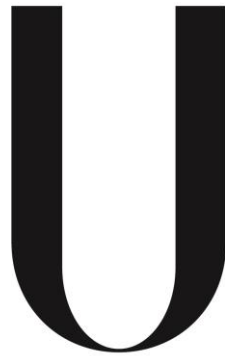


Universidade de Lisboa
Faculdade Ciências
Departamento de Engenharia Geográfica, Geofísica e Energia



LISBOA

UNIVERSIDADE
DE LISBOA

**Desenvolvimento de um sistema de monitorização do consumo e qualidade de
energia elétrica em sistemas trifásicos**

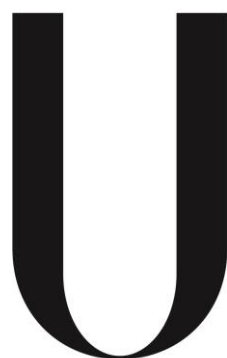
Duarte Sérgio Rodrigues Granado Santos

Dissertação

Mestrado Integrado em Engenharia da Energia e do Ambiente

2014

Universidade de Lisboa
Faculdade Ciências
Departamento de Engenharia Geográfica, Geofísica e Energia



LISBOA

UNIVERSIDADE
DE LISBOA

**Desenvolvimento de um sistema de monitorização do consumo e qualidade de
energia elétrica em sistemas trifásicos**

Duarte Sérgio Rodrigues Granado Santos

Dissertação de Mestrado Integrado em Engenharia da Energia e do Ambiente

Dissertação orientada pelo Professor Doutor Jorge Maia Alves
e coorientada pelo Engenheiro António Soares (FCUL)

2014

Abstract

The market for electric meters and energy monitoring systems is growing and still does not have a standardized technology in place for monitoring energy consumption. Several international and national pilot tests have been taking place with the use of various types of equipment's in order to determine their effects on energy consumption reduction, customer satisfaction and in some cases improving the quality of electricity. These energy reduction equipment's are also seen by many as a possible launching pad for the concept of smart-grid, because the readings and their real-time communication is the first step to generate a smart grid. This thesis has developed a device that meets these concepts, using results from pilot studies to shape its development and purpose. We developed energy motorization system capable of reading data like frequency, voltage, current, power factor, power (active, reactive and apparent) and energy (active and reactive) in single or three phase electrical grid depending on the purpose and manner of use intended . This system is composed of a development kit that has the integrated circuit MAX78630+PPM, Arduino UNO microcontroller, a GSM shield, and an RTC being able to carry out the energy motorization and data transmission every 5 seconds to an online application deployed in the cloud OpenShift. The application stores the data in a data base accessible only by the user, with detailed energy consumption information, and hosts a website that supports FusionCharts graphics for an easy and dynamic presentation of data to any user who has access to the Web Address. The system operates independently and works with a 12 V transformer.

Keywords: electric meters, energy monitoring systems, energy quality, real time communication, integrated circuit, MAX78630+PPM, microcontroller, Arduino UNO, PaaS Cloud, Open Shift, Three-Phase power grid, Single-Phase power grid.

Resumo

O mercado dos sistemas de contagem e monitorização de energia encontra-se em crescimento e não possui ainda uma tecnologia de uso padronizada. Vários testes internacionais e nacionais têm-se vindo a realizar com diversos tipos de equipamento com o objetivo de determinar os seus efeitos na redução de consumo, satisfação do cliente e em alguns casos o melhoramento da qualidade da energia elétrica, sendo que também são visto por muitos como uma possível plataforma de lançamento para o conceito de smart-grid, pois as leituras e sua comunicação em tempo real são o primeiro passo para gerar uma rede inteligente. Nesta dissertação desenvolveu-se um equipamento que vai ao encontro destes conceitos, usando resultados provenientes de estudos piloto para moldar o seu desenvolvimento e propósito. Desenvolveu-se um sistema de monitorização capaz de ler dados de frequência, tensão, corrente, fator de potência, potência (ativa, reativa e aparente) e energia (ativa e reativa) numa rede monofásica ou trifásica dependendo do propósito e forma de utilização pretendida. Este sistema usa na sua arquitetura um kit de desenvolvimento que possui um circuito integrado o MAX78630+PPM, um microcontrolador Arduino UNO, uma placa GSM, e um RTC, sendo capaz de proceder à monitorização e envio dos dados de 5 em 5 segundos para uma aplicação online implementada na nuvem OpenShift. A aplicação armazena os dados numa base de dados acessível ao utilizador, com dados detalhados de consumo, e aloja um website que suporta gráficos fusioncharts para uma apresentação comoda e dinâmica desses dados a qualquer utilizador que possua o seu endereço electrónico. O sistema opera autonomamente e funciona sendo alimentado à rede elétrica através de um transformador de 12 V.

Palavras-chave: contadores de energia, Monitores de energia, qualidade da energia, comunicação em tempo real, circuito integrado, MAX78630+PPM, microcontrolador, Arduino UNO, Nuvem PaaS, OpenShift, trifásico, monofásico.

Índice

1.	INTRODUÇÃO	1
1.1.	MOTIVAÇÃO	1
1.2.	ENQUADRAMENTO DA DISSERTAÇÃO	2
1.3.	PANORAMA NACIONAL	5
1.4.	FIABILIDADE DAS DIFERENTES TECNOLOGIAS DE TELECOMUNICAÇÕES	9
1.5.	TIPO DE LIGAÇÃO E TARIFA APLICADA	10
1.6.	OBJETIVOS E REQUISITOS DA DISSERTAÇÃO	11
2.	INTRODUÇÃO À TEORIA DE CONSUMO TRIFÁSICO	12
2.1.	CORRENTE ALTERNADA	13
2.2.	CIRCUITOS RLC	14
2.3.	POTÊNCIAS INSTANTÂNEA, ATIVA, REATIVA E APARENTE	16
2.3.1.	POTÊNCIA INSTANTÂNEA	16
2.3.2.	POTÊNCIA ATIVA	17
2.3.3.	POTÊNCIA REATIVA	17
2.3.4.	POTÊNCIA APARENTE	18
2.4.	FATOR DE POTÊNCIA	19
2.4.1.	INCONVENIENTES DA POTÊNCIA/ENERGIA REATIVA	19
2.4.2.	COMPENSAÇÃO DO FATOR DE POTÊNCIA	20
2.5.	PROBLEMAS DE QUALIDADE DA ENERGIA:	21
2.6.	SISTEMAS TRIFÁSICOS	21
2.6.1.	SISTEMAS TRIFÁSICOS Vs SISTEMAS MONOFÁSICOS	21
2.6.2.	SISTEMA EQUILBRADO E CONDUTOR NEUTRO	22
2.6.3.	TENSÕES SIMPLES E COMPOSTAS	23
2.6.4.	LIGAÇÃO DE RECEPTORES TRIFÁSICOS – TRIANGULO (Δ) E ESTRELA (Y)	24
2.6.5.	CÁLCULO DE POTÊNCIA DOS SISTEMAS TRIFÁSICOS	25
3.	ESTADO DA ARTE	25
3.1.	CONTADORES ELETROME CÂNICOS	25
3.2.	CONTADORES HÍBRIDOS	26
3.3.	CONTADORES ESTÁTICOS	26
3.3.1.	CONTADORES ELETRÓNICOS	26
3.3.2.	SISTEMAS DE MONITORIZAÇÃO DE ENERGIA	27
3.4.	FONTES DE ALIMENTAÇÃO	27
3.4.1.	BATERIAS	27
3.4.2.	TRANSFORMADOR AC/DC LIGADO À REDE	28
3.5.	SENSORES	28
3.5.1.	DE TENSÃO	28
3.5.2.	DE CORRENTE	28

3.6.	SERVIDOR WEB E WEBPAGE	30
3.6.1.	O COMPUTADOR COMO SERVIDOR – HARDWARE.....	30
3.6.2.	O SERVIDOR WEB COMO PROGRAMA – SOFTWARE.....	31
3.6.3.	COMPUTAÇÃO CLOUD “NUVEM”	31
4.	ARQUITETURA DO SISTEMA	32
4.1.	KIT DE DESENVOLVIMENTO MAX78630+PPM	32
4.1.1.	VISÃO GERAL DO SOC.....	33
4.1.1.1.	ENTRADAS ANALÓGICAS.....	34
4.1.1.1.1.	ENTRADAS DE TENSÃO	34
4.1.1.1.2.	ENTRADAS DE CORRENTE	34
4.1.1.2.	CONVERSOR DELTA SIGMA A/D	35
4.1.1.3.	PROCESSADOR DE MEDIÇÃO DE 24-BITS	35
4.1.1.4.	FLASH E RAM	35
4.1.1.5.	PINOS DIGITAIS DE ENTRADA E SAÍDA (I/O PINS)	35
4.1.1.6.	INTERFACES DE COMUNICAÇÃO	35
4.1.2.	DESCRIÇÃO FUNCIONAL E OPERAÇÃO.....	35
4.1.2.1.	ACESSO DE REGISTOS	36
4.1.2.2.	RÁCIO DA ATUALIZAÇÃO DE DADOS	37
4.1.2.3.	CANAIS DE MEDIÇÃO	37
4.1.2.3.1.	VOLTAGEM E VOLTAGEM RMS	37
4.1.2.3.2.	FREQUÊNCIA DA LINHA	38
4.1.2.3.3.	CORRENTE E CORRENTE RMS.....	38
4.1.2.3.4.	CALCULO DAS POTÊNCIAS.....	39
4.1.2.3.4.1.	POTENCIA ATIVA POR FASE.....	39
4.1.2.3.4.2.	POTENCIA REATIVA POR FASE.....	39
4.1.2.3.4.3.	POTÊNCIA APARENTE POR FASE.....	40
4.1.2.3.4.4.	FATOR DE POTÊNCIA	40
4.1.2.3.5.	POTÊNCIA ATIVA, REATIVA, APARENTE E FATOR DE POTÊNCIA TOTAL	40
4.1.2.3.6.	CÁLCULO DA ENERGIA.....	41
4.1.3.	FONTE CONECTADA EM ESTRELA, CARGA CONECTADA EM ESTRELA (Y-Y).....	42
3.1.1.	COMUNICAÇÃO.....	43
3.1.1.1.	INTERFACE UART.....	43
3.1.1.2.	USB -GUI	46
3.2.	SISTEMA DE AQUISIÇÃO DE DADOS “DATA LOGGER”	48
3.2.1.	ARDUINO UNO R3	49
3.2.1.1.	DESCRIÇÃO FUNCIONAL	49
3.2.1.2.	ARDUINO IDE-(INTEGRATED DEVELOPMENT ENVIRONMENT)	50
3.2.1.2.1.	CÓDIGO E COMPILAÇÃO	51
3.2.1.2.2.	CONFIGURAÇÃO DA COMUNICAÇÃO E BAUDRATE.....	51

3.2.1.2.3.	INTERFACE UART, PROTOCOLO ALTSOFTSERIAL	51
3.2.2.	RTC	52
3.2.3.	PLACA GSM/GPRS	52
3.2.4.	LEDS DE NOTIFICAÇÃO DE ENVIO DE DADOS.....	52
3.3.	SISTEMA DE MONITORIZAÇÃO DE ENERGIA	53
3.3.1.	INTEGRAÇÃO DO MAX78630+PPM NO COLETOR DE DADOS.....	53
3.3.2.	FLUXOGRAMA DO CÓDIGO DE MONITORIZAÇÃO DESENVOLVIDO	55
4.	NUVEM OPENSIFT	55
4.1.	CRIAR CONTA E INSTALAÇÃO DAS FERRAMENTAS DE CLIENTE	56
4.2.	CRIAR UMA APLICAÇÃO	58
4.3.	INSTALAR BASE DE DADOS MYSQL	58
4.4.	PHPMUADMIN E SINCRONIZAÇÃO COM O MONITOR DE ENERGIA	59
4.5.	FORMATO DO WEBSITE, GESTÃO E CONSULTA DOS DADOS.....	60
5.	RESULTADOS EXPERIMENTAIS	63
5.1.	MONTAGEM DA CAIXA PARA O DISPOSITIVO E DOS CABOS DE CONEXÃO PARA MEDIÇÕES TRIFÁSICAS E MONOFÁSICAS... 63	
5.1.1.	CONEXÃO – CARGAS TRIFÁSICAS	64
5.1.2.	CONEXÃO – CARGAS MONOFÁSICAS.....	64
5.2.	TESTE COM CARGA TRIFÁSICA – TORNO MECÂNICO	65
5.2.1.	RESULTADOS DA FREQUÊNCIA E DAS TENSÕES	65
5.2.2.	RESULTADOS DAS CORRENTES	66
5.2.3.	RESULTADOS DAS POTÊNCIAS.....	68
5.2.4.	RESULTADOS DO FATOR DE POTÊNCIA	71
5.2.5.	RESULTADOS DA ENERGIA	72
5.3.	TESTE COM CARGAS MONOFÁSICAS	73
5.3.1.	LEITURA E COMPARAÇÃO DA CORRENTE EFICAZ.....	73
5.3.2.	LEITURA DAS POTÊNCIAS	74
5.3.3.	LEITURA DA ENERGIA	75
6.	CONCLUSÕES	76
	REFERÊNCIAS	78
	ANEXOS.....	80

Índice de Tabelas

TABELA 1.3-1 - SÍNTESE DE CARACTERÍSTICAS E FUNCIONALIDADE DOS CONTADORES DA EDP-D	5
TABELA 1.3-2 – FUNCIONALIDADES EXISTENTES EM CONTADORES INSTALADOS ENTRE 2002 E 2012.	6
TABELA 1.3-3 – PREÇOS MÉDIOS DAS EBS INSTALADAS NO PILOTO DE ÉVORA ATÉ 2011.	7
TABELA 1.3-4 – TABELA DE REPRESENTAÇÃO QUALITATIVA DOS NÍVEIS DE CONSUMO APRESENTADOS NA FIGURA	9
TABELA 1.4-1 – DESEMPENHOS DE VÁRIAS TECNOLOGIAS DE TELECOMUNICAÇÃO DE INFORMAÇÃO EM APLICAÇÕES REMOTAS DE ANÁLISE DE CONSUMO ENERGÉTICO.	9
TABELA 1.5-1 – NA TABELA SÃO CONSIDERADAS E CARACTERIZADAS AS LIGAÇÕES EM BAIXA TENSÃO DIVIDIDA EM BAIXA TENSÃO NORMAL E BAIXA TENSÃO ESPECIAL, MÉDIA TENSÃO E ALTA TENSÃO, P - REPRESENTA A POTÊNCIA CONTRATÁVEL.	10
TABELA 1.5-2 – ESCALÕES DAS POTÊNCIAS CONTRATÁVEIS EM BAIXA TENSÃO NORMAL.	10
TABELA 1.5-3 – TARIFA SELEÇÃO – OPÇÃO 1 DA ENDESA, TARIFAS DE ELETRICIDADE E POTÊNCIA CONTRATADA.	11
TABELA 2.4.1-1 – TABELA DE PREÇOS DE REFERÊNCIA PARA CONSUMO EM BTE PELA EDP DISTRIBUIÇÃO.	20
TABELA 2.5-1 – INTERVALO DE FREQUÊNCIAS ESTIPULADA PARA A QUALIDADE DA REDE.	21
TABELA 2.5-2 – INTERVALO DE TENSÕES ESTIPULADA PARA A QUALIDADE DA REDE.	21
TABELA 4.1.2.1-1 – TABELA DE DESCRIÇÃO GENÉRICA DE UM REGISTO E O SEU TIPO.....	36
TABELA 4.1.2.1-2 – TABELA DEMONSTRATIVA DO TIPO DE REGISTO DESCRITO NO EXEMPLO 1.	36
TABELA 4.1.2.1-3 – TABELA DEMONSTRATIVA DO TIPO DE REGISTO DESCRITO NO EXEMPLO 2.	36
TABELA 4.1.2.3.1-1 – TABELA DE ENDEREÇOS, DESCRIÇÃO E ESCALA TEMPORAL DOS REGISTOS DE CÁLCULO DA TENSÃO.	37
TABELA 4.1.2.3.2-1 – TABELAS DE ENDEREÇOS, DESCRIÇÃO E ESCALA TEMPORAL DOS REGISTOS DA FREQUÊNCIA DA LINHA.....	38
TABELA 4.1.2.3.3-1 – TABELA DE ENDEREÇOS, DESCRIÇÃO E ESCALA TEMPORAL DOS REGISTOS DE CÁLCULO DA CORRENTE.	38
TABELA 4.1.2.3.4-1 – TABELA DE ENDEREÇOS, DESCRIÇÃO E ESCALA TEMPORAL DOS REGISTOS DE CÁLCULO DAS POTÊNCIAS ATIVA, REATIVA, APARENTE E FATOR DE POTÊNCIA.	39
TABELA 4.1.2.3.6-1 – TABELA DE ENDEREÇOS, DESCRIÇÃO E ESCALA TEMPORAL DOS REGISTOS DE CÁLCULO DAS ENERGIAS ATIVA E REATIVA, POSITIVAS E NEGATIVAS.....	41
TABELA 4.1.2.3.6-2 – TABELA DE ENDEREÇOS, DOS REGISTOS VSCALE E IFSCALE PARA O CÁLCULO DOS “PACOTES” DE ENERGIA.	42
TABELA 3.1.1-1 – PINOS DE LIGAÇÃO DAS INTERFASES DE COMUNICAÇÃO INDIVIDUAIS.	43
TABELA 3.1.1.1-1 – TABELA DESCRITIVA DO ENVIO DE UM PACOTE DE BYTES MESTRE	44
TABELA 3.1.1.1-2 – SUMÁRIOS DOS COMANDOS DOS PACOTES MESTRE.	44
TABELA 3.1.1.1-3 – TABELA DESCRITIVA DO ENVIO DE UM PACOTE DE BYTES ESCRAVOS.....	45
TABELA 3.1.1.1-4 – SUMÁRIOS DOS COMANDOS DOS PACOTES ESCRAVO.	45
TABELA 4.5-1 –TABELA DOS INDICADORES DE CONSUMO NA PÁGINA WEB CUSTOS INSTANTÂNEOS.....	63
TABELA 5.2.5-1 – CONSUMO DA ENERGIA ATIVA E REATIVA REGISTADO PELO PROTÓTIPO E PELO PEL 103.....	73

Índice de Figuras

FIGURA 1.2:1: ESTES SÃO OS RESULTADOS MÉDIOS DOS TESTES PILOTO. TAXAS DE SUCESSO, TAL COMO O IMPACTO DO CONSUMO DOMÉSTICO MÉDIO, CONTEÚDO INFORMATIVO, ETC. DENTRO DE UM MERCADO PARTICULAR NÃO SÃO REFLETIDOS NESTE GRÁFICO.	4
FIGURA 1.3:1: DISTRIBUIÇÃO MÉDIA DO NÚMERO DE CONTADORES INSTALADOS PELA EDP-D NOS SEUS RESPECTIVOS PERÍODOS DE TEMPO.	6
FIGURA 1.3:2: DEMONSTRA A EVOLUÇÃO DE CONTADORES INSTALADOS AO LONGO DOS ÚLTIMOS ANOS PELA EEM E EDA, APRESENTANDO O NÚMERO DE CONTADORES INSTALADOS EM FUNÇÃO DA IDADE (DATA DE INSTALAÇÃO) (KEMA, 2012).	6
FIGURA 1.3:3: DIVISÃO DO CONSUMO DOMÉSTICO EM GRUPOS DE CONSUMO DISTINTOS, DO PROJETO SMARTMETER 250 RESIDÊNCIAS.	8
FIGURA 1.3:4: GRÁFICO DE REPRESENTAÇÃO DOS NÍVEIS DE CONSUMO ESTUDADOS NO PROJETO SMARTMETER 250 RESIDÊNCIAS. ...	8
FIGURA 2:1: PRODUÇÃO DE TRÊS F.E.MS. POR MEIO DE UM ALTERNADOR TRIFÁSICO.	12
FIGURA 2:2: VETORES TENSÃO NUM SISTEMA TRIFÁSICO.	13
FIGURA 2.1:1: REPRESENTAÇÃO GRÁFICA DA EQUAÇÃO (2.3).	13
FIGURA 2.1:2: REPRESENTAÇÃO GRÁFICA DE CORRENTE AC EM FASE COM A TENSÃO, COM RESPECTIVOS VALORES EFICAZES INDICADOS.	14
FIGURA 2.2:1: CIRCUITO RLC SÉRIE	15
FIGURA 2.2:2: VETORES TENSÃO E CORRENTES RLC SERIE.	15
FIGURA 2.2:3: CIRCUITO RLC PARALELO.	15
FIGURA 2.2:4: VETORES TENSÃO E CORRENTES RLC PARALELO.	16
FIGURA 2.3.1:1: POTENCIA, TENSÃO E CORRENTE NUMA RESISTÊNCIA PURA.	17
FIGURA 2.3.3:1: POTÊNCIA TENSÃO E CORRENTE NUMA INDUTÂNCIA.	17
FIGURA 2.3.4:1: TRIÂNGULO DE POTÊNCIAS.	18
FIGURA 2.3.4:2: ENERGIA POR QUADRANTE.	18
FIGURA 2.4.2:1: COMPENSAÇÃO DO FATOR DE POTÊNCIA.	20
FIGURA 2.6.2:1: ALIMENTAÇÃO INDEPENDENTE DE TRÊS RECETORES IDÊNTICOS.	22
FIGURA 2.6.2:2: SISTEMA EQUILIBRADO DE CARGAS COM NEUTRO (CORRENTE NO NEUTRO É NULA).	22
FIGURA 2.6.2:3: TRANSPORTE DE ENERGIA ELÉTRICA TRIFÁSICA POR MEIO DE QUATRO CONDUTORES.	22
FIGURA 2.6.3:1: TENSÕES SIMPLES E COMPOSTAS, COMPOSIÇÃO EM ESTRELA.	23
FIGURA 2.6.3:2: REPRESENTAÇÃO VETORIAL DAS TENSÕES SIMPLES E COMPOSTAS.	23
FIGURA 2.6.4:1: LIGAÇÃO DE RECETORES EM TRIÂNGULO.	24
FIGURA 3.3.1:1: DIAGRAMA DE BLOCOS DE UM CONTADOR ELETRÔNICO.	26
FIGURA 3.4.1:1: CURVA DE DESCARGA DAS PILHAS ENERGIZER E95 D.	27
FIGURA 3.5.2:1: MEDIÇÃO DE CORRENTE MEDIANTE UMA RESISTÊNCIA DE SHUNT	28
FIGURA 3.5.2:2: SÍMBOLO DE UM TRANSFORMADOR DE CORRENTE (CT).	29
FIGURA 3.5.2:3: MEDIÇÃO DE CORRENTE MEDIANTE UMA BOBINA DE ROGOWSKI.	30
FIGURA 4.1:1: DIAGRAMA TÍPICO DE APLICAÇÃO DO KIT DE DESENVOLVIMENTO MAX78630+PPM.	33
FIGURA 4.1.1:1: DIAGRAMA DE BLOCOS DOS RECURSOS DISPONÍVEIS NO SoC MAX78630+PPM.	33
FIGURA 4.1.1.1.1:1: CONDICIONAMENTO DO SINAL DE TENSÃO, POR MEIO DE UM CIRCUITO DIVISOR DE RESISTÊNCIA.	34
FIGURA 4.1.1.1.2:1: CONDICIONAMENTO DO SINAL DE TENSÃO, PARA O CT CR8449-2500-N	34

FIGURA 4.1.2.3.1:1: DIAGRAMA DE BLOCOS COM OS PASSOS PARA O CÁLCULO DAS TENSÕES RMS (EFICAZES).	38
FIGURA 4.1.2.3.3:1: DIAGRAMA DE BLOCOS COM OS PASSOS PARA O CÁLCULO DAS CORRENTES RMS (EFICAZES).	38
FIGURA 4.1.2.3.4.1:1: DIAGRAMA DE BLOCOS COM OS PASSOS PARA O CÁLCULO DAS POTÊNCIAS ATIVAS.	39
FIGURA 4.1.2.3.4.2:1: DIAGRAMA DE BLOCOS COM OS PASSOS PARA O CÁLCULO DAS POTÊNCIAS REATIVAS.	40
FIGURA 4.1.2.3.4.3:1: DIAGRAMA DE BLOCOS COM OS PASSOS PARA O CÁLCULO DAS POTÊNCIAS APARENTES.	40
FIGURA 4.1.3:1: CONEXÕES DO KIT DE DESENVOLVIMENTO MAX78630+PPM PARA MEDIÇÃO DE CARGA COM TOPOLOGIA (Y-Y).42	
FIGURA 3.1.1:1: PINOS DE LIGAÇÃO DAS INTERFASES DE COMUNICAÇÃO PRESENTES NO KIT DE DESENVOLVIMENTO.	43
FIGURA 3.1.1.2:1: CONEXÃO À GUI VIA PORTA COM E RESPECTIVO BAUD RATE.	46
FIGURA 3.1.1.2:2: APRESENTA OS SEPARADORES DISPONÍVEIS NA INTERFASE GUI	46
FIGURA 3.1.1.2:3: IMAGEM DA INTERAÇÃO GRÁFICA DO SEPARADOR SCALING.	46
FIGURA 3.1.1.2:4: IMAGEM DA INTERAÇÃO GRÁFICA DO SEPARADOR DIRECT, COM UM EXEMPLO DE CÁLCULO DE UM VALOR DE TENSÃO.	47
FIGURA 3.1.1.2:5: RESULTADOS DOS FATORES DE ESCALA PARA O CÁLCULO DOS VALORES REAIS PARA CADA TIPO DE REGISTO.	47
FIGURA 3.1.1.2:6: IMAGEM DA INTERAÇÃO GRÁFICA DO SEPARADOR CALIBRATION, COM <i>LINE LOCK</i> DESCATIVADO E O PACOTE DE ENERGIA “BUCKET” DEFINIDO PARA 1 WH.	48
FIGURA 3.2:1: SISTEMA COLETOR DE DADOS (<i>DATA-LOGGER</i>) DO MAX78630+PPM	48
FIGURA 3.2.1:1: ESQUEMA DO ARDUINO UNO R3.	49
FIGURA 3.2.4:1: ESQUEMA DO CIRCUITO DE ATIVAÇÃO DOS LEDS DE AVISO DO ENVIO DE DADOS GSM/GPRS	52
FIGURA 3.3.1:1: LOCALIZAÇÃO DOS PINOS DE COMUNICAÇÃO UART (Tx, Rx) NO KIT DE DESENVOLVIMENTO DO MAX78630+PPM.	53
FIGURA 3.3.1:2: LOCALIZAÇÃO DOS PINOS DE ALIMENTAÇÃO DO KIT DE DESENVOLVIMENTO MAX78630+PPM EM DC.	53
FIGURA 3.3.1:3:ESQUEMA DO CIRCUITO DO SISTEMA DE MONITORIZAÇÃO DE ENERGIA DESENVOLVIDO NESTA DISSERTAÇÃO.	54
FIGURA 3.3.2:1: FLUXOGRAMA DO CÓDIGO USADO NA PROGRAMAÇÃO DO ARDUINO UNO R3	55
FIGURA 4.1:1: PLATAFORMA <i>ONLINE</i> PARA REALIZAR O REGISTO NA NUVEM OPENSIFT.	56
FIGURA 4.1:2: USO DA LINHA DE COMANDO PARA ACEDER À CHAVE DE ACESSO SSH USANDO O <i>EMAIL</i> E <i>PASSWORD</i> DO REGISTO <i>ONLINE</i>	57
FIGURA 4.2:1:LINHA DE COMANDOS GERANDO O DOMÍNIO DA APLICAÇÃO E A FINALIZAR A CRIAÇÃO DA APLICAÇÃO.	58
FIGURA 4.3:1: USO DA OPENSIFT WEB CONSOLE COMO MEIO DE CRIAR O CARTUCHO MYAQL 5.5	58
FIGURA 4.3:2: USO DO CARTUCHO PHPMYADMIN PARA ACEDER À BASE DE ADMINISTRAÇÃO DE DADOS.	59
FIGURA 4.4:1: FLUXOGRAMA DA ESTRUTURA DA INTERFACE DE COMUNICAÇÃO EQUIPAMENTO/UTILIZADOR DESENVOLVIDO.	60
FIGURA 4.5:1: PARTE DA JANELA DA PÁGINA INTRODUTÓRIA AO WEBSITE DE MONITORIZAÇÃO, COM OS SEPARADORES À VISTA DO UTILIZADOR.	60
FIGURA 4.5:2: PARTE DA JANELA DO SEPARADOR “CONSUMO ELÉTRICO”, COM OS GRÁFICOS DA POTÊNCIA EM TEMPO REAL.	61
FIGURA 4.5:3 PARTE DA JANELA DO SEPARADOR “ENERGIA”, COM OS GRÁFICOS DO CONSUMO DE ENERGIA DE 15 EM 15 MINUTOS.	61
FIGURA 4.5:4: PARTE DA JANELA DO SEPARADOR “QUALIDADE”, ONDE NOS GRÁFICOS O VERDE REPRESENTA BOA, O LARANJA ACEITÁVEL E O VERMELHO MÁ QUALIDADE.	61
FIGURA 4.5:5: GRAFICOS DO SEPARADOR “CUSTOS E IMPACTOS”	62
FIGURA 4.5:6: PARTE DA JANELA DO SEPARADOR “CUSTOS INSTANTÂNIOS”, ONDE O CONSUMIDOR PODE VER OS CUSTOS EM TEMPO REAL.	62
FIGURA 5.1:1: MONTAGEM DOS CABOS ELÉTRICOS NA FICHA TRIFÁSICA USANDO O CÓDIGO DE CORES STANDARD.	63

FIGURA 5.1.2: PROTÓTIPO DO DISPOSITIVO DE MONITORIZAÇÃO DESENVOLVIDO NA SUA FORMA FINAL.	64
FIGURA 5.1.1:1: SISTEMA DE CONEXÃO PARA A MEDIÇÃO DE SISTEMAS OU CARGAS TRIFÁSICAS	64
FIGURA 5.1.2:1: SISTEMA DE CONEXÃO PARA A MEDIÇÃO DE DISPOSITIVOS MONOFÁSICOS.	64
FIGURA 5.2:1: TORNO MECÂNICO QUE SERÁ A CARGA PARA A REALIZAÇÃO DESTES TESTES.	65
FIGURA 5.2:2: MONTAGEM DAS FERRAMENTAS PARA O INÍCIO DO TESTE DE COMPARAÇÃO DOS INSTRUMENTOS DE MEDIÇÃO	65
FIGURA 5.2.1:1: COMPARAÇÃO DAS LEITURAS DA FREQUÊNCIA ENTRE O PROTÓTIPO E O PEL 103	65
FIGURA 5.2.1:2: COMPARAÇÃO DE VÁRIAS LEITURAS DE TENSÃO LIDAS PELO PEL 103 E PELO PROTÓTIPO.	66
FIGURA 5.2.2:1 COMPARAÇÃO ENTRE OS VALORES DAS CORRENTES LIDAS PELO PROTÓTIPO E PELO PEL NAS FASES A E B.....	66
FIGURA 5.2.2:2: COMPARAÇÃO ENTRE O VALOR DA CORRENTE LIDA PELO PROTÓTIPO E PELO PEL NA FASE C.....	67
FIGURA 5.2.2:3 COMPARAÇÃO DA MÉDIA DAS FASES DE CORRENTES LIDAS PELO PROTÓTIPO E PELO PEL 103	67
FIGURA 5.2.3:1:VALORES DE POTÊNCIA ATIVA EM CADA FASE LIDOS PELO PROTÓTIPO E PELO PEL 103.	68
FIGURA 5.2.3:2: APRESENTAÇÃO DA POTÊNCIA TOTAL E DA MÉDIA DAS FASES LIDAS PELO PROTÓTIPO E PELO PEL 103.....	68
FIGURA 5.2.3:3: APRESENTAÇÃO DAS LEITURAS DAS POTÊNCIAS REATIVAS, EM CADA FASE, FEITAS PELO PROTÓTIPO E PELO PEL 103.	69
FIGURA 5.2.3:4: COMPARAÇÃO ENTRE AS LEITURAS DE POTÊNCIA REATIVA REALIZADAS NO PROTÓTIPO E REALIZADAS NO PEL 103.	69
FIGURA 5.2.3:5: LEITURA DOS VALORES DE POTÊNCIAS REATIVA MÉDIAS E SOMA DAS FASES LIDAS PELO PROTÓTIPO E PELO PEL 103.	69
FIGURA 5.2.3:6: APRESENTAÇÃO DAS LEITURAS DAS POTÊNCIAS APARENTES, EM CADA FASE, FEITAS PELO PROTÓTIPO E PELO PEL 103.	70
FIGURA 5.2.3:7: COMPARAÇÃO ENTRE AS LEITURAS DE POTÊNCIA APARENTE REALIZADAS NO PROTÓTIPO E REALIZADAS NO PEL 103.	70
FIGURA 5.2.3:8: LEITURA DOS VALORES DE POTÊNCIAS APARENTES MÉDIAS E SOMA DAS FASES LIDAS PELO PROTÓTIPO E PELO PEL 103.	71
FIGURA 5.2.4:1: VALORES DOS FATORES DE POTÊNCIAS DAS DIVERSAS FASES MEDIDAS PELO PROTÓTIPO E PELO PEL 103.	71
FIGURA 5.2.4:2: VALORES MÉDIOS DOS FATORES DE POTÊNCIA OBTIDOS ATRAVÉS DO PROTÓTIPO E DO PEL 103.	72
FIGURA 5.2.5:1: EVOLUÇÃO DO CONSUMO DE ENERGIA ATIVA E REATIVA NO DECORRER DO TESTE, NUM INTERVALO DE 15 MINUTOS.	72
FIGURA 5.3:1: ESTAÇÃO DE SOLDAR.....	73
FIGURA 5.3:2: VENTONHA	73
FIGURA 5.3.1:1: COMPARAÇÃO DO CONSUMO DAS CORRENTES DE CARGAS DE TESTE MONOFÁSICAS.	73
FIGURA 5.3.1:2: MEDIÇÕES ÀS CARGAS MONOFÁSICAS COM RECURSO A UMA PINÇA AMPERIMÉTRICA PARA COMPARAÇÃO DE RESULTADOS.....	74
FIGURA 5.3.2:1: DADOS DAS POTÊNCIAS ATIVA E REATIVA NO TESTE DAS CARGAS MONOFÁSICAS.	74
FIGURA 5.3.2:2: DADOS DAS POTÊNCIAS APARENTES NO TESTE DAS CARGAS MONOFÁSICAS.	75
FIGURA 5.3.2:3 DADOS DOS FATORES DE POTÊNCIA NO TESTE DAS CARGAS MONOFÁSICAS.	75
FIGURA 5.3.3:1: DADOS DA ENERGIA CONSUMIDA NO TESTE DAS CARGAS MONOFÁSICAS.	75

Abreviaturas e/ou Símbolos

AC/CA – *Alternating Current/Corrente Alternada*
ADC – *Analog to Digital Converter*
API – *Application Programming Interface*
BTE – *Baixa Tensão Especial*
BTN – *Baixa Tensão Normal*
CT – *Current Transformer*
DC/CD – *Direct Current/Corrente Direta*
DIOs – *Digital Input/Outputs*
EB – *Energy Boxes*
EDA – *Eletricidade dos Açores*
EDP-D – *EDP Distribuição*
EEM – *Empresa de Eletricidade da Madeira*
ERSE – *Entidade Reguladora dos Serviços Energéticos*
f.e.m – *Força Eletromotriz*
GUI – *Graphical User interface*
HTML – *HyperText Markup Language*
IDE – *Integrated Development Environment*
I/O – *Input/Outputs*
IEC/CEI – *International Electrotechnical Commission/ Comissão Eletrotécnica Internacional*
LAN – *local area network*
LSB – *Least significant Byte*
ORD's – *Operadoras da Rede de Distribuição*
PWM – *Pulse Width Modulation*
QEE – *Qualidade da Energia Elétrica*
RAM – *Random Access Memory*
SRAM – *Static Random Access Memory*
RHC – *Red Hat Customer*
SoC – *System on chip*
USB – *Universal Serial Bus*

Agradecimentos

O caminho para a realização desta dissertação teria sido muito mais difícil se não tivesse contado com a colaboração e apoio de inúmeras pessoas, as quais através dos seus conhecimentos, disponibilidade e atenção dada, tornaram possível a concretização dos objetivos propostos. Assim, quero deixar aqui o meu agradecimento a essas pessoas, não só por me ajudarem a completar esta dissertação, mas por também terem contribuído para a minha aprendizagem.

Agradeço ao coorientador, Engenheiro António Soares, pela enorme atenção, confiança e apoio demonstrado ao longo do decorrer de todo o trabalho, bem como pela disponibilidade prestada na discussão e resolução dos problemas decorrentes, orientação e revisão do trabalho escrito.

Agradeço ao meu orientador, Doutor Jorge Maia Alves, pelo interessante tema de dissertação proposto, e a todo o apoio prestado não só durante a dissertação mas durante todo o tempo que foi também meu coordenador de curso e professor.

Agradeço aos serviços técnicos da Faculdade de Ciências, por me terem auxiliado com algumas ferramentas importantes para os testes que realizei no âmbito da dissertação.

Agradeço à colega de curso, Francisca Fernandes, por ter contribuído para a concretização de alguns passos da dissertação, dada a nossa colaboração no decorrer da dissertação de ambos.

Agradeço ao meu amigo de longa data, João Pires, que graças aos seus conhecimentos de programação me auxiliou na construção do Website desenvolvido para a dissertação.

Agradeço a todos os meus amigos que estando presentes foram e são muitos importantes para o meu bem-estar, equilíbrio emocional e confiança de que conseguirei alcançar os meus objetivos.

Por fim agradeço o apoio incondicional dos meus pais, que me apoiaram estando sempre presentes ao longo de todo o meu percurso académico e profissional.

1. Introdução

Os sistemas de monitorização de energia o objetivo de medir o consumo elétrico de uma dada instalação ou carga, as quais estão habitualmente adaptadas a sistemas de consumo monofásico ou trifásico.

Estes sistemas têm vindo a sofrer um aumento da sua comercialização tem resultado também numa evolução constante dos antigos contadores eletromecânicos para os sistemas de aquisição de dados analógico-digitais, em que estes últimos possibilitam uma aquisição de dados mais elaborada e “inteligente”.

Esta evolução é uma consequência do aumento do número de conversores eletrónicos de potência utilizados, sobretudo na indústria, mas também pelos consumidores em geral. Em resultado disso é possível observar na rede elétrica uma crescente deterioração das formas de onda de corrente e tensão dos sistemas de potência. Os prejuízos económicos resultantes deste e de outros problemas associados aos sistemas elétricos são muito elevados, e por isso a questão da qualidade da energia elétrica entregue aos consumidores finais é hoje, mais do que nunca, objeto de grande preocupação. Segundo um relatório do EPRI (*Electric Power Research Institute*) os problemas relacionados com a qualidade da energia e quebras no fornecimento de energia custam à economia dos Estados Unidos mais de 119 mil milhões de dólares por ano (Martins, J.S., Couto, C., Afonso, J.L., 2003).

Nesse contexto a aplicação de uma abordagem de baixo custo pretende-se desenvolver um monitor de energia inteligente capaz de medir consumo trifásico, com capacidade para analisar a quantidade e qualidade do consumo de energia elétrica.

A qualidade de energia elétrica (QEE) envolve todas as partes do sector elétrico, desde o produtor de energia até aos consumidores e fabricantes de equipamentos, bem como entidades governamentais e de regulação. Uma fraca QEE pode resultar num aumento de custos e perigo para os equipamentos elétricos, sendo que a sua monitorização traz vantagens óbvias tanto para o mercado doméstico e serviços como também industrial e comercial (ISR-EDP, 2005).

A gestão da energia consumida é uma tarefa que engloba várias medidas, tais como o planeamento, a monitorização e a implementação de estratégias de controlo optimizadas. Geralmente o custo inicial da implementação destas medidas é relativamente baixo, principalmente se comparado com as possíveis melhorias de economias energéticas, produtividade e qualidade dos produtos. De facto, devido ao seu domínio transversal, os sistemas modernos de controlo de processo não são concebidos exclusivamente para a eficiência energética, mas também para a optimização da produção, da qualidade do produto e da segurança dos operadores dos equipamentos produtivos (Martins, et al., 2003).

1.1. Motivação

Neste momento decorrem inúmeras medidas e empreendimentos que impelem o desenvolvimento de sistemas de monitorização inteligentes, por forma a ajudar o consumo de energia mais eficiente bem como controlar a qualidade das redes elétricas.

Para combater o aumento da “poluição” eletromagnética, organizações como a CEI – Comissão Eletrotécnica Internacional (IEC – *International Electrotechnical Commission*) e o IEEE – Instituto dos Engenheiros Eletrotécnicos e Eletrónicos - têm elaborado normas visando limitar os conteúdos harmónico nos sistemas elétricos (Martins, et al., 2003).

A Diretiva 2009/72/CE (eletricidade) estabelece a obrigação dos Estados-Membros avaliarem a implementação de sistemas de contadores inteligentes de eletricidade. As diretivas anteriormente mencionadas foram transpostas para a legislação nacional através da aprovação dos decretos-lei n.º 78/2011 (eletricidade) ambos de 20 de Junho (Portaria n.º 231/2013, 2013).

No âmbito desta diretiva foram realizados estudos como “Empower Demand” encomendado pela ESMIG – European Smart Metering Industry Group – à VaasaETT e “Relatório 1E/G” encomendado pela ERSE (Entidade reguladora dos serviços Energéticos) à KEMA estão envolvidos no chamado “terceiro pacote energético¹” e têm o propósito de estudar o seguinte (Stromback, et al., 2011; KEMA, 2012):

- A avaliação económica de longo prazo de todos os custos e benefícios para o mercado, designadamente para operadores de rede, comercializadores e consumidores;
- O modelo de sistema inteligente economicamente mais racional e o prazo para a sua instalação;
- A importância do sistema de monitorização inteligente como base para produzir o melhor método em termos de custo/eficácia para tornar o consumidor uma peça mais envolvida.

A diretiva antevê desenvolvimento total o mais tardar até 2022, com 80% de consumidores equipados com sistemas de monitorização inteligente até 2020 (Portaria n.º 231/2013, 2013).

Reconhecendo esta situação, a ERSE tem procurado que a regulamentação do setor dinamize ações que contribuam para a promoção da eficiência energética no consumo. Em particular, no Regulamento Tarifário do setor elétrico estabelece-se um mecanismo competitivo de promoção de ações de gestão da procura, a implementar pelos comercializadores, operadores de redes e associações e entidades de promoção e defesa dos interesses dos consumidores, designado por Plano de Promoção da Eficiência no Consumo de energia elétrica. Através do qual promove-se, pela atribuição de incentivos mediante um procedimento concursal que se tem vindo a realizar desde o ano 2007, a execução de medidas que visam melhorar a eficiência no consumo de energia elétrica (PPEC, 2014).

Nesta dissertação pretende-se desenvolver o protótipo de um sistema de monitorização adaptado quer a consumos domésticos quer a consumos comerciais/industriais, sendo que o controlo da qualidade da energia é particularmente interessante do ponto de vista comercial/industrial, pois necessitam de mais consumos por parte de cargas não-lineares como, motores, transformadores e outros equipamentos similares, os quais consomem outro tipo de energia denominada de energia reativa. Um dos motivos de interesse está em controlar-se o impacto dessas cargas (denominadas indutivas) pois possuem impactos negativos tanto para o consumidor como para o produtor de energia.

Pretende-se assim produzir um sistema de monitorização de energia de baixo custo e elevada precisão capaz de medir o consumo e a qualidade de energia em instalações trifásicas. A aquisição e tratamento dos dados deverá ser feita por forma a tirar proveito das conclusões dos estudos acima mencionados e assim criar-se uma ferramenta aperfeiçoada para o aumento da eficiência do consumo de energia elétrica.

1.2. Enquadramento da dissertação

Pretende-se que as funcionalidades do sistema de monitorização desenvolvidos nesta dissertação se enquadrem dentro das tecnologias e metodologia que maiores impactos produzem nos consumidores. Assim, foram considerados os resultados dos estudos já mencionados, “**Empower Demand**” e “**Relatório 1E/G**” como guias para determinar que aspetos mais relevantes que o sistema deverá possuir.

O estudo “**Empower Demand**”, realizado a pedido da ESMIG, tem como propósito perceber o potencial e as limitações de um leque de programas de Dynamic Pricing e Feedback através do uso de monitores

¹ O “Third Energy Package” entrou em vigor a 9 de Setembro de 2009 e consiste em duas diretivas e três regulações que intencionam completar o processo de liberalização e integração do mercado energético da UE, bem como fortalecer e proteger os direitos do consumidor. O pacote inclui diretivas para o gás e eletricidade que requerem os estados membros da UE ‘assegurar a implementação de sistemas de monitorização inteligentes.

de consumo elétricos inteligentes. Para tal a VaasaETT recorre à comparação de 100 testes piloto por todo o mundo (Stromback, Et al., 2011).

Os programas *Dynamic Pricing* encorajam consumidores a desviarem os seus consumos das horas de pico de consumo para as horas de mais baixo consumo, permitindo reduzir os custos de distribuição, produção e fornecimento de energia. Isto é conseguido através de mecanismos de atribuição de tarifas dinâmicos que refletem melhor os custos de fornecimento de eletricidade. Os preços são elevados nos períodos de pico de consumo e diminuídos nos restantes períodos dependendo da procura e produção de energia (Stromback, Et al., 2011). Este programa não será mais abordado uma vez que não contribui para os objetivos do sistema de monitorização desenvolvido nesta dissertação.

Os Programas *Feedback* servem para tornar a energia visível e tornar o consumo de energia visível, aguçando a curiosidade do consumidor sobre como e quanta energia é gasta.

Nestes casos de estudo cada caso individual foi categorizado em um dos seguintes tipos de programa: Faturação Discriminada, Monitores Domésticos, Páginas Web, Monitores Ambiente, e uma mistura dos vários tipos de programas² (Stromback, Et al., 2011).

- Faturação Discriminada: Exemplo de apresentação de informação indireta e não em tempo real, sendo que por isso não refletem o consumo real de um dado mês. A diferença entre o consumo estimado médio e o uso real é feito no final do período de faturação ou quando um consumidor troca de distribuidor de eletricidade. A Faturação Discriminada irá cobrar o consumo real efetuado e providir um histórico do consumo do mês em questão com o mês anterior ou com o ano anterior no mesmo período. A fatura irá também providir informação sobre o quanto a respetiva habitação consumiu em comparação com habitações com características semelhantes. Ao contrário de faturas tradicionais onde as habitações recebem a sua fatura 4-6 vezes por ano, a Faturação Discriminada pode ser enviada até cerca de 1 vez por mês.
- Monitores Domésticos: Monitores que se encontram diretamente visíveis na habitação e que fornecem informação quase em tempo real sobre o consumo elétrico da habitação. Estes monitores providenciam às habitações uma informação e histórico em tempo real do consumo e custo da eletricidade. Outro Tipo de Informação pode ser adicionado aos monitores.
- Páginas Web: Oferecem uma forma alternativa de fornecer ao consumidor informação sobre o seu consumo elétrico. As Páginas Web são escolhidas como meio de fornecer informação porque são relativamente baratas. Estas dependem de sistemas de monitoração de eletricidade para coletar os dados necessários de consumo. Os espaçamentos entre cada medição de consumo instantâneo depende maioritariamente da forma como cada monitor lê a potência consumida ou qual a frequência com que a informação é transferida para o comercializador (ou retalhista).
- Monitores Ambiente: Diferem dos Monitores Domésticos na forma como apresentam a informação sobre o consumo elétrico sendo que eles não apresentam dados de consumo específico, mas sim sinais de aviso ao consumidor dos seus níveis de consumo e/ou de uma mudança nos preços de eletricidade. Muitos Monitores Ambiente tem o atributo de serem atrativos e intuitivos que traz ao consumidor potencial de aceitação.

² Os programas de combinação produziram impactos positivos e com melhores resultados que o programa individual.

A informação apresentada é categorizada em 8 formas: Comparação-Entre-Pares, Preço da eletricidade, Comparação de Histórico, Desagregação do consumo, Nível de consumo atualizado, Custo até à data (fatura), Poupanças relativas a períodos anteriores, Ambiente (Emissões de CO₂).³

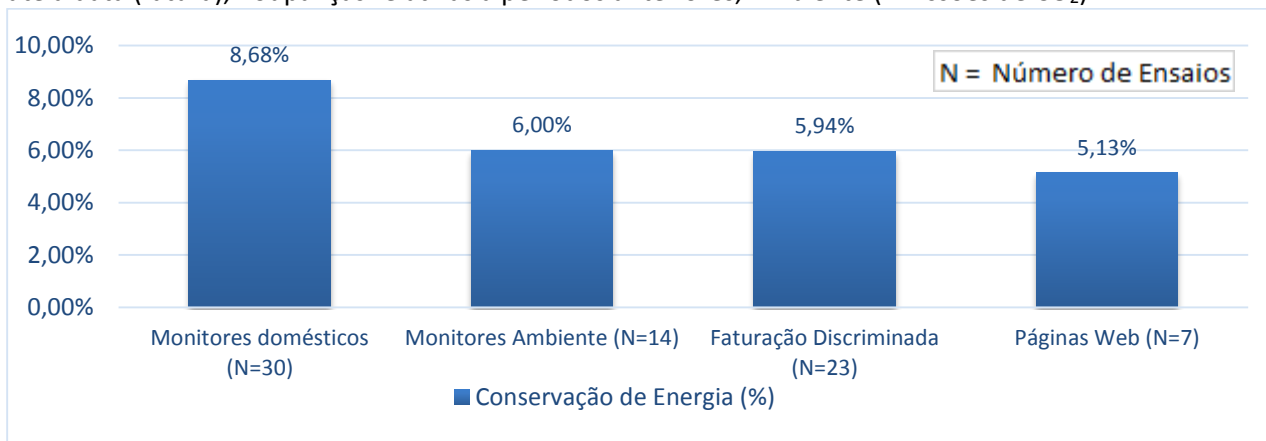


Figura 1.2:1: Estes são os resultados médios dos testes piloto. Taxas de sucesso, tal como o impacto do consumo doméstico médio, conteúdo informativo, etc. dentro de um mercado particular não são refletidos neste gráfico.

Assim os resultados principais do estudo “Empower Demand” demonstram que os consumidores reagem aos programas de Dynamic Pricing e Feedback positivamente, os resultados dos programas mantêm-se além dos 2-3 anos, sendo que os questionários realizados após os estudos piloto revelam que em média 75 – 90% dos participantes ficaram satisfeitos com o teste em que tomaram parte. Isso dito, os resultados variam largamente dentro de cada programa⁴.

Já o “**Relatório 1E/G**” concentra-se em resultados obtidos através de estudos do mercado português e estudos piloto realizados em Portugal, sendo que é possível observar neste a evolução dos monitores inteligentes em Portugal bem como as opiniões das ORD’s (Operadores da Rede de Distribuição), consumidores e outras conclusões (KEMA, 2012).

As evidências recolhidas face à alteração do comportamento dos consumidores face à disponibilização de uma melhor informação sobre o consumo das suas instalações são as seguintes:

- No Segmento Doméstico (BTN - Baixa Tensão Normal):

Após cerca de um ano de estudo e de dados, os resultados apontam para uma redução de 3,9% do consumo de eletricidade (com um intervalo de confiança de 95%, os resultados são uma redução entre 1,8% e 6%, ou seja, 2,1% de margem de erro) (KEMA, Relatório 1E/G., 2012).

- No Segmento Empresarial (BTE - Baixa Tensão Especial):

Verifica-se uma maior sensibilidade para a gestão energética. Também mostram satisfação com a iniciativa e com o feedback que conseguem obter através da informação *online* (KEMA, Relatório 1E/G., 2012).

As variações (reduções) já analisadas no caso do projeto Inovgrid⁵ variam entre valores na casa dos 2%, chegando a situações menos frequentes de diminuições superiores aos 20%.

³ Ver Glossário de Termos, anexo A.

⁴ Canadá foi o que obteve mais sucesso nos testes com monitores domésticos com 12% de reduções; Europa foi o segundo com 10% e os EUA o terceiro com 7%. A faturação discriminada na Europa foi 4.5 vezes mais efetivo que nos estados Unidos.

⁵ O InovGrid é um projeto da EDP-D que dota a rede elétrica de informação e equipamentos inteligentes capazes de automatizar a gestão da energia, melhorando assim a qualidade do serviço, diminuição de custos, e a eficiência energética, aumentando a sustentabilidade ambiental.

Das conclusões obtidas com clientes BTE, a EDP-D (EDP-Distribuição) realça os seguintes aspetos (KEMA, 2012):

- O acesso a informações detalhadas sobre o consumo de energia é relevante na identificação dos hábitos de consumo das instalações;
- Verifica-se que a representação dos diagramas de carga é uma forma fácil e compreensível de identificar os padrões de consumo e identificar oportunidades de melhoria
- O feedback direto obtido, disponibiliza os meios para monitorizar e avaliar os resultados das alterações introduzidas no consumo de energia;
- O acesso *online* também fornece informação útil para promover e partilhar as melhores práticas em matéria de medidas de economia de energia e de comportamentos;
- Alguns clientes têm demonstrado interesse em obter o mesmo detalhe de informação disponibilizado para outras instalações/locais de consumo;
- Quase todos os clientes que interagiram pessoalmente com a EDP - Distribuição classificam a abordagem como muito positiva;

Em termos globais, a EDP-D refere que, apesar de estudos internacionais apontarem para uma redução de consumo da ordem dos 5% a 9%, uma perspetiva mais conservadora e resultante de estudos de maior duração e com universos de contadores maiores, aponta para reduções de 3% a 6% (KEMA, 2012).

1.3. Panorama Nacional

Um levantamento da evolução dos equipamentos de monitorização em Portugal, até à data deste relatório, revela que a EDP-D tem 6.121.250, a EEM (Empresa de Eletricidade da Madeira) conta com 140.312 e a EDA (Eletricidade dos Açores) 119.946 contadores instalados em contratos BTN. Este número inclui contadores eletromecânicos, contadores estáticos e contadores híbridos. A distribuição pelos diferentes tipos de contadores é a seguinte (KEMA, 2012):

	<i>EDP-D</i>	<i>EEM</i>	<i>EDA</i>
• <i>Contadores eletromecânicos:</i>	4.418.969	75.264	82.965
• <i>Contadores híbridos:</i>	334.680		
• <i>Contadores estáticos:</i>	1.367.601	65.048	36.981

Os contadores instalados têm no geral as seguintes características e funcionalidades⁶

Tipo	Características	Funcionalidades
Eletromecânicos	Medição de energia ativa Classe de precisão 2	
Híbridos	Medição de energia ativa Classe de precisão 2	Multi-tarifa Relógio Integrado
Estáticos	Medição de energia ativa e ativa/reactiva Classe de precisão MID-A ou MIDB Display Comunicação sem fios ou Porta Série	Multi-Tarifa Relógio Integrado Registos históricos Potência Máxima Eventos

Tabela 1.3-1 - Síntese de características e funcionalidade dos contadores da EDP-D

⁶ Classe B – EN-50470 (Classe 1 (IEC-62053-21)) ou Classe A – EN-50470 (Classe 2 (IEC-62053-21)) são normas de precisão na medição de monitores de consumo energético, de acordo com a Diretiva Europeia MID.

O parque de contadores instalados da EDP-D é aproximadamente 70% monofásico e 30% trifásico, dos EEM 87% são monofásicos e 13% trifásicos e da EDA 84% são monofásicos e 16% trifásicos (KEMA, 2012).

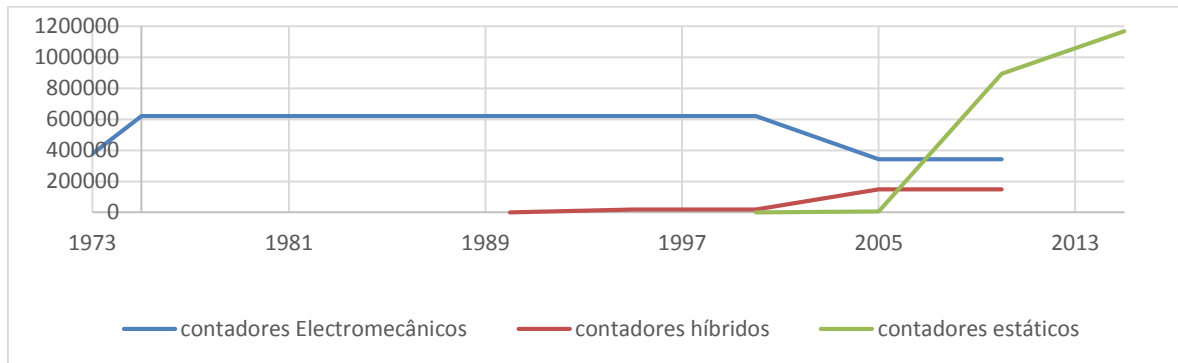


Figura 1.3:1: Distribuição média do número de contadores instalados pela EDP-D nos seus respectivos períodos de tempo.

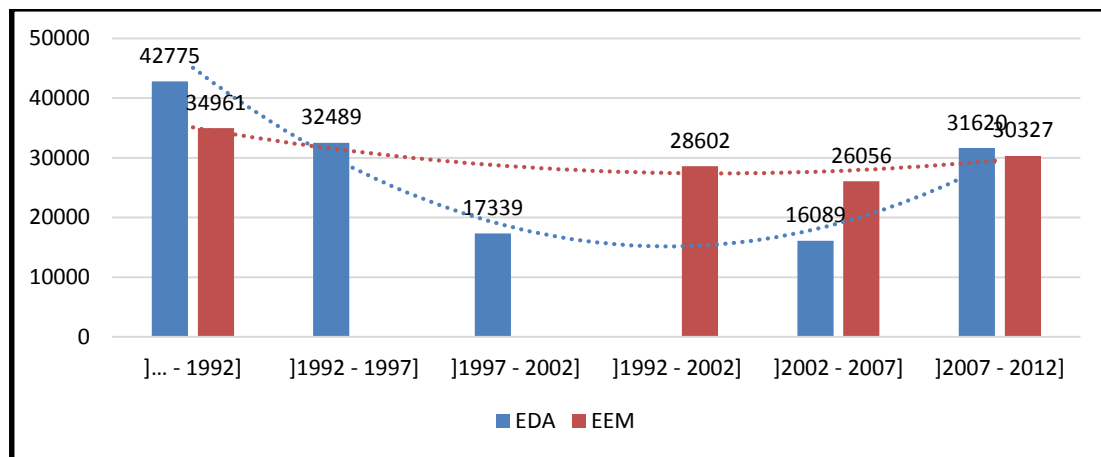


Figura 1.3:2: demonstra a evolução de contadores instalados ao longo dos últimos anos pela EEM e EDA, apresentando o número de contadores instalados em função da idade (data de instalação) (KEMA, 2012).

Todos os contadores eletromecânicos da EEM têm idade superior a 17 anos desde a data de instalação, sendo que os instalados mais recentemente são contadores estáticos multi-tarifa. Na EDA existem contadores eletromecânicos instalados até 2007, ano a partir do qual apenas foram instalados contadores estáticos. Cerca de 80% dos contadores estáticos (25% do parque total atualmente instalado) foram instalados nos últimos 5 anos até ao ano de 2012 (KEMA, 2012).

As seguintes funcionalidades surgem já em contadores instalados nos últimos 5 a 10 anos:	
• Registo de energia a 4 quadrantes	• Registo de eventos
• Registo de erros de sistema	• Perfil de carga
• Registo Fator de Potência	• Registo Ponta
• Monitorização de Tensão (V), Corrente (I), Potência (P)	• Ângulos fase
• Monitorização valores instantâneos de Tensão (V), Corrente (I), Frequência (f)	

Tabela 1.3-2 – Funcionalidades existentes em contadores instalados entre 2002 e 2012.

➤ **Relação entre custos e funcionalidades dos contadores (evidências recolhidas).**

- a) Informação disponível sobre custos de transmissão e processamento da informação associada.

No caso da EDP-D cada cliente possui uma contagem que permite a discriminação do consumo a cada período de 15 minutos. O custo para a transmissão de dados entre um contador inteligente e o sistema central, usando tecnologia GPRS, é atualmente de 1€/mês/cliente, com um limite de 25MB/mês. Este limite é suficiente para a quantidade de dados a transmitir diariamente (registos associados aos fechos, diagramas de carga com períodos de 15 minutos, eventos e outras operações pontuais). De acordo com a EDP-D, o custo para a transmissão de dados entre um concentrador⁷ e o sistema central, usando tecnologia GPRS, estima-se em 10€/mês/concentrador, correspondendo a 0,10€/mês/cliente se considerarmos 100 clientes por concentrador (KEMA, 2012).

- b) Custo médio dos contadores inteligentes.

As EBs- (Energy boxe's) instaladas em Évora foram produzidas no âmbito do ACE-InovGrid. Abaixo indicam-se os preços médios das EBs que foram instaladas no projeto piloto de Évora até 2011:

	PLC PRIME	GPRS
Monofásica	75€ - 85€	110€ - 115€
Trifásica	120€ - 125€	150€ - 160€

Tabela 1.3-3 – Preços médios das EBs instaladas no piloto de Évora até 2011.

Uma recente consulta ao mercado efetuada pela EDP Distribuição permite antecipar que o valor médio de EBs para 2012 seja mais baixo do que os valores apresentados na tabela acima, e que resultaram do desenvolvimento de protótipos específicos para o projeto piloto InovGrid (KEMA, 2012). Não é revelada a causa desta descida de preço, contudo é mencionada a redução dos custos de comunicação.

➤ **O projeto Smartmeter 250 residências**

Trata-se de um projeto intitulado “Contadores Inteligentes Para Decisões Eficientes” aprovado pelo PPEC, que visa a instalação de 250 Sistemas de monitorização inteligentes em residências, cujo objetivo é a redução dos consumos elétricos por análise dos perfis de consumo. A Metodologia usada consistiu em (Águas, Miguel, 2013):

- Ausência de custos de instalação de equipamentos de medição para o consumidor envolvido;
- Informar graficamente da evolução dos consumos;
- Criar um espírito de competição através da posição num ranking de um *website*.

O projeto usa como o equipamento o Cloogy® Home, produto desenvolvido pela empresa ISA, que se pode adquirir neste momento por 239.00 €, sendo que este inclui o seguinte (Águas, Miguel, 2013):

- 1 Sensor de corrente (permite medir um consumo monofásico, para consumos trifásico será necessário adquirir mais 2 sensores por 9.90 € Cada);
- 1 Transmissor (dispositivo que envia os dados de leitura dos sensores de corrente);
- 1 Concentrador;
- 1 Tomada Inteligente (permite ler o consumo individual uma carga ou várias ligadas a esta, cada tomada inteligente extra terá um custo adicional de 99.00 €);
- 1 Monitor Doméstico.

⁷ Dispositivo que recebe os dados dos sensores e os redireciona para o *modem* e monitor doméstico.

Sendo este um equipamento versátil, ele é direcionado para o cálculo de consumos de Potencia Ativa em tarifas BTN, não sendo capaz de fazer uma avaliação da qualidade da energia.

Este estudo, graças a uma avaliação de cargas de consumo individuais conseguiu chegar a uma conclusão interessante, dividindo o consumo doméstico nos seguintes tipos de grupos, como:

- Essenciais: equipamentos que necessitam estar sempre ligados, habitualmente apenas o frigorífico;
- Standby: equipamento que costumam estar ligados continuamente em consumo mínimo ou residual devido a descuido ou preguiça do utilizador (Computadores, Televisores, modem, Box da Tv, entre outros);
- Variáveis: equipamentos que ligamos apenas quando necessitamos (lâmpadas, máquina de lavar roupa, ferro de engomar, entre outros).

Verificou-se que na média dos 250 utilizadores esse consumo se distribuía como mostra a seguinte figura (Águas, Miguel, 2013):

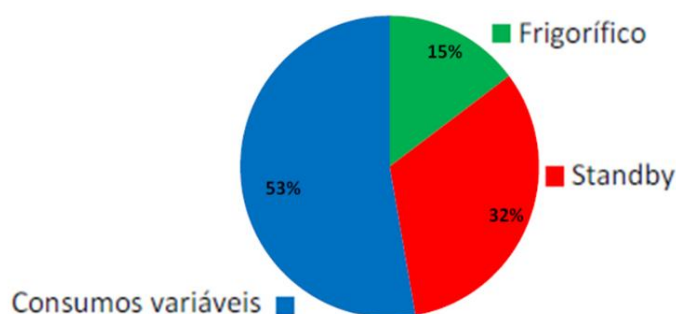


Figura 1.3:3: Divisão do consumo doméstico em grupos de consumo distintos, do projeto SmartMeter 250 residências.

Graças à compreensão da importância do frigorífico e dos standby na redução do consumo, este estudo revelou que só o consumo médio dos standby corresponde a 153€ da fatura elétrica anual deste grupo de teste (considerado o preço da eletricidade, só energia, de 0,17 €/kWh).

É possível então reduzir a fatura elétrica anual em grande parte eliminando o consumo desnecessário de equipamentos em standby. É possível concluir que também se pode reduzir bastante consumo trocando o frigorífico antigo por um mais atual e eficiente, uma vez que este estudo revela também diferenças de consumo de frigoríficos entre os 5 e os 50 % face ao total do consumo doméstico, o que se deve em parte à baixa eficiência destes.

Outro resultado interessante do estudo foi a divisão do grupo em 4 níveis de consumo, isto em grande parte devido a um estilo de vida mais consumidor por parte de alguns dos participantes do projeto.

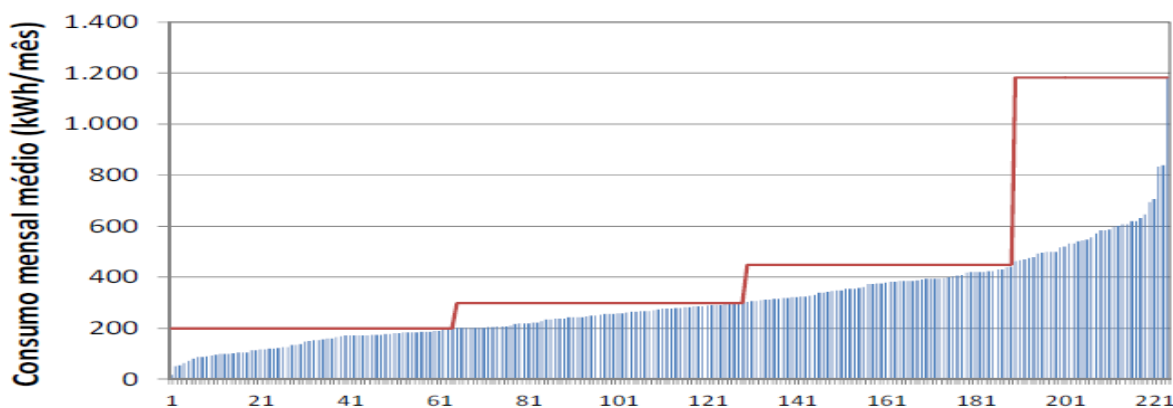


Figura 1.3:4: Gráfico de representação dos níveis de consumo estudados no projeto SmartMeter 250 residências.

Com base nestes dados estabeleceram-se indicadores de nível de consumo para um panorama nacional que podem ser interpretados de forma analítica, como indica a tabela seguinte:

Consumo [C- kWh/mês]	Nível de consumo
C<200	Baixo
200<C<300	Médio
300<C<450	Alto
450<C<1200	Muito Alto

Tabela 1.3-4 – tabela de representação qualitativa dos níveis de consumo apresentados na Figura

No final deste Estudo foi alcançada uma redução média no standby de 10%. Representando uma poupança de 20 €/ano (Águas, Miguel, 2013).

1.4. Fiabilidade das diferentes tecnologias de telecomunicações

Os principais projetos-piloto em BTN desenvolvidos pela EDP-D utilizaram as tecnologias PLC-DCSK, GPRS, RF Mesh e PLC PRIME⁸ para o acesso remoto ao contador inteligente. Face à experiência adquirida e aos dados recolhidos, as principais conclusões quanto à fiabilidade, desempenho e maturidade são as seguintes (KEMA, 2012):

PLC-DCSK			GPRS		
Fiabilidade	Desempenho	Maturidade	Fiabilidade	Desempenho	Maturidade
Aceitável	Insuficiente	Razoável	Boa	Bom	Boa
Resumo			Resumo		
Taxa de transmissão insuficiente para serviços exigentes ou para redes de grande dimensão (ex: recolha de diagramas de carga para todos os clientes); Tecnologia proprietária e com custos baixos, contudo necessita recorrer a outras tecnologias (ex: GPRS/3G/fibra) para comunicações entre o DTC/concentrador e os sistemas centrais.			Local e posição da antena é um fator a considerar para uma boa comunicação; Tecnologia Madura, aberta e normalizada; Custo determinado pelos operadores públicos destas redes e dependendo do seu tempo de vida previsto; Prevê-se que esta tecnologia se mantenha disponível durante bastante tempo.		

RF Mesh			PLC PRIME		
Fiabilidade	Desempenho	Maturidade	Fiabilidade	Desempenho	Maturidade
Boa	Bom	Razoável	Boa	Bom	Iniciante
Resumo			Resumo		
Local e posição da antena é um fator a considerar para uma boa comunicação; A tecnologia testada, embora mais específica e proprietária; Comunicações no segmento LAN (EB -> concentrador) apenas com os custos de gestão global da rede de comunicações; Necessidade de recorrer a outra tecnologia (ex: GPRS/3G/fibra) para comunicações entre o concentrador e os sistemas centrais.			Comunicações no segmento LAN-(redes locais de espaço limitado) (EB -> concentrador) apenas com os custos de gestão global da rede de comunicações; Comunicações no segmento LAN (EB -> concentrador) apenas com os custos de gestão global da rede de comunicações; Decorre um processo de proposta de normalização do PLC Prime, liderado pela Prime Alliance;		

Tabela 1.4-1 – Desempenhos de várias tecnologias de telecomunicação de informação em aplicações remotas de análise de consumo energético.

⁸ Ver Anexo A – Glossário, mas explicação mais detalhada das tecnologias.

1.5. Tipo de Ligação e tarifa aplicada

➤ Tipo de Ligação

Para fornecer ou receber energia elétrica torna-se necessário fazer uma ligação à rede elétrica de distribuição pública em baixa, média ou alta tensão.

Para tal, dever-se-á escolher a ligação mais adequada do ponto de vista técnico-económico, e solicitar ao distribuidor de energia um orçamento para a sua execução (EDP-D – Manual de ligações à rede elétrica do serviço público):

- **Baixa Tensão (BT)** – destina-se a clientes residenciais, lojas, escritórios e pequenas empresas, alimentadas ao nível de tensão 230/400 V.

- **BTN:** Para potências contratadas iguais ou inferiores a 41,4 kVA e uma potência mínima contratada de 1,15 kVA.
- **BTE:** Para potências contratadas superiores a 41,4 kW.

- **Média Tensão (MT)** – destina-se a empresas tipicamente com potências instaladas superiores a 100 kVA, alimentadas ao nível de tensão de 10 kV, 15 kV ou 30 kV, conforme a sua localização geográfica.

- **Alta Tensão (AT)** – destina-se a empresas tipicamente com potências instaladas superiores a 10 MVA, alimentadas ao nível de tensão de 60 kV.

É da responsabilidade do requisitante a escolha do nível de tensão, devendo ter-se em atenção os critérios técnicos e económicos, nomeadamente relacionados com as tarifas de consumo, investimento inicial, custos de manutenção e principalmente o nível da qualidade de serviço pretendida (EDP-D – Manual de ligações à rede elétrica do serviço público).

Baixa Tensão		Média Tensão	Alta Tensão
BTN	BTE	MT	AT
BT ≤ 1 kV		1 kV < MT ≤ 45 kV	45 kV < AT ≤ 110 kV
P ≤ 41,4kVA	P > 41,4 kW	P ≤ 10 MVA	P > 10 MVA

Tabela 1.5-1 – Na tabela são consideradas e caracterizadas as ligações em baixa tensão dividida em baixa tensão normal e baixa tensão especial, média tensão e alta tensão, P - representa a potência contratável.

Potencias Normalizadas

Em BTN as potências a requisitar ou a contratar até 41,4 kVA são normalizadas por escalões, conforme a Tabela abaixo.

Monofásico		Trifásico	
P(kVA)	I _n (A)	P(kVA)	I _n (A)
1,15	5	6,9	3 x 10
2,3	10	10,35	3 x 15
3,45	15	13,8	3 x 20
4,6	20	17,25	3 x 25
5,75	25	20,7	3 x 30
6,9	30	27,6	3 x 40
10,35	45	34,5	3 x 50
13,8	60	41,4	3 x 60

Tabela 1.5-2 – Escalões das potências contratáveis em Baixa Tensão Normal.

Em BTE não existem escalões definidos, podendo ser contratada ou requisitada qualquer valor de potência, acima de 41,4 kW (EDP-D – Manual de ligações à rede elétrica do serviço público).

➤ **Tarifa Aplicada**

O sistema de monitorização desta dissertação usa sensores de corrente com medição até 50 A, o que observando a Tabela 1-5 indica que a potência a contratar indicada será a de 34,5 kVA em regime BTN.

Observando uma tabela de preços de referência da ERSE atualizada (ERSE, 2014. Preços de Referência no Mercado Liberalizado de Energia) opta-se pela tarifa que se considera mais apropriada para esta dissertação, a TARIFA SELEÇÃO – OPÇÃO 1 da ENDESA.

Pot. Contratada (kVA)	Simples		Bi-Horário			Tri-Horário			
	Potência (€/dia)	Energia (€/kWh)	Potência (€/dia)	Energia (€/kWh)		Potência (€/dia)	Energia (€/kWh)		
				Fora Vazio	Vazio		Ponta	Cheias	Vazio
20,7									
27,6						1,0501			
34,5						1,3076	0,1620	0,1620	0,1620
41,4						1,5650			

Tabela 1.5-3 – TARIFA SELEÇÃO – OPÇÃO 1 da ENDESA, tarifas de eletricidade e potência contratada.

Os pré requisitos para a seleção desta tarifa são:

- Tarifa horária única, apropriada para perfil de consumo constante;
- Um Desconto de 40% no preço da energia em relação a uma tarifa de referência;
- Valor de potência contratada (€/Dia) de valor baixo.

Nota: Devido a esta tarifa ser de BTN e não BTE, os valores consumidos de Energia Reativa não serão tidos em consideração na fatura energética do consumidor (ERSE, 2014. Tarifas transitórias de Venda a Clientes).

1.6. Objetivos e requisitos da dissertação

Pretende-se desenvolver um protótipo de um sistema de monitorização de consumo e qualidade de energia elétrica em meio doméstico e industrial, usando sensores de corrente e de tensão, para recolha de dados, um EMP – (Energy Measurement Processor) o MAX78630+PPM para processamento de dados de consumo e qualidade de energia em tempo real em sistemas polifásicos, um Arduino UNO R3 para recolha e decifração dos dados, um RTC para obter as horas e calendário e uma placa GSM/GPRS para envio da informação para uma base de dados alojada num servidor para um servidor *online*. Pretende-se também desenvolver, alojado nesse servidor e ligado à base de dados, uma interface para o utilizador comum poder aceder aos dados de forma simples e instantânea.

Ou seja, que o protótipo desenvolvido vá de encontro com os resultados preconizados pelos estudos nacional e internacional “Empower Demand” e “Relatório 1E/G”, quer para consumos BTN e BTE.

Os requisitos mínimos para se conseguir desenvolver o protótipo são:

- Compressão das funcionalidades do MAX78630+PPM Evaluation Kit;
- Explorar as funcionalidades oferecidos pelo MAX78630+PPM Evaluation Kit com recurso à interface GUI (Graphical User interface) fornecida pelo fabricante;
- Programar o Arduino UNO para comunicar com o MAX78630+PPM;
- Desenvolver um programa de Arduino para comunicar e interpretar valores do MAX78630+PPM como a Frequência Composta, Voltagem, Corrente, Potência Ativa, Potência Reativa, Potência Aparente, Fator de Potência, Energia Ativa Consumida, Energia Ativa Produzida, Energia Reativa Indutiva e a Energia Reativa Capacitiva em todas as fases de consumo de uma rede/carga trifásica.
- Ser capaz de ler as variáveis mencionadas através da porta Série do Arduino.

- Uso de um RTC para controlar a leitura das variáveis indicadas do MAX78630+PPM pelo Arduino UNO, sendo estas feitas de 15 em 15 minutos no caso das variáveis de Energia e de 5 em 5 segundos para as restantes variáveis.
- Uso de uma shield de Arduino GSM/GPRS para enviar as variáveis mencionadas para uma base de dados MySQL, gerida por PHPmyadmin e alojada num Servidor online tipo PaaS⁹ OpenShift, onde MySQL e PHPmyadmin são subsistemas de software deste.
- Programação de código php importado para um dos alojamentos na conta OpenShift criada de forma gratuita por forma a criar um website de apresentação de informação em tempo real dos consumos e qualidade de energia elétrica medidos pelo MAX78630+PPM. Este código PHP tem inserido gráficos FusionCharts que estão contidos em bibliotecas API (Application Programming Interface) de JavaScript.
- Montagem de 2 tipos diferentes de conectores para leituras de cargas. Um para ler cargas trifásicas individuais e outro para ler o consumo individual até 3 cargas e respetivas qualidades energéticas.

2. Introdução à teoria de consumo trifásico

A energia elétrica, sendo utilizada de múltiplas maneiras, pode apresentar-se nos circuitos em diferentes formas, sendo as formas de corrente elétrica mais utilizadas as seguintes:

- Corrente contínua constante - conhecida por corrente contínua (CC, em Português, ou DC em Inglês).
- Corrente descontínua periódica sinusoidal - conhecida por corrente alternada (CA, em Português, ou AC em Inglês).

Foi com o início da comercialização da eletricidade que se iniciou a questão da opção entre corrente contínua (CC) e corrente alternada (CA). A partir de 1882, a CA foi adotada para o transporte e distribuição de energia elétrica em larga escala, pelas seguintes razões (Stromback, J., Dromacque, C., Yassin, M.H., 2011; ISR-EDP, 2005):

- A elevação e a redução de tensão são mais simples.
- Os alternadores (geradores de CA) são mais simples e têm melhor rendimento que os dínamos (geradores de CC).
- Os motores de CA, particularmente os motores de indução são mais simples e têm melhor rendimento que os motores de CC.
- A CA pode transformar-se facilmente em CC por intermédio de sistemas retificadores.

A maior parte dos alternadores geram tensões trifásicas, isto é, tem três bobinas idênticas e independentes, dispostas simetricamente no estator, formando ângulos de 120° entre si (ISR-EDP 2005):

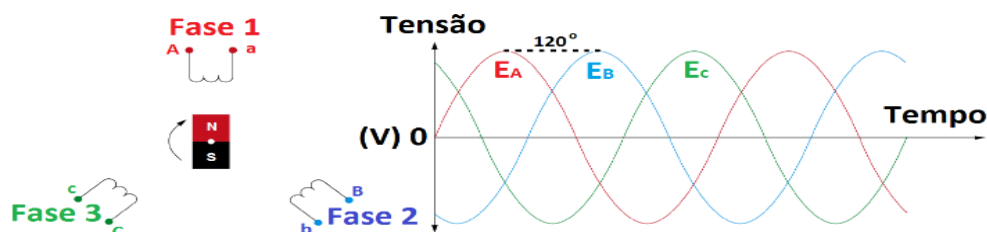


Figura 2:1: Produção de três f.e.ms. por meio de um alternador trifásico.

⁹ Servidor tipo plataforma: é uma categoria de serviços de computação nuvem que providencia uma plataforma e outros subsistemas de software (solution stack's) como serviço

Quando o rotor gira, induz-se em cada bobina uma f.e.m. (força eletromotriz) alternada sinusoidal. Estas f.e.m. têm igual amplitude máxima e estão desfasadas de 120° umas das outras, ou seja, de $1/3$ de período, e podem ser representadas pelas seguintes equações:

$$\begin{aligned} e_1 &= E_A \cdot \sin(\omega \cdot t) \\ e_2 &= E_B \cdot \sin(\omega \cdot t - 120^\circ) \\ e_3 &= E_C \cdot \sin(\omega \cdot t - 240^\circ) \end{aligned} \quad (2.1)$$

Assim, este alternador designa-se por **Alternador Trifásico**, dado que produz três tensões alternadas com fases diferentes. Tal como na corrente alternada monofásica, estas grandezas temporais podem representar-se vectorialmente:

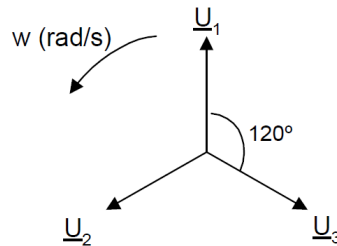


Figura 2.2: Vetores tensão num sistema trifásico.

Na maioria dos casos, os sistemas trifásicos são simétricos (equilibrados) e possuem impedâncias semelhantes em cada uma das suas fases. Cada fase pode ser tratada da mesma forma que um sistema monofásico. Assim, tensão, corrente e potência podem ser determinadas usando as mesmas equações e métodos que em circuitos monofásicos. Já os circuitos trifásicos assimétricos (desequilibrados) possuem condições especiais e a sua análise é mais complexa (Lab-Volt Ltd, 2010).

2.1. Corrente Alternada

O valor instantâneo de uma grandeza alternada sinusoidal u , pode representar-se matematicamente em função do tempo através da seguinte equação:

$$u(t) = U_m \times \sin(\omega \cdot t) \quad (2.2)$$

onde ω representa a velocidade angular (velocidade de rotação do alternador que gera a energia elétrica alternada sinusoidal) que representa-se em radianos por segundo (rad/s). A relação entre a velocidade angular, a frequência e o período é a seguinte:

$$\omega = 2\pi \times f = \frac{2\pi}{T} \quad (2.3)$$

Se considerarmos um vetor \mathbf{U} , de comprimento U_m , girando à velocidade ω , o valor instantâneo u será a projeção vertical desse vetor:

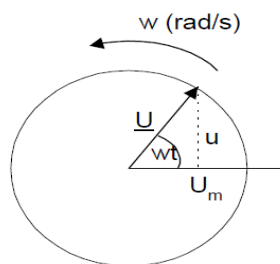


Figura 2.1.1: representação gráfica da equação (2.3)

Dado que a CA se repete periodicamente (ciclicamente), uma das características fundamentais é o valor do intervalo de tempo entre repetições (ou ciclos), ou seja, o **período** - **T**, cuja unidade é o **segundo** - **s**.

É comum utilizar-se uma outra característica da CA, diretamente relacionada com o período - a **frequência** - **f**.

$$f = \frac{1}{T} \quad (2.4)$$

A **Amplitude Máxima (Um)**, também designada por valor máximo ou valor de pico, a amplitude máxima é o valor instantâneo mais elevado atingido pela grandeza (tensão, corrente, f.e.m., etc.).

No caso de grandezas alternadas sinusoidais, o valor eficaz é $\sqrt{2}$ vezes menor que o valor máximo, independentemente da frequência (Fig. 2.1.2):

$$I = \frac{I_m}{\sqrt{2}} \approx 0.7 \times I_m \text{ e } U = \frac{U_m}{\sqrt{2}} \approx 0.7 \times U_m \quad (2.5)$$

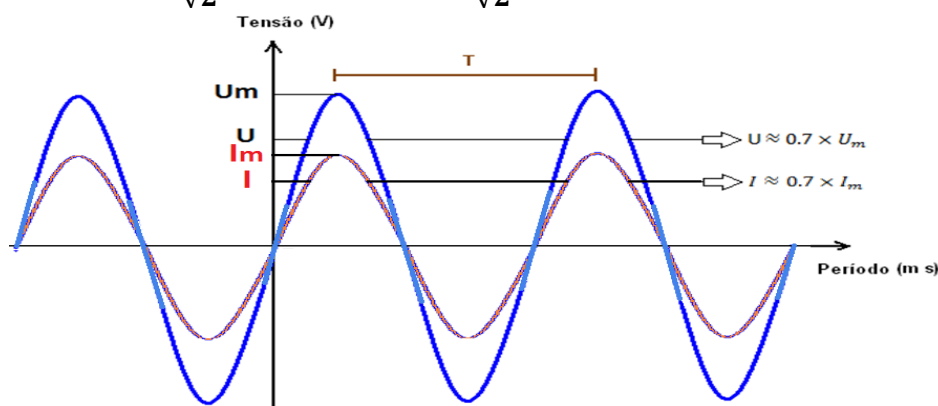


Figura 2.1.2: Representação gráfica de corrente AC em fase com a tensão, com respetivos valores eficazes indicados.

Note-se que:

- A relação de $\sqrt{2}$ entre o valor máximo e o valor eficaz só se verifica para ondas sinusoidais. Para outras formas de onda, a relação é diferente.
- O valor Eficaz da corrente e da Tensão também podem ser identificados como o valor RMS (root mean square) dos mesmos.
- O valor indicado pelos voltímetros e amperímetros, quando se efetuam medidas em AC, é o valor eficaz.
- Quando é referido um dado valor de uma tensão ou corrente alternada, este será sempre um valor eficaz, salvo se outro for explicitamente mencionado (Ferreira Alves, Mário, 1999).

2.2. Circuitos RLC

A análise de circuitos em corrente alternada (CA) implica o estudo do comportamento de três elementos elétricos básicos: resistência (R), indutância (L – bobina) e capacidade (C – condensador).

Na prática, todos os circuitos têm estes elementos. Embora alguns dos respetivos valores possam ser muito pequenos em relação aos outros e portanto desprezáveis. De facto, há sempre fenómenos indutivos e capacitivos inerentes a um circuito, ainda que possam ser pouco intensos (por exemplo, o problema dos parâmetros distribuídos em qualquer linha de transporte de energia elétrica).

➤ Em Série:

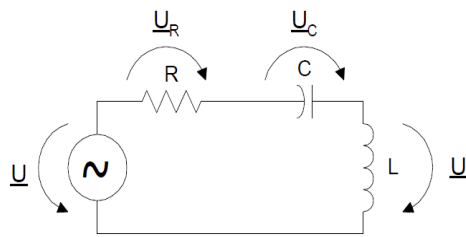


Figura 2.2:1: Circuito RLC série

A resistência R poderá incluir a resistência de outros elementos, como por exemplo a da bobina.

Pela Lei das Malhas: $U = U_R + U_C + U_L$

Distinguindo três situações diferentes:

$U_L > U_C$ ($X_L > X_C$)

Circuito Indutivo

$U_L < U_C$ ($X_L < X_C$)

Circuito Capacitivo

$U_L = U_C$ ($X_L = X_C$)

Circuito em Ressonância

Em termos vetoriais:

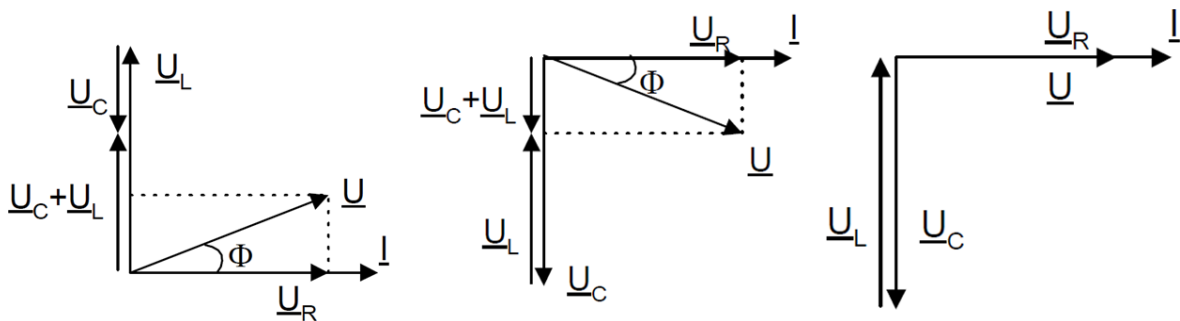


Figura 2.2:2: Vetores tensão e correntes RLC serie.

É observável que as tensões na capacitância e na indutância se anulam mutuamente. Esta situação (de ressonância) deve ser evitada, pois podem produzir-se sobretensões elevadas perigosas para pessoas e instalações (danificando isolamentos das maquinas elétricas, por exemplo).

➤ Em Paralelo:

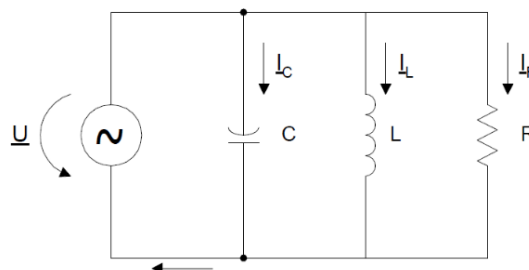


Figura 2.2:3: Circuito RLC paralelo.

Considerando, neste caso, que o circuito é ideal.

Pela Lei dos Nós: $I = I_R + I_C + I_L$

Tal como no circuito RLC série, distinguem-se três casos particulares:

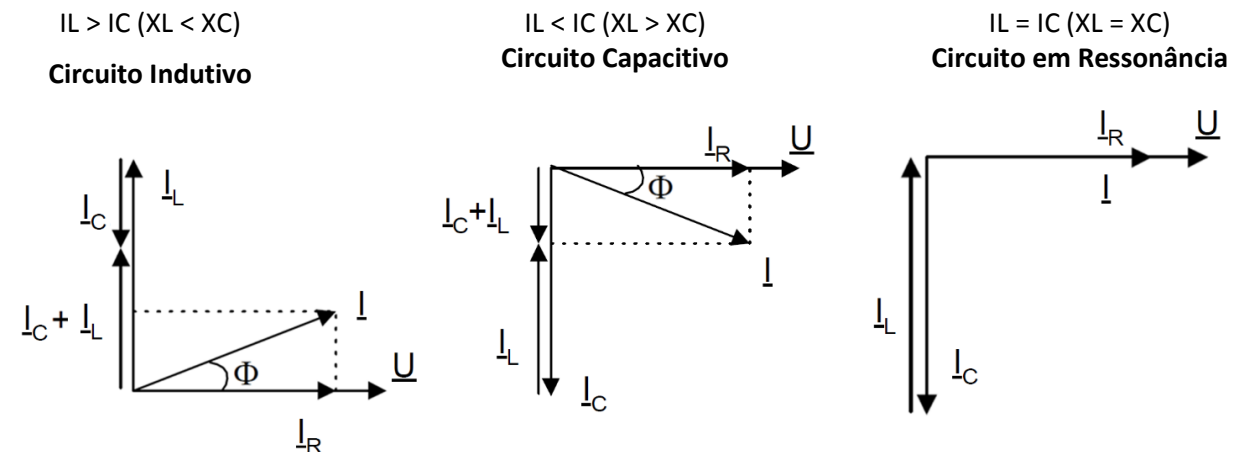


Figura 2.2:4: vetores tensão e correntes RLC paralelo.

Analogamente ao que acontecia com as tensões no circuito RLC série em ressonância, aqui são as correntes na capacitância e na indutância que se anulam mutuamente. Enquanto no circuito RLC série poderiam aparecer sobretensões, no circuito RLC paralelo são as correntes que podem ser demasiado elevadas (Ferreira Alves, Mário, 1999)

A impedância total dos circuitos pode ser calculada pela seguinte equação:

$$Z = \frac{U}{I} \quad (2.6)$$

Ao passo que a frequência de ressonância ($X_L = X_C$) - f_r que pode causar o perigo de sobretensão e sobrecorrente na rede, pode ser calculada pela seguinte equação:

$$X_L = X_C \leftrightarrow 2\pi f_r L = \frac{1}{2\pi f_r C} \quad (2.7)$$

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad (2.8)$$

2.3. Potências Instantânea, Ativa, Reativa e Aparente

Numa Rede Elétrica ideal, as tensões e correntes são puramente sinusoidais com uma frequência típica de 50 Hz ou 60 Hz. Contudo, como anteriormente visto, a presença de componentes e cargas não ideais nas redes reais geram tensões e correntes distorcidas (não sinusoidais) e desfasadas que contêm harmónicas (Svensson, Stefan, 1999).

2.3.1. Potência Instantânea

Considere-se um circuito ao qual se aplicou uma tensão

$$u(t_i) = U_m \times \sin(\omega \cdot t_i) \quad (2.9)$$

e que é percorrido pela corrente

$$i(t_i) = I_m \times \sin(\omega \cdot t_i + \Phi) \quad (2.10)$$

A potência dissipada em cada instante t_i – Potência instantânea – é igual ao produto de U por I .

Supondo que os valores máximos da tensão e da corrente são, $U_m = 1.5$ V e $I_m = 1$ A, apresenta-se na Figura 2.9 a representação gráfica da variação de potência em função da Tensão (U) e Corrente (I), no caso particular em que $\Phi=0$ equivalente a uma resistência pura.

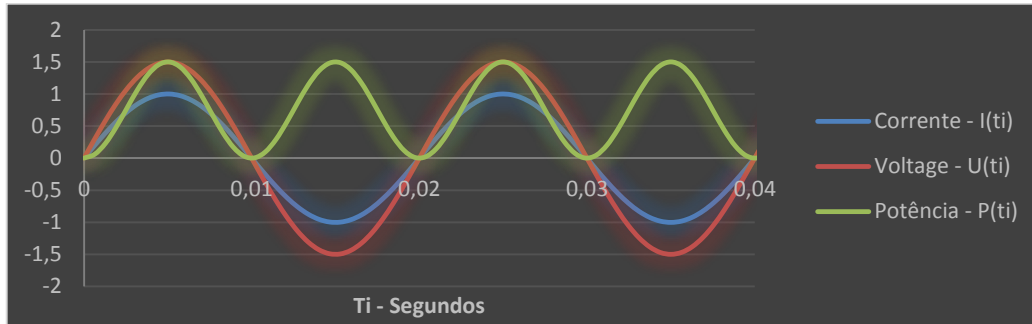


Figura 2.3.1:1: Potência, Tensão e Corrente numa Resistência pura.

O fato de a potência $P(t_i)$ ser sempre positiva significa que o circuito está a receber energia, estando neste caso a ser consumida na resistência (Ferreira Alves, Mário, 1999).

2.3.2. Potência Ativa

Há instantes em que a potência se anula, significando que a resistência não recebe potência e outros instantes em que a potência atinge o máximo. Na prática, apenas interessa o valor médio dessa potência (P), que corresponde no gráfico da (Fig. 2.3.1:1) ao valor médio da senoide de $P(t_i)$:

$$P = \frac{U_m I_m}{2} = \frac{\sqrt{2}U \cdot \sqrt{2}I}{2} = \frac{2UI}{2} = UI \quad (2.11)$$

Esta potência média é a Potência Ativa medida pelos sistemas de monitorização de energia e multímetros. A sua equação geral é:

$$P = RI^2 = UI \cdot \cos(\Phi) \quad (2.12)$$

Em que Φ é o ângulo entre a tensão e a corrente (Ferreira Alves, Mário, 1999).

2.3.3. Potência Reativa

A Potência Reativa pode ser positiva ou negativa, sendo que isso indica se a carga total é do tipo indutivo ou capacitivo (Svensson, Stefan, 1999).

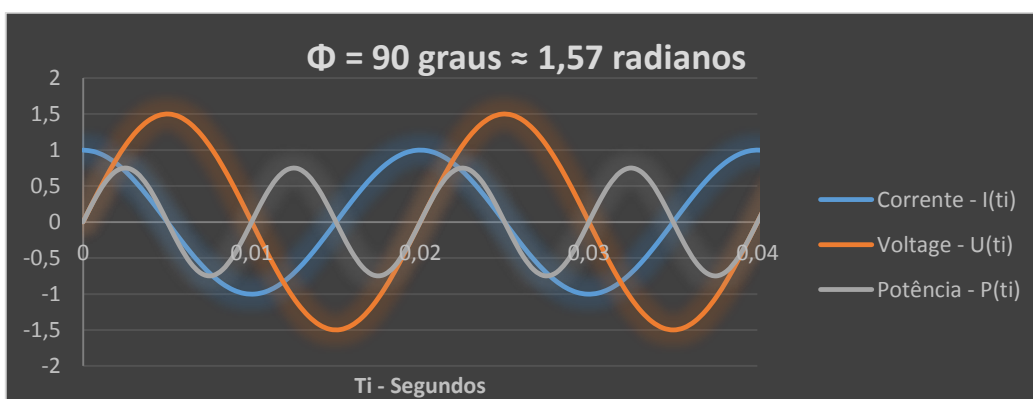


Figura 2.3.3:1: Potência tensão e corrente numa indutância.

Note-se que a potência instantânea $P(t_i)$ é alternadamente positiva e negativa, com uma frequência dupla da tensão e correntes existentes na indutância. Se a potência instantânea de um recetor é positiva, ele consome energia da fonte de alimentação. Nas alturas em que essa potência é negativa, esse recetor fornece energia à fonte de alimentação.

No caso da indutância, esta recebe e fornece energia, alternadamente, sendo a média nula, isto é, a energia recebida é igual à energia devolvida, pelo que não é dissipada. Ligando um Wattímetro para medir a potência Ativa, ele indica potência nula de $P = 0 \text{ W}$. Pode-se dizer que este o faseamento entre a corrente e tensão é zero podendo se dizer que estão em quadratura.

Apesar de não ser consumida, esta energia circula no circuito traduzindo-se numa corrente elétrica. A potência correspondente a esta energia oscilante designa-se por Potência Reativa e representa-se por Q .

No caso geral, para determinarmos a potência reativa de um elemento ou circuito, utilizamos a seguinte equação (Ferreira Alves, Mário, 1999):

$$Q = UI \cdot \sin(\Phi) \quad (2.13)$$

em que U e I são a tensão e corrente nesse elemento ou circuito e Φ é o ângulo entre tensão e corrente, sendo a sua unidade o Volt-Ampère Reactivo - VAR.

2.3.4. Potência Aparente

À potência que se consome num dado circuito AC com cargas resistivas e indutivas, atendendo à tensão e à intensidade da corrente que o percorre chama-se Potência Aparente. Esta potência representa-se por S , mede-se em Volt-Ampère - VA e pode ser determinada pela equação (Ferreira Alves, Mário, 1999):

$$S = U \cdot I = \sqrt{P^2 + Q^2} \quad (2.14)$$

Em termos vetoriais, pode-se representar o chamado triângulo de potências (caso indutivo):

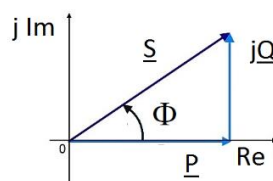


Figura 2.3.4:1: triângulo de potências.

Existe ainda o conceito de energia por quadrante, o qual surge na presença de Energia Ativa e Reativa em sistemas que podem tanto consumir como produzir energia. Na Figura 2.3.4.2 representa-se a distribuição de energia pelos quatro quadrantes (Gomes, André, 2012).

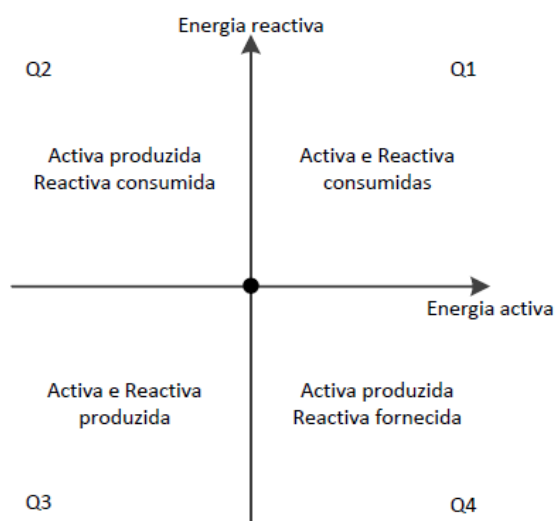


Figura 2.3.4:2: Energia por quadrante.

2.4. Fator de Potência

2.4.1. Inconvenientes da Potência/Energia Reativa

Embora só a potência ativa seja consumida, também a potência reativa representa um gasto para quem gera, transporta e distribui a energia, pois as perdas (*Efeito de Joule*) dependem da intensidade da corrente que percorre os condutores. Desta forma, ao fornecedor de energia interessa que não existam potências a oscilar na rede (reativas). Interessa portanto que a potência ativa **P** seja o mais próxima possível da potência aparente.

Conclui-se assim que pretende-se que a razão entre a potência ativa **P** e a potência aparente **S** seja o mais próximo do valor 1, sendo que o produto desta relação chama-se fator de potência, que pode ser calculado pela seguinte equação:

$$\frac{P}{S} = \frac{UI \cos(\Phi)}{UI} = \cos(\Phi) \quad (2.15)$$

A existência de fatores de potência inferiores a 1 nas instalações industriais deve-se aos recetores indutivos, maioritariamente motores elétricos (mas também outros, tais como lâmpadas fluorescentes), que são constituídos internamente por bobinas (indutâncias). Normalmente não existem recetores capacitivos (Ferreira Alves, Mário, 1999).

Podem enunciar-se alguns inconvenientes da existência de energia reativa nas instalações elétricas:

- Para o produtor de energia

*Uma qualquer estação de produção/Geração de energia elétrica possui um output de Potencia Aparente $S = UI$. Estando este a debitar a sua corrente máxima, a potência ativa **P** que ele está a produzir dependerá do Fator de Potência da instalação consumidora. Assim, se os utilizadores tiverem um baixo Fator de Potência implica que, para uma certa potência (ativa) a fornecer, o gerador de energia terá de ser construído para uma potência superior sendo, portanto, de maior volume e preço. O transformador elevador de tensão e toda a aparelhagem necessária (corte, seccionamento, proteção) têm de ser dimensionados para maiores intensidades. Tem-se assim que o produtor de energia exigirá que os utilizadores elevem o Fator de Potência das suas instalações ou que paguem uma quantia consoante a energia reativa que circula (Ferreira Alves, Mário, 1999).*

- Para o transportador e distribuidor de energia

Se uma linha, dimensionada para uma certa potência aparente ($S = UI$), vai alimentar instalações com fatores de potência baixos, implica que o investimento feito será mal aproveitado, pois transportará energia ativa (**P**) aquém da sua capacidade e, conseqüentemente, o consumidor receberá uma quantia baixa mesmo com a linha a plena carga.

De modo análogo, a mesma linha poderia alimentar mais instalações, desde que para as mesmas potências ativas os respetivos fatores de potência fossem superiores.

Quanto mais elevada é a intensidade de corrente que percorre uma linha, maiores são as perdas (quedas de tensão e *Efeito de Joule*), maior é o tamanho dos dispositivos de corte, seccionamento e proteção, assim como os transformadores que baixam a tensão das subestações e dos postos de transformação (Ferreira Alves, Mário, 1999).

- Para o utilizador de energia

Ao utilizador (consumidor) também interessa que o Fator de Potência seja o mais próximo possível de 1. No caso de um consumidor industrial, este terá de possuir uma subestação para diminuir a tensão sobredimensionada e por isso mais cara. Além disso para uma dada secção dos condutores de alimentação dos recetores, haverá maiores quedas de tensão e perdas de energia (que são contabilizadas e pagas). Poder-se-á nessa situação aumentar a secção dos condutores, o que aumenta

o custo da instalação. A aparelhagem de corte, seccionamento e proteção também terá de lidar com intensidades superiores. Se o fator de potência descer abaixo de um determinado limite, o consumidor será penalizado pelas entidades produtoras, transportadoras e distribuidoras, pagando o excesso de energia reativa (Ferreira Alves, Mário, 1999).

No caso de um consumidor que poderá usar o protótipo criado nesta dissertação, estando a usar uma tarifa BTE, este poderá pagar a energia reativa que consumir/produzir de acordo com a seguinte tabela (ERSE, 2014. Tarifas de Acesso às Redes):

TARIFA DE ACESSO ÀS REDES EM BTE		PREÇOS	
Potência		(EUR/kW.mês)	(EUR/kW.dia)*
	Horas de ponta	19,874	0,6534
	Contratada	1,118	0,0368
Energia activa		(EUR/kWh)	
	Horas de ponta	0,0489	
	Horas cheias	0,0424	
	Horas de vazio	0,0000	
	Horas de vazio normal	0,0221	
	Horas de super vazio	0,0199	
Energia reactiva		(EUR/kvarh)	
	Fornecida	0,0293	
	Recebida	0,0223	

* RRC art. 220.º, n.º 3

Tabela 2.4.1-1 – Tabela de preços de referência para consumo em BTE pela EDP Distribuição.

2.4.2. Compensação do Fator de Potência

Em instalações de alguma dimensão, tais como fábricas, é conveniente compensar baixos

Fatores de potência. Este “melhoramento” da instalação é vulgarmente efetuado recorrendo à utilização de condensadores em paralelo com os recetores, de modo a que a corrente “capacitiva” que neles circula anule (reduza ao máximo) a corrente “indutiva” dos recetores:

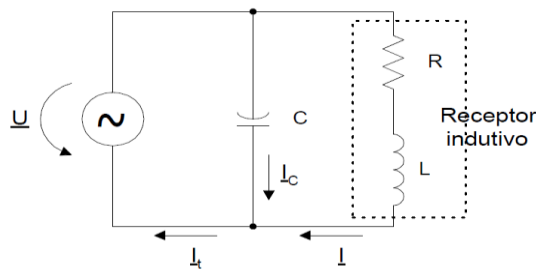


Figura 2.4.2:1: Compensação do fator de potência.

Para calcular a capacidade dos condensadores (podem ser vários associados em paralelo), pode aplicar-se a equação (2.16) para determinar a diferença entre a Potencia reativa real e a pretendida. De seguida aplicando-se a equação (2.17) consegue-se determinar a capacidade do condensador necessária para compensar a Potencia reativa em excesso.

$$Q_c = P \cdot \tan \Phi_i - P \cdot \tan \Phi_f \quad (2.16)$$

$$Q_c = X_c I_c^2 = X_c \left(\frac{U}{X_c} \right)^2 \leftrightarrow C = \frac{Q_c}{2\pi \cdot f \cdot U^2} \quad (2.17)$$

2.5. Problemas de qualidade da Energia:

Com o surgimento da Diretiva de Conselho nº 85/374 sobre responsabilidade de produtos defeituosos que no seu artigo 2º define a eletricidade como um produto, tornou-se obrigatório definir as suas características. Surge assim, depois de um longo processo a NP EN 50160:2001, com o termo de homologação nº 128/2001, de 2001.05.16. Com o título “Características da tensão fornecida pelas redes de distribuição pública de energia elétrica”, é composta por três partes, a saber (FEUP, 2014):

- Generalidades
- Características da alimentação em baixa tensão.
- Características da alimentação em média tensão.

A segunda parte da norma diz respeito às características da alimentação em baixa tensão, a qual aplicanda a BTN, destaca-se:

Frequência (fr): deve ser igual a 50 Hz (frequência da tensão de alimentação). Em condições normais de exploração, o valor médio da frequência fundamental em intervalos de 10 segundos deve estar compreendido entre os seguintes valores:

Intervalo (Hz)	Duração (Ano)
$49,5 \leq fr \leq 50,5$	99,50%
$47 \leq fr \leq 52$	100%

Tabela 2.5-1 – Intervalo de frequências estipulada para a qualidade da rede.

Tensão de alimentação: a tensão normalizada U_n para as redes de BT é 230 V entre fase e neutro para sistemas trifásicos com quatro condutores.

Variação da tensão de alimentação (em condições normais de exploração):

Intervalo (Volt)	Duração
$207 \leq U_n \leq 253$	95% Dos valores eficazes médios devem situar-se nesta gama
$195,5 \geq U_n \geq 253$	Todos os valores eficazes médios de 10 minutos devem situar-se nesta gama.

Tabela 2.5-2 – Intervalo de tensões estipulada para a qualidade da rede.

2.6. Sistemas Trifásicos

2.6.1. Sistemas Trifásicos Vs sistemas Monofásicos

Apresentam-se a seguir algumas vantagens dos sistemas trifásicos em relação aos monofásicos, nomeadamente ao nível da sua produção, transporte e utilização (Ferreira Alves, Mário, 1999):

- Considerando dois alternadores, um monofásico e outro trifásico, de igual volume e preço, o segundo tem uma potência aproximadamente 50% superior ao primeiro. Tal deve-se ao facto de haver um maior aproveitamento do perímetro do estator, isto é, há mais bobinas que são sede de f.e.ms. induzidas.
- O somatório da secção dos condutores necessários para transportar uma determinada potência é menor que nos sistemas monofásicos, em igualdade de condições de potência transportada, perdas e tensão nominal de transporte.
- Para transportar uma dada quantidade de energia são suficientes três (ou quatro, incluindo o neutro) fios em sistemas trifásicos, enquanto em monofásico seriam necessários seis fios de igual secção (ou dois de secção tripla).

- A capacidade dos sistemas trifásicos de produzir campos magnéticos girantes, permite a utilização dos motores assíncronos trifásicos, aparelhos simples, robustos e económicos que detêm a quase totalidade do mercado em tração elétrica industrial.
- A partir de um sistema trifásico podem obter-se três sistemas monofásicos (tal como em nossas casas).

2.6.2. Sistema Equilibrado e condutor Neutro

Considere-se um alternador a alimentarem três recetores idênticos (resistências, neste caso), um em cada fase:

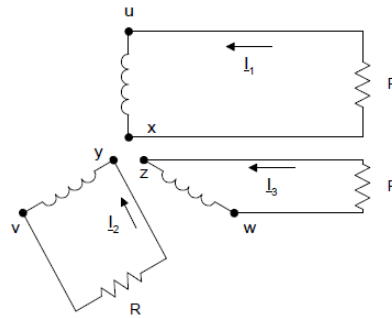


Figura 2.6.2:1: Alimentação independente de três recetores idênticos.

Para alimentar independentemente três recetores, é portanto necessário utilizar seis fios. Se os três recetores tiverem a mesma impedância, estes são percorridos por três corrente **I1**, **I2** e **I3**, com idêntico valor eficaz mas desfasadas de 120° (Ferreira Alves, Mário, 1999).

Diz-se então que o sistema está **equilibrado**, pois a soma das três correntes é sempre nula (a soma de três vetores iguais e desfasados de 120° é um vetor nulo).

Se reunirmos os três terminais x, y, z, num único ponto N, chamado de ponto neutro e substituímos os três condutores de retorno (vindos dos recetores) por um único condutor - condutor neutro (ou fio neutro), a corrente nesse condutor será nula e esquematiza-se da seguinte forma:

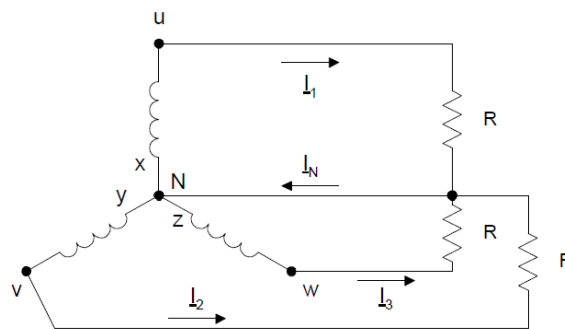


Figura 2.6.2:2: Sistema equilibrado de cargas com neutro (corrente no neutro é nula).

Pode desta forma distribuir-se a energia elétrica por meio de quatro condutores, sendo três designados por condutores de fase (ativos) ou simplesmente fases, em linguagem corrente. As três fases simbolizam-se normalmente pelas letras R, S e T. O condutor de neutro está normalmente ligado à terra, pelo que se encontra com potencial zero (Ferreira Alves, Mário, 1999).

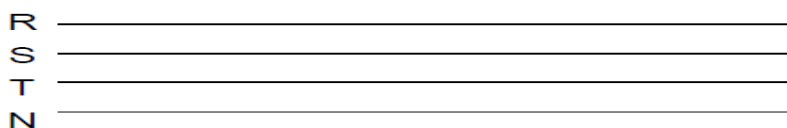


Figura 2.6.2:3: Transporte de energia elétrica trifásica por meio de quatro condutores.

2.6.3. Tensões Simples e Compostas

Num sistema trifásico existem diferentes tensões:

- **Tensões simples - U_S**

Tensão entre cada condutor de fase e o neutro. Nas redes de distribuição de baixa tensão, aproximadamente 230 V.

- **Tensões compostas - U_C**

Tensão entre dois condutores de fase. Nas redes de distribuição de baixa tensão, aproximadamente 400 V.

Na Figura 2.17 seguinte, U_{SN} é uma tensão simples e U_{ST} é uma tensão composta:

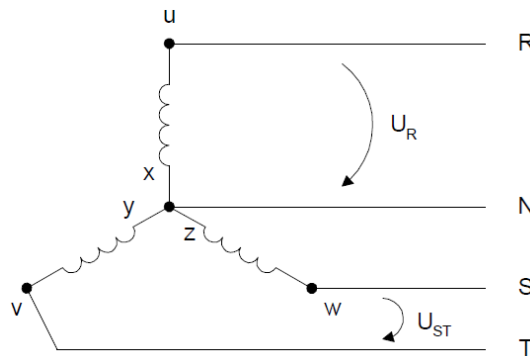


Figura 2.6.3:1: Tensões simples e compostas, composição em estrela.

Temos portanto três tensões simples e três tensões compostas distintas entre si:

- **Tensões simples:**

Tensão entre a fase R e a fase N - U_R

Tensão entre a fase S e a fase N - U_S

Tensão entre a fase T e a fase N - U_T

- **Tensões compostas:**

Tensão entre a fase R e a fase S - $U_{RS} = U_R - U_S$

Tensão entre a fase S e a fase T - $U_{ST} = U_S - U_T$

Tensão entre a fase T e a fase R - $U_{TR} = U_T - U_R$

Pode-se também representar estas tensões em termos vetoriais:

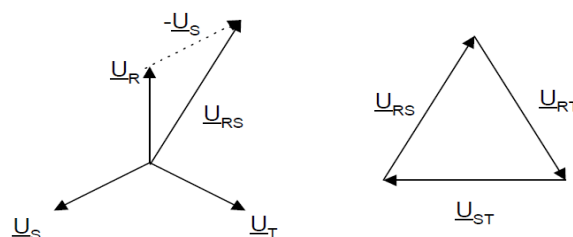


Figura 2.6.3:2: Representação vetorial das tensões simples e compostas.

Demonstra-se que o comprimento dos vetores das tensões compostas é $\sqrt{3}$ vezes superior ao das tensões simples, isto é:

$$U_C = \sqrt{3} \times U_S \quad (2.18)$$

De facto, para as redes de distribuição de baixa tensão, tem-se que:

$$U_S \approx 230 \text{ V} \quad (2.19)$$

$$U_C \approx \sqrt{3} \times 230 \approx 400 \text{ V} \quad (2.20)$$

Nas redes de distribuição BT, normalmente indicam-se as tensões do modo: 230/400 V.

2.6.4. Ligação de Recetores Trifásicos – Triângulo (Δ) e Estrela (Y)

O tipo de ligação mais comum nos geradores trifásicos é a ligação em estrela (Y), pois neste tipo de ligação pode-se alimentar facilmente as cargas monofásicas, devido à existência do fio neutro, ver (Fig.2.6.4:1)

Já os recetores trifásicos podem ser ligados de duas formas:

- Em **estrela - Y**
- Em **triângulo - Δ**

Na ligação de recetores **em estrela**, poderão ocorrer dois casos:

- Os recetores têm a mesma impedância - **sistema equilibrado**
- Os recetores têm impedâncias diferentes - **sistema desequilibrado**

Repare-se que num sistema em estrela equilibrado, o condutor neutro é dispensável (tal como foi referido atrás), isto é, ele pode ser retirado sem alteração do funcionamento dos recetores, já que a sua corrente é sempre nula. De facto, cada uma das linhas de fase faz de retorno em relação às outras duas. Há motores trifásicos cujas bobinas estão ligadas em estrela. Assim, poder-se-ia, apenas idealmente, alimentar o motor apenas com as três fases, dispensando-se o neutro, como a seguir se mostra.

No caso de um sistema em estrela desequilibrado, o somatório das correntes nas fases não é nulo, sendo indispensável a ligação no condutor de neutro.

Mesmo nos casos em que a estrela é normalmente equilibrada, não se deve eliminar o neutro, dado que se faltar uma fase (por corte de um dispositivo de proteção, por exemplo) estabelece-se um desequilíbrio de tensões.

Saliente-se ainda que se pretende equilibrar ao máximo os sistemas trifásicos, de modo a que a corrente no condutor do neutro seja a menor possível. Uma menor corrente no neutro tem como vantagem permitir a utilização de um condutor de menor secção, para as mesmas perdas energéticas. É por esse motivo que o condutor do neutro é normalmente mais fino que os condutores de fase (caso das linhas de transporte de energia elétrica com neutro).

Na ligação de recetores em triângulo, os recetores estão ligados entre as fases, tal como mostra a figura seguinte, para o caso de resistências:

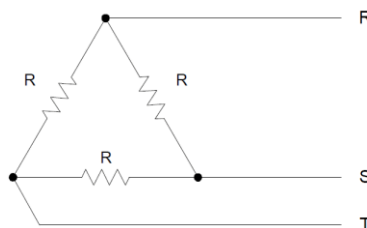


Figura 2.6.4:1: Ligação de recetores em triângulo.

Tal como na ligação de recetores em estrela, na ligação em triângulo poderão ocorrer dois casos:

- Os recetores têm a mesma impedância - **sistema equilibrado**
- Os recetores têm a impedâncias diferentes - **sistema desequilibrado**

A corrente num recetor (de fase) pode ser calculada dividindo a tensão compostas aos seus terminais pela sua impedância.

As correntes de linha podem ser determinadas de duas maneiras, consoante o equilíbrio ou desequilíbrio do sistema:

- Sistema equilibrado - as correntes nas linhas (R, S, T) são $\sqrt{3}$ vezes superiores às correntes nos recetores (correntes de fase).
- Sistema desequilibrado - as correntes nas linhas são determinadas em termos vetoriais, através da aplicação da Lei dos Nós de Kirchoff aos três nós.

Como conclusão pode dizer-se que nas configurações em estrela com neutro e em triângulo os recetores (monofásicos) funcionam independentemente uns dos outros.

2.6.5. Cálculo de Potência dos sistemas Trifásicos

Quer a carga seja equilibrada ou não, podem calcular-se (medir-se) as potências consumidas em cada fase e somar-se. Assim, somam-se as potências ativas aritmeticamente:

$$P_{Total} = P_R + P_S + P_T \quad (2.21)$$

As potências reativas têm de se somar algebricamente (tendo em conta se são indutivas ou capacitivas)

$$Q_{Total} = Q_R + Q_S + Q_T \quad (2.22)$$

No caso de **sistemas equilibrados** (triângulo ou estrela), podem ser usadas as seguintes equações:

$$P_{Total} = \sqrt{3} \cdot U_C \cdot I_L \cdot \cos(\Phi) \quad (2.23)$$

$$Q_{total} = \sqrt{3} \cdot U_C \cdot I_L \cdot \sin(\Phi) \quad (2.24)$$

$$S_{Total} = \sqrt{3} \cdot U_C \cdot I_L \quad (2.25)$$

onde:

- U_C – é a tensão composta (entre duas fases)
- I_L – é a corrente nas linhas

3. Estado da Arte

Neste capítulo, pretende-se descrever os princípios base de funcionamento por detrás das tecnologias atuais de monitorização de energia e de organização e funcionamento de um servidor web.

3.1. Contadores Eletromecânicos

Trata-se de um medidor de energia elétrica tipo indução através de um motor elétrico cuja interação de fluxos magnéticos produz movimentos no rotor com correntes elétricas. O medidor é composto por um estator, um rotor uma carcaça e um registador. Sendo que este ultimo regista, com uma relação pré-determinada, o número de rotações efetuadas pelo motor associado à quantidade de energia consumida. Isto porque a velocidade de rotação do disco é proporcional à potência da Carga (Mínguez, Agustín, 2007).

O principio de funcionamento do medidor tipo indução é baseado no fenómeno de indução eletromagnética, sendo que um condutor percorrido por uma corrente I na presença de um campo

magnético B , fica submetido a uma força cujo sentido é dado pela regra da mão direita e possui módulo que é dado por:

$$\vec{F} = \overline{B \cdot \vec{I} \cdot L \cdot \sin \alpha} \quad (2.26)$$

Onde L é o comprimento do condutor sobre ação do campo magnético B e α o ângulo compreendido entre o vetor do campo magnético e a direção do vetor $\vec{I} \cdot L$ no espaço.

Como já foi observado no Capítulo 1.3 é de longe a forma de contagem de energia mais usada em Portugal, existindo quase 1 contador para cada 2 habitantes. Contudo a sua instalação tem cessado nos últimos anos em detrimento de novas tecnologias.

3.2. Contadores Híbridos

São uma mistura entre tecnologia eletromecânica e eletrónica, sendo que a aquisição de dados é feita de forma semelhante a um contador eletromecânico, mas a informação de consumo obtida é processada de uma forma digital para o consumidor, conseguindo mesmo obter registos de consumo e os categorizar por tarifas horárias (EDP, 2014).

3.3. Contadores Estáticos

3.3.1. Contadores Eletrónicos

Usam tecnologia analógica-digital para ler e apresentar a informação de consumo de energia com um grau de complexidade e exatidão superior aos sistemas eletromecânicos. Dentro das principais vantagens da utilização de um contador eletrónico pode-se destacar: automação do processo de aquisição de dados e tratamento dos dados do consumidor para fins de facturamento, e ganhos na exatidão.

A figura 3.3.1.1 mostra o diagrama padrão de um medidor eletrónico, onde se pode ver os princípios de funcionamento do mesmo, assim como os componentes básicos necessários para se efetuar a medição de energia elétrica, as quais maioritariamente possuem arquitetura SAR ou Sigma Delta (Gomes, André, 2012).

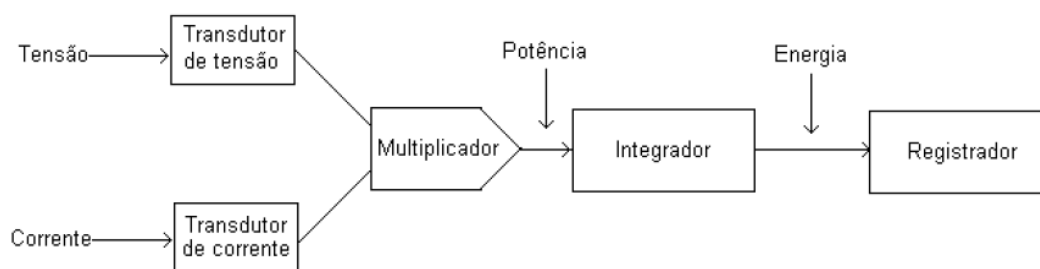


Figura 3.3.1.1: Diagrama de Blocos de um contador eletrônico.

O típico contador eletrónico dispõe de sensores (transdutores) de tensão e corrente, os quais reduzem os sinais para níveis mais baixos que possam ser convertidos por ADC's – (Analog to Digital Converter) e adequá-los de modo a serem multiplicados. A potência é obtida através do bloco multiplicador. A energia é obtida através do integrador, finalmente esse valor é armazenado no bloco registrador (Mínguez, Agustín, 2007)).

3.3.2. Sistemas de Monitorização de Energia

A tecnologia atual de medição eletrônica garante melhor exatidão que os medidores eletromecânicos, oferecendo informações detalhadas sobre o consumo, contudo existem sistemas que vão ainda mais longe ao tornarem o processamento da informação mais detalhado e dinâmico. Estes Sistemas de Monitorização de Energia fazem monitorização à distância via *modem*, GPRS, e outras tecnologias, dependendo do modelo, e ainda processamento das informações de consumo em tempo real criando diagramas e esquemas de análise deste. O propósito é tornar mais fácil a compreensão do consumo de energia por forma a tomar medidas de redução do consumo e melhoramento da qualidade de energia com menos variações e menos interrupções no seu fornecimento, quer por parte do comercializador de energia que por parte do próprio consumidor (Stromback, J., Dromacque, C., Yassin, M.H. 2011).

3.4. Fontes de alimentação

3.4.1. Baterias

A análise feita demonstra que o uso de pilhas recarregáveis no protótipo pretende determinar se será plausível o uso de pilhas recarregáveis para alimentar o protótipo de Monitorização de energia Desenvolvido nesta dissertação.

Começando por escolher o tipo de pilha recarregável a usar, a selecionada foi a ENERGIZER E95 D.

Recorrendo ao uso de um multímetro e uma fonte de alimentação apropriada de transformação AC/DC para o protótipo determinou-se a tensão e corrente DC de alimentação durante um pleno período de funcionamento do protótipo, chegando aos resultados médios de:

- $I_{in} = 0,190 A$
- $U_{in} = 12,22V$

Logo, uma vez que a pilha possui uma voltagem nominal de 1,2 V, serão necessárias $12,22/1,2 \rightarrow$ pelo menos 11 destas pilhas colocadas em série. ADC

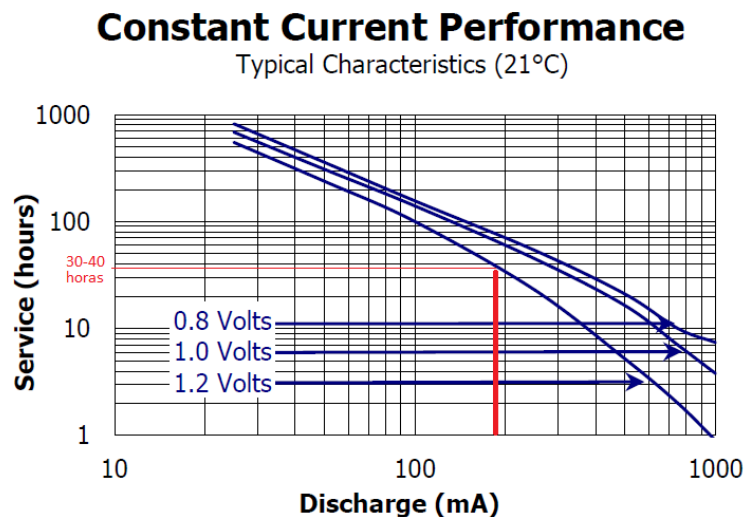


Figura 3.4.1.1: Curva de descarga das pilhas ENERGIZER E95 D.

Tendo em conta a curva de descarga fornecida pelo fornecedor das pilhas e o consumo de I_{in} medido, conclui-se observando a figura 3.4.1.1 que o tempo esperado de autonomia será entre as 30-40 horas.

Tal necessidade de consumo, torna inviável financeiramente o uso de baterias recarregáveis para alimentar o sistema de Monitorização de Energia.

3.4.2. Transformador AC/DC ligado à rede.

Através de uma medição do consumo do Protótipo Monitor de Energia ao longo de 30 minutos, enquanto este se encontrava a medir outras cargas, obteve-se dados de consumo da sua potência ativa, sendo o consumo médio medido de 3,39 W. Contudo este valor não pode ser considerado viável para cálculo da energia, uma vez que as medições estão condicionadas com a periodicidade do funcionamento do protótipo.

Contudo dado a inviabilidade do uso de baterias, a solução AC/DC torna-se a única solução economicamente viável para alimentar o protótipo desenvolvido nesta dissertação.

3.5. Sensores

Para o sistema deste projeto os sensores de correntes suportados são Transformadores de Corrente (CT), Bobinas de Rogowski e Resistências de Shunts.

3.5.1. De Tensão

Normalmente existem duas alternativas para sensores de tensão

- Transformador de Tensão
- Divisor resistivo

O transformador de tensão é um elemento volumoso e de elevado custo, mas a sua vantagem é o isolamento galvânico. Por outro lado, o divisor resistivo é relativamente simples de implementar, com menores dimensões e os seus componentes têm um custo muito mais baixo; porém, a sua desvantagem é não isolar o sistema de medida da rede, o que deve ser compensado com o dimensionamento do circuito de condicionamento de forma a tentar reduzir possíveis picos de tensão na rede, tipicamente introduzindo uma resistência à entrada de cada ADC. Pelas razões descritas optou-se pela utilização do divisor resistivo (Gomes, André, 2012).

O MAX78630+PPM implementa um divisor resistivo com um ADC bipolar, operando em um intervalo de -250 a +250 mV (Maxim Intergrated, 2014).

3.5.2. De Corrente

Tendo em conta o kit de desenvolvimento usado no sistema de monitorização desenvolvido, existem 3 opções de uso de sensores de corrente.

- Resistência de shunt
- Transformador de corrente (CT).
- Bobina de Rogowski

A medição de corrente através de uma resistência de shunt baseia-se no princípio de inserir uma resistência de muito baixo valor (quando comparada com a carga) em série com a carga. Através da medição do valor da tensão aos terminais desta resistência de forma precisa, torna-se possível calcular o valor da corrente. A tolerância da resistência de shunt deve ser suficientemente baixa para garantir, após a calibração do contador, a classe de precisão pretendida. O esquema deste tipo de sensor está representado na figura seguinte (Gomes, André, 2012).

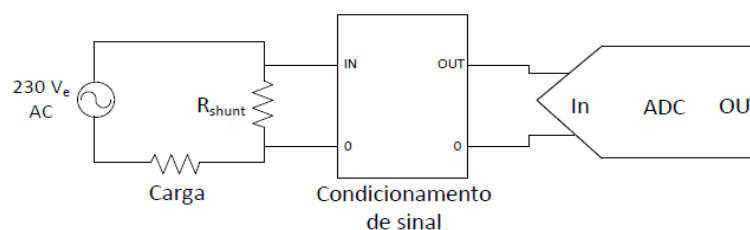


Figura 3.5.2:1: Medição de corrente mediante uma resistência de shunt

As vantagens deste método são a simplicidade do circuito, baixo custo e exatidão de medida, porém, a desvantagem está na potência dissipada. Uma vez que há dissipação de calor, verifica-se um aumento da temperatura da resistência com uma relação que é proporcional ao quadrado da corrente que atravessa a resistência e, por essa causa, para correntes elevadas utilizam-se outros tipos de sensores

Os transformadores de corrente – (CT) são dispositivos usados para reduzir de elevadas correntes primárias a baixas correntes secundárias, mais fáceis de medir. Como os transformadores de tensão tradicionais o rácio de espiras (enrolamentos) determina a relação entre a corrente de entrada e a corrente de saída. CT's de várias formas e tamanhos são usados como uma solução de interface entre correntes elevadas e dispositivos instrumentais.

Um CT é útil para medições de corrente AC. Atua tal e qual como um transformador de tensão, mas tipicamente tem apenas uma bobina primária (os condutores que transportam a corrente para ser medida). Ao contrário dos transformadores de tensão, não existe conexão física à linha que se encontra a ser medida.

O CT usa o campo magnético gerado pela corrente AC que corre através do cabo primário para induzir uma corrente secundária. O rácio no número de voltas no primário para o número de voltas no secundário determina a amplitude de corrente de saída do CT. A saída do CT atua como uma fonte de corrente (Elkor Technologies Inc, 2006).

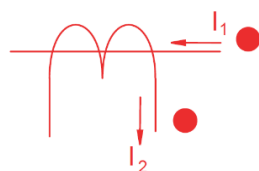


Figura 3.5.2:2: Símbolo de um Transformador de corrente (CT).

Para muitas aplicações de medição de corrente AC, a direção do CT, e a polaridade da saída das espiras é irrelevante. Contudo, para aplicações que requeiram medições de potência, a polaridade do CT deve ser tida em conta para evitar situações de inversão de fase. A tensão e a corrente medida devem ter a mesma polaridade para manter as medições do Fator de Potência e direção do fluxo de potência precisas e consistentes (Elkor Technologies Inc, 2006).

As Bobinas de Rogowski são usadas para medir corrente alterna. Elas funcionam usando o campo magnético produzido pelo condutor de corrente, sem terem a necessidade de fazer contacto elétrico com o condutor. Estas operam sobre um princípio simples, Uma “bobina com ar” é colocada à volta do condutor numa forma toroidal para que o campo magnético produzido possa induzir uma tensão na bobina. A bobina é efetivamente uma indutância mútua acoplada ao condutor a ser medido e a tensão de saída é proporcional à taxa de variação da corrente. Para completar um transdutor é integrado a esta tensão eletronicamente Figura 3.5.2.3 para providenciar uma saída que reproduz a forma ondulada da corrente. Esta combinação de bobine e transdutor produz um sistema onde a saída é independente da frequência, que possui uma resposta à fase precisa, e que pode medir correntes de onda complexas.

A saída do integrador pode ser usado com qualquer tipo de dispositivo indicador eletrónico, tal como um voltímetro, osciloscópio, sistema de proteção ou sistema de medição. As bobinas são enroladas quer numa forma flexível que é subsequentemente enrolada à volta do condutor a ser medido, ou numa forma rígida toroidal (D. A. Ward, J. La T. Exon, 1993).

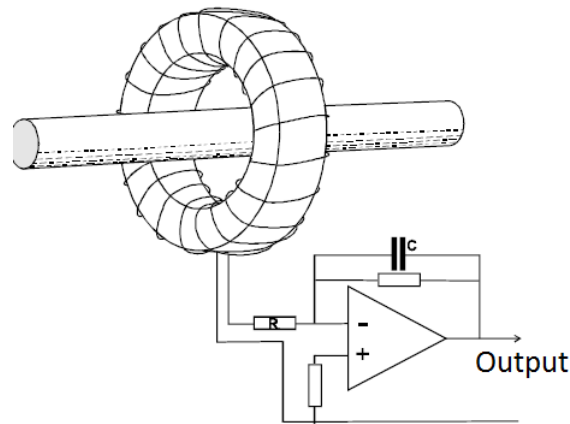


Figura 3.5.2:3: Medição de corrente mediante uma Bobina de Rogowski.

3.6. Servidor Web e Webpage

Um servidor Web pode ser duas coisas – um computador onde um Website é alojado ou o programa que gera o servidor e permite o acesso a um *website* através da Internet. Sendo que o termo tanto se pode referir a Hardware como a Software.

O processo de aceder a um *website* começa num browser web onde o utilizador insere o URL¹⁰ (*Uniform Resource Locator*) na barra de endereços. Cada página web possui um endereço único (URL) na Internet, o que significa que a mesma página não pode existir em dois locais. Então assim que o pedido é enviado pelo utilizador para aceder à página web, na verdade o que acontece é que o pedido URL é convertido em endereço de IP (*Internet Protocol*) que liga ao local onde o *website* se encontra alojado. O pedido é reenviado ao Computador Servidor que então passa pelo Software desse Servidor, onde este vai começar a procurar a página web requerida na memória do disco rígido. Ao encontrar o ficheiro, este envia uma resposta ao *browser* que transporta o ficheiro que é então interpretado como a página web requerida.

Uma página web típica não contém apenas texto, mas também ficheiros multimédia embebidos como imagem e animações Flash. Estes ficheiros “extra” são separados da página web e são buscados pelo *browser* ao servidor um a um. Apenas o *browser* determina como a página web é apresentada, o servidor web não possui qualquer controlo sobre esta (*Web developers notes, Website*).

3.6.1. O Computador como Servidor – Hardware

Um *website* é uma coleção de páginas web que são ficheiros digitais geralmente escritos usando linguagem HTML. Para que o Website se encontre disponível a qualquer momento, ele precisa ser armazenado ou “alojado” num computador que também se encontre sempre conectado à internet.

Alojar um *website* num computador pessoal requer um trabalho e monitorização constante, por isso a opção alternativa de comprar um servidor de alojamento a preços definidos pela exigência do serviço torna-se na grande maioria dos casos a melhor opção.

Um computador que seja um servidor deve ser rápido, ter uma grande capacidade de armazenamento em disco rígido e bastante memória RAM. Mas o mais importante é ter um endereço permanente de acesso à internet, conhecido como endereço IP. Se este mudar não será mais possível observar o Website (*Web Developers Notes, 2014*).

¹⁰ Localizador Padrão de Recursos, é o endereço de um recurso, disponível em uma rede. Seja a Internet, ou privada como uma intranet, ou de um ou vários dispositivos ligados entre si.

3.6.2. O servidor Web como programa – Software

Um programa servidor web é *software* que corre num sistema que aloja e corre o website. O seu propósito principal é servir páginas web, o que significa que espera por pedidos de browsers web (também conhecidos como clientes) e responde enviando os dados requeridos. Esta relação Servidor cliente é uma das bases de como a Internet funciona.

Apesar da sua principal função de servir conteúdos, uma implementação plena de HTTP também inclui formas de receber conteúdos dos clientes, como por exemplo para fazer *upload* de ficheiros (Web Developers Notes, 2014).

Muitos servidores web genéricos também suportam linguagem de servidor (*Server-side scripting*) usando ASP, PHP ou outras linguagem de programação. Isto significa que o comportamento do servidor da página web pode ser escrito em ficheiros separados (ex. vários ficheiros PHP que estão contidos no servidor e montam o suporte dinâmico da página web), enquanto o *software* do servidor mantém-se inalterado. Esta função permite então alojar vários ficheiros no servidor com conteúdo dinâmico, podendo interagir e modificar informações provenientes de bases de dados como MySQL.

3.6.3. Computação Cloud “Nuvem”

A computação em nuvem é um modelo que permite de forma conveniente, o acesso a uma rede omnipresente sob demanda de um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e lançados com um esforço mínimo de gerenciamento ou interação com o provedor de serviços.

Este modelo de nuvem é composto por cinco características essenciais, três modelos de serviço e quatro modelos de implantação (P.Mell, T.Grance, 2011).

Características essenciais¹¹:

- Self-service On-Demand.
- Acesso ampla de redes.
- Agrupamento de recursos.
- Rápida elasticidade.
- Serviço Mesurado.

Modelos de Serviço:

Software as a Service (SaaS): A capacidade fornecida ao consumidor é a utilização de aplicativos dos provedores em execução numa infraestrutura¹² tipo nuvem. As aplicações são acessíveis a partir de vários dispositivos, como um navegador da web (por exemplo, e-mail baseado na web), ou uma interface de programa. O consumidor não gerencia nem controla a infraestrutura de nuvem subjacente, incluindo rede, servidores, sistemas operacionais, armazenamento ou até mesmo recursos de aplicativos individuais, com a possível exceção de configurações de aplicativos limitados específicos do utilizador (P.Mell, T.Grance, 2011).

¹¹ Ver Glossário de Termos, anexo A.

¹² A infraestrutura de nuvem é o conjunto de *hardware* e *software* que permite as cinco características essenciais de computação em nuvem. A infraestrutura de nuvem pode ser visto como contendo tanto uma camada física e uma camada de abstração. A camada física consiste nos recursos de *hardware* que são necessários para apoiar os serviços de nuvem a ser prestado, e geralmente inclui servidores, armazenamento e componentes de rede. A camada de abstração consiste no *software* implantado em toda a camada física, que se manifesta as características essenciais da nuvem. Conceitualmente a camada de abstração fica acima da camada física

Platform as a Service (PaaS): A capacidade fornecida ao consumidor é a de implementar na infraestrutura tipo nuvem, aplicações criadas ou adquiridas pelo consumidor, bibliotecas, serviços e ferramentas suportadas pelos provedores. O consumidor não gerência ou controla a infraestrutura subjacente à nuvem, incluindo a rede, servidores, sistemas operacionais ou armazenamento, mas tem controlo sobre os aplicativos implementados e as definições de configuração, possivelmente, para o ambiente de hospedagem de aplicativos (P.Mell, T.Grance, 2011).

Infrastructure as a Service (IaaS): A capacidade fornecida ao consumidor é a de aprovisionar processamento, armazenamento, redes e outros recursos computacionais fundamentais em que o consumidor é capaz de implantar e executar *software* arbitrário, que pode incluir sistemas operativos e aplicativos. O consumidor não gerência nem controla a infraestrutura de nuvem subjacente, mas tem controlo sobre sistemas operacionais, armazenamento e aplicações implementadas. E possivelmente controlo limitado de alguns componentes da rede (por exemplo, firewalls do host) (P.Mell, T.Grance, 2011).

Modelos de Implantação:

Nuvem privada: A infraestrutura de nuvem é provisionada para uso exclusivo por uma só organização que agrupa vários consumidores (por exemplo, unidades de negócio). Pode ser detida, gerida e operada pela organização, uma terceira, ou alguma combinação destas.

Nuvem comunitária: A infraestrutura de nuvem é provisionada para uso exclusivo de uma comunidade específica de consumidores de organizações que têm preocupações comuns (considerações por exemplo, missão, requisitos de segurança, política e cumprimento). Pode ser detida, gerida e operada por uma ou mais das organizações na comunidade, uma terceira, ou alguma combinação destas.

Nuvem Pública: A infraestrutura de nuvem é provisionada para uso aberto ao público em geral. Pode ser detida, gerida e operada por uma empresa, acadêmico ou organização governamental, ou alguma combinação deles. Ela existe nas instalações do provedor da nuvem.

Nuvem Híbrida: A infraestrutura é uma composição de duas ou mais distintas infraestruturas de nuvem (privada, comunitária ou pública) que permanecem entidades únicas, mas são unidas por tecnologia padronizada ou proprietária que permite a portabilidade de dados e aplicações.

4. Arquitetura do sistema

4.1. Kit de desenvolvimento MAX78630+PPM

O kit de desenvolvimento MAX 78630+PPM demonstra as capacidades do SoC (system on chip) MAX 78630+PPM para monitorizar um sistema elétrico trifásico usando até 3 tensões e 3 correntes (sensores incluídos). Este Kit, ver (Fig. 4.1.1), pode ser conectado a um PC através de um cabo USB providenciando assim tanto alimentação elétrica como capacidade de comunicação com a placa. Uma interface gráfica baseada em Windows (GUI) permite uma comunicação simplificada com o aparelho de uma forma que o permite estudar (Maxim Intergrated, 2013).

As componentes do Sistema e definições do firmware¹³ estão desenhadas para operar com os seguintes intervalos de operação:

Voltagem	Corrente	Frequência da Linha
10 – 400V AC	10mA – 50A	46 – 64

¹³ Ver Glossário de Termos, anexo A.

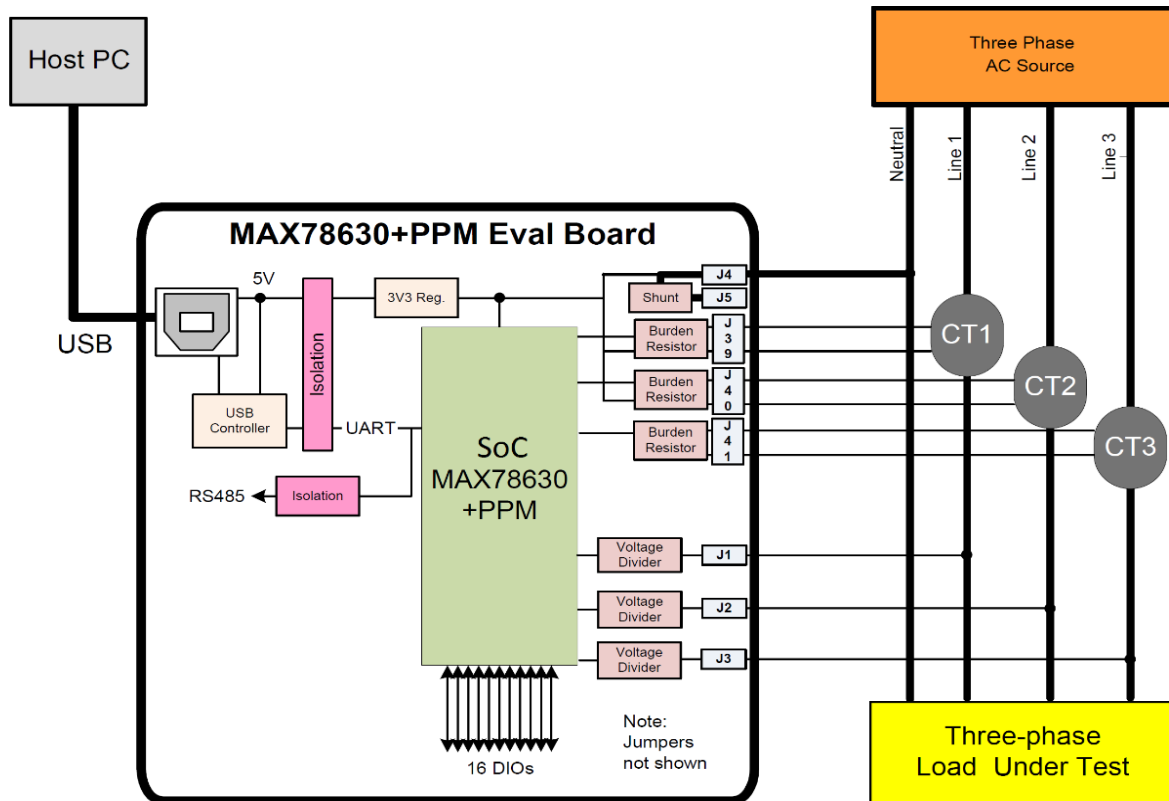


Figura 4.1.1: Diagrama típico de aplicação do Kit de desenvolvimento MAX78630+PPM

4.1.1. Visão Geral do SoC

O MAX78630+PPM providencia todos os blocos necessários para uma medição precisa de consumo AC, entre os quais encontram-se 6 entradas analógicas que funcionam como interface com os canais de medição de tensão e corrente. Tensões escalonadas são alimentadas dos sensores ao single converter front-end utilizando um conversor analógico digital (ADC) delta-sigma de alta resolução. Um processador embestado de 24-bits e firmware realizam toda a computação necessária e formatação de dados para um envio detalhado ao anfitrião (Host)¹⁴. Com memória flash integrada para armazenamento de coeficientes de calibração não voláteis e definições de configuração do dispositivo (Maxim Intergrated, 2014).

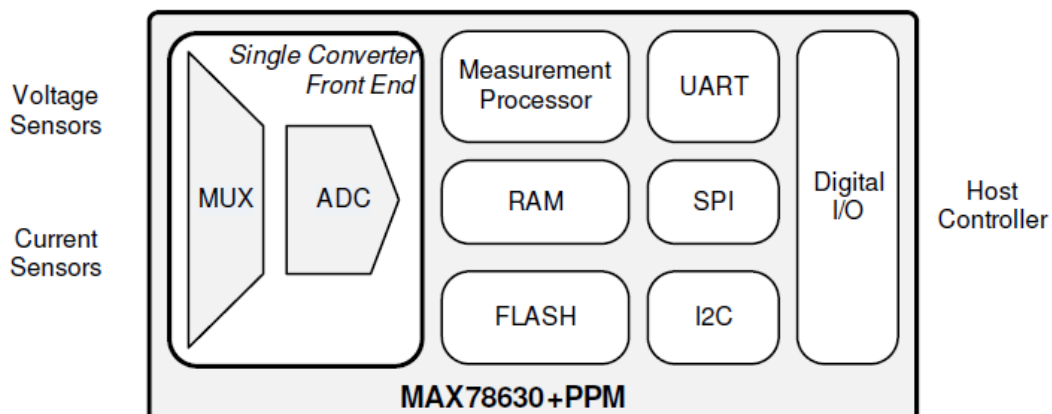


Figura 4.1.1.1: Diagrama de blocos dos recursos disponíveis no SoC MAX78630+PPM.

¹⁴ Trata-se do Anfitrião (PC ou Data Logger) que controla a entrada e saída de informação do MAX78630+PPM

4.1.1.1. Entradas analógicas

Até 6 sensores externos podem ser conectados ao MAX78630+PPM. As entradas analógicas em qualquer pino estão escalonadas no que diz respeito ao V_{3P3A} ¹⁵ para tensão com intervalo máximo de ± 250 mV. Considerando uma forma sinusoidal AC, a tensão eficaz máxima que se pode aplicar sobre os pinos é (Maxim Intergrated, 2014):

$$rmsMAX = \frac{250mV}{\sqrt{2}} = 176,78 mV_{RMS} \quad (4.1)$$

4.1.1.1.1. Entradas de tensão

Três pinos de entradas single-ended¹⁶ (AVA, AVB e AVC) podem ser conectados a sensores de tensão externos. A Figura abaixo mostra os circuitos de condicionamento de sinal para o exemplo não isolado (por meio de um circuito divisor de resistência), uma vez que esse é o caso presente do kit de desenvolvimento.

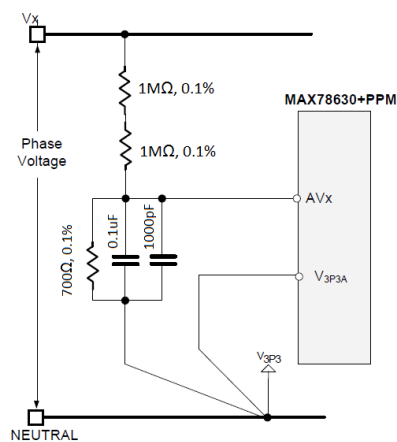


Figura 4.1.1.1.1:1: Condicionamento do sinal de tensão, por meio de um circuito divisor de resistência.

4.1.1.1.2. Entradas de corrente

Da mesma forma, três pinos de entrada single-ended (AIA, AIB e AIC) podem ser conectados a sensores de correntes externos. A figura seguinte mostra um circuito de condicionamento de sinal para os CT's fornecidos com o kit de desenvolvimento do MAX78630+PPM.

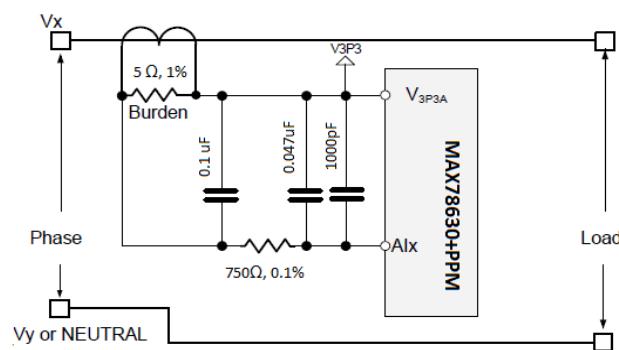


Figura 4.1.1.1.2:1: Condicionamento do sinal de tensão, para o CT CR8449-2500-N .

¹⁵ V3P3A – Pino fonte de alimentação usado no kit de desenvolvimento com tensão entre (-0.5V to 4.6V), ver Figuras 4.1.1.1.2:1 e 4.1.1.1.2:2

¹⁶ Ver Glossário de Termos, anexo A.

O kit desenvolvimento, placa e sensores incluídos (CR8449-2500-N) passam por um processo de calibração usando equipamento de medição de energia preciso. Notar que a placa é calibrada com os sensores que é enviada pela ordem de sensor I_x para CT_x , ou seja CT_1 para I_1 , CT_2 para I_2 , CT_3 para I_3 .

4.1.1.2. Conversor Delta sigma A/D

Um conversor de segunda ordem Delta-Sigma digitaliza as entradas analógicas. Os dados convertidos são então processados através de um filtro FIR (Finite Impulse Response).

4.1.1.3. Processador de medição de 24-Bits

O MAX78630 + PPM integra um processador de sinal de 24 bits de ponto fixo que executa todo o processamento de sinal digital necessário para a medição dos sinais recebidos para leitura. A funcionalidade e operação do dispositivo é determinada pela firmware e descrita no Capítulo 4.1.2.

4.1.1.4. Flash e RAM

O MAX78630 + PPM inclui 8 KB de memória flash no chip. A memória flash contém principalmente o código do programa, mas também armazena coeficientes, dados de calibragem e ajustes de configuração. O MAX78630 + PPM inclui 1,5 kb de RAM (random access memory) no chip, que contém os valores de registos de entrada e saída e é utilizada pela firmware para as suas operações.

4.1.1.5. Pinos Digitais de entrada e saída (I/O pins)

Há um total de dezasseis entradas / saídas digitais (DIOs) no dispositivo MAX78630+PPM. Alguns são dedicados à comunicação e configuração da interface série. Outros são I/O que podem ser usados como simples saídas push-pull¹⁷ sob controlo do utilizador ou encaminhado para sinais internos para propósitos especiais.

4.1.1.6. Interfaces de comunicação

O MAX78630+PPM inclui três opções para interfaces de comunicação: UART¹⁸, SPI e I2C. Uma vez que os pinos de I/O são compartilhados, apenas um modo é suportado de cada vez. A configuração de interface e endereço de pinos são amostradas ao ligar ou reiniciar para determinar qual das interfaces será ativa e para definir os endereços do dispositivo.

4.1.2. Descrição funcional e operação

Um conjunto de input (escrever), output (ler) e ler/escrever de registos são fornecidos para permitir acesso a dados para configurar o dispositivo. Os valores dos registos input (escrever) podem ser guardados na memória flash através de comandos específicos. Os valores guardados na memória flash serão carregados nestes registos ao ligar ou reiniciar por predefinição

¹⁷ Uma saída push-pull é um tipo de circuito eletrónico que usa um par de dispositivos ativos que alternadamente fornece ou absorve corrente de uma carga conectada.

¹⁸ Ver Capítulo 4.2.1.2.3.

4.1.2.1. Acesso de Registos

Todos os dados monitorizados pelo SoC são armazenados temporariamente sob a forma de Registos de 256 palavras (24-bits cada) que se encontram na sua memória, que podem ser acedidos através de interfaces de comunicação como UART, SPI ou I²C. Estes registos são endereçáveis sob a forma de bytes na porta UART e palavras nas portas SPI e I²C.

Estes consistem em registos de leitura (output), registos de escrita (input), e registos de leitura/escrita no caso do Registo de Comando¹⁹. Encontra-se uma lista em anexo com a descrição, localização e tipo de todos os registos presentes na memória flash do MAX78630+PP (Maxim Intergrated, 2014).

Tipos de Dados e formato de resultados

Os registos de entrada e saída possuem diferentes tipos de dados, dependendo da sua tarefa e função. A notação usada indica quando o número é assinalado, não-assinalado ou *bit-mapped*²⁰ e a localização da vírgula flutuante binária.

INT	Indica um inteiro com 24 bits	
S	Indica um valor de ponto fixo assinalado	S.nn
.	Indica um número de ponto fixo	
nn	Indica o número de bits à direita do ponto binário	

Tabela 4.1.2.1-1 – Tabela de descrição genérica de um registo e o seu tipo.

Exemplo1: S.21 é um número de ponto fixo assinalado, com 21 bits fracionados à direita do ponto binário, com um alcance de -4.0 a 4-2⁻²¹.

Posição do Bit	23	22	21	.	20	19	18	17	...	2	1	0
Multiplicador do Bit	(-2 ²³)	2 ²²	2 ²¹		2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	...	2 ⁻¹⁹	2 ⁻²⁰	2 ⁻²¹
Valor máximo	0	1	1		1	1	1	1	1	1	1	1
Valor mínimo	1	0	0		0	0	0	0	0	0	0	0

Tabela 4.1.2.1-2 – Tabela demonstrativa do tipo de Registo descrito no exemplo 1.

Exemplo2: S.20 é um número de ponto fixo assinalado, com 20 bits fracionados à direita do ponto binário, com um alcance de -8.0 a 8-2⁻²⁰.

Posição do Bit	23	22	21	20	.	19	18	17	...	2	1	0
Multiplicador do Bit	(-2 ²³)	2 ²²	2 ²¹	2 ²⁰		2 ⁻¹	2 ⁻²	2 ⁻³	...	2 ⁻¹⁸	2 ⁻¹⁹	2 ⁻²⁰
Valor máximo	0	1	1	1		1	1	1	1	1	1	1
Valor mínimo	1	0	0	0		0	0	0	0	0	0	0

Tabela 4.1.2.1-3 – Tabela demonstrativa do tipo de Registo descrito no exemplo 2.

¹⁹ Ver Capitulo 4.1.3

²⁰Um conjunto de bits que representa uma imagem gráfica, com cada grupo de bits a corresponder a pixel na imagem.

4.1.2.2. Rácio da Atualização de Dados

Os resultados de tensão instantânea e medição de corrente são atualizados a uma velocidade de leitura de 2700 amostras por segundo, sendo que geralmente são pouco úteis, uma vez que a essa velocidade de amostragem as leituras só podem ser acedidas através de uma interface alta velocidade, como a do protocolo SPI, outra razão está no fato que a apresentação de um detalhe de leituras tão denso e sem a suavização do sinal possui pouca utilidade. Esta forma de amostragem é denominada de “High-rate” sendo que existe um Registo, CYCLE, que incrementa cada atualização de uma amostra *high-rate* e apaga-a quando cria uma amostra denominada “low-rate”.

Uma amostra *low-rate*, consiste numa média de amostras High-rate dentro de um intervalo de acumulação. Este processo resulta na produção de um resultado low-rate, que através de um processo consecutivo de amostragem, gera uma suavização do sinal lido, aumentando a sua precisão e repetibilidade.

O intervalo de acumulação que define um resultado low-rate é definido por um número de amostras dentro de um intervalo de tempo pré-definido pelo utilizador no Registo SAMPLES, no caso deste projeto (540 amostras ou 0.2 segundos), uma vez que são os valores recomendados pelo fabricante. Resultados low-rate incluem tensões RMS e correntes RMS, frequência, potências, energia, e fator de potência.

O Registo FRAME é um contador que incrementa todos os intervalos de acumulação. Um Registo indicador de receção de dados STATUS, indica quando novos dados estão disponíveis (Maxim Intergrated, 2014).

4.1.2.3. Canais de medição

4.1.2.3.1. Tensão e Voltagem RMS

Os Valores instantâneos de tensão são carregados a uma velocidade de amostragem high-rate, enquanto os valores de tensão RMS são carregados após o intervalo de acumulação, já como valor low-rate.

Registo	Endereço byte	Descrição	Escala temporal
VA	87	Voltagem instantânea @ tempo t	high-rate
VB	8A		
VC	8D		
VA_RMS	90	Voltagem RMS do último intervalo	low-rate
VB_RMS	93		
VC_RMS	96		
VT_RMS	99	Média de VA_RMS, VB_RMS, VC_RMS	

Tabela 4.1.2.3.1-1 – Tabela de endereços, descrição e escala temporal dos registos de cálculo da tensão.

O MAX78630+PPM reporta as medições RMS verdadeiras para cada entrada. Um valor RMS é obtido através da soma dos quadrados dos valores instantâneos num dado intervalo de tempo (intervalo de acumulação) e então faz-se uma raiz quadrada do resultado depois de o dividir pelo número de amostras no intervalo.

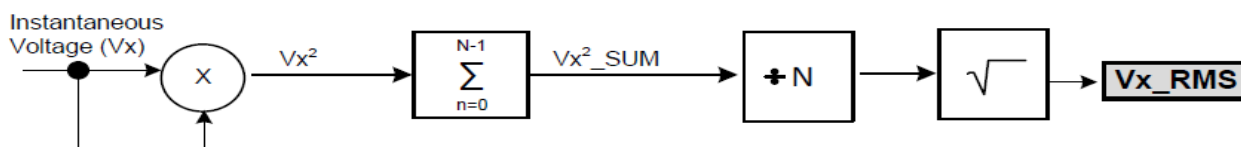


Figura 4.1.2.3.1:1: Diagrama de blocos com os passos para o cálculo das tensões RMS (eficazes).

4.1.2.3.2. Frequência da Linha

Esta saída é uma medição da frequência fundamental da fonte de tensão AC. Esta é derivada de um sinal compósito e por isso aplica-se a todas as três fases (numa única leitura por dispositivo).

Registo	Endereço byte	Descrição	Escala temporal
FREQ	180	Frequência da voltagem em CA	64 Ciclos de voltagem

Tabela 4.1.2.3.2-1 – tabelas de endereços, descrição e escala temporal dos registos da frequência da linha.

4.1.2.3.3. Corrente e corrente RMS

Os Valores instantâneos de corrente são carregados a uma velocidade de amostragem high-rate, enquanto os valores de corrente RMS são carregados após o intervalo de acumulação, já como valor low-rate.

Registo	Endereço byte	Descrição	Escala temporal
IA	87	Corrente instantânea	high-rate
IB	8A		
IC	8D		
IA_RMS	90	Corrente RMS	low-rate
IB_RMS	93		
IC_RMS	96		
IT_RMS	99	Média de IA_RMS, IB_RMS, IC_RMS	

Tabela 4.1.2.3.3-1 – tabela de endereços, descrição e escala temporal dos registos de cálculo da corrente.

O MAX78630+PPM reporta as medições RMS verdadeiras para entradas de corrente. Um valor RMS é obtido através da soma dos quadrados dos valores instantâneos de corrente num dado intervalo de tempo (intervalo de acumulação) e então faz-se uma raiz quadrada do resultado depois de o dividir pelo número de amostras no intervalo.

$I_{xRMS_OFF^2}$ – trata-se de um registo Offset²¹ usado neste caso para corrigir o valor da corrente RMS.

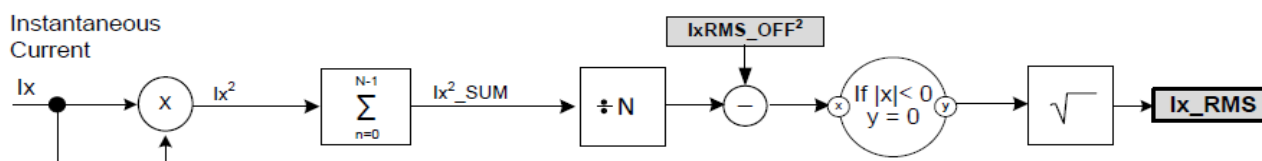


Figura 4.1.2.3.3:1: Diagrama de blocos com os passos para o cálculo das correntes RMS (eficazes).

²¹ Ver Glossário de Termos, anexo A.

4.1.2.3.4. Cálculo das Potências

Esta parte descreve detalhadamente os cálculos de fluxo de potência no MAX78630+PPM. A tabela em baixo, lista os resultados medidos disponíveis para potência em AC.

Registo	Endereço byte	Descrição	Escala temporal
WATT_A	11D	Potencia Ativa média (P)	low-rate
WATT_B	120		
WATT_C	123		
VAR_A	126	Potencia Reativa média (Q)	
VAR_B	129		
VAR_C	12C		
VA_A	12F	Potencia Aparente média (S)	
VA_B	132		
VA_C	135		
PF_A	165	Fator de Potência	
PF_B	168		
PF_C	16B		

Tabela 4.1.2.3.4-1 – tabela de endereços, descrição e escala temporal dos registos de cálculo das potências ativa, reativa, aparente e fator de potência.

4.1.2.3.4.1. Potencia Ativa por Fase

Os resultados da potência instantânea (PA,PB,PC) são obtidos multiplicando amostras de corrente e voltagem instantâneas. É então calculada a média da soma destes resultados dentro do intervalo de acumulação para computar a potência ativa média da respetiva fase (WATT_A,WATT_B,WATT_C).

Px_OFFS – trata-se de um registo Offset usado neste caso para corrigir o valor da potência ativa.

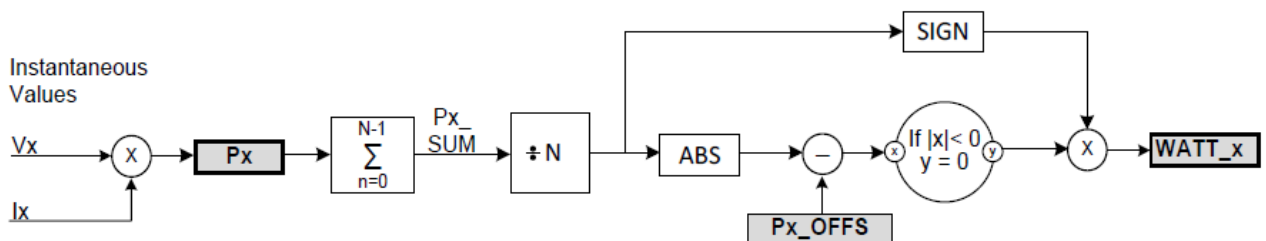


Figura 4.1.2.3.4.1.1: Diagrama de blocos com os passos para o cálculo das potências ativas.

4.1.2.3.4.2. Potencia Reativa por fase

Resultados instantâneos de potência reativa são calculados multiplicando-se as amostras instantâneas de corrente instantânea e a quadratura instantânea da tensão²². É então calculada a média da soma destes resultados dentro de um tempo de acumulação, para computar a potência reativa média (VAR_A, VAR_B, VAR_C).

Qx_OFFS – trata-se de um registo Offset usado neste caso para corrigir o valor da potência reativa.

²² Ver Capitulo 2.3.3.

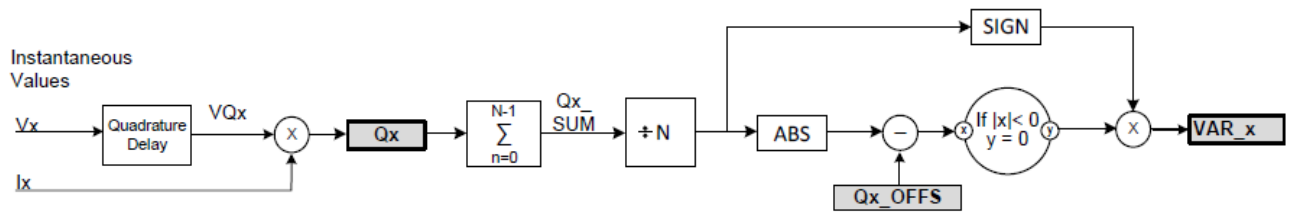


Figura 4.1.2.3.4.2:1: Diagrama de blocos com os passos para o cálculo das potências reativas.

4.1.2.3.4.3. Potência Aparente por Fase

A potência aparente é o produto dos dados low-rate da tensão RMS e corrente RMS. Offsets aplicados à corrente RMS corrente vão afetar os resultados de potência aparente.



Figura 4.1.2.3.4.3:1: Diagrama de blocos com os passos para o cálculo das potências Aparentes.

4.1.2.3.4.4. Fator de Potência

OS Registos do fator de potência capturam o rácio de potência ativa para potência reativa para o intervalo de acumulação mais recente. O sinal do fator de potência é determinado pelo sinal da potência ativa.

$$PF_A = \frac{WATT_A}{VA_A} \quad (4.1)$$

$$PF_B = \frac{WATT_B}{VA_B} \quad (4.2)$$

$$PF_C = \frac{WATT_C}{VA_C} \quad (4.3)$$

4.1.2.3.5. Potência Ativa, Reativa, Aparente e Fator de Potência Total

Os resultados das potências totais no sistema trifásico dependem de como a fonte AC, a carga e os sensores se encontram configurados. Para o caso deste projeto trata-se de uma ligação de corrente alternada conectada em estrela ou sistema trifásico *3V Delta-Connected*, onde a voltagem de todas as 3 fases (linha-neutro) são medidas. A linha neutro é a referência para todas as medidas de tensão.

Assim para este caso a forma de medir respetivamente a potência ativa, reativa e aparente será:

$$WATT_T = (WATT_A + WATT_B + WATT_C) \quad (4.4)$$

$$VAR_T = (VAR_A + VAR_B + VAR_C) \quad (4.5)$$

$$VA_T = (VA_A + VA_B + VA_C) \quad (4.6)$$

Estes cálculos não serão efetuados pelo MAX78630+PPM, uma vez que não existe necessidade se sobrecarregar o data logger com dados que são meras somas aritméticas. Assim os cálculos das potências totais será feita no código PHP que estará alojado no Website deste projeto, e assim acessível ao utilizador.

4.1.2.3.6. Cálculo da Energia

Os cálculos de energia estão incluídos no MAX78630 + PPM para minimizar o tráfego na interface com o host e simplificar o design do sistema. Resultados Low rate de medição de potência são multiplicados pelo número de amostras (Registo DIVISOR) para calcular a energia no último intervalo de acumulação. Os resultados de Energia são somados até atingirem uma dimensão "recipiente" definido pelo utilizador. Para cada recipiente de energia que é atingido, o valor nos registos de contagem é aumentado em um.

Todos os registos de contagem de energia são registos de saída de 24-bit low rate que contêm valores calculados ao longo de múltiplos intervalos de acumulação. Ambos os resultados de importação (positivo) e exportação (negativos) são fornecidos para a contagem de energia ativa e reativa.

Os resultados de energia são apagados após qualquer desativação ou reiniciação do dispositivo sendo que também podem ser limpos manualmente através do controlador (host) externo usando o bit REN no Registo COMMAND. Os registos CYCLES podem ser usados para detetar apagões no sistema (perdas dos dados de energia) e para saber o tempo entre as leituras de energia (Maxim Intergrated, 2014).

Registo	Endereço byte	Descrição	Escala temporal
WHA_POS	1DD	Contador de energia Ativa positiva por fase	low-rate
WHB_POS	1EF		
WHC_POS	201		
WHA_NEG	1E6	Contador de energia Ativa negativa por fase	
WHB_NEG	1F8		
WHC_NEG	20A		
VARHA_POS	213	Contador de energia Reativa positiva por fase	
VARHB_POS	225		
VARHC_POS	237		
VARHA_NEG	21C	Contador de energia Reativa negativa por fase	
VARHB_NEG	22E		
VARHC_NEG	240		

Tabela 4.1.2.3.6-1 – tabela de endereços, descrição e escala temporal dos registos de cálculo das energias ativa e reativa, positivas e negativas.

Tamanho dos “pacotes” de energia para os contadores de energia

O Registo BUCKET permite ao utilizador definir a unidade de medição para os registos de contagem. Este é um número não assinalado de ponto fixo de 48-bit com 24 bits de inteiro e 24 bits de parte fracionária.

Posição Bit	High Word						Low Word						
	23	22	...	2	1	0	.	23	22	21	...	1	0
Valor	2^{23}	2^{22}	...	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	...	3^{-23}	2^{-24}

A dimensão do pacote do Registo deve ser dimensionado tendo em conta o grau de precisão de medição pretendido, contudo é necessário ter em conta a ordem de grandeza das medições esperadas. Por forma a definir um pacote que impeça os acumuladores e contadores de encherem demasiado depressa. Para aumentar os contadores de energia para watt-hora, por exemplo, o valor do pacote de ser igual ao número de segundos numa hora (3600) multiplicado pela taxa de amostragem (2700 por segundo) e dividido pelo watts máximo da escala total (VSCALE x ISCALE).

$$\text{Watt - hora (Wh) "pacote"} = \frac{3600 \times 2700}{\text{VSCALE} \times \text{ISCALE}} \quad (4.7)$$

Registo	Endereço byte	Descrição	Escala temporal
VSCALE	144	Leitura limite do valor de tensão	Definida pelo "pacote"
IFSCALE	141	Leitura limite do valor de corrente	Definida pelo "pacote"

Tabela 4.1.2.3.6-2 – tabela de endereços, dos registos VSCALE e IFSCALE para o cálculo dos "pacotes" de energia.

Outra forma de atribuir um valor a este pacote é usando o *software* GUI.

No caso Desta dissertação os pacotes de energia para os contadores podem ser definidos facilmente no *software* Gui na secção de calibração. Uma vez que o equipamento se encontra em fase de testes, foi escolhido um pacote de um Wh para maior rigor nas medições, algumas de curta duração.

4.1.3. Fonte conectada em estrela, Carga conectada em estrela (Y-Y)

Como já pode ser estudado no capítulo 2.6.4 existem formas distintas de fornecer e alimentar correntes trifásicas, sendo que o MAX78630+PPM está preparado para 3 destas topologias:

1. Fonte conectada em estrela (Y), carga conectada em estrela (Y).
2. Fonte conectada em triângulo (Δ), carga conectada em triângulo (Δ).
3. Fonte conectada em estrela (Y), carga conectada em triângulo (Δ).

A que se irá configurar será a primeira por ser a topologia mais usada e apropriada para habitações tal como já foi mencionado no Capítulo 2.6.4.

Para sistemas conectados em estrela, todas as 3 fases (linha-neutro) de tensão são medidas. A linha neutra é a referência para todas as fontes de tensão.

Assim os passos para configurar a placa às especificações pretendidas é:

1. Colocar Jumper J11 na posição 2-3
2. Colocar Jumper J15 na posição 1-2
3. Conectar o cabo neutro da fonte para a carga e para o pino J4 (NIN)
4. Conectar a fonte AC: fase a para J1 fase B para J2 e fase C para J3.
5. Conectar o sensor de corrente CT1 à ligação J39, e fechar os jumpers J13 e J14.
6. Conectar o sensor de corrente CT2 à ligação J40, e fechar os jumpers J19 e J20.
7. Conectar o sensor de corrente CT3 à ligação J41, e fechar os jumpers J42 e J43.

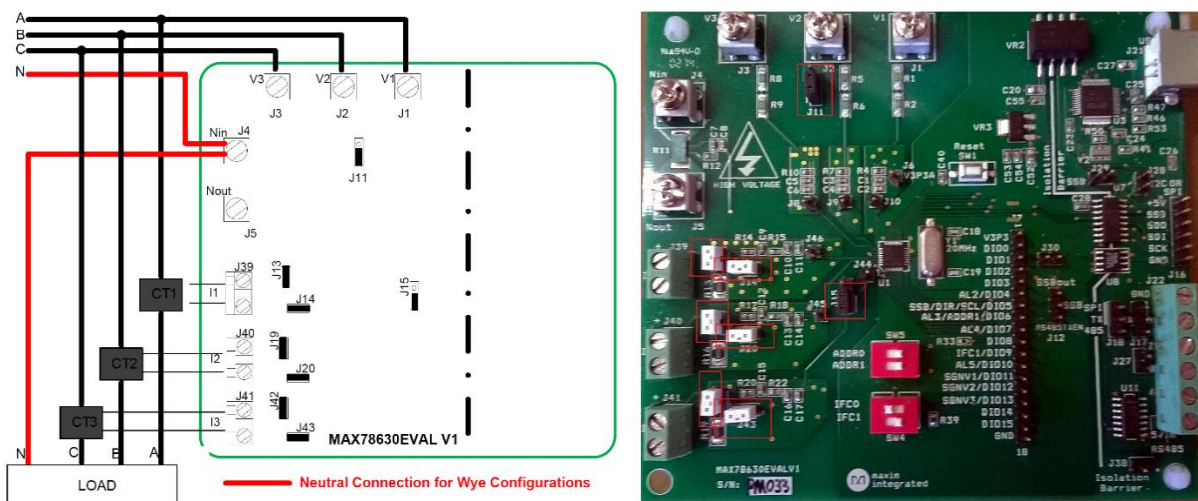


Figura 4.1.3.1: Conexões do kit de desenvolvimento MAX78630+PPM para medição de carga com topologia (Y-Y).

3.1.1. Comunicação

Todos os registos do utilizador possuem um tamanho de 156 palavras (de 24bits -3 bytes) estão contidos numa área na memória RAM e podem ser acedidos através das interfaces de comunicação UART, SPI ou I²C. Apesar de ser possível aceder a um único byte com algumas interfaces, é altamente recomendado que o utilizador aceda palavras ou múltiplas palavras de dados em cada transação.

Apenas uma interface pode ser ativada de cada vez. Os pinos de acesso à interface devem estar dispostos segundo a seguinte tabela para os dados poderem ser acedidos pelas respetivas interfaces.

Modo da Interfase	IFC0	IFC1
SPI	0	X (não importa)
UART	1	0
I2C	1	1

Tabela 3.1.1-1 – Pinos de ligação das interfases de comunicação individuais.

No kit de desenvolvimento essa seleção pode ser feita através dos pinos apresentados na Figura seguinte.

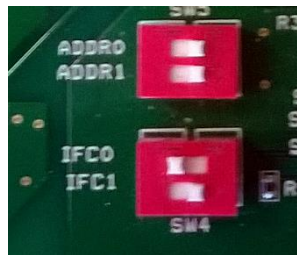


Figura 3.1.1.1: pinos de ligação das interfases de comunicação presentes no kit de desenvolvimento.

3.1.1.1. Interface UART

O dispositivo implementa um protocolo interface Serie Simples (SSI) na interfase UART que possui:

- Suporte para comunicações singulares ou de multi-pontos
- Controlo (direção) da transmissão para um transceptor RS-485
- Uso eficiente de uma interface serie de banda curta
- Verificação da integridade dos dados

A configuração padrão, e que será aplicada, é a de uma transmissão a 38400 baud, 8-bit, no-parity, 1 bit de interrupção, sem controlo de transmissão.

O protocolo SSI usado utiliza 8 bits para comunicação multi-ponto, com um alcance de 1 a 254.

Descrição: O protocolo SSI comanda o sistema de resposta suportando apenas um master e vários slaves. O host (mestre) envia o comando para um alvo escravo que verifica a integridade do pacote antes de enviar a resposta ou comando de execução. Caso isso falhe o escravo deverá enviar um código de erro, se a condição do pacote for incerta nenhuma mensagem será enviada (Maxim Intergrated, 2014).

Pacotes Mestre

Os Pacotes mestre começam sempre com um byte cabeça (header-0xAA) por propósitos de sincronização. O mestre enviará então outro byte (*byte count*), um que indica o tamanho do pacote inteiro (até 255 bytes) seguido de uma *payload* (com tamanho até 253 bytes) e por fim um byte final conhecido como *checksum*, que verificará a integridade de todo o pacote.

Ordem de envio dos bytes ->	Header	Byte Count	PAYLOAD	Checksum
	1º byte: (0xAA)	Nº de bytes no pacote	Carga de informação que o SoC receberá	Byte que verifica a integridade do pacote

Tabela 3.1.1.1-1 – Tabela descritiva do envio de um pacote de bytes Mestre

A *payload* pode conter um comando ou vários se o alvo já se encontrar selecionado. Também pode incluir o endereço do dispositivo, endereço dos registos e o mais importante, os dados. Todos dos múltiplos bytes são enviados e recebidos com o LSB (*Least significant Byte*) a ser lido primeiro, sendo essa a ordem de leitura.

Comando	Parâmetros	Descrição
80-8F		(não usado)
A0		Apaga endereço
A1	[Byte-L]	Estabelecer bits de leitura/escrita [7:0] – comando de endereçamento
A2	[Byte-H]	Estabelecer bits de leitura/escrita [15:8] – comando de endereçamento
A3	[Byte-L][Byte-H]	Estabelecer bits de leitura/escrita [15:0] – comando de endereçamento
A4-AF		(Reservado maiores endereços alvo)
B0-BF		(não usado)
C0		Desseleccionar alvo (alvo irá reconhecer)
C1-CE		Selecionar alvo 1 a 14 (alvo irá reconhecer)
CF	[byte]	Selecionar alvo 0 a 255 (alvo irá reconhecer)
D0	[Data...]	Escrever bytes estabelecidos pelo recolector da Contagem Bytes
D1-DF	[Data...]	Escrever de 1 a 15 bytes
E0	[byte]	Ler de 0 a 255 bytes
E1-EF		Ler de 1 a 15 bytes
F0-FF		(não usado)

Tabela 3.1.1.1-2 – Sumários dos comandos dos pacotes mestre.

Os utilizadores só tem de implementar os comandos que eles necessitam usar. Por exemplo apenas um comando de endereçamento é necessário – seja 0xA1 para sistemas com endereços de 8 bits, seja 0xA3 para sistemas com endereços 16 bits.

Nesta dissertação todos os pacotes Mestre possuem 7 bytes de comprimento, com a exceção de um que possui comprimento 10. Estes são usados para pedir ao MAX78630+PPM valores de corrente e tensão RMS, potências ativa, reativa e aparente, fatores de potência, valores de frequência e de energia Ativa e Reativa, positiva e negativa, e também é enviado um comando no final de cada mês para reiniciar a leitura dos pacotes de energia. Estes comandos podem ser consultados no anexo B (Maxim Intergrated, 2014).

Pacotes Escravo

O tipo de pacote escravo depende do tipo de comando recebido do dispositivo mestre, e do sucesso da execução do comando por parte do dispositivo escravo. Respostas padrão incluem dados reconhecidos com dados “Acknowledge” e dados não reconhecidos “Acknowledge Without Data”.

Acknowledge Without Data

Ordem de envio dos bytes ->

Acknowledge With Data	BYTE COUNT	READ DATA	CHECK SUM
---------------------------------	-----------------------------	----------------------------	----------------------------

Tabela 3.1.1.1-3 – Tabela descritiva do envio de um pacote de bytes Escravos.

Se nenhuns dados forem esperados do dispositivo escravo ou existir um erro na leitura, um único byte de resposta será enviado. Se a descodificação e leitura do código forem bem-sucedidas será expectado o envio de dados por parte do dispositivo escravo, este envio será em forma de pacote, semelhante ao do mestre, onde o *header* é substituído por um código de resposta e onde a *payload* contém os dados de leitura, tal como demonstra a tabela 4.1.4.1-3.

Comando	Descrição
0xAA	Reconhece os dados recebidos
0xAB	Reconhece os dados recebidos (half duplex)
0xAD	Reconhece sem dados recebidos
0xB0	Reconhecimento negativo (NACK).
0xBC	Comando não implementado
0xBD	Checksum falhou
0xBF	Pacote de bytes demasiado longo
- fora do tempo -	Qualquer condição demasiado difícil de responder

Tabela 3.1.1.1-4 – Sumários dos comandos dos pacotes Escravo.

Nesta dissertação o *data logger* que irá ter o papel de dispositivo mestre está preparado para receber naquilo que considera informação útil, pacotes de 6 bytes de comprimento. Sendo os 2 primeiros bytes o de reconhecimento e o de contagem do número de bytes do pacote, e sendo o último byte o *checksum*. O código desenvolvido está preparado para ler os bytes 5, 4 e 3 pela mesma ordem que foram descritos, pois o primeiro byte a ser descodificado será o LSB, por forma a retirar a informação de leitura necessária para codificação no valor pretendido (Maxim Intergrated, 2014).

Exemplo:

Pacote Mestre enviado: 0xAA 0x07 0xA3 0x80 0x01 0xE3 0x48 (este pacote está a pedir ao escravo o valor da frequência do sistema, pelo que recebe a resposta seguinte).

Pacote escravo enviado: 0xAA 0x06 0x45 0xFB 0x31 0xDF (este pacote contém a informação pedida ao escravo, sendo que os bytes que se querem ler são 0x31 -> 0xFB -> 0x45).

A leitura resulta da transformação do código binário que cada byte (cada byte com 8 casas binárias) representa unido pela ordem descrita em cima.

Logo: 0x31 -> 0xFB -> 0x45 => 00110001U11111011U01000101 =110001111101101000101
=(transformando para decimal) =3275589

A leitura está completa, agora compete apenas usar um fator de conversão para transformar o valor inteiro decimal, no valor de frequência que se pretende. O que será explicado no capítulo seguinte.

3.1.1.2.USB -GUI

A ligação USB do kit de desenvolvimento do MAX 78630+PPM é usada para fazer conexão a um software controlador de leituras denominado GUI, sendo que o envio e leitura de dados pode ser feito via protocolo UART ou SPI, dependendo da escolha que o utilizador faça usando os pinos presentes na Figura 4.1.4.1.

A GUI é uma ferramenta poderosa que permite ler e configurar todos os aspetos técnicos presentes no SoC MAX 78630+PPM. Sendo que também foi importante para a criação do código de comunicação usado no Arduino UNO, com o âmbito de desenvolver um *data logger*, ver Capítulo 4.2.1.

O uso da GUI é serviu como ferramenta para o processo de desenvolvimento do *software* desta dissertação, tendo a sua compreensão auxiliada pelo uso das data sheet's do MAX78630+PPM e do seu kit de desenvolvimento, de seguida irá se mencionar alguns aspetos da sua utilização.

Ao inicializar deve-se seleccionar a porta COM respetiva e o Baud rate de comunicação. É importante que este seja estabelecido a cerca de 38400, para que a conexão seja bem-sucedida (Maxim Intergrated, 2013).

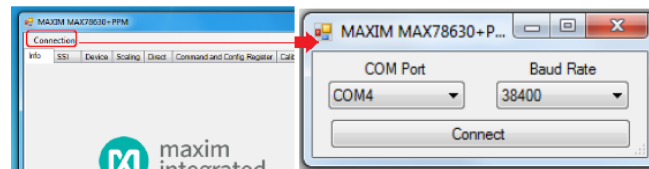


Figura 3.1.1.2:1: Conexão à GUI via porta COM e respetivo Baud rate.

Após a conexão ser bem-sucedida uma serie de separadores ficam disponíveis ao utilizador, sendo que estes são úteis para observar as capacidades do kit e configura-lo para as especificações pretendidas.

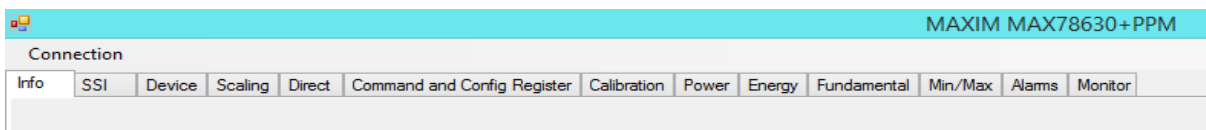


Figura 3.1.1.2:2: Apresenta os separadores disponíveis na interfase GUI

No Separador Scalling

A guia Scalling é usada para definir os fatores de escala para tensões, correntes e resultados de potência. Eles devem ser definidos de acordo com o circuito externo que está a ser usado e do processador 24 bits que os escala. Os fatores de escala são usados para converter os resultados numéricos obtidos a partir do dispositivo MAX78630+PMM para valores reais.

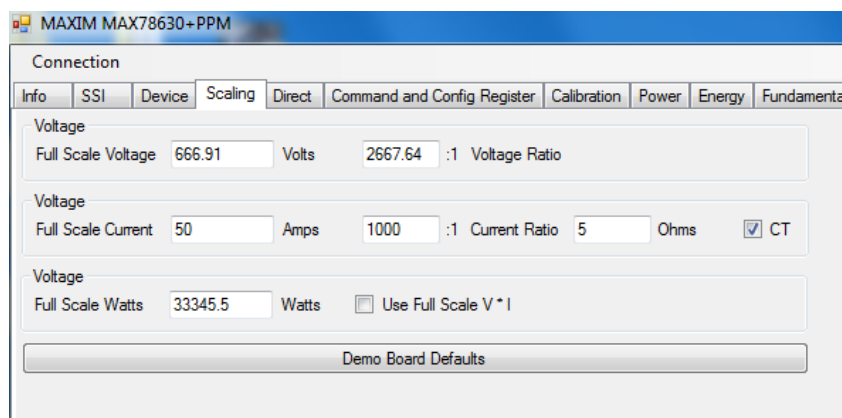


Figura 3.1.1.2:3: Imagem da interação gráfica do separador Scalling.

Os fatores escala podem ser obtidos a partir do separador Direct.

No Separador Direct

O separador Direct permite acesso a todos os registos e pode tanto ler como escrever registos. Os dados lidos dos pacotes escravos podem ser lidos já codificados em Decimal na coluna RAW DATA, onde de seguida são multiplicados pelo respetivo fator escala para se obter o valor real.

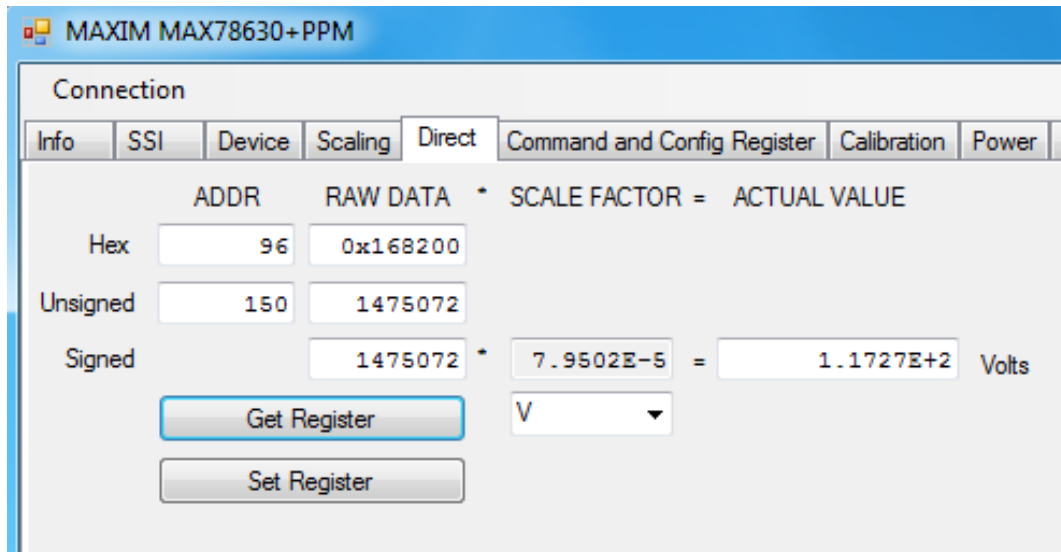


Figura 3.1.1.2:4: Imagem da interação gráfica do separador Direct, com um exemplo de cálculo de um valor de tensão.

É deste separador que se podem obter os valores de escala, sendo que este são demonstrados com baixo grau de precisão a comparar com os resultados apresentados. Então para se obter fatores de escala com maior precisão a partir da GUI, aplicou-se a equação (4.8) a diversos tipos diferentes de registos lidos, dependendo do tipo de fator de escala pretendido.

$$SCALE\ FACTOR\ (registro) = \frac{ACTUAL\ VALUE}{RAW\ DATA(registo)} \quad (4.8)$$

Os resultados apresentados na seguinte tabela:

Tipo de Registo	Fator de Escala
Corrente	0,00000596046186144157
Tensão	0,0000793945940258412
Potências	0,00396962860118015
Energias	1
Fator de Potencia	0,00000023841946872415
Frequência	0,0000152587672179127

Figura 3.1.1.2:5: resultados dos fatores de escala para o cálculo dos valores reais para cada tipo de registo.

Continuando o ultimo exemplo do capitulo 4.1.4.1., tem-se:

Resultado do valor inteiro era 3275589, sendo um valor de um registo de frequência, o fator de escala a ser aplicado é: 0,0000152587672179127

Calculando: $3275589 \times 0,0000152587672179127 = 49,9814500525554\ Hz \approx 49,91\ Hz$

O mesmo raciocínio pode ser aplicado para o cálculo de qualquer outro valor real.

No Separador Calibration

O separador Calibration é utilizado para calibrar todos os canais de medição. Os alvos de calibração podem ser visualizados e alterados e os parâmetros de escala são também apresentados por conveniência. Use as opções de calibragem para selecionar os parâmetros a serem calibrados. As calibrações e RAM não volátil são salvos através do botão fornecido.

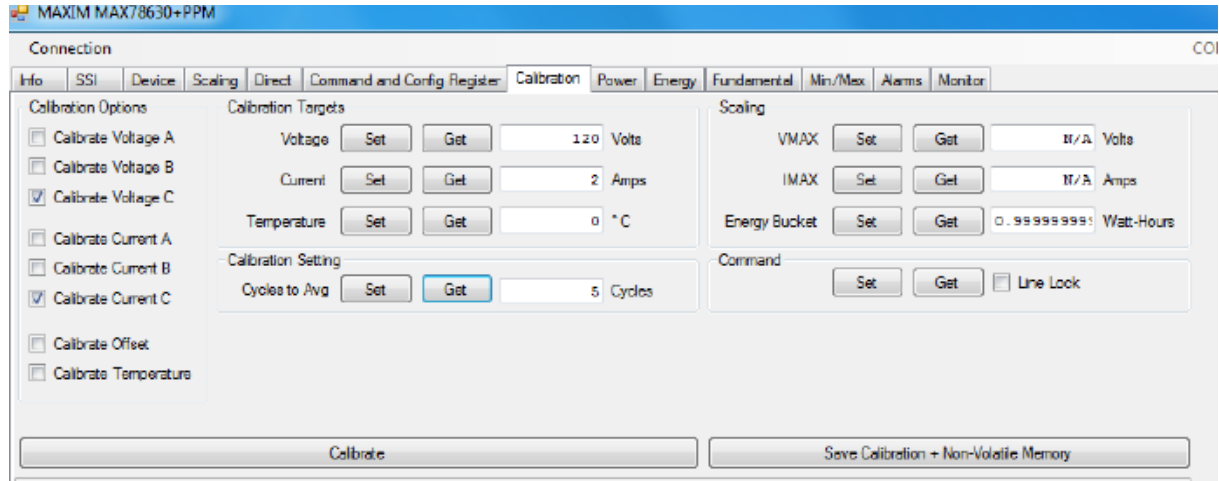


Figura 3.1.1.2:6: Imagem da interação gráfica do separador Calibration, com *line lock* desativado e o pacote de energia “bucket” definido para 1 Wh.

As únicas alterações que foram feitas na tabela de calibrações foi a desativação do *line lock*, pois cria as condições de taxa de atualização pretendidas e que se encontram descritas no capítulo 4.1.2.2., e a alteração do pacote de medição de energia para 1Wh, ou seja sempre que se consumir intervalos de 1 Wh o contador irá incrementá-los e apresentá-los.

3.2. Sistema de aquisição de dados “data logger”

Seguindo os objetivos da dissertação, que são os de criar um dispositivo capaz de fazer leituras de consumo e qualidade da energia elétrica trifásica, desenvolveu-se o dispositivo apresentado no esquema da figura 4.2.1.

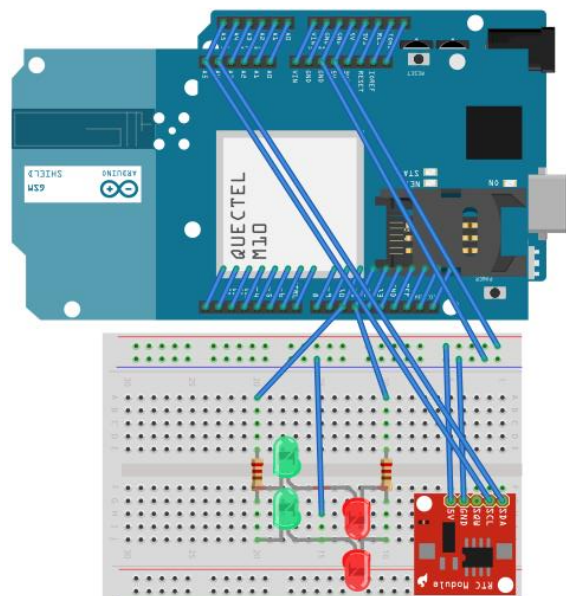


Figura 3.2:1: Sistema coletor de dados (*data-logger*) do MAX78630+PPM

O dispositivo que foi desenvolvido para funcionar como Mestre (Master) do MAX78630+PPM envolve uma série de aspetos, como o hardware utilizado na construção do protótipo, o software utilizado na sua programação e na construção da interface, além das diferentes tecnologias de comunicação entre o RTC/microcontrolador, Shield-GSM/microcontrolador, MAX78630+PPM/microcontrolador e microcontrolador/computador, ShieldGSM/Nuvem OpenShift.

Para construção do protótipo foram definidos uma série de outros aspetos como por exemplo, atingir os objetivos do mercado com maior procura, versatilidade da sua utilização e o facto de poder ser escalável. Esta escalabilidade é deveras importante, tendo em consideração a possibilidade do sistema poder facilmente evoluir por forma a atingir outros objetivos como a redução dos custos.

3.2.1. Arduino UNO R3

O Arduino UNO R3 é uma plataforma de desenvolvimento eletrónico de código aberto, que funciona em torno de um microcontrolador ATMEGA AVR de 8 bits. A placa tem capacidade para importar e exportar conteúdo, sendo a linguagem de programação utilizada denominada como *Wiring* do Arduino ou Arduino IDE (Arduino, 2014).

Apesar de existirem várias versões modelo do Arduino, todas têm como princípio comum a utilização do microcontrolador ATMEGA AVR, variando este em termos das suas capacidades de processamento.

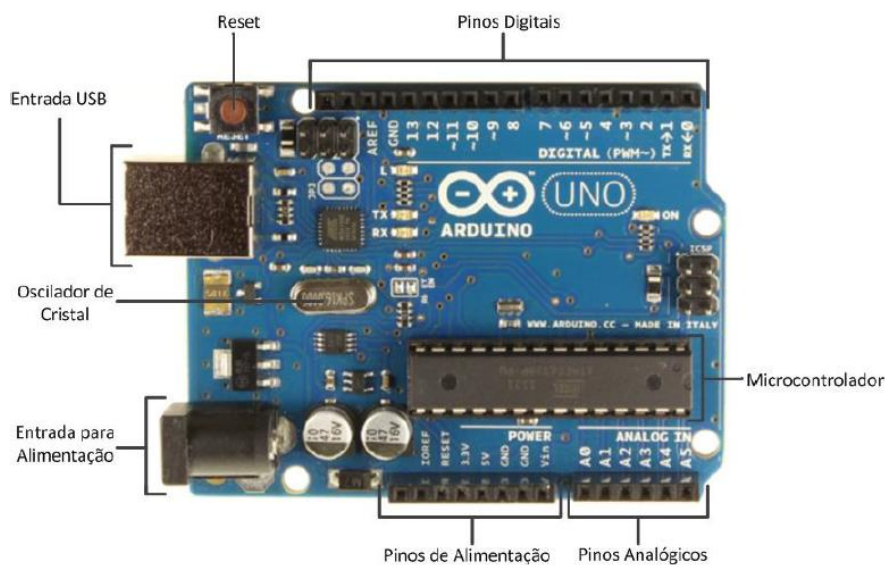


Figura 3.2.1:1: Esquema do Arduino Uno R3.

No âmbito desta dissertação, a escolha recaiu sobre o modelo Arduino Uno R3, por se ter demonstrado capaz de realizar todas as tarefas necessárias de executar por forma a atingir os objetivos desta dissertação, tendo mesmo chegado no fim perto do limite da sua capacidade, ver Capítulo 4.2.4.

3.2.1.1. Descrição funcional

A base para o funcionamento do Arduino Uno R3 é o seu microcontrolador ATmega328, que combina uma série de características como os três tipos de memórias utilizadas. Assim, no caso da memória flash existem 32 KB responsáveis pelo armazenamento dos programas carregados no Arduino a partir do Ambiente de Desenvolvimento, dos quais 0,5 KB são utilizados no *bootloader*.

Relativamente à memória SRAM (*Static Random Access Memory*) utilizada na execução e alteração dos dados presentes na memória flash, e que se assemelha à memória RAM dos computadores normais, o ATmega328 possui 2 KB.

Por sua vez este microcontrolador contém 1 KB de memória EEPROM- (*Electrically Erasable Programmable Read-Only Memory*), que permitem que os dados não sejam apagados quando o Arduino não é alimentado eletricamente.

O Arduino é capaz de fornecer energia a partir dos seus pinos a uma tensão máxima de 5V ou 3,3V, e a uma corrente máxima de 50mA no pino de 3,3V e de 40mA nos pinos de I/O e de 5V, sendo assim possível alimentar os sensores e componentes eletrónicos a ele conectados. Em relação à alimentação da própria placa esta pode ser feita de duas formas distintas, através de ligação USB ou por fonte de alimentação periférica. Existem dois locais onde a fonte de alimentação periférica se pode ligar no Arduino: o *jack*, para ligação de transformadores AC/DC com entrada fêmea compatível, ou o pino de entrada (Vin) e a terra (*Ground*) para ligação de uma bateria. Apesar de o Arduino estar apto a ser alimentado entre 6 a 20V, é recomendado pelo fabricante que seja alimentado por 7 a 12V.

Um dos aspetos essenciais à interação entre o Arduino e os sensores e componentes eletrónicos são os pinos de entrada e saída, que permitem ler e escrever dados. Assim existem na placa 14 pinos digitais, sendo que 6 deles permitem realizar PWM (*pulse width modulation*).

O Arduino Uno também possui 6 pinos analógicos ligados a um conversor analógico/digital (ADC), que possui 10bits, sendo que os pinos analógicos possuem uma entrada capaz de medir tensões entre os 0 e os 5 V. A precisão dessa medição provem da capacidade do ADC que sendo de 10 bits tem a capacidade de decompor uma tensão de intervalo de 5 volts em $2^{10}=1024$ níveis, ou seja aproximadamente 0,005 V de resolução.

Tal como qualquer outro microcontrolador, o Arduino possui um cristal com uma frequência de clock de 16 MHz, responsável pela velocidade de processamento das instruções realizadas.

Outra das características que torna o Arduino uma ferramenta tão utilizada em physical computing, é o facto de suportar várias formas de comunicação. Deste modo, o Arduino é capaz de comunicar, com computadores, microcontroladores ou sensores, através de I2C, SPI e UART (Arduino, 2014; Henriques, Ricardo, 2012).

3.2.1.2.Arduino IDE-(integrated development environment)

O Arduino uno pode ser programado com recurso a *software* de código aberto conhecido como Arduino IDE. Esta plataforma de programação permite escrever programas baseados na linguagem Wiring, sendo que se distinguem, pois o Arduino IDE é uma linguagem própria para programação de microcontroladores. Assim, quando se fala em Arduino IDE, pode-se definir como sendo a linguagem Wiring aplicada ao caso específico do Arduino. Ambos os ambientes de desenvolvimento são escritos em Java, contudo incluem uma biblioteca C/C++, também denominada Wiring, que simplifica as operações de entrada e saída no Arduino, abstraindo o utilizador do baixo nível da linguagem C/C++.

No Arduino IDE escreve-se o código, e através dos campos "*Board*" "*Serial Port*", indica-se qual o tipo de dispositivo Arduino a comunicar e a porta onde está ligado. A interfase é de uso simples e permite verificar e compilar o código desenvolvido para o Arduino recorrendo apenas a ícones presentes no monitor (Shahid, W., Dailey, N., M., 2012).

É o *bootloader* que permite o carregamento de novo código sem recurso de um compilador de código externo (Shahid, W., Dailey, N., M., 2012).

3.2.1.2.1. Código e compilação

Os programas de Arduino podem ser categorizadas em 3 partes, Estrutura, Valores e Funções. A estrutura base desta linguagem é simples, irá sempre conter pelo menos 2 partes/funções, sendo uma a função **setup ()**, que é a parte de preparação e a função **loop ()**, que é a parte de execução. Por isso o programa mais básico que se pode escrever em Arduino tem o seguinte formato:

```
void setup(){  
void loop() {}
```

Este programa já pode ser compilado para a memória flash do Arduino, marcando a sua base de funcionamento.

3.2.1.2.2. Configuração da comunicação e BaudRate

É definido usando a função **begin()** estando contida dentro da função **void setup () {}** e determina os parâmetros de comunicação serie com algum tipo de interface que o utilizador esteja a implementar, exceto no caso desta já se encontrar implementada numa biblioteca.

A função **begin()** possui dois parâmetros, a velocidade (baud) e a configuração (tamanho dos dados, paridade, bits de interrupção). Assim para implementar esta função devemos definir na função **setup ()** a forma de comunicação pretendida, seguida da função **begin (velocidade, configuração)**.

Um exemplo simples é o estabelecimento de uma comunicação com o Serial monitor, em que nesse caso o comando será: **Serial.begin (velocidade, configuração)**.

Observando o Capítulo 4.1.4.1. verifica-se que a configuração padrão do equipamento que se pretende conectar como escravo do Arduino, e que é a que será aplicada, é o de uma transmissão a 38400 baud, 8-bit, no-parity, 1 bit de interrupção.

Uma vez que serão estabelecidas, para a realização desta dissertação, 2 protocolos de comunicação serie, a do Serial monitor e a AltSoftserial, terá que se iniciar estas comunicações com os padrões de transmissão a 38400 baud, 8-bit no-parity, 1 bit de interrupção, uma vez que a configuração necessária é a configuração padrão do Arduino (Arduino, 2014).

Ter-se-á:

```
void setup()  
{  
  Serial.begin(38400);  
  altSerial.begin(38400);  
}
```

3.2.1.2.3. Interface UART, protocolo AltSoftSerial

O tipo de interface escolhida para as comunicações entre o Arduino UNO R3 e o MAX78630+PPM foi uma interface serie UART, sendo que por forma a não eliminar a possibilidade de ler informações coletadas pelo Arduino durante a utilização do equipamento, optou-se pelo uso de uma biblioteca SoftwareSerial para estabelecer uma comunicação série digital entre o Arduino e o MAX78360. Contudo após a instalação do Shield GSM Arduino, colocou-se o problema da biblioteca GSM também usar um protocolo softwareSerial para comunicar com a Shield GSM. Isto causou um problema de incompatibilidade.

Para resolver este problema de incompatibilidade, recorreu-se a um emulador de outra porta softwareSerial, que se encontra disponível através da seguinte referência (PJFR, AltSoftSerial Library). Instalando a Biblioteca e seguindo as instruções providenciadas, foi possível estabelecer 2 comunicações digitais Serie, resolvendo o problema.

3.2.2. RTC

Esta placa conhecida por RTC é um módulo capaz de comunicar o tempo real, usando um circuito integrado desenvolvido pela Maxim Integrated o DS1307. Este tem a capacidade de transmitir com elevada precisão o ano, mês, dia, hora, minutos e mesmo segundos. O módulo vem completamente montado e pré programado com o tempo atual, contudo foi necessário recalibrá-lo recorrendo a uma função de Arduino presente no anexo C. O módulo inclui também uma pequena pilha (CR1225 lithium coin cell battery) que irá permitir o funcionamento do RTC durante um período mínimo de 9 anos (17 anos tipicamente) sem o uso de uma alimentação externa de 5V (Maxim Intergrated, 2008).

O DS1307 RTC é acessível via interface de comunicação de protocolo I²C, sendo que o Arduino possui uma biblioteca **wire.h** que contem o código de acesso a este módulo, sendo apenas necessário iniciar na função **void setup ()**, configurar uma função para ler e decodificar os dados recebidos e aplicar as variáveis de medição do tempo conforme as especificações requeridas.

O módulo possui 5 canais, sendo que necessita usar apenas 4 neste caso particular, são eles:

- 5V: Fonte primária de energia. Quando a tensão é aplicada dentro de limites normais, o dispositivo torna-se completamente operacional podendo ler e escrever dados.
- GND: Terra
- SDA (Serial Data Input/Output): é a conexão de entrada/saída de dados da interfase série I²C.
- SCL (Serial Clock Line): é a entrada do relógio para a interfase I²C e é usado para sincronizar movimentação de dados na interface série.

3.2.3. Placa GSM/GPRS

A Shield GSM Arduino tem a capacidade de conectar o Arduino Uno R3 à internet usando a rede sem fios GPRS, basta inserir um cartão SIM de qualquer operadora de rede com serviço GSM/GPRS para poder estabelecer a conexão. A shield usa um rádio modem, o M10 da Quectel.

O M10 é um modem Quad-band GSM/GPRS que funciona às frequências GSM850MHz, GSM900MHz, DCS1800MHz e PCS1900MHz. Suporta os protocolos TCP/UDP e http através da conexão GPRS. A velocidade máxima de transferência de dados downlink e uplink é de 85.6 kbps.

A biblioteca GSM possui um vasto número de métodos para comunicar com a *shield*, esta usa os pinos digitais 2 e 3 para comunicar com o M10 usando um protocolo de comunicação *software serial*. O pino 2 está conectado ao pino Tx do M10 e o pino 3 ao seu pino Rx (Arduino, 2014).

3.2.4. Leds de notificação de envio de dados

Uma vez que este sistema foi desenvolvido para que seja possível ler remotamente os dados de consumo, sem nenhum tipo de monitor de energia (exceto um computador caso esteja ligado ao sistema via USB, com a visualização através do serial monitor), montou-se um sistema simples de identificação de envio de dados usando 2 saídas digitais (10 e 11) do sistema, 2 resistências de 220 Ω e 4 leds, 2 vermelhos 2 verdes.

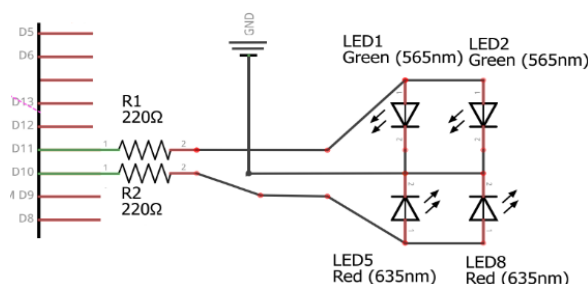


Figura 3.2.4:1: Esquema do circuito de ativação dos leds de aviso do envio de dados GSM/GPRS

A base de funcionamento é de que quando uma função de envio de dados `sendDataDb ()` é bem-sucedida, esta ativará o pino digital D11 sair para HIGH, mantendo-se ligado até que algum erro no envio dos dados ocorra. Nesse momento o pino D11 passará a LOW e o D10 passará a HIGH. Erros no envio de dados esporádicos são expectados, contudo se o sinal se mantiver vermelho durante bastante tempo, será um indicador de uma avaria de impossibilidade de enviar os dados para a Nuvem de dados *online*.

3.3. Sistema de monitorização de Energia

3.3.1. Integração do MAX78630+PPM no coletor de dados

Tal como foi anteriormente referido o MAX78630+PPM e o coletor de dados desenvolvido, usam um protocolo de comunicação UART. A ligação UART entre ambos atinge-se ligando os pinos digitais do coletor de dados Tx (pino 8) e Rx (pino 9) aos pinos ao kit de desenvolvimento do MAX78630+PPM nos pinos J17(Rx) e J18(Tx), como mostra a figura 4.3.1.1, respetivamente. Para tal é necessário remover os jumpers que ligam as cabeças nos pinos 1 e 2 tanto em J17 e J18, o que resulta na desativação da ligação USB do kit de Desenvolvimento do MAX78630+PPM.

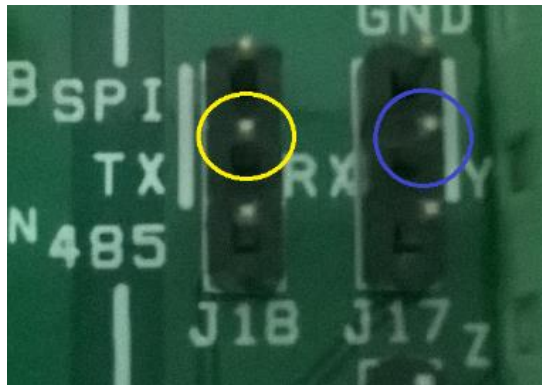


Figura 3.3.1:1: localização dos pinos de comunicação UART (Tx, Rx) no kit de desenvolvimento do MAX78630+PPM.

Para alimentar o MAX78630+PPM serão usados os pinos de alimentação de 5 volts e GND que o Arduino possui, sendo que neste caso os pinos de 5V e GND do Arduino serão ligados aos pinos de 5 V e GND do MAX78630+PPM, como mostra a figura 4.3.1.2, respetivamente.

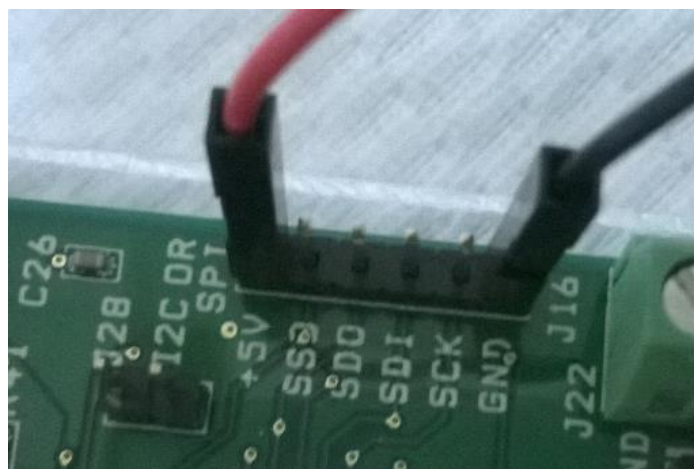


Figura 3.3.1:2: localização dos pinos de alimentação do kit de desenvolvimento MAX78630+PPM em DC.

Assim pode-se finalmente observar o produto final do sistema de monitorização de energia, composto pelo vários componentes já descritos ao longo desta dissertação.

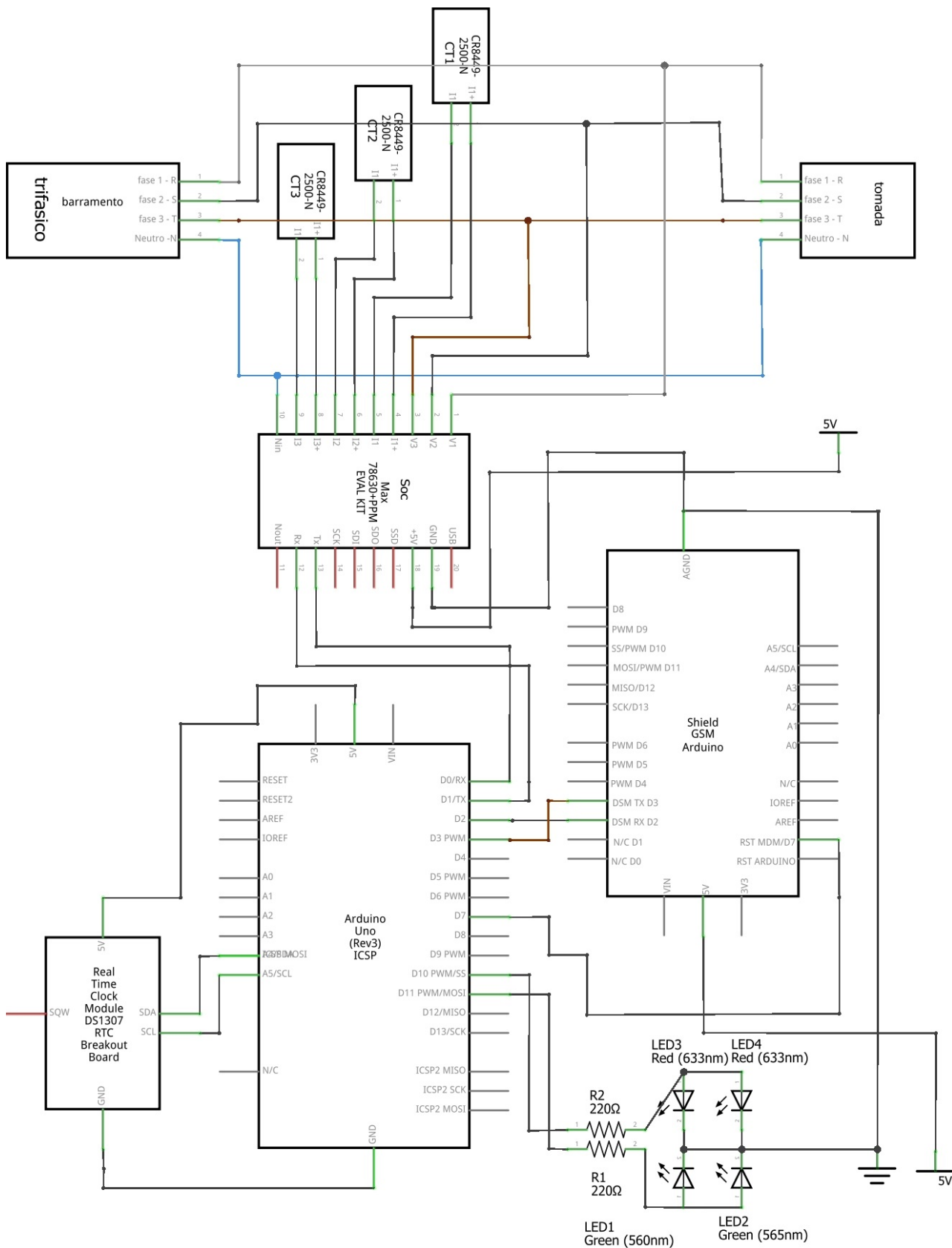


Figura 3.3.1:3:Esquema do circuito do sistema de monitorização de energia desenvolvido nesta dissertação.

3.3.2. Fluxograma do código de monitorização desenvolvido

O código desenvolvido é volumoso, sendo que ocupa 32.244 bytes dos 32.256 bytes máximos que a memória Flash do Arduino UNO R3 permite. Na figura seguinte segue um fluxograma representativo das operações realizadas por esse código.

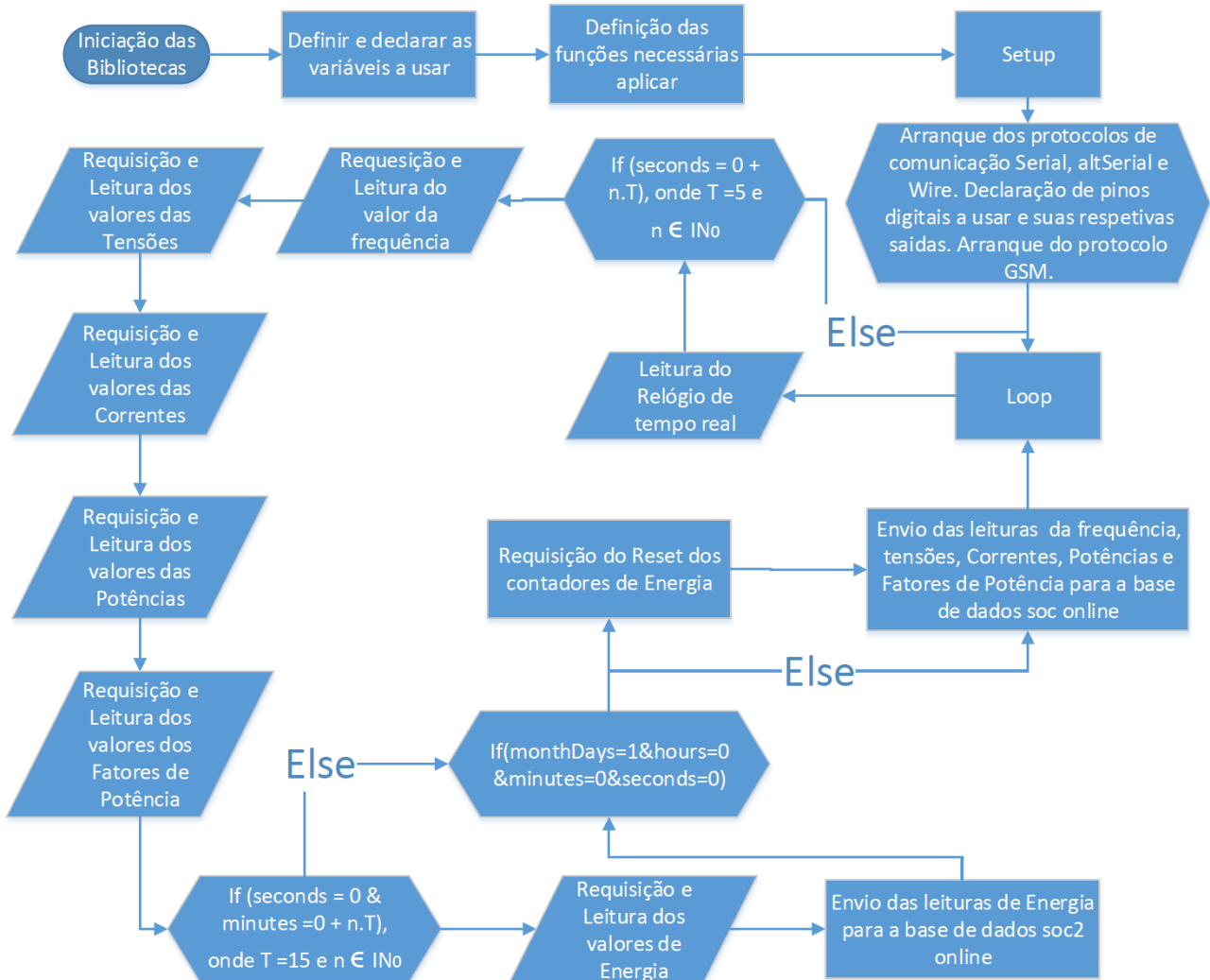


Figura 3.3.2.1: Fluxograma do código usado na programação do Arduino UNO R3

4. Nuvem OpenShift

OpenShift é uma plataforma Nuvem híbrida com um serviço tipo PaaS, provido pela empresa Red Hat. O software sobre qual corre o serviço é open source e encontra-se sob o nome OpenShift Origin, e está disponível no repositório baseado em Web hosting GitHub. Os utilizadores podem usar a Git para carregar aplicações Web em linguagem diferentes na plataforma.

OpenShift também suporta programas binários que são aplicações Web, desde que possam correr na plataforma Linux da Red Hat. Isto permite o uso de linguagem arbitrária e Frameworks²³. O OpenShift cuida da manutenção dos serviços subjacente à aplicação e dimensionamento da aplicação quando necessário.

²³ *Framework* é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação

O openShift possui 3 modelos de implementação, pública, privada e comunitária, sendo que para o âmbito desta dissertação será criada uma conta para que se possa usufruir do serviço PaaS Público.

4.1. Criar Conta e instalação das ferramentas de cliente

Acedendo ao *Website* da OpenShift, começaremos por criar uma conta do serviço PaaS público, por forma a mais tarde poder desenvolver-se a aplicação, Servidor e base de dados. O próprio *website* possui uma página de suporte que explica como usar as suas ferramentas. Contudo, em seguida serão descritos os passos básicos para atingir-se os resultados pretendidos no âmbito desta dissertação.

O endereço de *email* que será usado faz parte de um grupo da faculdade de Ciências da Universidade de Lisboa conhecido como Energia nas Escolas, sendo o endereço o seguinte: energiaescolas@fc.ul.pt.

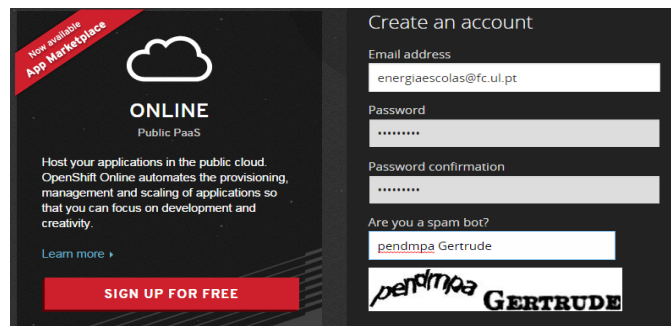


Figura 4.1.1: Plataforma *online* para realizar o registo na Nuvem OpenShift.

Feito o registo o próximo passo é estabelecer o ambiente de trabalho necessário para instalar e trabalhar as aplicações OpenShift. Deve-se então começar a trabalhar com os ambientes Web Console do OpenShift e Linha de comandos do próprio Windows.

- **OpenShift Web console**

A consola web permite criar e gerenciar as aplicações sem precisar instalar-se nada ou usar-se em alguns casos a linha de comando. A consola orienta o utilizador através da criação de cada aplicação e sugere recursos adicionais para complementá-lo. Exemplos incluem MongoDB, MySQL, ou Jenkins.

- **Linha de comandos do Windows com ferramentas (RHC)**

Com a instalação das ferramentas RHC é possível através da linha de comandos do Windows carregar aplicações, gerenciar domínios, controlar acesso às aplicações OpenShift e ter mesmo controlo completo sobre o ambiente Nuvem gerado.

Logo o próximo passo é instalar as ferramentas de cliente RHC:

1. Instalar Ruby

A versão recomendada a instalar é a Ruby 1.9.3, esta versão proporciona uma linguagem estável e uma lista extensiva de pacotes (gems) que são compatíveis e atualizadas.

Após o descarregamento do ficheiro que se encontra no seguinte endereço: <http://rubyinstaller.org/downloads/>, o utilizador deve seguir os seguintes passos: aceitar a licença, selecionar a opção “add Ruby executables to your PATH”, e finalizar a instalação do Ruby 1.9.3.

2. Instalar Git

O próximo passo é instalar o Git para Windows para que se possa sincronizar a fonte de aplicações local e sua aplicação OpenShift.

Após o descarregamento do ficheiro que se encontra no seguinte endereço: <http://msysgit.github.io/>, o utilizador deve seguir os seguintes passos: aceitar a licença, escolher a pasta onde instalar o repositório (recomenda-se uma pasta de fácil acesso ao utilizador, provavelmente organizada com outros documentos de trabalho), manter as componentes selecionadas, manter o nome para a pasta de início de menu, escolher “usar Git a partir da linha de comandos do Windows” (este passo é importante), escolher “checkout Windows-style. ...”, esperar que instale e finalizar a instalação do Git na linha de comando do Windows.

3. Instalar OpenShift gem

Depois do Ruby e Git estarem corretamente instalados, usa-se o gestor de pacotes RubyGems (incluído no Ruby) para instalar as Ferramentas de cliente do OpenShift.

Assim o próximo passo é abrir a linha de comandos (recomendado como administrador) e correr os seguintes comandos:

1. Passo: `> gem install rhc`

Após a instalação estar completa, escreve-se o novo comando seguinte:

2. Passo: `> rhc setup`

Após a correr este comando, a primeira coisa que esta vai perguntar é se se pretende continuar o processo usando o openshift.redhat.com, à qual se deve responder “sim” carregando *Enter*.

De seguida e como a figura seguinte indica, deve-se escrever o endereço com que se registou no *website* do OpenShift e *password* correspondente.

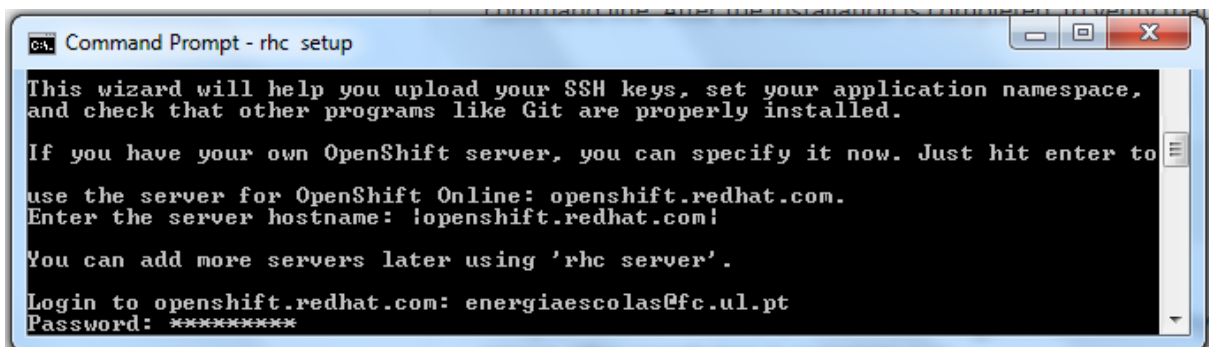


Figura 4.1:2: Uso da linha de comando para aceder à chave de acesso SSH usando o *email* e *password* do registo *online*.

Aceitando o endereço e a *password*, a janela irá perguntar se é pretendido gerar um “token”, ao que se deve responder:

3. Passo: `> yes`

De seguida a janela vai pedir algo crucial, que é o upload da chave de acesso SSH, ao que se responde:

4. Passo: `> yes`

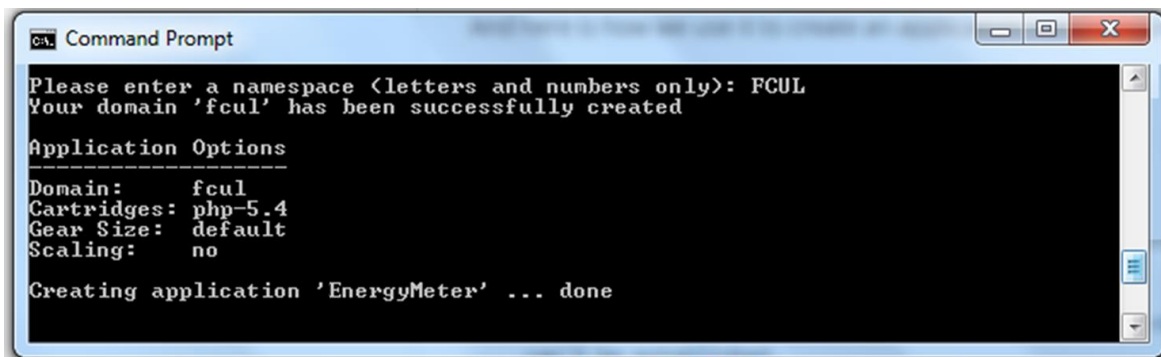
De seguida a janela vai perguntar se quer começar já a gerar um domínio, mas tendo a opção para deixar isso para outros passos mais consistentes, deverá responder com *Enter*.

E assim se conclui a instalação das ferramentas de cliente RHC, a partir de agora o utilizador tem a capacidade de criar a sua primeira aplicação no repositório OpenShift como se verá a seguir.

4.2. Criar uma aplicação

Com a instalação das ferramentas RHC completa, torna-se possível criar a primeira aplicação usando a Linha de comandos. Para efeitos de exemplo, irá se criar uma aplicação PHP 5.4. Antes de fazer um pedido de criação da aplicação, ter-se-á de navegar com a linha de comando até à diretoria onde se pretende que o código do aplicativo seja criado, neste caso será na Pasta Git criada durante a instalação deste. No final da criação da aplicação usando a linha de comando, irá clonar-se o repositório Git da aplicação para o computador, na mesma diretoria onde se executou o comando.

1. Passo: mudar para a diretoria pretendida e criar a aplicação, ver Figura: correr: `> rhc app create EnergyMeter PHP-5.4`
2. Passo: é pedido para dar um nome ao Domínio, o nome escolhido é, ver Figura: `>FCUL`
3. Passo: é perguntado se querará continuar a conexão, responder: `>yes`
4. Passo: terminada a execução do comando é criado um clone da aplicação que se criou no Openshift no repositório estabelecido no PC, ver Figura 5.2.1. Assim obtém-se uma forma fácil de colocar ficheiros criados por nós e fazer o *upload* destes para o repositório do servidor Web do OpenShift.



```
Command Prompt
Please enter a namespace (letters and numbers only): FCUL
Your domain 'fcul' has been successfully created

Application Options
-----
Domain:      fcul
Cartridges:  php-5.4
Gear Size:   default
Scaling:     no

Creating application 'EnergyMeter' ... done
```

Figura 4.2:1:Linha de comandos gerando o domínio da aplicação e a finalizar a criação da aplicação.

4.3. Instalar Base de dados MySQL

Usando agora a OpenShift Web console, irá se instalar um cartucho MySQL na aplicação criada.

O cartucho MySQL fornece MySQL no OpenShift. MySQL é uma base de dados de código aberto relacional de gestão do sistema. Com o OpenShift pode agora facilmente carregar e correr aplicações suportadas por MySQL usando os seus servidores e frameworks de escolha. O Openshift atualmente suporta MySQL 5.1 e 5.5.

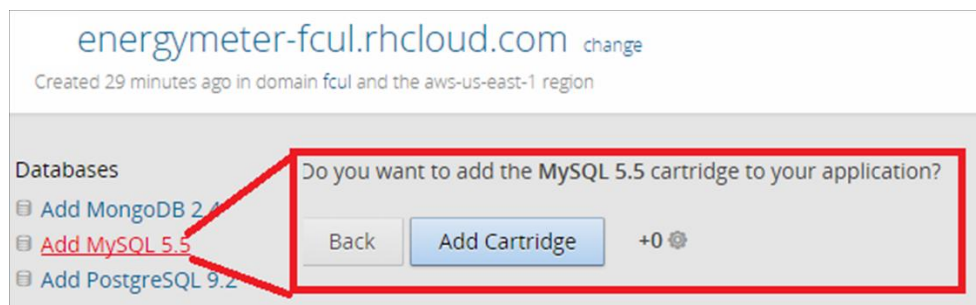


Figura 4.3:1: Uso da OpenShift Web console como meio de criar o Cartucho MySQL 5.5.

Observando a figura 5.3.1, pode ver-se que agora é possível entrar na aplicação criada e de seguida dar ordem para instalar o cartucho MySQL.

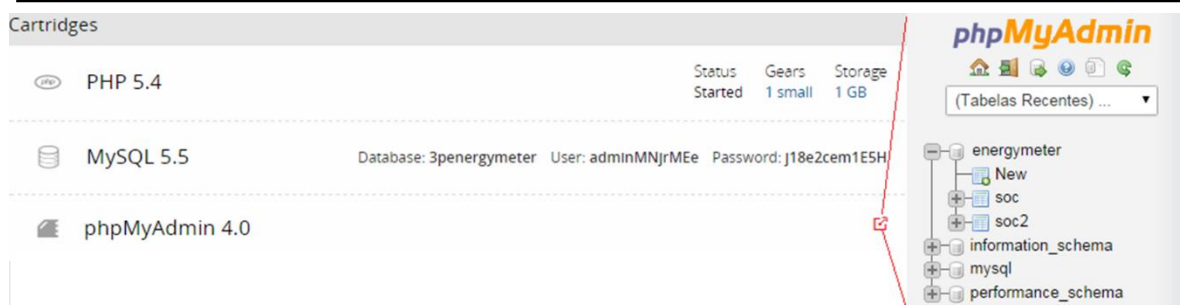


Figura 4.3.2: Uso do cartucho phpMyAdmin para aceder à base de administração de dados.

Criado o cartucho MySQL 5.5, pode-se agora proceder à instalação do cartucho phpMyAdmin para administrar a base de dados MySQL via browser, ver Figura 5.3.2.

4.4. phpMyAdmin e Sincronização com o Monitor de Energia

O phpMyAdmin é também uma ferramenta open source, escrito em PHP, que tenciona lidar com a administração do MySQL com recurso a um Browser web. Pode realizar diversas tarefas, tais como criar, modificar ou apagar bases de dados, tabelas, campos ou colunas, executando ordens SQL, ou gerir utilizadores e permissões.

Para aceder ao Browser phpMyAdmin basta carregar no ícone sinalizado na Figura 5.3.2, sendo que inicialmente o User e Password de acesso são os que se encontram apresentados na mesma Figura 5.3.2 de frente para o cartucho MySQL, Esta Palavra passe e nome de utilizador poderá posteriormente ser alterada.

Já no Browser phpMyAdmin é que se irá escolher ou criar a base de dados que se querará usar, neste caso o “energymeter” e suas subseqüentes tabelas de dados, soc e soc2.

Estas tabelas contêm variáveis que mediante da criação de 3 ficheiros PHP db_config.php, update.php e update_2.php se encontram sincronizadas com as variáveis lidas e enviadas pelo código Arduino do Sistema de monitorização de energia desenvolvido. Ver anexo D.

Para fazer o upload destes ficheiros de sincronização e dos restantes ficheiros desenvolvidos para a construção de um Website alojado neste Servidor, colocamo-los dentro da pasta criada que sincroniza com o OpenShift, e voltando à linha de comandos introduzimos os seguintes comandos:

1. `> git add .`
2. `> git commit -m "A checkin to my application"`

O commando “add” diz à Git que os ficheiros iram se tornar parte de uma verificação maior, e “commit” completa a verificação. A mensagem em frente do segundo comando é necessária para descrever a operação que se irá realizar.

Finalmente, pode-se enviar os ficheiros ou mudanças de ficheiros para a aplicação, este empurrão será feito pelo comando:

3. `>git push`

Feitos estes passos, tanto o website como o phpMyAdmin estão prontos para receber dados do sistema de monitorização desenvolvido no âmbito desta tese. A tabela soc receberá dados de 5 em 5 segundos e a tabela soc2 de 15 em 15 minutos.

A seguinte figura trata-se de uma fluxograma que resume em passos a forma como as diferentes funcionalidades deste sistema interagem, permitindo ao utilizador aceder ao conteúdo pretendido.

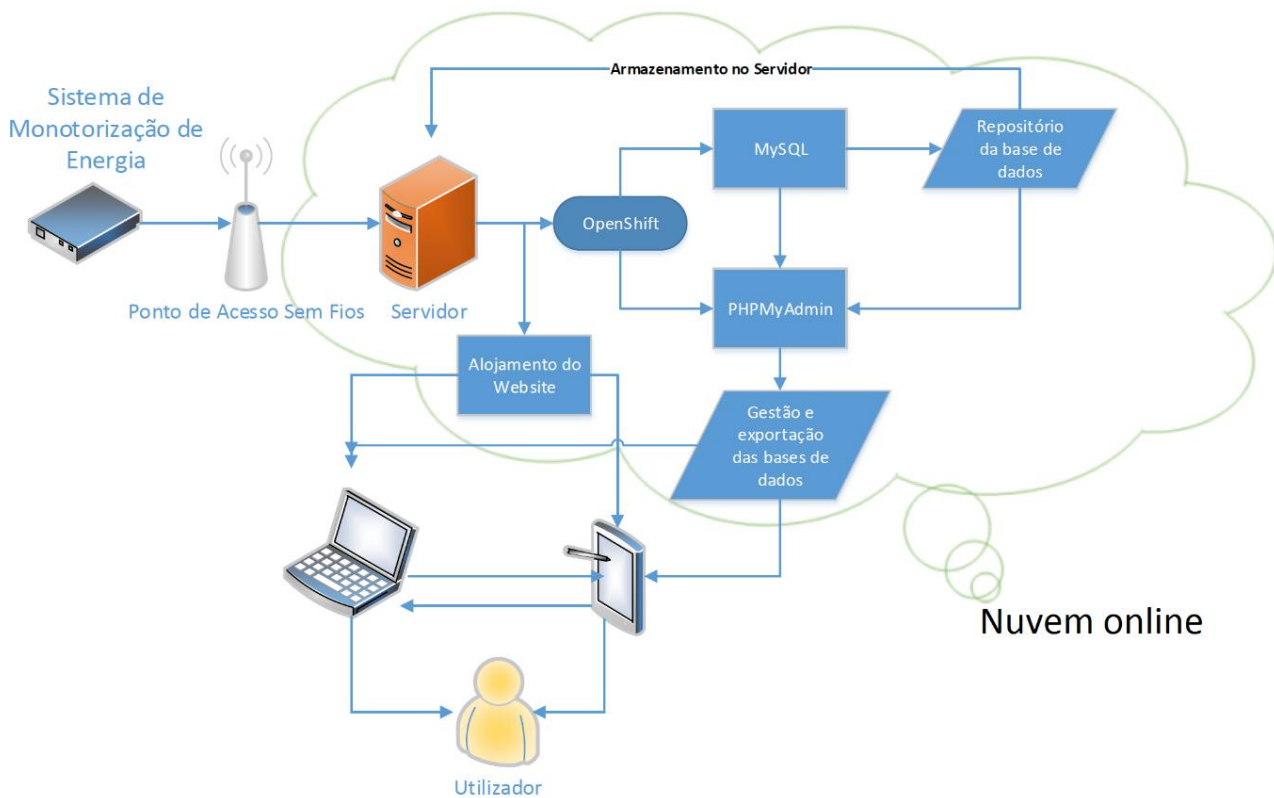


Figura 4.4:1: Fluxograma da estrutura da interface de comunicação Equipamento/Utilizador Desenvolvido.

4.5. Formato do Website, gestão e consulta dos dados

Criada a página Web Criada com o endereço, energymeter-fcul.rhcloud.com, qualquer utilizador pode aceder de forma fácil aos dados de monitorização de energia a serem lidos pelos Sistema de monitorização de energia que foi desenvolvido nesta dissertação.

O *website* é possui várias páginas web programadas em php, os ficheiros estão ligados à base de dados e tem capacidade de apresentar os dados de uma forma dinâmica recorrendo ao uso de gráficos FusionCharts que estão contidos em bibliotecas API de JavaScript. A Fusioncharts é uma empresa que oferece soluções aplicações gráficas para web developers que queiram concretizar projetos semelhantes.

O Website possui uma página introdutória para guiar os utilizadores no seu uso. Este encontra-se dividido em 5 separadores, os primeiros 3 para utilizadores mais conhecedores e as ultimas 2 de fácil leitura para qualquer utilizador.



Figura 4.5:1: Parte da janela da página introdutória ao Website de monitorização, com os separadores à vista do utilizador.

No Separador **Consumo elétrico**, o consumidor tem a oportunidade de consultar os seus consumos de potência quer seja Ativa, Reativa ou Aparente bem como as correntes eficazes correspondentes que se encontram a ser consumidas em tempo Real (5 em 5 segundos). É uma forma do consumidor analisar o comportamento do seu consumo.

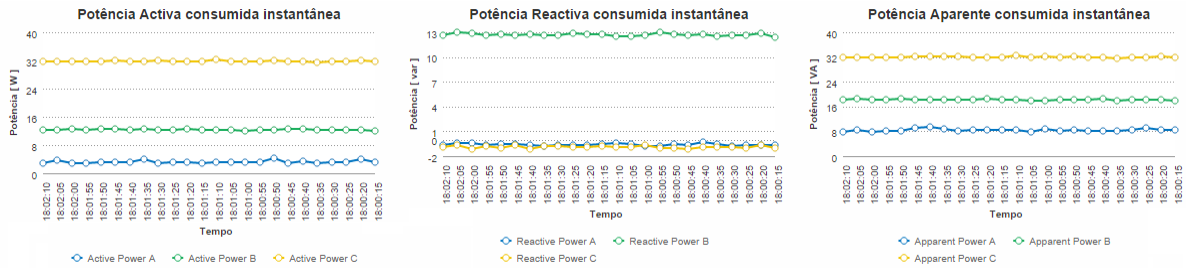


Figura 4.5:2: Parte da janela do Separador “Consumo Elétrico”, com os gráficos da potência em tempo real.

No Separador **Energia**, o consumidor tem oportunidade de verificar a evolução do seu consumo por fase e total, quer seja esta energia ativa ou Reativa. O consumidor tem assim a oportunidade de Avaliar o comportamento dos seus gastos energéticos.

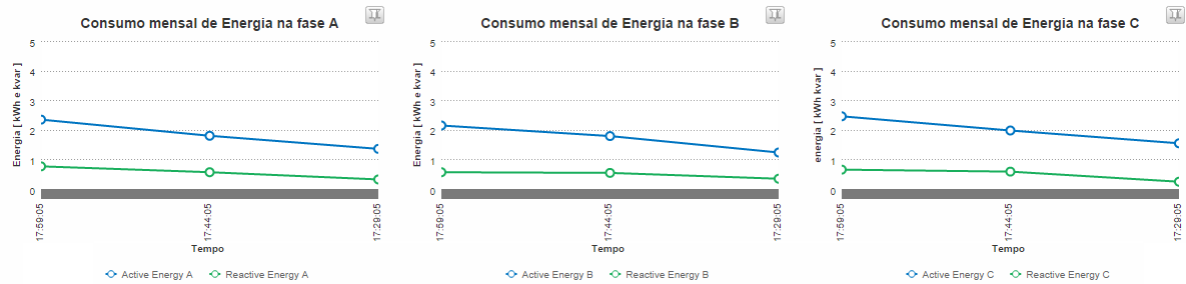


Figura 4.5:3 Parte da janela do Separador “Energia”, com os gráficos do consumo de energia de 15 em 15 minutos.

No Separador **Qualidade**, o consumidor tem a oportunidade de verificar a qualidade da sua rede elétrica ou do consumo dos seus equipamentos podendo observar a frequência elétrica da sua rede, os valores de Tensão e os Fatores de potência associados ao consumo de cada fase. As cores presentes nos gráficos são cores que ajudam o consumidor a verificar a qualidade de cada fase, sendo:

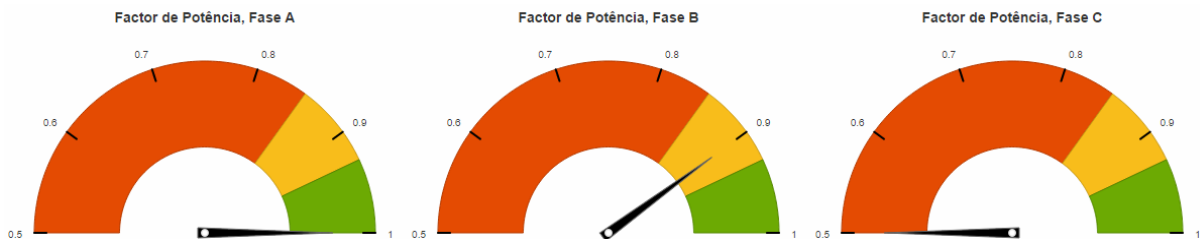


Figura 4.5:4: Parte da janela do Separador “qualidade”, onde nos gráficos o Verde representa boa, o laranja aceitável e o vermelho má qualidade.

Este índice de qualidade é estabelecido a partir de normas europeias de qualidade, não sendo necessariamente regra geral para todos os casos.

No Separador **Custos e Impactos**, o consumidor mais habitual pode verificar a evolução dos seus gastos energéticos em termos de Preço (€), CO₂ equivalente produzido e quilogramas equivalentes de petróleo consumidos em energia, desde o início da monitorização do equipamento ou desde o

início do mês caso o funcionamento esteja a ser ininterrupto. Contudo os gráficos mostram apenas os resultados das últimas 6 horas.

- Para o Cálculo do custo da Eletricidade é usada a tarifa elétrica já mencionada no Capítulo 1.5.
- Para o Calculo do CO₂ recorreu-se ao novo Decreto -Lei n.º118/2013 para obter-se os fator de conversão da energia elétrica para o seu equivalente de CO₂, sendo este de 0.114[kgCO₂/kWh].
- Para o Calculo dos kg equivalentes de petróleo recorreu-se ao decreto de lei anterior para se obter o fator de conversão de energia primária a utilizar, para a eletricidade, sendo este $F_{pu}=2,5$ [kWh_{EP}/kWh], sendo o poder calorifico do petróleo cru de aproximadamente 12,8 [kWh_{EP}/Kg], o que perfaz 0.196 [Kg_{EP}/kWh].

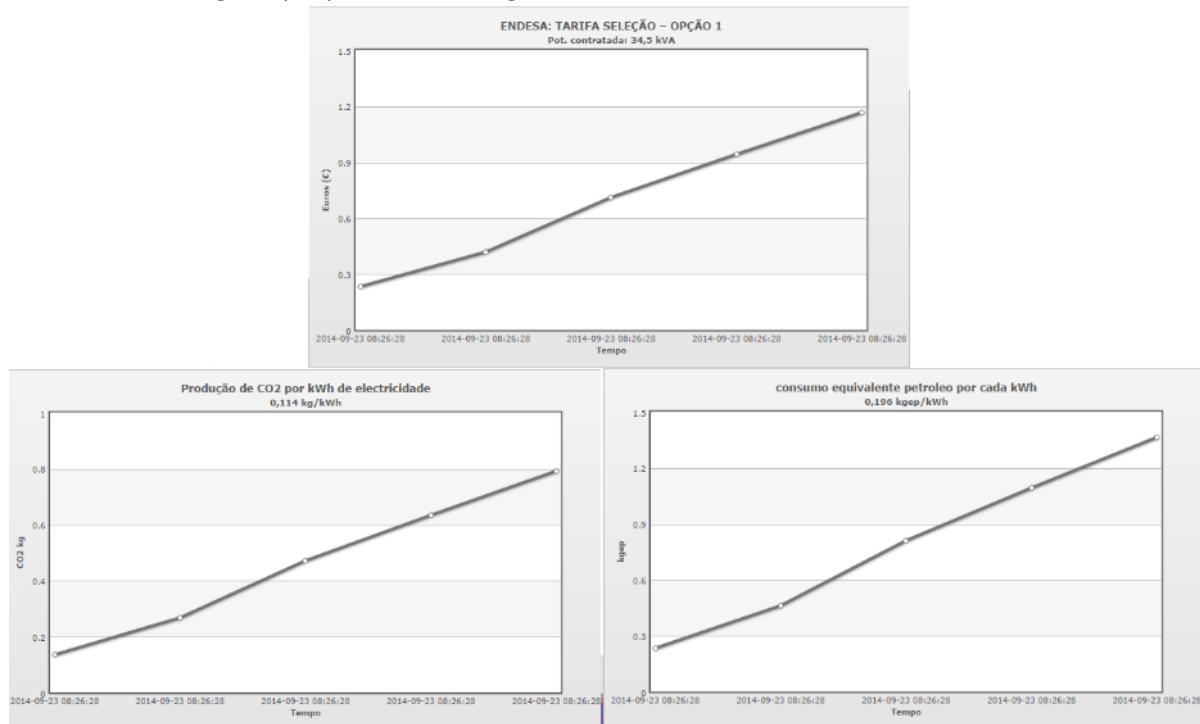


Figura 4.5.5: Graficos do Separador “Custos e Impactos”

Na etiqueta **Custos Instantâneos**, o utilizador pode ver em tempo real qual será o gasto de energia em euros nesse dia, caso o consumo naquele instante se mantenha. O Resultado é apresentado em euros e é apresentado na totalidade (soma das fases) e também dividido por cada fase. As cores ajudam o consumidor a determinar se ele esta a fazer um bom ou mau consumo naquele momento.

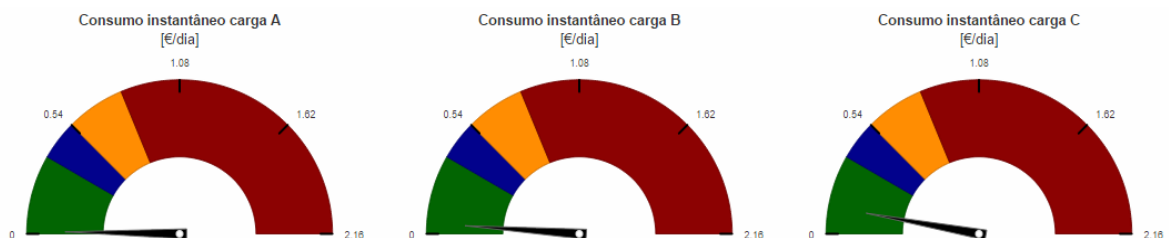


Figura 4.5.6: Parte da janela do Separador “Custos Instantâneos”, onde o consumidor pode ver os custos em tempo real.

O código de cores foi baseado em dados obtidos de um estudo realizado pela Lisboa E-Nova denominado **projeto Smartmeter 250 residências**. O sistema de cores aplicado segue a lógica dos resultados presentes no capítulo 1.3, sendo que pode ser interpretado, pela seguinte tabela:

Consumo [C- kWh]	Nível de consumo
$C < 1,08 \text{ €}$	Baixo
$1,08 \text{ €} < C < 1,62 \text{ €}$	Moderado
$1,62 \text{ €} < C < 2,43 \text{ €}$	Elevado
$2,43 \text{ €} < C$	Muito Elevado

Tabela 4.5-1 –Tabela dos indicadores de consumo na Página web Custos Instantâneos.

Outra forma útil e mais profunda de consultar os dados das medições efetuadas é através do acesso ao **phpMyAdmin**, onde o utilizador poderá importar os dados das medições realizadas sob vários formatos de ficheiros, os quais sendo:

- YAML
- Open document Spreadsheet
- CSV
- PHP array
- Code gen
- MédiaWiki table
- LaTeX
- SQL
- Taxy! Txt
- XML
- JSON
- PDF
- Microsoft Word 2000
- CSV para MS Excel
- OpenDocument Text

O formato mais vantajoso para o utilizador médio será certamente o CSV para MS Excel, tendo este sido usado várias vezes durante o curso desta dissertação para analisar em detalhe os consumos e qualidade das medições a serem realizadas, vários gráficos presentes nesta dissertação tiverem origem nesta exportação de dados.

5. Resultados experimentais

O propósito dos testes experimentais apresentados nesta dissertação, são os de validar as capacidades de monitorização de energia do protótipo desenvolvido, que será mencionado ao longo deste capítulo pelo nome Protótipo, realizando a monitorização de uma máquina trifásica e outros equipamentos monofásicos recorrendo-se ao uso tanto do protótipo como também de outros instrumentos usados como referência para comparação

5.1. Montagem da caixa para o dispositivo e dos cabos de conexão para medições trifásicas e monofásicas.

A caixa que contem o protótipo do sistema de monitorização foi feita de acrílico e de forma a comportar uma saída para uma ficha trifásica de 4 pinos, e 3 extensões de cabo HIFI para ligar aos sensores de corrente. A ficha trifásica possui cada uma das suas respetivas fases e neutro ligada aos sensores de tensão do Protótipo, respeitando sempre o mesmo código de cores para cada uma das fases e neutro, como mostra a seguinte figura.



Figura 5.1:1: montagem dos cabos elétricos na ficha trifásica usando o código de cores standard.

Assim ter-se-á a fase 1 (L1) de cor cinzenta, a fase 2 (L2) de cor preta, a fase 3 (L3) de cor castanha e o neutro (N) de cor azul. Já nas extensões dos sensores de corrente também se teve em atenção a polaridade dos sensores, sendo que a ligação que transmite a informação para a fase positiva possui uma risca vermelha que a identifica. Para finalizar foi colocado um interruptor no cabo de alimentação do dispositivo. O resultado desta montagem pode ver visto na seguinte figura.



Figura 5.1.1:2: Protótipo do dispositivo de monitorização desenvolvido na sua forma final.

5.1.1. Conexão – Cargas Trifásicas

Para ser possível registar leituras de cargas trifásicas, desenvolveu-se uma extensão trifásica recorrendo a 2 tomadas macho e fêmea de 5 pinos (4 fases + terra) de onde em ambas as extremidades existe a saída de uma ligação paralela que irá conectar ao sistema de monitorização desenvolvido e a outro sistema de motorização para os testes de comparação. Perto do fim das tomadas macho e fêmea que conectam à carga e à rede existe uma secção com os fios expostos por forma a se colocar os sensores de corrente. O resultado pode ser visto na seguinte figura.



Figura 5.1.1:1: sistema de conexão para a medição de sistemas ou cargas trifásicas

5.1.2. Conexão – Cargas Monofásicas

Reconhecido o potencial de usar-se o protótipo desenvolvido em meio doméstico, foi desenvolvida uma conexão construída à base de extensões, fichas duplas, e uma tomada trifásica. Realizada a montagem com as especificações necessárias contruiu-se uma conexão com capacidade de medir o consumo e qualidade da energia elétrica de até 3 dispositivos individuais, permitindo aprender mais sobre o seu consumo e funcionamento.



Figura 5.1.2:1: Sistema de conexão para a medição de dispositivos monofásicos.

5.2. Teste com carga Trifásica – Torno mecânico

No teste com o torno mecânico (carga trifásica) o equipamento de comparação é o PEL 103, que usa como sensores de corrente Bobinas de Rogowski modelo AmpFlex 193, com uma precisão de medição de corrente no intervalo 1 A de $\pm 5\%$ (Chauvin Arnoux, 2004). Já os documentos informativos dos sensores CR8449-2500-N não contêm especificações quanto ao seu grau de erro, sendo que estes estão calibrados com o kit de desenvolvimento para obter leituras de corrente entre os 10mA – 50A e de energia com precisão perto de 0,5%. Isto prevê antecipadamente diferenças nas medições de corrente e consequentemente nas medições que derivam desta, entre os equipamentos.



Figura 5.2:1: Torno mecânico que será a carga para a realização deste teste.



Figura 5.2:2: Montagem das ferramentas para o início do teste de comparação dos instrumentos de medição

A leitura dos resultados será efetuada a cada 5 segundos, estando os tempos de leitura de ambos os monitores sincronizados ao segundo, contudo sem garantia de sincronização exata.

Feitos os preparativos apresentados na figura 6.2.2, dá-se início ao teste às 17:52, terminando às 18:15, totalizando uma duração de 23 minutos.

5.2.1. Resultados da frequência e das tensões

Os resultados comparativos entre a medição da frequência da rede é bastante satisfatório, pois apesar de haver um ligeiro desfasamento entre as leituras, na ordem dos 0,04 Hz, existe um comportamento dinâmico semelhante entre ambos. As leituras feitas pelo Protótipo são em média mais próxima dos 50Hz espectados.

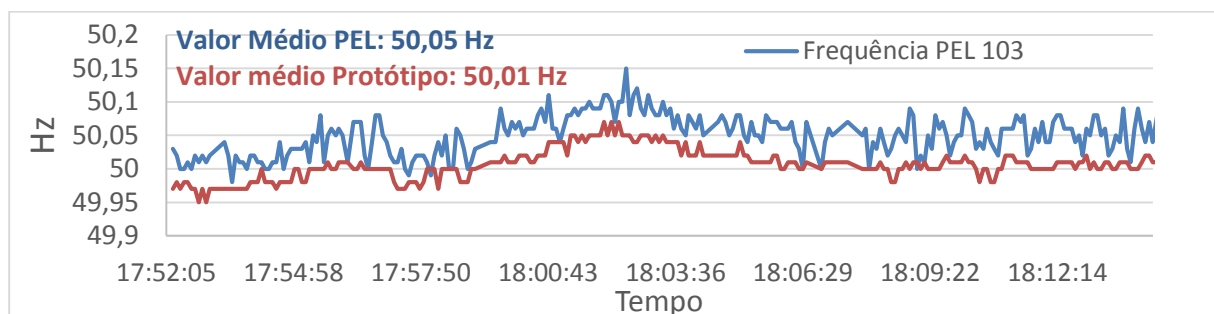


Figura 5.2.1:1: Comparação das leituras da frequência entre o Protótipo e o PEL 103

Os resultados obtidos da tensão medida em ambos os dispositivos também revela leituras muito próximas, as leituras nas fases A e B foram praticamente idênticas, sendo que apenas na fase C as leituras provenientes do Protótipo tornam-se temporariamente diferentes das medidas do PEL, contudo como é possível observar pelos gráficos que as leituras de ambos os monitores revelam um nível de proximidade muito elevado.

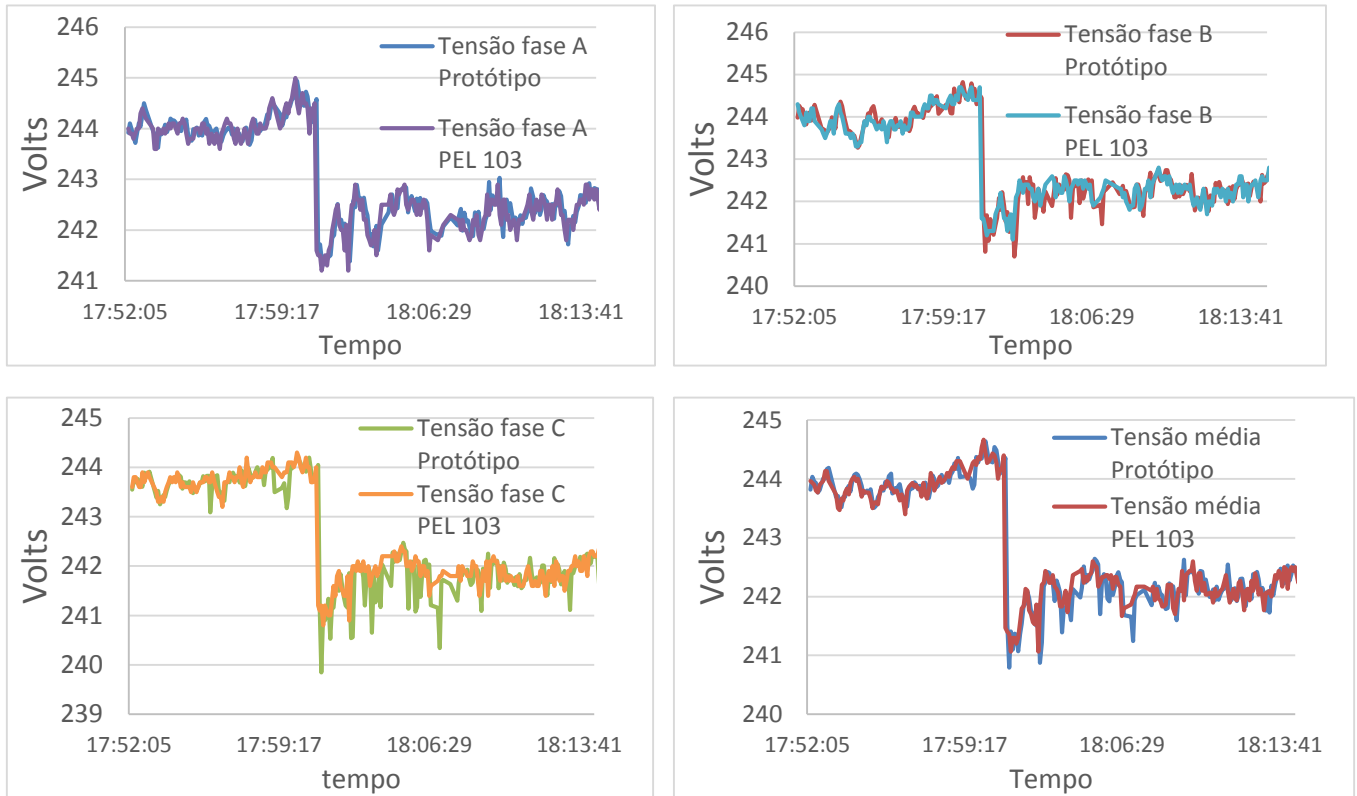


Figura 5.2.1:2: Comparação de Várias leituras de tensão lidas pelo PEL 103 e pelo Protótipo.

É possível ver que ambos os dispositivos reagiram a uma quebra de tensão súbita na rede, tendo essa descida sido registada ao mesmo tempo. Calculando o valor médio das 3 fases do Protótipo e do PEL durante o período de duração do teste, obtém-se um valor igual em ambos os monitores até à 2ª casa decimal de 242,82 V, o que é um indicador da extraordinária semelhança de recolha dos dados de tensão.

5.2.2. Resultados das correntes

No caso das correntes, os valores já não são tão similares como os anteriormente observados.

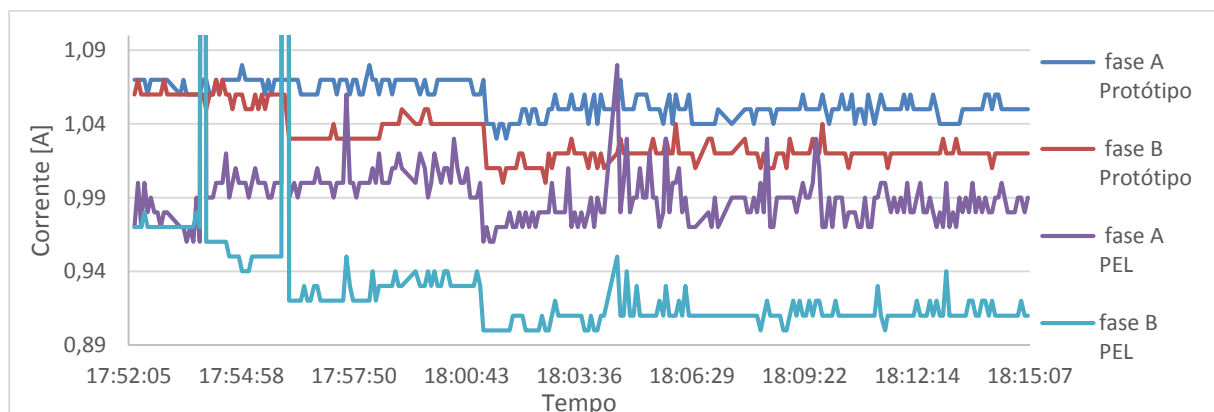


Figura 5.2.2:1 Comparação entre os valores das correntes lidas pelo Protótipo e pelo PEL nas fases A e B.

É possível observar (Fig. 5.2.2:1) que quer nas medições pelo Protótipo ou pelo PEL, tanto a fase A como a B tendem a ter leituras estáveis, contudo com uma diferença média de valor de 45mA na fase A e de 80 mA na fase B, o que referentemente aos valores da fase A e fase B lidos pelo Protótipo perfaz uma diferença de leitura de aproximadamente 4% e 8% respetivamente, o que é próximo do erro de leitura apresentado pelo fabricante dos sensores Rogowski para leituras de corrente dentro desta ordem de grandeza. Existe também um comportamento dinâmico semelhante entre as leituras, pois as suas variações possuem um elevado grau de coincidência. A apresentação dos dados pelo protótipo possui um comportamento mais suave que o do PEL (Fig. 5.2.2:1).

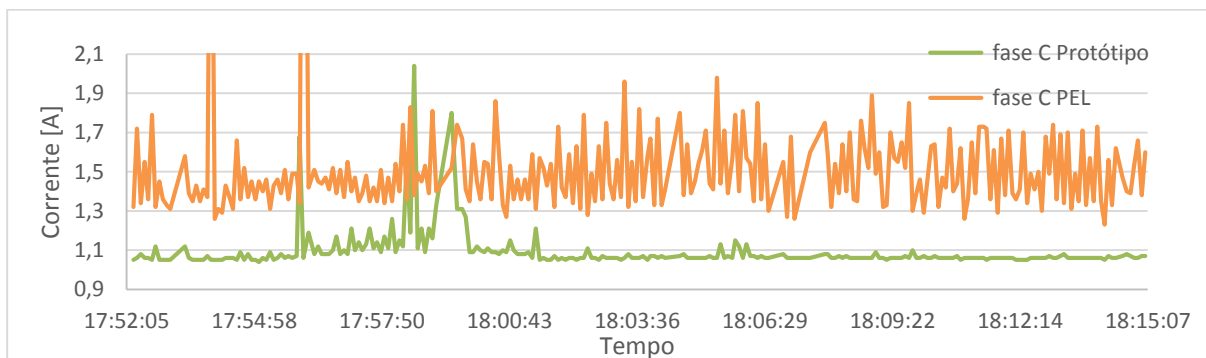


Figura 5.2.2:2: Comparação entre o valor da corrente lida pelo Protótipo e pelo PEL na fase C.

A fase C (Fig. 5.2.2:2) apresenta um comportamento anómalo quer nas medições do Protótipo quer do PEL, sendo que no Protótipo esse comportamento apenas tende a ocorrer no período entre as 17:55:59 e as 18:01:59, para de seguida tender a normalizar passando a se comportar de forma semelhante às outras fases A e B medidas pelo protótipo. Já no PEL (Fig 5.2.2:2) o comportamento é constantemente caótico, o que poderá indicar uma causa de dados mal lidos pelo sensor de corrente do PEL, uma vez que não existe nenhum comportamento dinâmico semelhante entre estas fases.

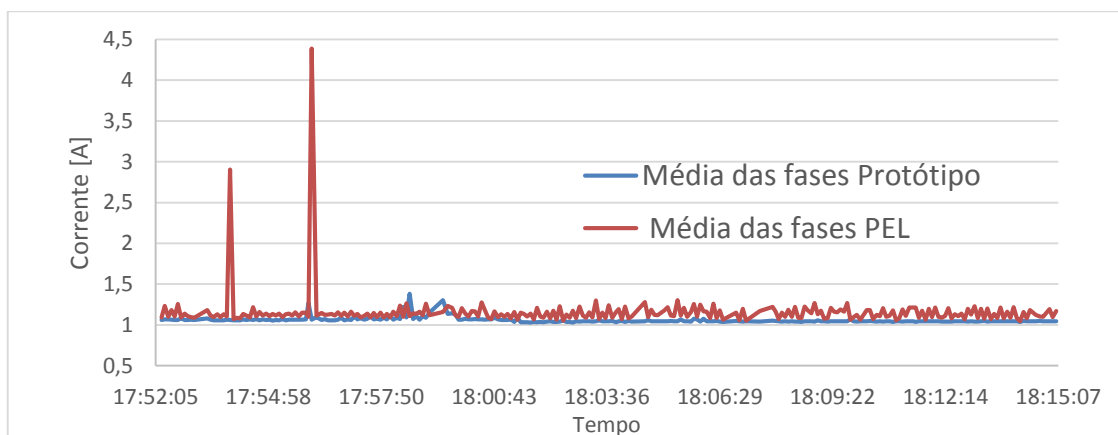


Figura 5.2.2:3 comparação da média das fases de correntes lidas pelo Protótipo e pelo PEL 103

Passando a uma observação geral das medições (Fig. 5.2.2:3) realizadas pelo protótipo e pelo PEL, deve-se mencionar a deteção de 2 leituras muito elevadas de corrente em todas as fases do PEL nos instantes 17:53:59 e 17:56:04. Esta leitura não ocorreu no protótipo e a sua causa mais provável de ser um erro na leitura dos dados por parte do PEL dadas as especificações técnicas do torno mecânico, embora é também possível que devido à amostragem ocorrer de 5 em 5 segundos, estes picos terem sido filtrados pelo Protótipo.

Apesar dos picos de correntes registados e do comportamento anómalo nas fases C, em especial a do PEL, pode-se considerar que a comparação das leituras possui um comportamento aceitável e com diferenças de grandezas de leitura justificáveis dada a sensibilidade dos sensores de corrente do PEL de $\pm 5\%$ para leituras na ordem de 1 A.

5.2.3. Resultados das potências

Já os valores de potência lidos (Fig. 5.2.3:1) apresentam diferenças significativas em termos de ordem de grandeza, contudo ainda é possível verificar aspetos comuns a ter em consideração.

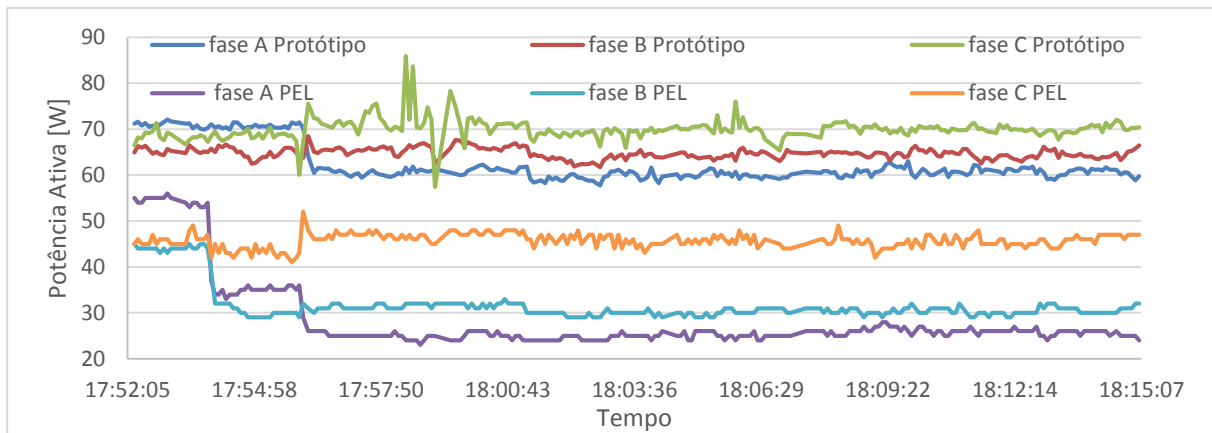


Figura 5.2.3:1:Valores de Potência Ativa em cada fase lidos pelo protótipo e pelo PEL 103.

O principal aspeto comum entre as leituras dos dois dispositivos (Fig. 5.2.3:1) é o comportamento dinâmico entre as fases com alterações semelhantes em ambos os dispositivos. É possível verificar-se que nos primeiros 4 minutos as fases com os valores mais altos em ambos os dispositivos são por ordem decrescente, começando na fase A seguida da fase C e da fase B, sendo que desse ponto em diante ocorre uma troca na ordem da sequência das grandezas já sendo a fase C com os valores mais elevados, seguida da fase B e da fase A. Isto revela aparentemente uma diferença no comportamento do consumo da carga, que é registada por ambos os dispositivos e que pode ter ocorrido devido à alteração do sentido de rotação do torno, esta alteração não foi registada no tempo mas sabe-se que ocorreu durante o decorrer do teste devido a fatores não controláveis.

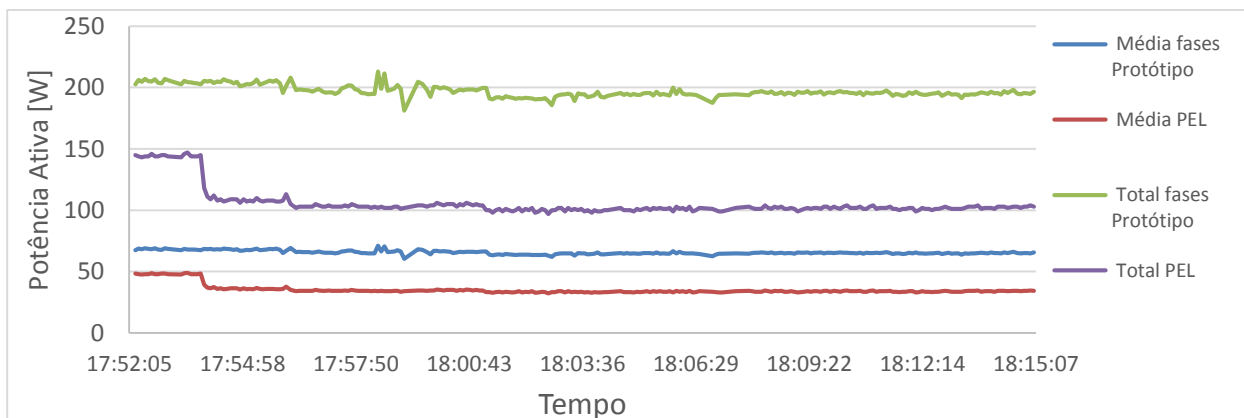


Figura 5.2.3:2: Apresentação da potência total e da média das fases lidas pelo protótipo e pelo PEL 103.

A grande diferença a ter em conta dos dados da potência ativa obtidos está na diferença da sua ordem de grandeza (Fig. 5.2.3:2), sendo que os valores de potência lidos pelo Protótipo são sensivelmente 2 vezes maiores dos que os lidos pelo PEL. Existe também um comportamento anómalo na fase C de ambos os equipamentos, sendo que no PEL os valores na fase C são bastante maiores quando comparado à diferença entre as outras 2 fases A e B. Na fase C do Protótipo é possível verificar que ainda existe um período de tempo em que o seu comportamento é mais anómalo, entre as 17:55:59 e as 18:00:59 (Fig. 5.2.3:2).

Na leitura dos dados da potência reativa o primeiro destaque vai para a leitura da fase C no PEL 103 que é completamente inconsistente com as outras leituras todas, incluído a leitura feita pela fase C do Protótipo (Fig.5.2.3:3).

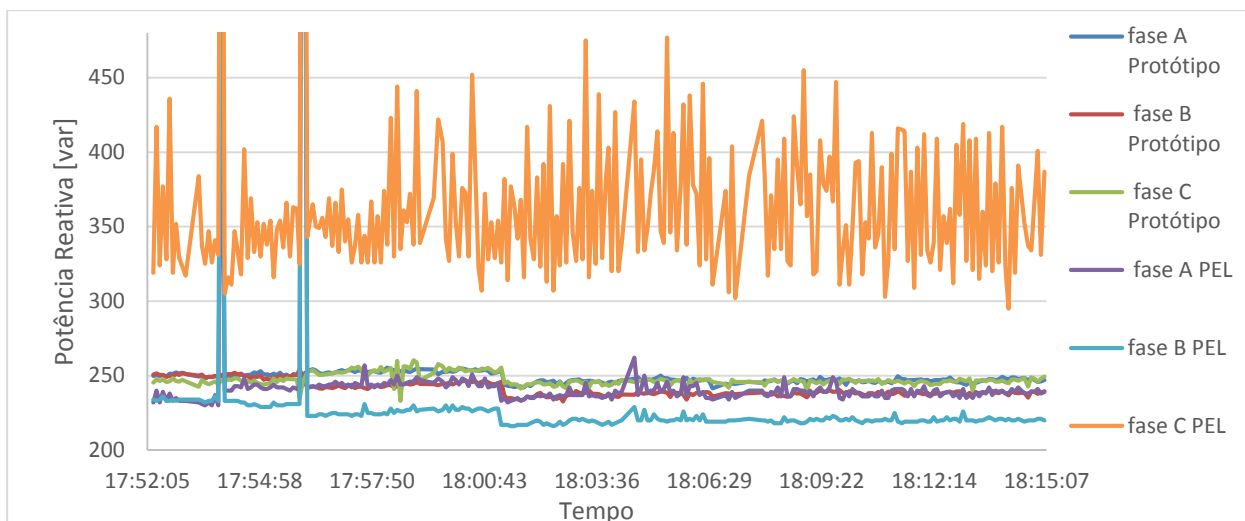


Figura 5.2.3:3: Apresentação das leituras das potências reativas, em cada fase, feitas pelo Protótipo e pelo PEL 103.

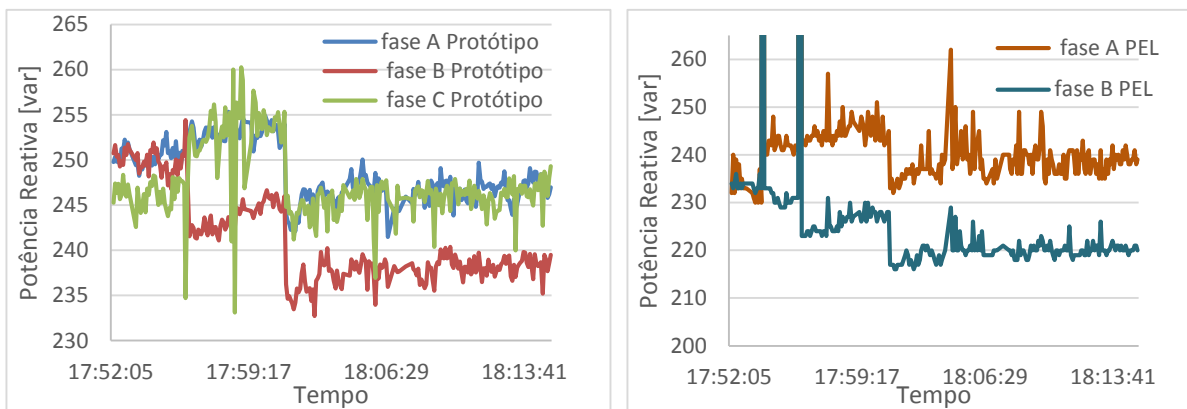


Figura 5.2.3:4: comparação entre as leituras de potência reativa realizadas no Protótipo e realizados no PEL 103.

Comparando os valores de leitura da potência reativa é possível verificar que ao contrário da potência ativa os da reativa são semelhantes em termos de ordem de grandeza. É observável também algum dinamismo coincidente entre as fases A e B de ambos os equipamentos (Fig 5.2.3:4). A fase C do Protótipo também regista valores que estão de acordo em termos de grandeza com os das restantes fases A e B, contudo continua a apresentar mais ruído no sinal (Fig. 5.2.3:4).

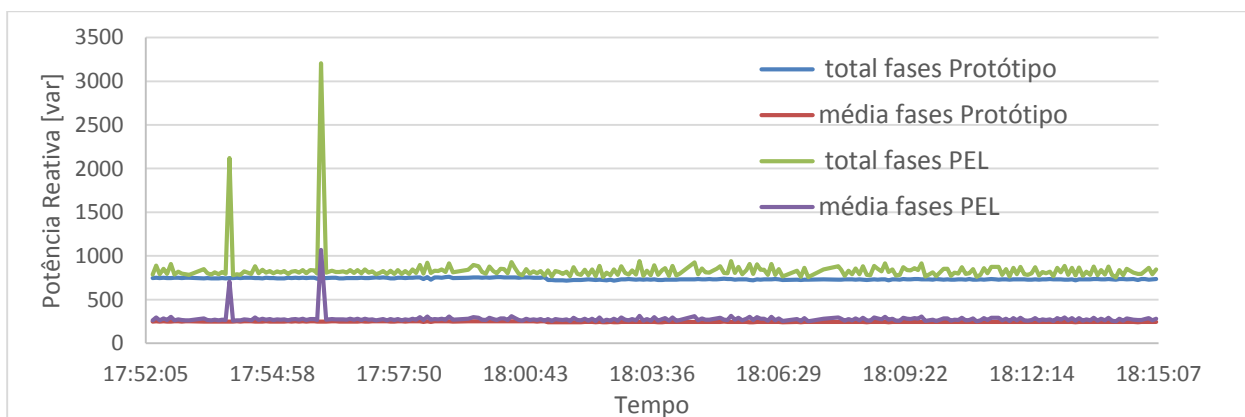


Figura 5.2.3:5: Leitura dos valores de potências reativa médias e soma das fases lidas pelo Protótipo e pelo PEL 103.

Por isso quando é feita as médias das fases e também a soma das mesmas (Fig. 5.2.3:5), as diferenças nas suas grandezas não é muito significativa, a maioria dessa diferença de grandeza recai somente sobre a leitura anormal da fase C apresentada pelo PEL 103. Na leitura da potência reativa do PEL os

dois picos que ocorrem entre as 17:53:59 e as 17:56:04, verificados também na leitura das correntes pelo PEL voltam a ocorrer (Fig. 5.2.3:5).

Passando à leitura dos dados das potências aparentes, é visível que o comportamento do gráfico é semelhante ao das potências reativas (Fig. 5.2.3:6).

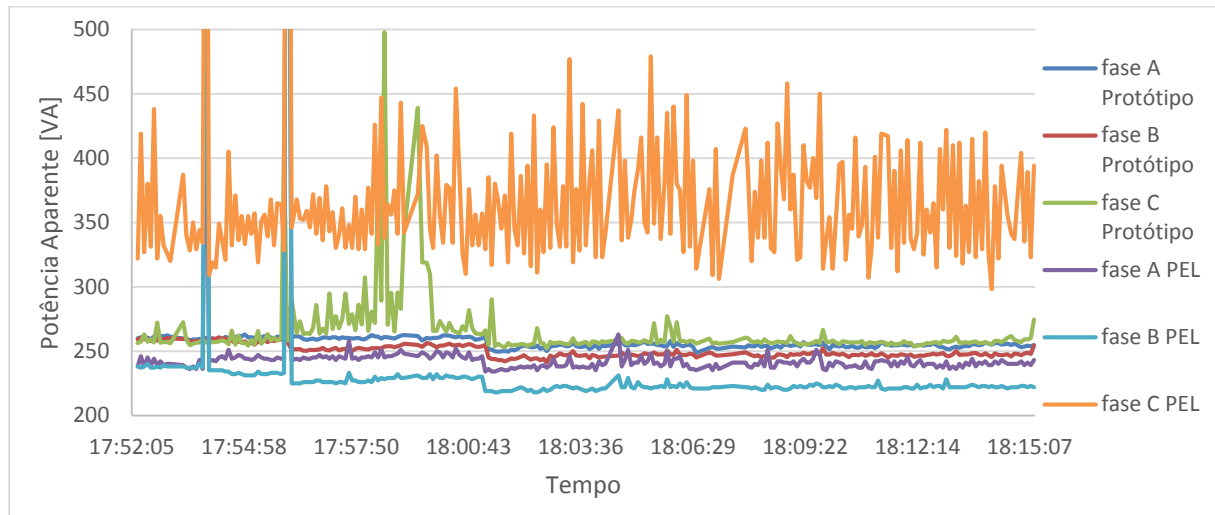


Figura 5.2.3:6: Apresentação das leituras das potências aparentes, em cada fase, feitas pelo Protótipo e pelo PEL 103.

Essa semelhança deve-se claramente à maioria do consumo realizado ser de energia reativa, principalmente nos dados apresentados pelo PEL. O comportamento anómalo descrito na leitura da potência ativa na fase 3 do protótipo, durante o período das 17h:55min e as 18h00min, é também visível no gráfico da potência aparente, sendo que a sua grandeza como valor foi bastante aumentada (Fig. 5.2.3:7 e Fig. 5.2.3:6).

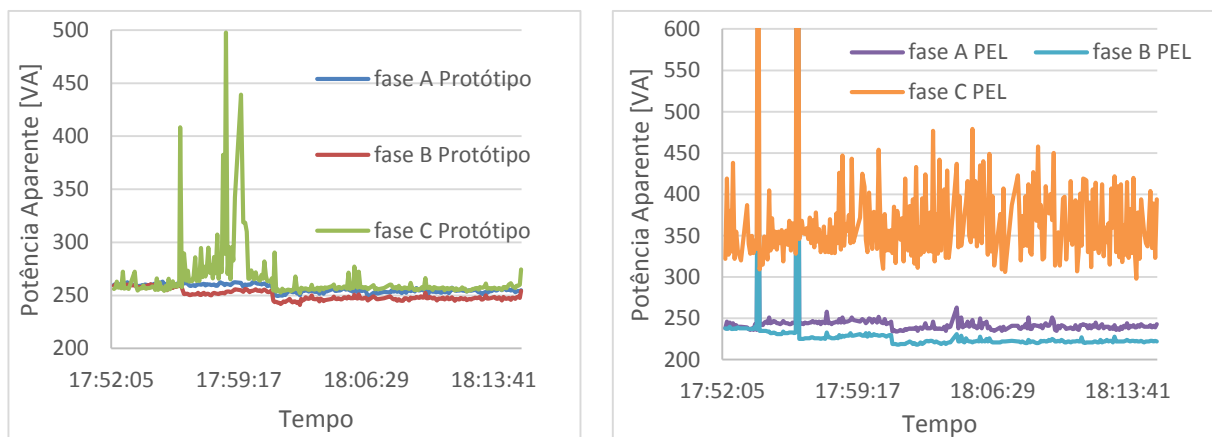


Figura 5.2.3:7: comparação entre as leituras de potência aparente realizadas no Protótipo e realizadas no PEL 103.

É possível notar-se a influência da potência ativa nos dados da potência aparente provenientes das leituras do Protótipo, por outro lado nas leituras de potência aparente pelo PEL (Fig. 5.2.3:7) a componente da potência ativa é de baixa influência devido à sua baixa ordem de grandeza face às leituras de energia reativa consumida. Tal como nos gráficos da potência ativa e reativa, neste também é visível algum dinamismo coincidente entre as leituras nas fases A e B de cada equipamento de monitorização.

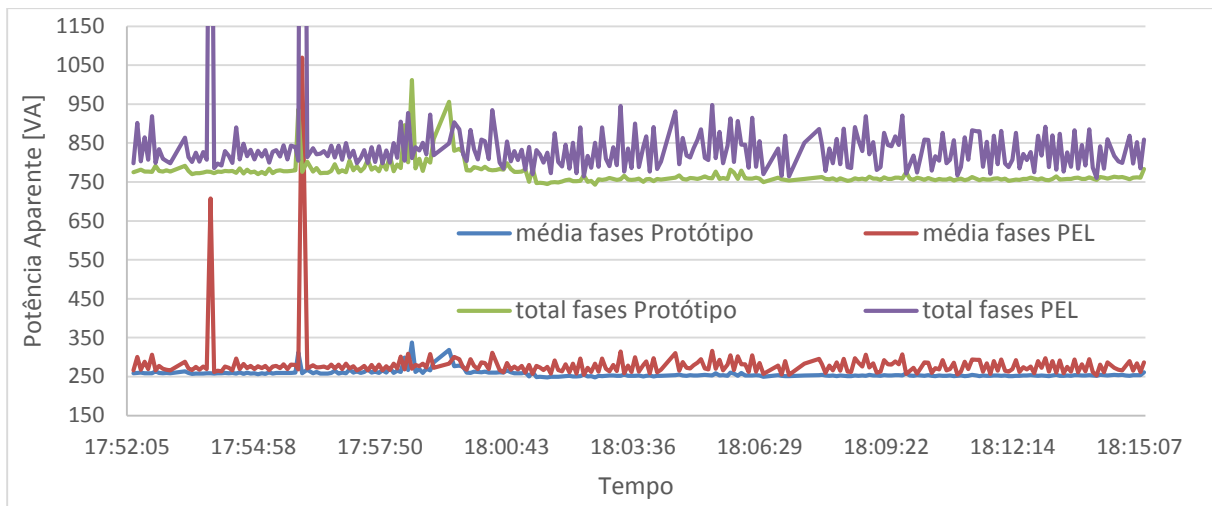


Figura 5.2.3:8: Leitura dos valores de potências aparentes médias e soma das fases lidas pelo Protótipo e pelo PEL 103.

As diferenças entre a potência aparente medida pelo Protótipo e pelo PEL 103 não são muito significativas (Fig. 5.2.3:8), sendo os fatores que apresentam maiores graus de diferenciação são o comportamento estranho da fase C do PEL na leitura de potência ativa e mais importante o seu comportamento na leitura de potência Reativa. Os picos de leitura da potência aparente nos momentos 17:53:59 e 17:56:04, (Fig. 5.2.3:8) têm origem na leitura dos picos de correntes lidos pelo PEL. Não é claro ainda o porque se eles se manifestarem na potencia reativa e aparente e não na ativa.

5.2.4. Resultados do fator de potência

Quanto aos valores dos fatores de potência, estes são um possível bom indicador do que se pode estar a passar com as leituras anómalas por forma a determinar quais dos dados serão os mais corretos.

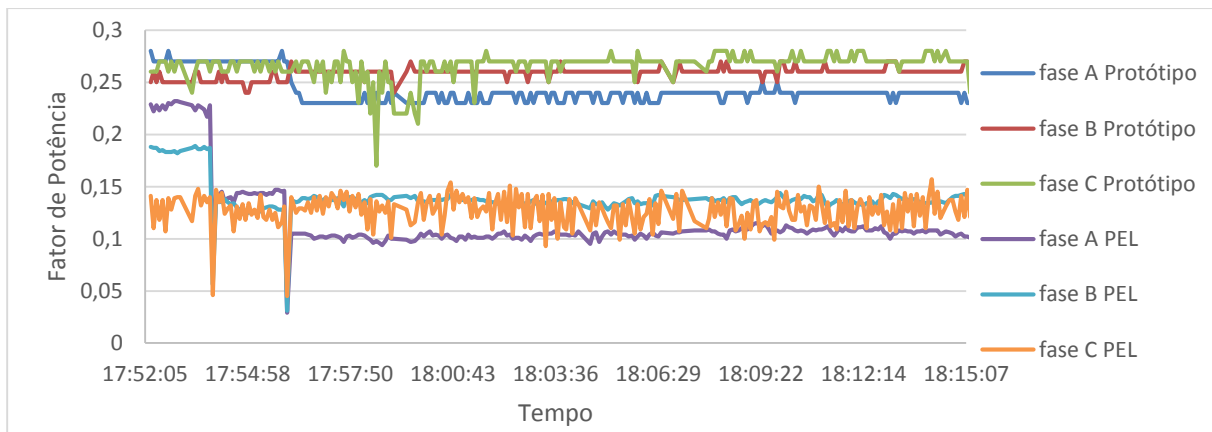


Figura 5.2.4:1: Valores dos fatores de potências das diversas fases medidas pelo Protótipo e pelo PEL 103.

A observação dos valores de Fatores de Potência na figura 5.2.3:9 vão de encontro aos resultados anteriormente observados, sendo que no caso do Protótipo os valores são mais elevados e constantes, sofrendo apenas uma pequena distorção na fase C durante o período de tempo já relatado nas medições de corrente e potência. Já os dados lidos pelo PEL são bastante inferiores, o que justifica a baixa potencia ativa em comparação às leituras do protótipo. O valor do fator de potencia na fase C lido pelo PEL também demonstra um comportamento anómalo que vai de encontro às observações já feitas tanto na potência reativa como aparente, não afetando visivelmente a potência ativa por o valor ser baixo.

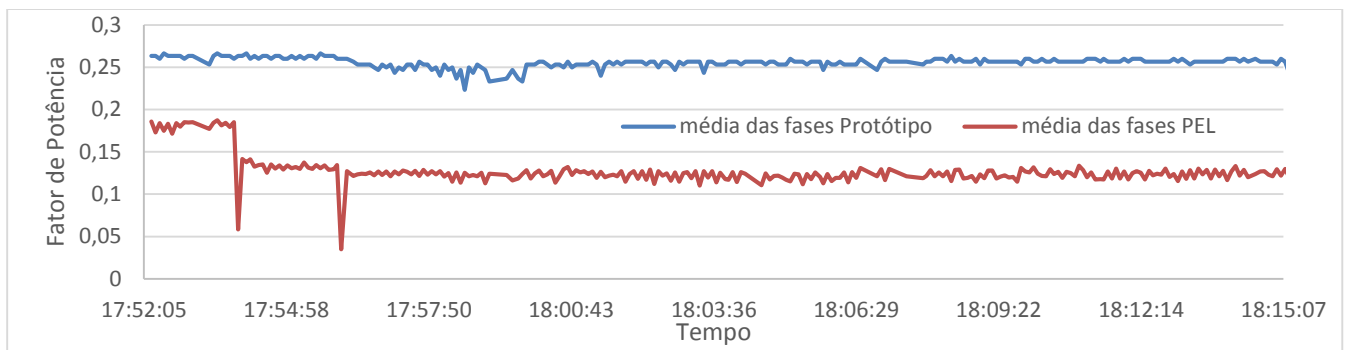


Figura 5.2.4.2: Valores médios dos fatores de potência obtidos através do Protótipo e do PEL 103.

É evidente que os picos de corrente que ocorrem nos dados recolhidos pelo PEL nos momentos 17:53:59 e 17:56:04, estão associados às quebras dos valores do fator de potência para valores na ordem dos 0,058 e 0,035 (Fig. 5.2.3:10). Tal acontecimento é bastante improvável o que é indicador de uma má leitura por parte do PEL nos valores de corrente, bem como do seu desfasamento face à tensão. O facto de os valores de potência ativa não terem aumentado com os picos de corrente, deve-se ao facto desta ser, segundo as leituras do PEL, quase puramente reativa.

Sendo um torno mecânico uma máquina que funciona através de um motor de indução, os valores baixos do fator de potência podem ser justificados pelo facto desta se encontrar em funcionamento sem esforço. Máquinas de indução comuns que estejam em funcionamento podem variar os seus fatores de potência sensivelmente entre valores de fator de potência de 0.85 a 0.17, dependendo de se encontrarem sobre esforço ou não respetivamente (Electrical installation wiki, 2014).

Poder-se-á dizer sem toda a certeza que o Protótipo desenvolvido teve um melhor desempenho nas medições de Potências efetuadas neste teste, muito provavelmente devido às pinças de Rogowski usadas que não são apropriadas para correntes com estas grandezas. Obstante disso os resultados obtidos pelo PEL 103 foram bastantes bons para comparação com os obtidos pelo Protótipo, tornando-se num indicador da sua operacionalidade.

5.2.5. Resultados da energia

Uma vez que o protótipo só é capaz de calcular pacotes de energia de 15 em 15 minutos, e dado a duração do teste realizado foi possível apenas de determinar com detalhe a comparação do consumo dos dois sistemas durante um período de 15 min, entre as 18h:00min e as 18h:15 min (Fig.5.2.4:1).

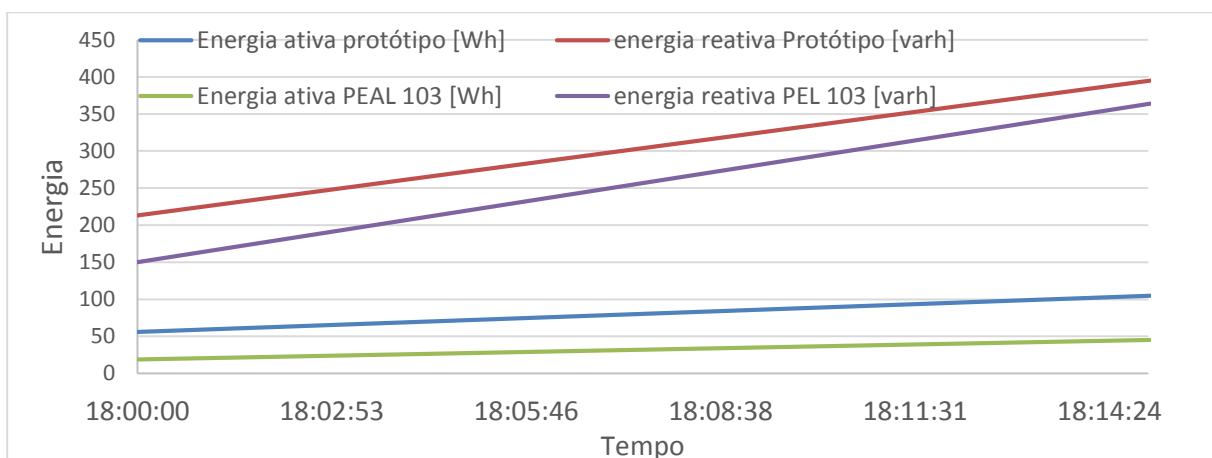


Figura 5.2.5.1: Evolução do consumo de energia ativa e reativa no decorrer do teste, num intervalo de 15 minutos.

Observando o gráfico é possível observar a evolução do consumo de energia nesse intervalo de 15 minutos, e calculando o consumo realmente efetuado durante esse período tem-se os seguintes resultados:

	Energia Ativa	Energia Reativa
Protótipo	49 [W]	182 [var]
PEL 103	26 [W]	214 [var]

Tabela 5.2.5-1 – Consumo da energia ativa e reativa registado pelo Protótipo e pelo PEL 103

Como seria de esperar os dados demonstram que, segundo o Protótipo, o torno mecânico consumiu cerca do dobro daquilo que os dados do PEL 103 aparentam indicar. Quanto ao consumo de Reativa, o PEL assume um maior consumo, o que é expectado dado a leitura anormal de reativa por parte do sensor da fase C do PEL 103.

5.3. Teste com cargas monofásicas

Usando o sistema de conexão para medição de cargas (dispositivos elétricos) em redes monofásicas realizou-se um teste para demonstrar a aplicabilidade deste sistema de conexão. Irá também ser usada, como termo de comparação das medições, uma pinça amperimétrica, sendo que o sistema de conexão permite a monitorização simultânea de 3 cargas, para as quais foram escolhidas 1 carga resistiva e uma indutiva para este teste.



Figura 5.3:1: Estação de soldar



Figura 5.3:2: Ventoinha

Uma vez que a tensão e a frequência são a mesma para as duas cargas, não se irá tratar dessas leituras. Contudo irá estudar-se o comportamento dos equipamentos em termos de correntes eficazes, potências, fatores de potência e consumo de energia.

5.3.1. Leitura e comparação da corrente eficaz

A figura seguinte compara as leituras feitas pelo protótipo, todas as cargas mantêm níveis de consumo estáveis.

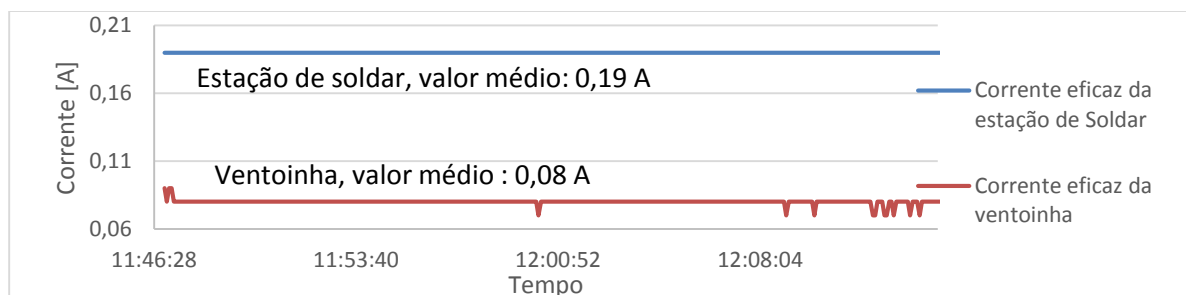


Figura 5.3.1:1: Comparação do consumo das correntes de cargas de teste monofásicas.

Usando a pinça amperimétrica será feito um novo teste de comparação (Fig. 5.3.1:2), neste caso somente às correntes eficazes de cada dispositivo. Será usada uma pinça amperimétrica pois não existe disponível outro equipamento de monitorização. A pinça amperimétrica nas suas especificações indica que possui uma precisão ($\pm 3,5\%$ de rdg + 20 dígitos) para correntes inferiores a 0,5 amperes.



Figura 5.3.1:2: Medições às cargas monofásicas com recurso a uma pinça amperimétrica para comparação de resultados.

Todas as medições registadas com as pinças amperimétricas obtiveram resultados que estão de acordo com os medidos pelo protótipo, sendo que para a estação de soldar o valor foi de 0,19 A e para a ventoinha de 0,071 A. O que se torna mais um indicador de que o protótipo se encontra a realizar leituras fiáveis.

5.3.2. Leitura das potências

As leituras dos dados das potências ativa e reativa também se mostram estáveis e coerentes, é possível verificar que a estação de soldar produz alguma, ainda que baixa, carga reativa. O que faz com que também possua características de uma carga capacitiva (Fig. 5.3.2:1).

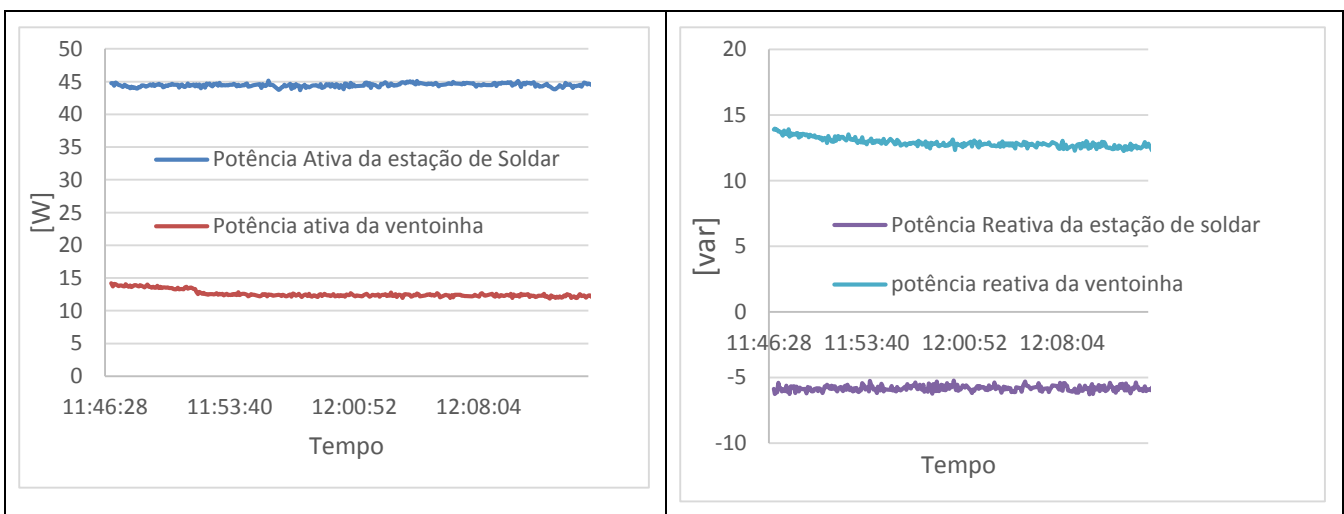


Figura 5.3.2:1: Dados das potências ativa e reativa no teste das cargas monofásicas.

No caso da carga resistiva (estação de soldar), a componente em potência ativa é positiva e supera em ordens de grandeza largamente a componente reativa (Fig. 5.3.2:1), o que se revela, como seria de esperar, uma característica das cargas resistivas. Já a carga indutiva (ventoinha) apresenta uma potencia ativa e reativa positivas e bastante semelhantes (Fig.5.3.2:1) em termos de ordem de grandeza, o que indica um desfaseamento de entre as fases de corrente e tensão de 45° ou $\frac{\pi}{4}$ e radianos. É possível antever por isso um fator de potência para esta carga de $\cos \frac{\pi}{4} \approx 0,7$.

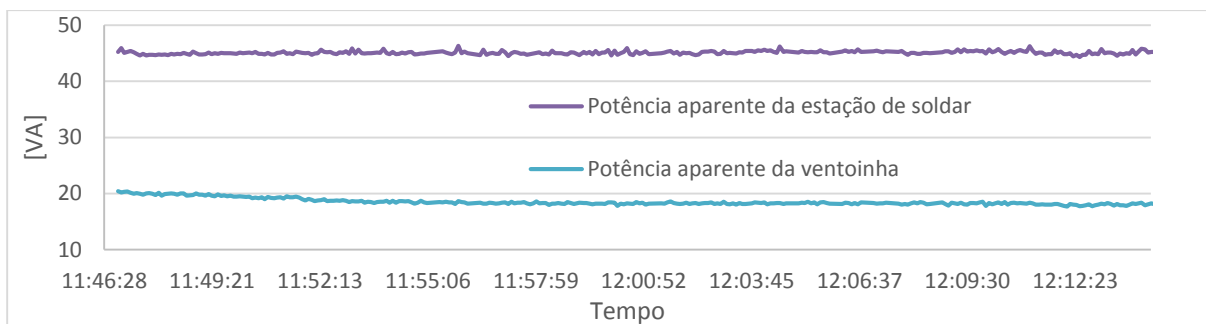


Figura 5.3.2:2: Dados das potências aparentes no teste das cargas monofásicas.

Os dados de potência aparente também se revelam estáveis e precisas (Fig.5.3.2:2).

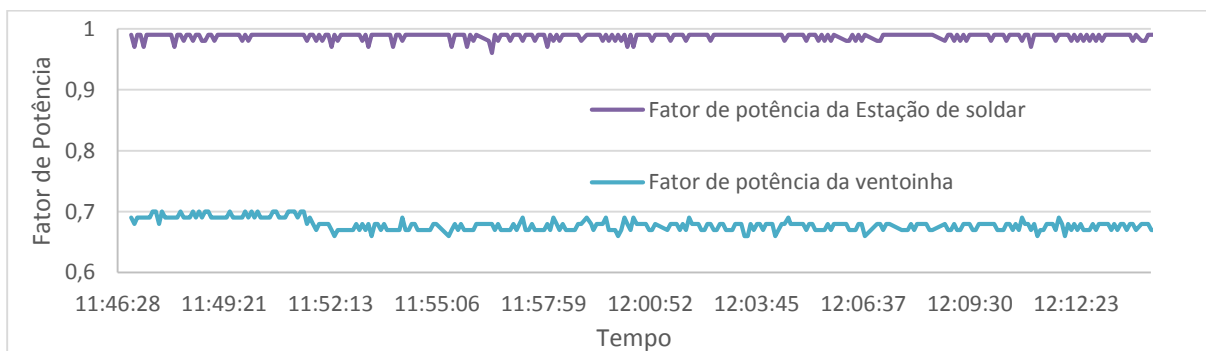


Figura 5.3.2:3 Dados dos fatores de potência no teste das cargas monofásicas.

Observando os dados dos fatores de potência é observável um valor elevado do fator de potência, perto de 1, por parte da carga resistiva, com ligeiras oscilações que derivam da geração de alguma carga reativa (Fig. 5.3.2:3). Isto ocorre possivelmente por a estação possuir condensadores no seu circuito elétrico por forma a controlar a variação de potência.

Já a o fator de potência da ventoinha corresponde ao previsto, tal previsão é óbvia pois uma produção igual de potência ativa e reativa é indicador de uma fator de potência nesta ordem de grandeza (Fig. 5.3.2:3).

5.3.3. Leitura da Energia

Os dados obtidos da energia consumida pelas cargas possui um comportamento esperado.

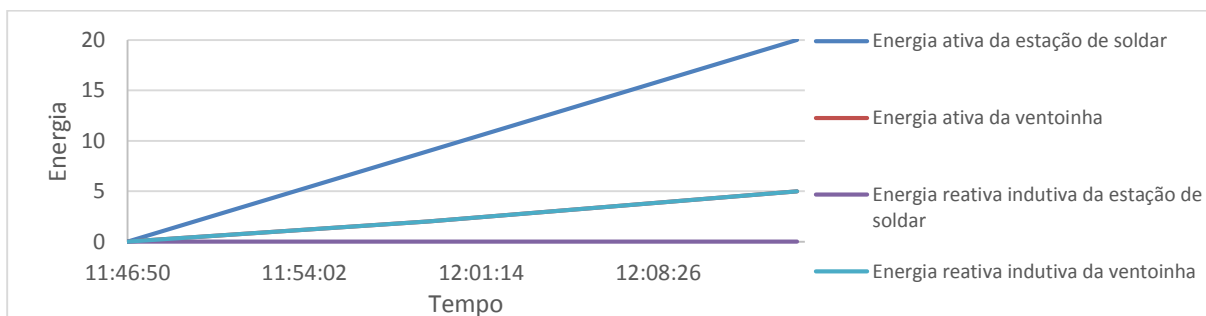


Figura 5.3.3:1: Dados da energia consumida no teste das cargas monofásicas.

Os valores de energia demonstram uma evolução de consumo linear, o que é esperado uma vez que as suas potências possuem grandezas constantes ao longo do tempo. É possível verificar que no período de tempo que decorreu a experiência, a carga resistiva só consumiu valores mesuráveis de energia ativa, o que é normal dada a sua natureza resistiva. Já interessantemente a ventoinha consumiu aproximadamente a mesma quantidade de energia ativa e reativa, dado o seu equilíbrio entre o consumo de ambas as potências.

6. Conclusões

O breve estudo comercial realizado nesta dissertação ao panorama atual do uso e aplicabilidade dos sistemas de monitorização de energia a nível nacional e internacional demonstra que existe neste momento interesse por parte tanto de consumidores domésticos e industriais bem como das ORD's de encontrar as melhores soluções para a monitorização do consumo de energia e também da sua qualidade. A nível político europeu a aprovação de diretivas como a 2009/72/CE (eletricidade) faz como que no futuro a disseminação destes equipamentos ocorra rapidamente, sendo que só em Portugal já existem instalados na rede da EDP-D no regime de consumo BTN mais de 1.400.000 contadores estáticos em contraste com 4.500.000 contadores eletromecânicos, número que irá certamente reverter nos próximos anos.

Assim recorrendo a estudos nacionais e europeus como o "Empower Demand", "Relatório 1E/G" e o projeto "Smartmeter 250 residências" determinou-se quais seriam os aspetos mais relevantes que o protótipo a desenvolver deverá possuir em termos de capacidades e forma de apresentação. A conclusão aponta para o desenvolvimento de um sistema de monitorização em tempo real de corrente trifásica/monofásica com a capacidade de registar o consumo de energia e potência (ativa, reativa e aparente) bem como apresentar dados da qualidade do consumo da rede elétrica, como a frequência, tensão e fator de potência. O Protótipo desenvolvido é adaptado para os mercados de consumo em BTN e BTE, e as suas principais características destacam-se em:

Forma de utilização simples, entrando em funcionamento automaticamente recorrendo apenas ao uso de uma botão de ligar e desligar;

O Protótipo não possui um monitor, sendo possível verificar o seu funcionamento através da observação de leds que sinalizam o envio de dados;

Capacidade de se adaptar a consumos na rede trifásica bem como de cargas individuais na rede monofásicas, graças a um sistema de adaptação desenvolvido;

Apresentação em tempo real dos dados num Website com gráficos dinâmicos e de leitura intuitiva, de fácil acesso via computador, Tablet ou smartphone. Aumentando a portabilidade e capacidade de visualização de dados em qualquer local com acesso à internet;

Apresentação no *website* de resultados de gasto em tempo real, consumo de CO₂, consumo equivalente em Kg de petróleo e gasto energético mensal em €.

Possibilidade de descarregar tabelas detalhadas do consumo e qualidade das leituras da rede acedendo de forma simples à base de dados do repositório na nuvem;

Utilização em lugares mais remotos. O uso de uma placa GSM faz com que seja necessária apenas a presença de uma rede de telecomunicação 2.5G para o envio dos dados de leitura.

Apesar de este ser um produto de medição para sistemas trifásicos, este tem a capacidade de através de uma extensão desenvolvida, monitorizar o consumo individual de carga em sistemas monofásicos. O que pode se apresentar como uma das principais vantagens para o consumidor doméstico, uma vez que o permite analisar quais os equipamentos que consomem mais, verificar os consumos parasitas existentes na residência e reduzir assim facilmente o valor da fatura elétrica.

Os testes realizados ao protótipo desenvolvido não permitiram verificar de forma conclusiva que todos os dados recolhidos têm a qualidade esperada. Isto deve-se ao facto dos equipamentos de referência e dispositivo experimental não serem adequados para os testes que se pretendiam realizar no âmbito desta tese. Contudo foi o suficiente para verificar as capacidades dinâmicas de leitura do protótipo e a validade da maior parte das grandezas medidas (com exceção da corrente e fator potência no regime trifásico) e o bom funcionamento de todo o sistema desde a recolha dos dados até à sua apresentação gráfica e armazenamento *online*.

Conclui-se que os objetivos propostos foram atingidos, sendo que o protótipo desenvolvido possui uma capacidade de leitura e apresentação de dados fácil, útil e capaz de competir com produtos de mercado de custo muito mais elevado e que se adequa às principais necessidades que tanto os consumidores das redes BTN e BTE procuram obter. Os dados são apresentados de uma forma intuitiva e de fácil acesso, procurando adaptar-se a uma combinação de resultados de testes feedback (estudos...) realizados em testes piloto internacionais por forma a seguir um conceito de leitura e apresentação de dados de consumo já com provas dadas. A tecnologia de comunicação usada é atual e neste momento é a que apresenta os melhores resultados com os custos mais baixos para uma aplicação generalizada na rede elétrica.

Referências

- Águas, Miguel, (2013). Lisboa e-nova: Smartmetering: um piloto em 250 residências de Lisboa, 11 de Dezembro de 2013.
- Arduino, (2014). Arduino Forum. Web site, 2014. <http://forum.arduino.cc/>
- Arduino, (2014). Referencies of Arduino. Web site, 2014. <http://arduino.cc/en/Reference/HomePage>
- Chauvin Arnoux, (2004). AmpFLEX™ Série A100 User's Manual
- Crmagnetics, (2014). High Ratio Wire Lead Transformers - CR8400 SERIES. Web site, 2014. <http://www.crmagnetics.com/Products/Assets/ProductPDFs/CR8400%20Series.pdf>
- D. A. Ward, J. La T. Exon, (1993). Using Rogowski coils for transient current measurements.
- D.R. Despacho n.º 15793-D/2013, (2013). Diário da República, 2.ª série — N.º 234 — 3 de dezembro de 2013.
- D.R. Despacho n.º 7253/2010, (2010). Diário da República, 2.ª série — N.º 80 — 26 de Abril de 2010.
- D.R. portaria n.º231/2013, (2013). Diário da República, 1.ª série — N.º 139 — 22 de julho de 2013.
- D.R. portaria n.º26/2013, (2013). Diário da República, 1.ª série — N.º 17 — 24 de janeiro de 2013.
- EDP, (2014). Contadores de Eletricidade. Web site, 2014. <https://energia.edp.pt/particulares/apoio-ao-cliente/contadores-de-eletricidade.aspx>
- EDP-D, (2011). Manual de ligações à rede elétrica do serviço público: Guia Técnico e Logístico de boas Práticas, 3ª edição, Junho de 2011.
- Electrical installation wiki, (2014). Schneider Electric – Practical values of power factor: Average power factor values for the most commonly-used equipment and appliances. Web site, 2014. http://www.electrical-installation.org/enwiki/Practical_values_of_power_factor
- Elkor Technologies Inc, (2006). AN0305, rev A.
- Energizer, (2014). ENERGIZER E95 Product Datasheet. Web site, 2014. <http://data.energizer.com/PDFs/e95.pdf>
- ERSE, (2014). Plano de promoção da eficiência no consumo de energia. Web site, 2014. <http://www.erse.pt/pt/planodepromocaodaeficiencianoconsumoppec/Paginas/default.aspx>
- ERSE, (2014). Preços de Referencia no Mercado Liberalizado de Energia Elétrica e Gás Natural em Portugal Continental, atualização de 16 Julho 2014.
- ERSE, (2014). Tarifas de Acesso às Redes. Web site, 2014. <http://www.erse.pt/pt/electricidade/tarifaseprecos/2014/Paginas/TarifasAcessoRedes2014.aspx>
- ERSE, (2014). Tarifas transitórias de Venda a Clientes Finais em Portugal continental em 2014.
- Ferreira Alves, Mário, (1999). ABC Dos Circuitos Eléctricos em Corrente Alternada.
- FEUP, (2014). Norma NP EN 50160:2001. Web site, 2014. http://paginas.fe.up.pt/~ee86007/pagina_norma_np_en_50160-2001.htm
- Gomes, André, (2012). Contador de Energia Elétrica Trifásico.
- Henriques, Ricardo, (2012). Desenvolvimento de um Sistema de Monitorização Remota do Consumo de Energia Elétrica e da Presença Humana em Edifícios
- IBERDROLA, (2008). Manual de boas práticas energéticas
- ISR-EDP (2005). Manual da qualidade da energia elétrica.

KEMA, (2012). Estudo sobre contadores inteligentes de electricidade e de gás natural: Relatório 1E/G: Situação atual e experiência com projectos-piloto em Portugal.

Lab-Volt Ltd, (2010). Three-Phase AC Power Circuits.

Magueijo, V., Fernandes, M^aC., Matos, H.A., Nunes, C.P., Calau, J.P., Carneiro, J., Oliveira, F. (2010). Medidas de eficiência energética aplicáveis à indústria Portuguesa: Um enquadramento tecnológico sucinto.

Martins, J.S., Couto, C., Afonso, J.L. (2003). Qualidade de energia Elétrica.

Maxim Intergrated, (2008). DS1307 64x8, Serial, I²C Real-Time Clock

Maxim Intergrated, (2013). MAX78630+PPM Evaluation Kit User Manual. Rev 0; 7/13.

Maxim Intergrated, (2014). MAX78630+PPM Energy Measurement Processor for Polyphase Monitoring Systems Data sheet. Rev 3; 8/14.

Mínguez, Agustín, (2007). Medidores de Energia Ativa: Funcionamento, práticas usuais, Principais ensaios e Analise das Fraudes mais comuns.

P.Mell, T.Grance, (2011). The NIST Definition of Cloud computing.

PJFR, (2014). AltSoftSerial Library. Web site, 2014. https://www.pjrc.com/teensy/td_libs_AltSoftSerial.html

Shahid, W., Dailey, N., M., (2012). Asian Institute of Technology Handout: Arduino “Getting Started” Tutorial

Stromback, J., Dromacque, C., Yassin, M.H. (2011). The potential of smart meter enabled programs to increase energy and systems efficiency: a mass pilot comparison “Empower Demand”.

Svensson, Stefan, (1999). Power measurement techniques for non-sinusoidal conditions – The significance of harmonics for the measurement of power and other AC quantities.

Web Developers Notes, (2014). What is webserver – a computer of a program?. Web site, 2014. http://www.webdevelopersnotes.com/basics/what_is_web_server.php

Anexos

A. Glossário de Termos.

1. **PLC-DCSK (Power Line Communication - Differential Chaos Shift Keying):** Sistema de comunicação que transmite dados através de cabos que são usados também para transmitir corrente alterna ou distribui-la aos consumidores, que no caso de comunicação DCSK, comprimentos de onda caóticos transmitem a informação em dois intervalos de tempo consecutivos.
2. **PLC - PRIME (Power Line Communication - PowerLine Intelligent Metering Evolution):** Tecnologia PLC pública, aberta e não-proprietária numa arquitetura de telecomunicações que suporta funcionalidades presentes e futuras de gestão de redes avançadas de monitorização. Sendo uma tecnologia recente, é possível que poderá reduzir o custo das telecomunicações em sistemas de monitorização elétrica inteligente.
3. **RF Mesh (Radio Frequency Mesh):** As redes mesh (malha) dos dias de hoje são frequentemente encontradas em aplicações industriais (por exemplo sistemas de medição e contadores). Trata-se de uma rede de comunicações constituída por nós e organizada numa topologia “malha”, que pode funcionar entre empresas, aplicações móveis entre veículos e baseada em outras tecnologias, como o Zibbee, para ser usada como um sensor de redes web.
4. **GPRS (General Packet Radio Service):** O núcleo da rede GPRS permite comunicações para transmitir pacotes IP para redes externas, como a internet. O subsistema GPRS é uma parte integrada da rede GSM (Global System for Mobile Communications)
5. **Comparação-Entre-Pares:** Consiste na comparação do consumo elétrico entre as habitações dos participantes e habitações de dimensão semelhante. Isto possibilita os participantes se eles usam mais ou menos eletricidade que os seus pares.
6. **Preço da eletricidade:** Indica o preço corrente da eletricidade por kWh. Não inclui faturas de eletricidade atualizadas.
7. **Comparação de histórico:** Mostra o consumo atual da habitação em comparação com os consumos realizados nos períodos pré-monitorização. Os consumidores podem saber se aumentaram ou reduziram o seu consumo em comparação com o mês anterior ou mesmo período de meses anteriores.
8. **Desagregação do consumo:** O consume elétrico doméstico é dividido entre os vários equipamentos elétricos domésticos. A profundidade e grau dessa divisão pode variar, mas usualmente é focado no consumo do forno, frigorífico, televisão e a iluminação. Permite aos consumidores ver o quanto estes equipamentos consomem e tomarem medidas apropriadas.
9. **Nível de consumo atualizado:** Apresenta o consumo até à data da habitação em kWh. Sozinho, não inclui o custo da eletricidade ou custo da fatura. Contudo, Se conjugada com os objetivos de consumo ou limites a não exceder, pode ser um poderoso incentivo para reduzir o consumo.
10. **Custo até à data (fatura):** Apresenta o custo da fatura da eletricidade até à data, que possibilita os consumidores avaliar os seus gastos e agirem sobre eles.

11. **Poupanças relativas a períodos anteriores:** compara a poupança de eletricidade da habitação relativa a períodos anteriores. As habitações teriam um determinado alvo para o seu consumo elétrico que seria uma percentagem de poupança a um período anterior.
12. **Ambiente (Emissões de CO₂):** Isto Mostra a quantidade de CO₂ que as habitações emitem devido ao seu consumo elétrico. Isto apresenta os custos ambientais ou consequências do consumo elétrico doméstico (a quantidade de CO₂ por kWh é calculado de forma diferente em cada país).
13. **Self-service em demanda.** Um consumidor pode unilateralmente provisionar recursos de computação, tais como tempo de servidor e armazenamento de rede, conforme a necessidade automaticamente sem a necessidade de interação humana com cada provedor de serviço.
14. **Acesso ampla de redes.** Recursos estão disponíveis através da rede e são acessíveis através de mecanismos padrão que promovem o uso de plataformas clientes heterogêneos (por exemplo, celulares, tablets, laptops e estações de trabalho).
15. **Agrupamento de recursos.** Recursos de computação do provedor são agrupados para servir a vários consumidores usando um modelo multi-tenant²⁴, com diferentes recursos físicos e virtuais atribuídos e realocados dinamicamente de acordo com a demanda do consumidor. Há um senso de independência local em que o cliente geralmente não tem qualquer controlo ou conhecimento sobre a localização exata dos recursos disponibilizados, mas pode ser capaz de especificar o local num nível mais alto de abstração (por exemplo, país, estado ou datacenter²⁵). Exemplos de recursos incluem o armazenamento, processamento, memória e rede de banda larga.
16. **Rápida elasticidade.** Capacidades podem ser elasticamente provisionadas e libertadas, em alguns casos, automaticamente, para escalar rapidamente para fora e para dentro, comensurável com a demanda. Para o consumidor, as capacidades disponíveis para provisionamento frequentemente parecem ser ilimitados e podem ser apropriadas em qualquer quantidade a qualquer momento.
17. **Serviço Mesurado.** Sistemas em nuvem automaticamente controlam e otimizam o uso dos recursos, aproveitando uma capacidade de medição, com algum nível de abstração apropriado para o tipo de serviço (por exemplo, armazenamento, processamento, banda larga e contas de usuários ativos). O uso de recursos pode ser monitorado, controlado e reportado, oferecendo transparência tanto para o provedor e consumidor do serviço utilizado.
18. **Firmware:** Nos sistemas eletrónicos e computação, firmware é a combinação de memória estática e código de programa e informação gravada neste. No caso do MA78630+PMM esta está contida no processador de medição de 24- Bits. Firmware encontra-se em equipamentos de memória não volátil e pode raramente ou mesmo nunca ser alterada ao longo do seu período de vida
19. **Conexão Single-Ended:** Uma entrada Single-ended mede a tensão entre o canal de entrada (+ ou -) e a terra (0) comum para todas as entradas. Um tipo de conexão alternativa é a denominada de Differential.

²⁴ É uma arquitetura em que uma única instância de uma aplicação de software serve múltiplos clientes. Cada cliente é chamado de tenant.

²⁵ É um repositório centralizado, para o armazenamento, gerenciamento e divulgação de dados e informações organizadas em torno de um corpo específico de conhecimento ou pertencente a uma empresa particular.

20. **Conexão Differential:** Mede a tensão entre duas entradas individuais que se encontram num intervalo comum. A medição é independente das medições de baixo nível (terra), o que a torna imune a ruído no sinal.
21. **Registo Offset** – Os registos Offset para cada entrada analógica contêm valores a ser subtraído das saídas ADC inalteradas com a finalidade de remover do sistema desvios inerente de corrente contínua de quaisquer valores calculados de potência e valores RMS.
22. **Memória RAM** – É um tipo de memória que permite a leitura e a escrita, utilizada como memória primária em sistemas eletrônicos digitais. É onde ficam armazenados os programas básicos operacionais, sendo um tipo de memória volátil.
23. **Memória Flash** – É semelhante à memória RAM, permitindo que múltiplos endereços sejam apagados ou escritos numa só operação. Em termos leigos, trata-se de um chip re-escrevível que, ao contrário de uma memória RAM convencional, preserva o seu conteúdo sem a necessidade de fonte de alimentação, sendo uma memória não volátil.
24. **ADC** – É um dispositivo eletrônico capaz de gerar uma representação digital a partir de uma grandeza analógica, normalmente um sinal representado por um nível de tensão ou intensidade de corrente elétrica.
25. **ADC-(SAR) – successive-approximation-register** – Representam a maioria do Mercado de ADC's. Estes produzem velocidades de amostragem até as 5 milhões de amostras por segundo com resoluções entre os 8 a 18 bits. A arquitetura SAR permite ADC's de baixo consumo e dimensão com elevado rendimento, dentro das exigências atuais.
26. **ADC – (Delta-Sigma)** – Os conversores Delta sigma (usados nesta dissertação) possuem uma vantagem sobre as arquiteturas SAR em termos de não necessitarem de um grande nível de calibração para obterem valores de resolução de 16 a 18 bits. Pelo facto da sua taxa de amostragem ser muito mais alta que a sua velocidade de processamento de informação.
27. **HTTP (Hypertext Transfer Protocol)** – É a fundação das transferências de informação na Internet. Possui um protocolo de pedido-resposta no modelo computacional cliente-servidor. Um cliente pode ser por exemplo um Browser e uma aplicação a correr alojada num webserver ser o servidor. O cliente submete um pedido em forma de mensagem http ao servidor, que por sua vez se correto é respondido com conteúdo por parte do servidor.
28. **API (Application programming interface)** – em programação computacional é uma aplicação que especifica uma componente de software em termos das suas operações, dos seus inputs e outputs. O seu principal propósito é o de definir um conjunto de funcionalidades que são independentes da sua implementação respetiva, permitido tanto variações de definição e implementação sem estas se comprometerem mutuamente.
29. **Baud Rate** – É a velocidade com que a informação é transmitida em um canal de comunicação em forma de bits. Uma Baud "9600" significa que a porta serie só consegue transmitir um máximo de 9600 bits por segundo, isto no caso unidade de informação for de 1 baud (1 bit). Se um baud for de 10 bits (um pacote de bits), a velocidade de transferência de bits continuará a ser de 9600 bits por segundo, a de transferência de baud será de 960 por segundo. A transferência de informação via porta série entre dispositivos deve ser feita à mesma velocidade baud.

B. Código de programação do Arduino IDE

```
#include <AltSoftSerial.h>
#include <Wire.h>
#include "GSM.h"
#define DS1307_ADDRESS 0x68
AltSoftSerial altSerial;

// PIN Number
#define PINNUMBER "3387"

// APN data
#define GPRS_APN    "internet.vodafone.pt" // replace your GPRS
APN
#define GPRS_LOGIN  "Vodafone" // replace with your GPRS login
#define GPRS_PASSWORD "vodafone" // replace with your GPRS
password
// initialize the library instance
GSMClient client;
GPRS gprsAccess;
GSM gsmAccess;
```

```
// URL, path & port (for example: arduino.cc)
char server[] = "energymeter-fcul.rhcloud.com";
int port = 80; // port 80 is the default for HTTP

byte temp0;
byte zero = 0x00; // byte do RTC
char byteAA = 0xAA;//0
char byte04 = 0x04;//1
char byteC0 = 0xC0;//2
char byte92 = 0x92;//3
char byteC1 = 0xC1;//4
char byte91 = 0x91;//5
char byte07 = 0x07;//6
char byteA3 = 0xA3;//7
char byte03 = 0x03;//8
char byte00 = 0x00;//9
char byteE3 = 0xE3;//10
char byteC6 = 0xC6;//11
char byte74 = 0x74;//12
char byte01 = 0x01;//13
char byte54 = 0x54;//14
```

```
char byte80=0x80;//15
char byte48=0x48;//16
char byte65=0x65;//17
char byte63=0x63;//18
char byte90=0x90;//19
char byte39=0x39;//20
char byte44=0x44;//21
char byte84=0x84;//22
char byteD5=0xD5;//23
char byteF4=0xF4;//24
char byte41=0x41;//25
char byte87=0x87;//26
char byte26=0x26;//27
char byteA2=0xA2;//28
char byte2F=0x2F;//29
char byte99=0x99;//30
char byte1D=0x1D;//31
char byteAB=0xAB;//32
char byte68=0x68;//33
char byte60=0x60;//34
char byte93=0x93;//35
```

```
char byte36=0x36;//36
char byteD8=0xD8;//37
char byteF1=0xF1;//38
char byte20=0x20;//39
char byte29=0x29;//40
char byteA8=0xA8;//41
char byte9F=0x9F;//42
char byte32=0x32;//43
char byte96=0x96;//44
char byte6B=0x6B;//45
char byte5D=0x5D;//46
char byte33=0x33;//47
char byteDB=0xDB;//48
char byteEE=0xEE;//49
char byte23=0x23;//50
char byteA5=0xA5;//51
char byte2C=0x2C;//52
char byte9C=0x9C;//53
char byte35=0x35;//54
char byteDD=0xDD;//55
char byteEB=0xEB;//56
```

```
char byteEF=0xEF;//57
char byteD9=0xD9;//58
char byte02=0x02;//59
char byteC9=0xC9;//60
char byte0A=0x0A;//61
char byteD3=0xD3;//62
char byteB0=0xB0;//63
char byte13=0x13;//64
char byteB4=0xB4;//65
char byte25=0x25;//66
char byte37=0x37;//67
char byteE6=0xE6;//68
char byteE2=0xE2;//69
char byteF8=0xF8;//70
char byteD0=0xD0;//71
char byteBD=0xBD;//72
char byte1C=0x1C;//73
char byte2E=0x2E;//74
char byte40=0x40;//75
double frequency;
double voltage_RMS_A;
```

```
double current_RMS_A;
double Active_Power_A;
double Reactive_Power_A;
double Apparent_Power_A;
double PF_A;
double voltage_RMS_B;
double current_RMS_B;
double Active_Power_B;
double Reactive_Power_B;
double Apparent_Power_B;
double PF_B;
double voltage_RMS_C;
double current_RMS_C;
double Active_Power_C;
double Reactive_Power_C;
double Apparent_Power_C;
double PF_C;
double Active_Energy_A;
double Active_Energy_B;
double Active_Energy_C;
double Reactive_Energy_A;
```

```
double Reactive_Energy_B;
double Reactive_Energy_C;
double PActive_Energy_A;
double PActive_Energy_B;
double PActive_Energy_C;
double PReactive_Energy_A;
double PReactive_Energy_B;
double PReactive_Energy_C;
void setup()
{
// Open serial communications and wait for port to open:
Serial.begin(38400);
altSerial.begin(38400);
for (int pinnumber=10; pinnumber<12; pinnumber++){
  pinMode(pinnumber,OUTPUT);
  digitalWrite(pinnumber, LOW);
}
Wire.begin();
gsmInit();
delay(100);
}
```

```
void loop(){
  Wire.beginTransmission(DS1307_ADDRESS);
  Wire.write(zero);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_ADDRESS, 7);

  int seconds = bcdToDec(Wire.read());
  int minutes = bcdToDec(Wire.read());
  int hours = bcdToDec(Wire.read() & 0b111111); //24 hour time
  int weekDays = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
  int monthDays = bcdToDec(Wire.read());
  int months = bcdToDec(Wire.read());
  int years = bcdToDec(Wire.read());

  if(seconds==0 | seconds==5 | seconds==10 | seconds==15 | seconds==
  20 | seconds==25 | seconds==30 | seconds==35 | seconds==40 | seco
  nds==45 | seconds==50 | seconds==55){
    Serial.println();
    Serial.print(months);
    Serial.print(F("/"));
    Serial.print(monthDays);
    Serial.print(F("/"));
  }
```

```
Serial.print(years);
Serial.print(F(" "));
Serial.print(hours);
Serial.print(F(":"));
Serial.print(minutes);
Serial.print(F(":"));
Serial.println(seconds);
Serial.print(F("Composite Frequency:"));
frequency=SoCreaderFreq (); //line frequency
Serial.print(frequency);
Serial.println();

start();
altSerial.print(byte65);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byte63);
delay(5);
Serial.print(F("Power Factor, fase A:"));
PF_A=SoCreaderPF (); // Power factor A
Serial.print(PF_A);
```

```
Serial.println();

start();
altSerial.print(byte90);
altSerial.print(byte00);
altSerial.print(byteE3);
altSerial.print(byte39);
delay(5);
Serial.print(F("RMS voltage, fase A:"));
voltage_RMS_A=SoCreaderVoltage (); // A RMS voltage
Serial.print(voltage_RMS_A);
Serial.println();

start();
altSerial.print(byteD5);
altSerial.print(byte00);
altSerial.print(byteE3);
altSerial.print(byteF4);
delay(5);
Serial.print(F("RMS Current, fase A:"));
current_RMS_A=SoCreaderCurrent (); // A RMS current
```

```
Serial.print(current_RMS_A);  
Serial.println();  
  
start();  
altSerial.print(byte1D);  
altSerial.print(byte01);  
altSerial.print(byteE3);  
altSerial.print(byteAB);  
delay(5);  
Serial.print(F("Active Power, fase A:"));  
Active_Power_A=SoCreaderPower (); // Active Power A  
Serial.print(Active_Power_A);  
Serial.println();  
  
start();  
altSerial.print(byte26);  
altSerial.print(byte01);  
altSerial.print(byteE3);  
altSerial.print(byteA2);  
delay(5);  
Serial.print(F("Reactive Power, fase A:"));
```

```
Reactive_Power_A=SoCreaderPower (); // Reactive Power A  
Serial.print(Reactive_Power_A);  
Serial.println();  
  
start();  
altSerial.print(byte2F);  
altSerial.print(byte01);  
altSerial.print(byteE3);  
altSerial.print(byte99);  
delay(5);  
Serial.print(F("Apparente Power, fase A:"));  
Apparent_Power_A=SoCreaderPower (); // Apparent Power A  
Serial.print(Apparent_Power_A);  
Serial.println();  
  
start();  
altSerial.print(byte68);  
altSerial.print(byte01);  
altSerial.print(byteE3);  
altSerial.print(byte60);  
delay(5);
```

```
Serial.print(F("Power Factor, fase B:"));
PF_B=SoCreaderPF (); // Power Factor B
Serial.print(PF_B);
Serial.println();

start();
altSerial.print(byte93);
altSerial.print(byte00);
altSerial.print(byteE3);
altSerial.print(byte36);
delay(5);
Serial.print(F("RMS Voltage, fase B:"));
voltage_RMS_B=SoCreaderVoltage (); // B RMS voltage
Serial.print(voltage_RMS_B);
Serial.println();

start();
altSerial.print(byteD8);
altSerial.print(byte00);
altSerial.print(byteE3);
altSerial.print(byteF1);
```

```
delay(5);
Serial.print(F("RMS Current, fase B:"));
current_RMS_B=SoCreaderCurrent (); // B RMS Current
Serial.print(current_RMS_B);
Serial.println();

start();
altSerial.print(byte20);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byteA8);
delay(5);
Serial.print(F("Active Power, fase B:"));
Active_Power_B=SoCreaderPower (); // Active Power B
Serial.print(Active_Power_B);
Serial.println();

start();
altSerial.print(byte29);
altSerial.print(byte01);
altSerial.print(byteE3);
```

```
altSerial.print(byte9F);
delay(5);
Serial.print(F("Reactive Power, fase B:"));
Reactive_Power_B=SoCreaderPower (); // Reactive Power B
Serial.print(Reactive_Power_B);
Serial.println();

start();
altSerial.print(byte32);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byte96);
delay(5);
Serial.print(F("Apparent Power, fase B:"));
Apparent_Power_B=SoCreaderPower (); // Apparent Power B
Serial.print(Apparent_Power_B);
Serial.println();

start();
altSerial.print(byte6B);
altSerial.print(byte01);
```

```
altSerial.print(byteE3);
altSerial.print(byte5D);
delay(5);
Serial.print(F("Power Factor, fase C:"));
PF_C=SoCreaderPF (); // Power Factor C
Serial.print(PF_C);
Serial.println();

start();
altSerial.print(byte96);
altSerial.print(byte00);
altSerial.print(byteE3);
altSerial.print(byte33);
delay(5);
Serial.print(F("RMS voltage, fase C:"));
voltage_RMS_C=SoCreaderVoltage (); // C RMS voltage
Serial.print(voltage_RMS_C);
Serial.println();

start();
altSerial.print(byteDB);
```

```
altSerial.print(byte00);
altSerial.print(byteE3);
altSerial.print(byteEE);
delay(5);
Serial.print(F("RMS Current, fase C:"));
current_RMS_C=SoCreaderCurrent (); // C RMS Current
Serial.print(current_RMS_C);
Serial.println();

start();
altSerial.print(byte23);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byteA5);
delay(5);
Serial.print(F("Active Power, fase C:"));
Active_Power_C=SoCreaderPower (); // Active Power C
Serial.print(Active_Power_C);
Serial.println();

start();
```

```
altSerial.print(byte2C);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byte9C);
delay(5);
Serial.print(F("Reactive Power, fase C:"));
Reactive_Power_C=SoCreaderPower (); // Reactive Power C
Serial.print(Reactive_Power_C);
Serial.println();

start();
altSerial.print(byte35);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byte93);
delay(5);
Serial.print(F("Apparent Power, fase C:"));
Apparent_Power_C=SoCreaderPower (); // Apparent Power C
Serial.print(Apparent_Power_C);
Serial.println();
if(minutes==0&&seconds==0 | minutes==15&&seconds==0 | minutes=
=30&&seconds==0 | minutes==45&&seconds==0){
```

```
start();
altSerial.print(byteDD);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byteEB);
delay(5);
Serial.print(F("Consumed Active energy, fase A:"));
Active_Energy_A=SoCreaderEnergy (); // active energy A
Serial.print(Active_Energy_A);
Serial.println();

start();
altSerial.print(byteEF);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byteD9);
delay(5);
Serial.print(F("Consumed Active energy, fase B:"));
Active_Energy_B=SoCreaderEnergy (); // Active Energy B
Serial.print(Active_Energy_B);
Serial.println();
```

```
start();
altSerial.print(byte01);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byteC6);
delay(5);
Serial.print(F("Consumed Active energy, fase C:"));
Active_Energy_C=SoCreaderEnergy (); // Active Energy C
Serial.print(Active_Energy_C);
Serial.println();

start();
altSerial.print(byte13);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byteB4);
delay(5);
Serial.print(F("Inductive Reactive energy, fase A:"));
Reactive_Energy_A=SoCreaderEnergy (); // Reactive energy A
Serial.print(Reactive_Energy_A);
```

```
Serial.println();

start();
altSerial.print(byte25);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byteA2);
delay(5);
Serial.print(F("Inductive Reactive energy, fase B:"));
Reactive_Energy_B=SoCreaderEnergy (); // Reactive Energy B
Serial.print(Reactive_Energy_B);
Serial.println();

start();
altSerial.print(byte37);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byte90);
delay(5);
Serial.print(F("Inductive Reactive energy, fase C:"));
Reactive_Energy_C=SoCreaderEnergy (); // Reactive Energy C
```

```
Serial.print(Reactive_Energy_C);
Serial.println();

start();
altSerial.print(byteE6);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byteE2);
delay(5);
Serial.print(F("Produced Active energy, fase A:"));
PActive_Energy_A=SoCreaderEnergy (); // active energy A
Serial.print(PActive_Energy_A);
Serial.println();

start();
altSerial.print(byteF8);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byteD0);
delay(5);
Serial.print(F("Produced Active energy, fase B:"));
```

```
PActive_Energy_B=SoCreaderEnergy (); // Active Energy B
Serial.print(PActive_Energy_B);
Serial.println();

start();
altSerial.print(byte0A);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byteBD);
delay(5);
Serial.print(F("Produced Active energy, fase C:"));
PActive_Energy_C=SoCreaderEnergy (); // Active Energy C
Serial.print(PActive_Energy_C);
Serial.println();

start();
altSerial.print(byte1C);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byteAB);
delay(5);
```

```
Serial.print(F("Capacitive Reactive energy, fase A:"));
PReactive_Energy_A=SoCreaderEnergy (); // Reactive energy A
Serial.print(PReactive_Energy_A);
Serial.println();

start();
altSerial.print(byte2E);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byte99);
delay(5);
Serial.print(F("Capacitive Reactive energy, fase B:"));
PReactive_Energy_B=SoCreaderEnergy (); // Reactive Energy B
Serial.print(PReactive_Energy_B);
Serial.println();

start();
altSerial.print(byte40);
altSerial.print(byte02);
altSerial.print(byteE3);
altSerial.print(byte87);
```

```
delay(5);
Serial.print(F("Capacitive Reactive energy, fase C:"));
PReactive_Energy_C=SoCreaderEnergy (); // Reactive Energy C
Serial.print(PReactive_Energy_C);
Serial.println();
delay(100);
sendDataDb2();
}
if(monthDays==1&&hours==0&&minutes==0&&seconds==0){
ResetEnergy () ;
}
delay(100);
sendDataDb();
}
}
double start(){
altSerial.print(byteAA);
altSerial.print(byte07);
altSerial.print(byteA3);
}
double SoCreaderFreq (){
```

```
altSerial.print(byteAA);
altSerial.print(byte07);
altSerial.print(byteA3);
altSerial.print(byte80);
altSerial.print(byte01);
altSerial.print(byteE3);
altSerial.print(byte48);
delay(5);
byte inBuffer[6];
int bufIdx = 0;
if (altSerial.available() > 0) {
while (altSerial.available() > 0) {
inBuffer[bufIdx] = altSerial.read();
bufIdx ++;
}
byte r =inBuffer[4];
byte t =inBuffer[3];
byte y =inBuffer[2];

double Dec=(r*65536.0+t*256.0+y)*0.0000152587672179127;
inBuffer[bufIdx] = 0;
```

```
    bufIndx = 0;
    return Dec;
}
}
double SoCreaderPower (){
    byte inBuffer[6];
    int bufIndx = 0;
    if (altSerial.available() > 0) {
        while (altSerial.available() > 0) {
            inBuffer[bufIndx] = altSerial.read();
            bufIndx ++;
        }
        byte r = inBuffer[4];
        byte t = inBuffer[3];
        byte y = inBuffer[2];
        if(r<128){
            double Dec=(r*65536.0+t*256.0+y)*0.00396962860118015;
            inBuffer[bufIndx] = 0;
            bufIndx = 0;
            return Dec;
        }
    }
```

```
    else{
        double Dec2= -((255.0-r)*65536.0+(255.0-t)*256.0+(255.0-
y))*0.00396962860118015;
        inBuffer[bufIndx] = 0;
        bufIndx = 0;
        return Dec2;
    }
    Serial.println();
    bufIndx = 0;
}
}
double SoCreaderVoltage (){
    byte inBuffer[6];
    int bufIndx = 0;
    if (altSerial.available() > 0) {
        while (altSerial.available() > 0) {
            inBuffer[bufIndx] = altSerial.read();
            bufIndx ++;
        }
        byte r = inBuffer[4];
        byte t = inBuffer[3];
        byte y = inBuffer[2];
```

```
double Dec=(r*65536.0+t*256.0+y)*0.0000793945940258412;
inBuffer[bufIdx] = 0;
bufIdx = 0;
return Dec;
}
}
double SoCreaderCurrent (){
byte inBuffer[6];
int bufIdx = 0;
if (altSerial.available() > 0) {
while (altSerial.available() > 0) {
inBuffer[bufIdx] = altSerial.read();
bufIdx ++;
}

byte r = inBuffer[4];
byte t = inBuffer[3];
byte y = inBuffer[2];

double Dec=(r*65536.0+t*256.0+y)*0.00000596046186144157;
```

```
inBuffer[bufIdx] = 0;
bufIdx = 0;
return Dec;
}
}
double SoCreaderPF (){
byte inBuffer[6];
int bufIdx = 0;
if (altSerial.available() > 0) {
while (altSerial.available() > 0) {
inBuffer[bufIdx] = altSerial.read();
bufIdx ++;
}
byte r =inBuffer[4];
byte t =inBuffer[3];
byte y =inBuffer[2];

if(r<128){
double Dec=(r*65536.0+t*256.0+y)*0.00000023841946872415;
inBuffer[bufIdx] = 0;
bufIdx = 0;
```

```
return Dec;
}
else{
double Dec2= -((255.0-r)*65536.0+(255.0-t)*256.0+(255.0-
y))*0.00000023841946872415;
inBuffer[bufIdx] = 0;
bufIdx = 0;
return Dec2;
}
}
}
double SoCreaderEnergy (){
byte inBuffer[6];
int bufIdx = 0;
if (altSerial.available() > 0) {
while (altSerial.available() > 0) {
inBuffer[bufIdx] = altSerial.read();
bufIdx ++;
}
byte r =inBuffer[4];
byte t =inBuffer[3];
byte y =inBuffer[2];
```

```
if(r<128){
double Dec=(r*65536.0+t*256.0+y);
inBuffer[bufIdx] = 0;
bufIdx = 0;
return Dec;
}
else{
double Dec2= -((255.0-r)*65536.0+(255.0-t)*256.0+(255.0-y));
inBuffer[bufIdx] = 0;
bufIdx = 0;
return Dec2;
}
}
}
void ResetEnergy (){
altSerial.print(byteAA);
altSerial.print(byte07);
altSerial.print(byteA3);
altSerial.print(byte00);
altSerial.print(byte00);
```

```
altSerial.print(byteE3);
altSerial.print(byteC9);
delay(5);

altSerial.print(byteAA);
altSerial.print(byte0A);
altSerial.print(byteA3);
altSerial.print(byte00);
altSerial.print(byte00);
altSerial.print(byteD3);
altSerial.print(byteB0);
altSerial.print(byte00);
altSerial.print(byte00);
altSerial.print(byte26);
delay(5);
}

byte bcdToDec(byte val) {
// Convert binary coded decimal to normal decimal numbers
return ( (val/16*10) + (val%16) );
```

```
}
void sendDataDb(){
if( client.connect(server, 80))
{
Serial.println(F("Connected to webserver"));
client.print(F("GET /update.php?"));

client.print(F("frequency="));
client.print(frequency);
client.print(F("&"));

client.print(F("voltage_RMS_A="));
client.print(voltage_RMS_A);
client.print(F("&"));

client.print(F("current_RMS_A="));
client.print(current_RMS_A);
client.print(F("&"));

client.print(F("Active_Power_A="));
client.print(Active_Power_A);
client.print(F("&"));

client.print(F("Reactive_Power_A="));
client.print(Reactive_Power_A);
```

```
client.print(F("&"));
client.print(F("Apparent_Power_A="));
client.print(Apparent_Power_A);
client.print(F("&"));
client.print(F("PF_A="));
client.print(PF_A);
client.print(F("&"));
client.print(F("voltage_RMS_B="));
client.print(voltage_RMS_B);
client.print(F("&"));
client.print(F("current_RMS_B="));
client.print(current_RMS_B);
client.print(F("&"));
client.print(F("Active_Power_B="));
client.print(Active_Power_B);
client.print(F("&"));
client.print(F("Reactive_Power_B="));
client.print(Reactive_Power_B);
client.print(F("&"));
client.print(F("Apparent_Power_B="));
client.print(Apparent_Power_B);
```

```
client.print(F("&"));
client.print(F("PF_B="));
client.print(PF_B);
client.print(F("&"));
client.print(F("voltage_RMS_C="));
client.print(voltage_RMS_C);
client.print(F("&"));
client.print(F("current_RMS_C="));
client.print(current_RMS_C);
client.print(F("&"));
client.print(F("Active_Power_C="));
client.print(Active_Power_C);
client.print(F("&"));
client.print(F("Reactive_Power_C="));
client.print(Reactive_Power_C);
client.print(F("&"));
client.print(F("Apparent_Power_C="));
client.print(Apparent_Power_C);
client.print(F("&"));
client.print(F("PF_C="));
client.print(PF_C);
```

```
client.println(F(" HTTP/1.1"));
client.println(F("Host:energymeter-fcul.rhcloud.com"));
client.println();
digitalWrite(10,LOW);
digitalWrite(11,HIGH);
delay(100);
}
else{
  digitalWrite(10,HIGH);
  digitalWrite(11,LOW);
  Serial.println(F("Could not connect to webserver"));
}
  client.flush();
  client.stop();
}
void sendDataDb2(){
  if( client.connect(server, 80))
  {
    Serial.println(F("Connected to webserver"));
    client.print(F("GET /update_2.php?"));
```

```
client.print(F("Active_Energy_A="));
client.print(Active_Energy_A);
client.print(F("&"));
client.print(F("Active_Energy_B="));
client.print(Active_Energy_B);
client.print(F("&"));
client.print(F("Active_Energy_C="));
client.print(Active_Energy_C);
client.print(F("&"));
client.print(F("Reactive_Energy_A="));
client.print(Reactive_Energy_A);
client.print(F("&"));
client.print(F("Reactive_Energy_B="));
client.print(Reactive_Energy_B);
client.print(F("&"));
client.print(F("Reactive_Energy_C="));
client.print(Reactive_Energy_C);
client.print(F("&"));
client.print(F("PActive_Energy_A="));
client.print(PActive_Energy_A);
```

```
client.print(F("&"));
client.print(F("PActive_Energy_B="));
client.print(PActive_Energy_B);
client.print(F("&"));
client.print(F("PActive_Energy_C="));
client.print(PActive_Energy_C);
client.print(F("&"));
client.print(F("PReactive_Energy_A="));
client.print(PReactive_Energy_A);
client.print(F("&"));
client.print(F("PReactive_Energy_B="));
client.print(PReactive_Energy_B);
client.print(F("&"));
client.print(F("PReactive_Energy_C="));
client.print(PReactive_Energy_C);

client.println(F(" HTTP/1.1"));
client.println(F("Host:energymeter-fcul.rhcloud.com"));
client.println();
delay(100);
}
```

```
else{
  Serial.println(F("Could not connect to webserver"));
}
client.flush();
client.stop();
}

void gsmInit(){
  Serial.println(F("Connecting GSM network..."));
  if(gsmAccess.begin(PINNUMBER)!=GSM_READY)
  {
    Serial.println(F("GSM PIN ERROR"));
    while(true);
  }
  Serial.println(F("GSM PIN OK"));
  delay(5000);
  // attach GPRS
  Serial.println(F("Attaching to GPRS with your APN..."));
  if(gprsAccess.attachGPRS(GPRS_APN,GPRS_LOGIN,GPRS_PASSWORD)!=GPRS_READY){
    Serial.println(F("GSM ATTACH ERROR"));
  }
  else{
```

```
Serial.println(F("GSM ATTACH OK"));  
}  
}
```

C. Código Arduino de Calibração do RTC

```
#include "Wire.h"  
#define DS1307_ADDRESS 0x68  
byte zero = 0x00; //workaround for issue #527  
  
void setup(){  
  Wire.begin();  
  Serial.begin(9600);  
  setDate(); //MUST CONFIGURE IN FUNCTION  
}  
  
void loop(){  
  printDate();  
  delay(1000);  
}  
  
void setDate(){
```

```
  byte seconds = 00; //0-59  
  byte minutes = 46; //0-59  
  byte hours = 16; //0-23  
  byte weekDays = 5; //1-7  
  byte monthDays = 5; //1-31  
  byte months = 9; //1-12  
  byte years = 14; //0-99  
  
  Wire.beginTransaction(DS1307_ADDRESS);  
  Wire.write(zero); //stop Oscillator  
  
  Wire.write(decToBcd(seconds));  
  Wire.write(decToBcd(minutes));  
  Wire.write(decToBcd(hours));  
  Wire.write(decToBcd(weekDays));  
  Wire.write(decToBcd(monthDays));  
  Wire.write(decToBcd(months));  
  Wire.write(decToBcd(years));  
  
  Wire.write(zero); //start
```

```
Wire.endTransmission();

}

byte decToBcd(byte val){
// Convert normal decimal numbers to binary coded decimal
return ( (val/10*16) + (val%10) );
}

byte bcdToDec(byte val) {
// Convert binary coded decimal to normal decimal numbers
return ( (val/16*10) + (val%16) );
}

void printDate(){

// Reset the register pointer
Wire.beginTransmission(DS1307_ADDRESS);
Wire.write(zero);
Wire.endTransmission();
```

```
Wire.requestFrom(DS1307_ADDRESS, 7);

int seconds = bcdToDec(Wire.read());
int minutes = bcdToDec(Wire.read());
int hours = bcdToDec(Wire.read() & 0b111111); //24 hour time
int weekDays = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
int monthDays = bcdToDec(Wire.read());
int months = bcdToDec(Wire.read());
int years = bcdToDec(Wire.read());

//print the date EG 3/1/11 23:59:59
Serial.print(months);
Serial.print("/");
Serial.print(monthDays);
Serial.print("/");
Serial.print(years);
Serial.print(" ");
Serial.print(hours);
Serial.print(":");
Serial.print(minutes);
```

```

Serial.print(":");
Serial.println(seconds);
}

```

D. Código PHP do Website

Update.php

```

<?php
require("db_config.php");
$conn = mysql_connect($db_server, $db_user, $db_pass) or die ('Error
connecting to mysql!');
mysql_select_db($db_database);

$frequency = $_GET['frequency'];
$voltage_RMS_A = $_GET['voltage_RMS_A'];
$current_RMS_A = $_GET['current_RMS_A'];
$Active_Power_A = $_GET['Active_Power_A'];
$Reactive_Power_A = $_GET['Reactive_Power_A'];
$Apparent_Power_A = $_GET['Apparent_Power_A'];
$PF_A = $_GET['PF_A'];
$voltage_RMS_B = $_GET['voltage_RMS_B'];

```

```

$current_RMS_B = $_GET['current_RMS_B'];
$Active_Power_B = $_GET['Active_Power_B'];
$Reactive_Power_B = $_GET['Reactive_Power_B'];
$Apparent_Power_B = $_GET['Apparent_Power_B'];
$PF_B = $_GET['PF_B'];
$voltage_RMS_C = $_GET['voltage_RMS_C'];
$current_RMS_C = $_GET['current_RMS_C'];
$Active_Power_C = $_GET['Active_Power_C'];
$Reactive_Power_C = $_GET['Reactive_Power_C'];
$Apparent_Power_C = $_GET['Apparent_Power_C'];
$PF_C = $_GET['PF_C'];

$query = "INSERT INTO soc(frequency,
        voltage_RMS_A, current_RMS_A, Active_Power_A,
        Reactive_Power_A, Apparent_Power_A, PF_A,
        voltage_RMS_B, current_RMS_B, Active_Power_B,
        Reactive_Power_B, Apparent_Power_B, PF_B,
        voltage_RMS_C, current_RMS_C, Active_Power_C,
        Reactive_Power_C, Apparent_Power_C, PF_C)

        VALUES($frequency,
        $voltage_RMS_A, $current_RMS_A, $Active_Power_A,
        $Reactive_Power_A, $Apparent_Power_A, $PF_A,

```

```

    $voltage_RMS_B, $current_RMS_B, $Active_Power_B,
    $Reactive_Power_B, $Apparent_Power_B, $PF_B,
    $voltage_RMS_C, $current_RMS_C, $Active_Power_C,
    $Reactive_Power_C, $Apparent_Power_C, $PF_C)";

if(!@mysql_query($query))
{
echo "&Answer; SQL Error - ".mysql_error();
return;
}
?>

```

Update2.php

```

<?php
require("db_config.php");
// You need to create a table on your MySQL server database. Here it is
called 'MY_TABLE'.

    // You will also have to define fields in the table.

    //I happen to name my fields the same as my GET variables.

$conn = mysql_connect($db_server, $db_user, $db_pass) or die ('Error
connecting to mysql');
mysql_select_db($db_database);

```

```

$Active_Energy_A = $_GET['Active_Energy_A'];
$Active_Energy_B = $_GET['Active_Energy_B'];
$Active_Energy_C = $_GET['Active_Energy_C'];
$Reactive_Energy_A = $_GET['Reactive_Energy_A'];
$Reactive_Energy_B = $_GET['Reactive_Energy_B'];
$Reactive_Energy_C = $_GET['Reactive_Energy_C'];
$PActive_Energy_A = $_GET['PActive_Energy_A'];
$PActive_Energy_B = $_GET['PActive_Energy_B'];
$PActive_Energy_C = $_GET['PActive_Energy_C'];
$PReactive_Energy_A = $_GET['PReactive_Energy_A'];
$PReactive_Energy_B = $_GET['PReactive_Energy_B'];
$PReactive_Energy_C = $_GET['PReactive_Energy_C'];

$query = "INSERT INTO soc2(Active_Energy_A, Active_Energy_B,
Active_Energy_C, Reactive_Energy_A, Reactive_Energy_B,
Reactive_Energy_C, PActive_Energy_A, PActive_Energy_B,
PActive_Energy_C, PReactive_Energy_A, PReactive_Energy_B,
PReactive_Energy_C)

    VALUES($Active_Energy_A, $Active_Energy_B, $Active_Energy_C,
$Reactive_Energy_A, $Reactive_Energy_B, $Reactive_Energy_C,

```

```
$PActive_Energy_A, $PActive_Energy_B, $PActive_Energy_C,
$PReactive_Energy_A, $PReactive_Energy_B, $PReactive_Energy_C);
```

```
if(!@mysql_query($query))
{
echo "&Answer; SQL Error - ".mysql_error();
return;
}
?>
```

db_config.php

```
<?php
    $db_server = "127.10.63.2:3306";
    $db_user = "Duarte";
    $db_pass = "FCUL";
    $db_database = "energymeter";
?>
```

navigation.php

```
<nav class="top-bar" data-topbar>
```

```
    <ul class="title-area">
<li class="name"><h1><a href="index.php">Introdução</a></h1></li>
</ul>
        <ul class="title-area">
            <li class="name"><h1><a href="Power.php">Consumo
Eletrico</a></h1></li>
        </ul>
        <ul class="title-area">
            <li class="name"><h1><a
href="Energy.php">Energia</a></h1></li>
        </ul>
        <ul class="title-area">
            <li class="name"><h1><a
href="qualidade.php">Qualidade</a></h1></li>
        </ul>
        <ul class="title-area">
            <li class="name"><h1><a href="Cl.php">Custos e
Impactos</a></h1></li>
        </ul>
        <ul class="title-area">
            <li class="name"><h1><a href="cost.php">Custos
Instantâneos</a></h1></li>
        </ul>
```

```
</nav>
```

index.php

```
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html;
charset=UTF-8">
    <link rel="stylesheet" type="text/css"
href="assets/css/app.css" media="screen">
    </script>
  </head>
  <body>
    <?php include('navigation.php') ?>
    <h1>Sistema de Monitorização Trifásico da FCUL</h1>
    <p>
      Seja bem vindo, este website foi desenvolvido
no âmbito de um projeto da Faculdade de Ciências da Universidade de
Lisboa que tem por objetivo desenvolver um sistema de monitorização
de consumo e qualidade de energia com aplicações domésticas e
Industriais.</p> <p>
Este website pode ser usado quer para monitorizar o consumo de uma
instalação trifásica, quer de cargas trifásicas ou mesmo de até 3 cargas
```

individuais (equipamentos elétricos eletrodomésticos), dependendo do adaptador do sistema a ser usado.

```
</p>
```

```
<h2>Âmbito</h2>
```

```
<p>
```

Pretende-se que um utilizador possa ser capaz de avaliar os seus próprios consumos de forma fácil e com custos reduzidos. A própria criação e utilização deste website é gratuita, estando este alojado num servidor open source gratuito da Red Hat o OpenShift.

```
</p> <p>
```

Esta página encontra-se dividida em 5 separadores, os primeiros 3 para utilizadores mais conhecedores e os últimos 2 de fácil leitura para qualquer utilizador.</p> <p>

No Separador Consumo Elétrico, o consumidor tem a oportunidade de consultar os seus consumos de potência quer seja Ativa, Reativa ou Aparente bem como as correntes eficazes correspondentes que se encontram a ser consumidas em tempo Real (5 em 5 segundos). É uma forma do consumidor analisar o comportamento do seu consumo.</p> <p>

No Separador Energia, o consumidor tem oportunidade de verificar a evolução do seu consumo por fase e total, quer seja esta energia Ativa, Reativa ou Aparente. O consumidor tem assim a oportunidade de Avaliar o comportamento dos seus gastos energéticos.</p> <p>

Na Separador Qualidade, o consumidor tem a oportunidade de verificar a qualidade da sua rede elétrica ou do consumo dos seus equipamentos podendo observar a frequência

elétrica da sua rede, os valores de Tensão e os Fatores de potência associados ao consumo de cada fase. As cores presentes nos gráficos são cores que ajudam o consumidor a verificar a qualidade de cada fase, Sendo:

Verde: Boa

Laranja: Aceitável

Vermelho: má

Este índice de qualidade é estabelecido a partir de normas europeias de qualidade, não sendo necessariamente regra geral para todos os casos.

No Separador [Custos e Impactos](Cl.php), o consumidor mais habitual pode verificar a evolução dos seus gastos energéticos em termos de Preço (€), CO2 equivalente produzido e quilogramas equivalentes de petróleo consumidos em energia, desde o início da monitorização do equipamento ou desde o início do mês caso o funcionamento esteja a ser interrompido. Contudo os gráficos mostram apenas os resultados das últimas 6 horas.

No Separador [Custos Instantâneos](cost.php), o utilizador pode ver em tempo real qual será o gasto de energia em euros nesse dia, caso o consumo naquele instante se mantenha. O Resultado é apresentado em euros e é apresentado na totalidade (soma das fases) e também dividido por cada fase.

As cores ajudam o consumidor a determinar se ele está a fazer um bom ou mau consumo naquele momento. O código de cores foi baseado em dados obtidos de um estudo realizado pela Lisboa E-Nova denominado projeto Smartmeter 250 residências. Sendo que neste caso as cores indicam:

Verde - Consumo Baixo

Azul - Consumo Moderado

Laranja - Consumo Elevado

Vermelho - Consumo Muito Elevado

Conclusões

Este projeto encontra-se a dar os primeiros passos e pretende evoluir por forma a encontrar as melhores formas de ajudar a redução do consumo de energia, a sua qualidade e também a valorização da eficiência energética dos edifícios. Outro objetivo é o aumento do conhecimento do público geral no que toca a questões de poupança e gestão de energia, tornando o processo mais pessoal e divertido. No futuro também será possível com este sistema os consumidores competirem entre si com o objetivo de reduzirem os seus consumos.

tsfatcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$data = array();
if ($result = $mysqli->query("SELECT time FROM soc ORDER BY time
DESC LIMIT 0,24")){
    while ($obj = $result->fetch_object()){
        $data[] = array("label"=>$obj->time);
    }
}
echo json_encode($data);
?>

```

tsfetcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {

```

```

    echo "500";
}
$data = array();
if ($result = $mysqli->query("SELECT time FROM soc2 ORDER BY time
DESC LIMIT 0,192")){
    while ($obj = $result->fetch_object()){
        $data[] = array("label"=>$obj->time);
    }
}
echo json_encode($data);
?>

```

Power.php

```

<html>
    <head>
        <meta http-equiv="Content-type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet" type="text/css"
href="assets/css/app.css" media="screen">
        <script type="text/javascript"
src="js/fusioncharts.js"></script>

```

```
<script type="text/javascript"
src="js/themes/fusioncharts.theme.fint.js"></script>
<script type="text/javascript">
var revenueChart = new FusionCharts({
  id: "stockRealTimeChart",
  type: 'realtimeline',
  renderAt: "chartContainer",
  width: "400",
  height: "360",
  dataFormat: "json",
});
var revenueChart2 = new FusionCharts({
  id: "stockRealTimeChart",
  type: 'realtimeline',
  width: "400",
  height: "360",
  dataFormat: "json",
});
var revenueChart3 = new FusionCharts({
  id: "stockRealTimeChart",
  type: 'realtimeline',
  renderAt: "chartContainer3",
```

```
width: "400",
height: "360",
dataFormat: "json",
});
var revenueChart4 = new FusionCharts({
  type: 'msline',
  renderAt: "chartContainer4",
  width: "400",
  height: "360",
  dataFormat: "json",
});
var revenueChart5 = new FusionCharts({
  id: "stockRealTimeChart",
  type: 'realtimeline',
  renderAt: "chartContainer5",
  width: "400",
  height: "360",
  dataFormat: "json",
});
var revenueChart6 = new FusionCharts({
```

```
        id: "stockRealTimeChart",
        type: 'realtimeline',
        renderAt: "chartContainer6",
        width: "400",
        height: "360",
        dataFormat: "json",
    });

    var categorias;

    (function fetch_timestamps(){
    var tsrequest = new XMLHttpRequest();
    tsrequest.onreadystatechange = function(){
    if(tsrequest.readyState==4 && tsrequest.status==200){
categorias = (eval(tsrequest.responseText));
    }
}
tsrequest.open("POST","tsfetcher.php",true);
tsrequest.send();
})();

    (function fetch_data(){
    setTimeout(function(){
    var xmlhttp;
```

```
        xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function(){
            if (xmlhttp.readyState==4 && xmlhttp.status==200){
data={
"chart": {
            "caption": "Potência Activa consumida instantânea",
            "subCaption": " ",
            "xAxisName": "Tempo",
            "yAxisName": "Potência [ W ]",
            "refreshinterval": "5",
            "theme": "fint",
            "defaultAnimation": "0",
            "labeldisplay": "rotate",
            "showvalues":"0"
        },
        "categories": [{
            "category": categorias
        }],
        "dataset": eval(xmlhttp.responseText)
    }
    revenueChart.setChartData(data,"json");
```

```
                fetch_data();
            }
        }
        xmlhttp.open("POST","fetchActivePower.php",true);
        xmlhttp.send();
    },3000))());

    (function fetch_data3(){
        setTimeout(function(){
            var xmlhttp3;
            xmlhttp3 = new XMLHttpRequest();
            xmlhttp3.onreadystatechange = function(){
                if (xmlhttp3.readyState==4 && xmlhttp3.status==200){
                    data3={
                    "chart": {
                        "caption": "Potência Reactiva consumida instantânea",
                        "subCaption": " ",
                        "xAxisName": "Tempo",
                        "yAxisName": "Potência [ var ]",
                        "theme": "fint",
                        "defaultAnimation": "0",
                        "refreshinterval": "5",
                        "labeldisplay": "rotate",
```

```
                "showvalues": "0"
            },
            "categories": [{
                "category": categorias
            }],
            "dataset": eval(xmlhttp3.responseText)
                }
            revenueChart3.setChartData(data3,"json");
                fetch_data3();
            }
        }
        xmlhttp3.open("POST","fetchReactivePower.php",true);
        xmlhttp3.send();
    },3000))());

    (function fetch_data2(){
        setTimeout(function(){
            var xmlhttp2;
            xmlhttp2 = new XMLHttpRequest();
            xmlhttp2.onreadystatechange = function(){
                if (xmlhttp2.readyState==4 && xmlhttp2.status==200){
                    data2={
```

```

"chart": {
  "caption": "Potência Aparente consumida instantânea",
  "subCaption": " ",
  "xAxisName": "Tempo",
  "yAxisName": "Potência [ VA ]",
  "theme": "fint",
  "defaultAnimation": "0",
  "refreshinterval": "5",
  "labeldisplay": "rotate",
  "showvalues": "0"
},
"categories": [{
  "category": categorias
}],
"dataset": eval(xmlhttp2.responseText)
}
revenueChart2.setChartData(data2,"json");
fetch_data2();
}
xmlhttp2.open("POST","fetchApparentPower.php",true);

```

```

xmlhttp2.send();
},3000))());
(function fetch_data4(){
setTimeout(function(){
var xmlhttp4;
xmlhttp4 = new XMLHttpRequest();
xmlhttp4.onreadystatechange = function(){
if (xmlhttp4.readyState==4 && xmlhttp4.status==200){
data4={
"chart": {
"caption": "Corrente eficaz na fase A",
"subCaption": " ",
"xAxisName": "Tempo",
"yAxisName": "Corrente [ A ]",
"theme": "fint",
"defaultAnimation": "0",
"refreshinterval": "5",
"labeldisplay": "rotate",
"showvalues": "0"
},
"categories": [{

```

```

        "category": categorias
    }},
    "dataset": eval(xmlhttp4.responseText)
        }
    revenueChart4.setChartData(data4,"json");
    fetch_data4();
    }
}
xmlhttp4.open("POST","fetchCurrentA.php",true);
xmlhttp4.send();
},3000))());

(function fetch_data5(){
    setTimeout(function(){
        var xmlhttp5;
        xmlhttp5 = new XMLHttpRequest();
        xmlhttp5.onreadystatechange = function(){
if (xmlhttp5.readyState==4 && xmlhttp5.status==200){
data5={
"chart": {
    "caption": "Corrente eficaz na fase B",
    "subCaption": " ",

```

```

        "xAxisName": "Tempo",
        "yAxisName": "Corrente [ A ]",
        "theme": "fint",
        "defaultAnimation": "0",
        "refreshinterval": "5",
        "labeldisplay": "rotate",
        "showvalues":"0"
    },
    "categories": [{
        "category": categorias
    }},
    "dataset": eval(xmlhttp5.responseText)
        }
    revenueChart5.setChartData(data5,"json");
    fetch_data5();
    }
}
xmlhttp5.open("POST","fetchCurrentB.php",true);
xmlhttp5.send();
},3000))());

(function fetch_data6(){

```

```

        setTimeout(function(){
            var xmlhttp6;
            xmlhttp6 = new XMLHttpRequest();
            xmlhttp6.onreadystatechange = function(){
if (xmlhttp6.readyState==4 && xmlhttp6.status==200){
data6={
"chart": {
    "caption": "Corrente eficaz na fase C",
    "subCaption": " ",
    "xAxisName": "Tempo",
    "yAxisName": "Corrente [ A ]",
    "theme": "fint",
    "defaultAnimation": "0",
    "refreshinterval": "5",
    "labeldisplay": "rotate",
    "showvalues":"0"
},
"categories": [{
    "category": categorias
}],
"dataset": eval(xmlhttp6.responseText)

```

```

        }
        revenueChart6.setChartData(data6,"json");
        fetch_data6();
        }
    }
    xmlhttp6.open("POST","fetchCurrentC.php",true);
    xmlhttp6.send();
},3000));
    FusionCharts.ready(function(){
        revenueChart.render("chartContainer");
        revenueChart2.render("chartContainer2");
        revenueChart3.render("chartContainer3");
        revenueChart4.render("chartContainer4");
        revenueChart5.render("chartContainer5");
        revenueChart6.render("chartContainer6");
    });
    </script>
</head>
<body>
    <?php include('navigation.php') ?>
    <h1>Resultados das potências</h1>

```

```

<div id="chartContainer" style="float:left">Chart Placeholder</div>
<div id="chartContainer3" style="float:left">Chart Placeholder</div>
<div id="chartContainer2" >Chart Placeholder</div>
    <h1>Resultados das Correntes</h1>
<div id="chartContainer4" style="float:left">Chart Placeholder</div>
<div id="chartContainer5" style="float:left">Chart Placeholder</div>
<div id="chartContainer6" >Chart Placeholder</div>
    </body>
</html>

```

fetchActivePower.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$Active_Power_A_data = array();
$Active_Power_B_data = array();
$Active_Power_C_data = array();

```

```

if ($result = $mysqli->query("SELECT Active_Power_A, Active_Power_B,
Active_Power_C FROM soc ORDER BY time DESC LIMIT 0,24")){
    while ($obj = $result->fetch_object()){
        $Active_Power_A_data[] = array("value"=>$obj-
>Active_Power_A);
        $Active_Power_B_data[] = array("value"=>$obj-
>Active_Power_B);
        $Active_Power_C_data[] = array("value"=>$obj-
>Active_Power_C);
    }
}
$Active_Power_A_series = array(
    "seriesname"=>"Active Power A",
    "data"=>$Active_Power_A_data
);
$Active_Power_B_series = array(
    "seriesname"=>"Active Power B",
    "data"=>$Active_Power_B_data
);
$Active_Power_C_series = array(
    "seriesname"=>"Active Power C",
    "data"=>$Active_Power_C_data
);

```

```

$data = array(
    $Active_Power_A_series,
    $Active_Power_B_series,
    $Active_Power_C_series
);
echo json_encode($data);
?>

```

fetchReactivePower.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$Reactive_Power_A_data = array();
$Reactive_Power_B_data = array();
$Reactive_Power_C_data = array();
if ($result = $mysqli->query("SELECT Reactive_Power_A,
Reactive_Power_B, Reactive_Power_C FROM soc ORDER BY time DESC
LIMIT 0,24")){

```

```

    while ($obj = $result->fetch_object()){
        $Reactive_Power_A_data[] = array("value"=>$obj-
>Reactive_Power_A);
        $Reactive_Power_B_data[] = array("value"=>$obj-
>Reactive_Power_B);
        $Reactive_Power_C_data[] = array("value"=>$obj-
>Reactive_Power_C);
    }
}
$Reactive_Power_A_series = array(
    "seriesname"=>"Reactive Power A",
    "data"=>$Reactive_Power_A_data
);
$Reactive_Power_B_series = array(
    "seriesname"=>"Reactive Power B",
    "data"=>$Reactive_Power_B_data
);
$Reactive_Power_C_series = array(
    "seriesname"=>"Reactive Power C",
    "data"=>$Reactive_Power_C_data
);
$data = array(

```

```

    $Reactive_Power_A_series,
    $Reactive_Power_B_series,
    $Reactive_Power_C_series
);
echo json_encode($data);
?>

```

fetchApparentPower.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$Apparent_Power_A_data = array();
$Apparent_Power_B_data = array();
$Apparent_Power_C_data = array();
if ($result = $mysqli->query("SELECT Apparent_Power_A,
Apparent_Power_B, Apparent_Power_C FROM soc ORDER BY time
DESC LIMIT 0,24")){
    while ($obj = $result->fetch_object()){

```

```

        $Apparent_Power_A_data[] = array("value"=>$obj-
>Apparent_Power_A);
        $Apparent_Power_B_data[] = array("value"=>$obj-
>Apparent_Power_B);
        $Apparent_Power_C_data[] = array("value"=>$obj-
>Apparent_Power_C);
    }
}
$Apparent_Power_A_series = array(
    "seriesname"=>"Apparent Power A",
    "data"=>$Apparent_Power_A_data
);
$Apparent_Power_B_series = array(
    "seriesname"=>"Apparent Power B",
    "data"=>$Apparent_Power_B_data
);
$Apparent_Power_C_series = array(
    "seriesname"=>"Apparent Power C",
    "data"=>$Apparent_Power_C_data
);
$data = array(
    $Apparent_Power_A_series,

```

```

        $Apparent_Power_B_series,
        $Apparent_Power_C_series
    );
    echo json_encode($data);
?>

```

fetchCurrentA.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$current_RMS_A_data = array();
if ($result = $mysqli->query("SELECT current_RMS_A FROM soc ORDER
BY time DESC LIMIT 0,24")){
    while ($obj = $result->fetch_object()){
        $current_RMS_A_data[] = array("value"=>$obj-
>current_RMS_A);
    }
}

```

```

$current_RMS_A_series = array(
    "seriesname"=>"Current RMS A",
    "data"=>$current_RMS_A_data
);
$data = array(
    $current_RMS_A_series,
);
echo json_encode($data);
?>

```

fetchCurrentB.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$current_RMS_B_data = array();
if ($result = $mysqli->query("SELECT current_RMS_B FROM soc ORDER
BY time DESC LIMIT 0,24")){
    while ($obj = $result->fetch_object()){

```

```

                $current_RMS_B_data[] = array("value"=>$obj-
>current_RMS_B);
            }
        }
    }
    $current_RMS_B_series = array(
        "seriesname"=>"Current RMS B",
        "data"=>$current_RMS_B_data
    );
    $data = array(
        $current_RMS_B_series,
    );
    echo json_encode($data);
?>

```

fetchCurrentC.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}

```

```

$current_RMS_C_data = array();
if ($result = $mysqli->query("SELECT current_RMS_C FROM soc ORDER
BY time DESC LIMIT 0,24")){
    while ($obj = $result->fetch_object()){
        $current_RMS_C_data[] = array("value"=>$obj-
>current_RMS_C);
    }
}
$current_RMS_C_series = array(
    "seriesname"=>"Current RMS C",
    "data"=>$current_RMS_C_data
);
$data = array(
    $current_RMS_C_series,
);
echo json_encode($data);
?>

```

Energy.php

```

<html>
    <head>

```

```
<meta http-equiv="Content-type" content="text/html;
charset=UTF-8">
<link rel="stylesheet" type="text/css"
href="assets/css/app.css" media="screen">
<script type="text/javascript"
src="js/fusioncharts.js"></script>
<script type="text/javascript"
src="js/themes/fusioncharts.theme.fint.js"></script>
<script type="text/javascript">
var revenueChart = new FusionCharts({
    type: "zoomline",
    renderAt: "chartContainer",
    width: "400",
    height: "360",
    dataFormat: "json",
});
var revenueChart2 = new FusionCharts({
    type: 'zoomline',
    renderAt: "chartContainer2",
    width: "400",
    height: "360",
    dataFormat: "json",
});
```

```
var revenueChart3 = new FusionCharts({
    type: "zoomline",
    renderAt: "chartContainer3",
    width: "400",
    height: "360",
    dataFormat: "json",
});
var revenueChart4 = new FusionCharts({
    type: "zoomline",
    renderAt: "chartContainer4",
    width: "400",
    height: "360",
    dataFormat: "json",
});
var revenueChart5 = new FusionCharts({
    type: "zoomline",
    renderAt: "chartContainer5",
    width: "400",
    height: "360",
    dataFormat: "json",
});
```

```
var revenueChart6 = new FusionCharts({
    type: 'zoomline',
    renderAt: "chartContainer6",
    width: "400",
    height: "360",
    dataFormat: "json",
});
var revenueChart7 = new FusionCharts({
    type: "zoomline",
    renderAt: "chartContainer7",
    width: "400",
    height: "360",
    dataFormat: "json",
});
var revenueChart8 = new FusionCharts({
    type: "zoomline",
    renderAt: "chartContainer8",
    width: "400",
    height: "360",
    dataFormat: "json",
});
```

```
var categorias;
(function fetch_timestamps(){
    var tsrequest = new XMLHttpRequest();
    tsrequest.onreadystatechange = function(){
if(tsrequest.readyState==4 && tsrequest.status==200){
    categorias = (eval(tsrequest.responseText));
    }
    }
    tsrequest.open("POST","tsfetcher2.php",true);
    tsrequest.send();
})();
(function fetch_data(){
    setTimeout(function(){
        var xmlhttp;
        xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function(){
if (xmlhttp.readyState==4 && xmlhttp.status==200){
data = {
"chart": {
    "caption": "Consumo mensal de Energia na fase A",
    "subCaption": "",
```

```

        "xAxisName": "Tempo",
        "yAxisName": "Energia [ kWh e kvar ]",
        "defaultAnimation": "0",
        "forceAxisLimits" : "1",
        "theme": "fint"
    },
    "categories": [{
        "category": categorias
    }],
    "dataset": eval(xmlhttp.responseText)
    }
    revenueChart.setChartData(data,"json");
    fetch_data();
    }
    xmlhttp.open("POST","fetchEnergyA.php",true);
    xmlhttp.send();
},3000)}}());

(function fetch_data3(){
    setTimeout(function(){
        var xmlhttp3;

```

```

        xmlhttp3 = new XMLHttpRequest();
        xmlhttp3.onreadystatechange = function(){
            if (xmlhttp3.readyState==4 && xmlhttp3.status==200){
                data3={
                    "chart": {
                        "caption": "Consumo mensal de Energia na fase B",
                        "subcaption": "",
                        "xaxisname": "Tempo",
                        "yaxisname": "Energia [ kWh e kvar ]",
                        "defaultAnimation": "0",
                        "forceAxisLimits" : "1",
                        "theme": "fint"
                    },
                    "categories": [{
                        "category": categorias
                    }],
                    "dataset": eval(xmlhttp3.responseText)
                }
                revenueChart3.setChartData(data3,"json");
                fetch_data3();
            }
        }
    }
}

```

```

    }
    xmlhttp3.open("POST","fetchEnergyB.php",true);
    xmlhttp3.send();
},3000)}}());

(function fetch_data2(){
setTimeout(function(){
    var xmlhttp2;
    xmlhttp2 = new XMLHttpRequest();
    xmlhttp2.onreadystatechange = function(){
if (xmlhttp2.readyState==4 && xmlhttp2.status==200){
data2={
"chart": {
    "caption": "Consumo mensal de Energia na fase C",
    "subcaption": "",
    "xaxisname": "Tempo",
    "yaxisname": "energia [ kWh kvar ]",
    "defaultAnimation": "0",
    "forceAxisLimits" : "1",
    "theme": "fint"
    },
"categories": [{

```

```

"category": categorias
}],
"dataset": eval(xmlhttp2.responseText)
    }
revenueChart2.setChartData(data2,"json");
    fetch_data2();
    }
    xmlhttp2.open("POST","fetchEnergyC.php",true);
    xmlhttp2.send();
},3000)}}());

(function fetch_data4(){
setTimeout(function(){
    var xmlhttp4;
    xmlhttp4 = new XMLHttpRequest();
    xmlhttp4.onreadystatechange = function(){
if (xmlhttp4.readyState==4 && xmlhttp4.status==200){
data4={
"chart": {
    "caption": "Consumo mensal de Energia Total",
    "subcaption": "",

```

```

        "xaxisname": "Tempo",
        "yaxisname": "Energia [ kWh e kvar ]",
        "defaultAnimation": "0",
        "forceAxisLimits" : "1",
        "theme": "fint"
    },

    "categories": [{
    "category": categorias
}],

    "dataset": eval(xmlhttp4.responseText)
    }
    revenueChart4.setChartData(data4,"json");
    fetch_data4();
    }
}

xmlhttp4.open("POST","fetchEnergytotal.php",true);
    xmlhttp4.send();
},3000)){});

(function fetch_data5(){
    setTimeout(function(){
        var xmlhttp5;

```

```

        xmlhttp5 = new XMLHttpRequest();
        xmlhttp5.onreadystatechange = function(){
    if (xmlhttp5.readyState==4 && xmlhttp5.status==200){
    data5={
    "chart": {
        "caption": "Produção mensal de Energia na fase A",
        "subcaption": "",
        "xaxisname": "Tempo",
        "yaxisname": "Energia [ kWh e kvar ]",
        "defaultAnimation": "0",
        "forceAxisLimits" : "1",
        "theme": "fint"
    },

    "categories": [{
        "category": categorias
    }],

    "dataset": eval(xmlhttp5.responseText)
    }
    revenueChart5.setChartData(data5,"json");
    fetch_data5();
    }

```

```

    }
    xmlhttp5.open("POST","fetchPEnergyA.php",true);
        xmlhttp5.send();
},3000)}}());

(function fetch_data6(){
    setTimeout(function(){
        var xmlhttp6;
        xmlhttp6 = new XMLHttpRequest();
        xmlhttp6.onreadystatechange = function(){
if (xmlhttp6.readyState==4 && xmlhttp6.status==200){
data6={
"chart": {
    "caption": "Produção mensal de Energia na fase B",
        "subcaption": "",
        "xaxisname": "Tempo",
        "yaxisname": "Energia [ kWh e kvar]",
        "defaultAnimation": "0",
        "forceAxisLimits" : "1",
        "theme": "fint"
                },
    "categories": [{

```

```

"category": categorias
    ]],
    "dataset": eval(xmlhttp6.responseText)
        }
    revenueChart6.setChartData(data6,"json");
        fetch_data6();
        }
    }
    xmlhttp6.open("POST","fetchPEnergyB.php",true);
        xmlhttp6.send();
},3000)}}());

(function fetch_data7(){
    setTimeout(function(){
        var xmlhttp7;
        xmlhttp7 = new XMLHttpRequest();
        xmlhttp7.onreadystatechange = function(){
if (xmlhttp7.readyState==4 && xmlhttp7.status==200){
data7={
"chart": {
    "caption": "Produção mensal de Energia na fase C",
        "subcaption": "",

```

```

        "xaxisname": "Tempo",
        "yaxisname": "Energia [ kWh e kvar]",
        "defaultAnimation": "0",
        "forceAxisLimits" : "1",
        "theme": "fint"
    },

    "categories": [{
    "category": categorias
}],

    "dataset": eval(xmlhttp7.responseText)
    }
    revenueChart7.setChartData(data7,"json");
    fetch_data7();
    }
}

xmlhttp7.open("POST","fetchPEnergyC.php",true);
xmlhttp7.send();
},3000))});

(function fetch_data8(){
    setTimeout(function(){
        var xmlhttp8;

```

```

        xmlhttp8 = new XMLHttpRequest();
        xmlhttp8.onreadystatechange = function(){
    if (xmlhttp8.readyState==4 && xmlhttp8.status==200){
    data8={
    "chart": {
        "caption": "Produção mensal de Energia Total",
        "subcaption": "",
        "xaxisname": "Tempo",
        "yaxisname": "Energia [ kWh e kvar ]",
        "defaultAnimation": "0",
        "forceAxisLimits" : "1",
        "theme": "fint"
    },

    "categories": [{
    "category": categorias
}],

    "dataset": eval(xmlhttp8.responseText)
    }
    revenueChart8.setChartData(data8,"json");
    fetch_data8();
    }

```

```

    }
    xmlhttp8.open("POST","fetchPEnergytotal.php",true);
    xmlhttp8.send();
},3000)}}());

    FusionCharts.ready(function(){
    revenueChart.render("chartContainer");
    revenueChart2.render("chartContainer2");
    revenueChart3.render("chartContainer3");
    revenueChart4.render("chartContainer4");
    revenueChart5.render("chartContainer5");
    revenueChart6.render("chartContainer6");
    revenueChart7.render("chartContainer7");
    revenueChart8.render("chartContainer8");
});

    </script>
</head>
<body>
<?php include('navigation.php') ?>
    <h1>Resultados da Energia Consumida por fase</h1>
<div id="chartContainer" style="float:left">Chart Placeholder</div>
<div id="chartContainer3" style="float:left">Chart Placeholder</div>

```

```

<div id="chartContainer2" >Chart Placeholder</div>
    <h1>Resultados da Energia Total Consumida</h1>
<div id="chartContainer4" >Chart Placeholder</div>
    <h1>Resultados da Energia Enviada por fase</h1>
<div id="chartContainer5" style="float:left">Chart Placeholder</div>
<div id="chartContainer6" style="float:left">Chart Placeholder</div>
<div id="chartContainer7" >Chart Placeholder</div>
    <h1>Resultados da Energia Total Enviada</h1>
<div id="chartContainer8" style="float:left">Chart Placeholder</div>
    </body>
</html>

```

fetchEnergyA.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$Active_Energy_A_data = array();
$Reactive_Energy_A_data = array();

```

```

if ($result = $mysqli->query("SELECT Active_Energy_A,
Reactive_Energy_A FROM soc2 ORDER BY time DESC LIMIT 0,192
")){
    while ($obj = $result->fetch_object()){
        $Active_Energy_A_data[] = array("value"=>$obj-
>Active_Energy_A*0.001);
        $Reactive_Energy_A_data[] = array("value"=>$obj-
>Reactive_Energy_A*0.001);
    }
}
$Active_Energy_A_series = array(
    "seriesname"=>"Active Energy A",
    "data"=>$Active_Energy_A_data
);
$Reactive_Energy_A_series = array(
    "seriesname"=>"Reactive Energy A",
    "data"=>$Reactive_Energy_A_data
);
$data = array(
    $Active_Energy_A_series,
    $Reactive_Energy_A_series
);

```

```

echo json_encode($data);
?>

```

fetchEnergyB.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$Active_Energy_B_data = array();
$Reactive_Energy_B_data = array();
if ($result = $mysqli->query("SELECT Active_Energy_B,
Reactive_Energy_B FROM soc2 ORDER BY time DeSC LIMIT 0,192
")){
    while ($obj = $result->fetch_object()){
        $Active_Energy_B_data[] = array("value"=>$obj-
>Active_Energy_B*0.001);
        $Reactive_Energy_B_data[] = array("value"=>$obj-
>Reactive_Energy_B*0.001);
    }
}

```

```

$Active_Energy_B_series = array(
    "seriesname"=>"Active Energy B",
    "data"=>$Active_Energy_B_data
);
$Reactive_Energy_B_series = array(
    "seriesname"=>"Reactive Energy B",
    "data"=>$Reactive_Energy_B_data
);
$data = array(
    $Active_Energy_B_series,
    $Reactive_Energy_B_series
);
echo json_encode($data);
?>

```

fetchEnergyC.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}

```

```

}
$Active_Energy_C_data = array();
$Reactive_Energy_C_data = array();
if ($result = $mysqli->query("SELECT Active_Energy_C,
Reactive_Energy_C FROM soc2 ORDER BY time DESC LIMIT 0,192")){
    while ($obj = $result->fetch_object()){
        $Active_Energy_C_data[] = array("value"=>$obj-
>Active_Energy_C*0.001);
        $Reactive_Energy_C_data[] = array("value"=>$obj-
>Reactive_Energy_C*0.001);
    }
}
$Active_Energy_C_series = array(
    "seriesname"=>"Active Energy C",
    "data"=>$Active_Energy_C_data
);
$Reactive_Energy_C_series = array(
    "seriesname"=>"Reactive Energy C",
    "data"=>$Reactive_Energy_C_data
);
$data = array(
    $Active_Energy_C_series,

```

```

        $Reactive_Energy_C_series
    );
    echo json_encode($data);
?>

```

fetchEnergytotal.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$Active_Energy_total_data = array();
if ($result = $mysqli->query("SELECT time, Active_Energy_A,
Active_Energy_B, Active_Energy_C, Reactive_Energy_A,
Reactive_Energy_B, Reactive_Energy_C FROM soc2 ORDER BY time
DESC LIMIT 0,192")){
    while ($obj = $result->fetch_object()){
        $valor1=$obj->Active_Energy_A+$obj-
>Active_Energy_B+$obj->Active_Energy_C;
        $valor2=$obj->Reactive_Energy_A*0.001+$obj-
>Reactive_Energy_B*0.001+$obj->Reactive_Energy_C*0.001;

```

```

        $Active_Energy_total_data[] =
array("value"=>$valor1*0.001);
        $Reactive_Energy_total_data[] =
array("value"=>$valor2);
    }
}
$Active_Energy_total_series = array(
    "seriesname"=>"Consumed Active Energy total",
    "data"=>$Active_Energy_total_data
);
$Reactive_Energy_total_series = array(
    "seriesname"=>"Inductive Reactive Energy total",
    "data"=>$Reactive_Energy_total_data
);
$data = array(
    $Active_Energy_total_series,
    $Reactive_Energy_total_series
);
echo json_encode($data);
?>

```

fetchPEnergyA.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$PActive_Energy_A_data = array();
$PReactive_Energy_A_data = array();
if ($result = $mysqli->query("SELECT PActive_Energy_A,
PReactive_Energy_A FROM soc2 ORDER BY time DESC LIMIT 0,192
")){
    while ($obj = $result->fetch_object()){
        $PActive_Energy_A_data[] = array("value"=>$obj-
>PActive_Energy_A*0.001);
        $PReactive_Energy_A_data[] = array("value"=>$obj-
>PReactive_Energy_A*0.001);
    }
}
$PActive_Energy_A_series = array(
    "seriesname"=>"PActive Energy A",
    "data"=>$PActive_Energy_A_data
);

```

```

$PReactive_Energy_A_series = array(
    "seriesname"=>"PReactive Energy A",
    "data"=>$PReactive_Energy_A_data
);
$data = array(
    $PActive_Energy_A_series,
    $PReactive_Energy_A_series
);
echo json_encode($data);
?>

```

fetchPEnergyB.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$PActive_Energy_B_data = array();
$PReactive_Energy_B_data = array();

```

```

if ($result = $mysqli->query("SELECT PActive_Energy_B,
PReactive_Energy_B FROM soc2 ORDER BY time DESC LIMIT 0,192
")){
    while ($obj = $result->fetch_object()){
        $PActive_Energy_B_data[] = array("value"=>$obj-
>PActive_Energy_B*0.001);
        $PReactive_Energy_B_data[] = array("value"=>$obj-
>PReactive_Energy_B*0.001);
    }
}
$PActive_Energy_B_series = array(
    "seriesname"=>"PActive Energy B",
    "data"=>$PActive_Energy_B_data
);
$PReactive_Energy_B_series = array(
    "seriesname"=>"PReactive Energy B",
    "data"=>$PReactive_Energy_B_data
);
$data = array(
    $PActive_Energy_B_series,
    $PReactive_Energy_B_series
);

```

```

echo json_encode($data);
?>

```

fetchPEnergyC.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$PActive_Energy_C_data = array();
$PReactive_Energy_C_data = array();
if ($result = $mysqli->query("SELECT PActive_Energy_C,
PReactive_Energy_C FROM soc2 ORDER BY time DESC LIMIT 0,192")){
    while ($obj = $result->fetch_object()){
        $PActive_Energy_C_data[] = array("value"=>$obj-
>PActive_Energy_C*0.001);
        $PReactive_Energy_C_data[] = array("value"=>$obj-
>PReactive_Energy_C*0.001);
    }
}
$PActive_Energy_C_series = array(

```

```

        "seriesname"=>"PActive Energy C",
        "data"=>$PActive_Energy_C_data
    );
    $PReactive_Energy_C_series = array(
        "seriesname"=>"PReactive Energy C",
        "data"=>$PReactive_Energy_C_data
    );
    $data = array(
        $PActive_Energy_C_series,
        $PReactive_Energy_C_series
    );
    echo json_encode($data);
?>

```

fetchPEnergytotal.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}

```

```

$PActive_Energy_total_data = array();
if ($result = $mysqli->query("SELECT time, PActive_Energy_A,
PActive_Energy_B, PActive_Energy_C, PReactive_Energy_A,
PReactive_Energy_B, PReactive_Energy_C FROM soc2 ORDER BY time
DESC LIMIT 0,192")){
    while ($obj = $result->fetch_object()){
        $valor1=$obj->PActive_Energy_A*0.001+$obj-
>PActive_Energy_B*0.001+$obj->PActive_Energy_C*0.001;
        $valor2=$obj->PReactive_Energy_A+$obj-
>PReactive_Energy_B+$obj->PReactive_Energy_C;
        $PActive_Energy_total_data[] =
array("value"=>$valor1);
        $PReactive_Energy_total_data[] =
array("value"=>$valor2*0.001);
    }
}
$PActive_Energy_total_series = array(
    "seriesname"=>"Produced Active Energy total",
    "data"=>$PActive_Energy_total_data
);
$PReactive_Energy_total_series = array(
    "seriesname"=>"Capacitive Reactive Energy total",
    "data"=>$PReactive_Energy_total_data

```

```

);
$data = array(
    $PActive_Energy_total_series,
    $PReactive_Energy_total_series
);
echo json_encode($data);
?>

```

qualidade.php

```

<html>
  <head>
    <meta http-equiv="Content-type" content="text/html;
charset=UTF-8">
    <link rel="stylesheet" type="text/css"
href="assets/css/app.css" media="screen">
    <script type="text/javascript"
src="js/fusioncharts.js"></script>
    <script type="text/javascript"
src="js/themes/fusioncharts.theme.zune.js"></script>
    <script type="text/javascript">
      var revenueChart1 = new FusionCharts({
        type: "angulargauge",

```

```

        renderAt: "chartContainer1",
          animation: "0",
          width: "400",
          height: "240",
          dataFormat: "json",
        });
      var revenueChart2 = new FusionCharts({
        type: "angulargauge",
        renderAt: "chartContainer2",
          animation: "0",
          width: "400",
          height: "240",
          dataFormat: "json",
        });
      var revenueChart3 = new FusionCharts({
        type: "angulargauge",
        renderAt: "chartContainer3",
          animation: "0",
          width: "400",
          height: "240",
          dataFormat: "json",

```

```
});  
var revenueChart4 = new FusionCharts({  
    type: "angulargauge",  
    renderAt: "chartContainer4",  
    animation: "0",  
    width: "400",  
    height: "240",  
    dataFormat: "json",  
});  
var revenueChart5 = new FusionCharts({  
    type: "angulargauge",  
    renderAt: "chartContainer5",  
    animation: "0",  
    width: "400",  
    height: "240",  
    dataFormat: "json",  
});  
var revenueChart6 = new FusionCharts({  
    type: "angulargauge",  
    renderAt: "chartContainer6",  
    animation: "0",
```

```
width: "400",  
height: "240",  
dataFormat: "json",  
});  
var revenueChart7 = new FusionCharts({  
    type: "angulargauge",  
    renderAt: "chartContainer7",  
    animation: "0",  
    width: "400",  
    height: "240",  
    dataFormat: "json",  
});  
(function fetch_data1(){  
    setTimeout(function(){  
        var xmlhttp1;  
        xmlhttp1 = new XMLHttpRequest();  
        xmlhttp1.onreadystatechange = function(){  
            if (xmlhttp1.readyState==4 && xmlhttp1.status==200){  
data1={  
"chart": {  
    "caption": "Factor de Potência, Fase A",
```

```
"subCaption": " ",
"lowerLimit": "0.5",
"upperLimit": "1",
"theme": "zune",
"defaultAnimation": "0"
},
"colorRange": {
"color": [
{
"minValue": "0.5",
"maxValue": "0.85",
"code": "#e44a00"
},
{
"minValue": "0.85",
"maxValue": "0.93",
"code": "#f8bd19"
},
{
"minValue": "0.93",
"maxValue": "1",
```

```
"code": "#6baa01"
}
]
},
"dials": {
"dial": [{
"value": eval(xmlhttp1.responseText)
}
}
}

revenueChart1.setChartData(data1,"json");
fetch_data1();
}
}
xmlhttp1.open("POST","pfa_fetcher.php",true);
xmlhttp1.send();
},3000)}());
FusionCharts.ready(function(){
revenueChart1.render("chartContainer1");
});
(function fetch_data2(){
```



```
    }  
    xmlhttp2.open("POST","pfb_fetcher.php",true);  
    xmlhttp2.send();  
},3000)}}());  
    FusionCharts.ready(function(){  
    revenueChart2.render("chartContainer2");  
});  
    (function fetch_data3(){  
    setTimeout(function(){  
        var xmlhttp3;  
        xmlhttp3 = new XMLHttpRequest();  
        xmlhttp3.onreadystatechange = function(){  
if (xmlhttp3.readyState==4 && xmlhttp3.status==200){  
data3={  
"chart": {  
    "caption": "Factor de Potência, Fase C",  
    "subCaption": " ",  
    "lowerLimit": "0.5",  
    "upperLimit": "1",  
    "theme": "zune",  
    "defaultAnimation": "0"
```

```
    },  
    "colorRange": {  
    "color": [  
    {  
    "minValue": "0.5",  
    "maxValue": "0.85",  
    "code": "#e44a00"  
    },  
    {  
    "minValue": "0.85",  
    "maxValue": "0.93",  
    "code": "#f8bd19"  
    },  
    {  
    "minValue": "0.93",  
    "maxValue": "1",  
    "code": "#6baa01"  
    }  
    ]  
    },  
    "dials": {
```

```
"dial": {  
    "value": eval(xmlhttp3.responseText)  
    }  
}  
revenueChart3.setChartData(data3,"json");  
    fetch_data3();  
}  
xmlhttp3.open("POST","pfc_fetcher.php",true);  
xmlhttp3.send();  
,3000));  
    FusionCharts.ready(function(){  
        revenueChart3.render("chartContainer3");  
    });  
    (function fetch_data4(){  
        setTimeout(function(){  
            var xmlhttp4;  
            xmlhttp4 = new XMLHttpRequest();  
            xmlhttp4.onreadystatechange = function(){  
                if (xmlhttp4.readyState==4 && xmlhttp4.status==200){
```

```
data4={  
"chart": {  
    "caption": "Voltagem, na Fase A",  
    "subCaption": "[Volts]",  
    "lowerLimit": "190",  
    "upperLimit": "260",  
    "theme": "zune",  
    "defaultAnimation": "0"  
    },  
    "colorRange": {  
    "color": [  
    {  
        "minValue": "190",  
        "maxValue": "207",  
        "code": "#e44a00"  
    },  
    {  
        "minValue": "207",  
        "maxValue": "225",  
        "code": "#f8bd19"  
    },  
    ]  
    }  
};
```

```

    {
      "minValue": "225",
      "maxValue": "235",
      "code": "#6baa01"
    },
    {
      "minValue": "235",
      "maxValue": "253",
      "code": "#f8bd19"
    },
    {
      "minValue": "253",
      "maxValue": "260",
      "code": "#e44a00"
    }
  ]
},
"dials": {
  "dial": [{
    "value": eval(xmlhttp4.responseText)
  }}
}

```

```

    }
  }
  revenueChart4.setChartData(data4,"json");
  fetch_data4();
}
}
xmlhttp4.open("POST","voltage_RMS_A_fetcher.php",true);
xmlhttp4.send();
},3000))());
FusionCharts.ready(function(){
  revenueChart4.render("chartContainer4");
});
(function fetch_data5(){
  setTimeout(function(){
    var xmlhttp5;
    xmlhttp5 = new XMLHttpRequest();
    xmlhttp5.onreadystatechange = function(){
      if (xmlhttp5.readyState==4 && xmlhttp5.status==200){
        data5={
          "chart": {
            "caption": "Voltagem, na Fase B",

```

```
"subCaption": "[Volts]",  
"lowerLimit": "190",  
"upperLimit": "260",  
"theme": "zune",  
"defaultAnimation": "0"  
},  
"colorRange": {  
"color": [  
{  
"minValue": "190",  
"maxValue": "207",  
"code": "#e44a00"  
},  
{  
"minValue": "207",  
"maxValue": "225",  
"code": "#f8bd19"  
},  
{  
"minValue": "225",  
"maxValue": "235",
```

```
"code": "#6baa01"  
},  
{  
"minValue": "235",  
"maxValue": "253",  
"code": "#f8bd19"  
},  
{  
"minValue": "253",  
"maxValue": "260",  
"code": "#e44a00"  
}  
]  
},  
"dials": {  
"dial": [{  
"value": eval(xmlhttp5.responseText)  
}]  
}  
}  
revenueChart5.setChartData(data5,"json");
```

```
                fetch_data5();
            }
        }
        xmlhttp5.open("POST","voltage_RMS_B_fetcher.php",true);
        xmlhttp5.send();
    },3000))());
    FusionCharts.ready(function(){
        revenueChart5.render("chartContainer5");
    });
    (function fetch_data6(){
        setTimeout(function(){
            var xmlhttp6;
            xmlhttp6 = new XMLHttpRequest();
            xmlhttp6.onreadystatechange = function(){
                if (xmlhttp6.readyState==4 && xmlhttp6.status==200){
                    data6={
                        "chart": {
                            "caption": "Voltagem, na Fase C",
                            "subCaption": "[volts]",
                            "lowerLimit": "190",
                            "upperLimit": "260",
```

```
                            "theme": "zune",
                            "defaultAnimation": "0"
                        },
                        "colorRange": {
                            "color": [
                                {
                                    "minValue": "190",
                                    "maxValue": "207",
                                    "code": "#e44a00"
                                },
                                {
                                    "minValue": "207",
                                    "maxValue": "225",
                                    "code": "#f8bd19"
                                },
                                {
                                    "minValue": "225",
                                    "maxValue": "235",
                                    "code": "#6baa01"
                                }
                            ]
                        }
                    }
                }
            }
        }
    }
}
```

```

        "minValue": "235",
        "maxValue": "253",
        "code": "#f8bd19"
    },
    {
        "minValue": "253",
        "maxValue": "260",
        "code": "#e44a00"
    }
]
},
"dials": {
    "dial": [{
        "value": eval(xmlhttp6.responseText)
    }
    ]
}
revenueChart6.setChartData(data6,"json");
fetch_data6();
}
}

```

```

xmlhttp6.open("POST","voltage_RMS_C_fetcher.php",true);
xmlhttp6.send();
},3000)}}());
FusionCharts.ready(function(){
revenueChart6.render("chartContainer6");
});
(function fetch_data7(){
setTimeout(function(){
var xmlhttp7;
xmlhttp7 = new XMLHttpRequest();
xmlhttp7.onreadystatechange = function(){
if (xmlhttp7.readyState==4 && xmlhttp7.status==200){
data7={
"chart": {
"caption": "Frequência Composta",
"subCaption": "[Hz]",
"lowerLimit": "46.90",
"upperLimit": "52.10",
"theme": "zune",
"defaultAnimation": "0"
},

```

```
"colorRange": {  
  "color": [  
    {  
      "minValue": "46.90",  
      "maxValue": "47.00",  
      "code": "#e44a00"  
    },  
    {  
      "minValue": "47.00",  
      "maxValue": "49.95",  
      "code": "#f8bd19"  
    },  
    {  
      "minValue": "49.95",  
      "maxValue": "50.05",  
      "code": "#6baa01"  
    },  
    {  
      "minValue": "50.05",  
      "maxValue": "52.00",  
      "code": "#f8bd19"
```

```
    },  
    {  
      "minValue": "52.00",  
      "maxValue": "50.10",  
      "code": "#e44a00"  
    }  
  ]  
},  
"dials": {  
  "dial": [{  
    "value": eval(xmlhttp7.responseText)  
  }  
]  
}  
revenueChart7.setChartData(data7,"json");  
fetch_data7();  
}  
xmlhttp7.open("POST","frequency.php",true);  
xmlhttp7.send();  
},3000))());
```

```

        FusionCharts.ready(function(){
            revenueChart7.render("chartContainer7");
        });
    </script>
</head>
<body>
    <?php include('navigation.php') ?>
    <h1>Resultados dos Factores de Potência:</h1>
<div id="chartContainer1" style="float:left">Chart Placeholder</div>
<div id="chartContainer2" style="float:left">Chart Placeholder</div>
<div id="chartContainer3" >Chart Placeholder</div>
    <h1>Resultados da Tensão:</h1>
<div id="chartContainer4" style="float:left">Chart Placeholder</div>
<div id="chartContainer5" style="float:left">Chart Placeholder</div>
<div id="chartContainer6" >Chart Placeholder</div>
    <h1>Resultados da frequência Composta:</h1>
<div id="chartContainer7" style="float:left">Chart Placeholder</div>
</body>
</html>

```

pfa_fetcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT pf_a FROM soc ORDER BY time
DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->pf_a;
    }
}
echo json_encode($data);
?>

```

pfb_fetcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}

```

```

}
if ($result = $mysqli->query("SELECT pf_b FROM soc ORDER BY time
DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->pf_b;
    }
}
echo json_encode($data);
?>

```

Pfc_fetcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT pf_c FROM soc ORDER BY time
DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->pf_c;
    }
}
echo json_encode($data);
?>

```

```

}
}
echo json_encode($data);
?>

```

Voltage_RMS_A_fetcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT voltage_RMS_A FROM soc ORDER
BY time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->voltage_RMS_A;
    }
}
echo json_encode($data);
?>

```

Voltage_RMS_B_fetcher.php

```
<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT voltage_RMS_B FROM soc ORDER
BY time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->voltage_RMS_B;
    }
}
echo json_encode($data);
?>
```

Voltage_RMS_C_fetcher.php

```
<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
```

```
    echo "500";
}
if ($result = $mysqli->query("SELECT voltage_RMS_C FROM soc ORDER
BY time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->voltage_RMS_C;
    }
}
echo json_encode($data);
?>
```

frequency.php

```
<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT frequency FROM soc ORDER BY
time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
```

```

        $data = $obj->frequency;
    }
}
echo json_encode($data);
?>

```

Cl.php

```

<html>
    <head>
        <meta http-equiv="Content-type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet" type="text/css"
href="assets/css/app.css" media="screen">
        <script type="text/javascript"
src="js/fusioncharts.js"></script>
        <script type="text/javascript"
src="js/themes/fusioncharts.theme.zune.js"></script>
        <script type="text/javascript">
            var Energy1Chart = new FusionCharts({
                type: 'line',
                renderAt: "Energy1ChartContainer",
                width: "750",

```

```

                height: "450",
                dataFormat: "json"
            });
            var Energy2Chart = new FusionCharts({
                type: 'line',
                renderAt: "Energy2ChartContainer",
                width: "750",
                height: "450",
                dataFormat: "json"
            });
            var Energy3Chart = new FusionCharts({
                type: 'line',
                renderAt: "Energy3ChartContainer",
                width: "750",
                height: "450",
                dataFormat: "json"
            });
            FusionCharts.ready(function(){
                Energy1Chart.render("Energy1ChartContainer");
            });
            (function fetch_Energy1_data(){

```

```
setTimeout(function(){
    var Energy1Request = new XMLHttpRequest();
    Energy1Request.onreadystatechange = function(){
if (Energy1Request.readyState==4 && Energy1Request.status==200){
data={
"chart": {
    "caption": "ENDESA: TARIFA SELEÇÃO – OPÇÃO 1",
    "subCaption": "Pot. contratada: 34,5 kVA",
    "xAxisName": "Tempo",
    "yAxisName": "Euros (€)",
    "refreshinterval": "5",
    "theme": "fint",
    "animation": "0",
    "showvalues":"0"
    },
    "data":eval(Energy1Request.responseText)
    }
    Energy1Chart.setChartData(data,"json");
    fetch_Energy1_data();
    }
    }
```

```
Energy1Request.open("POST","fetchCI1.php",true);
    Energy1Request.send();
    },3000));
    FusionCharts.ready(function(){
    Energy2Chart.render("Energy2ChartContainer");
    });
    (function fetch_Energy2_data(){
    setTimeout(function(){
        var Energy2Request = new XMLHttpRequest();
        Energy2Request.onreadystatechange = function(){
if (Energy2Request.readyState==4 && Energy2Request.status==200){
data2={
"chart": {
    "caption": "Produção de CO2 por kWh de electricidade",
    "subCaption": "0,114 kg/kWh",
    "xAxisName": "Tempo",
    "yAxisName": "CO2 kg",
    "refreshinterval": "5",
    "theme": "fint",
    "animation": "0",
    "showvalues":"0"
```

```

        },
        "data":eval(Energy2Request.responseText)
    }
    Energy2Chart.setChartData(data2,"json");
    fetch_Energy2_data();
    }
    }
    Energy2Request.open("POST","fetchCI2.php",true);
    Energy2Request.send();
},3000))());

    FusionCharts.ready(function(){
    Energy3Chart.render("Energy3ChartContainer");
});

    (function fetch_Energy3_data(){
    setTimeout(function(){
        var Energy3Request = new XMLHttpRequest();
        Energy3Request.onreadystatechange = function(){
if (Energy3Request.readyState==4 && Energy3Request.status==200){
data3={
"chart": {
    "caption": "consumo equivalente petroleo por cada kWh",

```

```

        "subCaption": "0,196 kgep/kWh",
        "xAxisName": "Tempo",
        "yAxisName": "kgep",
        "refreshinterval": "5",
        "theme": "fint",
        "animation": "0",
        "showvalues":"0"
    },
        "data":eval(Energy3Request.responseText)
    }
    Energy3Chart.setChartData(data3,"json");
    fetch_Energy3_data();
    }
    }
    Energy3Request.open("POST","fetchCI3.php",true);
    Energy3Request.send();
},3000))());

    </script>
</head>
<body>
    <?php include('navigation.php') ?>

```

```

        <h1>Resultados do Custo da Energia</h1>
<div id="Energy1ChartContainer">Chart Placeholder</div>
        <h1>Resultados do Consumo de CO2</h1>
<div id="Energy2ChartContainer">Chart Placeholder</div>
        <h1>Resultados do Consumo em Kgep</h1>
<div id="Energy3ChartContainer">Chart Placeholder</div>
        </body>
</html>
    
```

```

        $valor=$obj->Active_Energy_A*0.162*0.001+$obj-
>Active_Energy_B*0.162*0.001+$obj-
>Active_Energy_C*0.162*0.001+39.800075*0.001;
        $data[] = array("label"=>$obj->time,"value"=>$valor);
    }
}
echo json_encode($data);
?>
    
```

fetchCI1.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$data = array();
if ($result = $mysqli->query("SELECT time, Active_Energy_A,
Active_Energy_B, Active_Energy_C FROM soc2 ORDER BY time DESC
LIMIT 0,24")){
    while ($obj = $result->fetch_object()){
    
```

fetchCI2.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
$data = array();
if ($result = $mysqli->query("SELECT time, Active_Energy_A,
Active_Energy_B, Active_Energy_C FROM soc2 ORDER BY time DESC
LIMIT 0,24")){
    while ($obj = $result->fetch_object()){
    
```

```

                $valor=$obj->Active_Energy_A*0.001*0.114+$obj-
>Active_Energy_B*0.001*0.114+$obj->Active_Energy_C*0.001*0.114;
                $data[] = array("label"=>$obj->time,"value"=>$valor);
            }
        }
    }
    echo json_encode($data);
?>

```

```

                $data[] = array("label"=>$obj->time,"value"=>$valor);
            }
        }
    }
    echo json_encode($data);
?>

```

fetchCI3.php

```

<?php
require_once('db_config.php');
$db = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($db->connect_errno) {
    echo "500";
}
$data = array();
if ($result = $db->query("SELECT time, Active_Energy_A,
Active_Energy_B, Active_Energy_C FROM soc2 ORDER BY time DESC
LIMIT 0,24")){
    while ($obj = $result->fetch_object()){
        $valor=$obj->Active_Energy_A*0.001*0.196+$obj-
>Active_Energy_B*0.001*0.196+$obj->Active_Energy_C*0.001*0.196;
    }
}

```

cost.php

```

<html>
    <head>
        <meta http-equiv="Content-type" content="text/html;
charset=UTF-8">
        <link rel="stylesheet" type="text/css"
href="assets/css/app.css" media="screen">
        <script type="text/javascript"
src="js/fusioncharts.js"></script>
        <script type="text/javascript"
src="js/themes/fusioncharts.theme.zune.js"></script>
        <script type="text/javascript">
            var revenueChart1 = new FusionCharts({
                type: "angulargauge",
                renderAt: "chartContainer1",
                animation: "0",

```

```
        width: "400",
        height: "240",
        dataFormat: "json",
    });

    var revenueChart2 = new FusionCharts({
        type: "angulargauge",
        renderAt: "chartContainer2",
        animation: "0",
        width: "400",
        height: "240",
        dataFormat: "json",
    });

    var revenueChart3 = new FusionCharts({
        type: "angulargauge",
        renderAt: "chartContainer3",
        animation: "0",
        width: "400",
        height: "240",
        dataFormat: "json",
    });

    var revenueChart4 = new FusionCharts({
```

```
        type: "angulargauge",
        renderAt: "chartContainer4",
        animation: "0",
        width: "400",
        height: "240",
        dataFormat: "json",
    });

    (function fetch_data1(){
        setTimeout(function(){
            var xmlhttp1;
            xmlhttp1 = new XMLHttpRequest();
            xmlhttp1.onreadystatechange = function(){
                if (xmlhttp1.readyState==4 && xmlhttp1.status==200){
                    data1={
                        "chart": {
                            "caption": "Consumo instantâneo carga A",
                            "subCaption": "[€/dia]",
                            "lowerLimit": "0",
                            "upperLimit": "2.16",
                            "theme": "zune",
                            "defaultAnimation": "0"
```

```
    },  
    "colorRange": {  
      "color": [  
        {  
          "minValue": "0",  
          "maxValue": "0.36",  
          "code": "#006400"  
        },  
        {  
          "minValue": "0.36",  
          "maxValue": "0.54",  
          "code": "#00008b"  
        },  
        {  
          "minValue": "0.54",  
          "maxValue": "0.81",  
          "code": "#ff8c00"  
        },  
        {  
          "minValue": "0.81",  
          "maxValue": "2.16",
```

```
        "code": "#8b0000"  
      }  
    ]  
  },  
  "dials": {  
    "dial": [{  
      "value": eval(xmlhttp1.responseText)  
    }  
  ]  
  }  
  revenueChart1.setChartData(data1,"json");  
  fetch_data1();  
  }  
  xmlhttp1.open("POST","Cost1_fetcher.php",true);  
  xmlhttp1.send();  
  },3000));  
  FusionCharts.ready(function(){  
    revenueChart1.render("chartContainer1");  
  });  
  (function fetch_data2(){
```

```
setTimeout(function(){
    var xmlhttp2;
    xmlhttp2 = new XMLHttpRequest();
    xmlhttp2.onreadystatechange = function(){
        if (xmlhttp2.readyState==4 && xmlhttp2.status==200){
data2={
"chart": {
    "caption": "Consumo instantâneo carga B",
    "subCaption": "[€/dia]",
    "lowerLimit": "0",
    "upperLimit": "2.16",
    "theme": "zune",
    "defaultAnimation": "0"
    },
    "colorRange": {
    "color": [
    {
    "minValue": "0",
    "maxValue": "0.36",
    "code": "#006400"
    },

```

```

    {
    "minValue": "0.36",
    "maxValue": "0.54",
    "code": "#00008b"
    },
    {
    "minValue": "0.54",
    "maxValue": "0.81",
    "code": "#ff8c00"
    },
    {
    "minValue": "0.81",
    "maxValue": "2.16",
    "code": "#8b0000"
    }
    ]
    },
    "dials": {
        "dial": [{
            "value": eval(xmlhttp2.responseText)
        }
    ]
}
```

```
        }
    }
    revenueChart2.setChartData(data2,"json");
        fetch_data2();
    }
}
xmlhttp2.open("POST","Cost2_fetcher.php",true);
    xmlhttp2.send();
},3000))));
    FusionCharts.ready(function(){
        revenueChart2.render("chartContainer2");
    });

    (function fetch_data3(){
        setTimeout(function(){
            var xmlhttp3;
            xmlhttp3 = new XMLHttpRequest();
            xmlhttp3.onreadystatechange = function(){
                if (xmlhttp3.readyState==4 && xmlhttp3.status==200){
                    data3={
                        "chart": {
                            "caption": "Consumo instantâneo carga C",
```

```
                            "subCaption": "[€/dia]",
                            "lowerLimit": "0",
                            "upperLimit": "2.16",
                            "theme": "zune",
                            "defaultAnimation": "0"
                        },
                        "colorRange": {
                            "color": [
                                {
                                    "minValue": "0",
                                    "maxValue": "0.36",
                                    "code": "#006400"
                                },
                                {
                                    "minValue": "0.36",
                                    "maxValue": "0.54",
                                    "code": "#00008b"
                                },
                                {
                                    "minValue": "0.54",
                                    "maxValue": "0.81",
```

```

        "code": "#ff8c00"
    },
    {
        "minValue": "0.81",
        "maxValue": "2.16",
        "code": "#8b0000"
    }
]
},
"dials": {
    "dial": [{
        "value": eval(xmlhttp3.responseText)
    }
    ]
}
revenueChart3.setChartData(data3,"json");
    fetch_data3();
}
xmlhttp3.open("POST","Cost3_fetcher.php",true);
xmlhttp3.send();

```

```

},3000))());
    FusionCharts.ready(function(){
revenueChart3.render("chartContainer3");
});
    (function fetch_data4(){
setTimeout(function(){
    var xmlhttp4;
    xmlhttp4 = new XMLHttpRequest();
    xmlhttp4.onreadystatechange = function(){
        if (xmlhttp4.readyState==4 && xmlhttp4.status==200){
data4={
"chart": {
    "caption": "Consumo instantâneo total",
    "subCaption": "[€/dia]",
    "lowerLimit": "0",
    "upperLimit": "6.48",
    "theme": "zune",
    "defaultAnimation": "0"
    },
    "colorRange": {
    "color": [

```

```
{
  "minValue": "0",
  "maxValue": "1.08",
  "code": "#006400"
},
{
  "minValue": "1.08",
  "maxValue": "1.62",
  "code": "#00008b"
},
{
  "minValue": "1.62",
  "maxValue": "2.43",
  "code": "#ff8c00"
},
{
  "minValue": "2.43",
  "maxValue": "6.48",
  "code": "#8b0000"
}
]
```

```
},
"dials": {
  "dial": [{
    "value": eval(xmlhttp4.responseText)
  ]
}
}
revenueChart4.setChartData(data4,"json");
fetch_data4();
}
}
xmlhttp4.open("POST","Costtotal_fetcher.php",true);
xmlhttp4.send();
},3000)}}());
FusionCharts.ready(function(){
revenueChart4.render("chartContainer4");
});
</script>
</head>
<body>
<?php include('navigation.php') ?>
```

```

        <h1>Resultados dos Gastos Instantâneos de Electricidade, por
fase:</h1>
<div id="chartContainer1" style="float:left">Chart Placeholder</div>
<div id="chartContainer2" style="float:left">Chart Placeholder</div>
<div id="chartContainer3" >Chart Placeholder</div>
        <h1>Resultado do Gasto Instantâneo de Electricidade
Total:</h1>
<div id="chartContainer4" style="float:left">Chart Placeholder</div>
</body>
</html>
    
```

Cost1_fetcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT Active_Power_A FROM soc ORDER
BY time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->Active_Power_A*0.001*24*0.162;
    }
}
    
```

```

    }
}
echo json_encode($data);
?>
    
```

Cost2_fetcher.php

```

<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT Active_Power_B FROM soc ORDER
BY time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->Active_Power_B*0.001*24*0.162;
    }
}
echo json_encode($data);
?>
    
```

Cost3_fetcher.php

```
<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
    echo "500";
}
if ($result = $mysqli->query("SELECT Active_Power_C FROM soc ORDER
BY time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $data = $obj->Active_Power_C*0.001*24*0.162;
    }
}
echo json_encode($data);
?>
```

```
    echo "500";
}
if ($result = $mysqli->query("SELECT Active_Power_A, Active_Power_B,
Active_Power_C FROM soc ORDER BY time DESC LIMIT 0,1")){
    while ($obj = $result->fetch_object()){
        $total=($obj->Active_Power_A*0.001*0.162*24+$obj-
>Active_Power_B*0.001*0.162*24+$obj-
>Active_Power_C*0.001*0.162*24);
        $data = $total;
    }
}
echo json_encode($data);
?>
```

Costtotal_fetcher.php

```
<?php
require_once('db_config.php');
$mysqli = new mysqli($db_server,$db_user,$db_pass,$db_database);
if ($mysqli->connect_errno) {
```