



Mestrado em Engenharia Electrotécnica

Desenvolvimento de Aplicações para Monitorização de Consumos de Energia em Instalações Eléctricas

Trabalho Projeto apresentado para a obtenção do grau de Mestre em
Engenharia Electrotécnica
Especialização em Sistemas Industriais

Autor

Tiago Miguel Gonçalves Neves

Orientador

Victor Daniel Neto dos Santos

Professor do Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia de Coimbra

Coimbra, Setembro, 2018

“Se tudo o que soubesse fosse dado, não lhe dava tanto valor”

AGRADECIMENTOS

Após a conclusão deste trabalho, que representa o fim desta etapa, deixo os meus sinceros agradecimentos a todas as pessoas com as quais tive o privilégio de aprender e evoluir.

Ao Professor Doutor Victor Daniel Neto dos Santos deixo um agradecimento especial pelo apoio prestado e pela disponibilidade demonstrada no decorrer do presente trabalho.

Gostaria de agradecer a todos os amigos e colegas que me acompanharam nesta caminhada pelo ensino superior, pois sem eles este caminho não teria sido o mesmo.

Por último, mas não menos importante deixo um agradecimento muito especial aos meus pais por sempre me apoiarem e possibilitarem esta oportunidade e ao meu irmão pelo apoio e pela motivação dada ao longo destes anos de estudo no Instituto Superior de Engenharia de Coimbra.

RESUMO

As redes de energia inteligentes *smart grids* têm por objetivo melhorar a eficiência, fiabilidade e sustentabilidade da produção e consumo de energia elétrica, através de informação recolhida sobre a atividade dos produtores e consumidores.

A sua implementação irá tornar possível a integração de fontes de energia renováveis, a redução das perdas de transporte e distribuição de energia e a implementação de tarifas dinâmicas de energia. Aliado a este conceito encontra-se a tecnologia de *smart metering* que permite a recolha de informação em tempo real, relativa ao consumo de energia o que vai influenciar os consumidores a mudarem os seus hábitos, tornando-os mais conscientes em relação ao seu perfil de consumo atual.

A recolha dos dados referentes ao consumo energético dos consumidores, será efetuada por *smart meters*, que têm por função medir os consumos de energia elétrica de uma instalação, enviar as leituras efetuadas para uma unidade de armazenamento e processamento de dados, permitindo ao utilizador visualizar a informação recolhida, as estatísticas mais relevantes e os eventos detetados, em plataformas desenvolvidas para o efeito.

O objetivo deste projeto compreendeu o desenvolvimento de um dispositivo eletrónico com a capacidade de adquirir, registar e armazenar valores relativos à energia consumida, a partir de um contador de energia digital. O dispositivo de aquisição foi desenvolvido utilizando um microcontrolador de baixo custo para estabelecer a comunicação com o contador de energia. Os dados adquiridos são guardados numa base de dados e numa memória volátil as quais servem de base às plataformas de ICT desenvolvidas: uma em ambiente *web* e uma aplicação Android, as quais permitem ao consumidor consultar os dados recolhidos referentes ao seu consumo de energia.

Palavras-chave: *smart meter*, monitorização de energia, *smart grids*

ABSTRACT

Smart grids are aimed at improving the efficiency, reliability and sustainability of electricity production and consumption through information collected on the activity of producers and consumers.

Its implementation will make it possible to integrate renewable energy sources, reduce transport losses and distribute energy, and implement dynamic energy tariffs. Allied to this concept is the smart metering technology that allows real-time information on energy consumption, which will influence consumers to change their habits, making them more aware of their consumption profile.

The collection of energy consumption data from consumers will be carried out by smart meters, which are used to measure the electrical energy consumption of an installation, send the readings made to a storage and data processing unit, allowing the user to visualize the information collected, the most relevant statistics and the events detected, on platforms created for this purpose.

The objective of this project was the development of an electronic device with the capacity to acquire, register and store values related to the energy consumed, from a digital energy meter. The acquisition device was developed using a low cost microcontroller to establish communication with the energy meter. The data acquired will be stored in a database and in a flash memory which serve as the basis for the platforms developed: one in the web environment and another in the Android application format that allow the consumer to consult the data collected regarding their energy consumption.

Keywords: *smart meter, energy monitoring, smart grids*

ÍNDICE

AGRADECIMENTOS	v
RESUMO	vii
ABSTRACT	ix
ÍNDICE	xi
ÍNDICE DE FIGURAS	xv
ÍNDICE DE QUADROS	xvii
ÍNDICE DE SCRIPTS	xix
ABREVIATURAS	xxi
INTRODUÇÃO	- 1 -
ENQUADRAMENTO	- 1 -
OBJETIVOS	- 2 -
ADVANCED METERING INFRASTRUCTURE	- 3 -
Obtenção e leitura de dados	- 5 -
Instalação, configuração e manutenção do sistema de smart metering	- 5 -
Monitorização da qualidade da energia	- 7 -
Gestão dos dados sobre falhas de energia	- 7 -
Auxiliar as ações de resposta face à procura de energia	- 8 -
Suportar a geração de energia distribuídas	- 8 -
Gestão da rede de distribuição de energia	- 9 -
Gestão de avarias e interferências na rede de energia	- 9 -
Interação com os dispositivos de medição instalados	- 9 -
Gestão das medidas de eficiência da rede	- 9 -
Apresentação de mensagens	- 9 -
Erros na execução de uma tarefa	- 10 -
PLANO DE TRABALHO	- 10 -
Cronograma do Projeto	- 11 -
ESTRUTURA DA DISSERTAÇÃO	- 12 -
ESTADO DA ARTE	- 13 -
TECNOLOGIAS CABLADAS	- 13 -
<i>Ethernet</i>	- 13 -
<i>Power Line Carrier</i>	- 14 -
Comunicação Série	- 14 -

TECNOLOGIAS SEM FIOS.....	- 15 -
ZigBee.....	- 15 -
WirelessHART.....	- 17 -
6LOWPAN	- 19 -
Wi-Sun.....	- 22 -
EnOcean.....	- 23 -
Lora e LoraWan.....	- 25 -
Wi-Fi.....	- 27 -
POWER METERS DISPONÍVEIS NO MERCADO	- 29 -
SICAM P50 / P55.....	- 29 -
EM21 72D.....	- 30 -
ION8650 meter	- 31 -
CONCLUSÕES	- 33 -
IMPLEMENTAÇÃO DO SISTEMA	- 35 -
SISTEMA DE AQUISIÇÃO DE DADOS	- 35 -
Power meter 710.....	- 36 -
Conversor RS232 / RS485	- 38 -
MySQL	- 39 -
Apache	- 39 -
Protocolo Modbus.....	- 40 -
Aplicação desenvolvida em Matlab	- 41 -
Implementação utilizando um microcontrolador Raspberry Pi.....	- 44 -
SISTEMA DE INFORMAÇÃO	- 50 -
Base de dados (MySQL).....	- 52 -
Aplicação Web	- 57 -
Aplicação Android.....	- 74 -
CONCLUSÕES	- 87 -
RESULTADOS PRÁTICOS	- 89 -
RESULTADOS OBTIDOS	- 89 -
Termoventilador.....	- 89 -
Aquecedor a óleo	- 91 -
Torradeira	- 93 -
Fonte de alimentação dc laboratorial	- 95 -
Carregador de um laptop.....	- 96 -
Tostadeira	- 100 -
Microondas	- 102 -
Frigorífico	- 105 -
CONCLUSÕES	- 108 -

CONCLUSÕES E TRABALHO FUTURO	- 109 -
REFERÊNCIAS	- 111 -

ÍNDICE DE FIGURAS

<i>Figura 1 – Arquitetura de um sistema AMI. (Faisal, 2012)</i>	- 3 -
<i>Figura 2 – Camadas do protocolo ZigBee. (Di, 2017)</i>	- 16 -
<i>Figura 3 – “Superframe” WirelessHART. (Maass, 2017)</i>	- 18 -
<i>Figura 4 – Rede WirelessHART. (Künzel, 2017)</i>	- 19 -
<i>Figura 5 – Estrutura dos cabeçalhos 6LoWPAN comprimidos. (Olsson, 2014)</i>	- 20 -
<i>Figura 6 – Arquitetura de uma rede 6LoWPAN. (Qiu, 2018)</i>	- 22 -
<i>Figura 7 – Sistema Wi-Sun com comunicação multi-hop. (Mochizuki, 2016)</i>	- 23 -
<i>Figura 8 – Rede de controlo inteligente de iluminação utilizando tecnologia EnOcean. (Li, 2014)</i>	- 25 -
<i>Figura 9 – Topologia típica de uma rede LoRaWAN. (Usmonov, 2017)</i>	- 26 -
<i>Figura 10 – Power meter SICAM P50 / P55. (Siemens, 2016)</i>	- 30 -
<i>Figura 11 – Power meter EM21 72D. (Gavazzi)</i>	- 30 -
<i>Figura 12 – Power meter ION8650 meter. (Electric, 2018)</i>	- 31 -
<i>Figura 13 – Sistema de aquisição de dados de consumo de energia.</i>	- 36 -
<i>Figura 14 – Rede monofásica fase-neutro de 2 fios, 1 TI. (Electric, 2004)</i>	- 37 -
<i>Figura 15 – Ligação em simultâneo de aparelhos a dois fios. (Electric, 2004)</i>	- 38 -
<i>Figura 16 – Estrutura da mensagem em formato Modbus. (Modbus, 2018)</i>	- 41 -
<i>Figura 17 – Arquitetura ICT do sistema desenvolvido</i>	- 50 -
<i>Figura 18 – Estrutura da base de dados e relação entre tabelas</i>	- 53 -
<i>Figura 19 – Formulário de Login da aplicação Web</i>	- 59 -
<i>Figura 20 – Menu de falha do Login</i>	- 59 -
<i>Figura 21 – Menu de consulta dos valores em tempo real</i>	- 60 -
<i>Figura 22 – Menu de consulta dos valores num determinado espetro temporal</i>	- 61 -
<i>Figura 23 – Menu de atualização dos utilizadores.</i>	- 62 -
<i>Figura 24 – Menu de atualização dos utilizadores, com a opção de alterar um utilizador selecionada.</i> ...	- 65 -
<i>Figura 25 – Menu de atualização dos power meters</i>	- 66 -
<i>Figura 26 – Menu de atualização de power meters, com a opção “Alterar” selecionada.</i>	- 67 -
<i>Figura 27 – Menu de atualização de companhias</i>	- 69 -
<i>Figura 28 – Página para atualizar as relações entre companhias e utilizadores</i>	- 71 -
<i>Figura 29 – Menu de navegação da aplicação web</i>	- 72 -
<i>Figura 30 – Menu de navegação na aplicação, para utilizadores com permissão de consulta.</i>	- 72 -
<i>Figura 31 – Layout da página de login</i>	- 75 -
<i>Figura 32 – Layout do menu principal da aplicação android</i>	- 77 -
<i>Figura 33 – Menu de consulta dos valores referentes à energia consumida em tempo real</i>	- 79 -
<i>Figura 34 – Layout do menu “Energy Quality”.</i>	- 81 -
<i>Figura 35 – Layout do menu de consulta do preço da energia.</i>	- 82 -

<i>Figura 36 – Layout do menu de consulta dos consumos ao longo do tempo.</i>	<i>- 85 -</i>
<i>Figura 37 – Gráfico temporal da potência consumida pelo termoventilador.</i>	<i>- 90 -</i>
<i>Figura 38 – Gráfico temporal da energia consumida pelo termoventilador.</i>	<i>- 91 -</i>
<i>Figura 39 – Gráfico temporal da potência consumida pelo aquecedor a óleo.</i>	<i>- 92 -</i>
<i>Figura 40 – Gráfico temporal da energia consumida pelo aquecedor a óleo.</i>	<i>- 93 -</i>
<i>Figura 41 – Gráfico temporal da potência consumida pela torradeira.</i>	<i>- 94 -</i>
<i>Figura 42 – Gráfico temporal da energia consumida pela torradeira.</i>	<i>- 95 -</i>
<i>Figura 43 – Gráfico temporal da potência consumida pela fonte de alimentação DC.</i>	<i>- 95 -</i>
<i>Figura 44 – Gráfico temporal da energia consumida pela fonte de alimentação DC.</i>	<i>- 96 -</i>
<i>Figura 45 – Gráfico temporal da potência consumida pelo carregador de um PC.</i>	<i>- 97 -</i>
<i>Figura 46 – Gráfico temporal da energia consumida pelo carregador de um PC.</i>	<i>- 98 -</i>
<i>Figura 47 – Gráfico temporal do THD da tensão do carregador de um PC.</i>	<i>- 99 -</i>
<i>Figura 48 – Gráfico temporal do THD da corrente do carregador de um PC.</i>	<i>- 99 -</i>
<i>Figura 49 – Gráfico temporal do fator de potência do carregador de um PC.</i>	<i>- 100 -</i>
<i>Figura 50 – Gráfico temporal da potência consumida pela tostadeira.</i>	<i>- 101 -</i>
<i>Figura 51 – Gráfico temporal da energia consumida pela tostadeira.</i>	<i>- 101 -</i>
<i>Figura 52 – Gráfico temporal da potência consumida pelo microondas.</i>	<i>- 102 -</i>
<i>Figura 53 – Gráfico temporal da energia consumida pelo microondas.</i>	<i>- 103 -</i>
<i>Figura 54 – Gráfico temporal da corrente consumida pelo microondas.</i>	<i>- 104 -</i>
<i>Figura 55 – Gráfico temporal da variação da frequência na rede elétrica.</i>	<i>- 105 -</i>
<i>Figura 56 – Gráfico temporal da potência consumida pelo frigorífico.</i>	<i>- 106 -</i>
<i>Figura 57 – Gráfico temporal da energia consumida pelo frigorífico.</i>	<i>- 107 -</i>
<i>Figura 58 – Gráfico temporal do fator de potência do frigorífico.</i>	<i>- 108 -</i>

ÍNDICE DE QUADROS

<i>Quadro 1 – Planeamento das tarefas a realizar.</i>	- 11 -
<i>Quadro 2 – Especificações da camada física dos sistemas apresentados.</i>	- 15 -
<i>Quadro 3 – Camadas de atuação dos sistemas apresentados.</i>	- 27 -
<i>Quadro 4 – Especificações da camada física dos sistemas apresentados.</i>	- 28 -
<i>Quadro 5 – Especificações de power meters comerciais. (Electric, 2018) (Gavazzi) (Siemens, 2016)</i>	- 32 -
<i>Quadro 6 – Mensagem de pedido.</i>	- 41 -
<i>Quadro 7 – Mensagem de resposta.</i>	- 41 -
<i>Quadro 8 – Características do Raspberry Pi Model B.</i>	- 44 -
<i>Quadro 9 – Valores constituintes da tabela PM_ValUES.</i>	- 53 -
<i>Quadro 10 – Valores presentes na tabela users.</i>	- 54 -
<i>Quadro 11 – Valores constituintes da tabela PM.</i>	- 54 -
<i>Quadro 12 – Valores constituintes da tabela companies.</i>	- 55 -
<i>Quadro 13 – Valores constituintes da tabela comp_user.</i>	- 55 -

ÍNDICE DE SCRIPTS

<i>Script 1 – Inicialização da porta série.</i>	- 42 -
<i>Script 2 – Função de leitura dos registos do power meter.</i>	- 42 -
<i>Script 3 – Recolher valores do power meter.</i>	- 43 -
<i>Script 4 – Guardar valores na base de dados.</i>	- 43 -
<i>Script 5 – Inicialização do canal série.</i>	- 46 -
<i>Script 6 – Inicialização das variáveis de comunicação.</i>	- 47 -
<i>Script 7 – Recolha dos valores do power meter e da data e hora.</i>	- 48 -
<i>Script 8 – Comunicação com o servidor com a base de dados.</i>	- 48 -
<i>Script 9 – Registo de valores na base de dados.</i>	- 49 -
<i>Script 10 – Registo dos valores no dispositivo de armazenamento externo.</i>	- 49 -
<i>Script 11 – Validação dos dados de Login.</i>	- 60 -
<i>Script 12 – Registo de um novo utilizador.</i>	- 63 -
<i>Script 13 – Apresentação dos utilizadores.</i>	- 63 -
<i>Script 14 – Alterar utilizadores.</i>	- 64 -
<i>Script 15 – Apagar um utilizador.</i>	- 64 -
<i>Script 16 – Alterar um utilizador.</i>	- 65 -
<i>Script 17 – Registrar um power meter.</i>	- 66 -
<i>Script 18 – Alterar um power meter.</i>	- 67 -
<i>Script 19 – Apagar um power meter.</i>	- 67 -
<i>Script 20 – Recolher o nome de uma companhia.</i>	- 68 -
<i>Script 21 – Registrar uma companhia.</i>	- 68 -
<i>Script 22 – Chamada dos botões dinâmicos do formulário.</i>	- 70 -
<i>Script 23 – Criação de um botão dinâmico do formulário.</i>	- 70 -
<i>Script 24 – Botão do menu de administrador.</i>	- 72 -
<i>Script 25 – Botão do menu de consulta.</i>	- 73 -
<i>Script 26 – Início de sessão PHP.</i>	- 73 -
<i>Script 27 – Fim de sessão PHP.</i>	- 73 -
<i>Script 28 – Recolha dos parâmetros de Login.</i>	- 75 -
<i>Script 29 – Pedido ao servidor e resposta.</i>	- 76 -
<i>Script 30 – Iniciar o menu "Energy Price".</i>	- 78 -
<i>Script 31 – Recepção dos dados retornados pelo servidor.</i>	- 80 -
<i>Script 32 – Detecção de todos os "power meters" e criação do botão "dropdown".</i>	- 80 -
<i>Script 33 – Recepção dos dados retornados pelo servidor.</i>	- 80 -
<i>Script 34 – Apresentação dos últimos dados registados pelo "power meter".</i>	- 81 -
<i>Script 35 – Criação do botão de seleção da data.</i>	- 83 -
<i>Script 36 – Calculo do preço da energia.</i>	- 85 -

Script 37 – Recolha da grandeza a apresentar. - 86 -

Script 38 – Inicialização do gráfico. - 86 -

Script 39 – Construção do gráfico com valores e legenda. - 87 -

ABREVIATURAS

AAA	Authentication Authorization and Accounting
ADU	Application Data Unit
AES	Advanced Encryption Standard
AMI	Advanced Metering Infrastructure
ASK	Amplitude Shift Keying
BPSK	Binary Phase-Shift Keying
CSMA/CA	Carrier Sense Medium Access with Collision Avoidance
CSS	Chirp Spread Spectrum
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DSSS	Direct Sequence Spread Spectrum
ERP	EnOcean Radio Protocol
FD	Field Devices
FFD	Full Function Devices
FHSS	Frequency Hopping Spread Spectrum
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
GPIO	General Purpose Input/Output
GTS	Guarantee Time Slot
HAN	Home Area Network
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol

ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISM	Industrial, Scientific and Medical
JSON	JavaScript Object Notation
LMA	Local Mobility Anchor
LoRa	Long-Range™
LoWPAN	Low-Power Wireless Personal Area Network
LPWAN	Low Power Wide Area Network
MAC	Medium Access Control
NAN	Neighborhood Area Network
NM	Network Manager
NS	Network Server
OQPSK	Offset Quadrature Phase-Shift Keying
OSI	Open System Interconnection
PC	Personal Computer
PDU	Protocol Data Unit
RFD	Reduced Function Devices
SF	Spreading Factor
SNR	Signal to Noise Ratio
SQL	Structured Query Language
TDMA	Time Division Multiple Access

THD	Total Harmonic Distortion
TI	Transformador de Intensidade
TT	Transformador de Tensão
URL	Uniform Resource Locator
USB	Universal Serial Bus
WAN	Wide Area Network
WSN	Wireless Sensor Network
6LoWPAN	IPv6 LoWPAN
6MAG	6LoWPAN Mobile Access Gateway
6MN	6LoWPAN Mobile Nodes

INTRODUÇÃO

ENQUADRAMENTO

Em termos genéricos as redes de energia inteligentes *smart grids* têm como objetivo melhorar a eficiência, fiabilidade e sustentabilidade da produção e distribuição de energia elétrica usando a informação da atividade dos produtores e consumidores. A exploração das redes inteligentes permite também uma maior integração das fontes de energia renováveis, a redução das perdas de transporte e distribuição de energia e a implementação de tarifas dinâmicas de energia.

Abordando as *smart grids* do ponto de vista do consumidor de energia elétrica, será necessário dispor de mais informação sobre a energia consumida pelo utilizador residencial. Dessa forma será possível estudar com maior pormenor a distribuição dos consumos de energia ao longo de um dia, possibilitando ao utilizador realizar uma distribuição mais eficiente dos períodos de utilização da energia para cargas não prioritárias.

Por exemplo, um ciclo de lavagem de uma máquina de lavar loiça pode ser realizado em qualquer instante do dia, designadamente à noite quando a taxa de utilização da rede elétrica é menor e o preço da energia é mais reduzido. Ao realizar este balanceamento dos períodos de utilização da energia elétrica estamos a aumentar a eficiência da rede, pois não estamos a sobrecarregar a rede em horas de elevada utilização. A redução dos custos de operação e gestão da rede obtidas tem como consequência uma diminuição dos custos de energia para o consumidor. Para além disso, a redução da procura de energia em determinados períodos do dia irá também afetar o preço final da energia para os consumidores.

Em última análise as *smart grids* oferecem aos seus cliente maior controlo, disponibilizando um conjunto de informação sobre o seu perfil de consumo de energia, permitindo assim que este tome decisões relativamente sobre a utilização da energia proveniente da rede, com o intuito deste se tornar um consumidor mais eficiente.

OBJETIVOS

O objetivo deste projeto compreende o planeamento, desenvolvimento e implementação de uma solução tecnológica capaz de realizar a aquisição dos valores de energia consumida numa instalação elétrica residencial ou industrial, a partir de um contador de energia digital.

O dispositivo de aquisição desenvolvido utiliza um microcontrolador, e controladores capazes de estabelecer a comunicação série, e *Ethernet* e ainda unidades de memória *flash* que permitem o armazenamento dos dados recolhidos. Este microcontrolador irá comunicar com o contador de energia que previamente recolhera dados relativos ao consumo energético os quais são armazenados numa base de dados e numa memória volátil de forma a evitar perdas de informação.

Para tornar a consulta dos dados recolhidos mais intuitiva para o consumidor, será disponibilizada toda a informação relativa ao consumo de energia, numa página *web* e numa aplicação Android, que foram desenvolvidas para esse efeito.

Na elaboração deste projeto pretendeu-se:

- Definir a informação necessária às aplicações desenvolvidas;
- Escolher a tecnologia de comunicação que melhor se adequa aos requisitos do problema;
- Conceber e projetar um dispositivo de aquisição de dados;
- Analisar e testar o dispositivo de aquisição desenvolvido;
- Implementar uma base de dados, de suporte às aplicações desenvolvidas;
- Recolher dados relativos ao consumo de energia de diversos eletrodomésticos e outros dispositivos elétricos;
- Desenvolver um *website*;
- Desenvolver uma aplicação Android.

ADVANCED METERING INFRASTRUCTURE

Um sistema *Advanced Metering Infrastructure* (AMI) é constituída por quatro componentes: o *smart meter*, o concentrador ou coletor de dados, a rede de comunicação e o sistema AMI *headend*. A Figura 1 representa a arquitetura típica de um sistema AMI.

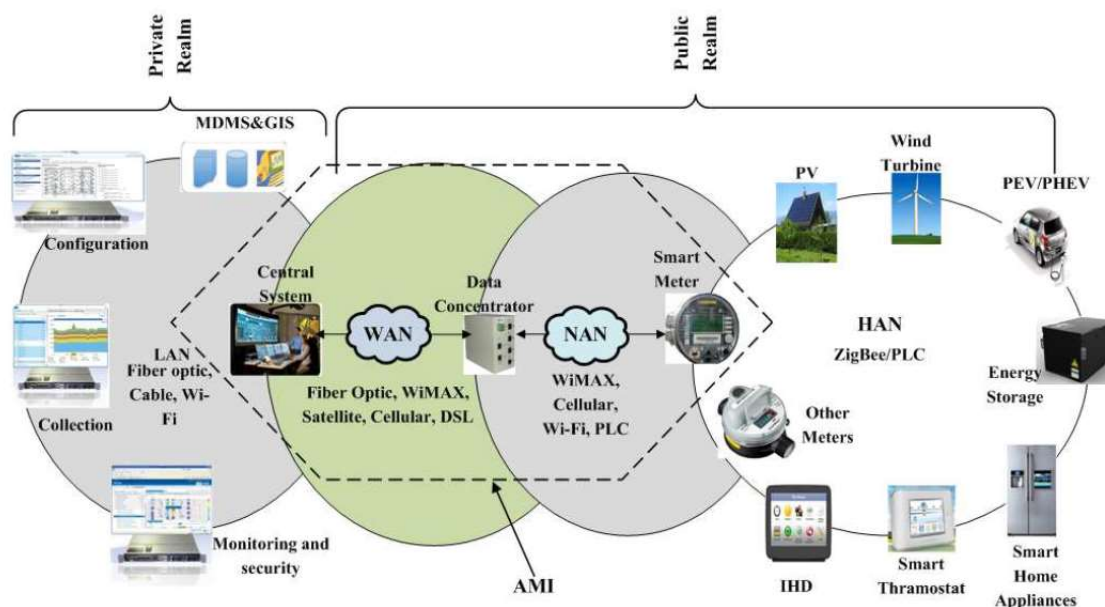


Figura 1 – Arquitetura de um sistema AMI. (Faisal, 2012)

O *smart meter* faz a ligação entre os eletrodomésticos e dispositivos elétricos existentes numa rede *Home Area Network* (HAN) e o concentrador de dados da rede *Neighborhood Area Network* (NAN). Conceptualmente o *smart meter* é composto por três componentes: o contador elétrico, a unidade de processamento e o módulo de comunicação. O contador elétrico de energia é responsável pela medição do consumo de energia elétrica. A unidade de processamento é responsável pelo processamento e armazenamento dos dados medidos pelo contador elétrico e enviados pelo módulo de comunicação e pelo controlo da comunicação entre o contador de energia e o módulo de comunicação. (Faisal, 2012)

Resumindo o *smart meter* é responsável pelas funções de medição, armazenamento e transmissão da informação relativa ao consumo ou geração de energia elétrica numa rede HAN. Adicionalmente, estes dispositivos monitorizam a qualidade da energia da rede elétrica.

O concentrador ou coletor de dados está localizado usualmente em subestações, e tem como função gerir a informação adquirida a partir de *smart meters* dispersos pelas várias redes HAN. O concentrador é responsável por efetuar a recolha e armazenamento da informação proveniente dos *smart meters*, e por enviar essa informação para um sistema central de análise e faturação da energia elétrica consumida. Estes dispositivos são também responsáveis pela deteção e configuração de novos *smart meters* instalados em redes HAN.

O sistema AMI compreende diversas redes de comunicação tais como: *Neighborhood Area Network* (NAN) e *Wide Area Network* (WAN), as quais são suportadas em diferentes tecnologias sem fios e cabladas. Por seu lado as comunicações entre o concentrador e o sistema central compreende as WAN. As NAN são responsáveis pela comunicação entre *smart meters*, que atuam como *gateway* ou pontos de acesso, para as HAN. As tecnologias Wi-Fi, satélite, Ethernet, fibra ótica e ZigBee podem ser utilizadas nas redes NAN. A rede WAN é responsável pela comunicação entre concentradores de dados e os *smart meters*.

O sistema AMI *headend* está localizado no lado do fornecedor de energia sendo responsável pela comunicação bidirecional com os AMI *meters*, de forma a executar leituras periódicas, detetar erros de comunicação, monitorizar a integridade do sistema AMI, implementar atualizações de *firmware* nos *smart meters* e configurar alterações na rede. O sistema AMI *headend* comunica ainda aos sistemas de faturação, de gestão da energia na rede e ao sistema de distribuição de energia elétrica, os dados recolhidos pelos *smart meters*. (Faisal, 2012)

Da análise da informação disponível em (ETSI, 2010) elencam-se as seguintes funcionalidades de uma rede AMI de entre elas refira-se as seguintes:

- Leitura remota da metrologia do fornecimento de energia elétrica. Capacidade de comunicar à distância o valor do registo do medidor a uma hora definida ou através de um pedido.
- Comunicação bidirecional entre o sistema de medição e o fornecedor. Capacidade do dispositivo comunicar a qualidade da energia, eventos na rede como o funcionamento incorreto de algum elemento da rede, o estado do medidor, entre outros parâmetros de interesse. Deve ainda ser possível ao fornecedor de energia configurar remotamente o dispositivo, fazer a atualização do *firmware* e do *software*, sincronizar o relógio interno para estabelecer comunicações e permitir o agendamento de medições.

De seguida serão apresentados exemplos de funcionalidades decorrentes da utilização da tecnologia *smart metering*.

- Obtenção e leitura de dados;
- Instalação, configuração e manutenção do sistema de *smart metering*;
- Monitorização da qualidade da energia;
- Gestão dos dados sobre falhas de energia;
- Auxiliar as ações de resposta face à procura de energia;
- Suportar a geração de energia distribuída;
- Gestão da rede de distribuição de energia;
- Gestão de avarias e interferências na rede de energia;
- Interação com os dispositivos de medição instalados;
- Gestão das medidas de eficiência da rede;
- Apresentação de mensagens.

OBTENÇÃO E LEITURA DE DADOS

Neste caso o sistema de informação *smart metering* reconhece o destino da informação recolhida e tem um endereço destino para enviar a informação. O sistema *smart metering* foi configurado para fornecer leituras de dados num determinado horário. A troca de informação acontece em resposta a um pedido efetuado ou de forma automática. Para uma leitura automática o sistema de informação *smart metering* regista a informação recolhida pelo medidor, assim como a data e hora da leitura. Numa leitura a pedido é efetuado o pedido de leitura, o sistema de *smart metering* valida o pedido e envia de volta a informação recolhida na leitura.

INSTALAÇÃO, CONFIGURAÇÃO E MANUTENÇÃO DO SISTEMA DE SMART METERING

Para efetuar a instalação e configuração de um sistema de *smart metering* é necessário que o sistema de informação tenha um endereço disponível para atribuir ao sistema de medição. No fim da instalação e configuração o dispositivo de medição comunica ao sistema de informação *smart metering* que a sua instalação, configuração ou manutenção foram finalizadas com sucesso ou se ocorreram erros.

Usualmente a instalação é agendada pela entidade responsável, que marca uma data e hora com o consumidor para instalar o dispositivo de medição. Uma vez instalado o dispositivo de medição, este comunica com o sistema de informação *smart metering* com o intuito de se registar no sistema. De seguida, o sistema de informação pede ao dispositivo a informação necessária após a sua inicialização, compreendendo medições e informação técnica do dispositivo, para finalizar a instalação.

A configuração do dispositivo é realizada pela entidade responsável pela rede, que envia as configurações a realizar a um determinado dispositivo, para o sistema de informação *smart metering*. De seguida, o sistema de informação valida o pedido e aplica as configurações ao dispositivo e envia uma mensagem para a entidade responsável a confirmar que as configurações foram alteradas.

É usual o sistema de informação identificar que uma secção da rede atingiu o fim do seu tempo de vida ou apresenta uma avaria e informa a entidade responsável pela manutenção da rede a solicitar uma ação de manutenção na secção da rede em causa. O sistema também pode solicitar a atualização do *software*. Durante uma atualização de *software* do sistema, a entidade responsável será notificada sobre qual o tipo de manutenção necessário, se é referente ao *software* ou ao *hardware*, ou a entidade responsável pela manutenção notifica o sistema de informação. De seguida a equipa de manutenção executa a manutenção conforme o tipo de manutenção necessário, o sistema de informação de *smart metering* recebe a confirmação de que a manutenção foi executada e notifica a entidade responsável pela manutenção. Finalmente o sistema de informação do sistema faz um diagnóstico para verificar se o problema foi devidamente resolvido com a intervenção efetuada. Caso o problema não fique resolvido é enviado um novo alerta à entidade responsável pela manutenção. Existem situações em que o processo de instalação, configuração ou manutenção não acontece como mencionado anteriormente, nestes casos o processo acontece da seguinte forma:

- O sistema de informação de *smart metering* não reconhece o pedido e considera inválido;
- O sistema de informação de *smart metering* não aplica a configuração definida;

- O sistema de informação de *smart metering* envia informação para a entidade responsável pela rede, com os detalhes dos erros ocorridos assim com a data e hora da ocorrência;
- A entidade responsável toma conhecimento da avaria;
- Se a manutenção de *hardware* for necessária então a entidade responsável marca uma visita ao local da instalação para instalar o novo hardware;
- A entidade responsável pela manutenção envia uma mensagem para o sistema de informação ou para o cliente diretamente, informando sobre a data e hora da manutenção;
- O sistema de informação apresenta uma mensagem ao consumidor com a data e hora da visita ate que este apague a mensagem ou limpe esta notificação;
- A entidade de manutenção visita o local e substitui o *hardware* necessário;
- O sistema de informação comunica as especificações do novo *hardware* à entidade responsável pela rede.

MONITORIZAÇÃO DA QUALIDADE DA ENERGIA

O responsável pela rede de energia envia um pedido ao sistema de informação de *smart metering* para obter informação sobre a qualidade da energia. O sistema de informação valida o pedido e retorna para a entidade gestora da rede a informação relativa à qualidade da energia juntamente com a data e hora da recolha de informação.

GESTÃO DOS DADOS SOBRE FALHAS DE ENERGIA

A gestão das falhas de energia pode ser dividida em duas categorias: as falhas não planeadas e as falhas de energia planeadas.

Quando ocorre uma falha não planeada o distribuidor envia um pedido ao sistema de informação de *smart metering* para recolher dados sobre a falha de energia, de seguida o sistema valida o pedido, recolhe e envia para o distribuidor toda a informação sobre a falha de energia juntamente com a data e hora do acontecimento.

Se por outro lado a falha de energia for planeada o distribuidor envia informação com o planeamento da falha de energia para o sistema de informação de *smart metering*, o sistema de informação recebe, valida e apresenta a informação ao consumidor, o consumidor aceita

a informação e o sistema de informação envia para o distribuidor uma mensagem a comunicar que a falha de energia planeada foi aceite. Se o consumidor não aceitar a falha de energia planeada então o sistema de informação assume que o pedido é inválido e comunica ao distribuidor o problema na validação do pedido sendo esta informação apresentada no interface do utilizador.

AUXILIAR AS AÇÕES DE RESPOSTA FACE À PROCURA DE ENERGIA

O operador do sistema envia uma mensagem para o sistema de informação de *smart metering* notificando o sistema de um evento de procura de energia, o sistema de informação após receber o pedido e validação do mesmo, apresenta a mensagem ao utilizador de que uma ação de controlo de cargas vai ser realizada. O consumidor depois lê a mensagem e escolhe se autoriza ou não essa ação, o sistema de informação posteriormente envia uma mensagem de volta para o operador notificando sobre a escolha do consumidor. Caso o consumidor escolha realizar a ação o sistema de informação envia para o operador uma mensagem com a quantidade de energia que está a ser consumida, para o operador verificar se o consumidor cumpriu com a ação de controlo de cargas.

SUPORTAR A GERAÇÃO DE ENERGIA DISTRIBUÍDAS

O distribuidor identifica as condições de produção desfavoráveis em relação ao mercado, depois decide que geradores de energia podem ser ligados para resolver o problema e depois envia um pedido para o sistema de informação de *smart metering* a pedir o aumento ou a diminuição da procura de energia, ou a exigir o aumento ou diminuição da produção. De seguida, o sistema de informação valida o pedido, o equipamento do consumidor toma conhecimento do pedido, o sistema de informação regista o fluxo bidirecional de energia entre o consumidor e o distribuidor e regista a qualidade da energia no consumidor. Posteriormente envia para o distribuidor informação sobre a energia na rede, e a sua qualidade, no fim o distribuidor recebe a informação enviada pelo sistema de informação e contabiliza a energia produzida pelo consumidor na sua fatura.

GESTÃO DA REDE DE DISTRIBUIÇÃO DE ENERGIA

O operador envia um pedido a pedir informação para o sistema de informação de *smart metering*. O sistema recebe e valida o pedido, depois envia uma resposta para o operador com a informação pedida, depois o operador toma medidas para otimizar o planeamento e utilização dos recursos.

GESTÃO DE AVARIAS E INTERFERÊNCIAS NA REDE DE ENERGIA

Quando ocorre uma avaria o sistema de informação envia para o operador uma notificação a reportar uma avaria no sistema, o operador depois pede detalhes sobre o evento ao sistema de informação, o sistema de informação valida o pedido e envia a informação pedida para o operador, de seguida o operador resolve o problema.

INTERAÇÃO COM OS DISPOSITIVOS DE MEDIÇÃO INSTALADOS

O sistema de informação de *smart metering* recebe uma instrução do operador, para interagir com um dispositivo, ou então de acordo com o horário definido necessita de interagir com um dispositivo, o sistema de informação apresenta a informação sobre esta interação ao utilizador e o utilizador decide se a ação ocorre ou não e depois é enviada uma mensagem para o operador comunicando a escolha do consumidor.

GESTÃO DAS MEDIDAS DE EFICIÊNCIA DA REDE

A entidade responsável pela eficiência envia uma mensagem para o sistema de informação de *smart metering* com o propósito de implementar ou propor uma medida de eficiência. O sistema valida a mensagem e apresenta a medida de eficiência na interface do utilizador, o consumidor depois decide se quer aprovar a medida de eficiência e o sistema de informação comunica ao operador a decisão do cliente e põe a medida em prática.

APRESENTAÇÃO DE MENSAGENS

O operador envia um pedido ao sistema de *smart metering*, para o sistema mostrar uma mensagem ao consumidor, o sistema valida a mensagem recebida e apresenta a mensagem ao consumidor, o consumidor depois decide se aprova a mensagem e o sistema envia uma mensagem ao operador a confirmar a decisão do consumidor.

ERROS NA EXECUÇÃO DE UMA TAREFA

Se o sistema de informação falhar ao executar uma tarefa, então envia uma mensagem ao responsável, que pode ser o operador da rede, o distribuidor ou o consumidor, com a informação referente ao erro ocorrido, que pode ser um erro de comunicação, de validação de uma mensagem, receção de um pedido, juntamente com a data e hora da ocorrência do erro. Se o destinatário da mensagem de erro não receber a mensagem então fica automaticamente a saber que ocorreu um erro de comunicação, pois não recebeu a mensagem esperada.

PLANO DE TRABALHO

O presente trabalho teve o seguinte plano de trabalhos que englobou os seguintes passos:

1. Pesquisa bibliográfica, preparação do projeto, relatórios intermédios (estado da arte e tecnologias relevantes).
2. Desenvolvimento do *software* de comunicação utilizando o *software* Matlab e desenvolvimento da base de dados destinada a guardar os dados adquiridos pelo *software* de comunicação.
3. Projeto e desenvolvimento de módulos eletrónicos de aquisição de dados de energia elétrica, utilizando um Raspberry Pi.
4. Instalação, configuração, testes e validação do módulo de aquisição de dados do *power meter*.
5. Desenvolvimento do interface homem-máquina, para interação com os dados recolhidos pelo módulo de comunicação.
6. Resultados experimentais relativos às soluções implementadas.
7. Relatório final definitivo do projeto.

Cronograma do Projeto

O presente projeto decorreu no Departamento de Engenharia Eletrotécnica do Instituto Superior de Engenharia de Coimbra, teve início no dia 17 de setembro de 2017. Durante este período foram desenvolvidas as tarefas presentes no Quadro 1.

O projeto compreendeu seis fases distintas, as quais tiveram lugar nas datas presentes no cronograma do Quadro 1.

Quadro 1 – Planeamento das tarefas a realizar.

Mês	Set.17	Out.18	Nov.18	Dez.18	Jan.18	Fev.18	Mar.18	Abril.18	Mai.18	Junho.18
Fase 1	X	X	X	X						
Fase 2	X	X								
Fase 3		X	X							
Fase 4			X	X	X	X	X			
Fase 5							X	X	X	
Fase 6							X	X	X	X

As diferentes fases compreenderam a realização das seguintes tarefas:

- Fase 1: Preparação do projeto, estudo do estado da arte e tecnologias relevantes.
- Fase 2: Projeto e implementação do *software* de comunicação e do módulo eletrónico de aquisição de dados.
- Fase 3: Testes e configuração do módulo desenvolvido e registo da informação numa base de dados a desenvolver para o efeito.
- Fase 4: Desenvolvimento de uma página *web* e de uma aplicação Android que interaja com a base de dados.
- Fase 5: Elaboração e entrega do relatório final provisório.
- Fase 6: Elaboração e entrega do relatório final definitivo de projeto.

ESTRUTURA DA DISSERTAÇÃO

A presente dissertação está dividida em cinco capítulos onde são apresentadas as várias fases do projeto.

No primeiro capítulo é realizado o enquadramento do tema, são apresentados os objetivos e as fases do presente projeto, bem como das funcionalidades do sistema de *smart metering* que se pretendem implementar.

No segundo capítulo é apresentado o estado da arte das redes *smart grid*, das tecnologias de comunicação utilizadas para implementar este tipo de sistemas na HAN, designadamente as características da camada física e lógica e dos seus protocolos.

No terceiro capítulo é apresentado o processo de desenvolvimento do sistema proposto neste projeto, assim como as suas funcionalidades. Os passos mais importantes compreendem o desenvolvimento do sistema de aquisição de dados, composto pelo *Power meter 710* e pelo Raspberry Pi. O desenvolvimento do sistema de informação, composto por uma base de dados e pelas duas aplicações, *web* e Android, desenvolvidas para apresentar ao consumidor a informação recolhida pelo sistema de aquisição são também apresentadas neste capítulo.

No quarto capítulo são apresentadas leituras referentes a vários eletrodomésticos presentes numa habitação. Este capítulo tem o objetivo de apresentar resultados que comprovem a validade do sistema para efetuar aquisição de dados de uma instalação elétrica referentes ao consumo de energia elétrica da mesma.

No quinto capítulo são apresentadas as conclusões relativas ao trabalho desenvolvido, assim como propostas para um trabalho futuro e melhoria do sistema desenvolvido.

ESTADO DA ARTE

O presente capítulo apresenta diversas tecnologias de comunicação existentes em sistemas de *smart metering* designadamente aquelas que estão implementadas nas HAN ao nível do consumidor e nas WAN ao nível da transmissão de energia. As tecnologias utilizadas podem cabladas ou sem fios, dependendo do ambiente de aplicação e dos requisitos. Este estudo entre as diferentes tecnologias de comunicação apresentadas servirá de base à escolha da tecnologia a utilizar para a transmissão de dados entre o *smart meter* e o concentrador na solução proprietária desenvolvida no presente projeto.

Adicionalmente são apresentados diversos *power meters* digitais de diferentes marcas e características presentes no mercado com o objetivo de se aferir as suas capacidades na aquisição dos valores de diferentes parâmetros que se pretendem disponibilizar ao utilizador de uma dada instalação elétrica.

TECNOLOGIAS CABLADAS

Ethernet

Ethernet é uma tecnologia de comunicação cablada, que pode ser do tipo coaxial, par trançado ou fibra ótica, baseada na norma IEEE 802.3. O facto de ser uma tecnologia cablada torna-a imune ao ruído, mas torna difícil implementar alterações na rede Ethernet. Na estrutura de uma rede Ethernet existem ainda dois dispositivos encarregues pelo encaminhamento das comunicações, nomeadamente *hubs* e *switches*. A velocidade de transmissão desta tecnologia pode variar entre 100 Mbps e 100 Gbps. A topologia tipicamente implementada neste tipo de redes é a topologia em estrela devido aos dispositivos responsáveis pelo encaminhamento das comunicações na rede. (Kuzlu, 2015)

Power Line Carrier

A tecnologia *Power Line Carrier* (PLC) utiliza a rede de transporte de energia elétrica para efetuar a comunicação entre dispositivos. Os dispositivos PLC injetam na rede um sinal de alta frequência, portadora, sobre a qual será transmitida informação, através da modulação desta onda com a informação. Esta tecnologia tem a vantagem de não necessitar de instalação de uma rede de comunicação, uma vez que utiliza a rede de energia elétrica já existente como rede de comunicação. Na Europa as frequências de transmissão utilizadas variam entre 85 Hz e 150 kHz, na América do Norte o valor da frequência pode ir até 540 kHz e em aplicações domésticas podem ser utilizadas frequências entre 20 Hz e 200 kHz. Existem duas especificações para a utilização da tecnologia PLC, que são HomePlug AV e HomePlug GP, sendo a segunda direcionada para o uso em aplicações em smart grids, as duas especificações foram desenvolvidas com base na norma IEEE 1901. Em relação à velocidade de transmissão a especificação HomePlug AV atinge velocidades de 200 Mbps e a especificação HomePlug GP atinge velocidades até 10 Mbps, sendo que em ambas as especificações o alcance atinge valores até 300 m. Para conferir segurança às comunicações efetuadas é utilizada a técnica de encriptação AES de 128 bits. (Kuzlu, 2015)

Comunicação Série

A comunicação série consiste em enviar os *bits* de um *byte* um de cada vez, num meio de comunicação cablado, um só fio, o que limita o número de dispositivos a comunicar a dois, um em cada extremo do cabo. Esta técnica é utilizada nas comunicações RS-232 e RS-485, que conseguem atingir velocidades de transmissão até 1 Mbps no caso das comunicações RS-232, mas habitualmente os dispositivos comunicam com uma velocidade máxima igual a 115.2 kbps. No caso das comunicações RS-485 a velocidade de transmissão pode atingir os 10 Mbps. Nas comunicações RS-232 e RS-485 é possível atingir alcances de 15 m e de 1200 m, respetivamente. (Kuzlu, 2015)

Quadro 2 – Especificações da camada física dos sistemas apresentados.

Tecnologias	Protocolo da camada física	Bandas de frequências		Velocidade de transmissão	Alcance	Técnica de encriptação
Ethernet	Norma IEEE 802.3			100 Mbps 100 Gbps		
PLC	Norma IEEE 1901	Europa	85 Hz a 150 kHz,	200 Mbps 10 Mbps	300 m	AES de 128 bits
		América do Norte	Até 540 kHz			
		Aplicações domésticas	20 Hz a 200 kHz			
Serial	RS-232			1 Mbps	15 m	
	RS-485			10 Mbps	1200 m	

TECNOLOGIAS SEM FIOS

ZigBee

A tecnologia Zigbee é simples e flexível em relação à sua implementação, pois baseia a sua arquitetura de funcionamento na norma internacional IEEE 802.15.4, para além de consumir pouca energia no seu funcionamento e comunicação. Está projetada para monitorizar sensores de baixa latência de dados e para controlo de redes *wireless*, pode ainda ser utilizada em automação de edifícios, em sistemas de segurança domésticos, e no controlo de redes industriais, em medições remotas e no desenvolvimento de *smart grids*.

Comparando com outras tecnologias o ZigBee destaca-se devido à sua fácil instalação e necessidade de poucos recursos para funcionar, disponibiliza ainda um conjunto de normas padrão, que permitem a sua fácil integração com outras tecnologias, garantindo a sua interoperabilidade. (Chati, 2006)

A tecnologia ZigBee pode ser utilizada usando as topologias em estrela, árvore ou malha, com um número máximo de nós igual a 65535. O protocolo ZigBee opera nas bandas *Industrial, Scientific and Medical* (ISM), 915 MHz (Américas), com 10 canais disponíveis, 868 MHz (Europa), com 1 canal disponível e 2.4 GHz, que utiliza 16 canais, com um alcance máximo que varia entre 10 m e 100 m, com uma largura de banda de 2 MHz e espaçamento entre canais de 5 MHz, o que permite interoperabilidade com a norma IEEE 802.11.b. O sistema utiliza a técnica *Direct Sequence Spread Spectrum* (DSSS) em conjunto com a modulação *Offset Quadrature Phase-Shift Keying* (OQPSK). A velocidade máxima de comunicação é igual a 20 kbps na banda dos 915 MHz, 250 kbps na banda de frequência de 868 MHz e 250 kbps na banda de frequência de 2.4 GHz. Os canais da banda dos 915 MHz utilizam modulação *Binary Phase-Shift Keying* (BPSK) o que resulta numa velocidade máxima de comunicação igual a 40 kbps. A sensibilidade do recetor deve ser superior a - 92 dBm para os canais na banda dos 915 MHz e superior a - 85 dBm para os canais na banda dos 2.4 GHz.

A tecnologia ZigBee utiliza o padrão IEEE 801.15.4 na camada física, denominada de camada PHY e na camada de ligação lógica, denominada por camada *Medium Access Control* (MAC), a disposição destas camadas, na arquitetura do protocolo ZigBee pode ser observada na Figura 2. As camadas de rede e de aplicação foram desenvolvidas pela *ZigBee Alliance*. (Di, 2017) (Santaji, 2017) (Marks, 2017) (Somani, 2012)

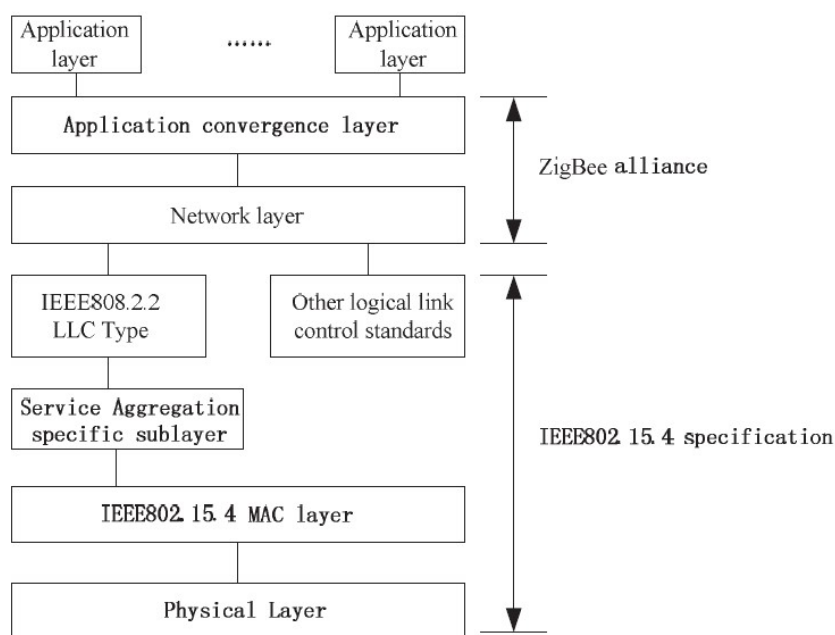


Figura 2 – Camadas do protocolo ZigBee. (Di, 2017)

A camada física da norma IEEE 802.15.4 é responsável pelas tarefas de ativação e desativação do transmissor de rádio, medição da qualidade da ligação rádio e seleção da frequência do canal.

A camada MAC é responsável pela transmissão e receção de pacotes de dados através da camada física. A camada MAC gere o acesso ao canal, é responsável pelo sinal farol, gestão do *Guarantee Time Slot* (GTS), validação de tramas e o *acknowledge* da receção de tramas.

Esta camada tem dois modos de funcionamento. O modo sem o sinal de farol, que utiliza a técnica ALOHA para gerir a transmissão de tramas e não utiliza o sinal de farol nem supertramas. E o modo com o sinal de farol, onde é enviado periodicamente um sinal de farol para outros dispositivos na rede de forma a sincronizar a comunicação entre os dispositivos da rede. A comunicação nesta camada é gerida utilizando a técnica *Carrier Sense Medium Access with Collision Avoidance* (CSMA/CA).

WirelessHART

O protocolo WirelessHART foi desenvolvido para monitorização e controlo de processos industriais, utilizando tecnologia sem fios, baseado na tecnologia HART.

A camada física do protocolo WirelessHART foi desenvolvida sobre o protocolo IEEE 802.15.4, combinando as técnicas de espalhamento espectral DSSS e *Frequency Hopping Spread Spectrum* (FHSS), para tornar a ligação de comunicação mais robusta.

Esta tecnologia é direcionada para comunicações de curta distância e baixa potência. Tem uma velocidade de transmissão de 250 kbps. O mecanismo FHSS utiliza 15 canais na banda de frequência dos 2.4 GHz. A codificação DSSS garante inerentemente a encriptação dos dados através da técnica de modulação M-ary. Para reforçar a segurança é utilizado o padrão de criptografia *Advanced Encryption Standard* (AES) com 128 bits de encriptação. (Supram, 2017)

Na camada lógica do protocolo é utilizado o mecanismo de acesso múltiplo *slotted Time Division Multiple Access* (TDMA). Este mecanismo divide cada canal de frequência em *timeslots*, atribuindo cada *timeslot* a um dispositivo, sendo que cada *timeslot* tem uma duração de 10 ms. A comunicação é definida numa supertrama, que consiste num intervalo de tempo fixo, com uma estrutura que pode ser consultada na Figura 3.

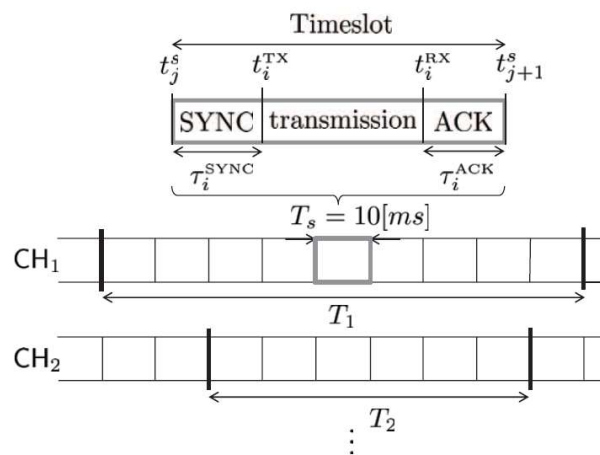


Figura 3 – “Superframe” WirelessHART. (Maass, 2017)

A camada de rede do sistema WirelessHART é baseada numa topologia de malha, centralizada e capaz de se reorganizar. Os dispositivos são capazes de encaminhar mensagens desempenhando dessa forma a função de roteadores de pacotes.

Na camada de aplicação de forma a manter a comunicação entre os dispositivos fonte e destino, existe a troca de mensagens de pedido e resposta entre eles, desta forma existe um tráfego constante de mensagens na rede o que mantém o tráfego entre os dispositivos ativo.

Tipicamente uma rede WirelessHART tem um *Network Manager* (NM), um *gateway*, um ou mais pontos de acesso e diversos dispositivos *Field Devices* (FD).

No servidor ou NM, que está ligado ao barramento de mais alto nível da rede, é alojada a aplicação que recebe e gere os dados recebidos pela rede WirelessHART. O NM é responsável pela gestão da rede, admissão de novos dispositivos à rede e distribuição dos recursos de comunicação. Este dispositivo é responsável pelo encaminhamento das mensagens na rede. De forma a criar caminhos de comunicação que proporcionem redundância, o NM utiliza caminhos de difusão para enviar mensagens dirigidas a todos os dispositivos na rede, para criar os caminhos destinados ao encaminhamento das mensagens específicas de cada dispositivo. (Künzel, 2017) (Supram, 2017) (Somkane, 2017) (Maass, 2017) (Arcari, 2017)

Na Figura 4 é apresentado um exemplo da arquitetura de uma rede WirelessHART.

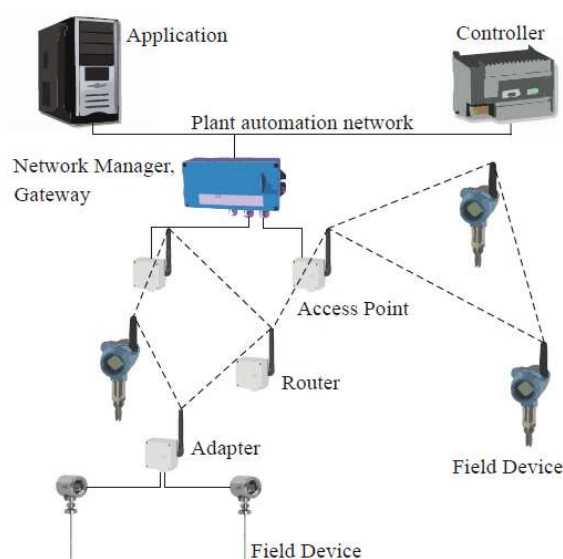


Figura 4 – Rede WirelessHART. (Künzel, 2017)

6LOWPAN

A *Internet Engineering Task Force* (IETF) estabeleceu o protocolo *Internet Protocol version 6* (IPv6) over *Low-power Wireless Personal Area Network* (LoWPAN), com o intuito de reduzir o consumo de energia das tecnologias *Wireless Sensor Network* (WSN), baseadas no protocolo IEEE 802.15.4, que acede à internet utilizando o protocolo IPv6.

O protocolo 6LoWPAN serve de camada de adaptação entre o protocolo IPv6 e a camada de ligação de dados que utiliza o protocolo IEEE 802.15.4.

Tendo em conta que a energia nos dispositivos de uma rede sem fios é limitada e como cada dispositivo não guarda toda a informação dos dispositivos no nó, não é viável o endereçamento utilizando *Dynamic Host Configuration Protocol* (DHCP). O protocolo IPv6 possui duas formas de configuração dos endereços: endereçamento fixo ou dinâmico. Para utilizar endereços dinâmicos é necessário criar endereços locais e efetuar a verificação relativa a possíveis endereços duplicados de forma a garantir a unicidade de cada endereço. O modo de endereços fixos necessita intervenção humana para definir os endereços manualmente.

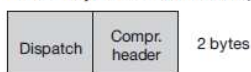
Os cabeçalhos das mensagens transmitidas são comprimidos, este padrão de *header* tem o nome de LOWPAN-MH. Existem três casos de compressão dos cabeçalhos, que são: na comunicação entre dispositivos na mesma rede 6LoWPAN, na comunicação para um dispositivo fora da rede conhecendo o prefixo do endereço desse dispositivo e na comunicação para fora da rede sem conhecer o prefixo do endereço do dispositivo no exterior da rede. No caso da comunicação dentro da mesma rede o cabeçalho pode ser comprimido a dois *bytes* pois apenas é possível comunicar com dispositivos na mesma rede. Na situação em que a comunicação é feita para um dispositivo no exterior da rede sabendo o prefixo do seu endereço é possível comprimir o cabeçalho até 12 *bytes*. Quando a comunicação é feita para o exterior da rede mas o prefixo do endereço do dispositivo no exterior da rede não é conhecido é possível comprimir o cabeçalho até 20 *bytes*. A estrutura dos cabeçalhos comprimidos é apresentada na Figura 5.

Primeiro é apresentado o cabeçalho IPv6, depois o cabeçalho 6LoWPAN comprimido no caso da comunicação dentro da mesma rede, depois quando a comunicação é feita para o exterior da rede sabendo o prefixo do endereço do dispositivo alvo e por último é apresentado o cabeçalho comprimido no caso em que a comunicação é feita para um dispositivo no exterior da rede sem saber o prefixo do seu endereço. O cabeçalho comprimido minimiza o tempo de comunicação e por sua vez a energia utilizada na comunicação, aumentando a autonomia e/ou tempo de vida útil da bateria do dispositivo. (Olsson, 2014)

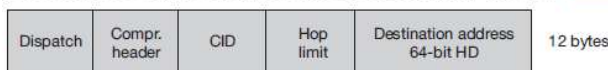
IPv6 header



1. Compressed header, FE80::CAFE:00FF:FE00:0100 → FE80::CAFE:00FF:FE00:0200



2. Compressed header, 2001::DEC4:E3A1:FE24:9600 → 2001::4455:84C6:39BB:A2DD



3. Compressed header, 2001::DEC4:E3A1:FE24:9600 → 2001::4455:84C6:39BB:A2DD

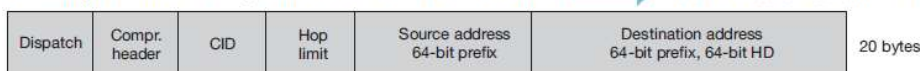


Figura 5 – Estrutura dos cabeçalhos 6LoWPAN comprimidos. (Olsson, 2014)

Uma rede 6LoWPAN é um conjunto de vários nós móveis, que são designados por 6LoWPAN *mobile nodes* (6MN), *gateways* de acesso 6LoWPAN *mobile access gateway* (6MAG), uma âncora local de mobilidade *Local Mobility Anchor* (LMA), e existe um servidor de autenticação, autorização e cálculo *Authentication Authorization and Accounting* (AAA). (Qiu, 2018)

- Nós móveis (6MN): são dispositivos móveis que medem e recolhem informação, podem ser *Full Function Devices* (FFD) ou *Reduced Function Devices* (RFD). Os FFD têm potência e memória suficiente para servirem de *router* e participam em topologias *peer-to-peer*. Os RFD apenas podem exercer a função de cliente na rede e são utilizados com uma topologia em estrela.
- Gateways (6MAG): são responsáveis por rastrear os movimentos dos 6MN na rede, dentro do seu domínio e comunicar as movimentações dos 6MN à LMA, através de um canal de comunicação estabelecido entre a LMA e a 6MAG.
- Âncora local de mobilidade (LMA): mantém os estados dos 6MN como ligados à rede e realiza a gestão dos prefixos da rede para cada nó da rede 6LoWPAN. A partir desse prefixo o 6MN pode configurar o seu endereço *Internet Protocol* (IP). A LMA exerce ainda a função de *router* para mensagens destinadas aos endereços IP referidos anteriormente.
- Servidor de autenticação (AAA): é responsável pela autenticação dos 6MN e pela autorização dos dispositivos permitidos a entrar na rede.

Os dispositivos 6MAG, LMA e AAA estão ligados através de uma ligação cablada, com elevada largura de banda e taxa de erros reduzida.

A topologia de uma rede 6LoWPAN é habitualmente em malha com ligações de curto alcance, como apresentado na Figura 6. Não suporta comunicações de *multicasting* para preservar energia. (Qiu, 2018) (Shaham, 2016) (Napiah, 2018) (Yousheng, 2017)

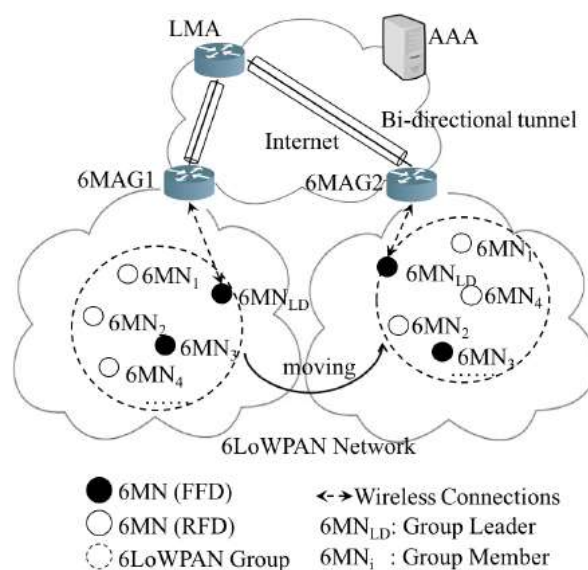


Figura 6 – Arquitetura de uma rede 6LoWPAN. (Qiu, 2018)

Wi-Sun

Wi-Sun é um padrão internacional de interoperabilidade com o protocolo IEEE 802.15.4g na camada física e com o protocolo IEEE 802.15.4/4e na camada de ligação de dados. A tecnologia Wi-Sun tem a capacidade de consumir pouca energia, aumentando dessa forma a autonomia da bateria dos dispositivos, o que cumpre o requisito dos dispositivos deste sistema funcionarem durante alguns anos sem ser efetuada qualquer operação de manutenção nos mesmos.

Esta tecnologia é implementada em sistemas de *smart metering* e em infraestruturas de medição avançada AMI, nas áreas da monitorização do consumo de gás, água e eletricidade. Num futuro próximo a tecnologia Wi-Sun será também aplicada em sistemas de comunicação, nomeadamente em sistemas de comunicações móveis de grande cobertura, em sistemas de monitorização de trânsito e em sistemas de recuperação de desastres.

A tecnologia Wi-Sun suporta comunicações bidirecionais e transmissão *multi-hop*. Utiliza a técnica de modulação *Gaussian Frequency Shift Keying* (GFSK) e utiliza uma banda de frequências próxima de 900 MHz, que oferece boas características de propagação. Como suporta comunicação multi-hop a área de cobertura é maior, reduzindo as regiões sem cobertura.

Nos sistemas Wi-Sun com uma grande área de cobertura existem dois tipos de dispositivos: de medição e os coordenadores. No *uplink* os dispositivos de medição enviam os dados recolhidos para o coordenador. No *downlink* o coordenador envia sinais de controlo para os dispositivos de medição.

Existem ainda uns dispositivos na rede designados de *routers* que são responsáveis por estabelecer a comunicação para fora da rede, designadamente com a Internet, possibilitando a cada dispositivo comunicar com um servidor localizado no exterior da rede.

Os dispositivos de medição e os coordenadores utilizam a mesma tecnologia rádio, logo os seus desempenhos e características de comunicação são iguais. Na Figura 7 é apresentado um exemplo de comunicação entre dispositivos numa rede Wi-Sun, utilizando a técnica *multi-hop*, que confere flexibilidade à rede que tipicamente utiliza uma topologia em malha.

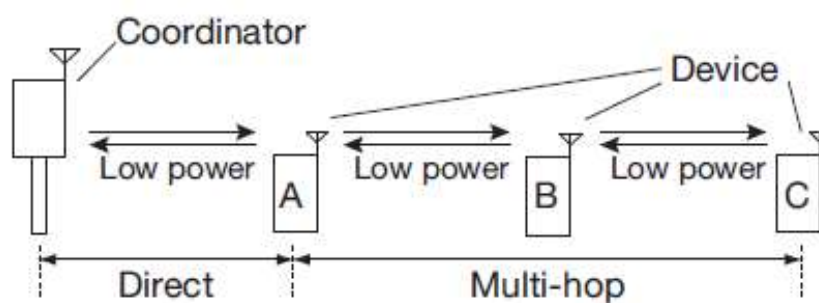


Figura 7 – Sistema Wi-Sun com comunicação *multi-hop*. (Mochizuki, 2016)

EnOcean

EnOcean é uma tecnologia de comunicação sem fios de potência reduzida. O seu funcionamento depende da energia recolhida do ambiente onde o dispositivo se encontra localizado podendo esta ser solar, térmica ou mecânica. Os dispositivos podem ser atuadores, controladores, sensores, *gateways* e centrais de controlo e monitorização. Esta tecnologia é usada em soluções para aplicações de automação de edifícios, sistemas inteligentes de iluminação e em *smart grids*. Esta tecnologia combina conversores de micro-energia com eletrónica de baixo consumo energético, o que torna esta tecnologia mais eficiente quando comparada com outras tecnologias como o ZigBee e o Bluetooth.

A tecnologia EnOcean pode ser implementada sem necessidade do uso de uma bateria, devido à sua capacidade de recolha de energia, o que reduz os custos de implementação e facilita a implementação da rede.

Esta tecnologia é baseada num modelo com quatro camadas, que compreende a camada física, a camada lógica de dados e a camada de rede, as quais são especificadas pelo modelo OSI, sendo a camada de aplicação especificada pela *EnOcean Alliance*.

- Camada física: especifica que os dados podem ser transmitidos utilizando dois protocolos diferentes, o *EnOcean Radio Protocol 1* (ERP1) no qual as mensagens são transmitidas na bandas de frequência dos 315 MHz ou dos 868.3 MHz, com uma velocidade de transmissão de 125 kbps, utilizando a técnica de modulação *Amplitude Shift Keying* (ASK), com um alcance 300 m em linha de vista para ambientes exteriores e de 30 m em ambientes interiores. Pode também ser utilizado o protocolo *EnOcean Radio Protocol 2* (ERP2) no qual as mensagens são transmitidas utilizando a banda de frequência compreendida entre os 902.875 MHz e os 928.35 MHz, utiliza a técnica de modulação *Frequency Shift Keying* (FSK), com uma velocidade de transmissão de 125 kbps, com um alcance igual ao anterior em ambientes interiores e de 300 m em ambientes exteriores.

- Camada de ligação de dados: utiliza a técnica MAC *unslotted ALOHA* em vez da técnica CSMA que permite ao dispositivo transmitir uma mensagem num dado instante e entrar imediatamente em modo de suspensão. A probabilidade de colisão de mensagens é reduzida utilizando mensagens muito curtas e utiliza um mecanismo em que ouve se existem mensagens no canal antes de enviar a mensagem *listen before talk*. Este mecanismo tem a capacidade de atrasar a transmissão da mensagem até ser recebida a trama esperada. É utilizada uma janela temporal de 1 ms para enviar a mensagem, repetindo a transmissão da mensagem até três vezes em janelas temporais diferentes, para garantir que não ocorrem colisões. Na receção da mensagem, esta é decodificada para verificar a sua integridade.

- Camada de rede: esta camada é responsável por aumentar o tamanho da rede EnOcean através de repetidores, quando a qualidade do sinal entre o transmissor e o recetor não atinge o nível de qualidade mínimo necessário. A camada de rede é ainda responsável por evitar colisões entre mensagens, tratar do endereçamento das mensagens e converter as mensagens em mensagens específicas para os dispositivos. (Li, 2014) (Arcari, 2017)

Na Figura 8 é possível observar um exemplo de uma rede de controlo inteligente de iluminação baseado na tecnologia EnOcean.

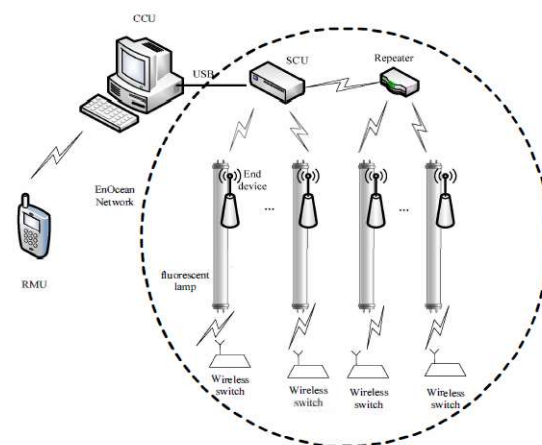


Figura 8 – Rede de controlo inteligente de iluminação utilizando tecnologia EnOcean. (Li, 2014)

Lora e LoraWan

Long-Range™ (LoRa) é uma tecnologia de comunicação rádio de baixa potência que proporciona extensas áreas de cobertura, *Low Power Wide Area Network* (LPWAN).

Na Europa o sistema LoRa utiliza as bandas de frequências ISM nos 868 MHz e nos 433 MHz. A banda de 868 MHz permite débitos compreendidos entre os 250 bps e os 50 kbps. Nos Estados Unidos a banda de frequências utilizada é a 915 MHz, que proporciona débitos compreendidos entre 980 bps e 21.9 kbps. Os transmissores rádio dos dispositivos utilizam o mecanismo de *listen-before-talk* para limitar o débito de mensagens para a rede. A tecnologia LoRa atua na camada física, com o objetivo de prolongar a duração das baterias dos dispositivos sem fios e estabelecer comunicações em grandes áreas, com velocidades de transmissão compreendidas entre 0.3 kbps e os 11 kbps.

O LoRa WAN é um protocolo para estabelecer comunicações que intervêm nas camadas de ligação e de rede do modelo OSI.

Na tecnologia LoRa o sinal é modulado usando uma técnica de espalhamento espectral, que espalha o sinal de informações por um canal com uma largura de banda maior que a do sinal original, este resultado é obtido através da geração de um sinal *chirp* que varia constantemente a frequência, denominado de *Chirp Spread Spectrum* (CSS).

Para garantir a adaptação da velocidade de transmissão e alcance do sinal, são utilizados fatores de espalhamento *spreading factor* (SF) que variam entre 7 e 12 e afetam o gradiente de variação da frequência. Refira-se que as larguras de banda utilizadas variam entre os 125 kHz e os 500 kHz.

Uma rede LoRa WAN é constituída por três tipos de dispositivos, o servidor de rede *Network Server* (NS), que é o coordenador e controlador da rede; os dispositivos de medição que geram e enviam dados para o NS; e as *gateways* que são responsáveis por encaminhar as mensagens entre o NS e os dispositivos de medição, tanto no percurso ascendente como no descendente. Numa rede LoRa é utilizada tipicamente a topologia em estrela, como se pode observar na Figura 9.

No percurso descendente, do NS para os dispositivos de medição, é o NS que escolhe a melhor *gateway* para encaminhar a mensagem para o dispositivo de medição. Esta escolha é realizada com base nos valores da relação sinal ruído *Signal to Noise Ratio* (SNR).

O modo de funcionamento da rede assume que cada dispositivo transmite a mensagem no percurso ascendente para várias *gateways*. As mensagens *downlink* só podem ser recebidas após uma mensagem *uplink* ser enviada, pois a mensagem *uplink* vai iniciar o processo em que o dispositivo de medição e o NS estabelecem dois canais de comunicação entre si, um canal é o definido no dispositivo de medição de acordo com a configuração da rede e o outro canal é escolhido entre o NS e o dispositivo de medição. Somente quando os dois canais estiverem estabelecidos é que a comunicação *downlink* pode ser estabelecida, caso o tempo de estabelecimento dos dois canais exceda o tempo limite a comunicação falha.

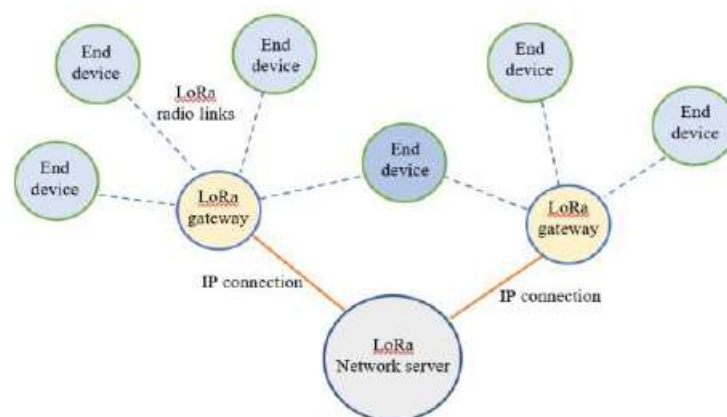


Figura 9 – Topologia típica de uma rede LoRaWAN. (Usmonov, 2017)

Este sistema utiliza dois tipos de mensagens distintas, as mensagens não confirmadas e as mensagens confirmadas. Normalmente numa rede LoRaWAN todas as mensagens *uplink* são enviadas sem garantia de uma receção bem-sucedida. Em situações em que é necessária garantia da receção é utilizada a técnica de *acknowledgment* (ACK) no *uplink*, assim sempre que o ACK não for recebido, o dispositivo de medição reenvia a mensagem. (Jörke, 2017) (Centenaro, 2017) (Usmonov, 2017) (Wang, 2017)

Wi-Fi

Atualmente a tecnologia Wi-Fi é das mais utilizadas nas comunicações sem fios. Esta tecnologia baseia-se na norma IEEE 802.11 e nas suas versões IEEE 802.11b, IEEE 802.11g, IEEE 802.11a e IEEE 802.11n. Dependendo da norma em que se baseia é possível obter velocidade de transmissão de 11 Mbps, 54 Mbps e 300 Mbps. As bandas de frequência de operação compreendem bandas nos 2.4 GHz, 3.5 GHz e 5 GHz, com um alcance até 100 m. (Kuzlu, 2015)

No Quadro 3 é possível verificar as diferenças no que concerne às camadas do modelo OSI das diversas tecnologias apresentadas anteriormente.

Quadro 3 – Camadas de atuação dos sistemas apresentados.

OSI	EnOcean	ZigBee	Wi-Sun	6LowPan	WirelessHart	LoRa
Aplicação	Aplicação	Aplicação	Aplicação	Aplicação	WirelessHART	
Apresentação	EnOcean	ZigBee				
Sessão			6LowPan			
Transporte		IEEE 802.15.4 MAC		IEEE 802.15.4 MAC	LoRa WAN	
Rede			IEEE 802.15.4 PHY			IEEE 802.15.4 PHY
Ligação de dados		IEEE 802.15.4 PHY		IEEE 802.15.4 PHY		
Física						

Quadro 4 – Especificações da camada física dos sistemas apresentados.

Tecnologias	Protocolo da camada física	Bandas de frequências	Velocidade de transmissão	Alcance	Modulação e encriptação	Inter-operabilidade
ZigBee	IEEE 802.15.4 PHY	2.4 GHz	250 kbps	10 m a 100 m		Norma IEEE 802.11.b
		Américas				
		915 MHz	20 kbps			
		Europa				
		868 MHz	250 kbps			
WirelessHart	IEEE 802.15.4,	2.4 GHz	250 kbps		DSSS FHSS	
6LowPan	Serve de camada de adaptação entre o protocolo IPv6 e a camada de ligação de dados que utiliza o protocolo IEEE 802.15.4					
Wi-Sun		Banda de frequências próxima de 900 MHz			GFSK	Protocolo IEEE 802.15.4g
EnOcean		ERP1: 315 MHz 868.3 MHz	ERP1: 125 kbps	ERP1 e ERP2: 300 m (exterior) 30 m (interior)	ERP1: ASK	
		ERP2: 902.875 MHz 928.35 MHz	ERP2: 125 kbps		ERP2: FSK	
LoRa		Europa				
		868 MHz 433 MHz	250 bps a 50 kbps			
		Estados Unidos				
		915 MHz	980 bps a 21.9 kbps			
Wi-Fi	IEEE 802.11b IEEE 802.11g IEEE 802.11a IEEE 802.11n	2.4 GHz 3.5 GHz 5 GHz	11 Mbps 54 Mbps 300 Mbps	100 m		

No Quadro 2 e Quadro 4 são apresentadas as especificações dos sistemas cablados e sem fios usualmente empregues nas redes HAN, e NAN dos sistemas AMI da camada física. Nomeadamente, sobre que protocolo funcionam, as frequências utilizadas para a comunicação, a velocidade de transmissão, o alcance do sinal, as técnicas de modulação e encriptação utilizadas e a interoperabilidade com outros protocolos e normas.

POWER METERS DISPONÍVEIS NO MERCADO

No Quadro 5 é possível comparar as funcionalidades de alguns *power meters* disponíveis hoje em dia no mercado, como as grandezas que cada *power meter* permite ler, a forma como podem ser ligados à rede eléctrica e os canais e protocolos que cada *power meter* utiliza para fazer a comunicação dos dados recolhidos. De entre os *power meters* existentes serão somente apresentados três, dos fabricantes Siemens, Carlo Gavazzi e Schneider Electric, com o intuito de mostrar a variedade existente no mercado em relação a fabricantes e *power meters*, embora as funcionalidades gerais dos *power meters* sejam semelhantes, uma vez que são projetados para a mesma tarefa, a leitura de dados relativos ao consumo de energia numa instalação eléctrica.

SICAM P50 / P55

Este *power meter* consegue medir valores relativos aos valores eficazes da tensão, corrente, potências ativa, reativa e aparente, fator de potência e os harmónicos da tensão e da corrente até ao 21º, THD por fase e a frequência. Estes valores podem ser apresentados no ecrã do *power meter* ou enviados através da porta de comunicação serie. Os valores registados são armazenados na memória de 1 MB do dispositivo, a qual permite armazenar os valores registados até 3 meses em caso de falha de energia.

É possível ligar o dispositivo a uma rede utilizando as seguintes ligações:

- Fase-neutro;
- Três fases equilibradas ou desequilibradas;
- Três fases com neutro.

Para comunicar com outros dispositivos o *power meter* dispõe de uma porta RS485 com um conector *standard* de 9 pinos D-sub. E pode comunicar utilizando os seguintes protocolos de comunicação:

- Profibus DP V1 em conformidade com EN 50170 Volume 2 e Modbus RTU / ASCII;
- Modbus RTU / ASCII e IEC 60870-5-103;

É possível configurar o dispositivo através do menu apresentado no ecrã do *power meter*, através da porta RS485 utilizando o *software* SICAM P Manager. É possível observar este *power meter* na Figura 10.



Figura 10 – Power meter SICAM P50 / P55. (Siemens, 2016)

EM21 72D

É um *power meter* trifásico capaz de efetuar medições das energias consumidas, da corrente (utilizando transformadores de intensidade), da tensão (conectando as fases diretamente ao *power meter* ou através de transformadores de tensão), o mesmo pode ser observado na Figura 11. Para comunicar os valores medidos dispõe de uma porta RS485 com ligação de dois fios, que permite distâncias de comunicação até 1000 m, que utiliza o protocolo MODBUS/RTU para comunicar.



Figura 11 – Power meter EM21 72D. (Gavazzi)

Ao ligar o dispositivo a uma instalação é possível efetuar os seguintes tipos de ligação:

- 3 fases equilibradas ou desequilibradas;
- 3 fases mais neutro;
- 3 fases sem neutro;
- 2 fases mais neutro;
- 1 fase com neutro.

ION8650 meter

Este *power meter*, que pode ser observado na Figura 12, é capaz de efetuar medições referentes ao consumo e qualidade da energia elétrica numa instalação, efetuando para o efeito as seguintes medições:

- Potências: aparente, ativa e reativa total e por fase;
- Tensão;
- Corrente;
- Frequência;
- Fator de potência total e por fase.



Figura 12 – Power meter ION8650 *meter*. (Electric, 2018)

Tem a capacidade de armazenar todos os valores medidos numa memória de 128 MB, assim como a capacidade de os comunicar para outro dispositivo através dos seguintes métodos:

- Ethernet RJ45;
- Infravermelhos;
- Bloco de terminais RS485;
- RS485/RS232 SUB-D 9.

Utilizando protocolos de comunicação como:

- TCP/IP;
- MODBUS RTU ou TCP.

Este *power meter* permite as seguintes topologias ao ser inserido numa instalação elétrica:

- 1 fase mais neutro;
- 3 fases sem neutro;
- 3 fases com neutro.

Quadro 5 – Especificações de *power meters* comerciais. (Electric, 2018) (Gavazzi) (Siemens, 2016)

Power meters	Grandezas lidas	Ligação à rede de energia	Comunicação
EM21 72D CARLO GAVAZZI	-Energias; -Tensões simples e compostas; -Correntes em cada fase; -Potências; -Fator de potência; -Frequência	-3 fases, 4 fios; -3 fases, 3 fios; -2 fases, 3 fios; -1 fase, 2 fios	-Canal RS485; -Protocolo MODBUS
ION8650 meter Schneider Electric	-Potências totais e por fase; -Tensões; -Correntes; -Frequência; -Fator de potência total e por fase;	-3 fases mais neutro; -3 fases; -1 fase mais neutro	-Ethernet RJ45; -Infrared; -RS485 terminal block; -RS485/RS232 SUB-D 9
SICAM P50/P55 SIEMENS	-Energias; -Tensões simples e compostas; -Correntes em cada fase; -Potências; -Fator de potência; -Frequência	-1 fase mais neutro; -3 fases mais neutro; -3 fases	-Modbus RTU / Profibus / IEC 60870-5-103

CONCLUSÕES

Existem várias tecnologias de comunicações possíveis de serem utilizadas nas redes de *smart grid*, em vários níveis como nas HAN ao nível do consumidor, ao nível da distribuição e nas WAN ao nível da transmissão de energia. As tecnologias aplicadas podem ser sem fios ou cabladas conforme o ambiente, em que os dispositivos se encontram sendo que se verifica uma maior utilização das tecnologias sem fios devido à sua flexibilidade e facilidade na instalação permitindo a sua instalação em sítios onde o acesso às tecnologias cabladas era difícil.

No entanto, dado que não existe limitações de espaço nem obstáculos entre o *power meter* e o concentrador no sistema desenvolvido optou-se pela utilização de uma tecnologia cablada.

Para garantir a interoperabilidade numa rede *smart grid*, existem várias especificações ao desenvolver dispositivos para uma rede *smart grid* e a desenvolver a rede, de forma geral estas tecnologias são desenvolvidas com base no modelo OSI e seguem um conjunto de critérios que especifica as forma de atuar dos dispositivos, como as suas funcionalidades na rede *smart grid*, obtenção e leitura de dados, monitorização da qualidade dos dados sobre a energia e facilitar ações de resposta face à procura de energia.

IMPLEMENTAÇÃO DO SISTEMA

SISTEMA DE AQUISIÇÃO DE DADOS

O presente capítulo apresenta as soluções técnicas desenvolvidas durante o presente projeto com o intuito de implementar uma solução integrada de *smart metering* para aplicação em instalações elétricas residenciais e industriais baseada nas tecnologias e dispositivos apresentados anteriormente.

Da análise dos requisitos do sistema optou-se pela utilização do *power meter* 710 da *Schneider* e pelo desenvolvimento de um sistema de aquisição de dados, concentrador, que faz parte integrante de um sistema de informação o qual compreende diversos módulos de *hardware*, responsáveis pela aquisição dos valores de consumo de energia de uma instalação elétrica e pela comunicação dos dados adquiridos para uma base de dados usando para o efeito um microcontrolador e/ou um computador pessoal.

Numa primeira fase deste projeto utilizou-se o *power meter* selecionado e um PC portátil como unidade de aquisição de dados, como se pode observar na Figura 13. O objetivo desta primeira solução compreendia unicamente a validação do conceito do sistema tendo-se implementado uma aplicação desenvolvida em Matlab para efetuar a comunicação entre o *power meter* e o PC; estudar as mensagens enviadas pelo *power meter* em resposta aos comandos de leitura e configuração, comunicar e registar os valores adquiridos numa base de dados externa, tendo em conta que este *software* possui um conjunto de funções que facilitam a análise das mensagens recebidas.

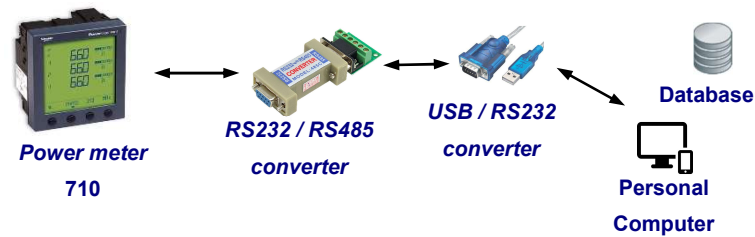


Figura 13 – Sistema de aquisição de dados de consumo de energia.

POWER METER 710

O *power meter* 710 da Schneider é o componente principal do sistema o qual possui um microcontrolador interno com registos de medida e interfaces de comunicação. Dada a variedade de registos que o *power meter* disponibiliza para consulta, verificou-se que nem todos possuíam relevante interesse para este projeto. Assim sendo, após a análise dos seus registos foram seleccionados os registos relativos às seguintes grandezas cujo valor se pretende analisar e registar:

- Energia aparente, ativa e reativa;
- Potência aparente, ativa e reativa;
- Fator de potência;
- Tensões nas fases A, B e C;
- Correntes nas fases A, B e C;
- Frequência;
- THD da corrente nas fases A, B e C;
- THD da tensão nas fases A, B e C.

Tendo em conta que os registos escolhidos para realizar a leitura são do tipo *float*, e seguem a norma IEEE 754, logo o valor devolvido pelo *power meter* terá (32 bits) que serão agrupados e convertidos para um número real.

No que concerne à instalação do *power meter* esta é dividida em três passos, que a seguir se descrevem:

- Definição do modo de ligação à instalação elétrica e ligação à mesma. Este passo está dependente do valor da tensão na rede e do tipo de rede, utilizada: monofásica ou trifásica;
- Ligação do *power meter* à rede de comunicação, através da porta RS485;
- Configuração dos Transformadores de Intensidade (TI's) e Transformadores de Tensão (TT's), da frequência da rede, do tipo de rede, da corrente média, definição das palavras passe e dos parâmetros de comunicação, como o endereço do *power meter* na rede, a taxa de transmissão (*baudrate*) e a existência ou não de *bits* de paridade.

Na instalação do *power meter* foi utilizado o esquema de ligação monofásico fase-neutro de dois fios, com um TI.

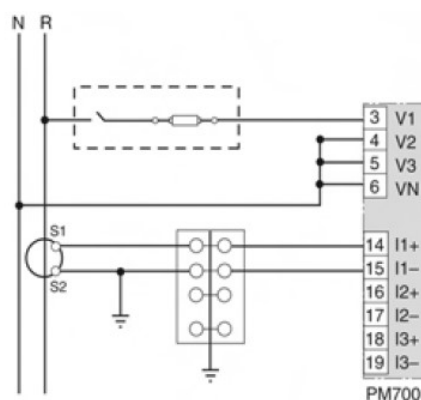


Figura 14 – Rede monofásica fase-neutro de 2 fios, 1 TI. (Electric, 2004)

De acordo com o esquema de ligações utilizado na instalação, os parâmetros de configuração do *power meter* vão mudar, nomeadamente o tipo de rede. Neste caso no menu de configuração do *power meter* foi necessário escolher o tipo de rede número 10, que corresponde ao tipo de rede apresentado na Figura 14.

Após proceder à realização dos passos anteriores o *power meter* encontra-se operacional e pronto a efetuar medições cujos dados serão enviados através da rede RS485 para um microcontrolador responsável pela solicitação do envio dos valores contidos nos registos referentes às leituras efetuadas e pela sua escrita numa base de dados, a qual faz parte integrante da solução *Information and Communication Technology* (ICT).

Após cada leitura, o *power meter* armazena os valores correspondentes, nos registos internos existentes para essa função. O endereço interno de cada um destes registos abrange uma gama de valores compreendida entre o número 4000 até ao registo número 4128. Os registos compreendidos entre 7000 e 7006 têm informação relativa à configuração do dispositivo, número do mesmo, endereço e versão do *software*.

O *power meter* utilizado permite executar comandos de leitura dos registos de manutenção, leitura dos registos de entrada, predefinição de registos individuais e de vários registos em simultâneo. Utiliza um barramento RS485 e o protocolo Modbus para efetuar a comunicação. A utilização deste protocolo é vocacionada para aplicações industriais, nomeadamente comunicações em redes industriais. Ao utilizar o barramento descrito para interligar os *power meters* numa rede é possível ligar até 31 dispositivos em simultâneo. A Figura 15 representa um exemplo da ligação em simultâneo dos *power meters* num barramento. Na solução implementada o endereço dos *power meters* assume valores entre 1 e 247, logo para inserir mais dispositivos numa rede será necessário implementar outro tipo de ligação entre os dispositivos, diferente da ligação em simultâneo num único barramento.

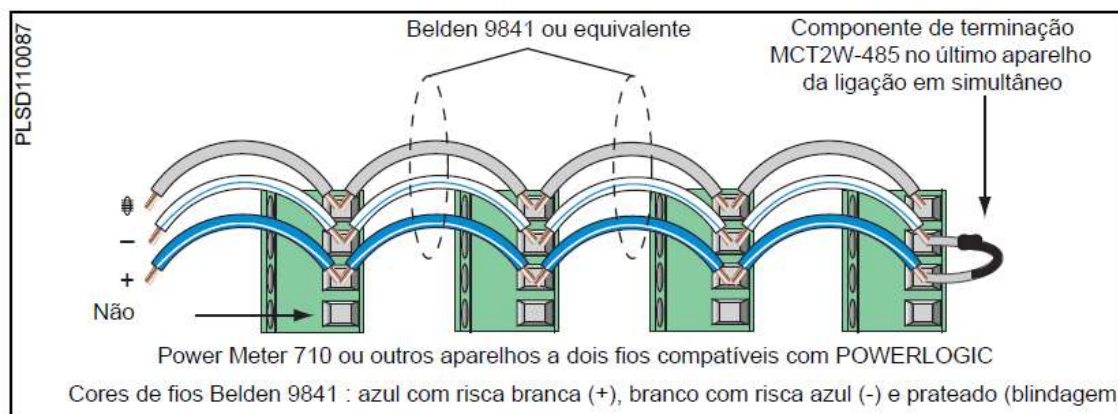


Figura 15 – Ligação em simultâneo de aparelhos a dois fios. (Electric, 2004)

CONVERSOR RS232 / RS485

O conversor RS232-RS485 é um equipamento de *hardware* destinado à conversão de dados entre os padrões RS232 de três fios (Tx, Rx e Gnd) e o padrão RS485 de dois fios (Data+ e Data-), sendo esta independente do protocolo utilizado na comunicação. O conversor tem a capacidade de ajustar e controlar o fluxo de dados e permite velocidades de comunicação até 115,2 kbps. A sua instalação requer apenas uma ligação à interface RS232 e outra à interface

RS485, e necessita de ser alimentado em corrente contínua, com uma tensão cujo valor se encontra no intervalo compreendido entre os 9 V e os 30 V, permitindo correntes até 300 mA. O canal de comunicação pode usar um dos três modos de comunicação comuns: *simplex*, *half-duplex* ou *full-duplex*.

MySQL

MySQL é um sistema de gestão de bases de dados *open source*, necessário para definir e parametrizar a base de dados responsável pelo armazenamento, acesso e processamento da informação recolhida e registada na base de dados.

As bases de dados desenvolvidas sobre o sistema MySQL são relacionais. Isto significa que utilizam diversas tabelas para armazenar informação, sendo que na definição da base de dados foram estabelecidas as relações entre as tabelas. Refira-se que esta propriedade torna o acesso à base de dados mais rápido. A linguagem utilizada para executar operações numa base de dados desenvolvida sobre o sistema MySQL é a linguagem *Structured Query Language* (SQL), que é a linguagem mais utilizada em bases de dados. Esta linguagem existe desde 1986, sendo que a versão utilizada é a versão desenvolvida no ano de 2003.

Este sistema é do tipo cliente/servidor, o que significa que suporta várias aplicações de diferentes clientes. (MySQL, 2018)

APACHE

Rob McCool desenvolveu em 1995 um *software* de servidores HTTP *daemon*. Alguns anos depois vários intervenientes corrigiram erros no *software* original e surgiu a necessidade de criar uma distribuição comum a todas as melhorias feitas no *software* HTTP *daemon*, através de oito intervenientes para o efeito foi criado o Apache Group. Em Abril de 1995 foi lançado a primeira versão oficial de servidor Apache. No dia 1 de Dezembro de 1995 é lançada a versão 1.0 do servidor Apache com novas funcionalidades e correção de erros existentes nas versões anteriores. De acordo com a empresa de serviços de *internet* Netcraft o servidor Apache ainda é hoje em dia o servidor mais utilizado na internet. (Apache2, 2018) (Apache Notes, 2018)

A versão mais recente deste *software* é a versão 2.4.33, sendo esta a versão mais estável do servidor Apache até a data. Segundo a Netcraft, em Maio de 2010 o Apache foi utilizado para suportar aproximadamente 54.68 % dos sites ativos na internet.

O servidor Apache pode ser instalado em vários sistemas operativos, como Windows e o Linux. A partir do servidor Apache será então possível processar páginas em linguagem PHP, comunicar com uma base de dados, responder a pedidos de clientes para consulta de uma determinada página de um *site*, entre outras funcionalidades. (Apache2, 2018) (Apache Notes, 2018)

PROTOCOLO MODBUS

Modbus é um protocolo de comunicação série criado em 1979, pelo fabricante de equipamentos Modicon. Este protocolo é dos mais antigos e dos mais utilizados em redes de controladores lógicos programáveis (PLC), para transmissão de informação entre dispositivos eletrónicos, em canais de comunicação série. (SModbus, 2018)

O protocolo Modbus quando utilizado sobre um canal RS485 efetua comunicações ponto-multiponto, isto significa que o servidor envia um pedido para a rede e o *power meter* com o endereço igual ao do pedido responde ao servidor. Utiliza portanto uma metodologia de servidor escravo. Na comunicação é utilizada uma mensagem, a qual se representa na Figura 16, designada por *Application Data Unit* (ADU) que tem diversos campos nomeadamente: o endereço do escravo, a mensagem *Protocol Data Unit* (PDU) e o código de erro (*Error Check*).

A mensagem PDU é constituída pelo campo *Function code*, que designa o comando que se pretende efetuar: leitura ou escrita e pelo campo *Data* que neste caso contém o endereço do primeiro registo do *power meter* a ser consultado e o número de registos que se pretende consultar. A resposta do *power meter* para o servidor é idêntica à mensagem enviada pelo servidor, as únicas alterações são as seguintes, onde era inserido o primeiro registo a ser lido é agora inserido o número de registos que o *power meter* consultou, e no campo onde era inserido o número de registos a serem consultados vêm agora os valores obtidos de cada um dos registos consultados neste pedido.

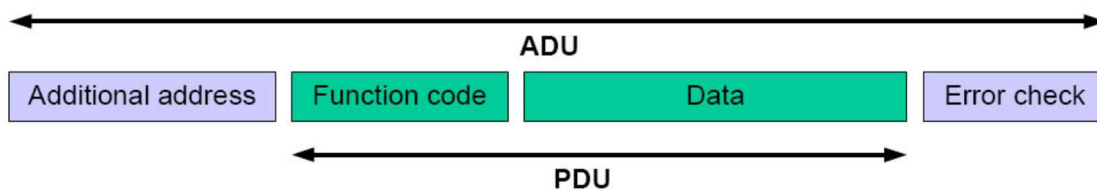


Figura 16 – Estrutura da mensagem em formato Modbus. (Modbus, 2018)

Exemplo de leitura dos registos de entrada

Mensagem de pedido:

Quadro 6 – Mensagem de pedido.

Endereço do dispositivo	Comando	Endereço do 1º registo		Quantidade de registos		CRC	
01	04	03	E7	00	02	C1	B8

A mensagem indica que vão ser lidos, dois registos de entrada, do *power meter* com o endereço 1, sendo que o primeiro registo a ser lido é o relativo ao endereço 999.

Mensagem de resposta:

Quadro 7 – Mensagem de resposta.

Endereço do dispositivo	Comando	Nº de registos devolvidos		Registos devolvidos	CRC	
01	04	00	08	XXXXXXXXXXXXXXXXXX	XX	XX

O dispositivo devolve oito registos, sendo que cada registo é representado por uma *word* (16 bits), que estão no campo *Data* da PDU, como demonstra a Figura 16.

Nota: Os valores apresentados nas tabelas anteriores estão no sistema de numeração hexadecimal.

APLICAÇÃO DESENVOLVIDA EM MATLAB

Para efetuar a comunicação com o *power meter* foi primeiro necessário definir todas as mensagens pedido a serem enviadas para o *power meter*, relativas às grandezas que se

pretendem consultar. Um exemplo das mensagens enviadas e da estrutura das mesmas pode ser encontrado na Figura 16 e no Quadro 6. Posteriormente foi necessário criar um objeto com os parâmetros da comunicação. Neste caso utilizou-se os seguintes parâmetros ‘COM3’, baudrate igual a ‘9600’ sem qualquer tipo de paridade. De seguida foi inicializado o canal de comunicação série através da função *fopen()* onde o parâmetro de entrada era o objeto definido anteriormente.

```
**** Inicialização da Porta Serie ****  
s = serial('COM3', 'BaudRate', 9600, 'DataBits', 8, 'Terminator', '');  
fopen(s);
```

Script 1 – Inicialização da porta série.

Após a inicialização do canal de comunicação é possível efetuar os pedidos necessários para a aquisição dos dados existentes em registos específicos do *power meter*. Para o efeito, definiu-se que a leitura dos mesmos seria feita com uma periodicidade de cinco segundos, com exceção dos valores de energia que seriam recolhidos de minuto em minuto. Para efetuar um pedido ao *power meter* foi criada uma função designada de *readPM*, esta função começa por escrever para o canal de comunicação a mensagem referente aos registos que pretendemos ler do *power meter*. De seguida é efetuada a leitura do canal de comunicação, que contém a mensagem de resposta do *power meter*.

```
**** Função de leitura dos registos do PM ****  
function [read_value] = readPM( s, msg)  
  
    fwrite(s, msg, 'uint8');  
    data = fread(s, 9, 'uint8');  
    val_hex = [dec2hex(data(4), 2) dec2hex(data(5), 2)  
dec2hex(data(6), 2) dec2hex(data(7), 2)];  
    read_value = typecast(uint32(hex2dec(val_hex)), 'single');  
  
end
```

Script 2 – Função de leitura dos registos do *power meter*.

A mensagem resposta, é uma sequência de *bytes*, a qual é colocada numa variável, de forma a agrupar os *bytes* recebidos numa *string* sendo esta de seguida processada de forma a recolher apenas os campos mensagem que contêm informação relativa aos dados solicitados ao *power meter*. Este processamento é realizado, extraindo apenas os quatro *bytes* da string referentes ao valor *float* que se pretende ler sendo realizada a conversão desses quatro *bytes*

para um valor do tipo *float*. No Matlab esta conversão é feita agrupando os quatro *bytes* numa *word* de 32 bits e de seguida efetuando um *typecast* para o tipo *uint32* sendo assim obtido o valor pretendido.

Os valores são posteriormente registados na base de dados, no programa são armazenados num vetor chamado *valores_BD*, utilizando a função *readPM* definida anteriormente.

```

%*** Recolher valores do PM ***
%...
%ler potencia aparente total
valores_BD(1,4) = readPM( s , msg4);

%ler potencia activa total
valores_BD(1,5) = readPM( s , msg5);

%ler potencia reactiva total
valores_BD(1,6) = readPM( s , msg6);
%...

```

Script 3 – Recolher valores do *power meter*.

O programa efetua então uma conexão com a base de dados do sistema, através da função *database*, de forma a armazenar os dados recolhidos, para ser possível enviar pedidos de registo de valores. A função *datainsert* insere os valores recolhidos do *power meter* na base de dados. No fim de cada registo a conexão com a base de dados é encerrada para libertar o canal de comunicação.

```

%*** Guardar na base de dados ***
conn = database('mydb','root',' ');
colnames =
{'energy_S','energy_P','energy_Q','power_S','power_P','power_Q','power_fa
c','voltage_A','voltage_B','voltage_C','current_A','current_B','current_C
','frequency','THD_currA','THD_currB','THD_currC','THD_voltA','THD_voltB'
,'THD_voltC','data_hora'};
datainsert(conn,'pm_values',colnames,{valores_BD(1,1) valores_BD(1,2)
valores_BD(1,3) valores_BD(1,4) valores_BD(1,5) valores_BD(1,6)
valores_BD(1,7) valores_BD(1,8) valores_BD(1,9) valores_BD(1,10)
valores_BD(1,11) valores_BD(1,12) valores_BD(1,13) valores_BD(1,14)
valores_BD(1,15) valores_BD(1,16) valores_BD(1,17) valores_BD(1,18)
valores_BD(1,19) valores_BD(1,20) date})
close(conn)

```

Script 4 – Guardar valores na base de dados.

IMPLEMENTAÇÃO UTILIZANDO UM MICROCONTROLADOR RASPBERRY PI

Após se ter validado o conceito procedeu-se à substituição da unidade de aquisição constituída pelo PC e pelo programa em Matlab por uma unidade que compreende um Raspberry Pi; um cartão de memória *flash*; uma PEN e um *shield* conversor TTL/RS232.

Como unidade de aquisição de dados nesta segunda versão do sistema utilizou-se o Raspberry Pi modelo B, tendo em conta as suas especificações, elevado desempenho e baixo custo para além do facto de possuir diversas interfaces e de permitir de uma forma simples a inclusão de módulos externos (*shields*).

De seguida são apresentadas as características com maior importância do Raspberry PI 1 modelo B.

Quadro 8 – Características do Raspberry Pi Model B.

Modelo	Raspberry Pi 1 B
System-on-Chip (SoC)	BCM2835
Memory chip (ROM)	256 MB
Alimentação	5V 1A
Processador	ARM11
Sistema operativo	Linux (Raspbian, Ubuntu, Windows 10 IoT Core, etc)
Saídas de vídeo áudio	HDMI (Full HD 1920x1080) e DSI/RCA
Portas USB	2 portas, 2.0
Conexão Ethernet	Porto RJ45, protocolo TCP/IP
Porto General-Purpose Input/Output (GPIO)	26 pinos, UART, I ² C, SPI Bus

Das características apresentadas as de maior interesse para este projeto são nomeadamente a capacidade de ter duas portas USB, o porto GPIO e o sistema operativo Raspbian. Estes elementos e características são essenciais para o Raspberry Pi executar a recolha de dados pretendida, de uma forma eficiente e eficaz.

Para incorporar o Raspberry Pi no sistema foi necessário configurar os parâmetros de comunicação, permitir a utilização da porta série do microcontrolador e adaptar o porto GPIO do Raspberry Pi para ser possível efetuar uma ligação com um canal de comunicação

RS232, que por sua vez é ligado a um adaptador de RS232 para RS485 para conseguir efetuar comunicações com o *power meter*.

A configuração dos parâmetros de comunicação é efetuada no programa desenvolvido, que o Raspberry Pi executa a quando da sua inicialização. A autorização para utilizar a porta série é dada uma vez quando se inicia o Raspberry Pi pela primeira vez. Embora o microcontrolador disponibilize uma porta série, está funciona com o nível de tensão de 3.3 V (TTL) e o canal série RS232 funciona com um nível de tensão de 5V. Assim sendo, é necessário utilizar um conversor adicional para converter os níveis de tensão permitindo assim a ligação do porto série do Raspberry Pi com o canal série que irá estar conectado com o *power meter*. Por isso é utilizado um conversor TTL/RS232, que passa o nível de tensão de 3.3 V para 5 V tornando possível a comunicação através de um canal RS232.

Para armazenar os dados recolhidos o Raspberry Pi terá de estar conectado a uma rede *Ethernet* para que seja possível aceder e efetuar o armazenamento de dados na base de dados.

De forma a permitir o armazenamento dos dados recolhidos foi necessário incluir o ficheiro *curl/curl.h* para ser possível comunicar com o servidor com a base de dados, sendo a comunicação realizada através da porta *Ethernet* do Raspberry Pi. Os dados recolhidos são posteriormente enviados para o servidor, para o *Uniform Resource Locator* (URL) especificado através do método GET, para um ficheiro PHP, que comunica com a base de dados e armazena os dados recebidos. O programa que executa o processo de recolha de dados e o seu armazenamento na base de dados foi implementado utilizando a linguagem de programação C, tendo em conta a sua versatilidade e o facto de ser uma linguagem de programação muito disseminada na programação de microcontroladores. A linguagem PHP foi utilizada no lado do servidor, tendo em conta que é uma linguagem de uso comum na programação em servidores.

O *software* desenvolvido foi dividido em três funções.

A função *read_val* responsável pelo envio do comando de leitura pela porta série e recolha do resultado do pedido. Para a inicialização do canal série e das variáveis necessárias ao funcionamento do *software* é definida uma função (*PM_710_read*) a qual invoca a função *read_val* para recolher dados, e posteriormente a função *PM_710_read* envia os dados recolhidos para o servidor para serem armazenados na base de dados.

De forma a aumentar a redundância no armazenamento dos dados recolhidos foi implementada uma funcionalidade que procede ao registo dos dados num dispositivo de armazenamento externo, sempre que não seja possível comunicar e registar os dados na base de dados, devido a problemas no canal de comunicação ou indisponibilidade do servidor. Esta funcionalidade evita a perda dos dados adquiridos e permite disponibilizar as leituras não só numa base de dados mas também noutra formato de armazenamento digital.

De igual forma ao realizado anteriormente pela aplicação em Matlab todos os dados são adquiridos com uma periodicidade de cinco segundos, com exceção dos valores da energia que são recolhidos a cada minuto.

No programa é utilizada a biblioteca *termios.h* que disponibiliza um conjunto de funções que permite iniciar um canal série, na porta série do Raspberry Pi. O canal é iniciado na localização */dev/ttyAMA0*, permitindo a escrita e leitura de dados, isto é para permitir o envio das mensagens ao *power meter* e a receção da resposta com os dados.

```
/** Inicialização do canal série */  
int uart0_filestream = -1;  
uart0_filestream = open("/dev/ttyAMA0", O_RDWR | O_NOCTTY | O_NDELAY);
```

Script 5 – Inicialização do canal série.

Para efetuar uma leitura dos dados é necessário utilizar a função *read_val* que foi criada com o propósito de efetuar pedidos ao *power meter*. Esta função escreve para o canal série a mensagem referente ao pedido que se pretendem realizar e de seguida fica à espera que a mensagem resposta apareça no canal série.

A função *write* retorna o valor “-1” sempre que ocorre um erro ao escrever para o canal série. Se o valor retornado pela função *write* for maior ou igual a zero significa que a escrita para o canal foi bem-sucedida e representa o número de *bytes* que foram enviados com sucesso.

De seguida o programa espera pela resposta do *power meter* no canal série utilizando a função *usleep*, caso não apresente qualquer tipo de resposta por parte do *power meter*, então assume-se que ocorreu um erro.

Caso a receção da resposta seja bem-sucedida é necessário determinar o número de *bytes* recebidos e verificar se o primeiro campo da resposta é igual ao endereço do *power meter* que se pretendia contactar. Caso o valor no segundo campo da mensagem resposta, seja igual

ao código da mensagem pedido, tal como foi explicado anteriormente, então a mensagem é válida e passa-se para a sua descodificação.

Tendo em conta que todos os valores recebidos, na implementação realizada, são do tipo *float* e seguem a norma IEEE 754, logo para serem enviados corretamente são necessários quatro *bytes*. De acordo com a referida norma os quatro *bytes* são processados de forma a obter um único valor do tipo *float*, esse valor é retornado pela função *read_val* e será processado na função *PM_710_read* de forma a ser armazenado na base de dados.

Para implementar as funções descritas foi criada a função *PM_710_read* que tem como parâmetro de entrada um número inteiro que serve para controlar a leitura dos valores de energia, este valor será explicado mais à frente quando for apresentado na função *main*.

A função *PM_710_read* primeiro define os parâmetros da comunicação do canal série, designadamente o baudrate a 9600 bits por segundo, a paridade (sem paridade) e que os bits serão agrupados em grupos de oito, formando assim um *byte*.

As *strings* que contêm as mensagens pedido que serão enviadas ao *power meter* no formato hexadecimal, cuja estrutura foi explicada anteriormente são inicializadas, juntamente com as variáveis do tipo *float* que irão receber os valores devolvidos pelo *power meter*, assim como as variáveis responsáveis pela aquisição da data e hora da recolha dos dados e uma *string* que será utilizada na escrita de valores para o dispositivo de armazenamento externo.

```

/** Inicialização das variáveis de comunicação */
struct termios options;
tcgetattr(uart0_filestream, &options);
options.c_cflag = B9600 | CS8 | CLOCAL | CREAD;
options.c_iflag = IGNPAR;
options.c_oflag = 0;
options.c_lflag = 0;
tcflush(uart0_filestream, TCIFLUSH);
tcsetattr(uart0_filestream, TCSANOW, &options);
//...
unsigned char ES[]={0x01, 0x04, 0x03, 0xE9, 0x00, 0x02, 0xA0, 0x7B};
unsigned char EP[]={0x01, 0x04, 0x03, 0xE7, 0x00, 0x02, 0xC1, 0xB8};
unsigned char EQ[]={0x01, 0x04, 0x03, 0xEB, 0x00, 0x02, 0x01, 0xBB};
//...
double energy_S=0, energy_P=0, energy_Q=0;
time_t rawtime, start_time;
struct tm *info;
char str[255];
FILE *fp;
char pwd[]="/home/pi/usb/dados_pm.csv";

```

Script 6 – Inicialização das variáveis de comunicação.

Na fase seguinte é chamada a função *read_val* onde se passa como parâmetros uma mensagem ADU de cada vez para pedir os valores ao *power meter*.

Depois com a função *localtime* incluída na biblioteca *time.h* é recolhida a data e hora da leitura e de seguida é construída a *string query* que será enviada para a base de dados.

```

/** Recolha dos valores do power meter e da data e hora **
//.../
power_S = read_val(uart0_filestream, &PS[0]);
power_P = read_val(uart0_filestream, &PP[0]);
power_Q = read_val(uart0_filestream, &PQ[0]);
//.../
time( &rawtime );
while( difftime(rawtime, start_time)<5 )
    time( &rawtime );
info = localtime( &rawtime );

```

Script 7 – Recolha dos valores do *power meter* e da data e hora.

Após a aquisição dos dados do *power meter*, o primeiro passo para registar os valores na base de dados compreende o estabelecimento da ligação ao servidor da base de dados, através do URL, que contém o endereço do servidor e do ficheiro que guarda os valores na base de dados, *smartgridtiago.hopto.org/appAndroidPHP_saveRPI.php* e utilizando o método GET da linguagem PHP os valores recolhidos são passados no mesmo URL.

```

/** Inicialização da comunicação com a base de dados **/
curl_global_init(CURL_GLOBAL_DEFAULT);
sprintf(str,
"http://smartgridtiago.hopto.org/appAndroidPHP_saveRPI.php/?energy_S=%f
&data_hora=%d-%d-%d+%d:%d:%d&PM_address=%d",energy_S,
ano,mes,dia,hora,minuto,segundo, PM_address);
curl_easy_setopt(curl, CURLOPT_URL, str);

```

Script 8 – Comunicação com o servidor com a base de dados.

Para efetuar o registo dos valores adquiridos na base de dados existe ficheiro *appAndroidPHP_saveRPI.php* no servidor que recebe através do método GET os valores enviados pelo Raspberry PI através do URL. Este ficheiro conecta-se com a base de dados e depois envia uma *query SQL* do tipo *INSERT* com os dados a guardar na base de dados. Após o registo retorna uma mensagem a confirmar se o registo dos dados foi bem-sucedido ou se ocorreu um erro.

```

/** Registo de valores na base de dados */
$con = mysqli_connect("localhost","root","","pm_db");

if(isset($_GET["energy_S"]) && isset($_GET["data_hora"]) &&
isset($_GET["PM_address"]))
{$sql= "INSERT INTO pm_values (energy_S, data_hora, PM_address) values
('".$_GET["energy_S"]."', '".$_GET["data_hora"]."',
".$_GET["PM_address"]."', " );
$rs = mysqli_query($con, $sql) or die(mysqli_error());}

```

Script 9 – Registo de valores na base de dados.

Por fim é aberto o ficheiro *dados_pm.csv* no dispositivo de armazenamento externo, com a função *fopen*, se o ficheiro não existir este é criado, se já existir os dados recolhidos são acrescentados aos dados existentes no ficheiro e no fim da escrita dos valores o ficheiro é encerrado com a função *fclose* sendo o canal série encerrado com a função *fclose*.

```

/** Registo dos valores na PEN */
fp = fopen(pwd,"a+");

if(fp==NULL)
printf("Impossivel abrir o
ficheiro %s\n",pwd);
else
{
printf("Ficheiro aberto com
sucesso\n");
fseek(fp, 0L, SEEK_END);
if (ftell(fp)==0)

fprintf(fp, "\"energy_S\";\"energy_P\";\"energy_Q\";\"
power_S\";\"
power_P\";\"power_Q\";\"power_fac\";\"voltage_A\";\"voltage_B\";\"voltage
_C\";\"current_A\";\"current_B\";\"current_C\";\"frequency\";\"THD_currA\
\";\"THD_currB\";\"THD_currC\";\"THD_voltA\";\"THD_voltB\";\"THD_voltC\";\"
data_hora\"\\n");

fprintf(fp, \"%f\";\"%f\";\"%f\";\"%f\";\"%f\";\"%f\";
\"%f\";\"%f\";\"%f\";\"%f\";\"%d-%d-%d %d:%d:%d\\n\",energy_S, energy_P,
energy_Q, power_S, power_P, power_Q, power_fac, voltage_A, voltage_B,
voltage_C, current_A, current_B, current_C, frequency, THD_currA,
THD_currB, THD_currC, THD_voltA, THD_voltB, THD_voltC,
(*info).tm_year+1900, (*info).tm_mon+1, (*info).tm_mday, (*info).tm_hour, (*i
nfo).tm_min, (*info).tm_sec);

fclose(fp);
}
//----- CLOSE THE UART -----
close( uart0_filestream);

```

Script 10 – Registo dos valores no dispositivo de armazenamento externo.

A função *main* é a função principal do programa e nesta função é declarada a variável *num* do tipo *int*, com o valor zero que vai servir para controlar quando é realizada a recolha dos valores relativos às energias ativa, reativa e aparente. É iniciado um ciclo infinito onde é chamada a função *PM_710_read*, onde é passado como parâmetro a variável *num*, esta variável é incrementada em uma unidade a cada iteração do ciclo e depois toma o valor do resto da sua própria divisão por doze. Apenas quando a variável *num* tiver o valor zero é que na função *PM_710_read* serão realizadas as leituras das energias aparente, ativa e reativa. Utiliza-se a divisão por doze porque as leituras são feitas a cada cinco segundos e como só queremos ler o valor das energias ativa, reativa e aparente a cada minuto, então significa que a cada doze leituras dos valores devolvidos pelo *power meter*, é que vamos ler o valor das energias.

SISTEMA DE INFORMAÇÃO

Após se ter procedido à instalação do(s) *power meter(s)* foi necessário desenvolver um sistema de informação responsável pelo armazenamento e gestão dos dados recolhidos os quais serão utilizados posteriormente por duas aplicações direcionadas aos seus utilizadores e/ou clientes. A arquitetura de ICT proposta, representada na Figura 17, contempla a existência dos seguintes elementos: um microcontrolador Raspberry Pi 2 com o sistema operativo Raspbian; um servidor de páginas *web* com uma base de dados associada e um conjunto de clientes que acedem à informação recolhida através de diversas plataformas.

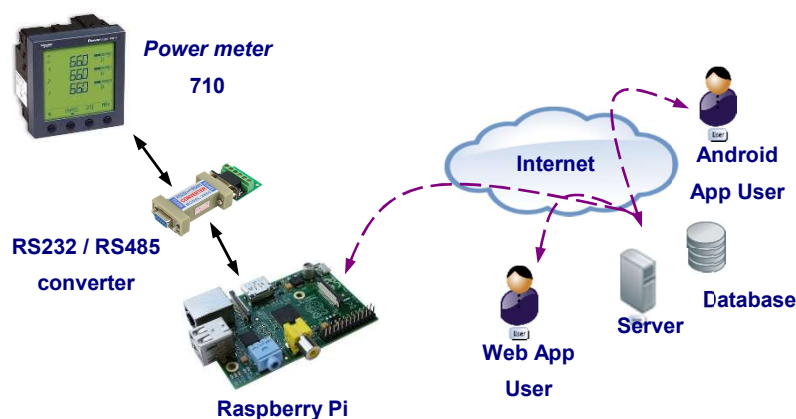


Figura 17 – Arquitetura ICT do sistema desenvolvido.

O servidor possui três pacotes instalados: o Apache, que tem por função alojar e apresentar as páginas *web*; o MySQL, que é responsável por criar uma base de dados e guardar os dados enviados pelo Raspberry Pi e por fim o phpMyAdmin, que apresenta uma interface gráfica para se proceder à gestão da base de dados.

A ligação ao servidor que suporta a base de dados é realizada usando o protocolo TCP / IP. Como tal, é necessário saber o endereço IP do servidor, de forma a registar a informação corretamente. Caso seja necessário, pode-se alojar no servidor um serviço *Domain Name Service* (DNS) para estar acessível através de um URL. Desta forma, caso o IP do servidor se altere, o serviço atualiza automaticamente o seu endereço.

Na arquitetura ICT apresentada a base de dados assume um papel crucial, pois efetua o armazenamento dos dados indispensáveis ao funcionamento do sistema. O sistema tem como objetivo adquirir informação referente ao consumo de energia elétrica numa instalação elétrica e guardar essa informação para que o utilizador do sistema possa consultar essa informação, para tal é necessário um elemento de armazenamento de dados, a base de dados neste caso.

Funcionalidades do sistema de informação:

A interface com o utilizador (página *web* / Aplicação Android) permite essencialmente a consulta dos dados referentes aos consumos de energia de uma dada instalação. De entre os dados que se podem consultar, refira-se:

- A consulta das leituras dos valores atuais de energia e potência, de uma instalação elétrica;
- A consulta dos valores referentes à qualidade da energia elétrica de uma instalação, como o *Total Harmonic Distortion* (THD) e o valor da frequência da rede elétrica;
- A consulta dos valores obtidos pelo *power meter* durante um intervalo de tempo definido pelo utilizador, relativo ao período de análise, permitindo obtenção de gráficos do consumo de energia numa base diária ou horária e a análise do valor da potência consumida ao longo do tempo;
- A consulta do preço da energia elétrica consumida numa instalação durante um intervalo de tempo, definido pelo utilizador.

BASE DE DADOS (MySQL)

Para registar, guardar e consultar toda a informação do sistema é indispensável a existência de uma base de dados. Uma base de dados compreende um conjunto de tabelas com vários campos predefinidos, denominados de atributos, os atributos representam os parâmetros da informação que será inserida em cada registo na tabela.

Para o desenvolvimento da solução descrita, foi selecionada a base de dados MySQL e utilizado o *software* MySQL Workbench. Esta solução permite projetar e estruturar uma base de dados de forma gráfica e intuitiva, proporcionando uma melhor perceção da sua estrutura e funcionamento.

Estrutura da base de dados

A estrutura da base de dados foi concebida de forma a registar toda a informação necessária ao correto funcionamento das aplicações desenvolvidas evitando redundâncias desnecessárias da informação inserida no sistema.

Assim sendo, existem cinco tabelas distintas, que podem ser consultadas na Figura 18, com informação específica às funcionalidades implementadas, de entre elas refira-se:

- *PM_VALUES*: tabela que contém os registos de todas as leituras efetuadas pelos *Power meters* registados no sistema;
- *users*: tabela que contém o registo dos utilizadores do sistema;
- *companies*: tabela que contém o registo das companhias fornecedoras de energia registadas no sistema;
- *comp_user*: tabela que regista a relação entre as companhias de energia e os utilizadores do sistema;
- *PM*: tabela que contém o registo de todos os *Power meters* presentes no sistema.

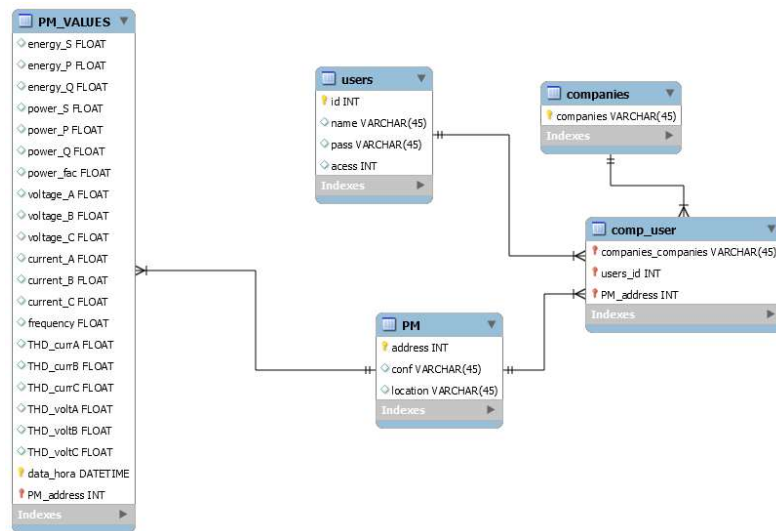


Figura 18 – Estrutura da base de dados e relação entre tabelas.

Tabela *PM_VALUES*:

Quadro 9 – Valores constituintes da tabela *PM_VALUES*.

Energy_S: energia aparente	Current_B: corrente na fase B
Energy_P: energia ativa	Current_C: corrente na fase C
Energy_Q: energia reativa	Frequency: frequência
Power_S: potência aparente	THD_currA: THD da corrente da fase A
Power: potência ativa	THD_currB: THD da corrente da fase B
Power_Q: potência reativa	THD_currC: THD da corrente da fase C
Power_fac: fator de potência	THD_voltA: THD da tensão da fase A
Voltage_A: tensão na fase A	THD_voltB: THD da tensão da fase B
Voltage_B: tensão na fase B	THD_voltC: THD da tensão da fase C
Voltage_C: tensão na fase C	Data_hora: data e hora do registo
Current_A: corrente na fase A	PM_adress: endereço do <i>Power meter</i>

Nesta tabela são registados os valores relativos ao consumo energético da instalação, para além desses valores ainda existem os atributos *Data_hora* e *PM_adress* que servem para identificar a data e hora do registo e o *power meter* ao qual a leitura faz referência, respetivamente. Os dois últimos atributos referidos constituem a chave primária da tabela *PM_VALUES*.

Tabela *users*:Quadro 10 – Valores presentes na tabela *users*.

Id: número identificador do utilizador
name: nome do utilizador
pass: palavra passe de acesso do utilizador
access: nível de acesso do utilizador

A tabela *users* tem quatro atributos. Os atributos *Id* e *name* fazem referência à identificação do utilizador, enquanto que os atributos *pass* e *access* estão envolvidos na segurança do sistema. O atributo *pass* permite que apenas o utilizador correspondente possa aceder ao sistema e o atributo *access* está relacionado com o nível de acesso ao sistema, pode ter dois valores, pode assumir o valor “1” no caso do utilizador registado ter permissão de administrador, ou pode-lhe ser atribuído o valor “2” no caso do utilizador registado ter acesso apenas para consulta. O atributo *Id* é a chave primária desta tabela, pois a partir deste atributo é possível identificar facilmente um utilizador.

Tabela *PM*:Quadro 11 – Valores constituintes da tabela *PM*.

address: endereço do <i>Power meter</i>
conf: configuração da instalação
location: localização da instalação

A tabela *PM* contém o registo dos *power meters* instalados no sistema. Através dos atributos *address*, que corresponde ao endereço do *power meter* na instalação, do atributo *conf*, que corresponde à forma como o *power meter* está ligado à instalação, se está ligado em estrela ou triângulo por exemplo e o atributo *location* que corresponde à localização física do *power meter*. Dos atributos descritos destaca-se o atributo *address* como chave primária desta tabela, pois é a partir dele que é possível identificar um registo de um *Power meter*.

Tabela *companies*:Quadro 12 – Valores constituintes da tabela *companies*.

companies: nome da companhia

Nesta tabela é feito o registo das companhias de eletricidade que fornecem energia aos utilizadores do sistema, para as identificar apenas é necessário o seu nome, daí esta tabela apenas ter um atributo que por sua vez é a chave primária desta tabela.

Tabela *comp_user*:Quadro 13 – Valores constituintes da tabela *comp_user*.

companies_companies: nome da companhia

users_id: número identificador do utilizador

PM_address: endereço do <i>Power meter</i>

Na tabela *comp_user* é registada a relação entre um utilizador, a companhia fornecedora de eletricidade e o *power meter*. Uma vez que o mesmo utilizador pode ter mais do que uma instalação e mais do que um fornecedor, daí a importância desta tabela. Nesta tabela os três atributos que a constituem fazem parte da chave primária pois só a combinação dos três permite identificar um registo.

No *software* MySQL Workbench é possível definir relações entre as tabelas da base de dados. Estas relações servem para definir a ligação entre atributos de diferentes tabelas, que têm como objetivo, no caso deste projeto, por exemplo, evitar que seja inserida informação na tabela de registo dos valores recolhidos com o endereço de um *power meter* que não esteja registado na tabela com o registo de todos os *power meters* registados no sistema.

Na criação da base de dados é importante definir dois termos importantes, que conforme a sua configuração no projeto da base de dados podem alterar completamente o funcionamento da base de dados. Estes termos são a chave primária e a chave estrangeira. Uma chave primária é definida por um ou mais atributos de uma tabela, que quando definidos como chave primária nunca se podem repetir.

Na nossa solução a chave primária da tabela com os valores recolhidos pelo *power meter* a chave primária é o atributo *data_hora*, que recebe a data e a hora a que a informação foi recolhida e o atributo *PM_adress*, que recebe o endereço do *power meter* que recolheu os dados guardados naquele registo. Esta chave primária ao estar definida desta forma coloca a condição de que em cada registo inserido na tabela a combinação entre a data e hora e o endereço do *power meter* não se podem repetir, esta é a definição de chave primária. A definição de chave estrangeira é mais simples, sendo que uma chave estrangeira é um atributo de uma tabela, que faz parte da chave primária de outra tabela. Neste caso, por exemplo na tabela com o registo dos utilizadores, a cada utilizador é atribuído um número de identificação, o *id* e este atributo é a chave primária da tabela *users*, mas este atributo também existe na tabela *comp_user*, que regista a relação entre os utilizadores, as companhias e os *power meters* registados no sistema, o que torna o atributo *comp_user* uma chave estrangeira.

Na construção da base de dados utilizada neste projeto foi utilizado o modelo relacional, que permite relacionar registos entre tabelas, o que oferece maior controlo ao inserir e apagar registos das tabelas constituintes da base de dados.

As tabelas utilizadas na base de dados estão normalizadas até à segunda forma normal. Cada atributo das tabelas contém valores atómicos, isto é, os valores estão o mais decompostos possível, por exemplo as energias estão decompostas até ao seu limite, temos o atributo referente à energia aparente, energia reativa e energia ativa, sendo ambas energias, na tabela estão divididos de forma a fazer referência aos três tipos de energias, isto é, os atributos contém valores atómicos, não havendo conjuntos de atributos repetidos, descrevendo a mesma característica. Como as duas condições anteriores foram respeitadas, diz-se que as tabelas estão na primeira forma normal. Uma tabela está na segunda forma normal quando está na primeira forma normal e os atributos que não pertencem à chave primária apenas dependem da totalidade da chave primária. Por exemplo na tabela que regista os valores lidos pelos *power meters*, cada registo que não seja a data e hora e o endereço do *power meter*, depende do conjunto data e hora e endereço do *power meter*, não depende só da data e hora ou só do endereço. Como nas tabelas criadas todos os atributos que não pertencem à chave primária satisfazem esta condição, então pode-se dizer que as tabelas estão na segunda forma normal.

O objetivo da normalização das tabelas foi evitar a redundância de dados, tornando a base de dados mais compacta e de mais fácil acesso e consulta.

APLICAÇÃO WEB

Com o intuito de disponibilizar os dados recolhidos ao utilizador do sistema foram projetadas duas aplicações de apresentação de dados, uma direcionada para o ambiente *web* e outra para ambiente Android.

Para construir a aplicação em ambiente *web* foram utilizadas as linguagens de programação direcionadas para apresentação de páginas *web* e de processamento de informação do lado do servidor. Essas linguagens são respetivamente o HTML e PHP. A linguagem HTML é processada no *browser*, no lado do cliente, enquanto que a linguagem PHP é processada no lado servidor quando é pedida a consulta da página, o PHP permite ter páginas dinâmicas que alterem a informação apresentada, conforme a escolha do utilizador, em relação à informação que pretende consultar.

Para definir o *layout* da aplicação e torná-la mais apelativa visualmente foi utilizada a biblioteca Bootstrap que permite projetar os elementos da aplicação de forma mais fácil e manter o seu aspeto de acordo com as tendências que se verificam em aplicações *web* hoje em dia.

No caso particular deste projeto, o servidor que aloja o *site* é o mesmo onde está a base de dados, utilizou-se um computador portátil, ligado à mesma rede *Ethernet* em que se encontra o Raspberry Pi. O programa XAMPP quando instalado num computador permite que este se comporte como um servidor pois instala o programa Apache, que permite lidar com pedidos de acesso ao servidor e processar código PHP, dois aspetos importantes para o desenvolvimento deste sistema. Permite ainda alojar no computador várias bases de dados, MySQL neste caso, para armazenamento e consulta de dados. De cada vez que um utilizador da rede queira aceder ao *site* basta inserir o URL http://localhost/dashboard/site_tese/login.html, com a particularidade de que a palavra *localhost* deve ser substituída pelo endereço IP do servidor na rede.

O *site* permite dois modos de acesso, um modo é destinado a utilizadores administradores do sistema e o segundo modo é destinado a utilizadores que apenas utilizam o sistema para consulta dos consumos energéticos da sua instalação. Para cumprir este objetivo o *site*

apresenta um *layout* com funcionalidades como consulta do consumo de energia, atualização de utilizadores, atualização de *power meters*, atualização de companhias e estabelecimento de relações entre companhias e utilizadores. O modo “utilizadores” possui somente as funcionalidades de consulta do consumo de energia.

A estrutura do *site* pode ser dividida nas páginas seguintes:

- Página de *Login*;
- Página de consulta do consumo de energia;
 - Consulta das energias e potências, com gráfico temporal num intervalo definido pelo utilizador;
 - Consulta dos últimos valores de tensão, corrente e THD da tensão e corrente recolhidos;
- Página de atualização de utilizadores;
- Página de atualização de *power meters*;
- Página de atualização de companhias;
- Página de estabelecimento de relações entre companhias e utilizadores;
- Página principal;
- Menu de escolha da função;
 - Página principal;
 - Ver consumos de energia;
 - Atualizar utilizadores;
 - Atualizar *power meters*;
 - Atualizar relações companhias utilizadores;
 - *Logout*;

Na página de *Login*, página *login.html*, é feita a verificação do utilizador, caso este esteja registado na base de dados, significa que é permitido o seu acesso à aplicação então é validado o *Login*.

Na Figura 19 é possível observar o formulário de recolha de informação.

Id do utilizador

Password

Submit

Figura 19 – Formulário de *Login* da aplicação *Web*.

Primeiro é recolhida a informação referente ao número identificador (ID) do utilizador e à sua palavra passe de acesso. Esta recolha é feita através de um formulário HTML, utilizando o método POST, como é possível observar na Figura 19. De seguida é chamada a página *form.php* que compara os dados recolhidos com os existentes na base de dados e caso o resultado da verificação seja positivo, então o utilizador é direcionado para a página *consumos.php*, no caso de ter permissão de administrador, ou para a página *consumos_consulta.php* no caso de só ter permissão para consulta de valores. Se o utilizador não estiver registado então é apresentada uma mensagem de erro e o utilizador pode voltar à página *login.html* para introduzir dados válidos para aceder à aplicação, como se pode observar na Figura 20.

Login falhou!
[Regressar ao login](#)

Figura 20 – Menu de falha do *Login*.

```
/** Validação dos dados de Login ***/
$utilizador= $_POST['user'];
$senha= $_POST['pass'];

session_start();
$_SESSION['user']=$_POST['user'];
$link = mysqli_connect($server, $user, $pass, $database) or
die(mysqli_error());
mysqli_select_db($link, $database) or die(mysqli_error());
$strSQL = "select * from users where id='".$utilizador.'" and
pass='".$senha.'"";
$rs = mysqli_query($link, $strSQL) or die(mysqli_error());
$linha=mysqli_fetch_array($rs);
```

```

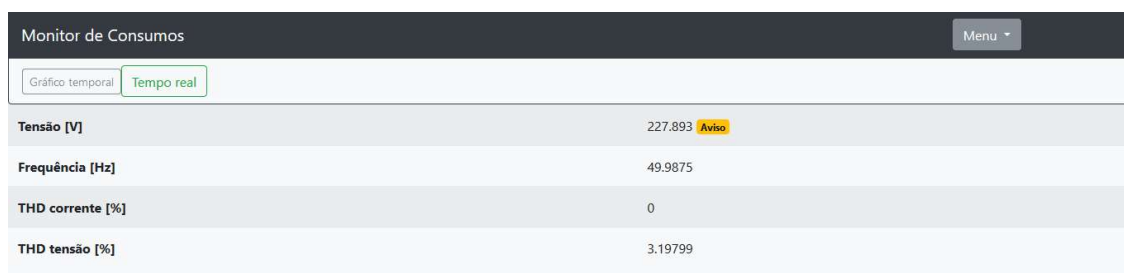
        if($_POST['user']!="" && $_POST['pass']!="") {
            if($linha[0]==$utilizador &&
$linha[2]==$senha) {
                if($linha[3]==1) {
                    include 'consumos.php';
                }

                if($linha[3]==2) {
                    include 'consumos_consulta.php';
                }
            }else{
                echo 'Login
falhou!'. "<br>". '<div>
                <a
href="login.html">Regressar ao login</a> </div>';
            }
        }else{
            echo 'Login falhou!'. "<br>". '<div>
            <a
href="login.html">Regressar ao login</a> </div>';
        }
    }

```

Script 11 – Validação dos dados de *Login*

Existem duas páginas de consulta de consumos uma para os administradores e outra para os utilizadores de consultas. A diferença entre estas duas páginas situa-se no menu, mais propriamente nas funcionalidades que o utilizador pode escolher. Este menu será explicado mais à frente. De seguida será explicado o funcionamento das páginas *consumos.php* e *consumos_consulta.php*, o qual é semelhante.



The screenshot shows a web interface titled 'Monitor de Consumos' with a 'Menu' dropdown. Below the title are two tabs: 'Gráfico temporal' and 'Tempo real'. The main content is a table with the following data:

Tensão [V]	227.893	Aviso
Frequência [Hz]	49.9875	
THD corrente [%]	0	
THD tensão [%]	3.19799	

Figura 21 – Menu de consulta dos valores em tempo real.

Na página *consumos.php* é apresentada uma tabela que contém os últimos valores lidos de tensão, corrente, THD da tensão e THD da corrente, na instalação, como mostra a Figura 21. O objetivo desta tabela é fornecer ao utilizador informação que permita analisar a qualidade

da energia elétrica na sua instalação, em tempo real. Por cima da tabela existe um menu onde é possível escolher uma de duas opções, a opção de tempo real que foi explicada neste parágrafo e a opção de gráfico temporal, onde o utilizador é encaminhado para a página *consumos_graph.php*, que é idêntica a esta página, com a diferença que no lugar da tabela tem um menu que permite ao utilizador escolher de entre os *power meters* registados, o que se pretende consultar, a data de início e de fim dos valores que pretende observar no gráfico que é apresentado abaixo deste menu. Como apresentado na Figura 22. O gráfico apresenta os valores das energias ativa, reativa e aparente no intervalo indicado e existe um segundo gráfico que apresenta os valores das potências ativa, reativa e aparente ao longo do tempo.

Ao submeter a data de início e fim da consulta é recarregada a página *consumos_graph.php* que recebe os valores inseridos pelo método POST. O servidor processa o pedido relativo aos valores compreendidos dentro da janela temporal definida, através do código PHP inserido nesta página e retorna a página com o gráfico com os valores pretendidos, o que a torna dinâmica e permite alterar os dados a analisar e visualizar conforme a escolha efetuada pelo utilizador.

Para apresentar os gráficos foi utilizada a biblioteca *Chart.js*, que permite criar gráficos utilizando código em linguagem JavaScript, que é inserida na página *consumos_graph.php*. Escolheu-se esta biblioteca de representação de gráficos porque dispõe de várias funcionalidades que permitem modificar o gráfico para o tornar mais apelativo ao utilizador.

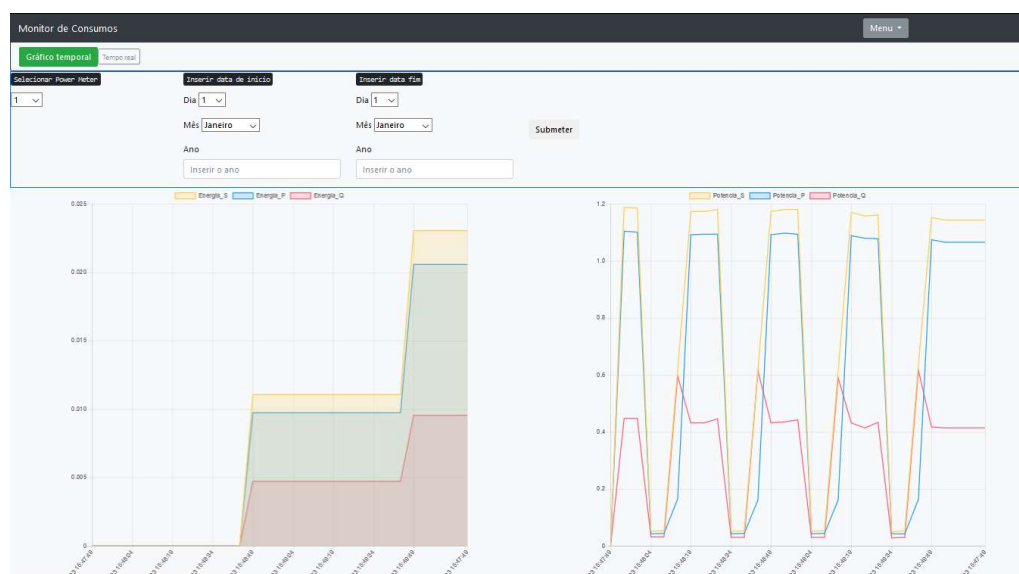


Figura 22 – Menu de consulta dos valores num determinado espetro temporal.

A funcionalidade de adicionar e apagar utilizadores do sistema é feita através da página *add_users.php*. Esta página apresenta o formulário onde se inserem os dados do novo utilizador a registar, como mostra a Figura 23, o qual, após ser preenchido e submetido chama a página *save_user.php*, que contém código PHP que recebe as variáveis inseridas no formulário, e efetua o processamento destes dados de forma a registar o novo utilizador na base de dados, do lado do servidor.

Figura 23 – Menu de atualização dos utilizadores.

Na página *save_user.php* utilizando código PHP é feita a conexão à base de dados através da função *mysqli_connect* e é selecionada a base de dados através da função *mysqli_select_db*. Posteriormente é procurado o maior número ID existente na tabela com o registo dos utilizadores e ao novo utilizador é atribuído o ID imediatamente acima desse, através das funções *mysqli_query* e *mysqli_fetch_array*. De seguida é enviada para a base de dados uma *query* SQL a pedir para inserir um novo utilizador com os seus dados usando a função *mysqli_query*. Caso o registo seja bem-sucedido é apresentada uma mensagem a comunicar que o registo foi bem-sucedido e um botão para retornar à página *add_users.php*, caso o registo falhe o comportamento é idêntico com a diferença de que a mensagem apresentada reporta uma falha no registo.

```

/** Registo de um novo utilizador */
$strSQL = "select MAX(id) from users;";
$rs = mysqli_query($link, $strSQL) or
die(mysqli_error());
$linha=mysqli_fetch_array($rs);
$id=$linha[0]+1;

$strSQL = "insert into users(id, name, pass, acess)
values('".$id."','".$nome."','".$passe."','".$nivel."');";

```

```

$rs = mysqli_query($link, $strSQL) or die(mysqli_error());
        if(isset($rs)){
                                echo 'Utilizador adicionado com
sucesso!';
        }else{ echo "Erro ao adicionar o utilizador.
</br>Tente outra vez."; }

```

Script 12 – Registo de um novo utilizador.

A funcionalidade de consulta dos utilizadores existentes, na página *add_users.php*, é feita através de um formulário onde o administrador insere o ID do utilizador que está à procura, de seguida através do método POST esse número é passado para a página *showusers.php*, que vai comunicar com a base de dados e verificar se o número existe. O processo de comunicação com a base de dados é similar ao processo explicado no parágrafo anterior onde se explica o processo de registo de um utilizador. A diferença agora verifica-se na *query SQL* que em vez de ser uma query de inserção de registos, é uma *query* de consulta, que utiliza o comando *sql select*. No caso de não ser inserido nenhum número na consulta, são consultados todos os registos na tabela dos utilizadores, como é possível observar na Figura 23.

```

/***/ Apresentação dos utilizadores */*
if(isset($_POST['search'])){
                                $nome=$_POST['search'];
                                $strSQL = "select * from users
where id like '". $nome. "%'";
        }else{$strSQL = "select * from users;";}

```

Script 13 – Apresentação dos utilizadores.

A pesquisa foi realizada dessa forma tendo em conta que ao iniciar a página *add_users.php* no lado direito da página é apresentada uma tabela com os utilizadores e essa tabela é criada através da página PHP *showusers.php*. O objetivo desta abordagem pretende que ao iniciar a página sejam apresentados os utilizadores existentes. No caso de não existir um número inserido para pesquisar, então são apresentados todos os registos existentes. A tabela apresenta seis colunas, a coluna com o ID do utilizador, a coluna com o nome do utilizador, outra com a palavra passe de acesso, a quarta coluna mostra o nível de acesso do utilizador e a quinta e sexta colunas apresentam um botão para apagar e modificar o registo do utilizador, respetivamente. Para construir esta tabela inseriu-se código HTML dentro do

código PHP e através da função *echo* o código HTML é passado para a página que o servidor devolve ao *browser*, para ser processada no navegador do utilizador.

```

/**** Alterar utilizadores ***/
echo '<tr>'.
'<td>'.$linha[0].'</td>'.
'<td>'.$linha[1].'</td>'.
'<td>'.$linha[2].'</td>'.
'<td>'.$linha[3].'</td>'.
'<td>
        <form method="post" action="delete_user.php">
        <div class="form-group">
        <input type="hidden" class="form-control"
value="'.$linha[0].'" name="deleteuser" />
        </div>
        <button type="submit" class="btn btn-
primary">Apagar</button>
</form>
</td>'.
'<td>
        <form method="post" action="change_add_users.php">
        <div class="form-group">
        <input type="hidden" value="'.$linha[0].'"
name="user_id" />
        </div>
        <button type="submit" class="btn btn-
primary">Alterar</button>
        </form>
</td>'.
'</tr>';

```

Script 14 – Alterar utilizadores.

Para eliminar um utilizador dos registos o processo de comunicação e envio da *string SQL* é idêntico ao explicado anteriormente. A única alteração acontece na *string SQL* que será uma *string* com o comando *delete*.

```

/**** Apagar utilizador ***/
$strSQL = "delete from users where id ='".$id."'";
$rs = mysqli_query($link, $strSQL) or die(mysqli_error());
if(isset($rs)){
echo 'Utilizador apagado com sucesso!';
}else{ echo "Erro ao apagar o utilizador. <br>Tente outra vez."; }

```

Script 15 – Apagar um utilizador.

No caso do administrador pretender alterar um utilizador, ao seleccionar a opção *Alterar* os campos com os atributos de cada utilizador referentes ao nome, palavra passe e nível de acesso ficam iguais aos campos do menu de adicionar um novo utilizador e o administrador

pode alterá-los, no fim basta seleccionar a opção guardar e as alterações realizadas ao utilizador são registadas. A página *change_add_users.php*, é invocada para criar a tabela com a lista dos utilizadores existentes, com os campos referentes aos atributos de cada utilizador alterados de forma a permitir a sua alteração. Apenas é possível fazer a alteração de um utilizador de cada vez. O exemplo desta funcionalidade pode ser observado na Figura 24.

Figura 24 – Menu de atualização dos utilizadores, com a opção de alterar um utilizador seleccionada.

O comando *SQL* enviado para a base de dados, para efetuar uma alteração num registo é o comando *update* que é utilizado na página *change_save_user.php* que está incluída no processo de alteração de um utilizador, pois é a página chamada para realizar a referida alteração.

```

//*** Alterar utilizador ***
$strSQL = "UPDATE users SET name='".$nome."', pass='".$passe."',
acess='".$nivel.'" WHERE id='".$id.'";";

$rs = mysqli_query($link, $strSQL) or die(mysqli_error());
        if(isset($rs)){
                                echo 'Utilizador alterado com
sucesso!';
        }else{echo "Erro ao alterar o utilizador. </br>Tente
outra vez."; }

```

Script 16 – Alterar um utilizador.

Para atualizar os *power meters* instalados no sistema foi criada a página *add_PM.php*, de atualização dos *power meters*. Esta página apresenta a funcionalidade de adicionar um novo *power meter* ou então consultar os *power meters* existentes, com a possibilidade de os apagar ou alterar, como é possível observar na Figura 25. Na secção de consulta foi adicionada a

funcionalidade de pesquisa pelo endereço do *power meter*, para o caso de o utilizador não encontrar o *power meter* que pretende consultar diretamente na tabela.

O método utilizado para registar um *power meter* novo é idêntico ao método descrito para registar um novo utilizador. A diferença verifica-se nos atributos a registar para cada *power meter*, que neste caso são o endereço do *power meter*, a configuração da instalação na rede e a localização da instalação. A *string SQL* que será enviada para a base de dados para registar o novo *power meter* será também diferente.

```

/**** Registrar um power meter ***/
$strSQL = "insert into pm(address, conf, location)
values ('".$nome."', '".$conf."', '".$loc."');";

$rs = mysqli_query($link, $strSQL) or die(mysqli_error());
    if(isset($rs)){
        echo 'PM adicionado com sucesso!';
    }else{ echo "Erro ao adicionar o PM. </br>Tente outra
vez."; }

```

Script 17 – Registrar um *power meter*.

Endereço	Configuração	Localização	Apagar	Alterar
1	aaaa	aaaa	Apagar	Alterar
2	aaaa	aaaa	Apagar	Alterar
3	fff	ne	Apagar	Alterar
10	estrela	n.e	Apagar	Alterar
222	LLL	X	Apagar	Alterar

Figura 25 – Menu de atualização dos *power meters*.

No menu de atualização dos *power meters* também é disponibilizada a funcionalidade de apagar ou alterar o registo de um *power meter*. A forma como estas duas ações se processam é idêntica à forma de processamento de um pedido de alteração ou para apagar um utilizador do sistema, no menu de atualização dos utilizadores. Os comandos *SQL* são idênticos aos

apresentados anteriormente. Para apagar é utilizado o comando *Delete* e para alterar, o comando *update*, sendo que o que se altera na *string* são os atributos da tabela, assim como a tabela em questão. Desta vez é possível alterar os atributos de configuração da instalação do *power meter* e a sua localização. A tabela onde serão realizadas as alterações é a tabela *pm* que contém o registo de todos os *power meters* do sistema. Na Figura 26 é possível observar as alterações da página de atualização dos *power meters* após ser selecionada a opção de alteração de um *power meter* registado no sistema.

```

//*** Alterar um power meter ***
$strSQL = "UPDATE pm SET conf='".$conf."', location='".$location."' WHERE
address='".$id."'";

```

Script 18 – Alterar um *power meter*.

```

//*** Apagar um power meter ***
$strSQL = "delete from pm where address = '".$id."'";

```

Script 19 – Apagar um *power meter*.

Endereço	Configuração	Localização	
1	aaaa	aaaa	Guardar
2	aaaa	aaaa	Guardar
3	fff	ne	Guardar
10	estrela	n.e	Guardar
222	LLL	X	Guardar

Figura 26 – Menu de atualização de *power meters*, com a opção “Alterar” selecionada.

A página para efetuar a atualização das companhias registadas no sistema ou registar uma nova está dividida em duas partes. Como é apresentado na Figura 27 o menu da esquerda permite registar uma nova companhia. O único atributo necessário para registar uma nova companhia é o nome da mesma, devido ao facto de as companhias terem nomes distintos. A

recolha do nome é realizada utilizando um formulário HTML, similar ao formulário utilizado nos menus de atualização dos utilizadores e de *power meters*.

```
/** Recolher nome de uma companhia ***/
<div class="container-fluid">
<div class="row justify-content-md-center">

  <div class="col-md-4">
  <p><kbd>Inserir novas companhias</kbd></p>
  <form method="post" action="save_company.php">
  <div class="form-group">
  <label for="adduser">Companhia</label>
  <input type="text" class="form-control" name="addcompany"
placeholder="Nome da companhia">
  </div>
  <button type="submit" class="btn btn-default">Submeter</button>
  </form>
  </div>
```

Script 20 – Recolher o nome de uma companhia.

O processo de registo na base de dados é idêntico aos processos de registo de utilizadores e *power meters*, com a diferença nos parâmetros que são passados na *string SQL*, assim como a tabela onde será efetuado o registo também muda, neste caso é utilizada a tabela *companies*. Quando o administrador seleciona a opção de submeter o novo registo é chamada a página *save_company.php*, que efetua a ligação com a base de dados, recolhe os valores passados no formulário HTML, cria a *string SQL Insert* com os valores inseridos no formulário e efetua o registo, após o comando de registo ser enviado, é realizado o controlo do registo da nova companhia, para verificar se este foi bem-sucedido sendo apresentada uma mensagem a indicar o sucesso ou o erro na inserção do novo registo.

```
/** Registrar uma companhia ***/
$strSQL = "insert into companies(companies) values('".$nome."');";
```

Script 21 – Registrar uma companhia.

Na Figura 27 no lado direito do menu de atualização das companhias é apresentada uma lista com as companhias registadas no sistema. Para além disso é incluindo um formulário de pesquisa por nome da companhia, que processa a pesquisa da mesma forma que esta é realizada no menu de atualização de utilizadores. Para cada companhia registada que é apresentada na lista existe um botão que quando selecionado apaga a respetiva companhia. Este processo mais uma vez é igual ao processo de apagar registos que foram referidos

anteriormente na secção referente à atualização de utilizadores e de *power meters*. Ao seleccionar o botão *Apagar* é chamada a página *delete_company.php*, que faz a ligação à base de dados, depois é criada a *string SQL* com o comando *Delete* e onde selecciona a tabela *companies* e dá ordem para apagar o registo seleccionado.

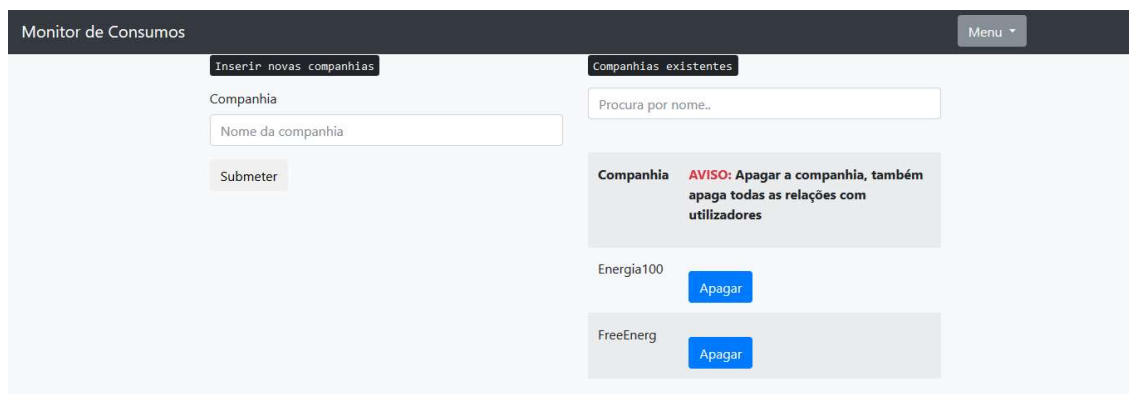


Figura 27 – Menu de atualização de companhias.

O último menu desenvolvido para a aplicação *web* tem por objetivo proporcionar ao administrador a possibilidade de estabelecer relações entre os utilizadores registados, as companhias e os *power meters* instalados no sistema. Desta forma é possível ver qual a companhia fornecedora de energia eléctrica que está associado a cada utilizador e qual o *power meter* utilizado na instalação do utilizador em causa.

O comportamento deste menu é idêntico ao comportamento dos menus de atualização explicados nas secções anteriores. Este menu é implementado a partir da página *add_comp_rel.php*, que à esquerda tem um formulário para escolher os valores que irão definir a relação entre utilizadores, companhias e *power meters*. A relação será registada após o utilizador seleccionar a opção *Submeter*. O registo será então efetuado pela página PHP *save_comprel.php* utilizando o mesmo processo utilizado para registar um utilizador na base de dados, através de uma *string SQL*. As relações serão registadas na tabela *comp_user* da base de dados.

Para criar os botões *dropdown* do formulário foi utilizado código PHP que é executado quando chamado dentro da página *add_comp_rel.php* que recolhe as companhias, utilizadores e *power meters* registados na base de dados e cria os botões para que na lista *dropdown* apareça todos os valores registados.

```

/** Chamada dos botões dinâmicos do formulário **
<form method="post" action="save_comprel.php">
    <?php include
'botao_dinamico_companies.php'; ?>
    <?php include
'botao_dinamico_users.php'; ?>
    <?php include
'botao_dinamico_PMs_comp.php'; ?>
    <button type="submit" class="btn btn-
default">Submeter</button>
</form>

```

Script 22 – Chamada dos botões dinâmicos do formulário.

```

/** Criação de um dos botões dinâmicos do formulário **
<?php
error_reporting(E_ERROR);

    $server="localhost";
    $user="root";
    $pass="";
    $database="pm_db";
$link = mysqli_connect($server, $user, $pass, $database) or
die(mysqli_error());
mysqli_select_db($link, $database) or die(mysqli_error());

$strSQL = "select * from companies;";

$rs = mysqli_query($link, $strSQL) or die(mysqli_error());

    echo '<p><kbd>Selecionar companhia</kbd></p>'.
    '<div class="form-group">'.
    '<label for="company"></label>'.
    '<select name="company">';

    while($linha=mysqli_fetch_array($rs)){
    echo '<option value="'.

    $linha[0].

    '>'.

    $linha[0].

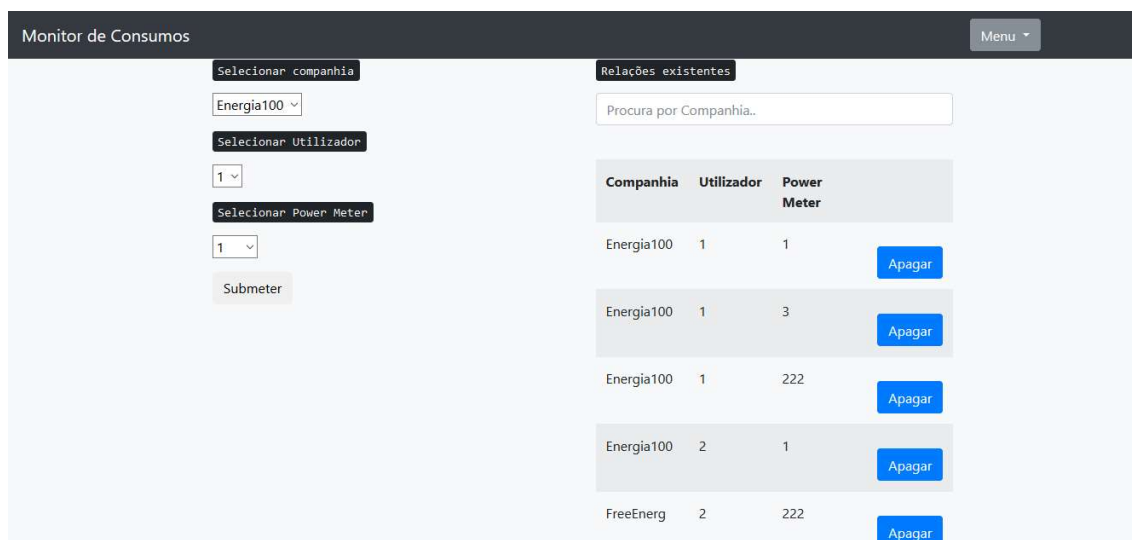
    '</option>';
    }
    echo '</select>'.
    '</div>';
mysqli_close($link);

?>

```

Script 23 – Criação de um botão dinâmico do formulário.

À direita na página de atualização da relação entre utilizadores e companhias é apresentada uma lista com as relações existentes, onde para cada relação registada é apresentada a companhia, o utilizador e o *power meter*, sendo esta lista criada utilizando o código PHP presente na página *showcomprel.php*. Adicionalmente é apresentado para cada relação um botão que permite ao administrador apagar a relação a que está associado. Ao seleccionar o botão *apagar* é chamada a página *delete_comprel.php* que recebe através do método POST a companhia, o utilizador e o *power meter* da relação que se pretende apagar e de seguida apaga da base de dados a relação em causa, utilizando o mesmo processo utilizado para apagar um utilizador da base de dados. De seguida na Figura 28 é apresentada a página de atualização da relação entre companhias e utilizadores, para representar visualmente como esta página funciona.



Companhia	Utilizador	Power Meter	
Energia100	1	1	Apagar
Energia100	1	3	Apagar
Energia100	1	222	Apagar
Energia100	2	1	Apagar
FreeEnergy	2	222	Apagar

Figura 28 – Página para atualizar as relações entre companhias e utilizadores.

Em todas as páginas da aplicação *web* existe uma barra no topo da página, com o botão *Menu*, este botão ao ser seleccionado apresenta um menu do tipo *dropdown* com todas as páginas às quais é possível aceder, de forma a permitir navegar pelas várias funcionalidades da aplicação *web*. O menu assume duas formas distintas, apresentado com todas as funções, no caso do utilizador da aplicação ter permissão de nível administrador, como é possível ver na Figura 29.

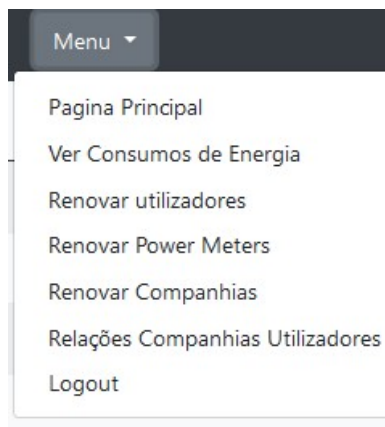


Figura 29 – Menu de navegação da aplicação *web*.

Ou contendo menos opções no caso do utilizador ter permissão para consultar os consumos apenas. Como é apresentado na Figura 30.

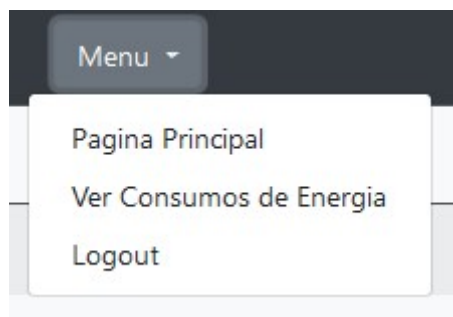


Figura 30 – Menu de navegação na aplicação, para utilizadores com permissão de consulta.

```

/** Botão Menu administrador **
<button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
    Menu
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
<a class="dropdown-item" href="capa.html">Pagina Principal</a>
<a class="dropdown-item" href="consumos.php">Ver Consumos de Energia</a>
<a class="dropdown-item" href="add_users.php">Atualizar utilizadores</a>
<a class="dropdown-item" href="add_PM.php">Atualizar Power meters</a>
<a class="dropdown-item" href="add_companies.php">Atualizar
Companhias</a>
<a class="dropdown-item" href="add_comp_rel.php">Relações Companhias
Utilizadores</a>
<a class="dropdown-item" href="login.html" action="logout.php">Logout</a>
</div>

```

Script 24 – Botão do menu de administrador.

```

/** ** Botão Menu consulta **
<button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
    Menu
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
    <a class="dropdown-item"
href="capa_consulta.html">Pagina Principal</a>
    <a class="dropdown-item"
href="consumos_consulta.php">Ver Consumos de Energia</a>
    <a class="dropdown-item" href="login.html"
action="logout.php">Logout</a>
</div>

```

Script 25 – Botão do menu de consulta.

Cada elemento do botão representa um *link* para a página respetiva. Exceto a opção de *Logout*. Esta opção efetua o *logout* do utilizador atual do sistema. Ao efetuar o *login* na aplicação é iniciada uma sessão, através da função PHP *session_start* e a partir daí é possível guardar numa variável da sessão o nome do utilizador atual do sistema, de forma a registar qual o utilizador com sessão iniciada.

```

/** ** Início de sessão PHP **
session_start();
$_SESSION['user']=$_POST['user'];

```

Script 26 – Início de sessão PHP.

Ao selecionar a opção *Logout* é chamada a página *logout.php* que encerra a sessão, através da função PHP *session_destroy*.

```

/** ** Fim de sessão PHP **
<?php
session_start();
session_destroy();
?>

```

Script 27 – Fim de sessão PHP.

APLICAÇÃO ANDROID

A aplicação em ambiente *android*, foi criada com o objetivo de proporcionar ao utilizador mais uma ferramenta para efetuar a consulta dos seus consumos e sensibilizá-lo para efetuar um consumo mais consciente e económico em termos de energia elétrica. A escolha deste ambiente decorre da massificação da utilização das tecnologias de comunicação móveis, nomeadamente *smartphones* e *tablets*, tornando a utilização da aplicação mais apelativa ao consumidor e aumentando as hipóteses de sucesso na sensibilização do cliente relativamente ao seu consumo energético. Para construir esta aplicação foi utilizado o ambiente de desenvolvimento de aplicações *Android Studio*, que utiliza as linguagens JAVA e XML, para programar as funcionalidades e para construir a parte gráfica da aplicação.

Para cumprir os objetivos enunciados a aplicação apresenta funcionalidades como a consulta da energia consumida e dos parâmetros de qualidade da energia em tempo real; a consulta da energia consumida num intervalo temporal definido, e a consulta do preço da energia consumida num período definido pelo utilizador.

A estrutura da aplicação pode ser dividida nas páginas seguintes:

- Página de *Login*;
- Página com o menu principal;
 - Página de consulta dos valores de tensão, corrente e frequência em tempo real;
 - Página de consulta dos valores THD registados em tempo real;
 - Página de consulta dos valores de energia e potência registados durante um período de tempo;
 - Página de consulta do preço da energia num determinado período de tempo;

A página de *Login* é a primeira página visível ao iniciar a aplicação. Nesta página o utilizador insere as suas credenciais de acesso: o número de identificação e a palavra passe para validar o seu acesso às funcionalidades da aplicação. Este processo é dividido em três fases, a primeira fase consiste na inserção do número de identificação e a palavra passe nos campos *User ID* e *Password*, visíveis nesta página, como é apresentado na Figura 31.

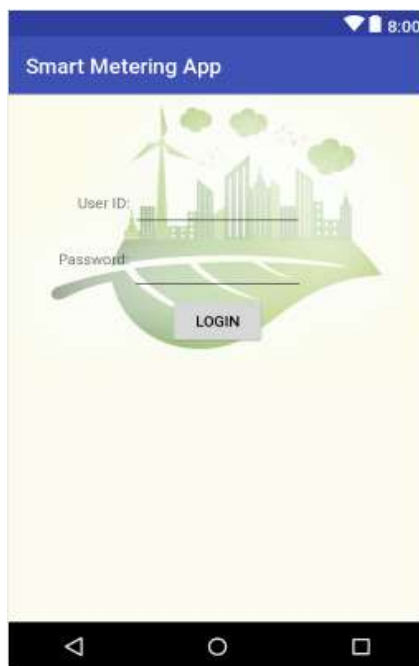


Figura 31 – Layout da página de login.

Os parâmetros inseridos para efetuar a operação de login são armazenados em duas variáveis criadas para este fim, designadas por *message_ID* e *message_Pass*.

```
//****Recolha dos parâmetros de Login *****/  
  
EditText editText_ID = (EditText) findViewById(R.id.editText);  
EditText editText_Pass = (EditText) findViewById(R.id.editText2);  
final String message_ID = editText_ID.getText().toString();  
final String message_Pass = editText_Pass.getText().toString();
```

Script 28 – Recolha dos parâmetros de Login.

De seguida a aplicação efetua um pedido ao servidor onde estão armazenados os dados de todos os utilizadores, solicitando que este confirme se as credenciais inseridas são válidas. As credenciais inseridas são enviadas para o servidor através do método GET da linguagem PHP, ou seja as credenciais são enviadas no URL. O servidor verifica se as credenciais são válidas e retorna uma mensagem com as credenciais registadas no sistema que coincidem com as credenciais inseridas pelo utilizador, caso contrário o servidor retorna campos vazios. Depois na aplicação é possível verificar se as credenciais inseridas são iguais às retornadas pelo servidor, desta forma a aplicação consegue encaminhar o utilizador para o menu principal caso as credenciais sejam válidas, ou apresentar uma mensagem de erro a informar

que as credenciais não são válidas para efetuar o login na aplicação. O pedido de confirmação ao servidor é feito da seguinte forma:

```
//****Pedido ao servidor e resposta *****/  
  
RequestQueue mRequestQueue;  
Cache cache = new DiskBasedCache(getCacheDir(), 1024 * 1024);  
Network network = new BasicNetwork(new HurlStack());  
mRequestQueue = new RequestQueue(cache, network);  
mRequestQueue.start();  
  
String url  
="http://smartgridtiago.hopto.org/appAndroidPHP_login.php?id=" +  
message_ID;  
  
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,  
new Response.Listener<String>() {...}, new Response.ErrorListener() {...});
```

Script 29 – Pedido ao servidor e resposta.

Na secção de código anterior primeiro é criada uma variável que será responsável por iniciar a conexão ao servidor. Depois é iniciada a memória *cache* do canal e definido o tamanho máximo das mensagens no canal. De seguida, ao criar a variável *network* é definido que a rede irá usar uma conexão HTTP. Na linha de código seguinte é criada a conexão, utilizando as definições definidas ao criar as variáveis *cache* e *network*. O último passo compreende o estabelecimento da conexão. Este processo é igual para as restantes funcionalidades da aplicação sempre que for necessário apresentar informação armazenada na base de dados, no servidor. Como é o caso da funcionalidade de apresentar os parâmetros de qualidade da energia consumida, os valores de tensão e corrente em tempo real, o cálculo do preço da energia e a apresentação da energia ou potência consumida num determinado intervalo de tempo.

O pedido dos dados para efetuar a validação do login do utilizador é efetuado utilizando a conexão descrita anteriormente. É enviado para o servidor um pedido para consultar a página http://smartgridtiago.hopto.org/appAndroidPHP_login.php. Após este pedido o servidor devolve para a aplicação o conteúdo da página, que são as credenciais registadas no sistema que coincidem com as inseridas pelo utilizador.

Na secção de código anterior existe uma função onde será feito o tratamento dos dados recebidos, nomeadamente “*Response.Listener<String>() {...}*” dentro desta função os

dados recebidos, que vêm no formato *JSON* são processados de forma a criar um vetor onde ficam guardados em cada campo o *ID* e a *password* do utilizador. De seguida é implementado um ciclo que percorre todo o vetor e compara cada campo com as variáveis *message_ID* e *message_Pass*. Caso ocorra uma correspondência entre as variáveis e um campo do vetor então a operação de *login* é bem-sucedida e o utilizador é direcionado para o menu principal da aplicação, caso contrário se não existir uma correspondência então é apresentada uma mensagem a informar que a operação de *login* não foi bem-sucedida.

A página referente ao menu principal é composta por vários ícones que quando selecionados dão acesso a uma funcionalidade distinta da aplicação, a disposição gráfica deste menu pode ser visualizada na Figura 32.



Figura 32 – Layout do menu principal da aplicação android.

Cada menu da aplicação, exceto a página de *Login*, tem um botão no canto superior esquerdo que permite voltar para o menu anterior, como é possível observar na Figura 32.

Os ícones foram criados ao adicionar um *ImageButton* e depois associar uma imagem a cada *ImageButton*, nas definições de cada ícone foi adicionada a funcionalidade *onClick* que

permite associar ao ícone uma função do ficheiro *java* referente ao menu principal da aplicação. Cada uma destas funções associadas a cada ícone é composta por duas linhas de código que cria uma variável *intent* que fica associada durante a sua criação ao menu a que o ícone diz respeito e depois é iniciado esse menu. Por exemplo ao seleccionar o ícone *Energy Price*, este vai chamar a função *price* do ficheiro *Main_menu.java*, que cria uma variável *intent* que está associada ao ficheiro *Price_Activity.java*, depois através da função *startActivity()* é aberto o menu referente ao preço da energia.

```
/**Iniciar o menu Energy Price ***/  
public void price(View view) {  
    Intent intent = new Intent(Main_Menu.this, Price_Activity.class);  
    startActivity(intent);  
}
```

Script 30 – Iniciar o menu "Energy Price".

Ao seleccionar no menu principal a opção *Actual Read* o utilizador é encaminhado para o menu onde são apresentados os últimos valores recolhidos pelo *power meter*. O objetivo deste menu é apresentar os valores referentes à energia consumida, potência instantânea, tensão e corrente, em tempo real, para que o utilizador tenha uma noção das variações ocorridas na energia elétrica que chega à sua instalação.

Para efetuar a consulta dos valores em tempo real o utilizador apenas tem de escolher de entre os *power meters* disponíveis, o que pretende consultar. Esta opção é implementada através de um botão *dropdown* que se situa no canto inferior direito da página, como é apresentado na Figura 33.



Figura 33 – Menu de consulta dos valores referentes à energia consumida em tempo real.

De seguida são apresentados os valores atuais presentes nos registos do *power meter* relativos aos valores da energia e potência: ativa, reativa e aparente, aos valores da tensão e da corrente nas fases da instalação, ou nas três fases caso seja uma instalação trifásica e é apresentada a frequência na rede. Desta forma, o utilizador pode verificar se os parâmetros referentes à energia que chega à sua instalação estão de acordo com o que foi estipulado com a empresa fornecedora.

No topo da página é apresentada a data da última leitura para o utilizador saber a que instante de tempo correspondem os valores apresentados.

Relativamente ao funcionamento interno deste menu, este pode ser dividido em duas partes. A primeira parte corresponde ao código que será executado quando a página é criada, este código *JAVA* é responsável por efetuar um pedido ao servidor, da mesma forma que foi explicado anteriormente na operação de *login*, com a diferença na página que a aplicação pede para consultar ao servidor. Agora a página que se pretende consultar é a página <http://192.168.1.66/appAndroidPHP.php>, que devolve uma lista com todas as entradas na tabela *pm_values* da base de dados.

Durante a resposta do servidor os dados são armazenados na variável *founderArray*.

```
//****Recepção dos dados retornados pelo servidor *****/
Valores[] founderArray = gson.fromJson(response, Valores[].class);
```

Script 31 – Recepção dos dados retornados pelo servidor.

Depois foi implementado um ciclo que percorre o vetor *founderArray*, com todos os dados retornados pelo servidor e cada vez que é detetada uma entrada com o endereço de um *power meter*, que ainda não tinha sido encontrado nos dados retornados pelo servidor, este é adicionado a um vetor *PMlist* criado para armazenar os endereços de todos os *power meters* que contêm dados que podem ser consultados.

```
//****Deteção de todos os power meters e criação do botão dropdown****/
final List<String> PMlist = new ArrayList<>();
...
for(int i=0; i<founderArray.length;i++){
    if(!PMlist.contains(founderArray[i].PM_address))
        PMlist.add(founderArray[i].PM_address);
}

spinnerArrayAdapter.setDropDownViewResource(R.layout.spinner_item);
spinner_PM.setAdapter(spinnerArrayAdapter);
```

Script 32 – Deteção de todos os "power meters" e criação do botão "dropdown".

Após o utilizador seleccionar qual o *power meter* que se pretende consultar, ao carregar no botão *refresh* a aplicação comunica com o servidor a solicitar para consultar a página <http://192.168.1.66/appAndroidPHP.php>, que devolve todos os registos presentes na tabela *pm_values*, este pedido é realizado da mesma forma que o pedido efetuado durante a operação de *login*, a única diferença é a página que se pretende consultar. Durante a resposta do servidor os dados são armazenados na variável *founderArray*.

```
//**** Recepção dos dados retornados pelo servidor *****/
Valores[] founderArray = gson.fromJson(response, Valores[].class);
```

Script 33 – Recepção dos dados retornados pelo servidor.

Depois é iniciado um ciclo que percorre o vetor *founderArray* e verifica em cada campo se o atributo *PM_address* é igual ao endereço do *power meter* escolhido pelo utilizador. Através de uma condição *if*, se for igual então os valores das energias, potência, tensões, correntes desse campo são apresentados no menu de leitura dos valores em tempo real, na aplicação. O ciclo termina quando percorrer todo o vetor *founderArray*, portanto os valores

apresentados são sempre os mais recentes, uma vez que quando foram devolvidos pelo servidor e armazenados no vetor manteve-se a sua ordem cronológica.

```

//****Apresentação dos últimos dados registados pelo power meter ****/
for (int i = 0; i < founderArray.length; i++) {
    if (founderArray[i].PM_address.equals(message)) {

        textView11.setText("P (KW): " +
Float.toString(founderArray[i].power_P));

        textView7.setText("Date of the current read: " +
founderArray[i].data_hora.replaceAll("\\s+", " "));
    }
}

```

Script 34 – Apresentação dos últimos dados registados pelo "power meter".

O menu de consulta dos valores de qualidade da energia, *Energy Quality*, é idêntico ao menu *Actual Read*, no aspeto gráfico e no seu funcionamento, a única diferença são os valores que são utilizados de entre todos os valores retornados pelo servidor. Neste menu são utilizados os valores de THD das tensões e correntes das fases do *power meter*, bem como do valor da frequência e do factor de potência da instalação. Na Figura 34 é apresentado o aspeto gráfico deste menu.



Figura 34 – Layout do menu "Energy Quality".

No menu principal ao escolher o ícone *Energy Price* o utilizador é encaminhado para o menu onde é apresentado o preço da energia consumida na instalação elétrica que este pretende consultar. Para o efeito é apresentado um quadro onde é discriminado todas as medições intervenientes no cálculo do preço final, nomeadamente o valor do preço da energia expresso

em €/kWh o qual irá ser multiplicado pelo valor da energia ativa consumida no período selecionado pelo utilizador. Para além disso existe um custo adicional relativo à potência contratada, o qual vai ser multiplicado pelo número de dias existentes dentro do intervalo temporal escolhido pelo utilizador. O valor final do custo da energia consumida agrega essas duas componentes. Ao iniciar uma consulta do preço que irá pagar, o utilizador tem de seleccionar qual o *power meter* que pretende consultar e a data de início e fim do período para o qual o utilizador pretende consultar o preço da energia elétrica. Estas três opções de escolha podem ser observadas na Figura 35, no topo do menu.



The screenshot shows the 'Smart Metering App' interface. At the top, there is a blue header with a back arrow and the text 'Smart Metering App'. Below the header, there is a section titled 'Select Power Meter' with a dropdown menu showing '1' and an 'ENTER' button. Below this, there are two date fields: 'Start date: 14-8-2017' and 'Ending date: 3-8-2018'. At the bottom, there is a table with three columns: 'Rate', 'Consumed Energy', and 'Price of energy'. The table contains two rows of data and a total row.

Rate	Consumed Energy	Price of energy
0.16 [€/Kwh]	0.56 [Kwh]	0.09 [€]
0.22 [€/day]	231 [days]	50.82 [€]
Total:		50.91 [€]

Figura 35 – Layout do menu de consulta do preço da energia.

Após ter seleccionado os parâmetros da consulta o utilizador selecciona o botão *Enter* sendo calculado o preço da energia e apresentado na tabela da Figura 35.

As funcionalidades gráficas explicadas anteriormente são aplicadas no ficheiro *activity_price.xml*. No ficheiro *Price_Activity.java* é implementado código que irá conferir às funcionalidades gráficas descritas, a funcionalidade de calcular o preço da energia elétrica consumida no período seleccionado.

Para o efeito foi necessário criar dois botões para inserir as datas e um botão para escolher o *power meter*. A forma como é criado o botão relacionado com a escolha do *power meter* é igual à forma como é criado no menu *Actual Read*. Os botões utilizados para inserir a data foram criados ambos da mesma forma, com variáveis distintas que vão receber as datas escolhidas pelo utilizador, uma relativa à data de início e a outra à data de fim da consulta. A sua criação passa por iniciar uma variável do tipo *Calendar*, denominada por *date* que irá receber a data atual a qual será representada na aplicação. Depois é ativada a propriedade

setOnClickListener na caixa de texto onde é apresentada a data a qual o utilizador seleciona para alterar a data, esta propriedade permite executar uma ação sempre que esta caixa de texto é selecionada. Neste caso a ação passa por iniciar uma variável *DatePickerDialog*, que uma vez iniciada apresenta no ecrã da aplicação um calendário onde o utilizador pode selecionar uma data. Após a escolha, a data é retornada em três variáveis incluídas na declaração da variável do tipo *DatePickerDialog*, que são as variáveis *year*, *monthOfYear* e *dayOfMonth*, por fim estas variáveis são apresentadas na caixa de texto onde o utilizador selecionou a data, mostrando assim a data escolhida. E também são armazenadas na variável *date1* do tipo *String* no formato, ano-mês-dia.

```

/****Criação do botão de data ****/

textView32.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        DatePickerDialog datePickerDialog = new
DatePickerDialog(Price_Activity.this, new
DatePickerDialog.OnDateSetListener() {

            public void onDateSet(DatePicker datePicker, int year, int
monthOfYear, int dayOfMonth) {

                String.format("%02d", monthOfYear) + "-" +
String.format("%02d", dayOfMonth) + " 00:00:00"; //+".*";

            }
        }, year, month, day);
    }
});

```

Script 35 – Criação do botão de seleção da data.

A selecionar o botão *Enter*, através da funcionalidade *onClick*, é acionada a função *sendMessage2*, que pede ao servidor os dados para calcular o preço da energia no período selecionado e depois efetua o cálculo do preço.

O processo de efetuar o pedido ao servidor é similar ao processo explicado nos menus anteriores, porque o método de comunicação utilizado é igual. O ficheiro acedido é um ficheiro que o servidor cria e onde coloca todos os valores registados referentes às energias, potências, tensões, correntes, THD, frequência, endereços de *power meters*, potência contratada em cada *power meter*, o preço da energia em €/Kwh e em €/dia, no formato GSON, que será posteriormente descodificado na aplicação.

Desta forma o servidor não fica sobrecarregado com ficheiro PHP, que seriam consultados conforme a função de consulta pretendida, apenas tem um ficheiro que recolhe todos os

dados necessários para o funcionamento dos menus da aplicação e na aplicação são apresentados os dados relevantes conforme o menu que se está a utilizar.

Após a receção dos dados, estes são processados de forma a extrair os dados relevantes para o menu do preço da energia elétrica. Este processamento é feito através de um ciclo que percorre todos os campos da mensagem de resposta do servidor e em cada campo retira apenas os valores referentes à energia ativa, preço da energia em €/Kwh, preço em €/dia e a data e hora desse registo. O resultado é um vetor com os valores da energia ativa, outro vetor com as datas de cada registo, uma variável com o preço em €/Kwh e outro em €/dia. Posteriormente é calculada a energia ativa consumida no intervalo de tempo selecionado e calculado o número de dias correspondente. O valor da energia consumida é calculado através da diferença entre o último valor de energia registado e o primeiro valor de energia registado no intervalo temporal escolhido. O preço da energia ativa é calculado multiplicando o valor anterior pelo preço da energia para a *power meter* em questão. Na fase seguinte através das datas dos registos, recolhidas, é calculada a diferença em dias, entre a última data registada e a primeira data registada. Como esta diferença representa o número de dias contabilizados para o consumo de energia na instalação, então esta diferença é multiplicado pelo preço recolhido em €/dia relativo ao valor de potência contratado. No fim para obter o preço total a pagar pelo consumo de energia elétrica na instalação são somadas as duas componentes, a componente relativa à potência contratada, calculada com base no número de dias de utilização e a componente referente à energia ativa consumida no período. No fim a soma das duas componentes é apresentada numa tabela como é possível observar na Figura 35, assim como as componentes que representam os cálculos intermédios, que podem servir para informar melhor o consumidor acerca do seu consumo energético.

```
/**Calculo do preço da energia ***/  
  
for (int i = 0; i < founderArray.length; i++) {  
    if (founderArray[i].PM_address.equals(message) &&  
founderArray[i].data_hora.compareTo(date1) >=0 &&  
founderArray[i].data_hora.compareTo(date2) <=0) {  
  
        data_graph.add(founderArray[i].energy_P);  
price=founderArray[i].energy_price;  
priceday=founderArray[i].energy_price_day;  
date.add(founderArray[i].data_hora);  
    }  
}  
  
energy=data_graph.get(data_graph.size() - 1) - data_graph.get(0);  
price_energy = energy * price;
```

```

long diff = lastDate.getTime() - firstDate.getTime();
long diffDays = diff / (24 * 60 * 60 * 1000);

price_energy_day = diffDays * priceday;

textView41.setText(String.format("%.2f", price_energy +
price_energy_day) + " [€]");

```

Script 36 – Calculo do preço da energia.

Ao seleccionar o menu *Get History* o utilizador é direccionado para o menu onde é apresentado um gráfico com as energias consumidas ou potências durante um período de tempo escolhido pelo utilizador referente ao *power meter* que o utilizador seleccionar.

O aspeto gráfico deste menu é visível na Figura 36.

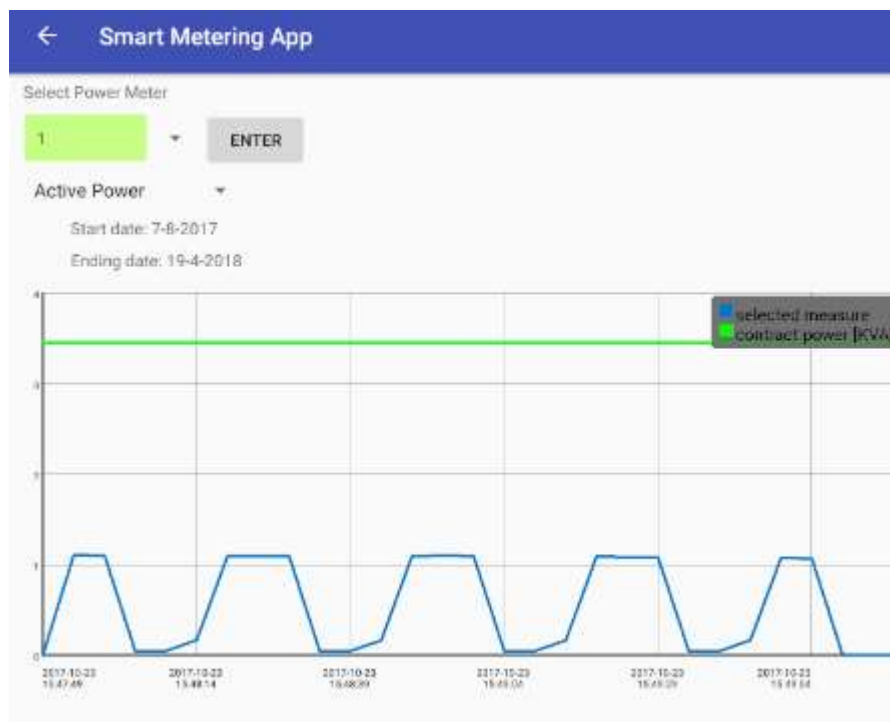


Figura 36 – Layout do menu de consulta dos consumos ao longo do tempo.

O funcionamento deste menu é idêntico ao funcionamento do menu *Energy Price* a diferença é que após o utilizador escolher o *power meter* que pretende consultar, a data de início e fim da consulta e o tipo de grandeza que pretende observar, o gráfico é apresentado de acordo com a seleção do utilizador. Este gráfico permite fazer *zoom* sempre que o utilizador pretender observar um conjunto de registos mais detalhadamente. No eixo vertical é

apresentada a medida referente à grandeza escolhida pelo utilizador e no eixo horizontal do gráfico é apresentada a data de cada registo.

Após o utilizador selecionar o botão *Enter*, é executada a função *sendMessage2* que efetua um pedido ao servidor para este devolver os registos com as medições efetuadas pelo *power meter*. O pedido, a receção e o processamento dos dados é realizado da mesma forma que no menu *Energy Price*, a diferença no processamento passa por verificar qual a grandeza que o utilizador pretende visualizar e então só os valores referentes a essa grandeza serão escritos no vetor *data_graph*.

```
//****Recolha da grandeza a apresentar ****/
if (textSpinner.equals("Active Energy")) {
    data_graph.add(founderArray[i].energy_P);
    stringleg.add(founderArray[i].data_hora);
}
```

Script 37 – Recolha da grandeza a apresentar.

Para apresentar o gráfico é utilizada a biblioteca *GraphView* que disponibiliza um conjunto de funcionalidades que permite tornar o gráfico mais perceptível e acrescentar legendas para que o utilizador tenha noção da grandeza que esta a ser apresentada e de como esta varia no tempo. Primeiro é iniciado o gráfico e a variável *series* do tipo *LineGraphSeries*, que recebe os dados referentes à grandeza escolhida pelo utilizador.

```
//****Inicialização do gráfico ****/
DataPoint[] dataPoints = new DataPoint[data_graph.size()];
for (int i = 0; i < data_graph.size(); i++) {
    dataPoints[i] = new DataPoint(i, data_graph.get(i));
    dataPoints2[i] = new DataPoint(i,
contract_power.get(i));

GraphView graph = (GraphView) findViewById(R.id.graph);
LineGraphSeries<DataPoint> series = new
LineGraphSeries<DataPoint>(dataPoints);
```

Script 38 – Inicialização do gráfico.

Depois é adicionado ao gráfico os valores referentes à grandeza escolhida, assim como uma segunda linha que representa o valor da potência contratada na instalação elétrica. Esta apresentação ocorre somente no caso da grandeza escolhida pelo utilizador ser a potência ativa, reativa ou aparente.

```
/**Preencher o gráfico */
graph.addSeries(series);
series.setTitle("selected measure");
graph.getViewport().setScalableY(true);

LineGraphSeries<DataPoint> series2 = new
LineGraphSeries<DataPoint>(dataPoints2);
graph.addSeries(series2);
```

Script 39 – Construção do gráfico com valores e legenda.

CONCLUSÕES

Na implementação prática deste projeto foi desenvolvido um sistema de aquisição de dados, que integra um Raspberry Pi e um *power meter* 710, que juntos representam o sistema de aquisição de dados.

Para o armazenamento dos dados recolhidos foi implementada uma base de dados num servidor no qual são registados os dados mais relevantes.

Após a implementação desta secção do projeto comprovou-se a validade do sistema desenvolvido para recolher e armazenar dados referentes ao consumo energético numa instalação elétrica.

A segunda secção do sistema corresponde ao desenvolvimento de um sistema de informação, constituído por uma aplicação *web* e uma aplicação Android, que foram desenvolvidas com o objetivo de apresentar os dados recolhidos ao consumidor, para que este tome conhecimento e possa tomar medidas em relação ao seu consumo energético. Após a implementação das duas aplicações verificou-se que estas cumpriam o objetivo destinado, que era apresentar os dados relativos ao consumo energético de uma instalação e sensibilizar o consumidor.

RESULTADOS PRÁTICOS

Neste capítulo serão apresentados e analisados resultados experimentais obtidos durante os testes de campo do sistema desenvolvido, com o objetivo de demonstrar as funcionalidades do sistema e a sua utilidade nas áreas da monitorização dos consumos, da gestão de energia e do *smart metering*. Os valores apresentados traduzem as medições efetuadas, utilizando diferentes eletrodomésticos presentes numa instalação elétrica residencial.

RESULTADOS OBTIDOS

TERMOVENTILADOR

De seguida serão apresentados os resultados referentes a um período de utilização de um termoventilador, com dois níveis de potência, nomeadamente 1 kW e 2 kW (potência máxima do aparelho) e com termostato regulável, com uma tensão de alimentação que pode variar entre 220 V e 240 V.

Os resultados obtidos foram obtidos durante um período de 26 minutos de utilização do termoventilador, onde se utilizou os dois níveis de potência, variou-se o termostato e também se utilizou o termoventilador no modo de ventoinha.

Na Figura 37 é apresentado um gráfico com os valores das potências aparente, ativa e reativa registados onde é possível observar em detalhe os níveis de potência do termoventilador, assim como o seu comportamento durante a transição entre potências e modos de funcionamento.

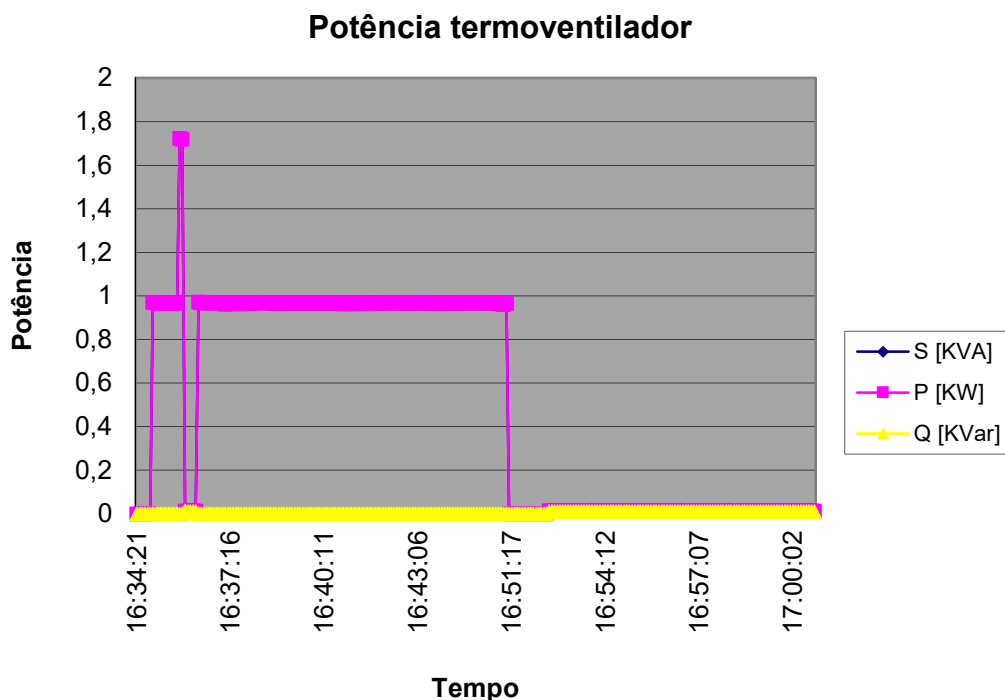


Figura 37 – Gráfico temporal da potência consumida pelo termoventilador.

Na Figura 37, entre as 16 horas e 34 minutos e as 16 horas e 35 minutos, observa-se o período em que o termonventilador funcionou no modo de aquecedor no nível de potência ativa de 1 kW, sendo que a potência ativa máxima consumida neste modo chega aos 0.96 kW. Às 16 horas e 35 minutos regista-se um pico de potência, sendo este o intervalo de tempo em que o termoventilador funcionou no nível de potência ativa igual a 2 kW, atingindo um nível máximo de potência ativa registado de 1.72 kW. Entre as 16 horas e 36 minutos e as 16 horas e 45 minutos foi o período em que o termoventilador esteve a funcionar como ventoinha, sendo que neste modo o consumo de potência é idêntico ao consumo no modo de aquecedor, cerca de 1 kW. Analisando o resultado, no referido período, verificou-se que o valor de potência ativa máxima consumida foi igual a 0.96 kW. Como se trata de um aparelho de natureza resistiva, esperava-se que a potência reativa consumida apresenta-se valores muito próximos de zero, pois o aparelho tem uma ventoinha o que implica a presença de um motor elétrico, que consome potência reativa dada a sua natureza de funcionamento, verificando-se valores na ordem dos 0,009 kVAR.

Na Figura 38 é apresentado o valor das energias presente nos registos do *power meter*, que representa o consumo cumulativo de energia do termoventilador ao longo do seu período de funcionamento desde o último *reset*.

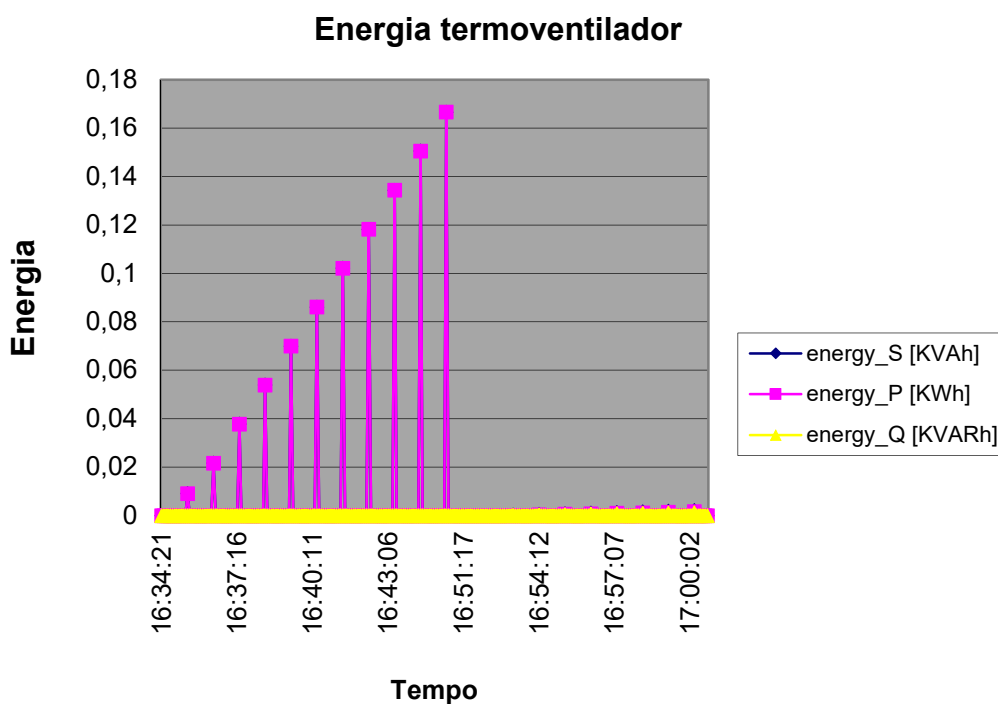


Figura 38 – Gráfico temporal da energia consumida pelo termoventilador.

Observando o gráfico da Figura 38 confirma-se que o termoventilador num período de aproximadamente 10 minutos em funcionamento consumiu aproximadamente 0.17 kWh.

Comparando os resultados obtidos com as especificações do termoventilador confirma-se a validade dos valores obtidos, e da solução técnica implementada para adquirir dados referentes ao consumo energético de uma instalação elétrica.

AQUECEDOR A ÓLEO

Foram recolhidos valores referentes ao consumo de energia de um aquecedor a óleo, com uma potência ativa máxima de 1.4 kW, durante um período de aproximadamente 30 minutos, na sua potência máxima. Durante este período temporal obtivemos os valores de potência ativa, reativa e aparente e energias ativa, reativa e aparente que são apresentados nas figuras 36 e 37.

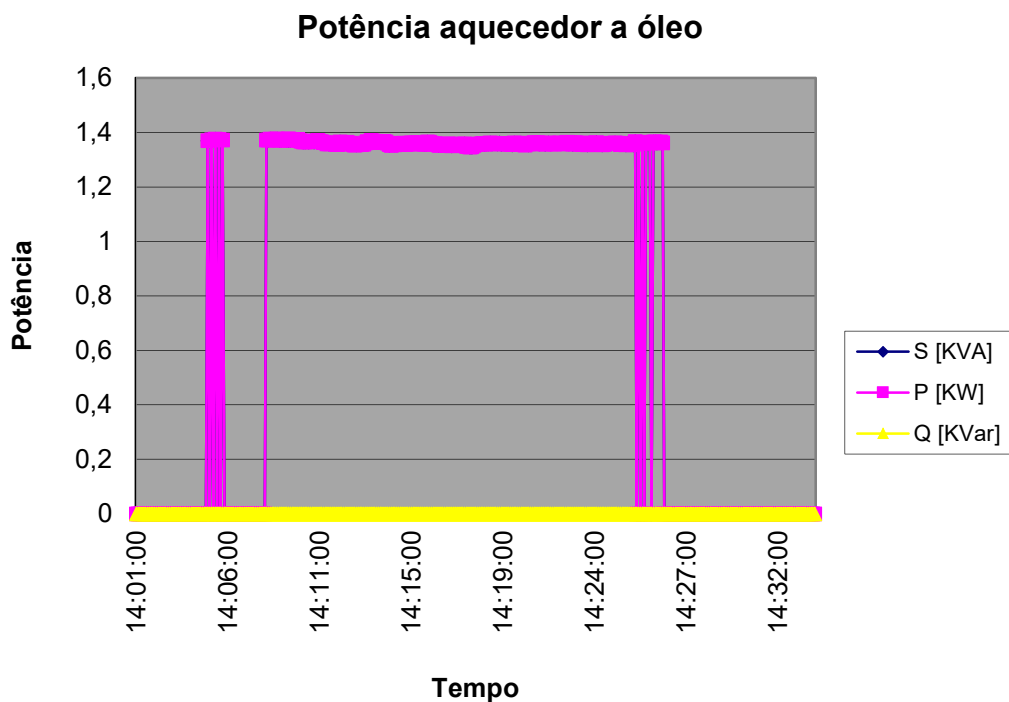


Figura 39 – Gráfico temporal da potência consumida pelo aquecedor a óleo.

Observando o gráfico da Figura 39 verifica-se que durante o período de tempo mencionado o aquecedor teve um consumo de potência praticamente constante, na ordem dos 1.37 kW. Como o aquecedor a óleo é uma carga resistiva, não apresenta consumo de potência reativa, resultando em valores de potência ativa e aparente iguais, como se constata na Figura 39.

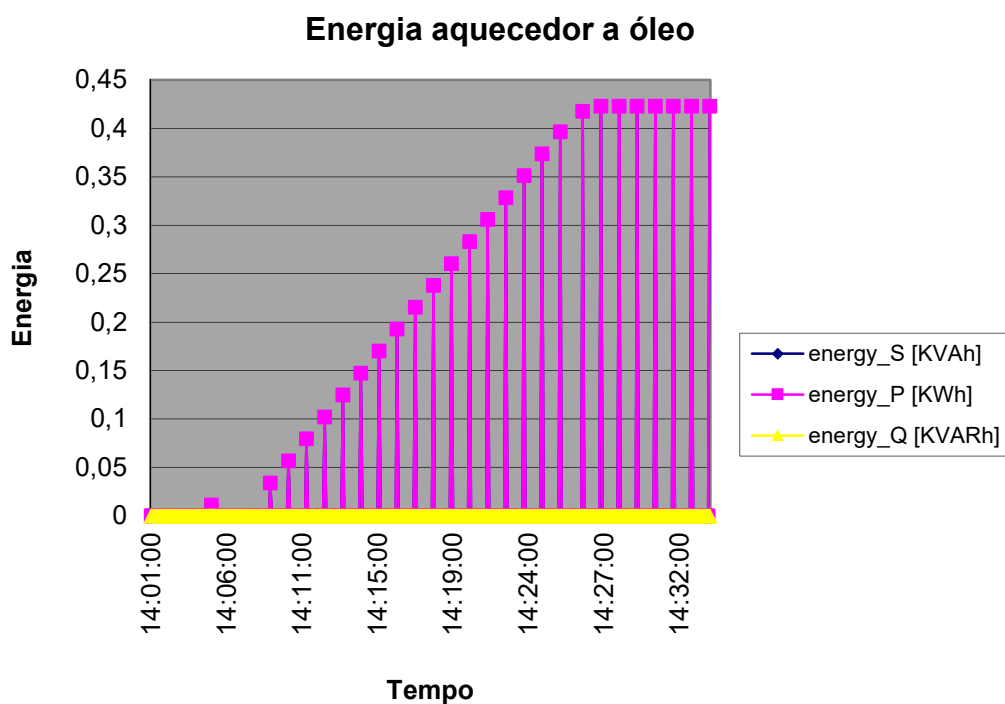


Figura 40 – Gráfico temporal da energia consumida pelo aquecedor a óleo.

Na Figura 40 é demonstrado em forma gráfica o valor da energia, presente nos registos do *power meter* relativa à utilização do aquecedor, verificando-se um consumo de energia ativa na ordem dos 0.423 kWh.

TORRADEIRA

De seguida serão apresentados os resultados referentes a um período de utilização de uma torradeira, com uma potência ativa de 1.150 kW, com uma tensão de alimentação que pode variar entre 220 V e 240 V, e sete programas distintos.

Os resultados obtidos referem-se a um período de 3 minutos de utilização, que corresponde à duração para fazer uma torrada, onde se utilizaram todos programas permitidos pelo aparelho, embora a variação entre estes programas não tenha afetado de forma visível a potência nem a energia consumida durante o período de utilização.

Na Figura 41 é apresentado um gráfico com os valores das potências: aparente, ativa e reativa registados onde é possível observar em detalhe o nível de potência ativa da torradeira durante um ciclo de funcionamento.

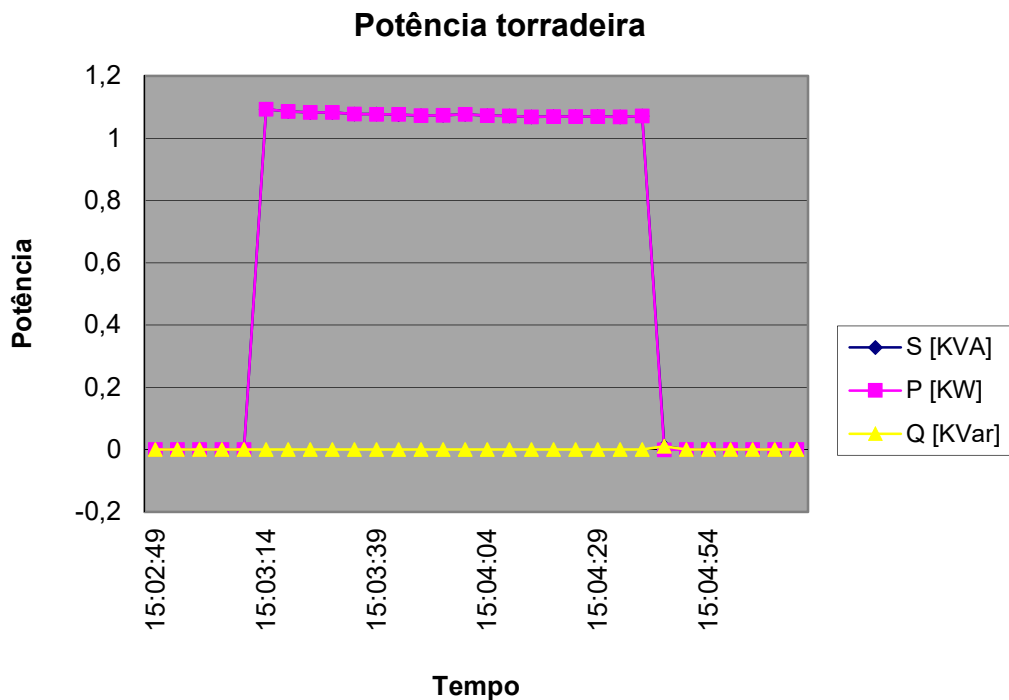


Figura 41 – Gráfico temporal da potência consumida pela torradeira.

Na Figura 41 observa-se que as potências ativa e aparente são iguais, este fenómeno deve-se ao facto de uma torradeira ser um aparelho resistivo, logo, não consome potência reativa. A potência ativa máxima consumida pela torradeira atinge os 1.09 kW e a mínima atinge os 1.068 kW, o que coincide com a potência máxima da torradeira, presente nas especificações técnicas.

Observando o gráfico da Figura 42 confirma-se que a torradeira num ciclo de funcionamento, 3 minutos, consumiu aproximadamente 0.027 kWh.

Mais uma vez, a análise dos resultados obtidos durante um ciclo de funcionamento de torradeira e a comparação dos mesmos com as especificações do aparelho, permite comprovar a validade do sistema desenvolvido para efetuar medições relativas ao consumo energético de uma instalação elétrica.

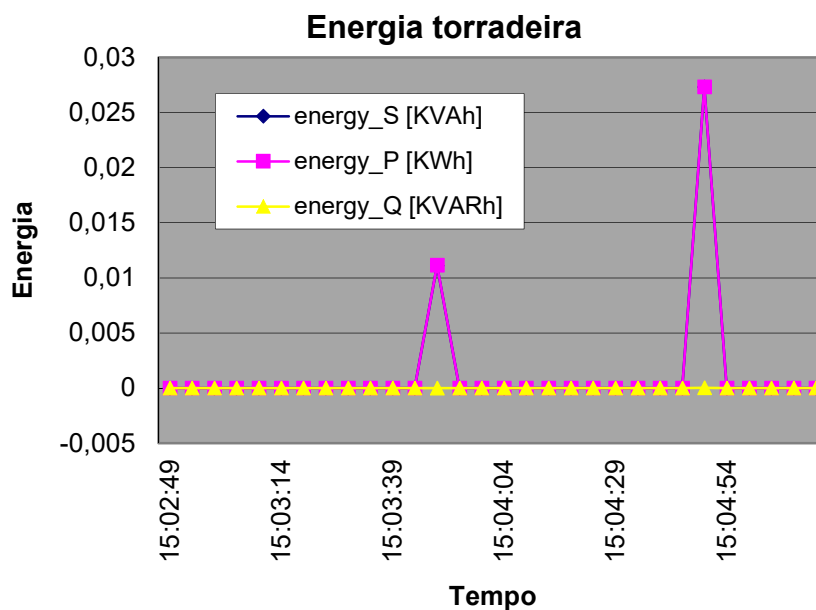


Figura 42 – Gráfico temporal da energia consumida pela torradeira.

FONTE DE ALIMENTAÇÃO DC LABORATORIAL

Foi utilizada uma fonte de alimentação DC, presente no laboratório de comunicações, a alimentar o conversor RS232/485 da solução apresentada. Procedeu-se à monitorização dessa carga durante 4 minutos. O valor da potência ativa esse período foi sensivelmente constante igual a 0.043 kW sendo o valor da potência reativa igual a 0.039 kVA, como é possível observar na Figura 43.

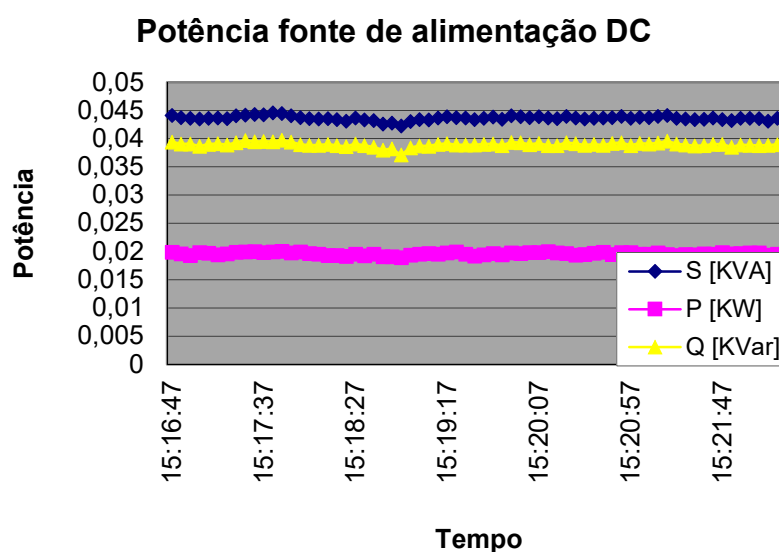


Figura 43 – Gráfico temporal da potência consumida pela fonte de alimentação DC.

No período de 4 minutos a fonte de alimentação DC teve um consumo de energia ativa na ordem dos 0.019 kWh e de energia reativa na ordem dos 0.039 kVARh, como é possível observar na Figura 44.

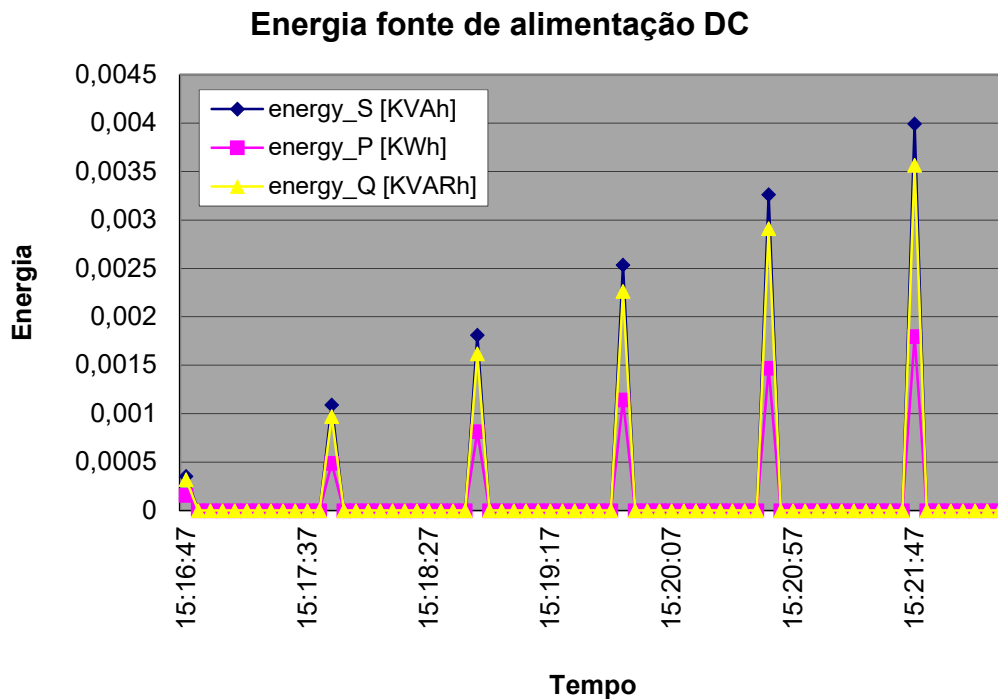


Figura 44 – Gráfico temporal da energia consumida pela fonte de alimentação DC.

A potência e energia reativa consumida pela fonte de alimentação DC justifica-se devido ao transformador e/ou bobine do conversor AC/DC que faz parte da fonte de alimentação, por isso verificam-se valores de potência reativa e energia reativa positivos.

CARREGADOR DE UM LAPTOP

Retiraram-se valores durante uma hora, de um carregador de um *laptop*, com uma tensão de entrada que pode variar entre 100 V e 240 V e que consome uma corrente de 2 A.

A partir dos valores obtidos verifica-se uma variação da corrente consumida pelo carregador, entre 1.5 A e 2.7 A. Na Figura 45 é apresentado um gráfico com a variação da potência ativa, reativa e aparente durante uma hora de funcionamento do carregador.

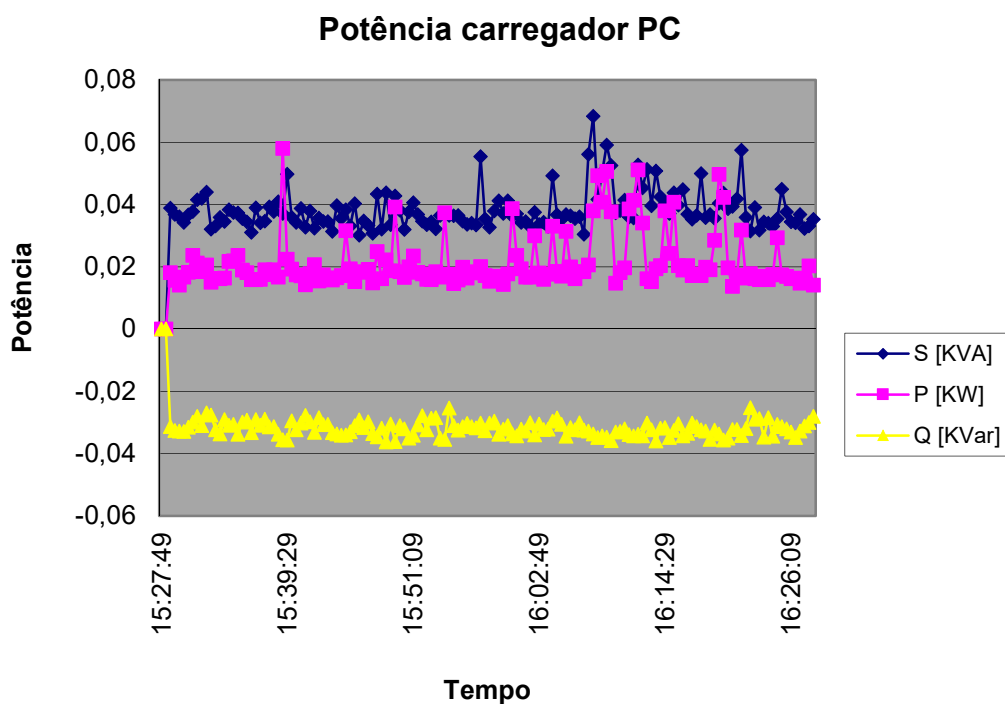


Figura 45 – Gráfico temporal da potência consumida pelo carregador de um PC.

Observando a Figura 45 verifica-se um consumo aproximadamente constante de potência ativa na ordem dos 20 W e de potência reativa na ordem dos -20 VAR. Este consumo negativo de potência reativa justifica-se devido ao facto do carregador utilizar condensadores para retificar a onda de tensão proveniente da rede elétrica, para fazer a conversão de corrente alternada para corrente contínua de forma a ser possível carregar a bateria do PC.

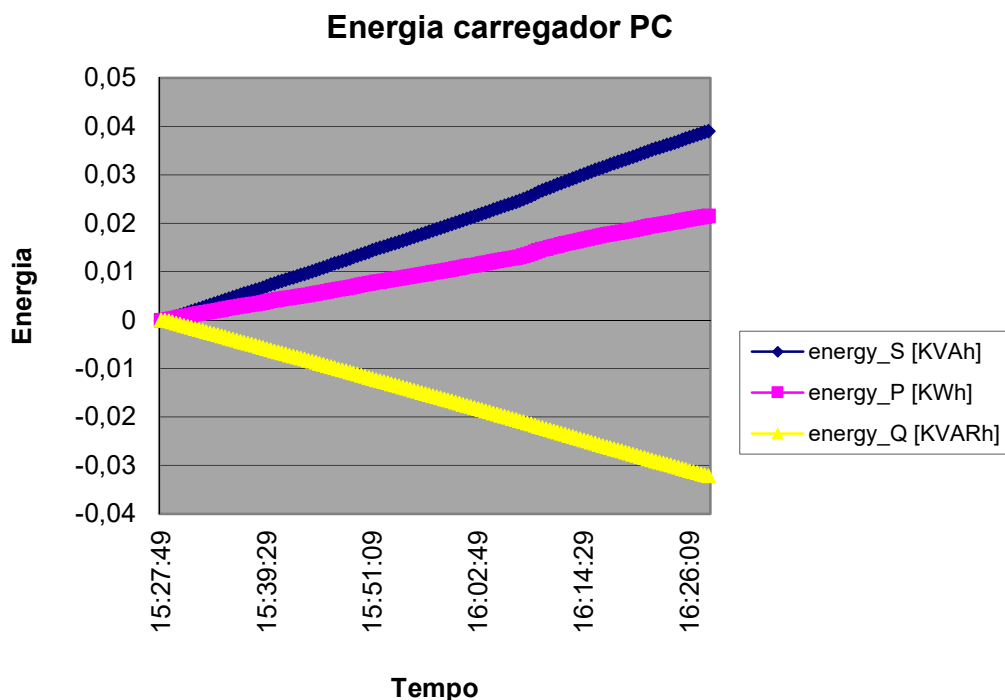


Figura 46 – Gráfico temporal da energia consumida pelo carregador de um PC.

Na Figura 46 observa-se o consumo de energia ativa, reativa e aparente do carregador, verificando-se um consumo de energia ativa no fim do período de utilização de 21.5 Wh e um consumo de energia reativa de -32 VARh.

Como um carregador de um *laptop* é um circuito eletrónico que transforma a tensão da rede numa fonte de tensão contínua de forma a alimentar a bateria do *laptop* que funciona em corrente contínua. Da análise dos resultados verifica-se uma distorção significativa nas ondas de tensão e corrente provenientes da rede. Esta distorção pode ser medida através da distorção harmónica total. De seguida na Figura 48 e Figura 47 são apresentadas os valores de THD da tensão e corrente na referida rede. Os valores de THD da tensão próximos de 3% devem-se ao facto do transformador do PT se encontrar saturado. Os valores de THD da corrente são elevados tendo em conta a distorção na onda da corrente, resultante do circuito de eletrónica de potência.

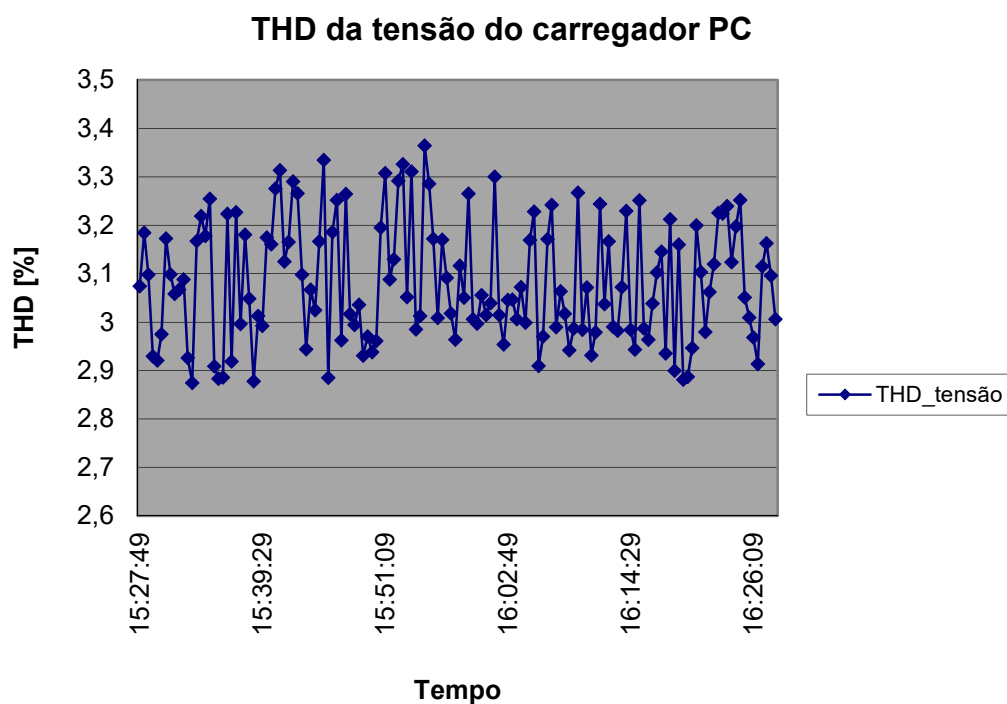


Figura 47 – Gráfico temporal do THD da tensão do carregador de um PC.

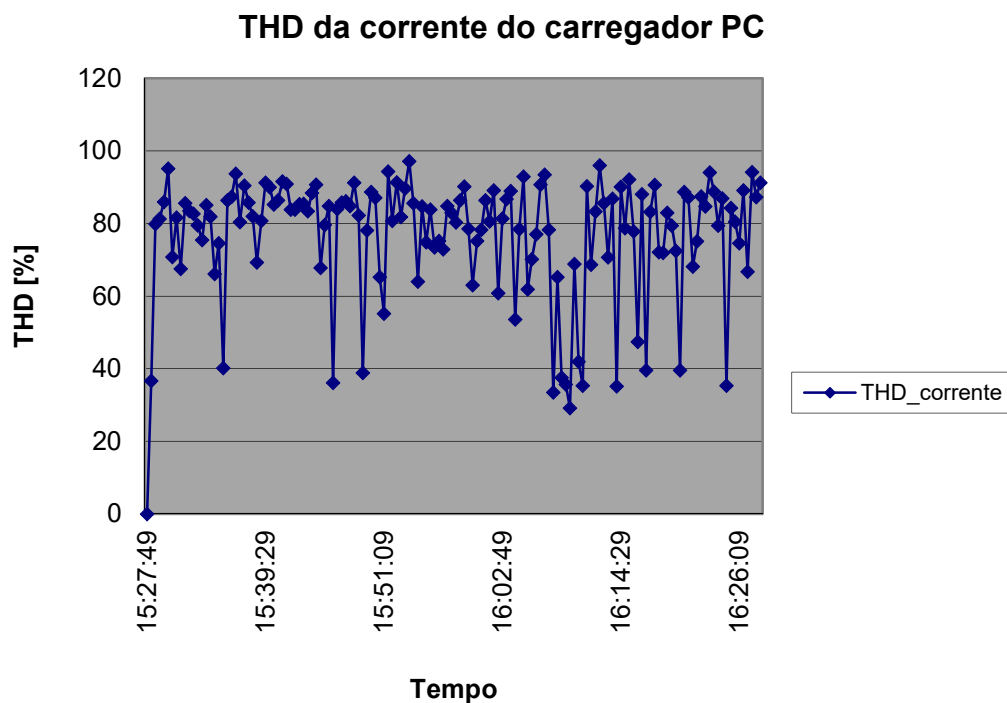


Figura 48 – Gráfico temporal do THD da corrente do carregador de um PC.

Uma vez que um carregador de um *laptop* é um conversor eletrónico AC/DC, o fator de potência registado é diferente de um, devido à sua componente reativa que faz com que este absorva energia, consumindo assim potência reativa. No caso apresentado observou-se dentro do período de funcionamento, uma variação do fator de potência entre 0.45 e 0.85, aproximadamente. Na Figura 49 é apresentado um gráfico onde é possível observar em pormenor a variação do fator de potência ocorrida durante um ciclo de carga do *laptop*.

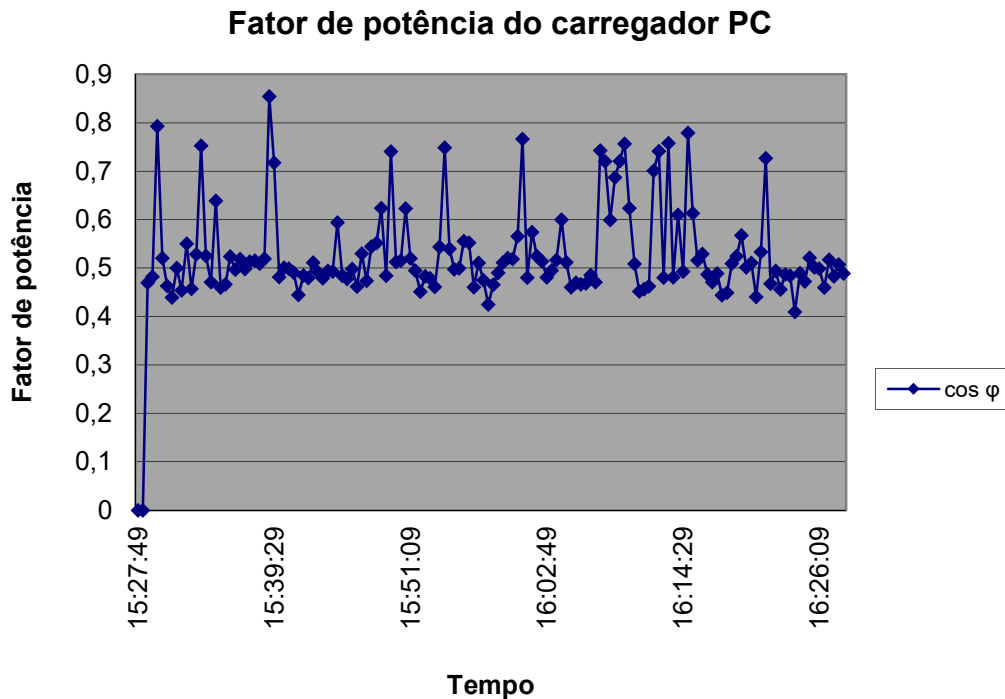


Figura 49 – Gráfico temporal do fator de potência do carregador de um PC.

TOSTADEIRA

De seguida serão apresentados os resultados referentes a um período de utilização de uma tostadeira, com uma potência ativa de 0,64 kW e apenas um nível de potência ativa, não sendo possível variar este parâmetro, com uma tensão de alimentação de 230 V.

Os resultados obtidos foram obtidos durante um período de 5 minutos de utilização, que corresponde à duração para fazer uma tosta.

Na Figura 50 é apresentado um gráfico com a variação dos valores das potências aparente, ativa e reativa registados durante o período de funcionamento, onde é possível observar em detalhe o nível de potência do aparelho.

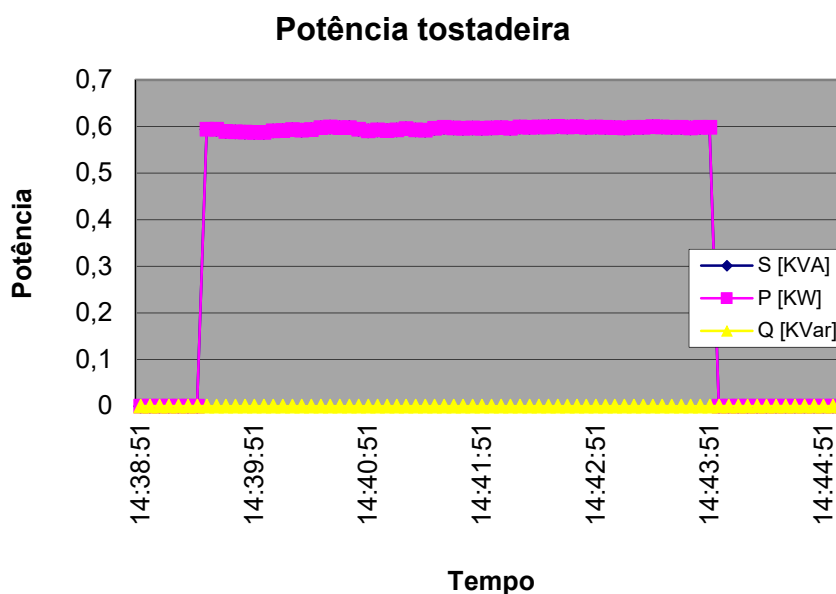


Figura 50 – Gráfico temporal da potência consumida pela tostadeira.

Sendo o aparelho de natureza resistiva, é natural que a potência reativa consumida pelo mesmo seja zero. Logo as potências ativa e aparente vão coincidir. Na Figura 50 verifica-se que o nível de potência ativa consumida é constante e que corresponde a 590 W. O valor de potência ativa obtido é próximo do valor dado nas especificações do aparelho (640 W).

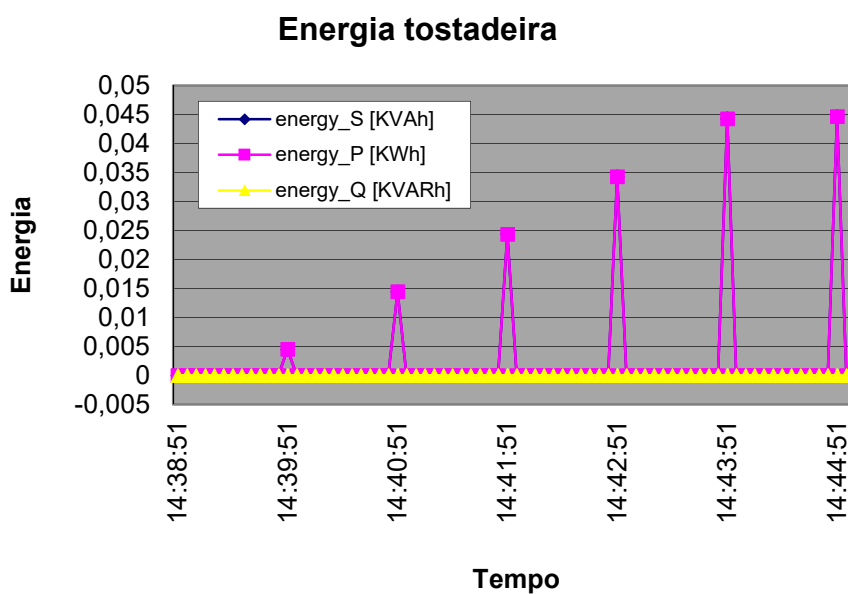


Figura 51 – Gráfico temporal da energia consumida pela tostadeira.

Observando a Figura 51 verifica-se um consumo de energia ativa na ordem dos 44.7 Wh, durante um ciclo de funcionamento.

MICROONDAS

De seguida serão apresentados os resultados referentes a um período de utilização de um microondas, com uma potência ativa de 1.2 kW, com uma tensão de alimentação que pode variar entre 230 V e 240 V.

Os resultados obtidos foram obtidos durante um período de 2 minutos de utilização, no nível de potência máxima. Na Figura 52 é apresentado um gráfico com os valores das potências aparente, ativa e reativa registados onde é possível observar em detalhe o nível de potência do microondas durante o ciclo de funcionamento.

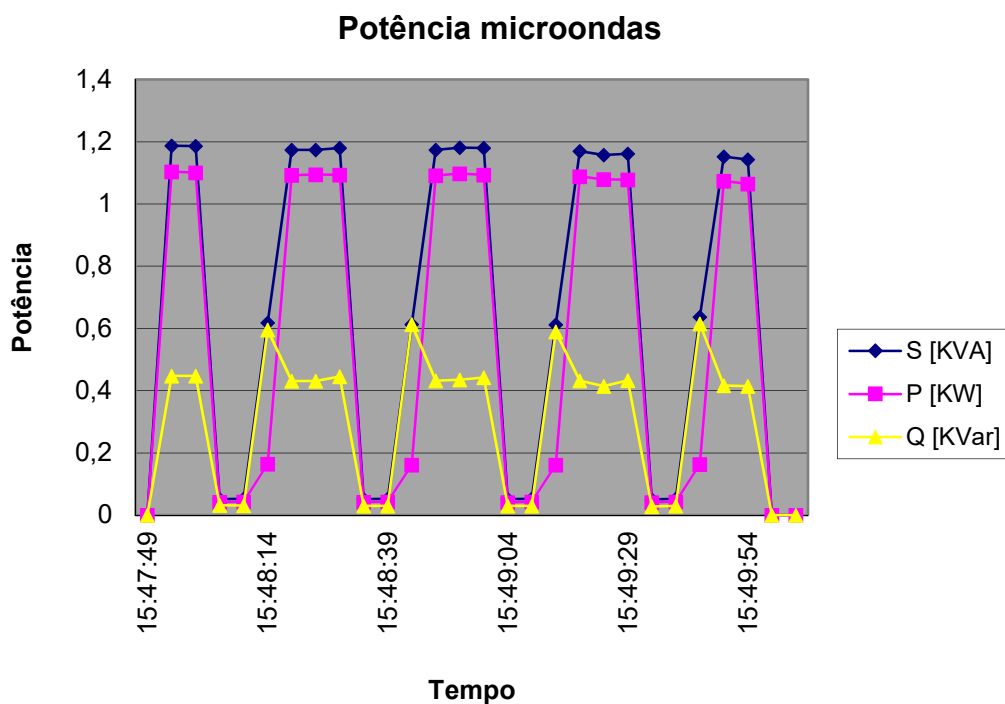


Figura 52 – Gráfico temporal da potência consumida pelo microondas.

Na Figura 52 observa-se uma diferença entre as potências ativa e aparente, este fenómeno deve-se ao facto de o microondas ser um aparelho que utiliza um *magnetrom* para gerar micro-ondas, com uma frequência igual a 2450 MHz, baseado em circuitos eletrónicos e cavidades ressonantes. A potência ativa máxima consumida pelo microondas atinge os

1.1 kW e a potência reativa máxima atinge os 440 W. A potência ativa máxima registada coincide com a potência máxima apresentada nas especificações.

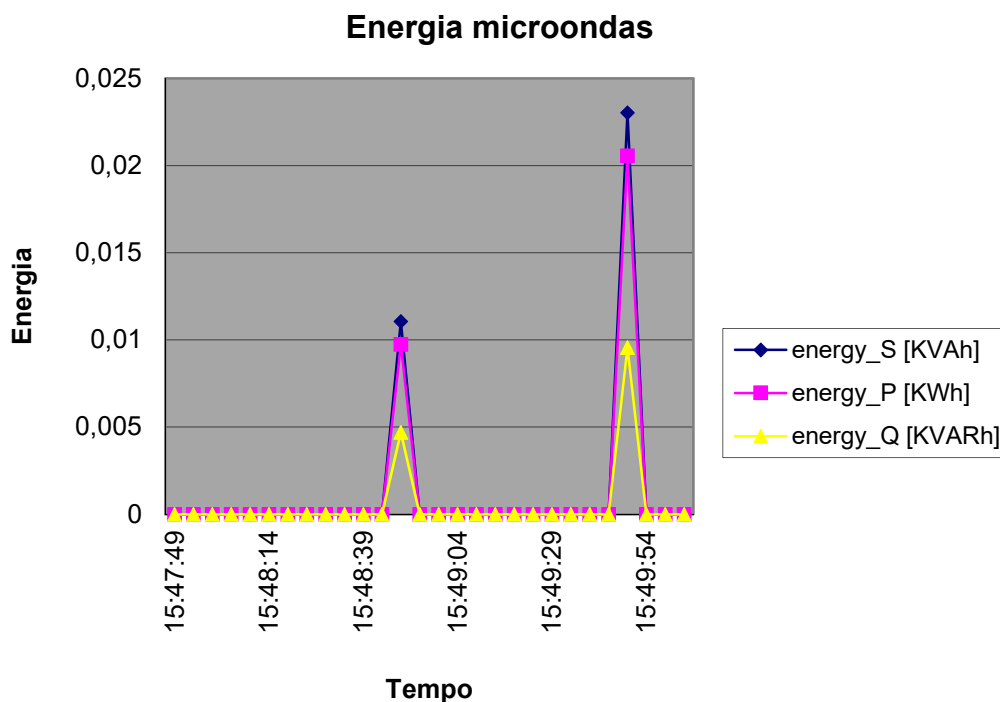


Figura 53 – Gráfico temporal da energia consumida pelo microondas.

Observando o gráfico da Figura 53 confirma-se que o microondas num ciclo de funcionamento, de 2 minutos, consumiu aproximadamente 20 Wh e 9.6 VARh, resultando num consumo de energia aparente de 23 VAh.

Mais uma vez, a análise dos resultados obtidos durante um ciclo de funcionamento do microondas e a comparação dos mesmos com as especificações do aparelho, permite comprovar a validade do sistema desenvolvido para efetuar medições relativas ao consumo energético de uma instalação elétrica.

Observando a corrente consumida pelo microondas durante um período de funcionamento, na Figura 54 verifica-se uma variação na corrente consumida pelo aparelho durante o tempo de funcionamento, esta variação acontece devido às características de funcionamento do *magnetrom* para gerar o campo eletromagnético. No caso apresentado a corrente variou entre os 0.23 A e os 5.2 A.

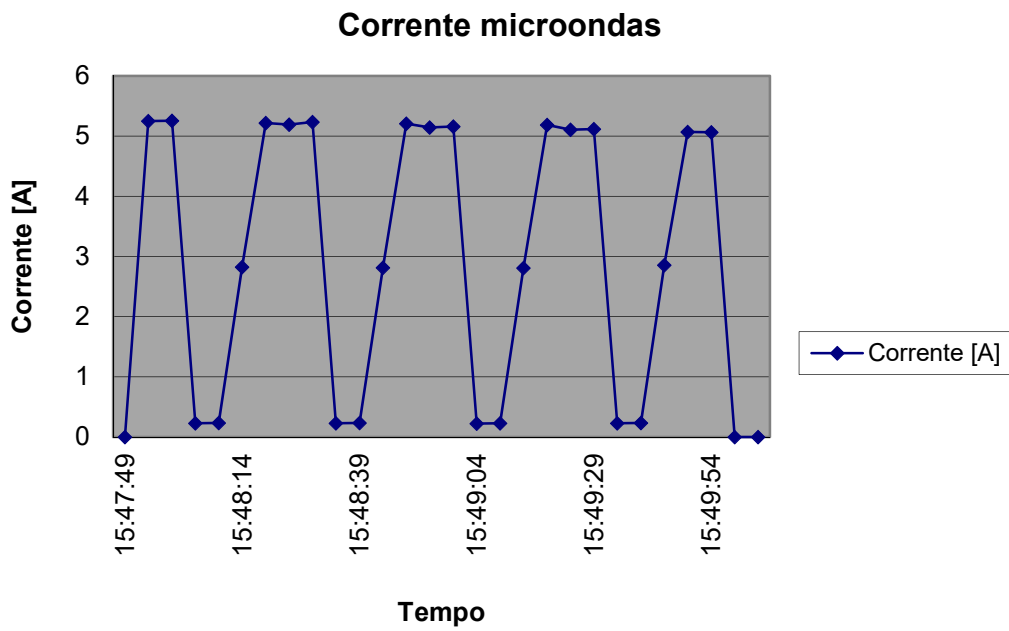


Figura 54 – Gráfico temporal da corrente consumida pelo microondas.

Observando a frequência na rede à qual o microondas estava ligado verifica-se uma variação desta, entre 49.95 Hz e 49.99 Hz. Esta variação está relacionada com as características das cargas da rede e com a velocidade de rotação dos geradores. Na Figura 55 é possível observar em pormenor esta variação.

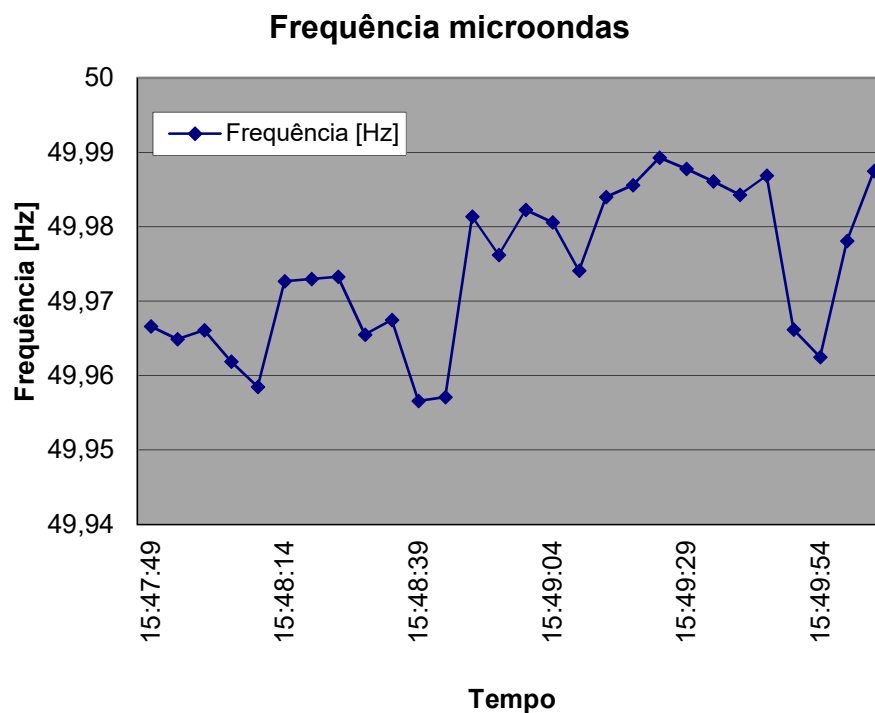


Figura 55 – Gráfico temporal da variação da frequência na rede elétrica.

FRIGORÍFICO

Retiraram-se valores durante aproximadamente quatro horas e 30 minutos, de um frigorífico, com uma tensão de entrada que pode variar entre 220 V e 230 V, que consome uma corrente de 0.7 A, e uma potência nominal de 110 W.

Na Figura 56 é apresentado um gráfico com a variação da potência ativa, reativa e aparente durante o tempo de funcionamento do frigorífico.

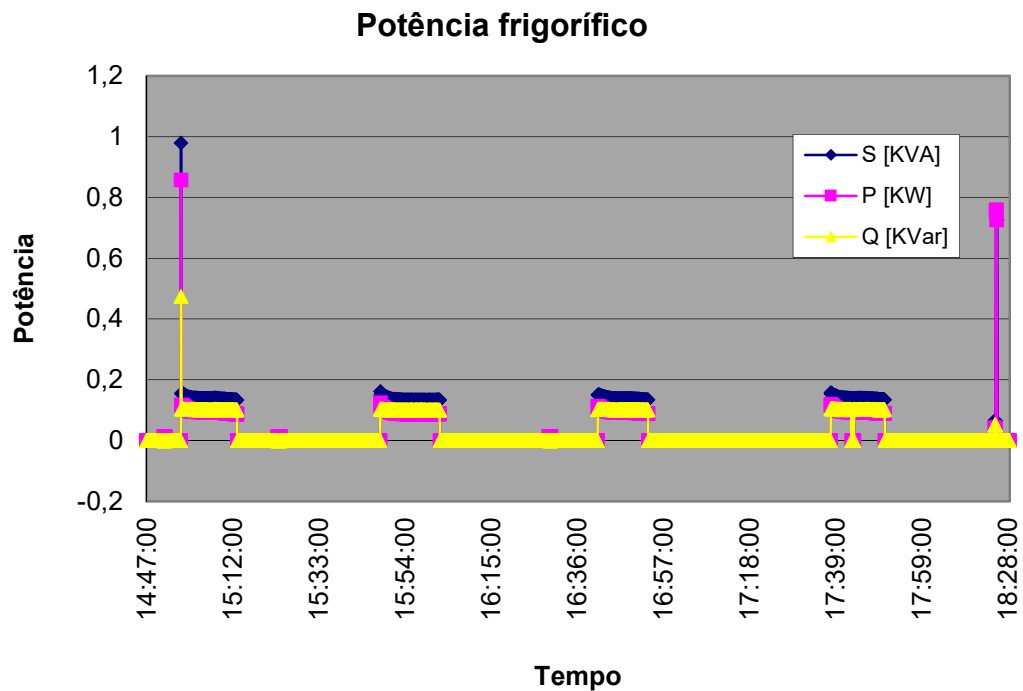


Figura 56 – Gráfico temporal da potência consumida pelo frigorífico.

Observando a Figura 56 verificam-se períodos onde existe consumo de potência ativa na ordem dos 0,14 kW e de potência reativa na ordem dos 0,1 kVAr. O consumo de potência reativa justifica-se devido ao facto do frigorífico ter um motor elétrico (carga indutiva) necessário ao funcionamento da máquina de frio. Os períodos de consumo de energia ocorrem com uma periodicidade de aproximadamente 40 minutos e com uma duração de 12 minutos. Estes períodos de consumo de energia refletem o funcionamento do compressor que é movido por um motor elétrico, ocorrem para que o frigorífico mantenha a temperatura no seu interior.

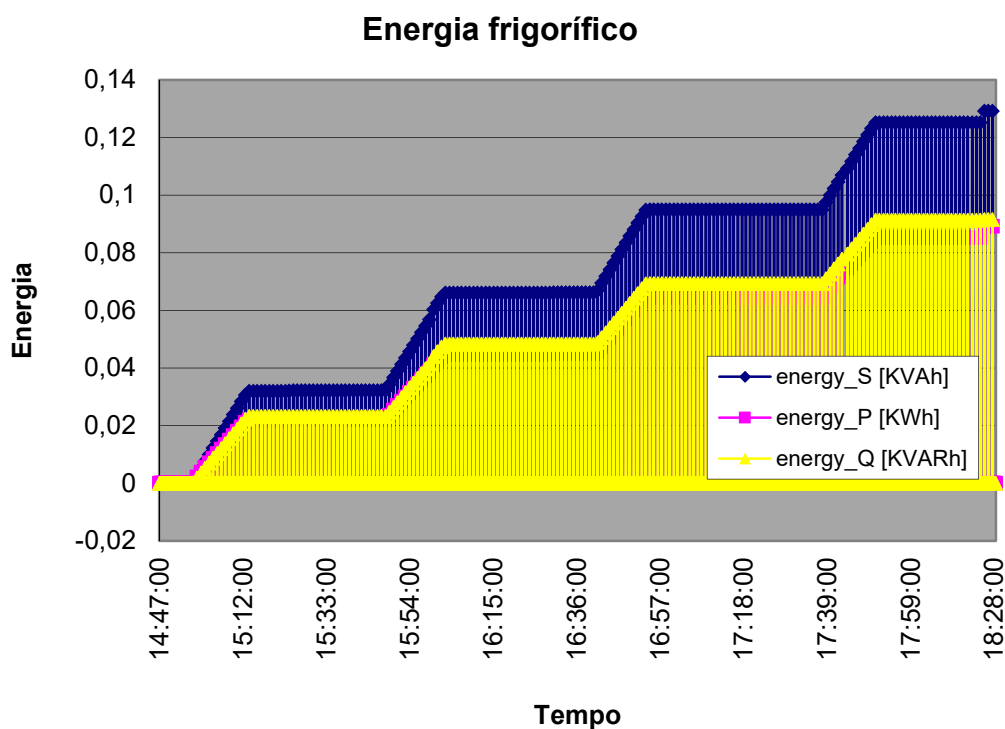


Figura 57 – Gráfico temporal da energia consumida pelo frigorífico.

Na Figura 57 observa-se o consumo de energia ativa, reativa e aparente do frigorífico, verificando-se um consumo de energia ativa no fim do período de utilização de 0.56 kWh e um consumo de energia reativa negativo de 0.06 kVARh, resultando num consumo de energia aparente de 0.62 kVAh.

Uma vez que um frigorífico tem uma componente indutiva, o fator de potência registado será diferente de um, tornando a potência aparente diferente da potência ativa. No caso apresentado observou-se dentro do período de funcionamento, um fator de potência próximo de 0.7 quando a máquina de frio estava em funcionamento. A Figura 58 apresenta um gráfico com a variação do fator de potência ocorrida durante um ciclo de leitura do frigorífico.

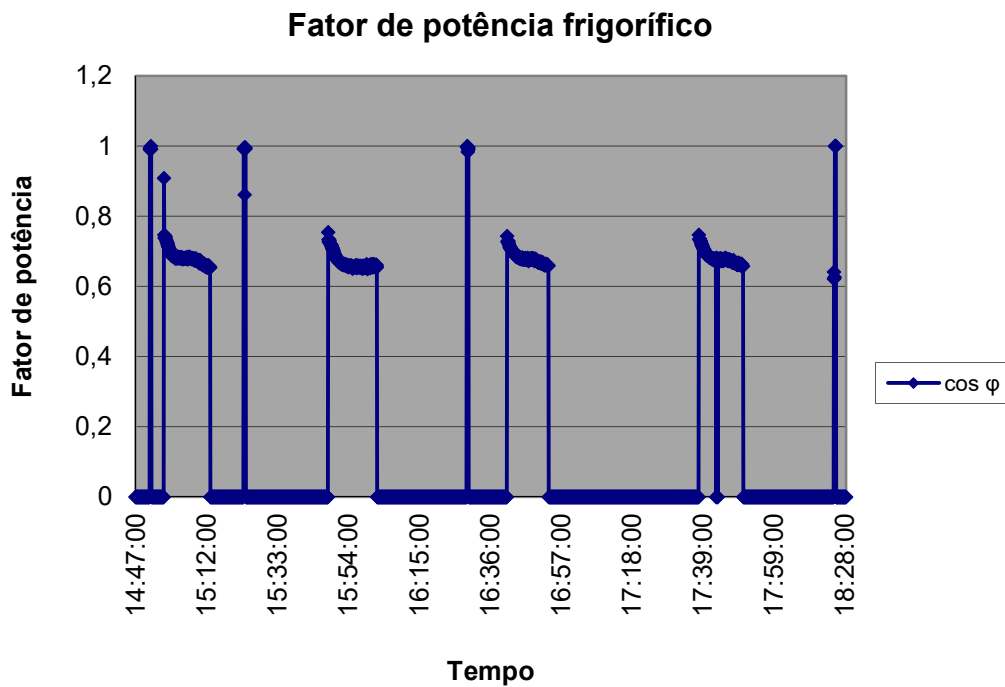


Figura 58 – Gráfico temporal do fator de potência do frigorífico.

CONCLUSÕES

Após a implementação do sistema de aquisição e apresentação de dados referentes ao consumo de energia numa instalação elétrica, estudou-se o consumo de vários equipamentos elétricos que costumam estar presentes nas nossas habitações, como frigoríficos, torradeiras, PC portáteis (carregador do PC) e aquecedores. A aquisição dos dados relativos a um ciclo de funcionamento dos eletrodomésticos utilizados permitiu comprovar a validade do sistema para fazer a aquisição de dados de uma instalação elétrica, através da comparação dos dados obtidos com as especificações dos eletrodomésticos utilizados.

CONCLUSÕES E TRABALHO FUTURO

O presente trabalho tinha por objetivo o desenvolvimento de aplicações para monitorização de consumos de energia em instalações elétricas. Para o efeito foi necessário implementar um *smart meter* proprietário para efetuar a monitorização do consumo de energia nas referidas instalações elétricas.

Numa primeira fase deste projeto utilizou-se o *power meter* que foi selecionado inicialmente PM710 da Schneider e um PC portátil como unidade de aquisição de dados. Esta primeira solução teve como objetivo a validação do conceito do sistema tendo-se desenvolvido uma aplicação em Matlab para efetuar a comunicação entre esse *power meter* e a base de dados alojada no portátil para efetuar o armazenamento dos dados recolhidos.

Posteriormente utilizando um Raspberry Pi foi possível tornar o sistema de aquisição de dados num dispositivo com capacidade para ser instalado facilmente numa instalação elétrica, com capacidade para adquirir valores relativos ao consumo de energia elétrica durante um período de tempo indeterminado e guardar os mesmos numa base de dados ou dispositivo de armazenamento externo.

Para completar o *smart meter* desenvolvido, de forma a cumprir o objetivo final deste projeto desenvolveram-se duas aplicações distintas para ambientes *web* e Android, que após a sua criação foram testadas e provaram a sua utilidade na sensibilização do consumidor, através da informação apresentada na interface das aplicações.

Para comprovar o funcionamento correto do sistema numa instalação elétrica, foram utilizados vários eletrodomésticos para efetuar leituras dos seus consumos num ciclo de funcionamento e posteriormente foram analisados os dados recolhidos, que comprovaram a validade do sistema para ser integrado numa rede *smart grid*, pois os resultados obtidos coincidiram com as especificações dos eletrodomésticos e foi possível obter resultados sem falhas do sistema desenvolvido durante os ciclos de funcionamento dos eletrodomésticos.

Considerando os resultados obtidos neste projeto e de modo a completar este estudo seria pertinente em trabalho futuro aprofundar os seguintes aspetos:

- Testar outros contadores digitais para além do *Power meter 710*;
- Testar o sistema desenvolvido numa habitação;
- Testar o sistema a uma escala maior, criando uma rede de dispositivos de aquisição de dados que comuniquem as leituras para uma base de dados;
- Aumentar o nível de segurança da informação que circula entre o sistema de aquisição de dados e a base de dados, assim como a informação que circula no sistema de informação;
- Adicionar outras funcionalidades pertinentes para a sensibilização do consumidor, às aplicações desenvolvidas e refinar a informação apresentada.

REFERÊNCIAS

- (Apache, 2018) Apache Development Notes. (Maio de 2018). Obtido de apache.org: <https://httpd.apache.org/dev/devnotes.html>
- (Arcari, 2017) Arcari, F. D., Costa, C., Pereira, C. E., Netto, J. S., Torres, G., Souza, M., Müller, I. (2017). Development of a WirelessHART – EnOcean Adapter for Industrial Applications. 2017 VII Brazilian Symposium on Computing Systems Engineering. IEEE. doi:10.1109/SBESC.2017.31
- (Centenaro, 2017) Centenaro, M., Vangelista, L., Kohnoy, R. (2017). On the Impact of Downlink Feedback on LoRa Performance. IEEE.
- (Chati, 2017) Chati, H. D., Salem, F. M. (2006). Sensing and Feedback Stimulation via The Wireless ZigBee Protocol. Proceedings of the 28th IEEE EMBS Annual International Conference. New York City, USA: IEEE.
- (Comer, 2017) Comer, D., Karandikar, R. H., Rastegarnia, A., Rouzbeh, F., Sruthi, P. C. (2017). WIST: Wi-SUN FAN Protocol Emulation Testbed. IEEE.
- (Di, 2017) Di, W., Ling, C., Lili, G., Yimei, W., Xuesong, J. (2017). Application Research of ZigBee Protocol in CPS Based on Multi-Agent. 2017 29th Chinese Control And Decision Conference (CCDC). Shenyang: IEEE.
- (Electric, 2004) Electric, S. (2004). Power Meter PM700 Manual de utilização. Lisboa: Schneider Electric.
- (Electric, 2018) Electric, S. (2018). Product data sheet Characteristics M8650A0C0E5E1B1A. Schneider Electric.
- (ETSI, 2010) ETSI. (2010). Machine-to-Machine communications (M2M); Smart Metering Use Cases. ETSI.

-
- (Faisal, 2012) Faisal, M. A. (2012). Securing Advanced Metering Infrastructure (AMI) in Smart Grid using Intrusion Detection System (IDS). Masdar: Masdar Institute of Science and Technology.
- (SModbus, 2018) Frequently Asked Questions What is Modbus. (Julho de 2018). Obtido de simplymodbus.ca: <http://www.simplymodbus.ca/faq.htm>
- (Gavazzi) GAVAZZI, C. (s.d.). Energy Management Energy Meter Type EM21 72D. CARLO GAVAZZI.
- (Habara, 2017) Habara, T., Mizutani, K., Harada, H. (2017). A Load Balancing Algorithm for Layer 2 Routing based Wi-SUN Systems. IEEE.
- (Jörke, 2017) Jörke, P., Böcker, S., Liedmann, F., Wietfeld, C. (2017). Urban Channel Models for Smart City IoT-Networks Based on Empirical Measurements of LoRa-links at 433 and 868 MHz. European Union.
- (Künzel, 2017) Künzel, G., Cainelli, G. P., Pereira, C. E. (2017). A Weighted Broadcast Routing Algorithm for WirelessHART Networks. 2017 VII Brazilian Symposium on Computing Systems Engineering. IEEE. doi:10.1109/SBESC.2017.32
- (Kuzlu, 2015) Kuzlu, M., Pipattanasomporn, M., Rahman, S. (2015). Review of Communication Technologies for Smart Homes/Building Applications. 2015 IEEE Innovative Smart Grid Technologies Conference (ISGT-ASIA). Bangkok, Thailand: IEEE.
- (Li, 2014) Li, X., Chen, G., Zhaot, B., Liang, X. (2014). A Kind of Intelligent Lighting Control System Using the EnOcean Network. IEEE.
- (Maass, 2017) Maass, A. I., Nešić, D., Postoyan, R., Dower, P. M., Varma, V. S. (2017). Emulation-based stabilisation of networked control systems over WirelessHART. 2017 IEEE 56th Annual Conference on Decision and Control (CDC). Melbourne, Australia: IEEE.
- (Marks, 2017) Marksteiner, S., Jiménez, V. J. E. , Vallant, H. and Zeiner, H. (2017). An Overview of Wireless IoT Protocol Security in the Smart Home Domain. 2017 Joint 13th CTTE and 10th CMI Conference on Internet of Things

-
- Business Models, Users, and Networks (pp. 1-8). Copenhagen: IEEE. doi:10.1109/CTTE.2017.8260940
- (Mochizuki, 2016) Mochizuki, K., Obata, K., Mizutani, K., Harada, H. (2016). Development and Field Experiment of Wide Area Wi-SUN System Based on IEEE 802.15.4g. IEEE.
- (Modbus, 2018) Modbus. (Julho de 2018). Modbus Application Protocol Specification V1.1b3. Obtido de modbus.org: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
- (OSI, 2018) Modelo OSI – Protocolos. (Fevereiro de 2018). Obtido de pplware.sapo.pt: <https://pplware.sapo.pt/tutoriais/networking/redes-sabe-o-que-e-o-modelo-osi/>
- (Monteiro, 2011) Monteiro, E., Boavida, F. (2011). Capítulo 3, Arquitecturas. Em E. B. Monteiro, Engenharia de Redes Informáticas 10ª Edição Atualizada e Aumentada (pp. 63-67). Lisboa: FCA - Editora de Informática,LDA.
- (MySQL, 2018) MySQL. (Maio de 2018). What is MySQL. Obtido de dev.mysql.com: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- (Napiiah , 2018) Napiiah, M. N., Idris, M. Y. I., Ramli, R., Ahmedy, I. (2018). Compression Header Analyzer Intrusion Detection System (CHA - IDS) for 6LoWPAN Communication Protocol. IEEE. doi:10.1109/ACCESS.2018.2798626
- (Obata, 2016) Obata, K., Mizutani, K., Harada, H. (2016). Carrier Frequency Offset Estimation Scheme for IEEE 802.15.4g Based Wide Area Wi-SUN Systems. IEEE ICC 2016 Ad-hoc and Sensor Networking Symposium. IEEE.
- (Okumura, 2017) Okumura, R., Mizutani, K., Harada, H. (2017). A broadcast protocol for IEEE 802.15.4e RIT based Wi-SUN systems. IEEE.
- (Olsson, 2014) Olsson, J., Instruments, T. (2014). 6LoWPAN demystified. Texas Instruments.

-
- (Qiu, 2018) Qiu, Y., Ma, M. (2018). Secure Group Mobility Support for 6LoWPAN Networks. IEEE Internet of Things Journal. doi:10.1109/JIOT.2018.2805696
- (Santaji, 2017) Santaji, S., Kangralkar, S. (2017). Implementation of Integrated Intelligent Power Management System Using Zigbee Protocol. International Conference on Communication and Signal Processing . India: IEEE.
- (Shaham, 2016) Shahamabadi, M. S., Ali, B. M., Noordin, N. K., Rasid, M. F. b. A. (2016). The Challenges of NEMO in 6LoWPAN Smart Building Area. 2016 International Conference on Identification, Information and Knowledge in the Internet of Things. IEEE. doi:10.1109/IKI.2016.110
- (Siemens, 2016) SIEMENS. (2016). SICAM – Power Quality and Measurements. Nuremberg, Germany: Siemens AG.
- (Somani, 2012) Somani, N. A., Patel, Y. (2012). ZIGBEE: A LOW POWER WIRELESS TECHNOLOGY FOR INDUSTRIAL APPLICATIONS. International Journal of Control Theory and Computer Modelling (IJCTCM) Vol.2, No.3.
- (Somkane, 2017) Somkane, P., Kongratana, V., Gulpanich, S., Tipsuwanporn, V., Wongvanich, N. (2017). A Study of Flow-Level Cascade Control with WirelessHART™ Transmitter using LabVIEW. 2017 17th International Conference on Control, Automation and Systems (ICCAS 2017). Ramada Plaza, Jeju, Korea: ICROS.
- (Supram, 2017) Supramaniam, T. , Ibrahim, R., Hassan, S. M., Bingi, K. (2017). Development of WirelessHART Adapter with Industrial Transmitter for Process Monitoring. Proc. of the 2017 IEEE International Conference on Signal and Image Processing Applications (IEEE ICSIPA 2017), Malaysia, September 12-14, 2017. IEEE.
- (Topologias, 2018) Tipos de topologia de rede. (Fevereiro de 2018). Obtido de tudosobrede.blogspot.pt:

<http://tudosobrerede.blogspot.pt/2009/05/tipos-de-topologia-de-rede.html>

- (Usmonov , 2017) Usmonov, M., Gregoretti, F. (2017). Design and Implementation of a LoRa Based Wireless Control for Drip Irrigation Systems. 2017 2nd International Conference on Robotics and Automation Engineering. IEEE.
- (Wang, 2017) Wang, S. Y., Chen, Y. R., Chen, T. Y., Chang, C. H., Cheng, Y. H., Hsu, C. C., Lin, Y. B. (2017). Performance of LoRa-based IoT Applications on Campus. IEEE.
- (Apache2, 2018) What is the Apache HTTP Server Project. (Maio de 2018). Obtido de apache.org: https://httpd.apache.org/ABOUT_APACHE.html
- (Yousheng, 2017) Yousheng, G., Lingyun, Y., Lijing, H. (2017). Addressing scheme based on three-dimensional space over 6LoWPAN for internet of things. 2017 IEEE 13th International Conference on Electronic Measurement & Instruments. IEEE.

