelectric.pt/documents/product-services/training/doctecnico_redes.pdf. [Acedido em 2014].

- [22] “ModBus,” [Online]. Available: <http://www.mecatronicaatual.com.br/educacao/1678-redes-de-clps-parte-2>. [Acedido em 2014].
- [23] P. -. C. P. Overview, “<http://www.rtaautomation.com/>,” [Online]. Available: <http://www.rtaautomation.com/technologies/profibus/>. [Acedido em 2014].
- [24] C. Cassiolato, “TECNOLOGIA PROFIBUS,” [Online]. Available: <http://www.smar.com/brasil/artigo-tecnico/tecnologia-profibus>. [Acedido em 2014].
- [25] “Profibus,” [Online]. Available: <http://www.mecatronicaatual.com.br/educacao/1156-redes-da-organizacao-profibus>. [Acedido em 2014].
- [26] Cisco, “Carrier Sense Multi-Access/Collision Detection (CSMA/CD),” [Online]. Available: http://www.cisco.com/en/US/tech/tk389/tk214/tk125/tsd_technology_support_sub-protocol_home.html. [Acedido em 2014].
- [27] A. D. L. Fernandes, “POWER LINE COMMUNICATION IN ENERGY MARKETS,” [Online]. Available: <http://www.cypress.com/?docID=31441>. [Acedido em 2014].
- [28] “Ariane Controls,” [Online]. Available: http://www.arianecontrols.com/documents/Narrowband_Powerline_Communication_Applications_and_Challenges_March2013.pdf. [Acedido em 2014].
- [29] “PLC,” [Online]. Available: http://www.teleco.com.br/tutoriais/tutorialkbns/pagina_3.asp. [Acedido em 2014].
- [30] K. D. M. D. K. K. T. A. R. Maciej, “Analysis of Industrial PROFINET in the Task of Controlling a Dynamic System,” [Online]. Available: <http://journals.bg.agh.edu.pl/AUTOMAT/2013.17.1/automat.2013.17.1.65.pdf>.
- [31] profibus.com, “PROFINET - Descrição do Sistema,” [Online]. Available: <http://www.profibus.com/nc/download/technical-descriptions-books/downloads/profinet-technology-and-application-system-description/download/15440/>. [Acedido em 2014].
- [32] U. o. A. S. Duesseldorf, “PROFINET ON PHOENIX CONTACT PLATFORM,” [Online]. Available: http://home.agh.edu.pl/~ipnet/Materials1/Module3/CML3_profinet_basics_students_checked.pdf. [Acedido em 2014].

- [33] ABB, “AC 800M PROFINET IO Configuration,” [Online]. Available: https://library.e.abb.com/public/f355a67551218ae7c1257dc0003298c5/3BDS021515-600_-_en_AC_800M_6.0_PROFINET_IO_Configuration.pdf. [Acedido em 2014].
- [34] PInternational, “Profibus and Profinet Glossary,” [Online]. Available: <https://www.google.pt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0CDEQFjAD&url=http%3A%2F%2Fwww.profibus.com%2Fnc%2Fdownload%2Ftechnical-descriptions-books%2Fdownloads%2Fglossary%2Fdownload%2F188%2F&ei=J2VsVbu4IMj2UvLQgLAO&usg=AFQjCNG2Ij7>. [Acedido em 2014].
- [35] “Profinet,” [Online]. Available: https://www.phoenixcontact.com/online/portal/br?1dmy&urile=wcm:path:/brpt/web/main/products/technology_pages/subcategory_pages/profinet/6b88e906-a40e-4922-8cc0-8dff087c29b1/6b88e906-a40e-4922-8cc0-8dff087c29b1. [Acedido em 2014].
- [36] “DeviceNet,” [Online]. Available: <http://www.smar.com/brasil/devicenet>. [Acedido em 2014].
- [37] WEG, “CANopen,” [Online]. Available: <http://ecatalog.weg.net/files/wegnet/WEG-plc300-comunicacao-canopen-10000849433-manual-portugues-br.pdf>. [Acedido em 2014].
- [38] “Tecnologias Wireless Usadas,” [Online]. Available: <http://www.jvl.dk/938/bluetooth-dominate-wlan>. [Acedido em 2014].
- [39] I. M. J. K. R. B. Milan Šimek, “Bandwidth Efficiency of Wireless Networks of WPAN, WLAN, WMAN and WWAN,” [Online]. Available: <http://www.elektrorevue.cz/en/articles/analogue-technics/0/bandwidth-efficiency-of-wireless-networks-of-wpan--wlan--wman-and-wwan-1/>. [Acedido em 2014].
- [40] D. G. S. Alexandre Baratella Lugli, “TECNOLOGIAS WIRELESS PARA AUTOMAÇÃO INDUSTRIAL: WIRELESS_HART, BLUETOOTH, WISA, WI-FI, ZIGBEE E SP-100.,” [Online]. Available: http://www.inatel.br/biblioteca/index.php/modelos-de-trabalhos/doc_download/6088-tecnologias-wireless-para-automacao-industrial-wireless-hart-bluetooth-wisa-wi-fi-zigbee-e. [Acedido em 2014].

- [41] “Wi-Fi,” [Online]. Available: http://www.connectblue.com/fileadmin/Connectblue/Web2006/Documents/In_the_news/IEB-July2012.pdf. [Acedido em 2014].
- [42] D. P. Dickinson, “So Many Wireless Technologies ... Which Is the Right One for My Application?,” [Online]. Available: http://www.digikey.com/Web%20Export/Supplier%20Content/PhoenixContact_277/pdf/phoenix-contact-wireless-technologies.pdf?redirected=1. [Acedido em 2014].
- [43] “Tecnologia Bluetooth: o que é e como funciona?,” [Online]. Available: <http://www.infowester.com/bluetooth.php>. [Acedido em 2014].
- [44] “Bluetooth Devices,” [Online]. Available: <http://www.macos.utah.edu/documentation/hardware/bluetooth.html>. [Acedido em 2014].
- [45] “Zigbee information,” [Online]. Available: <http://www.zigbees.com/>. [Acedido em 2014].
- [46] “ZigBee,” [Online]. Available: <http://www.rogercom.com/ZigBee/ZigBee.htm>. [Acedido em 2014].
- [47] “ZigBee,” [Online]. Available: <http://lurnq.com/lesson/Zigbee/>. [Acedido em 2014].
- [48] T. Lennvall, S. Svensson e F. Hekland, “A Comparison of WirelessHART and ZigBee for Industrial Applications,” *ABB Corporate Research*, 2008.
- [49] “WirelessHART,” [Online]. Available: <http://blog.pepperl-fuchs.us/blog/bid/350677/4-Key-Elements-of-WirelessHART>. [Acedido em 2014].
- [50] “WirlessHart,” [Online]. Available: <http://www.automacaoindustrial.info/o-protocolo-wirelesshart-parte-2/>. [Acedido em 2014].
- [51] S. Yamamoto, N. Maeda e M. Takeuchi, “World's First Wireless Field Instruments Based on ISA100.11a,” *IA Foundation Technology Center*, 2010.
- [52] “ISA,” [Online]. Available: <http://www.mecatronicaatual.com.br/educacao/1646-wireless-isa-100>. [Acedido em 2014].

- [53] G. Svheible, D. Dzung, J. Endresen e J. E. Frey, “Unplugged but Connected - Design and Implementation of a Truly Wireless Real-time Sensor/Actuator Interface,” *IEEE Industrial Electronics Magazine*, 2007.
- [54] R. Steigmann e J. Endresen, “Introduction to WISA,” *ABB*, 2006.
- [55] “Wisa,” [Online]. Available: http://www.millennialnet.com/MillennialNet/media/Resources_Media/WhitePapers/WhitePaper_IntroductiontoWISA_V2.pdf. [Acedido em 2014].
- [56] “Arduino Uno,” [Online]. Available: <http://arduino.cc/en/Main/arduinoBoardUno>. [Acedido em 2014].
- [57] “Raspberry Pi Modelo A e B,” [Online]. Available: <http://downloads.element14.com/raspberryPi1.html>. [Acedido em 2014].
- [58] “Raspberry Pi Modelo B+,” [Online]. Available: <http://www.raspberrypi.org/products/model-b-plus/>. [Acedido em 2014].
- [59] “Raspberry Pi,” [Online]. Available: <http://www.raspberrypi.org/downloads/>.
- [60] “SO Raspbian Interface,” [Online]. Available: <http://www.pcguaia.pt/2013/09/tudo-sobre-o-raspberry-pi-parte-1/>. [Acedido em 2014].
- [61] “STMicroelectronics,” [Online]. Available: <http://www.st.com/web/en/cn/catalog/mmc/FM141>. [Acedido em 2014].
- [62] STMicroelectronics, [Online]. Available: <http://www.st.com/web/en/catalog/mmc/FM141/SC1169>. [Acedido em 2014].
- [63] STMicroelectronics, [Online]. Available: http://www.st.com/web/en/resource/sales_and_marketing/promotional_material/flyer/flstm32.pdf. [Acedido em 2014].
- [64] “BeagleBoard,” [Online]. Available: <http://beagleboard.org/boards>. [Acedido em 2014].
- [65] “pcDuino,” [Online]. Available: http://www.linksprite.com/?page_id=806. [Acedido em 2014].

- [66] “STMicroelectronics,” [Online]. Available: http://www.st.com/web/en/resource/sales_and_marketing/promotional_material/brochure/brstm32f4.pdf. [Acedido em 2014].
- [67] “HC-05 Datasheet,” HC-05, [Online]. Available: http://www.rcscomponents.kiev.ua/datasheets/hc_hc-05-user-instructions-bluetooth.pdf. [Acedido em 2014].
- [68] “nRF24L01+,” [Online]. Available: <https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P>. [Acedido em 2014].
- [69] “STM32F4DISCOVERY,” [Online]. Available: <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419>. [Acedido em 2014].
- [70] “STM32F4DIS-BB,” [Online]. Available: <http://uk.farnell.com/stmicroelectronics/stm32f4dis-bb/expansion-stm32f4-cortex-m4-discovery/dp/2250205>. [Acedido em 2014].
- [71] MIT App Inventor 2, [Online]. Available: <http://appinventor.mit.edu/explore/about-us.html>. [Acedido em 2014].
- [72] MySQL, “Clientes,” [Online]. Available: <http://www.mysql.com/customers/>. [Acedido em 2014].
- [73] Calculadora RFM12b, [Online]. Available: <http://tools.jeelabs.org/rfm12b.html>. [Acedido em 2014].
- [74] “Jeelabs,” [Online]. Available: <http://jeelabs.org/2009/02/10/rfm12b-library-for-arduino>. [Acedido em 2014].
- [75] “Netcraft,” [Online]. Available: <http://news.netcraft.com/archives/2014/10/10/october-2014-web-server-survey.html>. [Acedido em 2014].
- [76] “PHP,” [Online]. Available: <http://php.net/>. [Acedido em 2014].
- [77] “UNdata,” [Online]. Available: http://data.un.org/Data.aspx?q=internet&d=WDI&f=Indicator_Code%3aIT.NET.USE.R.P2. [Acedido em 2014].

-
- [78] Softstarters, [Online]. Available: <http://www.softstarter.org/how-does-soft-start-work-959233.html>. [Acedido em 2014].
- [79] “CAN,” [Online]. Available: http://www.isobus.org.br/index.php?option=com_content&view=article&id=48&Itemid=27&lang=. [Acedido em 2014].
- [80] M. d. N. Rodrigues, “PROFINET - SIEMENS,” [Online]. Available: <http://pt.slideshare.net/mnrodrigues/profinet-siemens>. [Acedido em 2014].