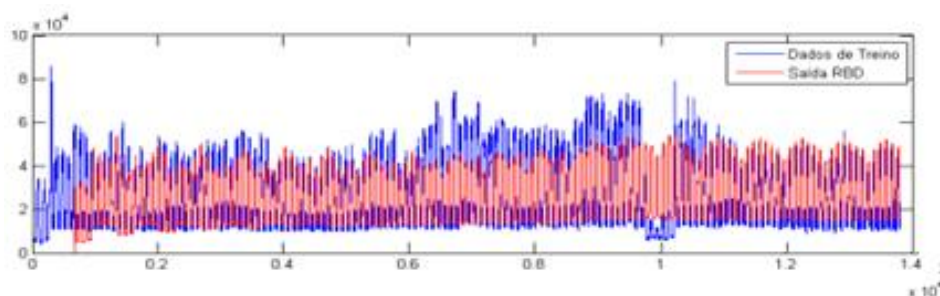


UNIVERSIDADE DO ALGARVE
INSTITUTO SUPERIOR DE ENGENHARIA



**SISTEMA DE MONITORIZAÇÃO E PREVISÃO
INTELIGENTE DE CONSUMOS ELÉTRICOS**

Hugo Miguel Conceição Costa

Dissertação para a obtenção do grau de Mestre em
Engenharia Elétrica e Eletrónica
(Ramo de Sistemas de Energia e Controlo)

Trabalho efetuado sob a orientação de:
Professor Doutor Jânio Miguel Evangelista Ferreira Monteiro
Professor Doutor Jorge Filipe Leal Costa Semião

Fevereiro de 2016

Declaração de Autoria de Trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

Hugo Miguel Conceição Costa

Copyright © 2016 por Hugo Miguel Conceição Costa

A Universidade do Algarve tem o direito, perpétuo e sem limites geográficos, de arquivar e publicitar este trabalho através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, de o divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Gostaria de agradecer ao Professor Doutor Jânio Miguel Evangelista Ferreira Monteiro, ao Professor Doutor Jorge Filipe Leal Costa Semião e aos técnicos António Costa e Nuno Portela pela disponibilidade para me orientarem e a paciência com que o fizeram e me ajudaram a ultrapassar as dificuldades que foram surgindo. Gostaria, ainda, de sublinhar os conhecimentos que me transmitiram, essenciais para continuar a minha formação e indispensáveis para o desenvolvimento deste trabalho.

Gostaria também de agradecer à minha família por todos os conselhos e apoio que me deram.

Resumo

Na atualidade, a eficiência energética é um assunto que tem cada vez mais importância, tanto na indústria como em instalações privadas. Investigadores e projetistas, aquando do desenvolvimento de novos projetos, tentam, sempre que possível, incluir a eficiência energética nos mesmos. Com a importância crescente deste tema, criam-se também projetos para incluir em instalações já existentes. Presente em algumas soluções de promoção da eficiência energética, encontram-se as redes de sensores. Estas redes são constituídas por elementos sensores (tensão, temperatura, luminosidade, etc), que comunicam com uma unidade central de processamento, por norma através de um protocolo sem fios, como *Wi-Fi* ou *ZigBee*. Com estas redes podemos obter informações sobre os elementos intervenientes na eficiência energética, como as potências elétricas e a temperatura, e enviar esses dados para a *Internet*. No entanto, só a monitorização dos intervenientes nem sempre é suficiente, na medida em que as anomalias nos consumos elétricos contribuem para o desperdício energético. Apesar de existirem inúmeros projetos que envolvem a monitorização de processos, nenhum deles envolve a deteção automática de anomalias. Assim, neste trabalho propõe-se um sistema que faça a monitorização dos consumos elétricos, ao mesmo tempo que verifica, de forma automática, se esses consumos são anómalos ou não.

Palavras-chave: Algoritmos de Previsão de Consumos, Aprendizagem Automática, Redes *Bayesianas*, Medição de Consumos.

Abstract

Nowadays, energy efficiency is an issue which has an increasing importance, both in industry as in private installations. Researchers and designers try, as much as possible, to include energy efficiency on new projects, but, with the growing importance of this theme, they also try to develop projects to include it in already existing installations. In some solutions that promote energy efficiency we find sensor networks. These networks are formed by sensor elements (voltage, temperature, light, etc.), which communicate with a central processing unit through a wireless protocol, such as Wi-Fi or ZigBee. With these networks, we can gather information about the elements present in energy efficiency gear, such as electrical power and temperature, and send that data to the Internet. However, the sole observation of those elements is not always enough, because anomalies on electrical consumptions contribute to the increase of energy waste. Although there are many projects involving the monitoring of processes, none of them involves the automatic detection of electrical anomalies. Therefore, in this work we propose a system that monitors electrical consumptions while, at the same time, checks, in an automatic way, if those consumptions are anomalies or not.

Keywords: Consumption Forecast, Machine Learning, Bayesian Networks, Power Consumption Measurements.

Índice

Capítulo 1	Introdução	1
1.1	<i>O Problema</i>	1
1.2	<i>Objetivos da Dissertação</i>	2
1.3	<i>Estrutura da Dissertação</i>	2
Capítulo 2	Estado da Arte	5
2.1	<i>Projetos para monitorização de variáveis e com controlo remoto</i>	5
2.1.1	Monitorização de processos	5
2.1.2	Monitorização e atuação	10
2.1.3	Monitorização e atuação com inteligência	12
2.2	<i>Protocolos de comunicação em redes de sensores</i>	12
2.2.1	<i>ZigBee</i>	13
2.2.2	<i>6LoWPAN</i>	15
2.3	<i>Redes Bayesianas</i>	16
2.3.1	Obtenção de Probabilidades	18
2.3.2	Cadeias de <i>Markov</i>	20
2.3.3	Redes <i>Bayesianas</i> Dinâmicas	21
2.3.4	Algoritmo de <i>Viterbi</i>	28
2.4	<i>Conclusão preliminar</i>	30
Capítulo 3	Módulo de Medição e Monitorização Remota	33
3.1	<i>Obtenção dos Valores de Tensão e Corrente no Módulo de Medição de Consumos</i>	34
3.2	<i>Medição de Consumos em Quadros Elétricos Trifásicos</i>	37
3.2.1	Circuito medição de tensão	37
3.2.2	Circuito de medição de corrente	39
3.2.3	Obtenção dos Valores das Potências e do Fator de Potência no Módulo de Medição dos Consumos	41
3.3	<i>Módulo de Controlo de Cargas Elétricas Trifásicas</i>	44
3.4	<i>Módulo de Aquisição de Dados Ambientais</i>	45
3.5	<i>Apresentação das Caraterísticas do Sistema Trifásico</i>	49
3.6	<i>Proposta de um novo protótipo da tomada elétrica trifásica</i>	52
3.7	<i>Módulo demonstrador</i>	54
Capítulo 4	Mecanismo de Detecção de Consumos Anómalos	57
4.1	<i>Implementação</i>	60
4.2	<i>Avaliação de Resultados</i>	60
Capítulo 5	Conclusões	63
5.1	<i>Disseminação</i>	64
Bibliografia		65
Anexos		67

<i>Anexo A. Fluxograma do Algoritmo de Viterbi</i>	67
<i>Anexo B. O modelo OSI</i>	68
<i>Anexo C. Componentes utilizados</i>	69
<i>a. Raspberry Pi modelo B+</i>	69
<i>b. Conversor AC-DC VIGORTRONIX – VTX-214-010-105</i>	70
<i>c. Sensor de corrente SCT-013-000</i>	71
<i>d. Conversor Analógico-Digital (ADC) MCP3008</i>	73
<i>e. Zolertia Z1</i>	74
<i>f. Relé Omron G4A</i>	75
<i>g. Sensor de temperatura LM35DZ</i>	75
<i>h. Painel solar MC-SP0.8-NF-GCS</i>	76
<i>i. Conversor DC-DC TSR 0.5-2433</i>	76
<i>j. XBee série 1</i>	76
<i>k. AC-DC RAC01-12SC</i>	77
<i>Anexo D. Biblioteca do Conversor AC-DC RAC01-12SC da RECOM POWER</i>	77
<i>Anexo E. Código Para Obter as Tensões e as Correntes Trifásicas</i>	77
<i>Anexo F. Código Para Calcular as Potências e o Fator de Potência Trifásico</i>	78
<i>Anexo G. Código Para a Aquisição Dos Dados Ambientais</i>	85
<i>Anexo H. Código Para Previsão Dos Consumos Elétricos</i>	86

Lista de Figuras

Figura 2.1– Diagrama de blocos do projeto “Sistema Automático de Medida Usando um Sistema GSM”. Fonte: [1].	6
Figura 2.2 – Topologia em estrela. Fonte: [13].	14
Figura 2.3 – Topologia em malha. Fonte: [13].	14
Figura 2.4 – Topologia em árvore. Fonte: [13].	14
Figura 2.5 – Rede <i>Bayesiana</i> do exemplo da dor de costas. Editado de [15].	16
Figura 2.6 – Tabelas de probabilidades da rede <i>Bayesiana</i> da dor de costas. Editado de [15].	17
Figura 2.7 – Cadeia de Markov exemplificando um mercado bolsista. Fonte: [18].	20
Figura 2.8 – Ambiente para testes. Editado de [19].	22
Figura 2.9 – Rede <i>Bayesiana</i> de “ <i>Realtime Recognition of Complex Daily Activities Using Dynamic Bayesian Network</i> ”. Fonte: [19].	23
Figura 2.10 – Rede <i>Bayesiana</i> Dinâmica de “ <i>Using Bayesian Networks for Daily Activity Prediction</i> ”. Fonte: [20].	24
Figura 2.11 – Gráfico com as previsões dos intervalos de tempo entre o início da atividade atual e o início da atividade seguinte. Fonte: [20].	28
Figura 2.12 – Gráfico dos intervalos de tempo entre o início do pequeno-almoço e a toma da medicação. Fonte: [20].	28
Figura 2.13 – Estrutura da rede para avaliação do estado de saúde do paciente. Fonte: [21].	29
Figura 2.14 – Proposta de uma rede <i>Bayesiana</i> .	31
Figura 3.1 – Arquitetura do sistema.	33
Figura 3.2 – Interligação <i>SPI</i> . Fonte:[22].	34
Figura 3.3 – Ligação <i>SPI</i> entre o <i>Raspberry</i> e o <i>ADC</i> .	35
Figura 3.4 – Comunicação do <i>ADC</i> .	35
Figura 3.5 – Circuito de medição da tensão. Fonte: [22].	37
Figura 3.6 – Formas de onda das tensões das 3 fases obtidas no <i>Raspberry Pi</i> .	39
Figura 3.7 – Circuito da medição de corrente. Fonte: [22].	40
Figura 3.8 – Formas de onda das correntes das 3 fases calculadas pelo <i>Raspberry Pi</i> .	41
Figura 3.9 – Placa que mede os consumos elétricos em quadros trifásicos.	41
Figura 3.10 – Desenho da placa que mede os consumos elétricos em quadros trifásicos.	42
Figura 3.11 – Circuito de controlo do relé. Fonte: [22].	44
Figura 3.12 – Placa que controla os consumos elétricos em tomadas trifásicas.	45
Figura 3.13 – Desenho da placa que controla os consumos elétricos em tomadas trifásicas.	45
Figura 3.14 – Circuito de aquisição dos dados ambientais.	46

Figura 3.15 – <i>PCB</i> da placa de aquisição de dados ambientais.	46
Figura 3.16 – <i>PCB</i> final do módulo de aquisição de dados.	47
Figura 3.17 – Módulo de aquisição dos dados ambientais.	47
Figura 3.18 – Diagrama de blocos do processo de medição e cálculo das características do sistema trifásico.....	49
Figura 3.19 – <i>Dashboard</i> recetor dos valores calculados.	50
Figura 3.20 – <i>Template</i> da página <i>web</i>	51
Figura 3.21 – Proposta alternativa de uma tomada elétrica trifásica.	53
Figura 3.22 – Módulo de Medição dos Consumos Elétricos Trifásicos.	54
Figura 3.23 – Módulo de Controlo de Cargas Elétricas Trifásicas.	55
Figura 4.1 – Esquema da Rede Bayesiana Dinâmica utilizada.....	58
Figura 4.2 – Comparação entre consumo real e estimado, utilizando a Rede <i>Bayesiana</i> Dinâmica, em que o gráfico superior corresponde à fase anterior a novembro de 2014 e a parte inferior à fase de posterior.....	62
Figura 6.0.1 – Fluxograma da função viterbi para implementação do algoritmo de Viterbi para a análise do estado de saúde do paciente. Fonte: [21].....	67
Figura 6.0.2 – O modelo <i>OSI</i>	68
Figura 6.0.3 – <i>Raspberry Pi</i> modelo B+.	69
Figura 6.0.4 – <i>Pinout</i> do <i>Raspberry Pi</i> modelo B+.	70
Figura 6.0.5 – Conversor AC-DC.	70
Figura 6.0.6 – SCT-013-000.	71
Figura 6.0.7 – Gráfico dos resultados obtidos.	72
Figura 6.0.8 – Ligação interna do SCT-013-000	73
Figura 6.0.9 – <i>Pinout</i> do MCP3008.	73
Figura 6.0.10 – Zolertia Z1.	74
Figura 6.0.11 – Relé G4A.	75
Figura 6.0.12 – Biblioteca do conversor AC-DC RAC01-12SC da RECOM POWER.	77

Lista de Tabelas

Tabela 2.1 – Especificações dos protocolos <i>ZigBee</i> e <i>6LoWPAN</i> . Fonte: [14].	15
Tabela 2.2 – Resultados obtidos pelo modelo adotado. Fonte: [19].	24
Tabela 2.3 – Características de teste. Fonte: [20].	25
Tabela 2.4 – Precisão obtida pelo CRAFT. Fonte: [20].	26
Tabela 2.5 – Exemplos das características da atividade seguinte a prever. Fonte: [20].	26
Tabela 2.6 – Resultados obtidos pelo CEFA. Fonte: [20].	27
Tabela 3.1 – <i>Bits</i> de configuração.	36
Tabela 3.2 – Medição das variáveis ambientais.	49
Tabela 3.3 – Medições de potências.	51
Tabela 4.1 – Resultados obtidos para a Rede <i>Bayesiana</i> Dinâmica.	61
Tabela 6.0.1 – Características do <i>Raspberry Pi</i> modelo B+.	69
Tabela 6.0.2 – Características do conversor AC-DC.	71
Tabela 6.0.3 – Resultados obtidos nas medições das correntes no transformador de corrente.	72
Tabela 6.0.4 – Características do MCP3008.	74
Tabela 6.0.5 – Características do relé Omron G4A.	75
Tabela 6.0.6 – Características do sensor de temperatura.	75
Tabela 6.0.7 – Características do painel solar.	76
Tabela 6.0.8 – Características do conversor DC-DC.	76
Tabela 6.0.9 – Características do <i>XBee</i> série 1.	76
Tabela 6.0.10 – Características do retificador RAC01-12SC.	77

Lista de Acrónimos

ADC	<i>Analog-to-Digital Converter</i>
AMT	Aeromodelo Telecomandado
AVAC	Aquecimento, Ventilação e Ar Condicionado
CAN	<i>Controller Area Network</i>
CEFA	<i>Current features and activity to predict the next activity</i>
CRAFFT	<i>Current activity and features to predict the next features</i>
DCP	Distribuição Conjunta de Probabilidade
EM	<i>Expectation-Maximization</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Global Communications</i>
HMI	<i>Human-Machine Interface</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IGBT	<i>Insulated-Gate Bipolar Transistor</i>
IP	<i>Internet Protocol</i>
LED	<i>Light-Emitting Diode</i>
LSB	<i>Least Significant Bit</i>
SMD	<i>Surface Mount Device</i>
MCU	<i>Microcontroller Unit</i>
MSB	<i>Most Significant Bit</i>
OSI	<i>Open Systems Interconnection</i>
PAN	<i>Personal Area Network</i>
PCB	<i>Printed Circuit Board</i>
PWM	<i>Pulse Width Modulation</i>
SPI	<i>Serial Peripheral Interface</i>

USART	<i>Universal Synchronous – Asynchronous Receiver – Transmitter</i>
UAlg	Universidade do Algarve
WWAN	<i>Wide Wireless Area Network</i>
WLAN	<i>Wireless Local Area Network</i>

Capítulo 1 **Introdução**

Atualmente, dá-se cada vez mais importância à eficiência energética, fazendo com que projetistas e investigadores de todo o mundo tentem criar novas soluções para reduzir os desperdícios e, conseqüentemente, os custos, otimizando assim a utilização dos recursos disponíveis. Um exemplo de uma solução já existente é a casa “inteligente” [1][31], em que se pode controlar desde a iluminação e a temperatura, até à quantidade de comida existente no frigorífico, utilizando sensores, protocolos de comunicação, unidades de processamento centralizadas e atuadores. A solução mencionada já representa um avanço na redução dos desperdícios energéticos, na medida em que, por exemplo, quando é de dia, a casa abre as persianas e, quando é de noite, fecha-as. Com este controlo, evita-se ligar as luzes se for de dia. Este é um exemplo de monitorização e atuação.

Contudo, os projetos que permitem a monitorização não têm deteção automática de anomalias, implicando a intervenção humana que tem que cruzar uma grande quantidade de informação. Neste trabalho abordamos a eficiência energética através da monitorização e deteção de consumos elétricos anómalos. O objetivo é criar um sistema “inteligente” que “aprende” os consumos elétricos e gera alertas se, num determinado instante, esses consumos são muito diferentes dos previstos.

1.1 O Problema

A melhoria da eficiência energética de uma instalação pode passar pela introdução de medidas passivas, recorrendo por exemplo a materiais com melhor isolamento, ou mesmo à substituição de equipamentos por outros com melhor eficiência. Pode ainda basear-se na introdução de medidas ativas, que permitem monitorizar, detetar e controlar equipamentos, por forma a reduzir o seu consumo ou o custo associado.

No âmbito das medidas ativas encontramos hoje várias soluções que permitem já efetuar a monitorização de diversos pontos de consumo, gerando alarmes sempre que esses consumos ultrapassem limites superiores ou inferiores. No entanto os consumos ditos

normais de uma instalação dependem de uma série de fatores, tais como o dia da semana, a hora do dia, a temperatura exterior, a radiação solar, a ocupação do edifício, entre outros. Não é de admirar, por isso, que a deteção de consumos anómalos continue a ser realizada com base na intervenção de um profissional, ou de uma pessoa que regularmente analise os dados. Ainda assim, a análise dos dados é complexa, uma vez que é necessária uma extrema capacidade de correlação das variáveis, para que o utilizador consiga identificar se o consumo que analisou era anómalo ou normal. Se se quiser que haja a deteção automática de consumos elétricos anómalos, é necessário o desenvolvimento de um algoritmo para esse efeito que pode ser bastante complexo, sendo que a maior parte dos algoritmos de deteção existentes baseiam-se em regressões (lineares ou quadráticas) que só têm em consideração os valores de saída obtidos anteriormente, desprezando as variáveis de entrada.

1.2 Objetivos da Dissertação

A proposta para a resolução deste tipo de problemas, consiste na criação de um sistema “inteligente” que detete anomalias nos consumos elétricos. Pretende-se que o sistema “aprenda” o que se passa e que gere alertas sempre que, num determinado instante, o consumo de energia elétrica seja bastante diferente do previsto. A aprendizagem tem como objetivo a adaptação do sistema às novas condições do meio exterior. Assim, quando houver um consumo elétrico anómalo, o sistema deteta-o de forma automática, isto é, sem que haja um ser humano a correlacionar os dados e a chegar à conclusão de que há essa anomalia.

A arquitetura do *hardware* proposto suporta-se num pequeno computador que efetua o processamento dos dados, complementado com sensores de temperatura e radiação e sistemas de comunicação sem fios, para transmissão dos dados recolhidos pelos sensores.

1.3 Estrutura da Dissertação

Este documento está organizado como a seguir se explica. No capítulo 2 será feita a análise do estado da arte, que compreende um levantamento dos projetos relacionados com a deteção de anomalias e monitorização de parâmetros, que estejam a ser, ou que já estejam, desenvolvidos, incluindo materiais e procedimentos utilizados, bem como alguma teoria

mais relevante. No capítulo 3, será feita a apresentação e a descrição dos componentes, procedimentos e algoritmos para a medição e monitorização de parâmetros. O capítulo 4 será dedicado em exclusivo à previsão dos consumos energéticos, apresentando os algoritmos desenvolvidos. No capítulo 5, resumem-se as conclusões e apontam-se as perspetivas de trabalho futuro. Este documento termina com a bibliografia, onde se encontram as referências, e com os anexos, onde se encontra uma explicação mais detalhada dos componentes utilizados, os códigos fonte gerados e outras informações relevantes.

Capítulo 2 Estado da Arte

Atualmente existem inúmeros e variados sistemas de monitorização, sendo que alguns já contêm atuação.

Este capítulo mostrará alguns projetos de monitorização e de controlo de processos, desde a simples monitorização de processos até aos complexos sistemas de monitorização e atuação nos mais diversos ambientes.

2.1 Projetos para monitorização de variáveis e com controlo remoto

As redes de sensores estão assentes em dois pilares: por um lado, a medição de variáveis como, por exemplo, a temperatura, a pressão ou a radiação; por outro lado, a transmissão desses valores, utilizando por exemplo redes *ZigBee*, *Wi-Fi*, ou a rede *GSM* (*Global Systems for Mobile Communications*).

Nos subcapítulos que se seguem, apresentam-se alguns exemplos de aplicações que contêm monitorização e controlo remoto. No final de cada exemplo, é feita uma comparação com a solução proposta nesta dissertação.

2.1.1 Monitorização de processos

A monitorização remota de variáveis tem sido utilizada em vários projetos de redes de sensores. Nesta secção analisaremos alguns desses projetos, classificando-os com base na tecnologia utilizada para a transmissão dos dados.

A utilização de módulos de comunicação GSM é comum a vários projetos. Refira-se como exemplo o trabalho realizado em 2001 por Paulo Santos, da Universidade do Algarve (UAIG), que num artigo denominado de “Sistema Automático de Medida Usando um

Sistema GSM” [1], descreve um sistema instalado numa boia que monitoriza a qualidade da água e envia os dados recolhidos via *GSM* para uma estação em terra. Nesse artigo, o autor expõe que os objetivos do projeto foram o desenho e implementação de um sistema automático de medida, com aplicação na medição da qualidade da água, tendo como características a operação em tempo real, ser constituído por uma ou mais estações de medida e uma estação de processamento, apresentar transdutores de arquitetura aberta à inclusão de componentes de vários fabricantes e ter acesso às coordenadas de posição da estação de medida.

O diagrama de blocos que serviu de base para a implementação do projeto é o que se encontra na Figura 2.1. Os dados recolhidos pelos sensores são convertidos de valores analógicos para digitais, para que o microcontrolador possa processá-los, e depois estes dados são enviados por *GSM* através de *modems* para o computador em terra, onde são armazenados e novamente processados. O autor refere ainda que foram utilizados multiplexadores (circuitos integrados que enviam para a saída um ou mais dos valores presentes na entrada, dependendo do sinal de comando que é enviado para os mesmos e do número de saídas disponível), os quais permitem a redução de custos. Por exemplo, se tivermos 10 sensores, com os respetivos transdutores, e se utilizarmos um multiplexador de 10 para 2, não precisaremos de adquirir um microcontrolador com 10 canais analógicos, mas sim, 1 de apenas 2 canais analógicos, tornando o produto final mais barato.

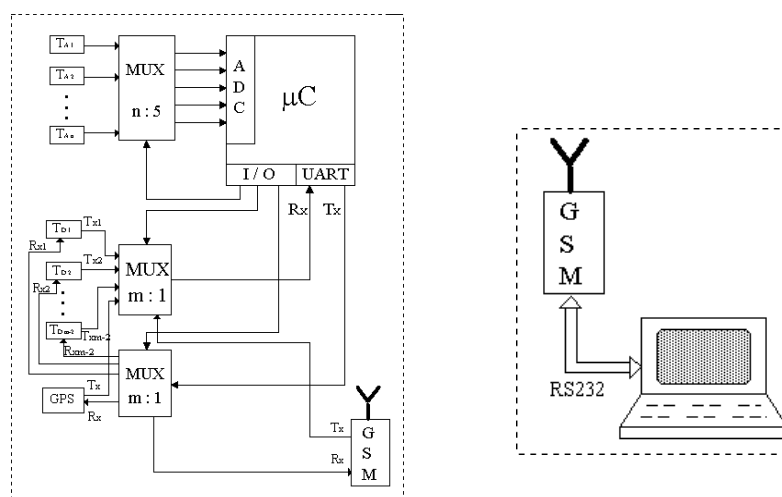


Figura 2.1– Diagrama de blocos do projeto “Sistema Automático de Medida Usando um Sistema GSM”. Fonte: [1].

Este projeto usou um sistema *GSM*, o que constitui uma mais-valia, uma vez que utiliza as antenas de telecomunicações das operadoras para enviar os dados da boia para a

estação em terra, sendo um sistema de comunicação interessante para o envio de dados a grandes distâncias. Contudo, a solução *GSM* não é adequada para o nosso projeto, porque os transmissores *GSM* consomem muita energia, contrariando as medidas de eficiência energética que pretendemos implementar.

Em outros casos optou-se pela utilização de comunicações por ZigBee. Por exemplo, o projeto desenvolvido por Francisco J. Espinosa-Faller e Guillermo E. Rendón-Rodríguez, da *Universidad Marista de Mérida*, no México, intitulado “*A ZigBee Wireless Sensor Network for Monitoring an Aquaculture Recirculating System*” (2012) [2], utiliza o protocolo *ZigBee* na comunicação entre os sensores e o computador. Este projeto monitoriza um sistema experimental de recirculação aplicado em aquacultura e é constituído por sensores de temperatura, de oxigénio dissolvido, de água e de pressão do ar, os quais estão ligados a módulos de comunicações, dispostos em malha, que comunicam utilizando o protocolo *ZigBee*. Os dados fornecidos pelos sensores são mostrados e armazenados num programa, que gera alertas via *SMS* e via *e-mail* quando os valores medidos ultrapassam valores limite definidos. A página *web* mostra os valores mais recentes da base de dados e fica em permanente atualização. A página foi programada utilizando a linguagem HTML, PHP e AJAX, e a base de dados utilizada foi a MySQL [2].

Este projeto é bastante semelhante ao que foi desenvolvido nesta dissertação, na medida em que ambos se baseiam a recolha e análise de dados vindos de sensores, para posteriormente decidir sobre geração de alertas para o utilizador. Contudo, utilizam materiais e métodos diferentes, nomeadamente na unidade de processamento, linguagens e *output* da página *Web* e módulos *ZigBee*, sendo que os alertas gerados resultam da ultrapassagem de níveis mínimos ou máximos. Na solução desenvolvida nesta dissertação, os alertas gerados resultam do cruzamento dos dados obtidos com o histórico dos resultados.

Em 2013, uma empresa denominada *Dedicated Engines Lda*, sediada em *Letchworth Garden City, Hertfordshire*, Reino Unido, começou a utilizar um projeto chamado “*Temperature Monitoring & Control with ZigBee*” [3], que utiliza uma rede fechada de sensores de temperatura, os quais comunicam para a unidade central de processamento utilizando o protocolo *ZigBee* para controlar a temperatura em locais de trabalho e em armazéns. A empresa queria criar uma rede de sensores *wireless* de baixo consumo e chegou à conclusão que o protocolo *ZigBee* seria a única hipótese de conjugar robustez com baixo

consumo. Os dispositivos de monitorização da temperatura emparelham automaticamente com a estrutura principal *ZigBee*, graças a um coordenador colocado em cada saída dessa mesma estrutura, que liga diretamente os novos dispositivos à unidade base local da *Dedicated Engines*. Os valores da temperatura são assim enviados, juntamente com a informação já fornecida pelo sistema de monitorização já existente.

Apesar de mais simples que os sistemas anteriormente referidos, dado que mede apenas valores de temperatura, este projeto tem algumas características semelhantes às que pretendemos utilizar no trabalho apresentado nesta dissertação, uma vez que assume a preocupação com o baixo consumo energético.

Outro projeto que utiliza o protocolo *ZigBee* e que está relacionado com a medição de temperaturas é o “*Zigbee Enabled Remote Temperature Monitor System for High-Voltage Substations*” [4], um projeto apresentado numa conferência em *Xian, Shaanxi*, China, em finais de Maio de 2012. Uma vez que nas subestações de alta tensão, contactos fracos podem gerar excesso de calor que derivam em incêndios, surgiu a necessidade de desenvolver este projeto que utiliza sensores de temperatura para gerar alarmes, quando esta sobe acima de um determinado valor. Os dados recolhidos são enviados para um computador central através do protocolo *ZigBee*.

Mais uma vez, encontramos como semelhanças entre este projeto e o que estamos a desenvolver, o protocolo e a função do sensor utilizados. No entanto, este projeto não proporciona o acesso à *Internet* para visualização dos dados recolhidos, o que, no nosso caso, é fundamental.

Numa tese de mestrado desenvolvida por Tomé Emanuel Rodrigues Matos, na Faculdade de Ciências e Tecnologia da Universidade de Coimbra em 2008 e intitulada de “*Remote Vital Signs Monitoring: Instrumentation Module*” [5], é apresentado um sistema de cuidados de saúde portátil que permite “uma contínua e segura monitorização dos pacientes, evitando a alteração da sua rotina diária”, como é descrito pelo próprio. O sistema é constituído por um equipamento médico, com sensores, e um módulo de transmissão. Os testes de diagnósticos que foram integrados no equipamento médico foram: um eletrocardiógrafo para monitorizar possíveis doenças cardiovasculares; um sensor de oximetria do pulso que, indiretamente, mede a quantidade de oxigénio que a hemoglobina transporta; e um acelerómetro triaxial para saber se o paciente caiu. O autor diz que “foram

usados dois tipos de *sockets*: um que suporta tecnologia *Bluetooth* e outro que suporta tecnologia *ZigBee*”. Isto permite amplificar a escolha das soluções de monitorização. Depois, este módulo poderá enviar os dados para profissionais de saúde através de comunicações *wireless* de longo alcance, tais como *Wi-Fi*, *GSM/GPRS (General Packet Radio Service)* e *Ethernet*, alterando a tecnologia de comunicação através de um sistema *plug – and – play*.

Este trabalho é semelhante ao nosso, na medida em que recolhe dados utilizando sensores e envia-os via rede sem fios. As principais diferenças são os *outputs* dos diferentes sensores, o sistema de transmissão sem fios poder ser alterado consoante as necessidades, e o fato de, mais uma vez, o projeto não enviar os dados para a *Internet*.

Em 2009 foi desenvolvida por Ivo Daniel G. Moreno, na Faculdade de Ciências e Tecnologia da Universidade de Coimbra, uma tese de mestrado em engenharia física intitulada “Rede de Sensores Inteligentes Sem Fios Para a Supervisão de Parques de Reservatórios Industriais” [6]. O objetivo da tese foi, segundo o autor, “criar um sistema de aquisição de dados distribuído, embebido e não cablado, otimizado para a monitorização remota de parques de reservatórios de matérias potencialmente explosivas, como são os das refinarias petroquímicas”. Os dados recolhidos por cada unidade podem ser enviados para uma base de dados definida pelo utilizador, no complexo industrial ou em qualquer parte do mundo. A comunicação entre as unidades e a central é feita utilizando tecnologia *wireless*. O autor afirma que “a comunicação entre os vários módulos que formam a unidade de aquisição de dados, uma vez que estes estão em posições adjacentes ou distanciados por escassos metros, é feita através de uma ligação ponto-a-ponto cablada, usando o protocolo *CAN*”. Dado que a tecnologia de transmissão de dados para o exterior utilizada foi *wireless*, o protocolo de comunicações escolhido foi o IEEE 802.15.4, uma vez que o baixo consumo característico dos *transceivers* IEEE 802.15.4 é na ordem dos *miliWatt*. A norma IEEE 802.15.4 opera na banda dos 2,4 GHz. No entanto, as bandas utilizadas são entre os 868 e os 868,6 MHz, de 1 canal na Europa, e entre os 902 e os 928 MHz, de até 30 canais na América do Norte. A velocidade de transmissão máxima é de 250 kbit/s, se a distância de comunicação máxima for de 10 metros. O autor salienta que “uma vez que uma comunicação baseada no protocolo IEEE 802.15.4 nunca atinge potências superiores a algumas dezenas de *miliWatts* (tipicamente, entre 20 a 30 mW), e de forma a melhorar o alcance rádio dos módulos, estes usam uma antena em chicote (*whip antenna*) colocada no seu exterior.”

Tal como no trabalho da presente dissertação, este projeto usa a norma IEEE 802.15.4, que será detalhada mais à frente, mas o projeto descrito utiliza o protocolo *CAN*, do tipo cablado e desenhado para transmissão dos dados em tempo real, o que não é muito importante no nosso caso. O uso da antena em chicote é uma opção viável porque aumenta a potência do sinal, permitindo um aumento da distância a que este pode chegar.

Apesar destes projetos já envolverem monitorização, esta por si só não chega, na medida em que se for necessário atuar sobre qualquer processo, não é possível fazê-lo remotamente.

2.1.2 Monitorização e atuação

Neste subcapítulo iremos analisar projetos que envolvem monitorização e atuação sobre processos, realçando o método de atuação. Estes projetos podem ser classificados ao nível da atuação em sistemas sem fios [7][8] ou misto [9].

Em 2013 na Anna University, Chennai, Índia, foi desenvolvido um projeto que utiliza sensores de luminosidade, o “*Intelligent Street Lighting Using a ZigBee Network of Devices and Sensors*” [7]. Este projeto consistiu em utilizar um conjunto de sensores na iluminação pública que detetam a presença de pessoas, verificam a intensidade luminosa num determinado instante e ajustam a luz emitida em conformidade, promovendo assim a eficiência energética. As informações recolhidas pelos sensores (*Light-Dependent Resistor* ou *LDR*) são enviadas via *ZigBee*, ponto a ponto, até chegarem ao terminal de controlo. Dependendo dos dados recebidos, o sistema de controlo dirá ao sistema de iluminação para ligar ou desligar as luzes através da comunicação *wireless*.

Segundo os autores, a maior parte dos sistemas de comunicação em sistemas de iluminação pública ainda é *USART (Universal Synchronous – Asynchronous Receiver – Transmitter)* e Porto Escravo Paralelo, significando assim, que este projeto constitui um avanço, na medida em que utiliza o protocolo *ZigBee*. Além deste ponto, em comum com o projeto apresentado nesta dissertação, utiliza-se um sensor de luz.

Na Faculdade de Ciências e Tecnologia da Universidade do Algarve foi escrita, em 2012, uma tese denominada por “*Aeromodelo Telecomandado Controlado Remotamente Através da Internet Para Caracterização do Ambiente Através de Imagens e Vídeo*” [8], cujo autor foi João Pedro Rosado Landeiro. Como o próprio afirma, “o objetivo desta dissertação é controlar um aeromodelo (avião) telecomandado (AMT - Aeromodelo Telecomandado) por rádio frequência através da *Internet*.” Este aeromodelo está ligado à *Internet* através de 1 das 3 ligações implementadas: *Wireless Local Area Network* (WLAN); *Wide Wireless Area Network* (WWAN) e o controlo clássico por rádio frequência, caso as duas primeiras ligações falhem, que funciona na gama de frequências dos 2,4 GHz.

Apesar de não receber dados de sensores, este projeto tem informações interessantes, como por exemplo a forma de alterar a frequência da onda portadora da informação, promovendo a segurança. Embora o nosso projeto não seja comandar um avião ou carro, a forma como se pode controlar e proteger o equipamento é um aspeto importante.

Em Março de 2010, na Faculdade de Engenharia da Universidade do Porto (FEUP), Paulo Filipe Loureiro Ferreira escreveu uma dissertação intitulada “*Sistema de Controlo e Monitorização Remota de uma Unidade Industrial de PSA*” [9]. O objetivo do trabalho desenvolvido nesta dissertação, segundo o autor, resultou do fato da empresa cliente (*SysAdvance*) possuir sistemas de produção de oxigénio com poucos sensores e, além disso, autómatos que corriam algoritmos simples, pouco ou nada parametrizáveis, o que requeria a intervenção de um técnico especializado em programação de autómatos cada vez que se quisesse fazer uma reconfiguração. Pegando nestas limitações, o cliente pediu à FEUP um produto novo e inovador “capaz de controlar e implementar várias situações de funcionamento e reagir a ordens de emergência / stop de uma forma controlada, sem colocar em perigo as pessoas ou os bens materiais”. Deveria, ainda, recolher os dados em tempo real e registá-los periodicamente numa base de dados, para que, posteriormente, pudessem ser descarregados e processados. O autor cita que “no PC remoto foi criada uma aplicação visual e intuitiva que tem como função parametrizar o sistema, mais especificamente o nº de etapas em cada macroetapa, o tempo de duração de cada etapa, o nº de ciclos de cada macroetapa e quais as saídas digitais a ativar em cada etapa. Essa aplicação foi também usada para visualizar em tempo real as variáveis monitorizadas e observar o comportamento temporal dessas variáveis guardando um registo em memória externa do PLC”. Esta aplicação

comunica com o PC remoto via *GPRS* (no caso do utilizador remoto) e utiliza um interface *HMI* (*Human-Machine Interface*) para utilizadores que se encontrem no local.

Apesar destes projetos já permitirem a atuação sobre processos, requerem a intervenção humana para fazer essa atuação. A subsecção seguinte mostra um projeto que tem a capacidade de decidir sozinho, tanto em função do comportamento do utilizador como em função das medidas necessárias, para promover a eficiência energética.

2.1.3 Monitorização e atuação com inteligência

Em 2011, uma empresa americana sediada em Palo Alto, Califórnia chamada *Nest Labs*, [10] desenvolveu um termostato que possui a capacidade de “aprendizagem” [11]. Este termostato tem como função controlar os sistemas AVAC (Aquecimento, Ventilação e Ar Condicionado) para manter constante a temperatura num compartimento. Apesar de, à primeira vista, ser um termostato semelhante aos outros, este tem a capacidade de “aprender”. Isto é, para além de controlar os sistemas AVAC de forma a cumprir um padrão definido inicialmente, o equipamento tem ainda a capacidade de criar um calendário personalizado, onde constam todos os valores definidos pelo utilizador e um histórico dos consumos energéticos. Com este calendário, o termostato sabe, para cada dia, quais os hábitos do consumidor e ajusta-se quando há alterações. Por exemplo, se o utilizador chega a casa às 16h e quer uma temperatura de 21°C, o termostato pode ligar-se às 15h e 45min e executar um aquecimento/arrefecimento gradual de forma automática. Para além disso, o termostato pode ser acedido, tanto localmente como em qualquer lugar, através de qualquer equipamento que consiga aceder à *Internet*.

Este projeto tem algumas características semelhantes às que aqui propomos, uma vez que a medição da temperatura e a aprendizagem são comuns a ambos os projetos.

2.2 Protocolos de comunicação em redes de sensores

Sendo o *ZigBee* um dos tipos de redes de baixo consumo mais utilizado nos projetos referidos anteriormente, em seguida descrevemo-lo de forma sumária. Descrevemos também

o seu sucessor, o *6LoWPAN*, que não tendo sido mencionado nos projetos anteriores é implementado mais adiante, suportando o conceito de *Internet of Things*.

2.2.1 ZigBee

O *ZigBee* é um protocolo de comunicação sem fios muito utilizado na criação de PANs (*Personal Area Networks*). Os aparelhos que proporcionam essa comunicação são de baixo consumo, e isso permite ligar os dispositivos *ZigBee* a baterias, evitando assim a necessidade de ligá-los à rede elétrica e permitindo também a implementação de mecanismos de carga das baterias usando as energias “verdes”. Apesar de se basear na norma *standard* IEEE 802.15.4, o protocolo é aberto e os dispositivos que tenham os módulos *ZigBee* instalados podem funcionar em qualquer parte do planeta, permitindo implementar o conceito “*Internet of Things*” (isto é, conectar todos os aparelhos à *Internet*). Empresas como a Comcast [12] (Filadélfia, Pensilvânia, Estados Unidos da América), NTT [12] (Japão), EchoStar [12] (Estados Unidos), Rogers [12] (Canadá), Videocon [12] (Índia), entre outras, implementam soluções usando o *ZigBee*. A tecnologia *ZigBee* pode ser uma alternativa simples e barata a outros sistemas de comunicação sem fios, como o *Bluetooth* ou o *Wi-Fi*. Como o *ZigBee* é um protocolo implementado em dispositivos de baixo consumo, as distâncias de transmissão do sinal são entre os 10 e os 100 metros, dependendo da potência de saída e das características do ambiente.

Cada rede *ZigBee* tem que ter obrigatoriamente um coordenador que a inicia e mantém os dispositivos associados. Na topologia em estrela, representada na Figura 2.2, toda a informação fornecida pelos e para os *ZigBee Endpoints* tem que passar, obrigatoriamente, pelo *ZigBee Coordinator*. Sendo assim, o *ZigBee Coordinator* é responsável por criar e gerir a rede que envolve tanto o *Coordinator* como os *Endpoints*. Na topologia em malha da Figura 2.3, um terceiro dispositivo, chamado *ZigBee Router*, substitui o *Coordinator*, ficando o último a registar as entradas e as saídas. Esta distribuição da rede permite a optimização do trânsito de informação e mantém o fluxo da mesma caso exista uma falha em algum dispositivo de encaminhamento. Na última topologia (Figura 2.4), o *Coordinator* continua a ser o gestor da rede. No entanto, enquanto os *Endpoints* das ramificações secundárias podem comunicar entre si enviando a informação para o *Router*, este só pode comunicar com outro *Router* depois de ter enviado a informação para o *Coordinator*.

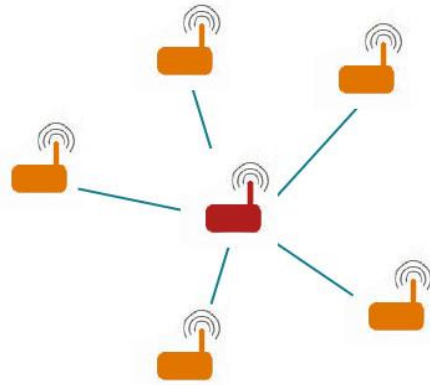


Figura 2.2 – Topologia em estrela. Fonte: [13].

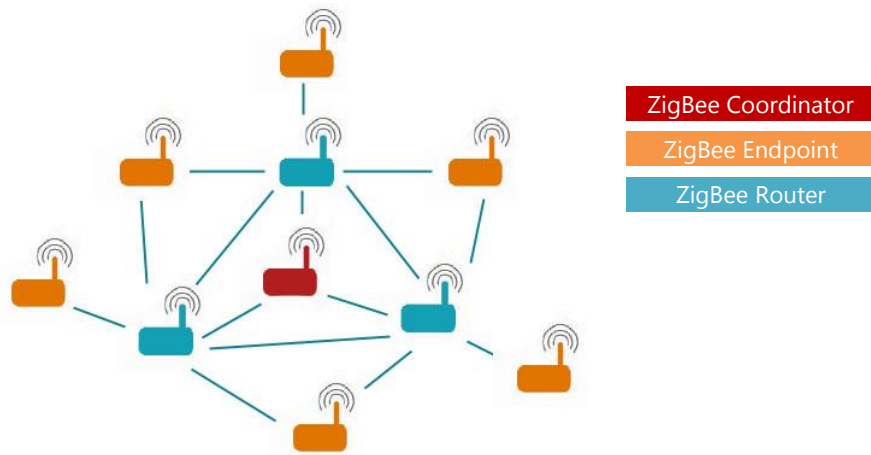


Figura 2.3 – Topologia em malha. Fonte: [13].

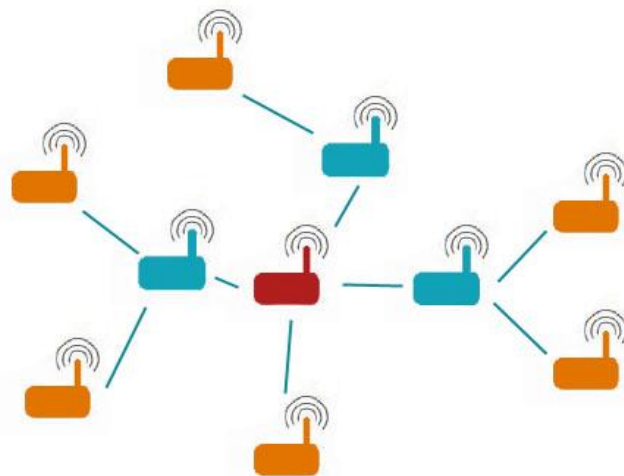


Figura 2.4 – Topologia em árvore. Fonte: [13].

Dado que a norma IEEE utilizada define que a velocidade máxima de transmissão é de 250 kbit/s a 10 m, este protocolo não tem como finalidade a transmissão de uma grande quantidade de dados. Assim, o *ZigBee* deve ser utilizado em sistemas que precisem de enviar poucos dados, superando, na parte dos consumos energéticos, alguns protocolos mais utilizados, como o *Bluetooth*, o *Wi-Fi* ou o *GSM*, permitindo por isso a alimentação através de baterias.

2.2.2 6LoWPAN

O protocolo *6LoWPAN* nasceu da ideia de que o “protocolo da *Internet* podia e devia ser aplicado em pequenos dispositivos” e que estes dispositivos, dotados de reduzidas capacidades de processamento, deviam participar no conceito da *Internet* das coisas [14]. Como foram criados módulos que utilizam o protocolo *6LoWPAN*, módulos que serão descritos detalhadamente no Capítulo 3, apresenta-se nesta subsecção uma pequena descrição sobre o protocolo mencionado.

Para implementar a comunicação sem fios entre os módulos de medição e controlo das cargas trifásicas e a unidade de processamento central, optámos pelo protocolo *6LoWPAN*, porque permite implementar o conceito de *Internet of Things*. Por outras palavras, este protocolo serve para dotar os dispositivos, que possuem características de processamento e de memória reduzidas, com a capacidade de comunicar, utilizando a *stack* de protocolos *IP* (*Internet Protocol*) [14].

Protocolos	ZigBee	6LoWPAN
Camada de Aplicação	Camada de Aplicação	
Camada de Sessão		
Camada de Transporte		
Camada de Rede	Camada de Rede do ZigBee	IPv6/6LoWPAN
Camada de Ligação de Dados	IEEE 802.15.4	
Camada Física		

Tabela 2.1 – Especificações dos protocolos *ZigBee* e *6LoWPAN*. Fonte: [14].

Pela Tabela 2.1, verificamos que a única diferença entre *ZigBee* e *6LoWPAN* reside na camada de Rede do modelo *OSI* (*Open Systems Interconnection*), isto porque o *ZigBee* não suporta de forma nativa a comunicação com os sensores via protocolo *IP*.

2.3 Redes Bayesianas

As redes *bayesianas* assumem hoje uma importância elevada, porque permitem implementar algoritmos de *machine learning*, isto é, aprendizagem automática. Refira-se a este nível a patente do projeto da *Nest* [10], em que as redes *Bayesianas* aparecem como uma possível tecnologia utilizada na aprendizagem dos parâmetros externos e nos algoritmos de otimização.

Uma rede *bayesiana* é assim uma estrutura que resume as relações entre cada uma das suas variáveis aleatórias (nós), através de tabelas de probabilidades. Cada tabela probabilística pode depender de outros nós dessa mesma rede. Com estas tabelas, o *software* pode indicar, por exemplo, qual a probabilidade de determinado acontecimento, sabendo que algo já aconteceu.

Apesar destas redes se basearem em probabilidades, em muitos casos estas últimas são desconhecidas, pelo que precisam de ser criadas. Dependendo do grau de desconhecimento, a construção da rede torna-se mais ou menos complexa. Para ajudar a compreender melhor este conceito, vamos recorrer a um exemplo típico da origem da dor de costas de um trabalhador [15]. A rede encontra-se na Figura 2.5.

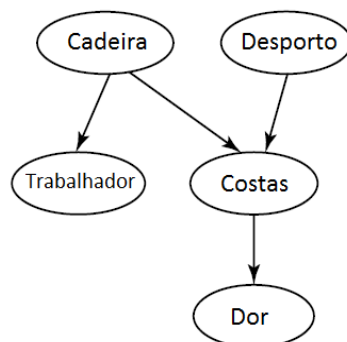


Figura 2.5 – Rede Bayesianas do exemplo da dor de costas. Editado de [15].

Este exemplo representa uma pessoa que pode sofrer uma lesão nas costas, representada pela variável Costas. Esta lesão pode provocar uma dor de costas, representada pela variável Dor. As origens da lesão podem ser um exercício de desporto mal feito ou uma cadeira desconfortável, representadas pelas variáveis Desporto e Cadeira, respetivamente. A variável Trabalhador representa um trabalhador que apresenta queixas semelhantes às de uma pessoa que se senta numa cadeira desconfortável, sendo que a possível causa seja uma cadeira não confortável.

Estas redes são definidas através da Distribuição Conjunta de Probabilidade (DCP), isto é, a interceção das probabilidades de um conjunto de variáveis n é o produto de todas as probabilidades condicionadas do nó em estudo por todos os nós pai.

Matematicamente [15]:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)). \quad (1)$$

Para este exemplo a DCP da rede é:

$$P(Ca, De, T, Co, Do) = P(Ca) \times P(De) \times P(T|Ca) \times P(Co|Ca, De) \times P(Do|Co) \quad (2)$$

As variáveis Cadeira (Ca) e Desporto (De) não têm dependências (nós pai), por isso não são condicionadas por estes; a variável Costas (Co) tem 2 dependências (Cadeira e Desporto), ou seja, é condicionada pelo acontecimento conjunto destas 2 variáveis e a variável Dor (Do) apenas tem como dependência a variável Costas, ficando apenas condicionada por esta.

A Figura 2.6 apresenta um exemplo das tabelas de probabilidades relativas ao esquema da Figura 2.5.

P(Ca=T)	P(Ca=F)
0.8	0.2

P(De=T)	P(De=F)
0.02	0.98

Ca	P(T=T Ca)	P(T=F Ca)
T	0.9	0.1
F	0.01	0.99

Ca	De	P(Co=T Ca,De)	P(Co=F Ca,De)
T	T	0.9	0.1
T	F	0.2	0.8
F	T	0.9	0.1
F	F	0.01	0.99

Co	P(Do=T Co)	P(Do=F Co)
T	0.7	0.3
F	0.1	0.9

Figura 2.6 – Tabelas de probabilidades da rede *Bayesiana* da dor de costas. Editado de [15].

Pelas tabelas de probabilidades da Figura 2.6, se alguém se queixar de dores, existe uma probabilidade de 70% de essa dor ser de uma lesão nas costas.

2.3.1 Obtenção de Probabilidades

Na maior parte dos casos, as tabelas de probabilidades dos nós são desconhecidas, fazendo com que a rede tenha de “aprendê-las”. Esta aprendizagem consiste na utilização dos dados para determinar os parâmetros da função de probabilidade que melhor representa os dados obtidos. Para fazê-lo, utilizam-se duas soluções que partem dos dados observados para estimar a probabilidade inicial.

A primeira dessas soluções é denominada de *Bayesian Parameter Estimation* e consiste em utilizar o teorema de *Bayes*, partindo de uma função de probabilidade à *priori*. Esta função funciona como um “palpite” de como será a distribuição dos dados. A seguir, utiliza-se o teorema de *Bayes* e os dados para calcular a função de probabilidade à *posteriori*, ou seja:

$$P(\theta | \text{dados}) = \frac{P(\text{dados} | \theta) \times P(\theta)}{P(\text{dados})} \quad (3)$$

Em (3), θ representa a probabilidade à *priori*, $P(\theta | \text{dados})$ representa a probabilidade à *posteriori* e $P(\text{dados})$ representam as probabilidades obtidas a partir dos dados.

Uma outra solução denomina-se de *Maximum Likelihood Estimation* (MLE). O que este algoritmo faz é estimar a distribuição de probabilidades p que maximiza:

$$\max P(\text{Dados} | p) = \max \prod_{i=1}^m P(\text{resultado}(i) | p) \quad (4)$$

Para perceber como funciona vamos ver um exemplo. Se tivermos uma moeda que desconhecemos a probabilidade de sair cara e a lançarmos 6 vezes saindo cara 4 vezes e coroa 2 vezes, a probabilidade de sair cara p deve ser estimada de acordo com a expressão anterior. Isto neste caso corresponde a maximizar a expressão $p^4(1-p)^2$.

Para calcular a probabilidade que maximiza a função de probabilidade, devemos procurar os seus máximos. Para tal deriva-se a função de probabilidade, em ordem a p , e iguala-se a 0. Matematicamente:

$$\frac{\partial f_p}{\partial p} = 0 \quad (5)$$

O resultado obtido é:

$$0 = 2p^3(3p^2 - 5p + 2) \quad (6)$$

Neste caso verificamos que o máximo se encontra em $p=0$, $p=1$ e $p=2/3$.

Os valores 0 e 1 não servem, porque ambos fazem com que o resultado da função de probabilidade seja 0.

Veamos outro exemplo com mais lançamentos. Considerando que efetuamos 80 lançamentos e que saem 49 cara, vamos agora admitir que queremos maximizar $P(\text{Caras}=49/p)$, em que a probabilidade p pode assumir qualquer valor entre 0 e 1, isto é:

$$P(\text{Caras} = 49 | p) = p^{49}(1-p)^{31} \quad (7)$$

As soluções desta equação são: $p=0$, $p=1$ e $p=49/80$.

Mais uma vez, os valores 0 e 1 não servem, porque ambos fazem com que o resultado da função de probabilidade seja 0. Por isso, a solução $p=49/80$ é aquela que maximiza a função de probabilidade. Por outras palavras, o valor de $p=49/80$ é chamado o estimador da maior probabilidade.

Estes exemplos mostraram-nos como calcular o valor da probabilidade utilizando os dados. Se, em algum nó, não tivermos dados, devemos recorrer ao algoritmo de maximização da expectativa (*Expectation-Maximization*, EM). Para uma informação mais detalhada sobre o algoritmo EM, consulte a referência [16].

Em alguns casos, podemos saber os nós mas não as relações entre os mesmos. Nestes casos, o *software* precisa de criar a estrutura. Este processo é complicado, uma vez que o número de combinações entre os nós escolhidos é superexponencial. Para tentar contornar este problema, recorre-se ao método *Markov Chain Monte Carlo* (MCMC). Para informação mais detalhada sobre o método MCMC, consulte a referência [17].

No entanto, existe uma variação das redes *Bayesianas*, que são as redes *Bayesianas Dinâmicas* (RBD). As redes *Bayesianas* dinâmicas têm relações entre os diferentes nós semelhantes àquelas que surgem nas cadeias de *Markov*. Por isso, na secção que se segue, encontra-se o essencial das cadeias de *Markov*.

2.3.2 Cadeias de *Markov*

As cadeias de *Markov* são redes constituídas por nós discretos que mostram as transições entre estados ou nós. Estas cadeias permitem-nos ver qual será a transição de estados mais provável num determinado tempo t .

Vamos analisar um exemplo. Num determinado mercado bolsista, as tendências podem alternar entre 3 estados: *bull*, *bear* e *stagnant* e sabe-se que, na semana seguinte, o estado pode manter-se ou modificar-se. Uma cadeia de *Markov* que representa estes estados encontra-se na Figura 2.7:

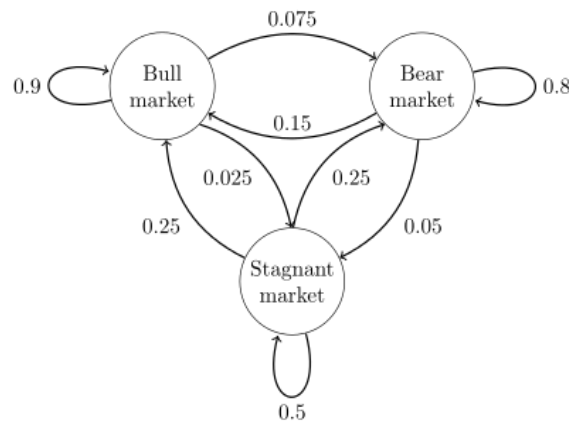


Figura 2.7 – Cadeia de Markov exemplificando um mercado bolsista.
Fonte: [18].

Esta figura mostra que, na semana seguinte, o estado do mercado tende a ser o da semana atual. Supondo que queremos saber qual será o estado do mercado em $t+3$, e sabendo que os estados 1, 2 e 3 são, respetivamente, *bull*, *bear* e *stagnant*, e que, atualmente, estamos no estado *bear*, temos que, dado P :

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \quad (8)$$

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix} \quad (9)$$

em que P representa as probabilidades associadas aos estados à matriz de transição. Como queremos saber qual será o estado em $t+3$, este pode ser obtido utilizando a seguinte fórmula:

$$x^{t+3} = x^{t+2} \times P = x^{t+1} \times P^2 = x^t \times P^3 \quad (10)$$

Esta fórmula diz que o estado seguinte é o produto do estado atual pela matriz de transição. Substituindo as incógnitas:

$$x^{t+3} = [0.3575 \quad 0.56825 \quad 0.07425] \quad (11)$$

$$x^{t+3} = [0 \quad 1 \quad 0] \times P^3 \quad (12)$$

Os resultados indicam-nos que no tempo $t+3$, existe 35.75% de probabilidade de transitar para uma semana *bull*, 56.83% de se manter no estado *bear* e 7.43% de passar para uma semana *stagnant*.

Existem algumas variações das cadeias de *Markov* que devem ser tidas em conta: as de tempo homogéneo ou invariáveis no tempo e as de ordem m . As de tempo homogéneo dizem que a probabilidade do estado seguinte ter o valor x , dado que esse estado, atualmente, tem o valor y , é igual à probabilidade do estado atual ter o valor x , dado que o estado anterior tinha o valor y . Do ponto de vista matemático:

$$P(X_{n+1} = x | X_n = y) = P(X_n = x | X_{n-1} = y) \quad (13)$$

As cadeias de ordem m são uma generalização das cadeias homogéneas, em que a probabilidade do estado atual tomar o valor x vai ser condicionada pelos m estados anteriores, isto é:

$$P(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_{n-m} = x_{n-m}), n > m \quad (14)$$

2.3.3 Redes *Bayesianas* Dinâmicas

As redes *bayesianas* dinâmicas são iguais às redes *bayesianas* normais, com a particularidade de que as primeiras podem alterar a sua estrutura e/ou as suas tabelas de probabilidades ao longo do tempo. Para além disso, os nós no tempo $t+1$ dependem dos nós no tempo t e as tabelas de probabilidades podem-se alterar.

Para que no decorrer desta dissertação se pudesse aplicar as redes *Bayesianas* dinâmicas à determinação de consumos anómalos, foi realizada uma análise de diversos trabalhos. Na subsecção seguinte são mostrados alguns desses exemplos.

2.3.3.1 Exemplos de Aplicação em Projectos Reais

Nesta secção analisam-se dois trabalhos práticos que aplicam os conceitos de redes *Bayesianas* Dinâmicas em situações reais.

O primeiro projeto foi desenvolvido por Chun Zhu e Weihua Sheng e é intitulado de “*Realtime Recognition of Complex Daily Activities Using Dynamic Bayesian Network*” [19]. Utiliza redes *Bayesianas* Dinâmicas para determinar as atividades presentes e prever as atividades futuras de um indivíduo. Na Figura 2.8, ilustra-se a zona de testes usada para o efeito.

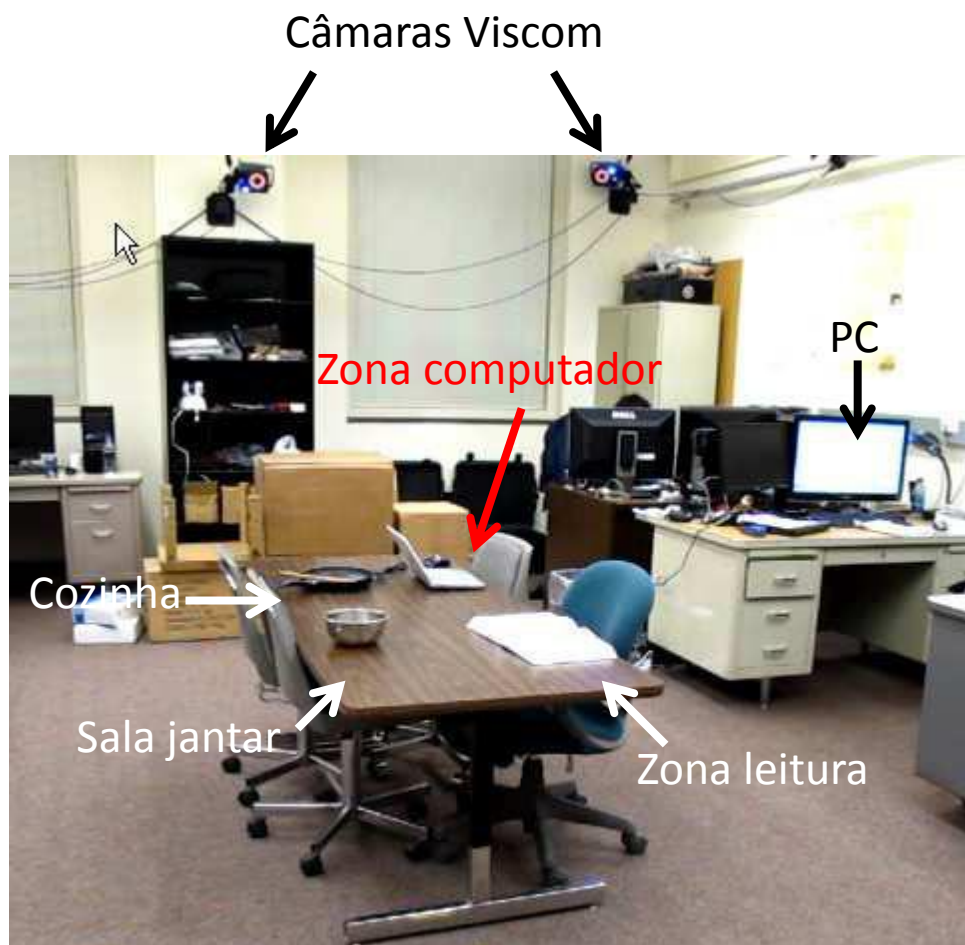


Figura 2.8 – Ambiente para testes. Editado de [19].

As câmaras servem para localizar o utilizador; a mesa central representa as diferentes áreas a estudar e o PC recebe os dados das 3 câmaras e dos sensores do utilizador presentes na cintura, no joelho e na mão. A rede *Bayesiana* utilizada é a seguinte:

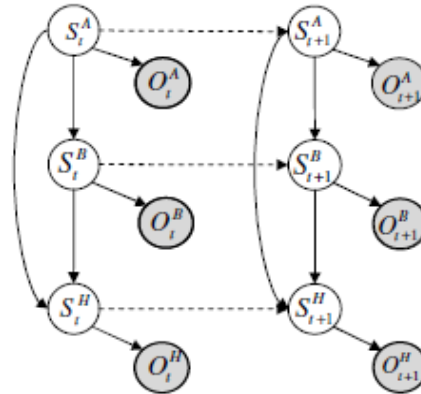


Figura 2.9 – Rede Bayesiana de “Realtime Recognition of Complex Daily Activities Using Dynamic Bayesian Network”. Fonte: [19].

Nas redes *Bayesianas* Dinâmicas, as cores seguem uma norma: os balões que estão a cinzento representam as variáveis observáveis, e os que estão a branco são as variáveis não observáveis. As variáveis são: O_t^A , que representa os dados das câmaras Viscom no tempo atual t ; O_t^B , que representa os dados do sensor do corpo no tempo atual t ; O_t^H , que representa os dados do sensor da mão no tempo atual t ; S_t^A , que é a estimativa da área onde se encontra a pessoa no tempo atual t ; S_t^B , que é a estimativa da atividade do corpo no tempo atual t ; e S_t^H , que é a estimativa da atividade da mão no tempo atual t . As variáveis que têm $t+1$ são as que pretendemos prever, ou seja, são as atividades futuras.

Utilizando a rede *Bayesiana* representada na Figura 2.9 e os seus princípios dados no início do subcapítulo, verificamos que as atividades desenvolvidas pela mão vão depender das atividades do corpo e do local onde o indivíduo se encontra. Assim, a probabilidade de cozinhar na sala deve ser quase nula. Para verificar se o sistema efetuou as verificações corretas, foi criada uma “verdade indiscutível”. Esta verdade foi criada usando rótulos que eram gerados quando o utilizador carregava num botão de um PDA. Esses rótulos, constituídos pela localização e pelas atividades da mão e do corpo, eram fornecidos por uma câmara digital.

Os valores a negrito na Tabela 2.2 são os valores corretos e indicam a percentagem de decisões corretas utilizando a “verdade indiscutível” como referência. Tal como é visível, a

exatidão da previsão variou entre um mínimo de 76% (para a utilização do rato) até um máximo de 100%.

Verdade	Estimativa do Algoritmo											Exatidão
	Parado	Sentar para levantar	Levantar para sentar	Em pé	Andar	Escrever com o teclado	Utilizar o rato	Virar uma página	Cozinhar	Comer	Misto	
Parado	1											1
Sentar para levantar		0.92			0.08							0.92
Levantar para sentar			0.90		0.06						0.04	0.90
Em pé				1								1
Andar			0.02		0.98							0.98
Escrever com o teclado						0.83	0.08				0.09	0.83
Utilizar o rato						0.05	0.76				0.19	0.76
Virar uma página								0.85			0.15	0.82
Cozinhar									0.82		0.18	0.82
Comer										0.80	0.2	0.80

Tabela 2.2 – Resultados obtidos pelo modelo adotado. Fonte: [19].

Num segundo projeto, desenvolvido por Ehsan Nazerfard e Diane J. Cook e intitulado “*Using Bayesian Networks for Daily Activity Prediction*” [20], também se utilizam redes Bayesianas Dinâmicas para prever a próxima atividade desenvolvida por um indivíduo.

As previsões são feitas em 2 passos: previsão das características da próxima atividade e previsão da próxima atividade. A rede criada para este efeito foi a seguinte:

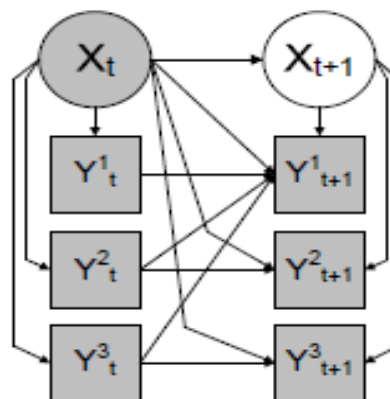


Figura 2.10 – Rede Bayesiana Dinâmica de “Using Bayesian Networks for Daily Activity Prediction”. Fonte: [20].

As variáveis X_t, Y_t^1, Y_t^2 e Y_t^3 representam, respetivamente, a atividade atual, a localização da atividade atual, a hora do dia em que é executada a atividade atual e o dia da semana em que é executada a atividade atual, respetivamente.

Para fazerem testes, os investigadores utilizaram dois apartamentos, cada um deles com um quarto apenas, com as características presentes na Tabela 2.3.

	Apartamento 1	Apartamento 2
Nº sensores movimento	20	18
Nº sensores portas	12	12
Nº de residentes	1	1
Nº de eventos recolhidos	371.925	254.918
Tempo dispensado	6 meses	4 meses

Tabela 2.3 – Características de teste. Fonte: [20].

A rede ilustrada na Figura 2.10 representa o CRAFFT (Using **CuRrent Activity and Features to predict the next FeaTures**). Pela rede, verifica-se que as características da atividade seguinte vão depender das características da atividade atual. A determinar cada hora do dia e do dia da semana da atividade seguinte é obtida pela expressão:

$$y_{t+1}^* = \arg \max_{y_{t+1}} P(Y_{t+1} = y_{t+1} | Y_t = y_t, X_t = x_t) \quad (15)$$

Esta equação traduz que a característica da atividade seguinte é obtida pela maximização da probabilidade dessa característica ter o valor y_{t+1} , sabendo que o seu valor atual é y_t e a atividade atual é x_t .

Para melhor compreender este processo, a tabela seguinte mostra alguns exemplos de características de atividades que os investigadores quiseram prever.

$(X_t, Y_t^1, Y_t^2, Y_t^3)$	$(Y_{t+1}^1, Y_{t+1}^2, Y_{t+1}^3)$
(Banho, Banheira, Manhã, *)	(Casa banho, Manhã, *)
(Banho, Banheira, Noite, *)	(Quarto, Noite, *)
(Comendo, Sala jantar, Anoitecer, 2-7)	(Sala, Anoitecer, 2-7)
(Comendo, Sala jantar, Anoitecer, 1)	(Porta principal, Anoitecer, 1)
(Dormindo, Quarto, Noite, 7)	(Cozinha, Manhã, 1)

Legenda: 1 – Segunda, 2 – Terça, 3 – Quarta, 4 – Quinta, 5 – Sexta, 6 – Sábado, 7 - Domingo

Tabela 2.4 – Precisão obtida pelo CRAFFT. Fonte: [20].

Assim, os investigadores verificaram que, por exemplo, se o utilizador estivesse a tomar banho, na banheira, numa manhã de qualquer dia da semana, a maior probabilidade correspondia a que, a seguir, o utilizador permanecesse na casa de banho (na manhã de qualquer dia da semana). Já o mesmo não acontece à noite. É importante relembrar que, nesta fase, o CRAFFT não prevê a atividade seguinte, apenas a sua localização.

Os resultados obtidos neste passo foram os que se encontram na Tabela 2.5.

	Precisão da previsão das características da próxima atividade em função da atividade atual		
	Localização	Hora do dia	Dia da semana
Apartamento 1	46.60%	88.10%	96.74%
Apartamento 2	60.94%	86.41%	96.48%

Tabela 2.5 – Exemplos das características da atividade seguinte a prever. Fonte: [20].

Os resultados indicam que as melhores previsões se verificam para o dia da semana da atividade seguinte. Conseguimos ainda concluir que a previsão do comportamento do utilizador do apartamento 2 aproxima-se mais do que realmente acontece.

Feita esta previsão, é introduzido o CEFA (CurrEnt Features and activity to predict the next Activity). O que este sistema faz é tentar prever a próxima atividade a partir da atividade atual. Revendo a Figura 2.10, a equação que traduz a previsão da atividade seguinte é:

$$x_{t+1}^* = \arg \max_{x_{t+1}} P(X_{t+1} = x_{t+1} \mid X_t = x_t) \quad (16)$$

A previsão da próxima atividade é igual ao argumento que maximiza a probabilidade da próxima atividade ter o valor x_{t+1} , sabendo que a atividade atual tem o valor x_t .

Os resultados obtidos foram os seguintes:

	Precisão da previsão próxima atividade
Apartamento 1	45.41%
Apartamento 2	60.05%

Tabela 2.6 – Resultados obtidos pelo CEFA. Fonte: [20].

Conjugando estes resultados com os presentes na Tabela 2.6, concluímos que a localização da atividade atual tem influência na previsão da próxima atividade, mas que os valores da eficácia do método não são elevadas.

Outra previsão que se pode fazer utilizando o modelo apresentado é o intervalo de tempo entre o início de uma atividade e o início da seguinte. Para este efeito, os autores utilizaram um intervalo de tempo e construíram um modelo de Gauss, utilizando o algoritmo de maximização da expectativa (EM).

Os resultados obtidos foram os que se encontram na Figura 2.11.

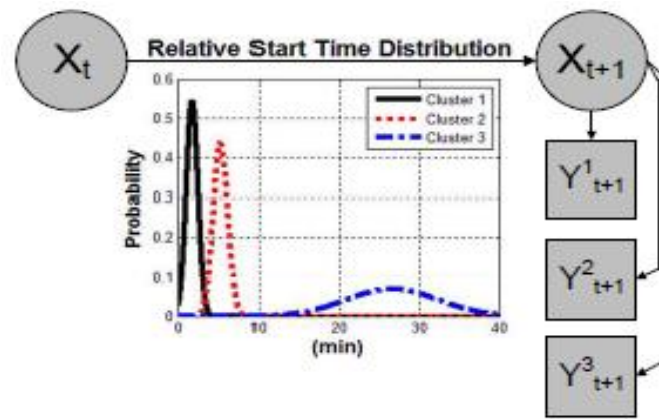


Figura 2.11 – Gráfico com as previsões dos intervalos de tempo entre o início da atividade atual e o início da atividade seguinte. Fonte: [20].

Os resultados mostram que os intervalos de tempo mais habituais variam entre, aproximadamente, 3 minutos e 6 minutos. Um exemplo que os autores consideraram foi o intervalo de tempo entre o pequeno-almoço e a toma da medicação. A figura seguinte mostra as probabilidades dos intervalos de tempo entre essas atividades mencionadas.

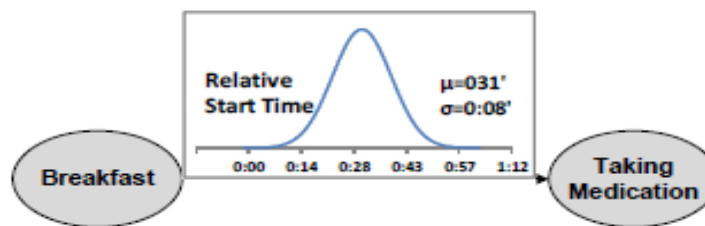


Figura 2.12 – Gráfico dos intervalos de tempo entre o início do pequeno-almoço e a toma da medicação. Fonte: [20].

Analisando os resultados presentes na Figura 2.12, verifica-se que o mais provável é que um indivíduo demore 31 minutos entre o início do pequeno-almoço e a toma da medicação. Caso tal não se verifique, o sistema pode gerar um aviso ao utilizador a informar para não se esquecer de tomar a medicação.

2.3.4 Algoritmo de Viterbi

Na secção anterior foram apresentados exemplos práticos da aplicação das redes Bayesianas para previsões de acontecimentos. No entanto, às vezes é importante tentar saber qual foi a sequência de estados que levou à apresentação de determinado acontecimento. O

algoritmo de *Viterbi* tem essa finalidade. Sendo apresentado um número de resultados e conhecendo a estrutura da rede, o algoritmo permite encontrar qual a sequência de estados mais provável que levou aos resultados obtidos. Este algoritmo não é de previsão, mas sim de “*debugging*”, ou seja, de detecção de erros.

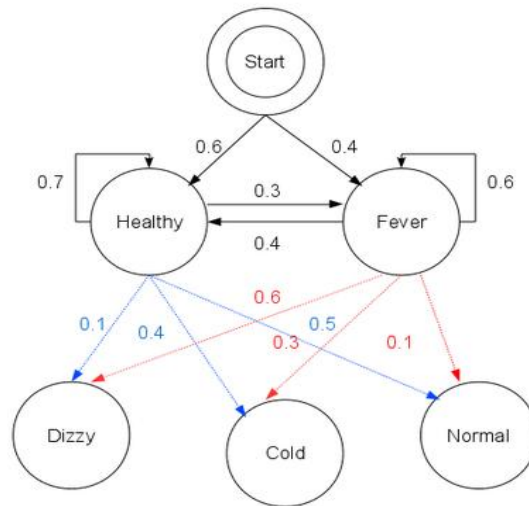


Figura 2.13 – Estrutura da rede para avaliação do estado de saúde do paciente. Fonte: [21].

Para melhor mostrar a utilidade do algoritmo, segue-se um exemplo: numa determinada clínica, as pessoas que chegam estão saudáveis ou com febre; como o médico não sabe à partida qual o estado de saúde da pessoa, faz perguntas aos pacientes e estes respondem se se sentem normais, enjoadas ou com frio. Este processo repete-se a cada dia. A estrutura da rede é a que se encontra na Figura 2.13. Antes de continuar a explicação, vamos apresentar as equações do algoritmo que permitem a resolução do problema.

Supondo que temos um modelo de *Markov* escondido (certos estados não são observáveis) com espaços de estados S , as probabilidades de estar no estado i (π_i) e a de transitar do estado i para o estado j ($a_{i,j}$) e as saídas y_1, y_2, \dots, y_t , a sequência de estados mais provável x_1, x_2, \dots, x_t que produz os dados observados e o estado final k é dada pelas seguintes relações recursivas:

$$V_{1,k} = P(y_1 | k) \times \pi_k \quad (17)$$

$$V_{t,k} = \max_{x \in S} (P(y_t | k) \times a_{x,k} \times V_{t-1,x}) \quad (18)$$

$V_{t,k}$ representa a probabilidade da sequência de estados mais provável que leva a que as primeiras t observações tenham k como estado final.

A complexidade deste algoritmo é de $O(T \times S^2)$ sendo T o número de saídas e S o número de estados.

Regressando ao problema que deu origem a este subcapítulo, num determinado dia um paciente entra na clínica e diz que no 1º dia se sentia normal, no 2º dia se sentia com frio e no 3º dia se sentia enjoado. Com estes resultados, o médico tenta encontrar a evolução do estado de saúde do paciente, ou seja, a sequência de estados mais provável que levaram aos resultados observados.

A função *viterbi* utiliza como argumentos: *obs* é a sequência de observações tais como ['normal', 'cold', 'dizzy']; *states* são as variáveis ocultas; *start_p* são as probabilidades iniciais; *trans_p* são as probabilidades de transição de estados; e *emit_p* são as probabilidades de se obter o veredicto (normal, enjoado, frio).

Depois de aplicado o algoritmo de *Viterbi*, a conclusão a que se chegou foi que, provavelmente, o paciente estava normal, mas que tinha contraído febre no 3º dia.

Apesar deste exemplo aparentemente não se relacionar com o nosso projeto, ele faz sentido na medida em que demonstra a utilidade do algoritmo de *Viterbi* para a deteção de anomalias em situações em que se pretende verificar qual a causa que conduziu a uma anomalia. O fluxograma que se encontra no Anexo A representa os passos necessários para se implementar o algoritmo de *Viterbi*.

2.4 Conclusão preliminar

Depois de apresentados alguns trabalhos na área das redes de sensores, complementados com conceitos base na área das redes *Bayesianas*, segue-se uma conclusão preliminar sobre que tipos de redes e protocolos devem ser usados no presente trabalho.

No nosso caso, queremos prever o consumo de energia elétrica de um edifício com o objetivo de detetar possíveis anomalias. Assim, para obter os dados ambientais usaremos sensores de temperatura e de radiação solar, porque, como já foi referido, os equipamentos que consomem mais energia elétrica são os da classe AVAC. Esses dados serão transmitidos através do *ZigBee*, porque é um protocolo de baixo consumo e porque não precisamos de transmissões a grandes distâncias. Para detetarmos as anomalias precisamos de saber qual foi o consumo real. Para esse efeito, foram desenvolvidos dois módulos: um de medição dos consumos elétricos trifásicos e outro de controlo remoto das cargas trifásicas.

Na Figura 2.14 identificam-se os principais fatores que interferem no consumo energético e que serão o ponto de partida para a comparação do consumo real com o previsto. Assim, mede-se a temperatura e a luminosidade utilizando sensores adequados e os resultados serão analisados em função das horas do dia e dos dias da semana, que podem interferir com a ocupação do edifício. Com estes dados poderemos prever o consumo, compará-lo com o consumo efetivo e concluir, por essa via, se existem ou não anomalias.

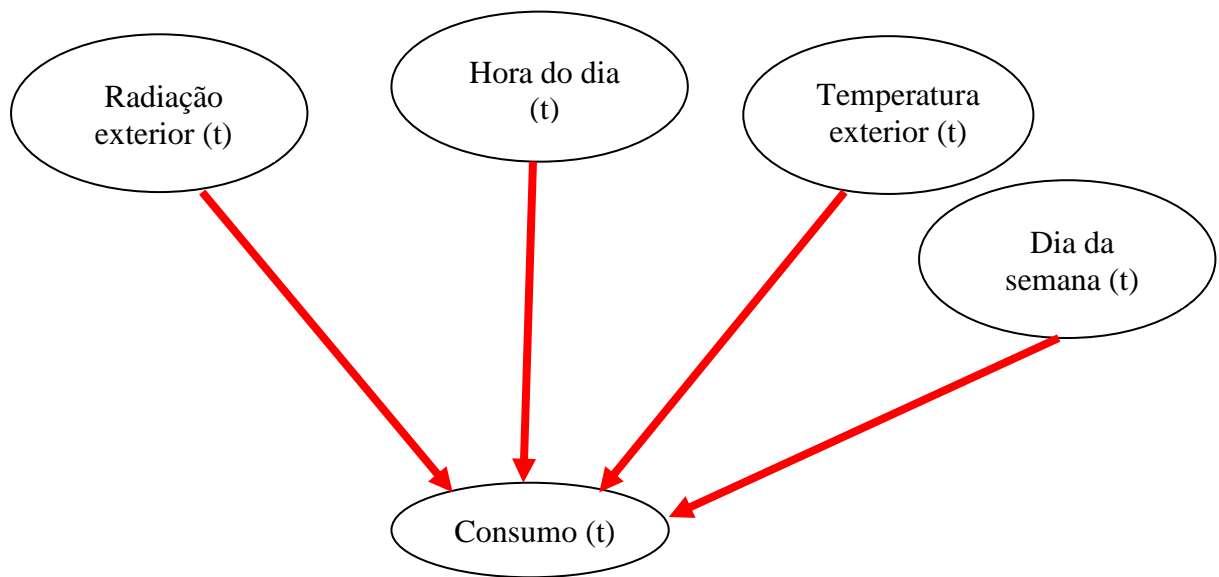


Figura 2.14 – Proposta de uma rede *Bayesiana*.

Capítulo 3 Módulo de Medição e Monitorização Remota

Para desenvolver um sistema inteligente de deteção de consumos elétricos anómalos, foram criados módulos para medir as potências elétricas trifásicas, a temperatura exterior e a radiação solar. Para controlo de cargas elétricas desenvolveu-se, também, um módulo de comutação de cargas trifásicas. A interação entre cada um dos módulos encontra-se representada na Figura 3.1.

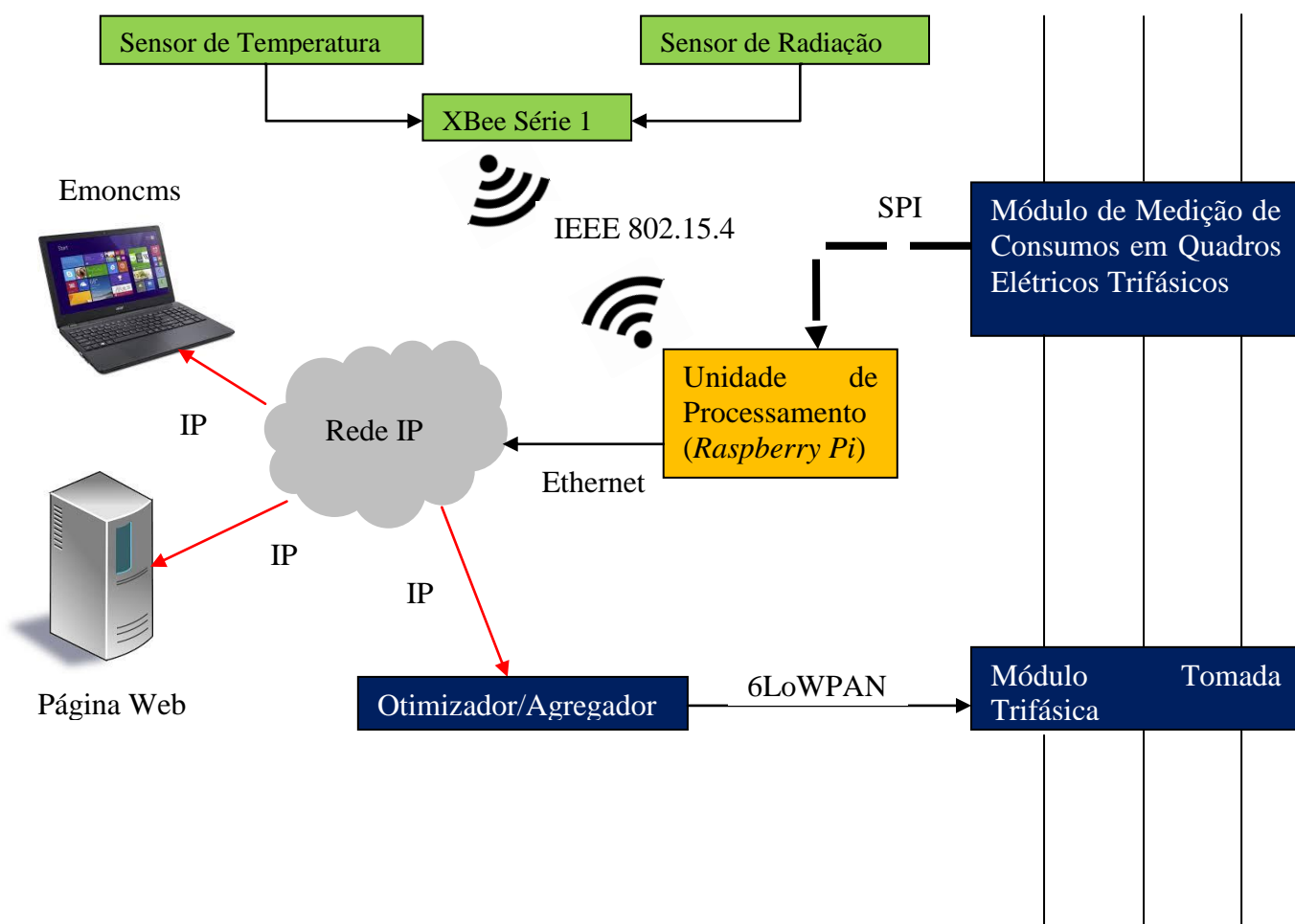


Figura 3.1 – Arquitetura do sistema.

A arquitetura deste capítulo começa com uma abordagem à comunicação *SPI* (*Serial Peripheral Interface*), que é utilizada para a obtenção das tensões e correntes através do *ADC* (*Analog-to-Digital Converter*). A seguir é feita uma explicação dos circuitos para a obtenção das tensões e correntes. Este capítulo termina com as implementações efetuadas, tanto ao nível de *hardware* como de *software*.

3.1 Obtenção dos Valores de Tensão e Corrente no Módulo de Medição de Consumos

Os sinais de tensão e corrente são obtidos através de um processo de amostragem utilizando um *ADC* externo (MCP3202), sendo posteriormente enviados para um *Raspberry Pi* através de uma interface *SPI*. O *Raspberry Pi* é também responsável por alimentar com uma tensão de 3,3 V os elementos do circuito de medição de corrente, de tensão e o módulo da comunicação.

A interface *SPI* consiste numa interface síncrona que utiliza 4 ligações, tal como apresentado na Figura 3.2 [22].

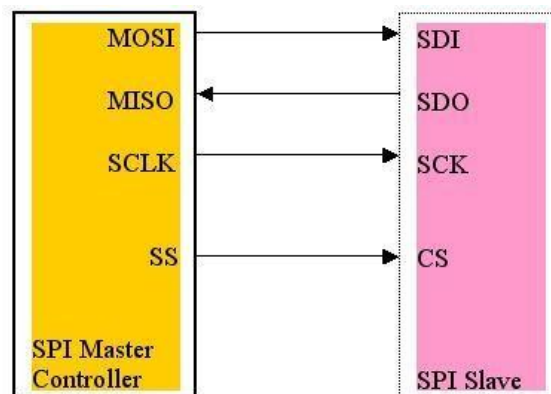


Figura 3.2 – Interligação *SPI*. Fonte:[22].

A frequência máxima do relógio (pinos SCLK e SCK) depende da tensão da alimentação do *ADC*. De acordo com o *datasheet*, para a tensão da alimentação 2,7 V, a frequência máxima do relógio é igual a 0,9 MHz. Se se considerar que a dependência é linear, para 3,3 V a frequência máxima é igual a 1,1 MHz. Nos testes seguintes é utilizada a frequência de 1 MHz.

A Figura 3.3 representa o esquema de ligação para obtenção dos dados do sistema de energia através do ADC, que converte os valores num código digital. O pino “CH0” está ligado à saída do circuito de medição da tensão da fase 1, o “CH1” está ligado à saída do circuito de medição da corrente da fase 1, o pino “CH2” está ligado à medição da tensão da fase 2, o “CH3” está ligado à medição da corrente da fase 2, o pino “CH4” está ligado à medição da tensão da fase 3 e o “CH5” está ligado à medição da corrente da fase 3.

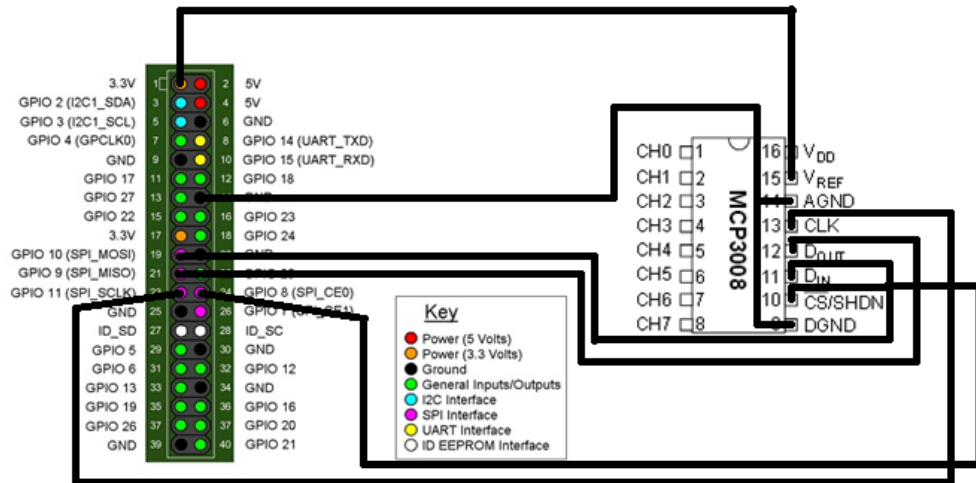


Figura 3.3 – Ligação SPI entre o Raspberry e o ADC.

Na Figura 3.4 apresenta-se a comunicação série entre o ADC e o Raspberry Pi, para modo bus 0, device 0. O valor do bus indica a forma de comunicação entre o Raspberry Pi e o ADC e o valor do device indica o endereço do dispositivo com o qual se quer comunicar.

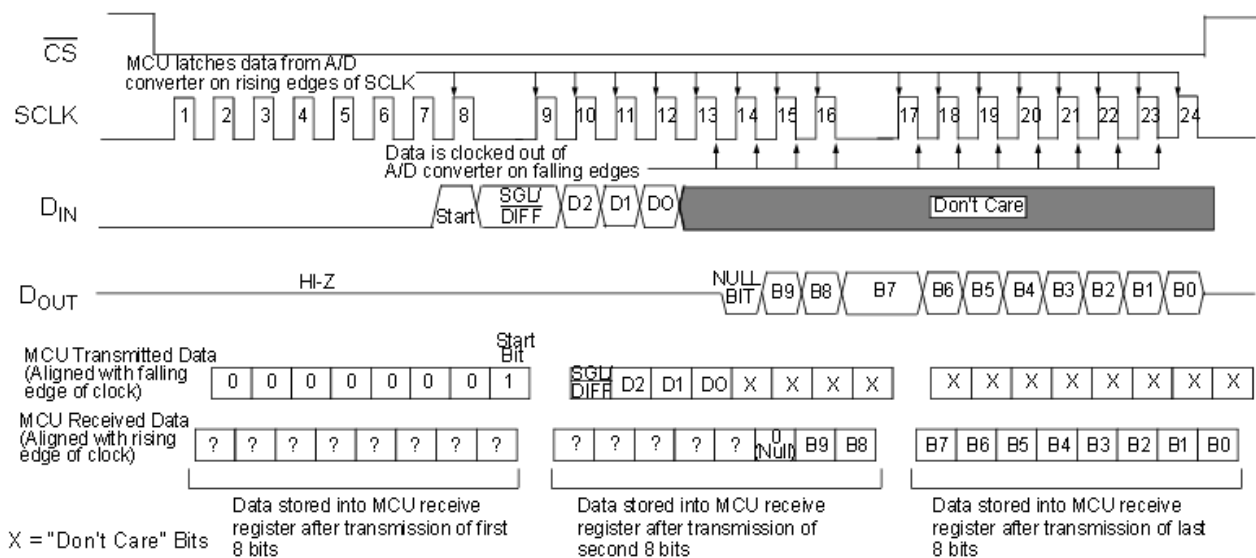


Figura 3.4 – Comunicação do ADC.

Control Bit Selections				Input Configuration	Channel Selection
Single / Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

Tabela 3.1 – Bits de configuração.

Como mostra a Figura 3.4, para obter uma amostra é necessário enviar três grupos de 8 bits. O primeiro contém apenas o *start bit* (a '1') que corresponde ao bit menos significativo. Os quatro bits seguintes, do segundo byte, traduzem quatro bits de configuração, sendo que o bit mais significativo define o modo (1 - *single-ended*, 0 - *differential*), e os 3 bits seguintes servem para seleccionar o canal. O valor do modo indica quão precisa será a leitura. Por exemplo, se o ADC estiver num ambiente onde o ruído elétrico é frequente, deve-se utilizar o modo 0 para eliminar ao máximo esse ruído. As combinações possíveis são apresentadas na Tabela 3.1. O terceiro grupo contém *bits don't care*. Os dados são enviados no modo *full-duplex*, ou seja, o Raspberry e o ADC transmitem ao mesmo tempo. Enquanto o Raspberry Pi envia os 4 bits de configuração na segunda palavra, os 5 bits enviados pelo ADC são *don't care*. Depois de recebidos os bits de configuração, o ADC começa a enviar os bits de informação, deixando, por isso, de importar o conteúdo dos dados enviados pelo Raspberry (*don't care*). Assim, depois de um *null bit* o ADC envia os primeiros 3 bits mais significativos dos 10 que constituem o código digital (*digital output code*). Os últimos 8 bits do código são transmitidos pelo ADC como sendo o último grupo. O ADC de 10 bits tem assim $2^{10} = 1024$ níveis, sendo o nível 0 correspondente a 0V e o nível 1023 a VCC que, neste caso é igual a 3,3 V (DC). Assim, o código digital pode ser convertido para a tensão utilizando a fórmula que se segue:

$$V_{in} = \frac{\text{código_digital}}{1023} \times 3.3 \quad (19)$$

Para a frequência do *SPI* escolhida, a frequência de amostragem é igual a 900 Hz, ou 18 pontos por período, por fase.

3.2 Medição de Consumos em Quadros Elétricos Trifásicos

Este módulo tem como objetivo a medição das características de um sistema elétrico trifásico. Nos subcapítulos 3.1.1. e 3.1.2. serão descritos os circuitos de medição de tensão e de corrente para cada fase, respetivamente.

3.2.1 Circuito medição de tensão

Na Figura 3.5 é apresentado o circuito de medição da tensão. Começa-se por utilizar um transformador para baixar a tensão e ao mesmo tempo isolar o circuito. Para além disso, as resistências R1, R2 e R3 formam um divisor de tensão que permite baixar mais a tensão proveniente do transformador. Como o *ADC* é unipolar (consegue ler só as tensões positivas), deve ser introduzida uma componente contínua ao sinal, com as resistências R2 e R3 a formarem outro divisor de tensão para esse propósito. O condensador C1 serve para isolar a parte alternada da parte contínua e o condensador C2 serve para implementar um filtro passa-baixo [22] que eliminará o ruído superior à frequência do sinal.

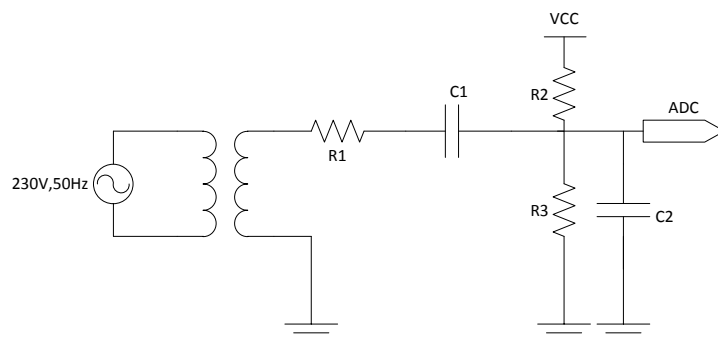


Figura 3.5 – Circuito de medição da tensão. Fonte: [22].

É utilizado um transformador de isolamento que baixa a tensão para 18 V (amplitude). Devido à gama de variações da tensão na rede elétrica, assume-se que no pior caso a tensão

variará entre até $\pm 115\%$ acima/abaixo de ± 18 V. A resistência R_1 é igual a 8710Ω , para reduzir a corrente e a dissipação de potência nestes componentes, e a componente DC é introduzida através do divisor de tensão R_2 - R_3 e é de $3,3$ V. As abreviações usadas em seguida são:

- $V_{ADC\ max}$ – tensão máxima permitida no ADC
- V_{DC} – componente contínua
- V_{CC} – tensão de alimentação DC
- V_{AC} – amplitude da componente alternada
- V_T – tensão na saída do transformador no pior caso

As relações no circuito são:

$$\left\{ \begin{array}{l} V_{CC} = 3,3 \\ V_T = 18,2 \times 1,15 \\ V_{ADC\ max} = 3 \\ R_1 = 8710 \\ V_{DC} = \frac{V_{ADC\ max}}{2} = \frac{R_3}{R_2 + R_3} \times V_{CC} \\ V_{AC} = \frac{V_{ADC\ max}}{2} = \frac{R_2 \parallel R_3}{R_1 + R_2 \parallel R_3} \times V_T \end{array} \right. \quad (20)$$

Para garantir estas condições tem-se que $R_2 \geq 1479,3103 \Omega$ e $R_3 \leq 1232,7586 \Omega$. Neste sentido foram escolhidos os seguintes valores:

$$\left\{ \begin{array}{l} R_2 = 1402\Omega \\ R_3 = 1,2k\Omega \end{array} \right. \quad (21)$$

O valor escolhido para R_2 , sendo ligeiramente menor do que tinha sido calculado no parágrafo anterior, resulta numa componente contínua maior (evitando erros de medição no ADC).

O condensador C_1 implementa um filtro passa-alto e a frequência de corte deve ser menor ou igual a 50 Hz. No cálculo da capacidade do condensador considerou-se a frequência de corte máxima de 5 Hz, para evitar a atenuação da onda no ADC. Isto deve ser feito porque, num filtro real, o ganho na frequência de corte é de -3 dB. A resistência equivalente que o condensador vê é igual à soma de R_1 com o paralelo entre: R_2 , R_3 e a resistência na entrada do ADC. Uma vez que a resistência de entrada do ADC é muito maior que as restantes, desprezando-se a primeira, temos:

$$\left\{ \begin{array}{l} f_{corte} \leq 5 \\ f_{corte} = \frac{1}{2\pi C_1 (R_2 \parallel R_3 + R_1)} \Leftrightarrow C_1 \geq 3,4\mu F \end{array} \right. \quad (22)$$

Neste sentido optou-se por utilizar um condensador de 4,7 μF .

As resistências em conjunto com o condensador C2 implementam um filtro passa-baixo. Neste caso, a frequência de corte do mesmo deverá ser maior ou igual a 50 Hz. No cálculo da capacidade do condensador considerou-se a frequência de corte máxima de 300 Hz, para evitar a atenuação da onda no ADC. Isto deve ser feito porque, mais uma vez, num filtro real, o ganho na frequência de corte é de -3dB. A resistência equivalente que o condensador vê é igual a paralelo entre R1, R2 e R3 e a resistência na entrada do ADC (desprezada porque é grande). Assim:

$$\left\{ \begin{array}{l} f_{corte} \geq 300 \\ f_{corte} = \frac{1}{2\pi C_2 (R_2 \parallel R_3 \parallel R_1)} \Leftrightarrow C_2 \leq 0,88 \mu\text{F} \end{array} \right. \quad (23)$$

Para este condensador optou-se por utilizar um valor de 0,68 μF .

Os resultados obtidos nas medições feitas encontram-se na Figura 3.6.

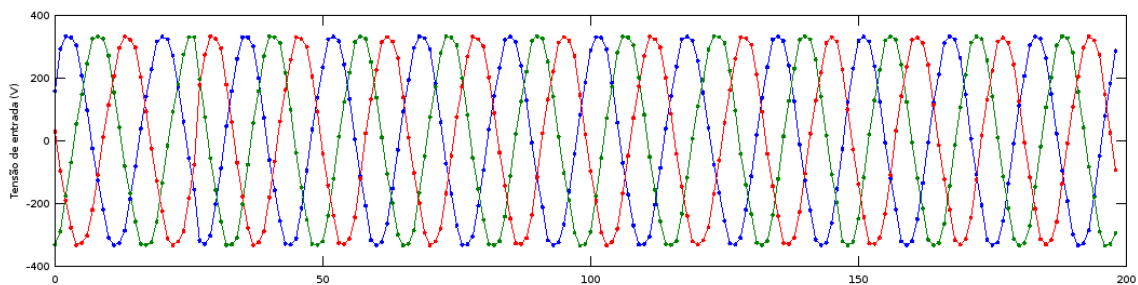


Figura 3.6 – Formas de onda das tensões das 3 fases obtidas no Raspberry Pi.

3.2.2 Circuito de medição de corrente

Abaixo, na Figura 3.7, apresenta-se o circuito de medição de corrente. Nesta figura, Rb representa a resistência de *Burden*. Esta resistência teve de ser inserida porque o transformador de corrente não possuía uma resistência integrada mas 2 diodos *zener* com o objetivo de evitar os picos de tensão caso houvesse alguma desconexão no secundário do transformador de corrente. A tensão através desta resistência é proporcional à corrente medida. Como o ADC é unipolar (consegue ler só as tensões positivas) deve ser introduzida a componente contínua ao sinal à semelhança do que foi feito no circuito de medição de tensão. As resistências R1 e R2 formam um divisor de tensão para este propósito, cuja tensão de alimentação é de 3,3 V. O condensador C1 serve para implementar um filtro passa-baixo

[22]. Considera-se ainda que a resistência de entrada do *ADC* é muito grande (quando comparada com estas resistências).

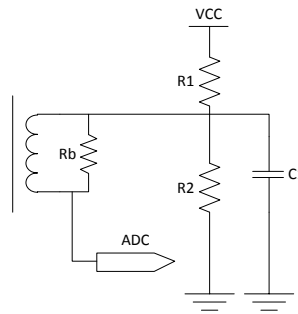


Figura 3.7 – Circuito da medição de corrente. Fonte: [22].

Foi escolhida uma resistência de 10Ω para R_b . A relação entre as resistências é:

$$\left\{ 1,5 = \frac{R_2}{R_1 + R_2} \times 3,3 \Leftrightarrow R_2 = 0,835 \times R_1 \right. \quad (24)$$

Foram escolhidos os seguintes valores:

$$\left\{ \begin{array}{l} R_1 = 140\Omega \\ R_2 = 1,2k\Omega \end{array} \right. \quad (25)$$

Estas resistências, em conjunto com o condensador C_1 implementam um filtro passa-baixo. Neste caso, a frequência de corte deve ser maior ou igual a 50 Hz. No cálculo da capacidade do condensador considera-se a frequência de corte mínima de 300 Hz, para evitar a atenuação da onda no *ADC*. Isto deve ser feito porque, mais uma vez, num filtro real, o ganho na frequência de corte é de -3dB. A resistência equivalente que o condensador vê é igual ao paralelo entre R_1 e R_2 . Portanto:

$$\left\{ \begin{array}{l} f_{corte} \geq 300 \\ f_{corte} = \frac{1}{2\pi C_1 (R_1 \parallel R_2)} \Leftrightarrow C \leq 0,82\mu F \end{array} \right. \quad (26)$$

Foi escolhido um condensador de $0,68\mu F$.

Os resultados obtidos nas medições feitas encontram-se na Figura 3.8.

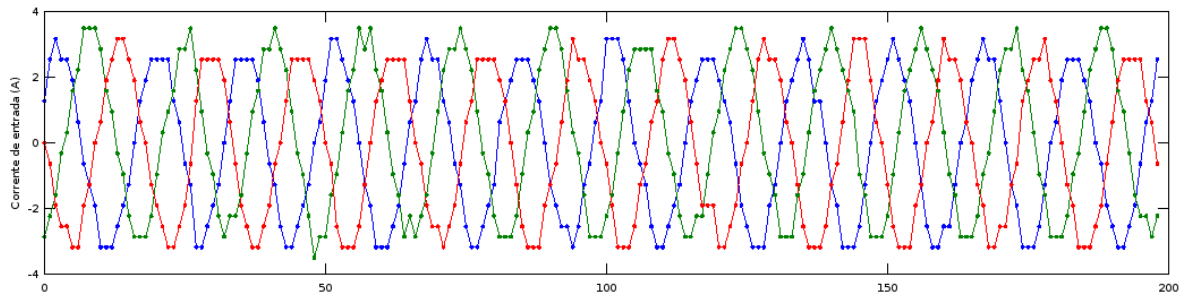


Figura 3.8 – Formas de onda das correntes das 3 fases calculadas pelo *Raspberry Pi*.

O código que permitiu a obtenção das formas de onda, tanto da tensão como da corrente, encontra-se no Anexo E.

3.2.3 Obtenção dos Valores das Potências e do Fator de Potência no Módulo de Medição dos Consumos

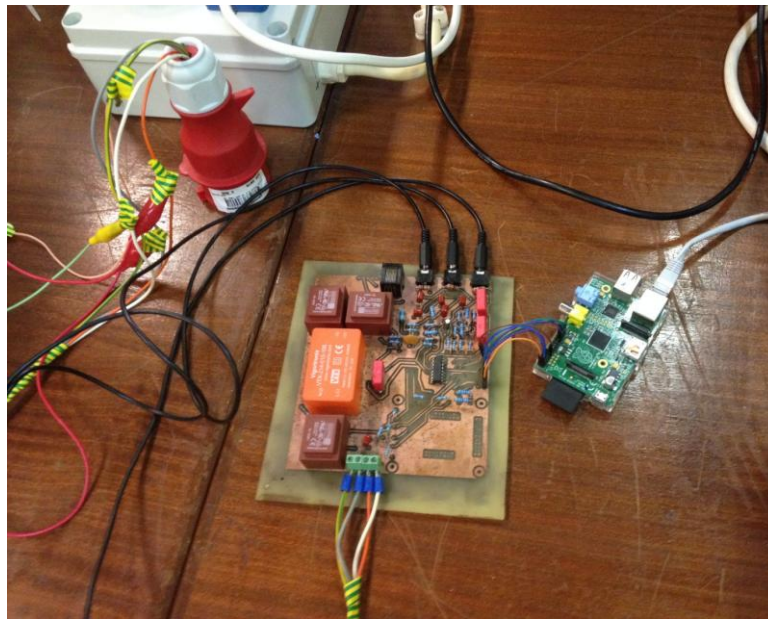


Figura 3.9 – Placa que mede os consumos elétricos em quadros trifásicos.

Para obtenção das características do sistema elétrico trifásico foi feita um *PCB* (*Printed Circuit Board*). A placa que mede os consumos elétricos trifásicos é a que se encontra representada na Figura 3.9. O desenho da placa da figura anterior encontra-se na Figura 3.10.

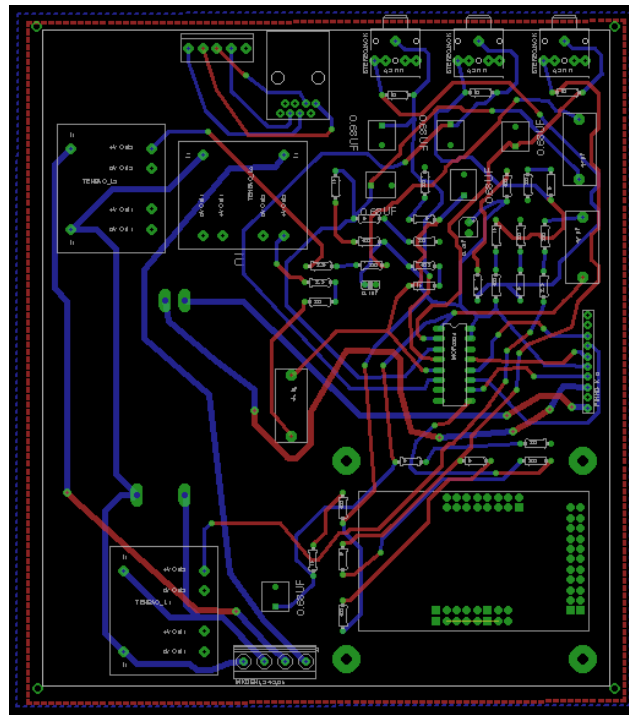


Figura 3.10 – Desenho da placa que mede os consumos elétricos em quadros trifásicos.

O programa que foi inserido no *Raspberry Pi*, que por sua vez foi colocado a funcionar junto com o *PCB*, começa por obter as curvas das tensões e das correntes das 3 fases, usando o seguinte código em Python (Anexo E) (só ilustrado para a fase 1).

```
resp_adc=spi.xfer2([1, (8+canal)<<4, 0])
adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
v1_volts=adc_out*3.3/1023
v1.append(v1_volts)

resp_adc=spi.xfer2([1, (8+canal)<<4, 0])
adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
i1_volts=adc_out*3.3/1023
i1.append(i1_volts)
```

A função `xfer2()` serve para receber um grupo de bytes através do porto *SPI* do *Raspberry Pi*, converte o valor para *volts* e anexa-o ao vetor correspondente. Este ciclo é executado 200 vezes. Depois os valores têm de ser referidos ao primário. Para tal, devido à unipolaridade do *ADC*, tem que se retirar a componente contínua do sinal e multiplicar o resultado por um fator multiplicativo, como mostra o excerto de código em baixo (Anexo F) desenvolvido em Octave.

```
medio_V1=mean(tensao_v1);
tensao_V1=270.607.*(tensao_v1-medio_V1);
V1_ef=V1_ef*(1-constante)+max(tensao_V1)/sqrt(2)*constante;
```

Este fator multiplicativo foi obtido medindo o valor eficaz da tensão no sistema de energia e dividindo-o pelo valor eficaz da tensão no *ADC*. O valor final do fator multiplicativo resultou da média de três medições consecutivas efetuadas. A última linha do código acima serve para aplicar um filtro passa-baixo em *software*, eliminando mais algumas oscilações que possam surgir nas leituras. Para calcular a corrente seguiu-se o mesmo caminho, como mostra o código seguinte (Anexo F).

```
medio_I1=mean(corrente_i1);
corrente_i1_AC=corrente_i1-medio_I1;
corrente_I1=196.364.*corrente_i1_AC;
I1_ef=I1_ef*(1-constante)+max(corrente_I1)/sqrt(2)*constante;
```

No entanto, a obtenção do fator multiplicativo da corrente foi diferente daquele que foi obtido para a tensão. Para este caso, mediu-se a tensão pico-a-pico no *ADC* da componente alternada aplicando-se, posteriormente, uma relação linear entre a corrente no sistema de energia e a tensão pico-a-pico que esta produzia na entrada do *ADC*. A constante que relacionava estas duas grandezas era o fator multiplicativo da corrente. O valor final do fator multiplicativo resultou da média de três medições consecutivas efetuadas.

Depois de obtidos os valores das tensões e das correntes das 3 fases têm-se todos os dados para calcular as potências e o fator de potência. Para calcular as potências, utilizou-se a Transformada de Concórdia. Esta transformada, que assenta numa matriz de constantes, permite passar de um sistema 1, 2, 3 (fase 1, fase 2, fase 3) para um sistema α , β , 0 (fase α , fase β , fase 0). A vantagem da utilização desta transformada é que, para um sistema equilibrado, isto é, um sistema que possui a mesma frequência e os valores eficazes nas 3 fases, a componente 0 é eliminada. Assim, libertam-se recursos computacionais e os valores obtidos são os eficazes. O código correspondente à transformação do sistema é o seguinte:

```
Tensao_concordia=M_concordia*[tensao_V1;tensao_V2;tensao_V3];
Corrente_concordia=M_concordia*[corrente_I1;corrente_I2;corrente_I3];
```

Por fim, calculam-se as potências ativa, reativa e aparente e o fator de potência utilizando:

```
Potencia_activa=Potencia_activa*(1-constante)-
constante*(Tensao_alfa.*Corrente_alfa+Tensao_beta.*Corrente_beta);
Potencia_reactiva=Potencia_reactiva*(1-constante)-constante*(-
Tensao_alfa.*Corrente_beta+Tensao_beta.*Corrente_alfa);
Potencia_activa_media=mean(Potencia_activa);
Potencia_reactiva_media=mean(Potencia_reactiva);
Potencia_aparente=Potencia_aparente*(1-
constante)+constante*(sqrt(Potencia_activa_media^2+Potencia_reactiva
_media^2));
Factor_potencia=Factor_potencia*(1-
constante)+constante*(Potencia_activa_media/Potencia_aparente);
```

Esta parte do código calcula as potências ativa, reativa e aparente e o fator de potência aplicando uma função cujo valor seguinte resulta da soma entre o valor atual e o anterior, ambos afetados por pesos. Este método serve para filtrar algum ruído (para além daquele que já foi filtrado pelo *hardware*).

O código completo encontra-se no anexo Anexo F.

3.3 Módulo de Controlo de Cargas Elétricas Trifásicas

Nesta subsecção é feita uma descrição do módulo que é responsável por comutar as cargas trifásicas. Este módulo efetua a ligação ON/OFF da carga a jusante, sempre que a unidade de processamento queira.

Na Figura 3.11 é apresentado o circuito que é utilizado para controlar o relé [22].

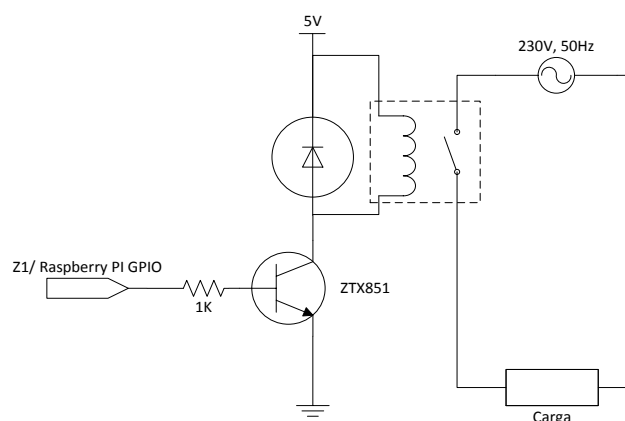


Figura 3.11 – Circuito de controlo do relé. Fonte: [22].

O transístor ZTX851 é utilizado para controlar a tensão através da bobina com o sinal na base. Assim, comandando o sinal na base do transístor, controlamos o estado de funcionamento da carga. A resistência que se encontra na base do transístor serve para limitar a corrente na base do mesmo, e o díodo em anti-paralelo, com a parte indutiva do relé, serve para proteger o transístor dos picos de tensão quando se desliga o relé. A placa de circuito impresso (PCB, *Printed Circuit Board*) que foi utilizada para controlar as cargas trifásicas é a que se encontra na Figura 3.12.

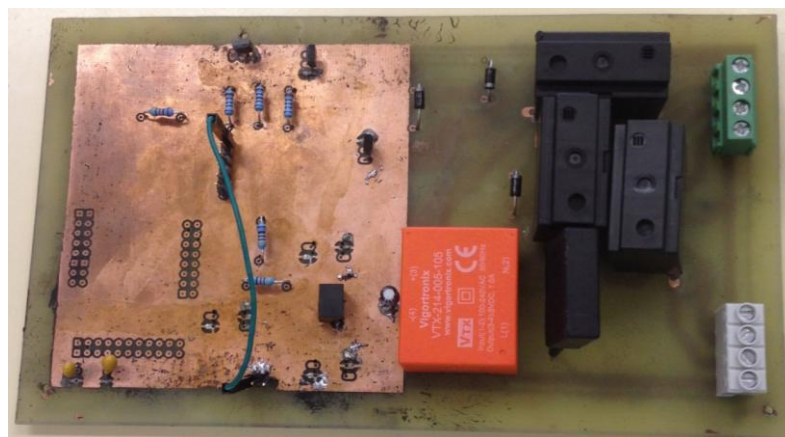


Figura 3.12 – Placa que controla os consumos elétricos em tomadas trifásicas.

O desenho do PCB da Figura 3.12 é o que se encontra na Figura 3.13.

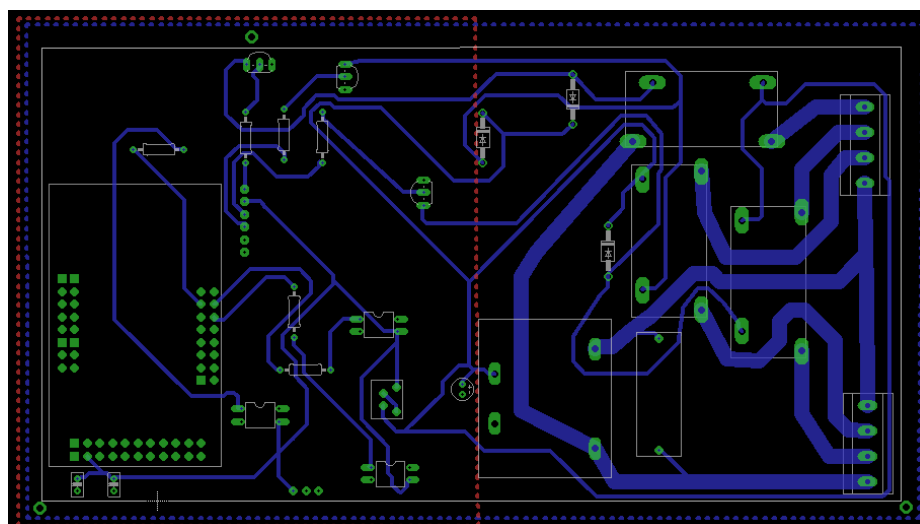


Figura 3.13 – Desenho da placa que controla os consumos elétricos em tomadas trifásicas.

3.4 Módulo de Aquisição de Dados Ambientais

Para além das placas de medição de consumos em quadros elétricos e controlo de cargas elétricas trifásicas, desenvolveu-se uma placa que permitiu a medição de dados ambientais, como a temperatura e a radiação. A Figura 3.14 representa o circuito de aquisição e de transmissão dos dados na fase de desenvolvimento.

Os componentes *XBee* suportam a transmissão dos dados desde o *XBee* remoto até ao *XBee* central. Acoplado ao *XBee* remoto estão um sensor de temperatura (LM35DZ), um

painel solar, que fará de sensor de radiação solar, um *LED* de sinalização do estado do transmissor, para verificação da sincronização entre os *XBee*, e um conversor DC-DC, para a alimentação do transmissor remoto, que reduz a tensão de 5V na sua entrada para 3,3V na sua saída. O desenho da placa de circuito impresso encontra-se representado na Figura 3.15. O *XBee* central encontra-se ligado ao *Raspberry Pi*.

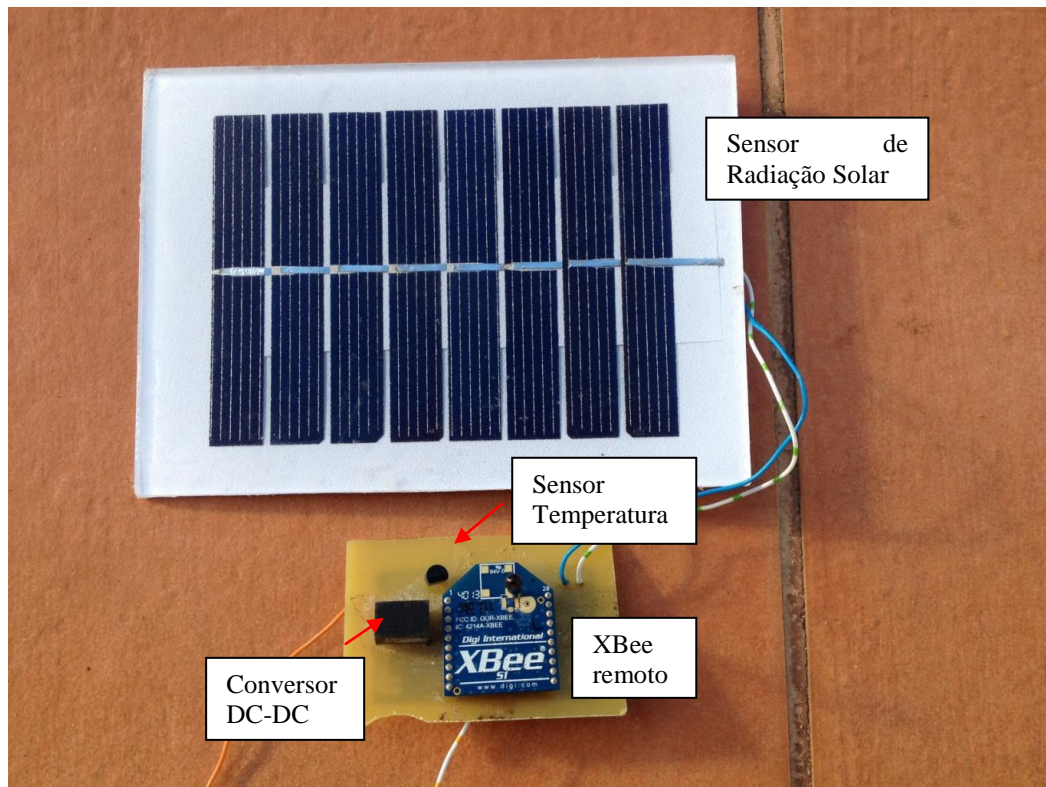


Figura 3.14 – Circuito de aquisição dos dados ambientais.

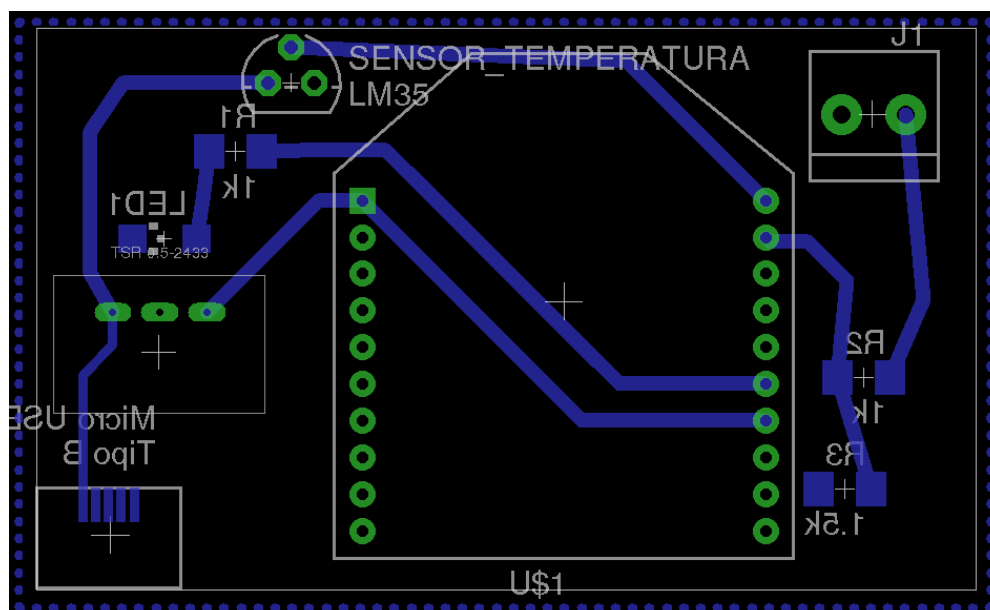


Figura 3.15 – PCB da placa de aquisição de dados ambientais.

Durante os testes para a análise do funcionamento desta placa, verificou-se que o *LED* de sinalização parava de piscar ao fim de 5 segundos, permanecendo aceso. Testes aprofundados apresentaram a ausência do transmissor na rede utilizada após os 5 segundos e até nova sincronização.

O problema residia nas pistas que passam por baixo do transmissor, que provocavam campos magnéticos que interferiam com o funcionamento do mesmo. Assim, o *PCB* final é o que se encontra na figura em baixo apresentando-se o resultado na Figura 3.16. Este *PCB* está completamente operacional resolvendo, assim, o problema surgido.

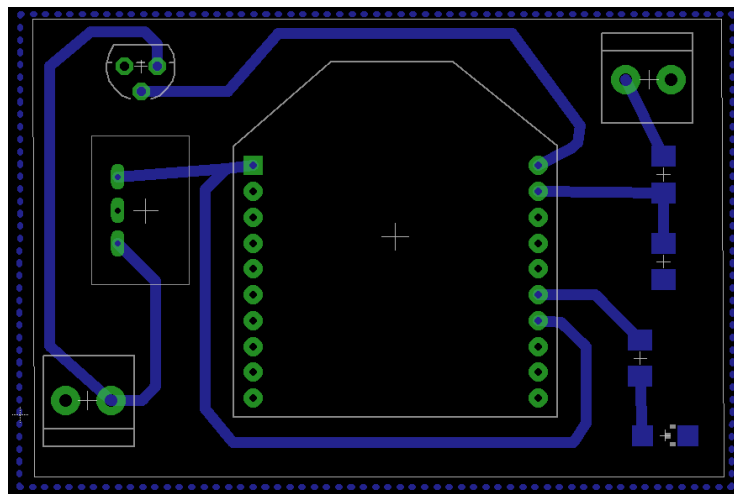


Figura 3.16 – *PCB* final do módulo de aquisição de dados.

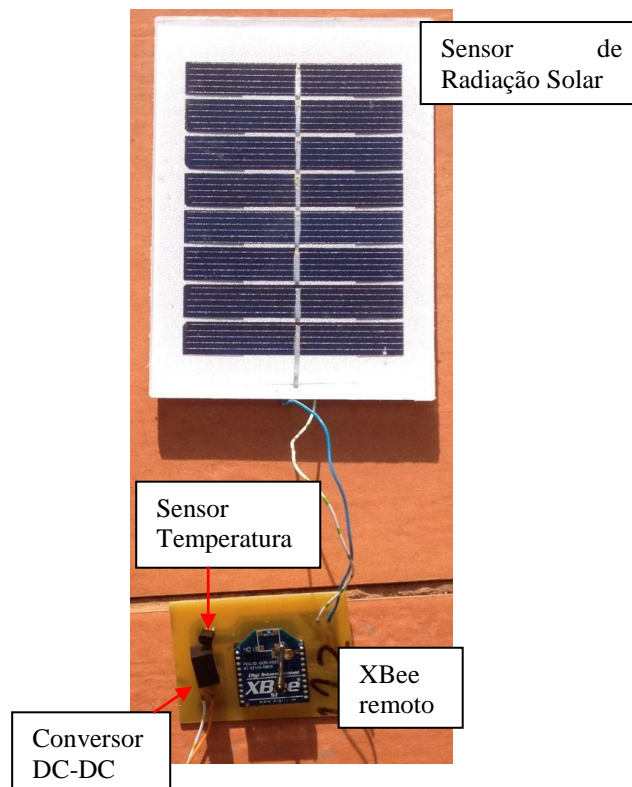


Figura 3.17 – Módulo de aquisição dos dados ambientais.

A implementação física da placa da Figura 3.16 encontra-se na Figura 3.17.

Em primeiro lugar, o programa, desenvolvido em Python, que opera no *Raspberry Pi*, faz um pedido de dados ao *XBee* remoto através da função (Anexo G):

```
dados = xbee.wait_read_frame()
```

Para adquirir os valores da temperatura e da radiação, o programa extrai os valores que se encontram nas variáveis `adc-0` e `adc-1`, respetivamente, através da função (Anexo G):

```
Temperatura_decimal = float(dados['samples'][0]['adc-0'])
Tensao_saida_adc_1_decimal = float(dados['samples'][0]['adc-1'])
```

Estes valores representam a temperatura e a tensão à saída de um divisor de tensão, em decimal, pelo que é necessário converter para °C (no caso da temperatura) e W/m² (no caso da radiação). A utilização de um divisor de tensão para efetuar a leitura da tensão do painel, advém da possibilidade da tensão aos terminais do painel exceder a tolerância máxima do transmissor. Assim, os valores nas unidades correspondentes são calculados através das fórmulas (Anexo G):

```
Temperatura_volts = Temperatura_decimal*3.3/1023
Temperatura_celsius = Temperatura_volts/0.01

Tensao_saida_adc_1_volts = Tensao_saida_adc_1_decimal*3.3/1023
Tensao_painel = (2.5/1.5)*Tensao_saida_adc_1_volts
Corrente_painel = (Tensao_painel-Tensao_saida_adc_1_volts)/1000
Radiacao = 4.348*pow(10,3)*Corrente_painel+4.348*pow(1, (-3))*(exp(Tensao_painel/0.389)-1)
```

Depois de obtidos os valores, estes são enviados para a plataforma *Emoncms* através do seguinte código (Anexo G):

```
_Temperatura_celsius = str(Temperatura_celsius)
_Radiacao = str(Radiacao)

urllib2.urlopen("http://emoncms.org/input/post.json?json={Temperatura:"+_Temperatura_celsius+"&apikey="+apikey)

urllib2.urlopen("http://emoncms.org/input/post.json?json={Radiacao:"+_Radiacao+"&apikey="+apikey)
```

O código completo encontra-se no Anexo G.

Os resultados obtidos pelo módulo encontram-se na Tabela 3.2.

Número do teste	Temperatura (°C)	Radiação (W/m ²)
1	19,03	3,7
2	19,03	2,93
3	19,03	2,41
4	19,03	3,25
5	19,03	3,03
6	19,03	2,62
7	20,32	1,31
8	20,97	1,93
9	20,32	1,75
10	19,35	1,66

Tabela 3.2 – Medição das variáveis ambientais.

Infelizmente não foi possível saber se os valores de radiação estavam corretos, porque não tínhamos um sensor de referência que medisse a radiação em W/m², mas os valores referentes à temperatura estavam corretos.

3.5 Apresentação das Características do Sistema Trifásico

Enquanto são obtidos os dados ambientais, o Módulo de Medição dos Consumos Elétricos Trifásicos calcula as tensões, correntes e potências trifásicas, e ainda o fator de potência. O diagrama de blocos que ilustra o processo de medição e cálculo das características do sistema trifásico encontra-se na Figura 3.18.

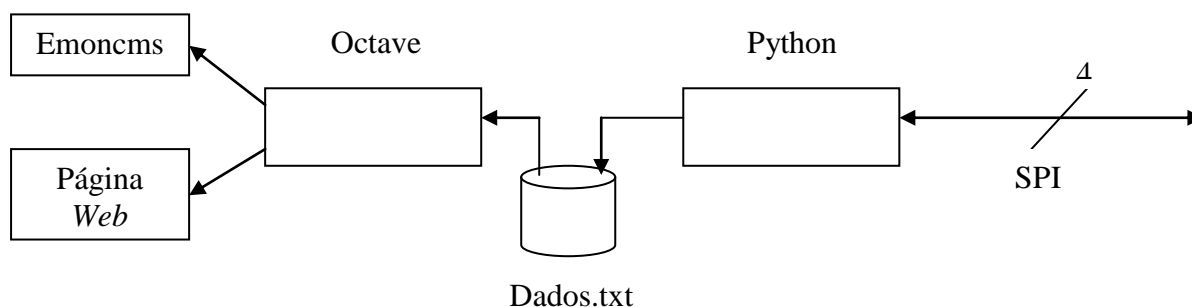


Figura 3.18 – Diagrama de blocos do processo de medição e cálculo das características do sistema trifásico.

O código em Octave é o responsável pela medição das potências ativa, reativa e aparente, e ainda do fator de potência. O programa começa por executar o código em Python,

cujo objetivo é fazer a recolha dos dados do *ADC*, convertê-los em tensão e escrever os resultados num ficheiro chamado *Dados.txt*. Terminada a execução do programa, o Octave abre o ficheiro onde foram escritos os dados e guarda-os em variáveis locais, tendo o cuidado de ignorar as linhas em branco. Depois, usando um fator multiplicativo previamente calculado, o programa calcula quais as tensões e correntes na rede utilizando uma função cujo valor seguinte resulta da soma entre o valor atual e o anterior, ambos afetados por pesos. A seguir, o código faz uma transformação de sistemas de coordenadas, passando de um sistema 1,2,3 para um sistema $\alpha, \beta, 0$. Este código repete-se infinitamente (Anexo F). Os primeiros 10 comandos *system* servem para enviar os dados para a plataforma *Emoncms* [23] da *Open Energy Monitor*, que coloca os dados recebidos em *inputs* (Anexo F). A seguir, no *site* da *Emoncms*, o utilizador tem de associar esses *inputs* a *feeds*, que por sua vez servirão de referência aos *dashboards* que se queiram criar, como por exemplo o que se encontra na Figura 3.19.



Figura 3.19 – Dashboard recetor dos valores calculados.

Os restantes comandos *system* servem para criar uma página *web* (fora do *Emoncms* e presente na Figura 3.20) com os mesmos dados, utilizando um modelo inicial, isto porque o comando *sed* não altera o *template*, criando uma cópia baseada nesse *template*, mas com a *string* substituída (Anexo F).

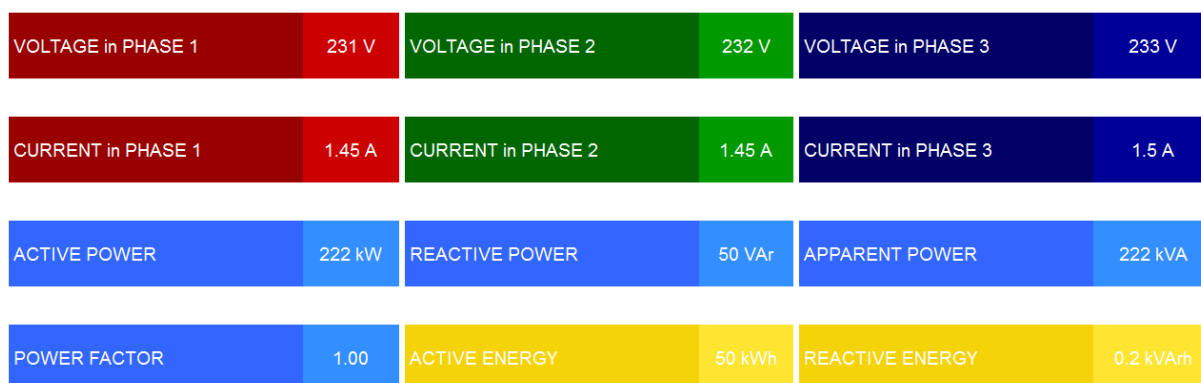


Figura 3.20 – Template da página web.

Os resultados da medição dos diversos parâmetros definidos são apresentados na Tabela 3.3.

Aparelho	Potência	Parâmetro	Parâmetros medidos		Erro absoluto	Erro relativo
			Rasp. PI	Wattímetro		
Sem carga	0 VA	Vrms	240,28	240,986	-0,706	-0,2938%
		Irms	0,31969	0,003	0,31669	99,0616%
Resistências	1380 VA	Vrms	239,31	240,0622	-0,7522	-0,3143%
		Irms	2,3006	2,21	0,0906	3,9381%
		P	1582,5	1576	6,5	0,4107%
		Q	37,057	16,9741	20,0829	54,1946%
		S	1582,9	1576,1	6,3	0,3981%
		Pf	0,99976	0,9999	-0,00014	-0,0140%
Resistências	2760 VA	Vrms	238,93	238,157	0,773	0,3235%
		Irms	4,4744	4,39	0,0844	1,8863%
		P	3080,2	3105	-24,8	-0,8051%
		Q	88,751	46,7654	41,9856	47,3072%
		S	3085,8	3105,4	-19,6	-0,6352%
		Pf	0,99882	0,9999	-0,00108	-0,1081%
Resistências+Bobinas	1020 VA	Vrms	242,74	239,1385	3,6015	1,4837%
		Irms	1,7663	1,478	0,2883	16,3223%
		P	713,28	730,9	-17,62	-2,4703%
		Q	725,65	722,7848	2,8652	0,3948%
		S	1012,6	1027,9	-15,3	-1,5110%
		Pf	0,7031	0,7111	-0,008	-1,1378%

Tabela 3.3 – Medições de potências.

Pela análise dos resultados obtidos, verifica-se que, quando a carga é resistiva, os erros da potência reativa são elevados porque a resolução do ADC não é muito elevada e o ruído

existente na corrente é alto. Com isto, fica difícil obter um valor próximo do 0 absoluto através da fórmula do cálculo da potência reativa.

$$\begin{aligned} Potencia_reativa = & Potencia_reativa \times (1 - Constante) - \\ & - Constante \times (Tensao_alfa \times Corrente_beta + Tensao_beta \times Corrente_alfa) \end{aligned} \quad (27)$$

Os erros verificados em potências baixas surgem, fundamentalmente, de erros na medição das correntes. Assim, para melhorar esta precisão, pode-se optar por utilizar um transformador de corrente de melhor precisão, que não seja de núcleo interrompido, um transformador de melhor qualidade, ou utilizar um sensor de corrente de efeito de *Hall* com maior precisão.

Uma vez que esta placa foi desenhada para fazer medições de potências em quadros elétricos trifásicos, onde as potências são normalmente mais elevadas, consideramos que os resultados obtidos são aceitáveis. Isto não impede que se possa optar, no futuro, por utilizar um transformador de corrente que não seja de núcleo interrompido e que não implique alterações significativas, nem ao esquema do circuito, nem ao algoritmo.

3.6 Proposta de um novo protótipo da tomada elétrica trifásica

Depois de realizados os testes, verificou-se que o *PCB* da Figura 3.13 não pode servir como protótipo final porque o isolamento entre as pistas a 400 V e os planos de massa não são suficientes para evitar o aparecimento de arcos elétricos, comprometendo assim a instalação, a parte responsável pelo transporte de energia não tem capacidade de conduzir as potências que são utilizadas nas instalações trifásicas e, devido a um desvio no alinhamento entre as camadas superior e inferior, existe a possibilidade de, ao menor desvio de algum componente, existir um curto-circuito.

Como na fase final do projeto não houve tempo de produzir outro protótipo físico, apresenta-se aqui o projeto de um novo protótipo. Para este novo protótipo, sugeriu-se as seguintes alterações:

- Substituição dos relés e dos respetivos diodos por transístores de potência;
- Colocação dos transístores de potência no circuito principal e não na *PCB*;

- Remoção dos optocopladores;
- Utilizar o mesmo sinal de comando para os 3 transístores.

Implementando as alterações descritas, obteve-se o seguinte desenho para o *PCB*.

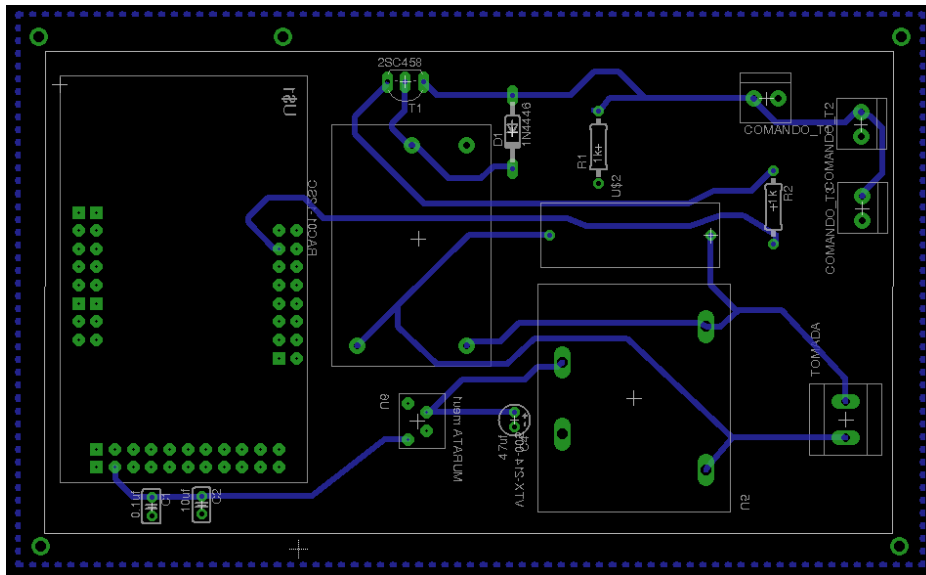


Figura 3.21 – Proposta alternativa de uma tomada elétrica trifásica.

As razões que levaram a implementar as alterações foram:

- Os transístores de potência são mais compactos do que os relés e servem para o mesmo efeito. Isto implica que se tenha de dimensionar os dissipadores de calor. Estes transístores devem ser colocados em série com o circuito a comandar, podendo-se ligar e desligar a carga sem a preocupação da capacidade de condução da potência por parte da placa de circuito impresso.
- A inserção de um mesmo sinal de comando para os 3 transístores de potência serve para eliminar ao máximo o atraso da abertura/fecho de transístores de fases diferentes, minimizando as correntes de circulação entre fases.

Comparando a Figura 3.13 com a Figura 3.21, conclui-se que não só se retira a dependência de uma tomada trifásica, como se reduz bastante o tamanho do *PCB*, mantendo a sua funcionalidade. Depois de alguma pesquisa, concluiu-se que os transístores a utilizar

seriam os *Insulated-Gate Bipolar Transistor* (IGBT), porque podem conduzir correntes elevadas tendo uma queda de tensão em condução baixa. Contudo, teve que ser adicionado um novo conversor AC-DC RAC01-12SC da RECOM POWER, uma vez que, para conduzir correntes elevadas, a tensão entre a gate e o emissor tinha que ser maior do que aquela que o conversor da VIGORTRONIX disponibilizava. Concluiu-se, ainda, que devem ser utilizados 2 transístores por fase para que quando os transístores estejam à condução, estes possam dissipar o calor sem queimar. Esta ideia teve que ser implementada por força da informação fornecida pelos *datasheets* analisados. Devido à ausência de um símbolo e de um pacote no *software Eagle* [24] do conversor AC-DC, houve a necessidade de se implementar uma biblioteca para o conversor referido anteriormente. Essa biblioteca encontra-se no Anexo C.k.

3.7 Módulo demonstrador

Na fase final do projeto, os módulos de medição dos consumos elétricos trifásicos e de controlo de cargas trifásicas encontravam-se em funcionamento. As Figura 3.22 e Figura 3.23 representam respetivamente os módulos de medição de consumos elétricos e de controlo de cargas trifásicas, nas suas versões finais.



Figura 3.22 – Módulo de Medição dos Consumos Elétricos Trifásicos.



Figura 3.23 – Módulo de Controlo de Cargas Elétricas Trifásicas.

Capítulo 4 Mecanismo de Detecção de Consumos Anómalos

No capítulo anterior foram feitos vários módulos de medição e cálculo das características do sistema trifásico. No entanto, para detetar anomalias nos consumos elétricos, estes módulos só por si não chegam. É necessário utilizar um algoritmo que permita, em função dos dados históricos, efetuar as previsões de consumo, para depois, em função da discrepância entre o consumo real e o estimado, detetar as situações de alarme. Neste capítulo será feita uma explicação dos procedimentos adotados para fazer as previsões recorrendo a redes *Bayesianas*.

As redes *Bayesianas* convencionais representam as probabilidades de uma forma estática, não variante no tempo. Para os modelos em que as probabilidades variam, têm que se utilizar as Redes *Bayesianas* Dinâmicas (RBD).

As RBD foram apresentadas pela primeira vez por Dean e Kanazawa em 1988 [25], como uma extensão das redes *Bayesianas* [26]. Neste sentido, uma RBD permite utilizar um modelo gráfico probabilístico para descrever o nível de incerteza numa variedade de aplicações. As RBD permitem também a análise de fenómenos complexos e o apoio à decisão em situações altamente variáveis e interligadas [25] ou quando os dados são ambíguos.

Nas RBD, o momento temporal das variáveis é apresentado por um conjunto de variáveis aleatórias $X_t = X_{1t} \dots X_{nt}$. Se o sistema depender apenas do estado imediatamente anterior, ele é denominado de cadeia de *Markov* de 1ª ordem, com a sua distribuição dada por [25]:

$$P(X^t | X^{t-1}) = \prod_{i=1}^n P(X_i^t | X_{pai}(X_i^t)) \quad (28)$$

onde $X_{pai}(X_i^t)$ representa o nó antecessor de X_i^t . Neste capítulo iremos utilizar este tipo de RBD, representado por uma cadeia de *Markov* de 1ª ordem, apresentada na Figura 4.1. Nesta figura, os nós mais escuros indicam os dados X_i^t e os restantes os estimadores.

A deteção de consumo anómalo faz-se pela comparação entre o consumo efetivamente medido (no período de um quarto de hora) com aquele que é obtido pelo modelo (i.e. o estimado). O consumo estimado é obtido a partir de dois estimadores, identificados na Figura 4.1 como Estimadores de Consumo A e B.

O Estimador A utiliza como dados de entrada o dia da semana e hora atuais, não entrando em linha de conta com a temperatura exterior e com a radiação solar (29). No caso do Estimador B para prever o consumo, utilizam-se os dados: do dia da semana, da hora do dia, da temperatura exterior e da radiação solar (30).

$$w = \text{matriz}(A, \text{diadasemana}, \text{horadodia} - 1, :) \quad (29)$$

$$w = \text{matriz}(B, \text{diadasemana}, \text{horadodia} - 1, \text{indicetemperatura}, \text{indiceradicao}, :) \quad (30)$$

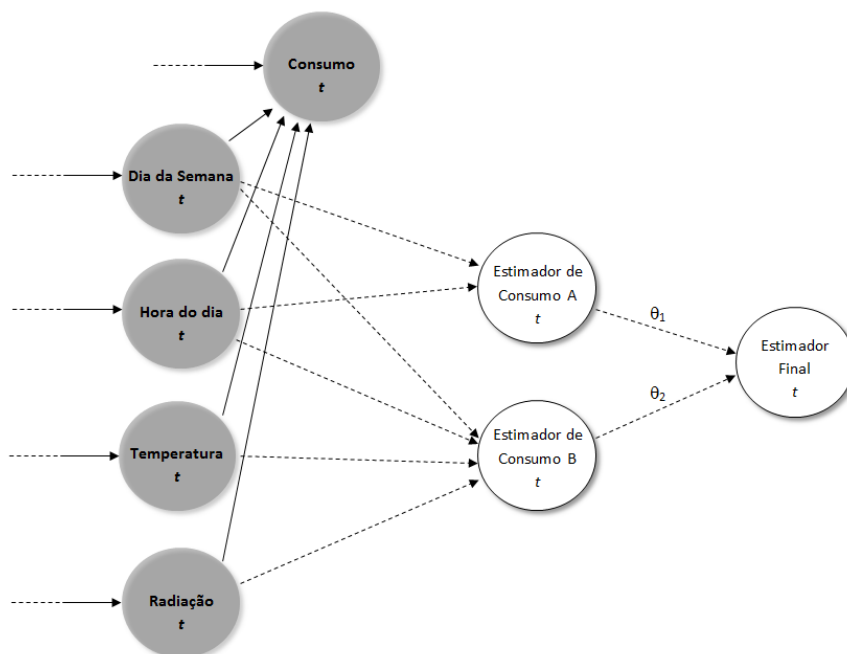


Figura 4.1 – Esquema da Rede Bayesiana Dinâmica utilizada.

A previsão de consumo faz-se com base na distribuição de probabilidades de consumo verificadas até então, para as mesmas condições de entrada (dia da semana e hora do dia; temperatura exterior e radiação solar).

O resultado apresentado pelo Estimador Final dá preferência ao estimador B pois este considera mais variáveis. No entanto, e uma vez que para efetuar a estimação de consumo se utilizam dados do passado, nem sempre o estimador B tem todos os dados que necessita. Sempre que tal acontece recorre-se ao Estimador A que, sendo menos esparso, consegue fazer previsões ao final de uma semana.

Segue-se a descrição das variáveis utilizadas no modelo:

- Dia da Semana: um valor inteiro a variar entre 1 e 7 que traduz o dia da semana em que nos encontramos, onde o valor 1 corresponde ao domingo.
- Hora do Dia: um valor inteiro entre 1 e 24 que traduz a hora do dia em que nos encontramos, onde o valor 1 corresponde ao período entre as 0 e a 1 hora.
- Temperatura: um valor discreto que reflete a temperatura exterior, fragmentado em intervalos de 5°C, onde o valor 1 representa o intervalo entre -10°C a -5°C, o valor 2 representa o intervalo entre -5°C a 0°C, etc.
- Radiação: um valor discreto que reflete a radiação solar, fragmentado em intervalos de 500 W/m², em que o valor 1 representa a radiação entre 0 e 500 W/m², etc.

Neste caso, para que o histograma de consumos traduza uma distribuição de probabilidades normal, o processo de discretização de consumos é acompanhado por uma conversão de escala, traduzido pela equação (31).

$$\text{Índice_de_Consumo} = 1 + \text{round}(P_{\text{consumida}}^{1/3}) \quad (31)$$

A aproximação do histograma de consumos a uma distribuição normal permite utilizar a média aritmética dos valores medidos como método de *Maximum Likelihood Estimation*. A equação (31) introduz também um processo de discretização não linear que permite dar uma maior precisão a valores baixos de consumo, traduzindo-se por isso no final em menores erros percentuais quando as grandezas medidas são baixas.

De acordo com as características da distribuição normal, 95,4% dos valores medidos devem estar a uma distância de dois desvios padrão (σ) da média. Assim, dado um nível de consumo medido, produz-se a geração de um alarme sempre que o mesmo esteja a mais que δ vezes o desvio padrão dos consumos anteriores, em que δ pode ser ajustado ao longo do funcionamento do equipamento.

Ao fim de uma semana de operação, o Estimador de Consumo A começa a reportar dados de estimação, pois tem em consideração apenas os mesmos dias da semana e o mesmo intervalo horário da(s) semana(s) anterior(es). Em relação ao estimador B, este só efetua estimações quando verificou no passado as mesmas condições de radiação e temperatura exterior, de acordo com os intervalos acima descritos.

4.1 Implementação

Neste subcapítulo é feita uma explicação das partes mais importantes do código implementado em Octave.

Partindo das bases apresentadas no início do capítulo, foram criadas duas matrizes, A e B referentes ao Estimador de Consumo A e ao Estimador de Consumo B, respetivamente, sendo que as dimensões têm em linha de conta o número de entradas e de saídas que cada estimador utiliza. Durante a aprendizagem, cada estimador converte os dados presentes no ficheiro “DadosOlhao.txt” em índices e depois adiciona uma unidade de consumo no índice correspondente (retirado do Anexo H).

```
indicetemperatura=2 + ceil(temperatura/5);  
indiceradiacao=ceil(radiacao/500)+1;
```

Para melhor compreender esta ideia, vamos supor que o Estimador B verifica que, às 15h:45min do dia 7 de Setembro de 2014, com a radiação de 900 W/m^2 e a temperatura a 23°C , existiu um consumo de 12000 W . Em primeiro lugar, o código converte o dia da semana, a hora do dia, a radiação, a temperatura e o consumo em índices (neste caso, a radiação corresponde ao índice 3, a temperatura ao índice 7 e o consumo ao índice 24). Por fim, adiciona mais uma unidade de consumo no ponto $matrizB(1,12+1,7,3,24)$. No final, o ponto com o número maior é aquele que mais vezes se repetiu na fase de aprendizagem.

Na fase de previsão o algoritmo começa por verificar se ambos Estimadores indicam previsões. Se sim, o valor estimado corresponde ao Estimador B. Caso haja apenas um Estimador que indica previsões, o valor final corresponde ao valor indicado por esse Estimador. Pode também acontecer que não hajam previsões por parte dos Estimadores. A forma como é executada a previsão é a seguir apresentada (Anexo H).

```
if (isempty(za) && ~isempty(zb))  
    Estimativa=EstimativaB;  
end  
if (~isempty(za) && isempty(zb))  
    Estimativa=EstimativaA;  
end
```

4.2 Avaliação de Resultados

Para efetuar a avaliação dos resultados, utilizaram-se dados de medição de consumo do edifício da Biblioteca Municipal de Olhão, obtidos no período compreendido entre 8 de abril

de 2014 e 31 de Janeiro de 2015, e intercalados por períodos de 15 minutos. Em seguida descrevemos cada um desses resultados para o modelo assente em redes *Bayesianas Dinâmicas*.

A Tabela 4.1 apresenta os resultados obtidos para o modelo da Rede *Bayesiana Dinâmica*, apresentada na Tabela 4.1. Abreviações utilizadas:

- MSE: Média da Soma dos Erros Quadrados:
$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y} - y)^2$$
- MRE: Média da Soma dos Erros Relativos:
$$MRE\% = \frac{100}{N} \sum_{i=1}^N \left| \frac{\hat{y} - y}{y} \right|$$

Estimador de Consumo	Dados de 8/Abr/2014 a 31/Out/2014		Dados de 01/Nov./2014 a 31/Jan./ 2015	
	MSE (Wh)	MRE %	MSE (Wh)	MRE%
A	$7,8 \times 10^{10}$	27,6	$4,5 \times 10^6$	32,7
B	$5,8 \times 10^{10}$	28,3	$2,0 \times 10^6$	21,9
Final	$8,1 \times 10^{10}$	27,8	$2,5 \times 10^6$	22,7

Tabela 4.1 – Resultados obtidos para a Rede *Bayesiana Dinâmica*.

Os resultados obtidos mostram que este modelo consegue estimar com um erro final de 22,7% o consumo da instalação. Quando comparamos os Estimadores de Consumo A e B, verificamos que o B consegue obter um erro menor (21,9%). No entanto, apesar de ser mais preciso, para efetuar previsões requer que condições semelhantes de temperatura exterior e radiação se tenham verificado no passado, o que nem sempre acontece. Já o estimador de consumo A, não sendo tão preciso, consegue sempre devolver estimativas após uma semana de monitorização de consumos.

Como se pode observar, as soluções obtidas nos dois casos são distintas na quantidade de parâmetros que usam, mas idênticas em termos de especificações de avaliação. A diferença relevante está na complexidade dos modelos e nos instantes passados que são necessários guardar em memória para implementar tais modelos. Note-se que o pior caso obriga a usar 79 parâmetros por entrada e um atraso de 30 minutos ($\tau = 2$) entre amostras. Em relação ao valor de MRE, em %, dos dados de teste, de sublinhar que é um valor idêntico ao obtido pelos melhores modelos das restantes abordagens em discussão.

A Figura 4.2 mostra as comparações entre o consumo real e estimado antes e depois de novembro.

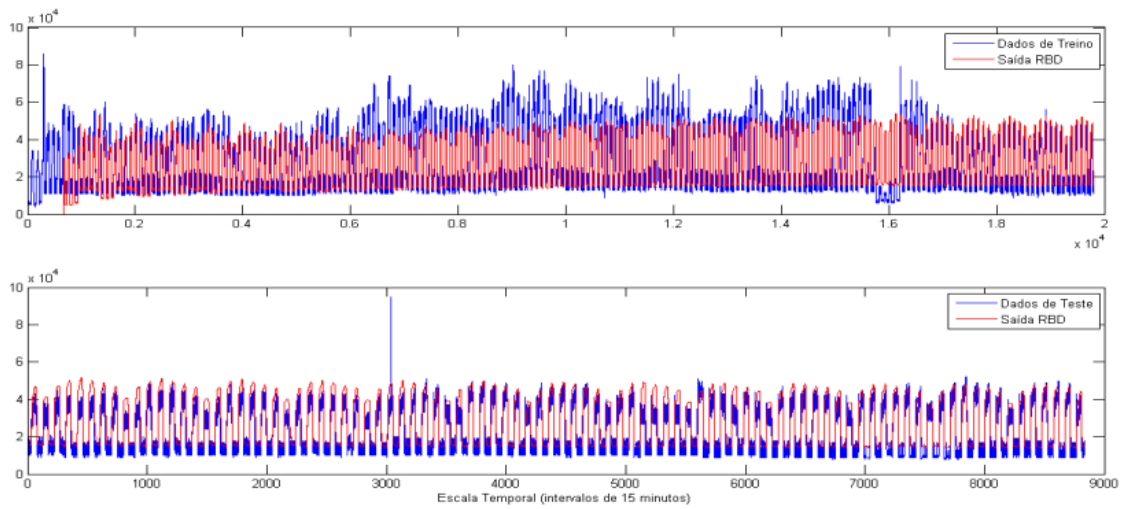


Figura 4.2 – Comparação entre consumo real e estimado, utilizando a Rede Bayesiana Dinâmica, em que o gráfico superior corresponde à fase anterior a novembro de 2014 e a parte inferior à fase de posterior.

Capítulo 5 Conclusões

O trabalho realizado no decorrer desta dissertação compreende duas grandes partes: a criação de um sistema para medição e monitorização remota (das potências trifásicas e dados ambientais), e o desenvolvimento de um mecanismo de deteção de consumos anómalos.

Em relação ao sistema trifásico, verifica-se que existem erros relativos elevados nas medições para correntes no sistema baixas. Isto acontece porque o ruído é superior ao sinal que se queria analisar e porque o sensor de corrente utilizado suporta medições de até 100 A. Este problema pode ser resolvido utilizando um sensor de corrente que não seja de núcleo interrompido, ou substituindo o sensor por um outro cujo limite seja inferior a 100 A. No entanto, os resultados podem-se considerar positivos, porque a média dos erros, no caso de resistências puras (1380 W), foi de 7.99% e, no caso dos circuitos com resistências e bobinas (1020 VA), foi de 2.18%. Dado que as instalações trifásicas têm potências elevadas, os erros relativos tendem a diminuir sempre que as medidas aumentam. Os módulos implementados já estão em funcionamento, à exceção da proposta do novo módulo de controlo da carga trifásica que ainda não foi feita, uma vez que o projeto entretanto terminou.

Apesar dos resultados serem satisfatórios, há aspetos nas placas de circuito impresso que podem ser melhorados: a inserção de dispositivos de proteção em todos os módulos; a implementação do novo módulo de comando das cargas trifásicas (apresentado no subcapítulo 3.6); a substituição dos transformadores monofásicos de 2 saídas por transformadores de uma só saída; a substituição das resistências e condensadores do tipo *through hole* pelo tipo *SMD (Surface Mount Device)*, no caso do módulo de monitorização dos consumos trifásicos; substituição dos transmissores *XBee* série 1 por *XBee* série 2, porque o primeiro não implementa o *ZigBee*; e a substituição dos fios de alimentação por um conector *micro-USB* ou pilhas, a fim de se eliminar a utilização de uma fonte de alimentação exterior dedicada de difícil disponibilidade no mercado, no caso do módulo de aquisição dos dados ambientais. Este módulo ainda não está inserido num contexto real, mas pretende-se adicionar ao módulo demonstrador, para que se possa implementar o algoritmo de previsão dos consumos na UAlg.

Em relação aos resultados relativos pelo algoritmo de estimação de consumos elétricos, estes foram relativamente bons (22.7%), uma vez que a variabilidade dos consumos é considerável. Considera-se ainda que este algoritmo pode sofrer uma melhoria se outros fatores forem considerados, como por exemplo a ocupação do edifício e/ou se alterar a estrutura da rede inicialmente considerada. Antes de se aplicar este algoritmo na UAlg, pretende-se inserir as melhorias já anteriormente referidas, analisar os resultados e, posteriormente, retirar conclusões.

5.1 Disseminação

O trabalho desenvolvido nesta dissertação foi parcialmente incluído no Projeto QREN *Managing The Intelligence* (QREN-30260) refletido no relatório:

- Relatório de Milestone 6.2, PARTE II - Implementação da placa "Controlo e Medição de Potências Consumidas Trifásicas"- Objeto Inteligente nº2, 30-06-2015.

No que diz respeito à disseminação foi também publicado o artigo:

- Costa, H., Cabrita, C., Monteiro, J., Semião, J., “Mecanismo de Detecção de Consumos Anómalos em Redes Energéticas Inteligentes”, 14ª Conferência sobre Redes de Computadores (CRC2015), Évora, 19-20 Novembro, 2015.

Bibliografia

- [1] Santos, Paulo. **Sistema Automático de Medida Usando o Sistema GSM**, Escola Superior de Tecnologia, Universidade do Algarve, 2001.
- [2] Espinosa-Faller, Francisco J. e Rendón-Rodríguez, Guillermo E. **A ZigBee Wireless Sensor Network for Monitoring an Aquaculture Recirculating System**, Universidad Marista de Mérida, México, 2012.
- [3] Telegesis. <http://www.telegesis.com/casestudy/temperature-monitoring-control-with-zigbee/>. Acedido a 27/01/2015.
- [4] IEEE Xplore.
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6341902&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6341902. Acedido a 10/10/2014.
- [5] Matos, Tomé Emanuel Rodrigues. **Remote Vital Signs Monitoring: Instrumentation Module**, Faculdade Ciências e Tecnologia, Coimbra, 2008.
- [6] Moreno, Ivo Daniel G. **Redes de Sensores Inteligentes Sem Fios Para a Supervisão de Parques de Reservatórios Industriais**, Faculdade de Ciências e Tecnologia, Coimbra, 2009.
- [7] Intelligent Steet Lighting Using a ZigBee Network of Devices and Sensors. <http://www.grin.com/en/e-book/213749/intelligent-street-lighting-using-a-zigbee-network-of-devices-and-sensors>. Acedido a 11/10/2014.
- [8] Landeiro, João Pedro Rosado. **Aeromodelo Telecomandado Controlado Remotamente Através da Internet Para Caracterização do Ambiente Através de Imagens e Vídeo**, Faculdade Ciências e Tecnologia, UAlg, 2012.
- [9] Ferreira, Paulo Filipe Loureiro. **Sistema de Controlo e Monitorização Remota de uma Unidade Industrial de PSA**, Faculdade de Engenharia, Porto, 2010.
- [10] Nest, site oficial. <https://nest.com/>. Acedido a 15/01/2015.
- [11] Google patentes. <http://www.google.com/patents/WO2013058820A1?cl=en>. Acedido a 19/01/2015.
- [12] ZigBee Alliance, site oficial. <http://zigbee.org/zigbee-for-developers/>. Acedido a 09/01/2015.
- [13] Saleiro, Mário e Ey, Emanuel. **Introdução ao ZigBee – Uma Abordagem Prática**, Escola Superior de Tecnologia, Universidade do Algarve.
- [14] Protocolo 6LoWPAN. <https://en.wikipedia.org/wiki/6LoWPAN>. Acedido a 17/09/2015.
- [15] I., Ben-Gal. **Bayesian Networks**, 2007.
- [16] Expectation – Maximization.
https://en.wikipedia.org/wiki/Expectation%20%80%93maximization_algorithm.
Acedido a 18/09/2015.
- [17] MCMC. https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo. Acedido a 18/09/2015.
- [18] Cadeias de Markov. http://en.wikipedia.org/wiki/Markov_chain. Acedido a 23/03/2015.

- [19] Zhu, Chun e Sheng, Weihua. *Realtime Recognition of Complex Daily Activities Using Dynamic Bayesian Network*, 2011.
- [20] Nazerfard, Ehsan e Cook, Diane J. *Using Bayesian Networks for Daily Activity Prediction*. In: Proceedings of the AAAI Workshop on Plan, Activity and Intent Recognition, pp 32–38, 2013.
- [21] Algoritmo de Viterbi. http://en.wikipedia.org/wiki/Viterbi_algorithm. Acedido a 23/03/2015.
- [22] Nugin, Sergey. **Relatório Milestone revisto**, 2015.
- [23] Emoncms, site oficial. <http://emoncms.org/>. Acedido a 11/09/2015.
- [24] Eagle, site oficial. <http://www.cadsoftusa.com/>. Acedido a 11/09/2015.
- [25] T. Dean e K. Kanazawa, **Probabilistic temporal reasoning**, 1988.
- [26] J. Pearl, **Probabilistic reasoning in intelligent systems: networks of plausible inference**. Morgan Kaufmann Publishers Inc., 1988.
- [27] Monteiro, Jânio Miguel Evangelista Ferreira. **Redes e Protocolos Internet**, capítulo 1, Instituto Superior de Engenharia, Universidade do Algarve.
- [28] Standard IEEE 802.15.4. http://en.wikipedia.org/wiki/IEEE_802.15.4. Acedido a 17/10/2014.
- [29] Open Energy Monitor, site oficial. <http://openenergymonitor.org/emon/>. Acedido a 11/09/2015.
- [30] Tecdomo. <http://www.tecdomo.pt/domotica.htm>. Acedido a 11/10/2014.
- [31] Neobit. <http://www.neobit.pt/>. Acedido a 11/10/2014.
- [32] ZigBee overview. <https://docs.zigbee.org/zigbee-docs/dcn/07/docs-07-5482-01-0mwg-zigbee-alliance-overview.pdf>. Acedido a 22/01/2015.
- [33] Areal Energia, site oficial. <http://www.areal-energia.pt/>. Acedido a 13/07/2015.
- [34] Previsão. <https://en.wikipedia.org/wiki/Forecasting>. Acedido a 20/07/2015.
- [35] Introdução ao Raspberry Pi B+. <http://www.rs-online.com/designspark/electronics/eng/blog/introducing-the-raspberry-pi-b-plus>. Acedido a 24/07/2015.

Anexos

Anexo A. Fluxograma do Algoritmo de Viterbi

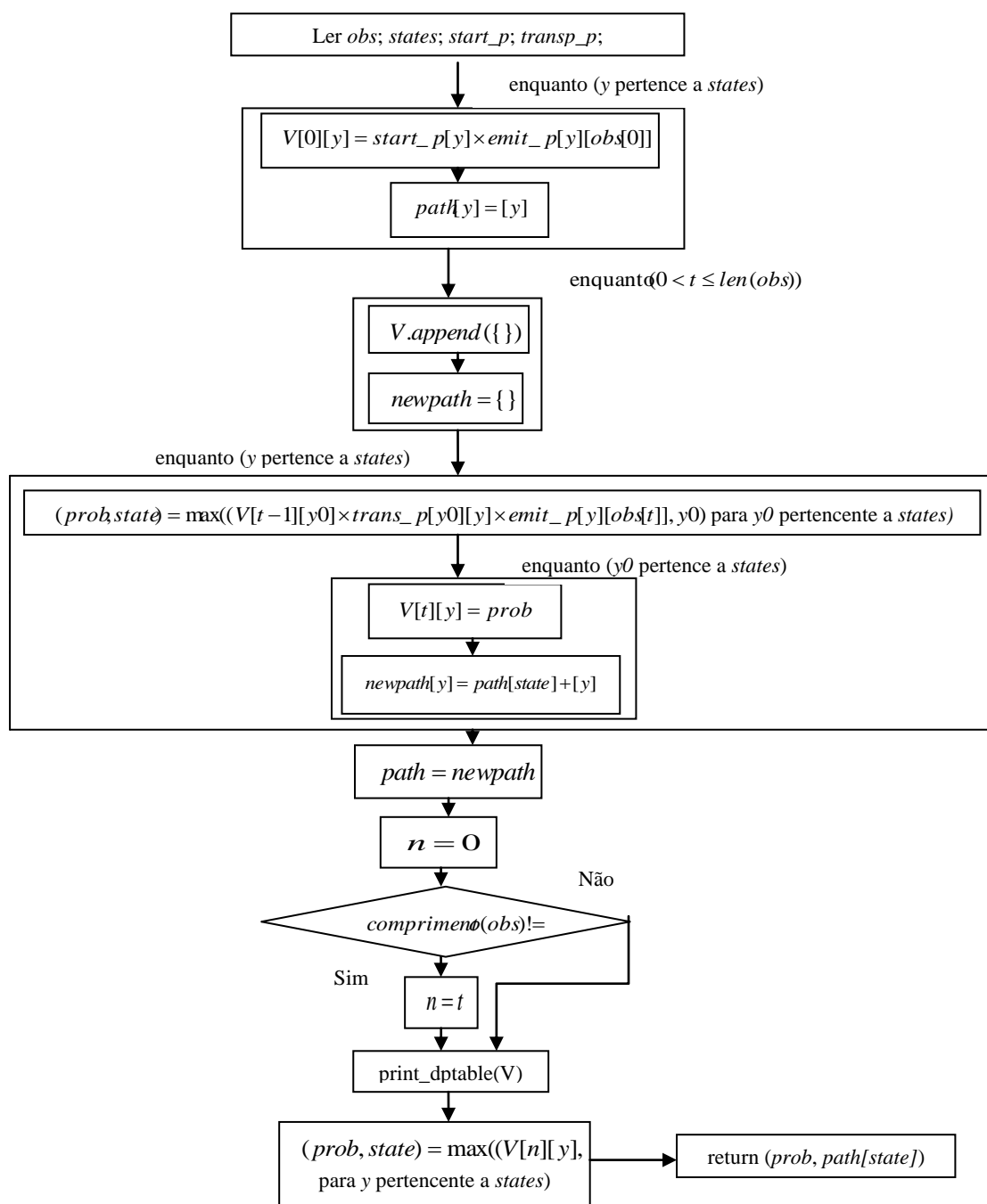


Figura 6.0.1 – Fluxograma da função viterbi para implementação do algoritmo de Viterbi para a análise do estado de saúde do paciente. Fonte: [21].

Anexo B. O modelo OSI

O modelo OSI [27] é uma arquitetura geral de como são feitas as comunicações dentro do mesmo dispositivo, ou entre dispositivos diferentes, e que é composto por 7 camadas, como mostra a Figura 6.0.2.

Camada de Aplicação
Camada de Apresentação
Camada de Sessão
Camada de Transporte
Camada de Rede
Camada de Ligação de Dados
Camada Física

Figura 6.0.2 – O modelo OSI.

A camada Física é a responsável pela transmissão dos bits no meio físico [27]; a camada de Ligação de Dados tem como função garantir a fiabilidade da comunicação num determinado troço da rede, podendo fornecer mecanismos de controlo de fluxo e de erros [27]; a camada de Rede estabelece, mantém e termina conexões e assegura funções de endereçamento e de *routing* [27]; a camada de Transporte implementa a qualidade do serviço, isto é, assegura uma transmissão fiável extremo a extremo, ao mesmo tempo que também cria um mecanismo para controlo de erros [27]; a camada de Sessão faz a sincronização entre as entidades de aplicação comunicantes [27]; a camada de Apresentação converte o formato dos dados num formato adequado, fornecendo, ainda, meios para a utilização de sintaxes que possibilitem a conversão [27] e a camada de Aplicação providencia acesso ao modelo OSI para os utilizadores, através de aplicações como o HTTP [27]. Cada camada mencionada comunica com a camada consecutiva no mesmo sistema terminal ou com a mesma camada de um outro sistema terminal.

As normas IEEE definem as características da camada Física e da camada de Ligação de Dados. A norma IEEE 802.15.4 em particular define que a velocidade de transmissão dos dados a 10 metros é de 250 kbit/s [28], que as frequências usadas são 868, 915 e 2450 MHz [28] e que a rede pode operar nas topologias estrela ou ponto-a-ponto [28].

Anexo C. Componentes utilizados

a. Raspberry Pi modelo B+

O *Raspberry Pi* é um computador baseado num sistema de placa única. A Figura 6.0.3 demonstra o dispositivo.

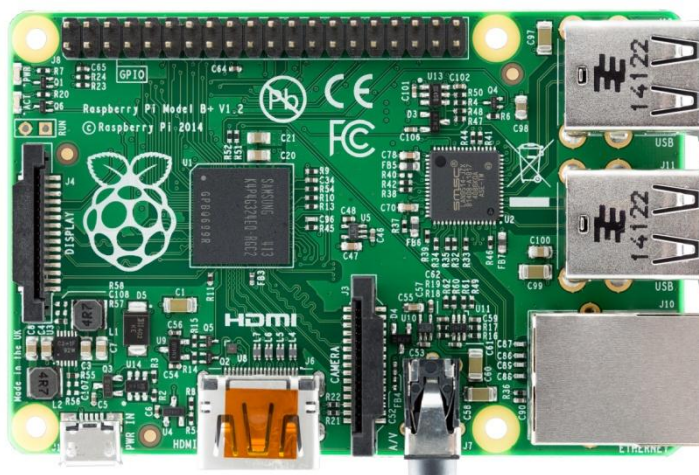


Figura 6.0.3 – *Raspberry Pi* modelo B+.

As características principais do dispositivo são apresentadas na tabela seguinte:

Chip	Broadcom BCM2835 SoC
CPU	700 MHz Low Power ARM1176JZ-F
GPU	Broadcom VideoCore IV
Memory	512MB SDRAM shared with GPU
Network	onboard 10/100 Ethernet RJ45 jack
USB 2.0	4 (Integrated USB hub)
Video Output	HDMI, Composite RCA
Audio Output	3.5mm jack, HDMI
Storage	MicroSD card slot
Peripheral connectors	16xGPIO, UART, I ² C bus, SPI bus
Power rating	700 mA (3.5 W)
Power source	5V via MicroUSB or GPIO header
Dimensions	8.6cm x 5.4cm x 1.7cm

Tabela 6.0.1 – Características do *Raspberry Pi* modelo B+.

A Figura 6.0.4 representa o *pinout* deste dispositivo.

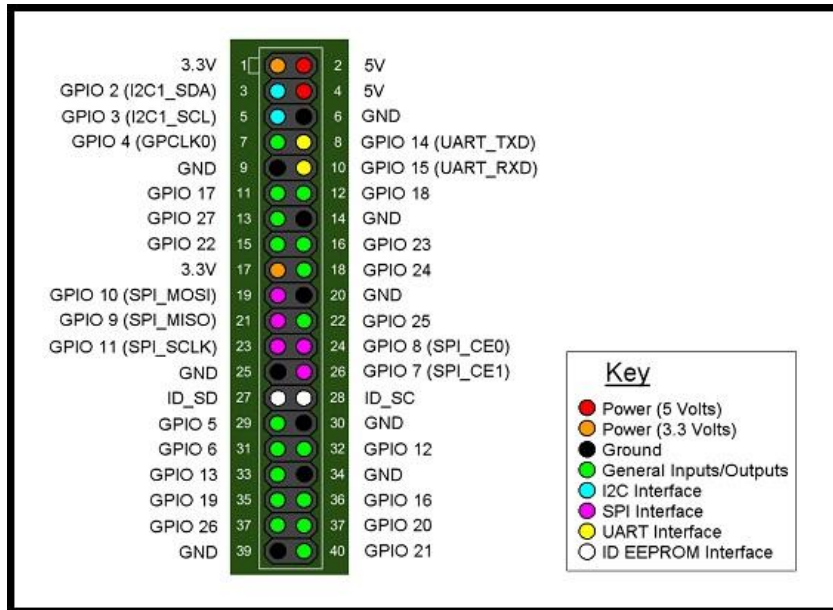


Figura 6.0.4 – Pinout do Raspberry Pi modelo B+.

b. Conversor AC-DC VIGORTRONIX – VTX-214-010-105

A Figura 6.0.5 apresenta o conversor AC-DC utilizado no Módulo de Medição dos Consumos.



Figura 6.0.5 – Conversor AC-DC.

Este conversor permite alimentar o circuito DC a partir da tensão alternada da rede, fornecendo uma tensão contínua com a amplitude de 5 V e uma corrente máxima de 2 A. As características principais são apresentadas na Tabela 6.0.2.

INPUT Pin 1: L Pin 2: N	Voltage Range	90 ~ 264VAC / 47 ~ 63Hz (120VDC ~ 370VDC)
	Current	250mA Max (100VAC) / 150mA Max (240VAC)
	Inrush Current	25A Max (100VAC) / 35A Max (240VAC)
	No Load power	0.3W Max
	Leakage Current	0.3 mA Max
OUTPUT Pin 3: DC+ Pin 4: DC-	Voltage Accuracy	3% +/-
	Ripple	<50mV Max (Vp-p)
	Noise	<1% Vout Max (Vp-p)
	Efficiency	>70%
	Protection	Over Power, Over Voltage & Short Circuit Protection

Tabela 6.0.2 – Características do conversor AC-DC.

c. Sensor de corrente SCT-013-000

Este sensor contém um transformador de corrente que pode ser montado sem a interrupção do condutor. A sua ilustração encontra-se na Figura 6.0.6.



Figura 6.0.6 – SCT-013-000.

A *Open Energy Monitor* [29] fez os testes para as medições do primário e do secundário do sensor, e os resultados obtidos foram os que seguem na tabela seguinte.

Primary Turns (Amps per turn = 5.04)	Primary Current (A)	Measured Sec current (No.1) (mA)	Measured Sec current (No.2) (mA)	Design Sec current (mA)	Error (No.1) (%)	Error (No.2) (%)
0	0	0	0	0	0	0
2	10.08	4.98	4.88	5.04	-1.19	-3.17
4	20.16	9.87	9.91	10.08	-1.09	-1.69
6	30.24	15.00	14.85	15.12	-0.79	-1.79
8	40.32	19.87	19.94	20.16	-1.44	-1.09
10	50.40	24.8	24.8	25.2	-1.59	-1.59
12	60.48	29.9	29.5	30.24	-1.12	-2.45
14	70.56	35.0	34.9	35.28	-0.79	-1.08
16	80.64	40.1	39.5	40.32	-0.55	-2.03
18	90.72	45.0	44.9	45.36	-0.79	-1.01
20	100.8	49.4	49.8	50.40	-1.98	-1.19

Tabela 6.0.3 – Resultados obtidos nas medições das correntes no transformador de corrente.

Traduzindo esta tabela num gráfico, o resultado é:

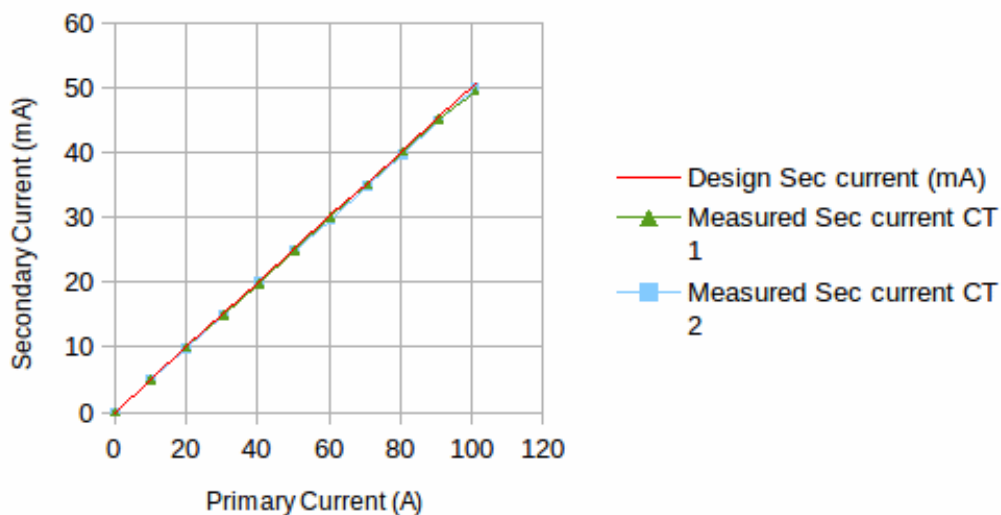


Figura 6.0.7 – Gráfico dos resultados obtidos.

A ligação interna do transformador de corrente encontra-se na Figura 6.0.8.

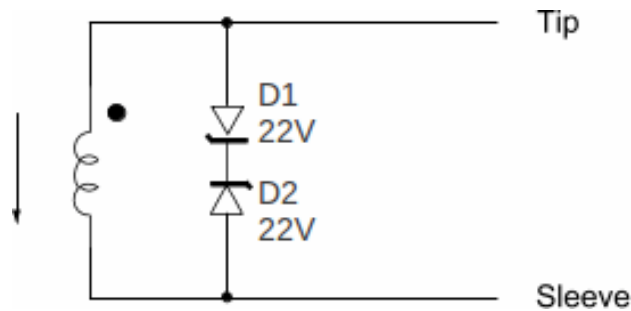


Figura 6.0.8 – Ligação interna do SCT-013-000

d. Conversor Analógico-Digital (ADC) MCP3008

O MCP3008 da Microchip é um conversor analógico-digital de 8 canais que suporta os modos *single-ended* e *pseudo-differential*. O pinout está representado na Figura 6.0.9

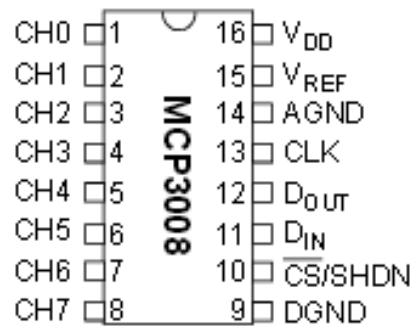


Figura 6.0.9 – Pinout do MCP3008.

As características principais do MCP3008 são apresentadas na tabela seguinte:

Operating voltage	2.7-5.5 V
Throughput Rate	200 ksps for Vdd=Vref=5 V
	75 ksps for Vdd=Vref=2.7 V
Resolution	10 bit
Input Voltage Range for CH0 or CH1 in Single-Ended Mode	Vss...Vdd V
Input Voltage Range for IN+ in Pseudo-Differential Mode	IN-...Vdd+IN-
Input Voltage Range for IN- in Pseudo-Differential Mode	Vss-100...Vss+100 mV
Clock Frequency	3.6 MHz for Vdd=5 V
	1.35 MHz for Vdd=2.7 V

Tabela 6.0.4 – Caraterísticas do MCP3008.

e. Zolertia Z1

O Zolertia Z1 é um módulo sem fios de baixo consumo que serve para implementar uma rede de sensores sem fios (em inglês Wireless Sensor Networks - WSN). A Figura 6.0.10 apresenta as capacidades de expansão do dispositivo.

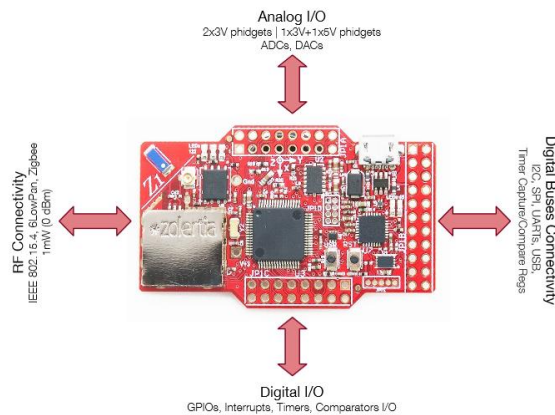


Figura 6.0.10 – Zolertia Z1.

f. Relé Omron G4A

A Figura 6.0.11 apresenta o Relé Omron G4A.



Figura 6.0.11 – Relé G4A.

As características principais do Relé Omron G4A são apresentadas na tabela seguinte.

Max. switching current	20 A
Max. switching voltage	250 VAC
Coil rated voltage	5 VDC
Coil rated current	180 mA
Coil resistance	27.8 Ω

Tabela 6.0.5 – Características do relé Omron G4A.

g. Sensor de temperatura LM35DZ

Parâmetro	Condições de teste	LM35D			Unidade
		Típico	Limite testado	Limite teórico	
Precisão	$T_A=25^{\circ}\text{C}$	± 0.6	± 1.5		$^{\circ}\text{C}$
	$T_A=T_{\text{max}} (150^{\circ}\text{C})$	± 0.9		± 2	
	$T_A=T_{\text{min}} (1.5^{\circ}\text{C})$	± 0.9		± 2	
Não linearidade	$T_{\text{min}} \leq T_A \leq T_{\text{max}},$ $-40^{\circ}\text{C} \leq T_j \leq 125^{\circ}\text{C}$	± 0.2		± 0.5	$^{\circ}\text{C}$
Ganho	$T_{\text{min}} \leq T_A \leq T_{\text{max}},$ $-40^{\circ}\text{C} \leq T_j \leq 125^{\circ}\text{C}$	10		9.8	$\text{mV}/^{\circ}\text{C}$

Tabela 6.0.6 – Características do sensor de temperatura.

h. Painel solar MC-SP0.8-NF-GCS

Potência máxima (Wp)	0.8
Tensão à potência máxima (V)	3.85
Corrente à potência máxima (A)	0.21
Tensão de circuito aberto (V)	4.8
Corrente de curto-circuito (A)	0.23

Tabela 6.0.7 – Características do painel solar.

i. Conversor DC-DC TSR 0.5-2433

Tensão entrada (V)	4.75 – 32
Tensão de saída (V)	3.3
Corrente saída máxima (A)	0.5

Tabela 6.0.8 – Características do conversor DC-DC.

j. XBee série 1

Símbolo	Característica	Condição	Mínimo	Típico	Máximo	Unidade
V_{IL}	Tensão de entrada a '0'	Todas as entradas digitais	-	-	$0.35 \cdot V_{CC}$	V
V_{IH}	Tensão de entrada a '1'	Todas as entradas digitais	$0.7 \cdot V_{CC}$	-	-	V
V_{OL}	Tensão de saída a '0'	$I_{OL}=2\text{mA}, V_{CC} \geq 2.7\text{V}$	-	-	0.5	V
V_{OH}	Tensão de saída a '1'	$I_{OH}=-2\text{mA}, V_{CC} \geq 2.7\text{V}$	$V_{CC}-0.5$	-	-	V
I_{IN}	Corrente de fuga na entrada	$V_{IN}=V_{CC}$ ou GND, pinos como <i>inputs</i> , por pino	-	0.025	1	μA
TX	Corrente na transmissão	$V_{CC}=3.3\text{V}$	-	45	-	mA
RX	Corrente na recepção	$V_{CC}=3.3\text{V}$	-	50	-	mA
PWR-DWN	Corrente em poupança de energia	Parâmetro SM=1	-	<10	-	μA

Tabela 6.0.9 – Características do XBee série 1.

k. AC-DC RAC01-12SC

Tensão entrada (VAC)	90-277
Frequência (Hz)	40-63
Tensão saída (VDC)	12
Corrente saída (mA)	83

Tabela 6.0.10 – Características do retificador RAC01-12SC.

Anexo D. Biblioteca do Conversor AC-DC RAC01-12SC da RECOM POWER

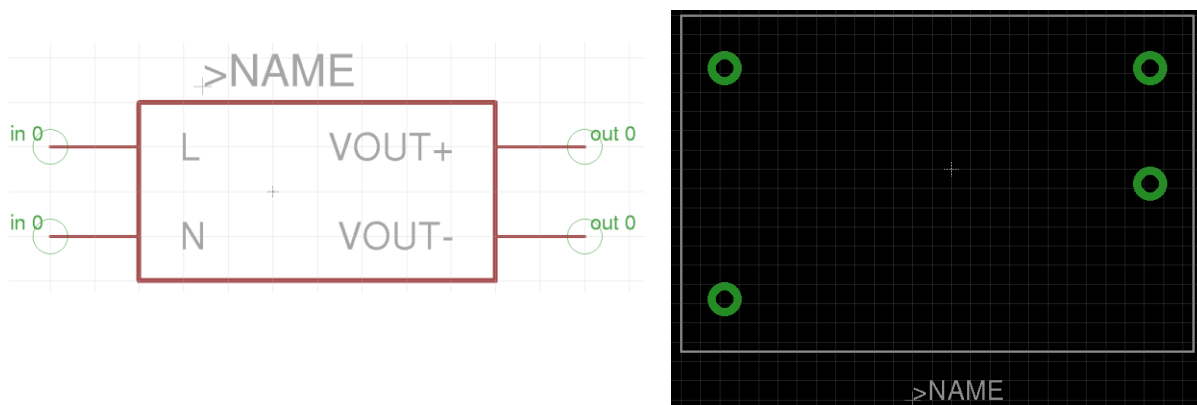


Figura 6.0.12 – Biblioteca do conversor AC-DC RAC01-12SC da RECOM POWER.

Anexo E. Código Para Obter as Tensões e as Correntes Trifásicas

```
import spidev

spi=spidev.SpiDev()
spi.open(0,0)
spi.max.speed.hz=1650000

v1=[]
i1=[]
v2=[]
i2=[]
v3=[]
i3=[]

for x in range(1,200)
    canal=0
    resp_adc=spi.xfer2([1,(8+canal)<<4,0])
    adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
```

```
v1_volts=adc_out*3.3/1023
v1.append(v1_volts)

canal=1
resp_adc=spi.xfer2([1, (8+canal)<<4,0])
    adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
i1_volts=adc_out*3.3/1023
i1.append(i1_volts)

canal=2
resp_adc=spi.xfer2([1, (8+canal)<<4,0])
    adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
v2_volts=adc_out*3.3/1023
v2.append(v2_volts)

canal=3
resp_adc=spi.xfer2([1, (8+canal)<<4,0])
    adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
i2_volts=adc_out*3.3/1023
i2.append(i2_volts)

canal=4
resp_adc=spi.xfer2([1, (8+canal)<<4,0])
    adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
v3_volts=adc_out*3.3/1023
v3.append(v3_volts)

canal=5
resp_adc=spi.xfer2([1, (8+canal)<<4,0])
    adc_out=((resp_adc[1]&3<<8)+resp_adc[2])
i3_volts=adc_out*3.3/1023
i3.append(i3_volts)

print v1
print "\n"
print i1
print "\n"
print v2
print "\n"
print i2
print "\n"
print v3
print "\n"
print i3
spi.close()
```

Anexo F. Código Para Calcular as Potências e o Fator de Potência Trifásico

```
constante=0.2;

V1_ef=0;
V2_ef=0;
V3_ef=0;

I1_ef=0;
I2_ef=0;
I3_ef=0;
```

```
Potencia_activa=0;
Potencia_reactiva=0;
Potencia_aparente=0;
Factor_potencia=0;

while 1
    for i=1:20
        system('sudo python Leitura_SPI.py > Dados.txt');

        FP=fopen('Dados.txt','rt');

        linha=fgetl(FP);
        while isempty(linha)
            linha=fgetl(FP);
        end
        V1=sscanf(linha(2:length(linha-1)),'%f,');

        linha=fgetl(FP);
        while isempty(linha)
            linha=fgetl(FP);
        end
        I1=sscanf(linha(2:length(linha-1)),'%f,');

        linha=fgetl(FP);
        while isempty(linha)
            linha=fgetl(FP);
        end
        V2=sscanf(linha(2:length(linha-1)),'%f,');

        linha=fgetl(FP);
        while isempty(linha)
            linha=fgetl(FP);
        end
        I2=sscanf(linha(2:length(linha-1)),'%f,');

        linha=fgetl(FP);
        while isempty(linha)
            linha=fgetl(FP);
        end
        V3=sscanf(linha(2:length(linha-1)),'%f,');

        linha=fgetl(FP);
        while isempty(linha)
            linha=fgetl(FP);
        end
        I3=sscanf(linha(2:length(linha-1)),'%f,');

        fclose(FP);

        tensao_v1=V1';
        tensao_v2=V2';
        tensao_v3=V3';

        corrente_i1=I1';
        corrente_i2=I2';
        corrente_i3=I3';

        t=0:length(tensao_v1)-1;

        medio_V1=mean(tensao_v1);
        tensao_V1=270.607.*(tensao_v1-medio_V1);
```

```

        V1_ef=V1_ef*(1-
constante)+max(tensao_V1)/sqrt(2)*constante;

        medio_V2=mean(tensao_v2);
        tensao_V2=270.607.*(tensao_v2-medio_V2);
        V2_ef=V2_ef*(1-
constante)+max(tensao_V2)/sqrt(2)*constante;

        medio_V3=mean(tensao_v3);
        tensao_V3=270.607.*(tensao_v3-medio_V3);
        V3_ef=V3_ef*(1-
constante)+max(tensao_V3)/sqrt(2)*constante;

        medio_I1=mean(corrente_i1);
        corrente_i1_AC=corrente_i1-medio_I1;
        corrente_I1=196.364.*corrente_i1_AC;
        I1_ef=I1_ef*(1-
constante)+max(corrente_I1)/sqrt(2)*constante;

        medio_I2=mean(corrente_i2);
        corrente_i2_AC=corrente_i2-medio_I2;
        corrente_I2=196.364.*corrente_i2_AC;
        I2_ef=I2_ef*(1-
constante)+max(corrente_I2)/sqrt(2)*constante;

        medio_I3=mean(corrente_i3);
        corrente_i3_AC=corrente_i3-medio_I3;
        corrente_I3=196.364.*corrente_i3_AC;
        I3_ef=I3_ef*(1-
constante)+max(corrente_I3)/sqrt(2)*constante;

        M_concordia=sqrt(2/3)*[1 -1/2 -1/2;0 sqrt(3)/2 -
sqrt(3)/2;1/sqrt(2) 1/sqrt(2) 1/sqrt(2)]; %Matriz de Concordia

        Tensao_concordia=M_concordia*[tensao_V1;tensao_V2;tensao_V3];
        Corrente_concordia=M_concordia*[corrente_I1;corrente_I2;corrente_I3
];

        %Eliminar a onda que não interessa (Gama)
        Tensao_alfa=Tensao_concordia(1,1:length(Tensao_concordia));
        Tensao_beta=Tensao_concordia(2,1:length(Tensao_concordia));

        Corrente_alfa=Corrente_concordia(1,1:length(Corrente_concordia));

        Corrente_beta=Corrente_concordia(2,1:length(Corrente_concordia));

        %Cálculo das potências activa, reactiva, aparente e factor de
potência
        Potencia_activa=Potencia_activa*(1-constante)-
constante*(Tensao_alfa.*Corrente_alfa+Tensao_beta.*Corrente_beta);
        Potencia_reactiva=Potencia_reactiva*(1-constante)-
constante*(-Tensao_alfa.*Corrente_beta+Tensao_beta.*Corrente_alfa);
        Potencia_activa_media=mean(Potencia_activa);
        Potencia_reactiva_media=mean(Potencia_reactiva);
        Potencia_aparente=Potencia_aparente*(1-
constante)+constante*(sqrt(Potencia_activa_media^2+Potencia_reactiva
_media^2));

        Factor_potencia=Factor_potencia*(1-
constante)+constante*(Potencia_activa_media/Potencia_aparente);

        FP=fopen('Potencias','w');

```

```

        _V1_ef=num2str(V1_ef, 5);
        _V2_ef=num2str(V2_ef, 5);
        _V3_ef=num2str(V3_ef, 5);
        _I1_ef=num2str(I1_ef, 5);
        _I2_ef=num2str(I2_ef, 5);
        _I3_ef=num2str(I3_ef, 5);
        _Potencia_aparente=num2str(Potencia_aparente, 5);

    _Potencia_activa_media=num2str(Potencia_activa_media, 5);

    _Potencia_reactiva_media=num2str(Potencia_reactiva_media, 5);
    _Factor_potencia=num2str(Factor_potencia, 5);
    json=strcat("{\"I1_rms\": \"\", _I1_ef, \"\", \"I2_rms\":
\", _I2_ef, \"\", \"I3_rms\": \"\", _I3_ef, \"\", \"V1_rms\": \"\", _V1_ef, \"\",
\"V2_rms\": \"\", _V2_ef, \"\", \"V3_rms\": \"\", _V3_ef, \"\", \"desf\": \"x\",
\"S\": \"\", _Potencia_aparente, \"\", \"P\": \"\", _Potencia_activa_media,
\", \"Q\": \"\", _Potencia_reactiva_media, \"\", \"PF\": \"\",
_Factor_potencia, \"\"}");
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp Potencias /var/www');

    FP=fopen('V1_ef.json', 'w');
    json=strcat("{\"V1_rms\": \"\", _V1_ef, \"\"}");
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp V1_ef.json
/var/www/measure/phase1/voltage');

    FP=fopen('V2_ef.json', 'w');
    json=strcat("{\"V2_rms\": \"\", _V2_ef, \"\"}");
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp V2_ef.json
/var/www/measure/phase2/voltage');

    FP=fopen('V3_ef.json', 'w');
    json=strcat("{\"V3_rms\": \"\", _V3_ef, \"\"}");
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp V3_ef.json
/var/www/measure/phase3/voltage');

    FP=fopen('I1_ef.json', 'w');
    json=strcat("{\"I1_rms\": \"\", _I1_ef, \"\"}");
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp I1_ef.json
/var/www/measure/phase1/current');

    FP=fopen('I2_ef.json', 'w');
    json=strcat("{\"I2_rms\": \"\", _I2_ef, \"\"}");
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp I2_ef.json
/var/www/measure/phase2/current');

```

```

        FP=fopen('I3_ef.json', 'w');
        json=strcat('{"I3_rms": "', _I3_ef, '"}');
        fprintf(FP, json);
        fclose(FP);

        system('sudo cp I3_ef
/var/www/measure/phase3/current');

        FP=fopen('apparent_power.json', 'w');
        json=strcat('{"S": "', _Potencia_aparente, '"}');
        fprintf(FP, json);
        fclose(FP);

        system('sudo cp apparent_power.json
/var/www/measure/threephasepower/apparent_power');

        FP=fopen('active_power.json', 'w');
        json=strcat('{"P": "', _Potencia_activa_media,
''}');
        fprintf(FP, json);
        fclose(FP);

        system('sudo cp active_power.json
/var/www/measure/threephasepower/active_power');

        FP=fopen('reactive_power.json', 'w');
        json=strcat('{"Q": "', _Potencia_reactiva_media,
''}');
        fprintf(FP, json);
        fclose(FP);

        system('sudo cp reactive_power.json
/var/www/measure/threephasepower/reactive_power');

        FP=fopen('power_factor.json', 'w');
        json=strcat('{"PF": "', _Factor_potencia, '"}');
        fprintf(FP, json);
        fclose(FP);

        system('sudo cp power_factor.json
/var/www/measure/threephasepower/power_factor');
        end

disp('Valor eficaz da Tensão V1')
V1_ef
disp('Valor eficaz da Tensão V2')
V2_ef
disp('Valor eficaz da Tensão V3')
V3_ef
disp('Valor eficaz da Corrente I1')
I1_ef
disp('Valor eficaz da Corrente I2')
I2_ef
disp('Valor eficaz da Corrente I3')
I3_ef

Potencia_activa_trifasica=mean(Potencia_activa);
Potencia_reactiva_trifasica=mean(Potencia_reactiva);

if Potencia_activa_trifasica<0
    disp('Valor eficaz da Potência Activa Gerada')

```

```

        Potencia_activa=mean(Potencia_activa)
    else
        disp('Valor eficaz da Potência Activa Consumida')
        Potencia_activa=mean(Potencia_activa)
    end

    if Potencia_reactiva_trifasica<0
        disp('Valor eficaz da Potência Reactiva Capacitiva')
        Potencia_reactiva=mean(Potencia_reactiva)
    else
        disp('Valor eficaz da Potência Reactiva Indutiva')
        Potencia_reactiva=mean(Potencia_reactiva)
    end

    disp('Valor eficaz da Potência Aparente')
    Potencia_aparente=mean(Potencia_aparente)
    disp('Factor de Potência')
    Factor_potencia=mean(Factor_potencia)

    tempo=cputime()/3600;
    Energia_activa=tempo*Potencia_activa*10^(-3);
    Energia_reactiva=tempo*Potencia_reactiva*10^(-3);

    FP=fopen('active_energy.json', 'w');
    _Energia_activa=num2str(Energia_activa, 10);
    json=strcat('{"Active_Energy": "', _Energia_activa, '"}');
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp active_energy.json
/var/www/measure/threephaseenergy/active_energy');

    FP=fopen('reactive_energy.json', 'w');
    _Energia_reactiva=num2str(Energia_reactiva, 10);
    json=strcat('{"Reactive_Energy": "', _Energia_reactiva, '"}');
    fprintf(FP, json);
    fclose(FP);

    system('sudo cp reactive_energy.json
/var/www/measure/threephaseenergy/reactive_energy');

    comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Tensao
V1:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
',V1_ef);
    system(comando);
    comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Tensao
V2:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
',V2_ef);
    system(comando);
    comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Tensao
V3:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
',V3_ef);
    system(comando);
    comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Corrente
I1:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
',I1_ef);
    system(comando);

```

```

comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Corrente
I2:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
', I2_ef);
system(comando);
comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Corrente
I3:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
', I3_ef);
system(comando);
comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Potencia
Aparente:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
', Potencia_aparente);
system(comando);
comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Potencia
Activa:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
', Potencia_activa_media);
system(comando);
comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Potencia
Reactiva:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
', Potencia_reactiva_media);
system(comando);
comando=sprintf('wget -qO-
"http://emoncms.org/input/post.json?json={Factor
Potencia:%f}&apikey=b821d53574b4b3f7eb243d6175d612a0" > /dev/null
', Factor_potencia);
system(comando);

comando=sprintf("sed -e 's/231/%.2f/g' Valores_Medidos.html >
Valores_Medidos2.html", V1_ef);
system(comando);
comando=sprintf("sed -e 's/232/%.2f/g' Valores_Medidos2.html >
Valores_Medidos3.html", V2_ef);
system(comando);
comando=sprintf("sed -e 's/233/%.2f/g' Valores_Medidos3.html >
Valores_Medidos4.html", V3_ef);
system(comando);
comando=sprintf("sed -e 's/ola/%.2f/g' Valores_Medidos4.html >
Valores_Medidos5.html", I1_ef);
system(comando);
comando=sprintf("sed -e 's/ole/%.2f /g' Valores_Medidos5.html >
Valores_Medidos6.html", I2_ef);
system(comando);
comando=sprintf("sed -e 's/oli/%.2f/g' Valores_Medidos6.html >
Valores_Medidos7.html", I3_ef);
system(comando);
comando=sprintf("sed -e 's/packman/%.2f/g' Valores_Medidos7.html >
Valores_Medidos8.html", Potencia_aparente);
system(comando);
comando=sprintf("sed -e 's/222 kW/%.2f W/g' Valores_Medidos8.html >
Valores_Medidos9.html", Potencia_activa_media);
system(comando);
comando=sprintf("sed -e 's/mario/%.2f/g' Valores_Medidos9.html >
Valores_Medidos10.html", Potencia_reactiva_media);
system(comando);
comando=sprintf("sed -e 's/luigi/%.2f/g' Valores_Medidos10.html >
Valores_Medidos11.html", Factor_potencia);
system(comando);

```

```

comando=sprintf("sed -e 's/xancho/%.2f/g' Valores_Medidos11.html >
Valores_Medidos12.html", Energia_activa);
system(comando);
comando=sprintf("sed -e 's/shazam/%.2f/g' Valores_Medidos12.html >
Valores_Medidos13.html", Energia_reactiva);
system(comando);
system('sudo cp Valores_Medidos13.html /var/www/index.html');
end

```

Anexo G. Código Para a Aquisição Dos Dados Ambientais

```

# -*- coding: utf-8 -*-

import serial
from xbee import XBee
import urllib2
from math import exp, pow

# Configuração da porta série

porta = serial.Serial('/dev/ttyAMA0', baudrate = 9600, timeout =
None, rtscts = False, xonxoff = False)

# Abre a porta UART0
xbee = XBee(porta)

porta.flushInput()

# Número de série do Xbee remoto
numero_serie_remoto = '\x00\x13\xA2\x00\x40\xBA\x37\xC2'

# Apikey do Emoncms
apikey = 'b821d53574b4b3f7eb243d6175d612a0'

# Ler os valores do sensor remoto
while True:
    dados = xbee.wait_read_frame()

# Processamento dos dados
    Temperatura_decimal = float(dados['samples'][0]['adc-0'])
    Luminosidade_decimal = float(dados['samples'][0]['adc-1'])

    Temperatura_volts = Temperatura_decimal*3.3/1023
    Temperatura_celsius = Temperatura_volts/0.01 # Pelo datasheet do
sensor (10mv/°C)

# Obtenção da radiação
    Tensao_saida_adc_1_volts = Tensao_saida_adc_1_decimal*3.3/1023
    Tensao_painel = (2.5/1.5)*Tensao_saida_adc_1_volts
    Corrente_painel = (Tensao_painel-Tensao_saida_adc_1_volts)/1000
    Radiacao = 4.348*pow(10,3)*Corrente_painel+4.348*pow(1,(-
3))* (exp(Tensao_painel/0.389)-1)

# Envio para a Internet
    _Temperatura_celsius = str(Temperatura_celsius)
    _Lux = str(Lux)

```

```

urllib2.urlopen("http://emoncms.org/input/post.json?json={Temperatura:+"_Temperatura_celsius+"}&apikey="+apikey)

urllib2.urlopen("http://emoncms.org/input/post.json?json={Luminosidade:+"_Lux+"}&apikey="+apikey)

filenames = ['/var/www/Cabecalho_HTML.html',
'/var/www/Grafico_Construcao.html', '/var/www/Rodape_HTML.html']
with open('/var/www/templog.html', 'w') as outfile:
    for fname in filenames:
        with open(fname) as infile:
            outfile.write(infile.read())
outfile.close()

```

Anexo H. Código Para Previsão Dos Consumos Elétricos

```

FP=fopen('DadosOlhao.txt','rt');
%FP=fopen('DadosBoliqueime.txt','rt');
%FP=fopen('DadosAlbufeira.txt','rt');
%FP=fopen('DadosLoule.txt','rt');

%FP=fopen('DadosTavira.txt','rt');

%FP=fopen('ProducaoBoliqueime.txt','rt');
FPout=fopen('dados_gradual.txt','wt');
ERROMEDIO=0;
ERRO2MEDIO=0;
LINHAS=0;
EM1=0;

% Lê uma linha do ficheiro
linha=fgetl(FP);
% 7 dias por semana, 24 horas por dia, 15 níveis de temperatura, 6
níveis de radiacao, 50 níveis de consumo
matrizA=zeros(7,24,15,6,50);
matrizA2=zeros(7,24,15,6,50);

% 7 dias por semana, 24 horas por dia, 50 níveis de consumo
matrizB=zeros(7,24,50);
matrizB2=zeros(7,24,50);

PontosA=1;
PontosB=1;

DadosOut1=[];
DadosOut2=[];
while ~feof(FP)
    newline='';
    counter=1;
    % retira espaços no meio da linha
    for i=1:length(linha)
        if linha(i)~=0
            newline(counter)=linha(i);
            counter=counter+1;
        end
    end
end
end

```

```

if ~isempty(newline)
    if length(newline)>1
        % Analisa linha

        % substitui ',' por '.' para carregarmos como floats
        posi=find(newline==',' );
        newline(posi)='.';

        % faz a partição da linha para ler data e restantes
dados
        dados=sscanf(newline,'%d-%d-%d %d:%d %f %f %d');
        if ~isempty(dados)
            if dados(1)<2014
                % uniformiza formatos de ficheiros de txt
                dados=[2014 dados(1:5) ' 0' dados(6:8)];
            end
        end

        if length(dados)==10
            % se é meia noite
            if dados(5)+dados(6)==0
                % copia dados do dia anterior para permitir
fazer previsão
                matrizA2=matrizA;
                matrizB2=matrizB;
            end

            diadasemana=weekday(datestr([dados(4), dados(3), dados(2), 12, 0, 0]));
            horadodia=dados(5);
            radiacao=dados(8);
            temperatura=dados(9);
            Pconsumida=dados(10);
            % Para mostrar dados
            [diadasemana horadodia temperatura Pconsumida];
            indicetemperatura=2 + ceil(temperatura/5);
            if temperatura<0
                disp('Temperatura negativa')
                %break;
            end
            %ESCALOESPOTENCIA=7500;
            ESCALOESPOTENCIA=200;
            indicePconsumida= 1 + round(Pconsumida^(1.0/3.0)); %
soma 1 para evitar indice 0
            indiceradiacao=ceil(radiacao/500)+1;

            %[x, posmaximo]=max([matrizG(diadasemana, horadodia+1, indicetemperatura,
a, :)]);

            % CONSELHEIRO B
            zb=[];

            w=matrizB2(diadasemana, horadodia+1, indicetemperatura, indiceradiacao,
:);

            for i=1:length(w)
                zb=[zb i*ones(1, w(i))];
            end
            if (~isempty(zb))
                EstimativaB=(mean(zb)-1.0)^3.0;
            end

            % CONSELHEIRO A
            za=[];

```

```

w=matrizA2(diadasemana,horadodia+1,:);
for i=1:length(w)
    za=[za i*ones(1,w(i))];
end
if (~isempty(za))
    EstimativaA=(mean(za)-1.0)^3.0;
end

Estimativa=-1;
if (~isempty(za) && ~isempty(zb))

%Estimativa=(PontosA*EstimativaA+PontosB*EstimativaB)/(PontosA+PontosB);

    Estimativa=EstimativaA;
    % atualiza pontos comparando estimativa com
valor real
    if abs(Pconsumida-EstimativaA)<abs(Pconsumida-
EstimativaB)

        PontosA=PontosA+1;
    else
        PontosB=PontosB+1;
    end
end
if (isempty(za) && ~isempty(zb))
    Estimativa=EstimativaB;
end
if (~isempty(za) && isempty(zb))
    Estimativa=EstimativaA;
end
[PontosA PontosB];
% Verifica se há histórico
if Estimativa>-1
    % se sim: coloca previsão
    newline2=sprintf('%d %d-%d-%d %d:%d:%d %d %d %d
\n', dados(1), dados(2), dados(3), dados(4), dados(5), dados(6), dados(7), P
consumida, Estimativa, 0); % (mean(z)-1.0+std(z))^3.0);
    if (dados(3)==11 || dados(3)==12 || dados(3)==1)
        DadosOut2=[DadosOut2;
Pconsumida, Estimativa];
    else
        DadosOut1=[DadosOut1;
Pconsumida, Estimativa];
    end
    fwrite(FPout, newline2);
    if (dados(3)==10 && EM1==0)
        EM1=ERROMEDIO/LINHAS;
        E2M=ERRO2MEDIO;
        ERROMEDIO=0;
        ERRO2MEDIO=0;
        LINHAS=0;
    end
    if (Pconsumida>0)
        ERROMEDIO=ERROMEDIO+abs(Estimativa-
Pconsumida)/Pconsumida;
        ERRO2MEDIO=ERRO2MEDIO+ERROMEDIO^2;
        LINHAS=LINHAS+1;
        [dados(3) ERROMEDIO/LINHAS];
    end
else
    % se não: coloca apenas dados medidos
    if (dados(3)==11 || dados(3)==12 || dados(3)==1)
        DadosOut2=[DadosOut2; Pconsumida, 0];
    end
end

```

```
        else
            DadosOut1=[DadosOut1; Pconsumida, 0];
        end
        newline2=sprintf('%d %d-%d-%d %d:%d:%d %d\n', dados (1), dados (2), dados (3), dados (4), dados (5), dados (6), dados (7), Pconsumida);
        fwrite (FPout, newline2);
    end

matrizA(diadasemana, horadodia+1, indicetemperatura, indiceradiacao, indicePconsumida)=matrizA(diadasemana, horadodia+1, indicetemperatura, indiceradiacao, indicePconsumida)+1;

matrizB(diadasemana, horadodia+1, indicePconsumida)=matrizB(diadasemana, horadodia+1, indicePconsumida)+1;
    %[diadasemana, horadodia+1,
matriz (diadasemana, horadodia+1, indicetemperatura, indiceradiacao, indicePconsumida)]
    end
end
end

% Lê a próxima linha do ficheiro
linha=fgetl (FP);
end
fclose all;
beep

EM1
E2M
ERROMEDIO/LINHAS
ERRO2MEDIO/LINHAS

subplot (2,1,1)
plot (DadosOut1 (:,1))
hold on
plot (DadosOut1 (:,2), 'r')
axis ([0 2e4 0 10e4])
subplot (2,1,2)
plot (DadosOut2 (:,1))
hold on
plot (DadosOut2 (:,2), 'r')
```