



**Centro de ciências exactas e da Engenharia**

**SmartEnergy – Rede inteligente de gestão de energia  
eléctrica em edifícios**

Luís Miguel Vieira Fernandes de Aguiar

(Licenciado)

Projecto para obtenção do grau de Mestre em  
Engenharia de Telecomunicações e Redes

Orientador: Doutor João Dionísio Simões de Barros

**Júri**

Presidente: Doutor Alberto de Jesus Nascimento

Vogais: Doutor Luís Armando de Aguiar Oliveira Gomes

Doutor João Dionísio Simões de Barros

**Setembro 2011**



## Resumo

Os sistemas de domótica são cada vez mais utilizados para gerir a energia conseguindo-se uma redução do custo. Devido à facilidade de interligação com os sistemas de iluminação e gestão ambiental de uma casa, um sistema de domótica é uma plataforma adequada para a implementação de sistemas de despacho de energia capazes de interagir com a disponibilidade da energia da rede eléctrica.

Um sistema de despacho de energia monitoriza o estado a rede eléctrica e efectua o controlo das cargas ligadas à rede. A proliferação de sistemas de despacho domésticos proporciona uma gestão mais eficiente da rede eléctrica, trazendo benefícios económicos. A taxa de aumento significativo destes sistemas de despacho domésticos só será mais atractiva se o custo for reduzido para uma maior quantidade de consumidores.

Neste projecto de mestrado foi implementado um protótipo de um sistema de automação capaz de interligar diferentes protocolos de automação, de forma a aproveitar as características de cada protocolo para se obter um custo mais reduzido e simplificar a instalação. Foi desenvolvida uma *gateway* que utiliza um protocolo interno que faz a abstracção entre os protocolos. Esta camada de abstracção protocolar permite a normalização do desenvolvimento de aplicações, da gestão da rede e da comunicação entre equipamentos.

Para implementar o sistema de despacho foi desenvolvida uma aplicação que corre sobre o sistema de automação, permitindo gerir o estado da rede eléctrica através da activação e desactivação de cargas.

O sistema de automação efectua a troca de mensagens entre os diferentes protocolos de uma forma genérica. O custo do sistema foi reduzido através da escolha de actuadores X10, reduzindo o custo dos actuadores em aproximadamente 45% em comparação com um equipamento KNX semelhante.

O sistema de gestão de energia demonstrou melhorar a eficiência e fiabilidade da rede, respondendo às variações do consumo e produção.

**Palavras-chave:** Domótica, *gateway* de múltiplos protocolos, despacho de energia, XBee, X10, KNX.

## Abstract

Home automation systems are being increasingly used to manage energy consumption, reducing energy cost. Because of the easy interconnection with the lighting and environmental systems of a home, the home management system is a suitable platform for the implementation of demand response systems, capable of interacting with the availability of power from the grid.

The demand response system monitors the power grid and controls the loads connected to the grid. The proliferation of domestic demand response systems enables a more efficient management of the power grid, bringing with it economic benefits. The increase rate of such demand response systems will only be more attractive if the cost is reduced for a greater number of consumers.

In this Master's project a prototype of an automation system capable of interconnecting different automation protocols was implemented, in order to take advantage of the characteristics of each protocol and to provide a lower system cost and easier installation. A gateway was developed, that uses an internal protocol and makes an abstraction of the protocols. This protocol abstraction layer allows the standardization of application development, network management and communication between devices.

To implement the demand response system, an application that runs on the automation system was developed, allowing it to manage the power grid state through the activation and deactivation of loads.

The automation system performs the exchange of messages between different protocols in a generic way. The system cost was reduced by the selection of X10 actuators, reducing in approximately 45% the cost of the actuators in comparison to a similar KNX device.

The management system demonstrated an improvement of the energy efficiency and network reliability, by responding to the variations of the production and consumption of energy.

**Keywords:** Domotics, multi-protocol gateway, automatic demand response, XBee, X10, KNX

## Agradecimentos

Aos meus pais e família que permitiram e incentivaram-me a chegar aqui.

À minha namorada que me apoiou, incentivou e apresentou grande paciência ao longo deste tempo todo.

Ao meu orientador, professor Dionísio Barros pela paciência, orientação e apoio disponibilizados ao longo da elaboração desta tese.

Aos Engenheiros Filipe Santos e Maurício Rodrigues, pelas sugestões e apoio no desenvolvimento deste projecto.

Ao Engenheiro Célio Silva da empresa RC automação, pelo apoio prestado ao longo desta tese.

A todos os colegas pelo apoio nas horas de estudo e desenvolvimento do projecto.

# Índice

Resumo .....	i
Abstract.....	ii
Agradecimentos .....	iii
Lista de acrónimos .....	vi
Índice de figuras.....	viii
Índice de tabelas.....	x
1 Introdução .....	1
1.1 Motivação.....	1
1.2 Enquadramento.....	2
1.3 Objectivos.....	5
1.4 Organização.....	5
1.5 Contribuições originais .....	6
2 Estado da arte .....	7
2.1 Domótica (Automação doméstica).....	7
2.2 Conversores de protocolos .....	21
2.3 <i>Smart grids</i> (Redes eléctricas inteligentes) .....	23
3 Sistema de automação.....	29
3.1 Protocolos escolhidos para o protótipo .....	30
3.2 Protótipo da <i>gateway</i> .....	32
3.3 <i>Firmware</i> de uniformização de protocolos .....	38
4 Sistema inteligente de gestão de energia .....	67
4.1 Principais elementos.....	68
4.2 Gestor da <i>Gateway</i> .....	73
4.3 Web GUI .....	77

4.4	Despacho de energia automático.....	80
5	Testes do sistema de despacho de energia .....	85
5.1	Cenário 1 – Consumo e produção de referência de energia em regime estacionário.....	86
5.2	Cenário 2 – Produção de referência normalmente superior ao consumo de energia .....	89
5.3	Cenário 3 – Produção de referência normalmente inferior ao consumo de energia .....	90
5.4	Cenário 4 – Aumento e diminuição rápida do consumo .....	91
5.5	Cenário 5 – Redução rápida da produção de energia .....	92
5.6	Simulação do cenário 1 com variação do peso da energia.....	93
5.7	Simulação do cenário 1 com 6 equipamentos .....	97
6	Conclusões e trabalhos futuros .....	99
6.1	Conclusões .....	99
6.2	Trabalhos futuros.....	101
	Referências.....	102
	Anexo A: Tabela de códigos X10.....	105
	Anexo B: Esquema de ligações da <i>gateway</i> .....	106
	Anexo C: Desenho PCB da <i>gateway</i> .....	107
	Anexo D: Lista de APIs XBee .....	108
	Anexo E: Fotografia do protótipo implementado .....	109
	Anexo F: Código C do <i>firmware</i> da <i>gateway</i> .....	110
	Anexo G: Tabela dos atributos e comandos da camada de aplicação PIntern.....	110
	Anexo H: Código JAVA da aplicação .....	111
	Anexo I: Tabelas dos valores dos cenários de teste do sistema de despacho .....	111
	Anexo J: Registos da execução do sistema de despacho de energia.....	116

## Lista de acrónimos

**AES** – *Advanced Encryption Standard* (Padrão de encriptação avançado)

**API** – *Application Programming Interface* (Interface de programação da Aplicação)

**AT** – *ATtention* (Atenção)

**AVR** – *Advanced Virtual RISC*

**BAS** – *Building Automation System* (Sistema de automação de edifício)

**CNP** – *Control Network Protocol* (protocolo de rede de controlo)

**CPU** – *Central Processing Unit* (Unidade central de processamento)

**CSMA / CA** – *Carrier Sensing Multiple Access / Collision Avoidance*

**CSMA / CD** – *Carrier Sensing Multiple Access / Collision Detection*

**DES** – *Data Encryption Standard* (Padrão de encriptação de dados)

**DEST** – *Destination* (Destino)

**DMA** – *Direct Memory Access* (Acesso directo à memória)

**DRE** – *Data Registry Empty* (Registo de dados Vazio)

**DSM** – *Demand Side Management* (Gestão no lado da demanda)

**DSSS** – *Direct-Sequence Spread Spectrum* (Espalhamento espectral por sequência directa)

**EHB** – *European Home Systems*

**EIB** – *European Installation Bus*

**ETS** – *Engineering Tool Software*

**FIFO** – *First In First Out* (Primeiro a entrar primeiro a sair)

**GSM** – *Global System for Mobile Communications* (sistema global para comunicações móveis)

**GUI** – *Graphical User Interface* (interface gráfico com o utilizador)

**HAN** – *Home Automation Network* (Rede de Automação doméstica)

**HVAC** – *Heating, Ventilation and Air Conditioning* (Aquecimento, Ventilação e Ar condicionado)

**I<sup>2</sup>C** – *Inter-Integrated Circuit*

**IP** – *Internet Protocol*

**IrDA** – *Infrared Data Association*

**ISM** – *Industrial Scientific and Medical*

**MAC** – *Medium Access Control*

**MCU** – *MicroController Unit* (microcontrolador)

**OSI** – *Open Systems Interconnection*

**PCB** – *Printed Circuit Board* (Placa de circuito impresso)

**PIN** – *Personal Identification Number* (Numero de identificação pessoal)

**PL** – *Power Line* (Rede eléctrica)

**PLC** – *PowerLine Communication* (Comunicação pela rede eléctrica)

**RF** – **Rádio Frequência**

**RISC** – *Reduced Instruction Set Computer* (computador com conjunto de instruções reduzido)

**RMI** – *Remote Method Invocation* (Invocação de métodos remota)

**RTP** – *Real Time Pricing* (Preço em tempo real)

**RXC** – *Reception Completed* (Recepção completa)

**SIM** – *Subscriber Identify Module* (Modulo de identificação do subscritor)

**SMS** – *Short Message Service* (Serviço de mensagens curtas)

**SPI** – *Serial Peripheral Interface*

**SRAM** – *Static Random Access Memory*

**SRC** – *Source* (Fonte, origem)

**SVG** – *Scalable Vector Graphics*

**TOU** – *Time Of Use* (Tempo de utilização)

**TP / FT** – *Twister Pair / Free Topology* (Par entrançado / topologia livre)

**UART** – *Universal Asynchronous Receiver/Transmitter*

**USART** – *Universal Synchronous Asynchronous Receive Transmit*

**USB** – *Universal Serial Bus* (Barramento série universal)

**ZDO** – *ZigBee Device Object*

**ZIF** – *Zero Insertion Force*

## Índice de figuras

Figura 1: Contribuição por fonte de energia primária da Empresa de Electricidade da Madeira [9]	3
Figura 2: Integração de Sistemas [14]	8
Figura 3: Elementos básicos de um sistema de automação	9
Figura 4: Forma de onda da transmissão X10 [17]	12
Figura 5: Formato dos códigos X10 transmitidos[17]	12
Figura 6: Topologias físicas de um sistema KNX [16]	14
Figura 7: Pilha protocolar <i>ZigBee</i> [19]	16
Figura 8: Topologias suportadas no meio TP/FP-10 [4]	19
Figura 9: Arquitectura protocolar de uma <i>gateway</i> [24]	22
Figura 10: <i>Gateway</i> Multiprotocolo Harmony [26]	22
Figura 11: Comparação do diagrama de carga sem sistema de despacho, com sistema RTP e com sistema TOU [31]	26
Figura 12: Interface remoto para gestão do edifício [32]	27
Figura 13: Interface de opções de configuração do sistema de despacho [32]	28
Figura 14: Arquitectura do sistema proposto	29
Figura 15: Arquitectura do Hardware da Gateway	32
Figura 16: Placa PCB do microcontrolador XMEGA128A1	34
Figura 17: Dimensões e Numeração dos pinos do módulo XBee [36]	36
Figura 18: Esquema de pinos do adaptador UART-USB [37]	36
Figura 19: <i>Pinout</i> do módulo XM10E	37
Figura 20: Placa PCB do módulo GSM	37
Figura 21: Diagrama do processo de conversão proposto	39
Figura 22: Comparação do número de algoritmos para conversão da forma directa e indirecta calculados com as equações (3.1) e (3.2)	40
Figura 23: Formato da trama de ligação de dados do protocolo PIntern	41
Figura 24: Formato da trama da camada de rede do protocolo PIntern	41
Figura 25: Formato da trama da camada de aplicação do protocolo PIntern	43
Figura 26: Representação de objectos no protocolo PIntern	44

Figura 27: Diagrama de interligação dos <i>buffers</i> .....	45
Figura 28: Representação do buffer FIFO circular .....	46
Figura 29: Segmento de código que efectua incremento de um índice do <i>buffer</i> .....	47
Figura 30: segmento de código da função de configuração de uma porta UART do microcontrolador.....	48
Figura 31: segmento de código para obtenção de um ponteiro para um pacote num buffer .....	49
Figura 33: segmento de código que determina camada de aplicação e chama a função apropriada .....	50
Figura 32: segmento de código que obtém o endereço de 64 <i>bits</i> do pacote XBee.....	50
Figura 34: Formato da trama da camada de aplicação florsens .....	51
Figura 35: segmento de código para preencher o pacote XBee com os dados do pacote PIntern	52
Figura 36: Segmento de código que configura os pinos de entrada e saída e temporizador .....	54
Figura 37: Segmento de código que chama as funções para complementar o códigos X10 .....	55
Figura 38: Segmento de código que coloca no pino de saída a sequência de <i>bits</i> do pacote X10	56
Figura 39: Formato das mensagens SMS para interagir com a rede.....	61
Figura 40: Diagrama do despacho dos pacotes na <i>gateway</i> .....	63
Figura 41: Diagrama do escalonamento de despacho de pacotes da <i>gateway</i> .....	64
Figura 42: Diagrama Entidade-Associação da base de dados segundo representação Crow's Foot .....	71
Figura 43: Página Web de interface gráfico .....	78
Figura 44: Gráficos do factor de incentivo de energia.....	82
Figura 45: Gráficos do factor de incentivo de conforto .....	83
Figura 46: Diagramas de consumo e produção de referência com e sem sistema de despacho, cenário 1 .....	87
Figura 47: Diagrama de consumo dos equipamentos 0 e 1, cenário 1 .....	87
Figura 48: Mensagem enviada pelo sistema de despacho a indicar acção realizada .....	88
Figura 49: Diagramas de consumo e produção de referência com e sem sistema de despacho, cenário 2.....	89
Figura 50: Diagrama de consumo dos equipamentos 0 e 1, cenário 2.....	90
Figura 51: Diagramas de consumo e produção com e sem sistema de despacho, cenário 3 .....	90
Figura 52: Diagrama de consumo dos equipamentos 0 e 1, cenário 3.....	91

Figura 53: Diagramas de consumo e produção com e sem sistema de despacho, cenário 4 .....	91
Figura 54: Diagrama de consumo dos equipamentos 0 e 1, cenário 4.....	92
Figura 55: Diagramas de consumo e produção com e sem sistema de despacho, cenário 5 .....	92
Figura 56: Diagrama de consumo dos equipamentos 0 e 1, cenário 5.....	93
Figura 57: Gráfico da variação do custo de energia para diferentes valores do peso da energia .	93
Figura 58: Gráfico da variação da temperatura ao longo do dia utilizado pelo sistema.....	94
Figura 59: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com peso de energia de 60%.....	95
Figura 60: Diagrama de consumo dos equipamentos 0 e 1, cenário 1 com peso de energia de 60% .....	95
Figura 61: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com peso de energia de 60% analisando somente o período actual .....	96
Figura 62: Diagrama de consumo do equipamento 0, cenário 1 com peso de energia de 75% ....	96
Figura 63: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com 6 equipamentos com margem de 20 MW .....	97
Figura 64: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com 6 equipamentos com margem de 5 MW .....	98

## Índice de tabelas

Tabela 1: Tabela dos endereços do protocolo PIntern .....	42
Tabela 2: Equipamentos do sistema de despacho .....	85

# 1 Introdução

Neste projecto de mestrado pretendeu-se desenvolver um sistema de gestão distribuída de energia capaz de gerir o equilíbrio energético (entre a produção e consumo de electricidade). O sistema dinâmico de gestão de energia permitiu otimizar a geração distribuída baseada em fontes de energia renováveis com níveis de produção variáveis.

Pretendeu-se implementar um protótipo de um sistema de automação que fosse caracterizável pelo baixo custo e fácil instalação. O sistema deveria realizar um serviço de despacho de energia que respondesse de forma a equilibrar os níveis de produção e consumo.

O sistema protótipo foi realizado segundo uma arquitectura agregadora de protocolos de automação, onde o processamento de dados e gestão das redes era efectuado num só ponto (computador). O sistema foi desenvolvido de forma genérica e pode ser modificado para uma arquitectura distribuída, para instalação em prédios industriais de médias ou grandes dimensões.

A redução de custo foi efectuada pela integração de protocolos comerciais existentes, através de uma *gateway*, sendo possível escolher os actuadores e sensores mais adequados, definindo assim as características do sistema com as características dos protocolos utilizados.

## 1.1 Motivação

Tradicionalmente, a gestão de energia eléctrica é feita de forma centralizada e actuando somente na regulação da produção. A grande incerteza ao nível do consumo obriga a haver uma reserva de produção de energia, para responder às grandes variações dinâmicas do consumo. Essa reserva, sendo necessária para garantir a estabilidade na rede de energia eléctrica, contribui para a redução da eficiência de gestão da energia eléctrica.

A necessidade actual de utilizar fontes de energia renováveis, cuja produção não é totalmente controlável, tem exigido o controlo da energia na parte do consumo.

Para melhor gerir o balanço energético, parte da regulação de energia eléctrica pode ser efectuada do lado do consumidor. Este conceito, que é normalmente designado por DSM (*Demand Side Management*), permite aproximar a produção do consumo, sendo necessária a automatização dos edifícios para melhorar a gestão da rede de energia eléctrica e melhorar a eficiência energética [1].

Neste projecto foi desenvolvido um protótipo idealizado para os consumidores domésticos. Justifica-se esta escolha pelo grande número de potenciais utilizadores de sistemas de automação, que exigem sistemas de baixo custo e baixa complexidade.

As soluções existentes adoptam normalmente um único protocolo de automação. Essa limitação restringe a flexibilidade na escolha dos equipamentos, podendo contribuir para o aumento do custo. Algumas destas soluções individuais não possuem as funcionalidades e equipamentos necessários para sistemas de automação capazes de responder às necessidades da rede eléctrica [2]-[4].

## **1.2 Enquadramento**

A utilização eficiente da energia é uma preocupação cada vez maior na sociedade, devido ao receio de esgotamento de alguns recursos energéticos, nomeadamente dos combustíveis fósseis, e do conseqüente aumento do custo da energia [5].

O equilíbrio energético é essencial hoje em dia para garantir a fiabilidade da rede eléctrica. O equilíbrio energético é, idealmente, obtido quando o nível de energia produzida é igual à energia consumida. O equilíbrio da produção/consumo pode ser abordado de ambos os lados da “balança”: lado da produção e lado do consumo [6].

Este projecto focaliza o estudo na contribuição do lado do consumo, em função da disponibilidade de energia e das necessidades do consumo. Segundo a filosofia da indústria de produção eléctrica, o sistema apresenta melhor eficiência energética quando a diferença entre os níveis de máximo e mínimo do consumo for a menor possível [6].

### **1.2.1 Produção de energia eléctrica**

Normalmente, o equilíbrio energético é gerido pela intervenção no lado da produção, regulando a geração de energia eléctrica para responder à solicitação dinâmica do consumo [6].

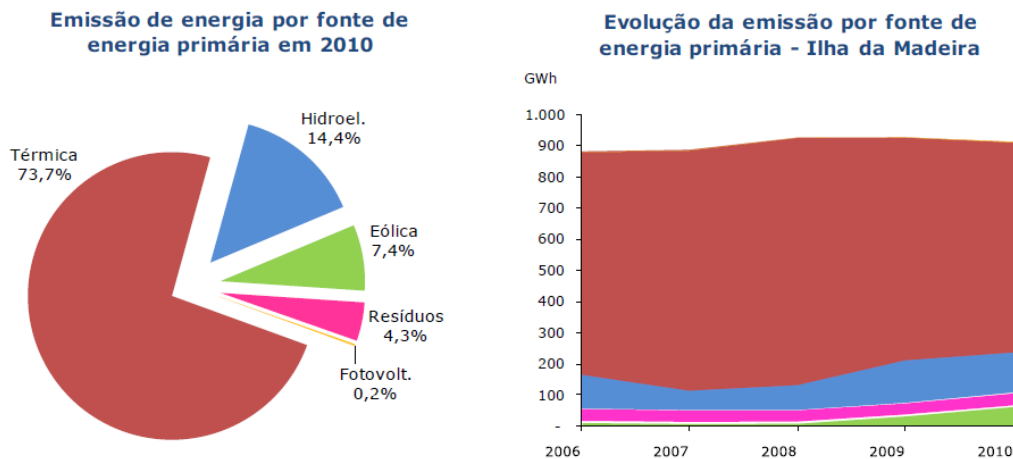
A principal fonte de energia mundial é a combustão de combustíveis fósseis, correspondendo aproximadamente a 80% do consumo mundial. Esta fonte de energia é muito atractiva pela sua capacidade de armazenar energia num pequeno volume, e facilidade de distribuição [7], [8].

Hoje em dia, devido ao grande uso de combustíveis fósseis, a população mundial depara-se com os problemas das alterações climáticas (provavelmente devido à grande emissão de CO<sub>2</sub>), poluição do ar e água, conflitos sobre o acesso aos recursos, etc [7], [8].

Como as principais fontes de energia eléctrica advêm da queima de combustíveis, produtoras de CO<sub>2</sub> de gases que produzem efeito de estufa, serão eventualmente necessárias políticas de redução da dependência deste tipo de fontes de energia [7], [8].

No entanto, o estilo de vida moderno exige cada vez mais consumo de energia eléctrica por pessoa. O crescimento da população e da qualidade de vida tem exigido a construção de centrais eléctricas para satisfazer as necessidades de consumo de energética [5].

A título de exemplo local, a Empresa de Electricidade da Madeira, cuja contribuição das fontes de energia primárias estão indicadas na Figura 1, apresenta uma percentagem da produção de energia de 73,7% efectuada através da queima de combustível.



**Figura 1: Contribuição por fonte de energia primária da Empresa de Electricidade da Madeira [9]**

Actualmente, há uma tendência na Madeira de aumentar a produção através da utilização de fontes de energia renováveis, como a fotovoltaica, eólica e hídrica [9], [10]. Como pode ser observado na Figura 1, gráfico da direita, verifica-se que desde o ano de 2008 a produção através de energia eólica (7,4% em 2010), hídrica (14,4% em 2010) e solar (0,03% em 2009 e 0,2% em 2010) estão a aumentar, e a térmica a reduzir (76,95% em 2009 e 73,7% em 2010) [9], [10].

Em todo o mundo tem igualmente ocorrido uma crescente adopção da geração de energia através de fontes renováveis. Estas fontes renováveis, apesar de benignas na maior parte dos casos, em alguns casos implicam algum impacto social e ambiental [8].

A variação dos níveis de potência produzidos através destas fontes de energia origina dificuldades na gestão da rede eléctrica, como [11]:

- Dificuldades de agendamento de produção, devido às incertezas introduzidas nas previsões de disponibilidade de energia;
- Cuidados adicionais para maiores amplitudes de flutuação na produção de energia;
- Dificuldade de previsão de rápidas flutuações.

Para suportar a geração distribuída a partir de fontes renováveis (com níveis de produção variáveis) é necessária a utilização de uma rede eléctrica capaz de prever o comportamento e responder de forma inteligente, para prover um serviço fiável, eficiente e económico. Uma rede deste género entra dentro do conceito de uma *SmartGrid* (Rede eléctrica inteligente) [12].

### 1.2.2 Consumo de energia eléctrica

Outra forma de influenciar o equilíbrio energético é agir no lado do consumo de forma a reduzi-lo permitindo atingir o equilíbrio entre consumo e produção de energia.

Para a redução do consumo existem muitas estratégias, sendo algumas das principais apresentadas em seguida [7], [8]:

- Melhoria dos isolamentos das paredes, tectos e chão;
- Utilização de lâmpadas com maior eficiência;
- Utilização de electrodomésticos mais eficientes, como frigoríficos, máquinas de lavar, computadores, etc;
- Introdução ou melhoria de sistemas de controlo que gerem o consumo de energia.

A mudança de hábitos pode ser uma forma de melhorar o consumo, através de acções como desligar o transformador de um equipamento que não está a ser utilizado, até ao agendamento da utilização de um equipamento para alturas mais propícias. Estas mudanças de hábitos podem ser efectuadas através de um sistema de gestão de energia.

A área de automação doméstica (Domótica) tem ao longo dos anos avançado consideravelmente em termos tecnológicos, até ao ponto de realizar acções e prestar serviços que previamente eram considerados um objectivo inatingível. Um sistema de domótica integra sistemas de controlo de iluminação, ambiente e acesso. Sistemas de segurança e gestão de energia também podem ser integrados [13].

Um sistema de gestão de energia permite a redução do consumo de energia e um uso mais eficiente da mesma. Num destes sistemas a gestão é efectuada de forma automática, isto é, além da configuração do sistema não é necessária a intervenção humana.

A redução do consumo pode advir de acções relativamente simples, como desligar a luz automaticamente quando não está ninguém num quarto, até acções mais complexas como detectar e desligar um equipamento que não está em utilização como uma TV, carregadores, etc. A utilização eficiente de energia é efectuada através da análise e agendamento de períodos para activar e desactivar equipamentos, dependendo de factores como o custo da energia, luminosidade num quarto, temperatura interior e exterior, etc.

### 1.3 Objectivos

Os objectivos a atingir neste projecto de mestrado eram os seguintes:

1. Revisão do estado de arte dos protocolos industriais utilizados em casas inteligentes para a gestão da energia eléctrica em edifícios;
2. Desenvolver um sistema baseado em *hardware* para interligar redes industriais, de forma a compatibilizar e uniformizar a comunicação entre sensores, actuadores e *software* de gestão;
3. Desenvolvimento de uma aplicação informática para comunicar com o sistema de *hardware* de interligação de redes industriais para gerir a rede inteligente de energia eléctrica em edifícios;
4. Construção de um protótipo para testar experimentalmente o desempenho do sistema desenvolvido.

### 1.4 Organização

Este trabalho encontra-se dividido em 6 capítulos.

No capítulo 1, Introdução, é efectuada uma breve introdução ao trabalho com indicação dos objectivos, motivação e enquadramento do trabalho do projecto. Também são apresentadas as contribuições originais introduzidas pelo projecto.

No capítulo 2, Estado da arte, é apresentado o estado da arte, onde é feita uma revisão literária das soluções existentes na área dos sistemas e protocolos de automação, conversores de protocolos e sistemas de despacho de energia.

No capítulo 3, Sistema de automação, é apresentado um sistema de automação, indicando a arquitectura e elementos básicos a utilizar. Também são indicados os protocolos escolhidos para integração, apresentando as contribuições dos mesmos para o sistema. O *hardware* utilizado para implementação do sistema é também descrito. O *firmware* que permite a integração dos protocolos é especificado.

No capítulo 4, Sistema inteligente de gestão de energia, é descrito o sistema de gestão de energia implementado, através da descrição dos seus elementos e funcionalidades.

No capítulo 5, Testes do sistema de despacho de energia, são apresentados os resultados dos testes efectuados, em vários cenários do sistema de despacho

No capítulo 6, Conclusões e trabalhos futuros, são apresentadas as conclusões obtidas ao longo do trabalho. Também são apresentados futuros trabalhos a desenvolver.

## 1.5 Contribuições originais

Este trabalho apresenta contribuições originais nas áreas de conversão de protocolos, sistemas de automação e gestão do consumo de energia doméstico.

Implementação de um sistema de automação que realiza a integração de protocolos através de uma *gateway*, permitindo maior flexibilidade na escolha dos equipamento de forma a reduzir o custo do sistema e aumentar a facilidade de instalação.

Desenvolvimento de um processo de conversão de forma indirecta para facilitar a expansão do número de protocolos suportados. A conversão de forma indirecta implica a utilização de um protocolo comum, sendo neste projecto implementado um protocolo simples que serve de adaptação aos diferentes protocolos.

Implementação de um centro de processamento da *gateway*, utilizando um microcontrolador com programa (*Firmware*) desenvolvido em linguagem C, para realizar a conversão e encaminhamento de pacotes.

Criação de um algoritmo de despacho de energia que analisa os níveis de consumo e produção, para uma gestão eficiente do consumo através da activação e desactivação de cargas. Estes algoritmos têm em consideração uma componente de conforto.

## 2 Estado da arte

Neste capítulo é feita uma revisão bibliográfica para resumir o estado da arte das tecnologias e soluções existentes.

Como se pretendia desenvolver um sistema de automação residencial começa-se por fazer um estudo do estado da arte de sistemas de domótica. Os sistemas de domótica seguem protocolos para a transferência da informação entre os diversos equipamentos, por isso, apresentam-se os diversos protocolos e analisam-se as suas capacidades.

Em seguida, é feita a revisão do estado da arte dos conversores de protocolo existentes, que permitem a integração de diversos sistemas e protocolos.

Como este projecto integra-se dentro do conceito da *SmartEnergy* é essencial também realizar-se uma revisão sobre as redes eléctricas inteligentes (*Smart Grids*) e sobre os sistema de despacho de energia automáticos (*Automatic Demand Response*).

### 2.1 Domótica (Automação doméstica)

A palavra “Domótica” tem origem na palavra *Domus* (palavra latina que significa casa) e na combinação das palavras automática, informática e robótica [13].

“Edifícios inteligentes” já são construídos há várias décadas, mas só nos últimos anos deixaram de se limitar a casas de luxo onde um sistema centralizado era, normalmente, utilizado para gerir as funções da casa. Nos últimos 10 anos, com o aparecimento de soluções para todo o tipo de casas com baixos custos de instalação e utilização, a domótica tem-se tornado cada vez mais popular [13]. Segundo alguns autores, pode-se dizer que: “Hoje, a domótica é, mais do que um luxo ou extra, uma necessidade.” [13].

A domótica surgiu em Portugal à 20-25 anos. Esta actividade apresenta como principal objectivo o comando e controlo de casas/edifícios ditas inteligentes.

Nos primeiros sistemas as funções da casa eram controladas por uma estrutura central única, pensada e projectada de raiz.

Hoje em dias os sistemas de domótica podem realizar de forma integrada o controlo e gestão dos diversos recursos existentes numa residência (Figura 2).

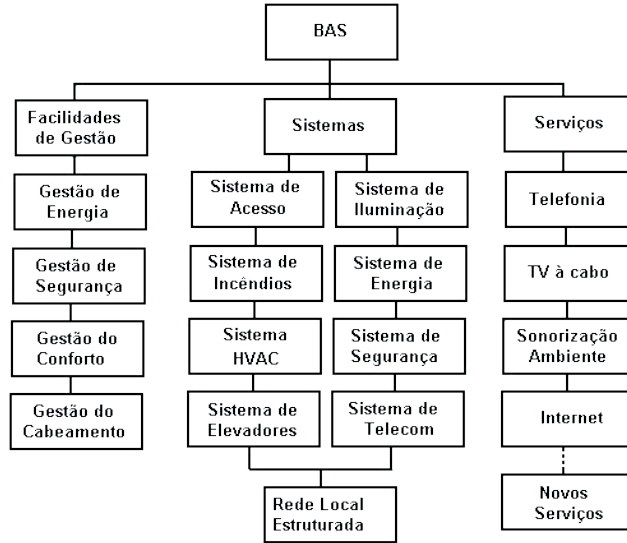


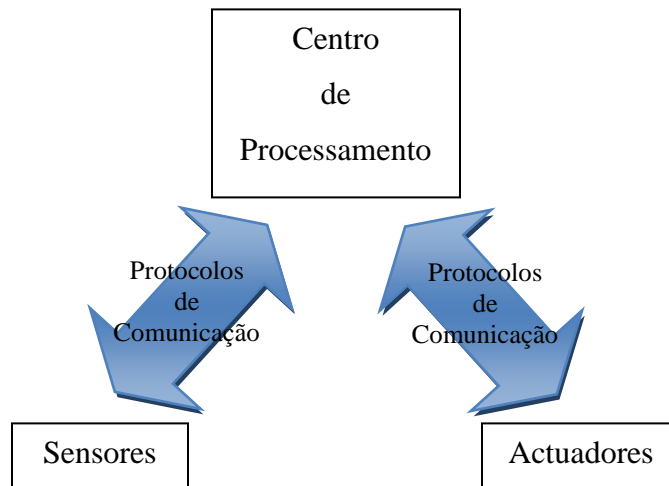
Figura 2: Integração de Sistemas num BAS (*Building Automation System*) [14]

Geralmente um sistema de domótica é constituído por diversos sistemas (ou subsistemas) que permitem o acesso e monitorização dos recursos domésticos, de forma a proporcionar ao utilizador maior conforto e segurança. Os sistemas (ou subsistemas) também abrangem áreas de entretenimento, poupança energética, apoio a idosos ou deficientes [13].

Notou-se, no mercado, o aparecimento de soluções de domótica especializadas para cada área de aplicação, que originou uma grande diversidade de sistemas e protocolos de comunicação. Esta diversidade pode causar a incompatibilidade de comunicação entre sistemas que utilizam protocolos distintos, dificultando a integração dos subsistemas de automação doméstica.

### 2.1.1 Elementos de um sistema de domótica

A nível comercial existem sistemas de automação com diferentes características [2]. De forma genérica, os elementos básicos essenciais dos sistemas de automação estão representados na Figura 3.



**Figura 3: Elementos básicos de um sistema de automação**

Os elementos básicos de um sistema de automação e suas características principais são:

- Os **sensores** são responsáveis pela aquisição dos dados e permitem enviar, para o sistema de automação inteligente, informação sobre os parâmetros que se pretende gerir;
- Os **Actuadores** são os responsáveis por actuar nos elementos da casa que se pretendem controlar;
- O **Centro de Processamento** é onde está localizado o sistema de decisão. Este recebe todos os dados dos sensores, interpretando e processando-os, para determinar a acção a exercer. Num sistema centralizado existe um único centro de processamento que se encontra interligado com todos os equipamentos da rede de automação. Nos sistemas distribuídos o processamento pode ficar distribuído quer nos sensores e quer nos actuadores da rede, libertando recursos do sistema central de decisão para poder decidir as acções mais importantes;
- Os **Protocolos de Comunicação** regem como é efectuada a troca de informação e o meio de transmissão, utilizando protocolos e um meio físico (fibra óptica, ar (RF - Rádio frequência), rede eléctrica, infravermelhos, etc.).

Os protocolos de automação existentes não se limitam a definir as camadas mais baixas do modelo OSI (*Open Systems Interconnection*), nomeadamente, camada física, camada de enlace e camada de rede (Camadas de comunicação) [2]-[4], [15], [16].

A definição das várias camadas do modelo OSI pelos protocolos permite garantir não só a comunicação mas também a interoperabilidade entre os equipamentos.

Os equipamentos seguem, normalmente, um único protocolo restringindo a selecção de equipamentos que podem ser integrados num sistema de automação com garantia de interoperabilidade [2].

Este impedimento não permite grande flexibilidade no projecto e instalação de sistemas de automação doméstica. Um sistema com alguma flexibilidade permite a escolha das funcionalidades, controlo de custo e dimensionamento mais adequados para cada tipo de aplicação [2], [4], [16].

### **2.1.2 Protocolos de automação**

Os equipamentos de domóticas existentes no mercado seguem, normalmente, protocolos direccionados para a automação de edifícios (protocolos de controlo). Existe no entanto uma grande variedade de protocolos, com características e capacidades distintas [2]-[4], [15], [16]. Neste capítulo enumeram-se e resumem-se os principais protocolos comerciais existentes e apresentam-se as suas principais características

#### **2.1.2.1 Protocolo X10**

O protocolo X10 foi um dos primeiros protocolos de automação residencial introduzidos no mercado. Tendo sido desenvolvido pela *Pico Electronics* em 1974 e introduzido no mercado em 1978 pela *RadioShack*, é considerado o pai dos protocolos de domótica. Este título é lhe atribuído devido ao seu amplo uso e idade [2].

Apesar da sua existência no mercado já ultrapassar os 30 anos, e de terem surgido inúmeros protocolos, este continua em grande uso, especialmente para projectos amadores de domótica [2].

O que torna este protocolo tão atractivo é o seu baixo custo, simplicidade de utilização e instalação, apresentando uma característica *Plug & Play* [2].

Como já referido anteriormente, o protocolo X10 apresenta facilidade de instalação, não sendo necessária a passagem de cablagem adicional, pois este utiliza a rede eléctrica doméstica para a transmissão dos sinais entre os diversos equipamentos [2].

Um sistema baseado no protocolo X10 é composto por dois componentes básicos:

- **Controladores:** responsáveis por enviar os sinais de comando. Os controladores podem ser comandos remotos, temporizadores, acesso telefónico ou sensores. Um controlador pode ser utilizado para enviar comandos para qualquer receptor da rede X10;
- **Receptores:** recebem os sinais de comando e são responsáveis pela realização da acção indicada. Dividem-se em duas classes: iluminação, onde são efectuadas acções de ligar/desligar, aumento ou diminuição da intensidade, e electrodomésticos onde são somente realizadas acções de ligar/desligar.

De uma forma resumida as vantagens do protocolo X10 são:

- Baixo custo;
- Facilidade de instalação;
- Baixa complexidade.

Apesar das suas vantagens o protocolo X10 apresenta algumas desvantagens que o torna indesejável para muitos projectos de domótica. As desvantagens do protocolo X10 são:

- Baixa taxa de transmissão;
- Baixa fiabilidade.

Uma das desvantagens, a baixa taxa de transmissão, deve-se ao método de transmissão. Cada *bit* é transmitido a cada transição pelo nível zero do sinal da rede o que implica taxas de transmissão de dados extremamente baixas (100 *bits/s*).

A fiabilidade, normalmente é inferior a 70%. Essa baixa fiabilidade deve-se à sua grande vulnerabilidade aos ruídos introduzidos na rede eléctrica por alguns equipamentos electrónicos e à atenuação do sinal provocada por equipamentos com baixa impedância de entrada [2].

O protocolo X10 define as camadas físicas, ligação de dados, rede e aplicação do modelo OSI. A camada física do X10 define como meio de propagação dos sinais, a cablagem da rede eléctrica. A transmissão dos *bits* é sincronizada com a travessia pelo zero da rede, sendo enviado a cada travessia um *bit* na forma de impulsos de 120 kHz. O código de linha utilizado representa o “1” lógico com o envio de impulsos com uma duração de 1 ms e o “0” pela ausência dos impulsos. Para redes eléctricas trifásicas os impulsos deverão ser transmitidos de forma a coincidirem com as travessias pelos zeros das três fases (Figura 4).

A amplitude dos impulsos não é definida pelo protocolo, podendo variar entre os 4 e 20 volts, o que não permite um alcance de transmissão superior a 30 metros. O alcance está também dependente do estado da rede eléctrica e dos equipamentos electrónicos ligado à rede. Melhorias em termos de alcance podem ser obtidas com o uso de filtros [2].

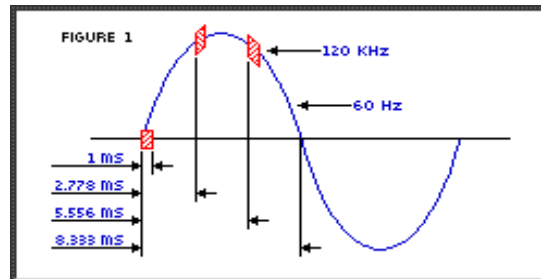


Figura 4: Forma de onda da transmissão X10 [17]

Para a transmissão completa de uma trama são necessários 11 ciclos da rede (22 travessias pelo zero implicando 22 *bits* por trama) como pode ser visto na Figura 5. Para a transmissão de um comando é necessário o envio de duas tramas (trama de endereço e trama de função), uma para indicar o endereço do equipamento que irá executar o comando e o outro código para indicar qual o comando a executar. Ao receber a trama de endereço, o dispositivo correspondente, entra em modo activo escutando os comandos que são enviados, podendo-se enviar múltiplos comandos enquanto este se mantiver activo, isto é, enquanto não for endereçado outro dispositivo.

Para evitar falsas interpretações de tramas estas devem ser transmitidas em grupos de duas tramas (mesma trama repetida duas vezes) separadas por 3 ciclos da rede. A repetição das tramas melhora ligeiramente a fiabilidade do protocolo, mas faz com que o tempo de envio de um comando seja elevado, chegando a demorar perto de 1 segundo [15].

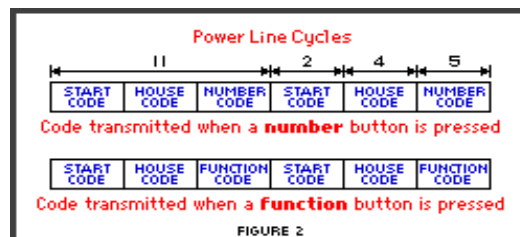


Figura 5: Formato dos códigos X10 transmitidos[17]

Como pode ser visto na Figura 5, cada código é constituído por um *Start code*, um *House code* e um *Number/Function code*. O *House code*, normalmente representado pelas letras de “A” a “P” (16 códigos), representa parte do endereço do dispositivo, podendo ser visto como

o código de grupo (ou casa) do dispositivo. O *Number code*, mais conhecido por *Unit code*, representa o identificador específico do dispositivo, que em conjunto com o *House code* forma o endereço do dispositivo na rede. No total podem ter-se 256 endereços de dispositivos distintos, ou seja, 16 grupos com 16 endereços distintos cada. O *Function code* representa a acção a ser realizada pelo equipamento [2]. A tabela dos códigos X10 é apresentada no Anexo A.

Cada trama deve ser enviada com uma codificação específica. O *House e Number/Function codes* devem ser enviados sob a forma de complemento real, isto é, se numa travessia pelo zero foi enviado um “1” na próxima travessia deverá ser enviado um “0” e caso tenha sido enviado um “0” a próxima travessia deverá conter um “1”. Esta codificação em complemento não se aplica ao *Starter code* cujo valor é sempre “1110”.

O protocolo X10 também apresenta o suporte para RF com modulação em frequência de uma portadora com frequência de 310 MHz. Também suporta IR (*InfraRed*) que em conjunto com o RF são utilizados em comandos remotos. O RF também é utilizado para interligar equipamentos X10 que se encontram em redes eléctricas isoladas [2], [15].

Um sistema X10 funciona segundo uma topologia mestre-escravo, onde um equipamento mestre (controlador X10) envia comandos para os equipamentos “escravos” (exemplo: Receptor X10). Os equipamentos “escravos” não apresentam “inteligência” para efectuarem acções sozinhos, sendo necessária a intervenção de um utilizador ou de uma aplicação no lado do controlador para enviar os comandos.

### 2.1.2.2 *Protocolo KNX*

É um protocolo de comunicação europeu, criado a 14 de Abril de 1999, com o principal objectivo de evitar a importação de material eléctrico dos mercados japoneses e americanos, através da definição de um único protocolo europeu para automação de residências e edifícios [3].

O protocolo KNX, muitas vezes referido como EIB, foi desenvolvido através da “convergência” de outros protocolos europeus já existentes, nomeadamente EHB, BatiBus e EIB. Esta junção deu origem à *KONNEX association* que é a associação que gere o protocolo [3].

Utilizando como base o EIB, o protocolo desenvolveu-se sobre uma arquitectura descentralizada onde a comunicação entre os elementos é efectuada ponto a ponto. Esta abordagem demonstra o crescente interesse no desenvolvimento de sistemas distribuídos cuja

inteligência encontra-se espalhada por cada elemento da rede, ao invés de concentrada num ponto central.

A interligação entre os elementos da rede pode ser efectuada segunda uma topologia de barramento, anel, estrela, árvore ou mista (Figura 6).

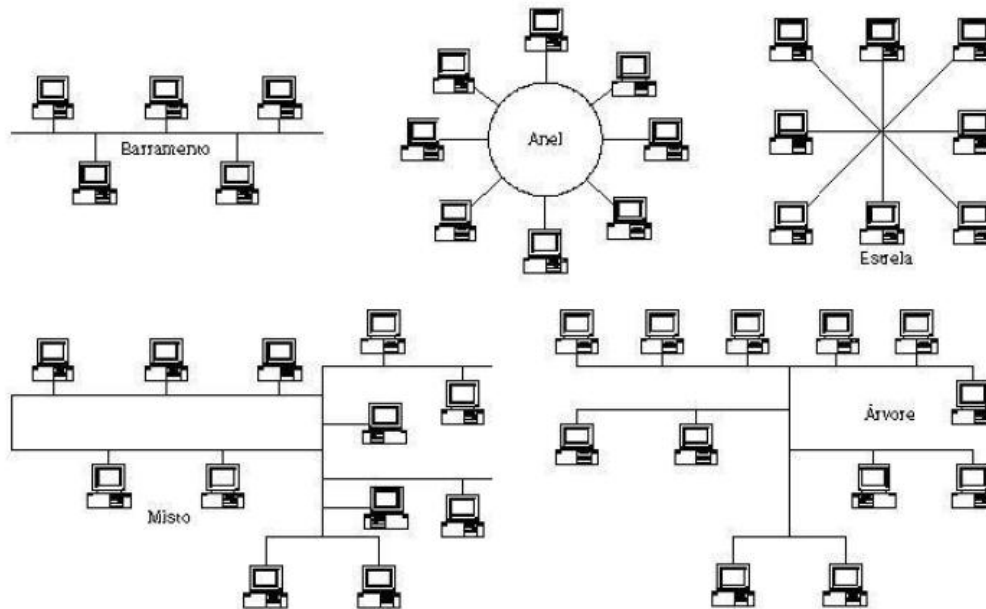


Figura 6: Topologias físicas de um sistema KNX [16]

Cada elemento apresenta um endereço individual de 16 *bits*, o que possibilita a implementação de uma rede com 65536 elementos.

É possível encontrar-se no mercado inúmeros equipamentos, de diversos fabricantes, que seguem o protocolo KNX/EIB. Os equipamentos são certificados pela *KONNEX association* garantindo assim a interoperabilidade dos equipamentos mesmo que estes sejam de fabricantes diferentes.

As vantagens do protocolo KNX são [16]:

- Elevada capacidade de equipamentos;
- Flexibilidade na escolha da topologia da rede;
- Certificação de equipamentos;
- Suporte para diversos meios físicos de transmissão.

As desvantagens do protocolo KNX são [16]:

- Complexidade de instalação;
- Custo elevado dos equipamentos.

Trata-se de um protocolo relativamente complexo cuja instalação, configuração e manutenção necessita de software especializado (ETS, *Engineering Tool Software*).

Outra desvantagem é o custo dos equipamentos que comparativamente com equipamentos de outros protocolos (como o X10) é extremamente caro.

O protocolo KNX define quase todas as camadas protocolares indicadas pelo modelo OSI.

Na definição da camada física do protocolo é indicado o suporte de uma grande variedade de meios físicos por onde os sinais podem ser transmitidos [18]:

- Cablagem com pares entrançados (TP1) – é utilizado um par de cabos entrançados para transmissão de dados e alimentação. A transmissão é efectuada de forma assíncrona sendo a unidade de transferência o carácter. Suporta comunicação bidireccional em *half-duplex* e implementa o CSMA/CA para evitar colisões de pacotes. Podem ser utilizadas todas as topologias (estrela, malha, árvore, etc.);
- Comunicação através de rede eléctrica (PL 110) – A transmissão é efectuada por pacotes em *half-duplex* bidireccional. Dados com uma taxa de 1200 *bit/s* são enviados numa portadora de 110 kHz, modulada com a técnica de espalhamento espectral;
- Comunicação por rádio frequência – Utiliza uma portadora de 868,3 MHz modulada em frequência, com os dados codificados em *Manchester*;
- Também apresenta soluções para suporte de meios como *Ethernet*, *Bluetooth*, *WiFi* e *FireWire*.

Cada meio físico apresenta um pequeno segmento da camada de ligação de dados é unido a um segmento genérico que proporciona o controlo de acesso ao meio e controlo da ligação lógica.

A camada de rede é normalmente só de interesse para os nós que efectuam o encaminhamento das tramas. Esta camada encarrega-se de enviar confirmações e controlar o número de saltos da trama.

A camada de transporte define suporte para 4 tipos de comunicação entre os nós da rede:

- *Multicast* (um para muitos)
- *Broadcast* (um para todos)
- Ponto a ponto sem ligação
- Ponto a ponto com ligação

A camada de aplicação oferece uma grande variedade de serviços para a aplicação que gere o sistema de automação. Os serviços estão dependentes do tipo de comunicação utilizado na camada de transporte, sendo os serviços associados a comunicações ponto a ponto utilizados para a gestão da rede, e os serviços associados a *multicast* direccionados para a operação da rede.

### 2.1.2.3 Protocolo ZigBee

Em 2004, o protocolo *ZigBee* foi lançado como uma solução para aplicações de baixo consumo e baixo custo. Foi um projecto iniciado pela *Philips*, que continuou a ser desenvolvido pela *ZigBee Alliance* [2].

O *Zigbee* foi desenvolvido a pensar nas aplicações com sensores sem fios cuja alimentação é fornecida por baterias (aplicações de baixo consumo). Devido ao suporte de construção automática de rede, que pode seguir topologias como a árvore, malha e estrela, o *ZigBee* tornou-se uma óptima solução para a distribuição de sensores e controladores numa residência, introduzindo ou expandindo um sistema de automação sem grande custo adicional [2].

O baixo preço dos equipamentos torna-o ainda mais atractivo para sistemas de automação residenciais, juntamente com a facilidade de instalação (não é necessário a passagem de cablagem).

O protocolo *ZigBee* define várias camadas do modelo OSI, tal como é representado na Figura 7.

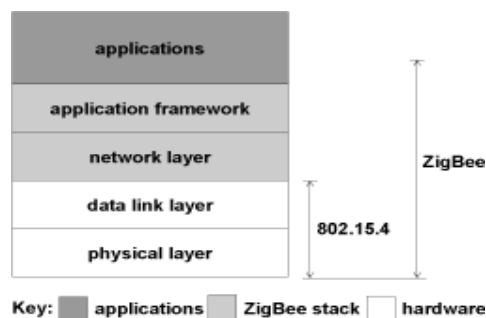


Figura 7: Pilha protocolar *ZigBee* [19]

As camadas física e de ligação de dados são definidas pelo protocolo IEEE 802.15.4 [2], [19]. Refira-se que a camada física do *ZigBee* suporta a comunicação por rádio frequência (RF) em três bandas ISM, nomeadamente 868 MHz, 915 MHz e 2,4 GHz. A técnica de modulação do sinal utilizada é o DSSS (*Direct Sequence Spread Spectrum*), fazendo com que o sinal se propague no meio (atmosfera) de forma semelhante a um ruído que só pode ser decodificado com a sequência pseudo-aleatória correcta. Utilizando as sequências pseudo-aleatórias apropriadas é possível fazer a separação dos sinais, permitindo assim a construção de vários canais de comunicação. O acesso a esses canais é controlado via CSMA/CD (*Carrier Sensing Multiple Access / Collision Detection*) [2].

As velocidades de transmissão podem ir dos 20 kbps (com 1 canal) até 250 kbps (com 16 canais) dependendo da banda de frequências utilizada. Com uma potência de transmissão de 1 mW obtêm-se alcances entre os 10 e 100 metros (no exterior). A baixa taxa de transmissão de dados, baixo consumo e curto alcance tornam este padrão de comunicação *Wireless* propício para sistemas de controlo e monitorização, que não necessitam de taxas de dados elevadas, necessitam de flexibilidade e facilidade na implementação das redes cujo alcance não é elevado [2], [20].

A camada MAC (*Medium Access Control*) do *ZigBee*, integrada na camada de ligação de dados, trabalha com 4 tramas com funções distintas:

- Trama de dados – transmissão normal de dados;
- Trama de resposta – controlo de fluxo;
- Trama de comando MAC – criação e configuração da rede;
- Trama “*Beacon*” – funcionalidades especiais dos controladores.

A camada de rede *ZigBee* permite a implementação das topologias malha, estrela e árvore. Independentemente da topologia as redes *ZigBee* contêm três tipos de nós [2]:

- Coordenador – Encarregue de fazer a construção da rede, através da indicação das ligações entre os vários nós por meio de uma *Binding table*;
- Router – Realiza o encaminhamento de mensagens para os nós vizinhos e permite expandir o alcance máximo da rede;
- *End devices* – São os nós associados à aplicação. Estes contêm os *End Points* que correspondem aos actuadores e sensores da aplicação. Estes nós são a origem e destino da informação da rede *ZigBee*.

Na camada de aplicação são definidos vários objectos do dispositivo *ZigBee* (ZDO) que determinam a funcionalidade e comportamento do dispositivo na resposta a eventos do sistema.

O *ZigBee Application Profile* define os conjuntos de *clusters Zigbee* a utilizar para uma determinada área de aplicação (automação doméstica, medição automática, monitorização industrial, etc.). Os *clusters Zigbee* por sua vez definem um conjunto de atributos e comandos para uma determinada área funcional (iluminação, climatização, segurança, etc.).

Existem também no mercado equipamentos que implementam o protocolo *ZigBee* só até à camada de rede inserindo uma API (*Application Programming Interface*), permitindo assim o desenvolvimento de camadas de aplicação personalizadas [2], [20].

#### 2.1.2.4 Protocolo LonTalk

O protocolo *LonTalk* é uma de várias implementações do protocolo CNP (*Control Network Protocol*) desenvolvido e gerido pela *Echelon*. Desenvolvido para suportar aplicações de controlo que englobam diversas áreas, é a base principal da plataforma de sistemas de controlo *LONWORKS*. A plataforma serve de base de desenvolvimento e operação de um sistema de controlo distribuído [4].

O protocolo define cada uma das 7 camadas do modelo OSI, sendo optimizado para aplicações de controlo. A definição de todas as camadas proporciona uma solução de comunicação robusta, que satisfaz os requisitos de muitas aplicações actuais e futuras. Algumas das características principais do CNP são [4]:

- Entrega eficiente de pequenas mensagens;
- Fiabilidade na entrega de mensagens;
- Detecção de mensagens duplicadas;
- Múltiplos meios de comunicação;
- Baixo custo de dispositivos, instalação e manutenção;
- Uso eficiente da largura de banda dos canais.

O protocolo suporta múltiplos meios de transmissão como [4]:

- Pares de cabos entrançados;
- Rede eléctrica (*Power Line Communications, PLC*);
- Rádio frequência;

- Fibra óptica;
- Infravermelhos.

Cada meio tem uma especificação própria que define o alcance, taxas de *bits* e topologias que são suportadas. A rede *LONWORKS* (rede que utiliza o protocolo *LonTalk*) é composta por um ou mais canais, sendo a forma física desta dependente do meio utilizado (portadora com uma determinada frequência em RF, um par de cabos em TP (*Twisted Pair*), etc.) [4].

A camada física TP/FT-10 (*Twisted Pair / Free Topology*) especifica a ligação utilizando um par de cabo entrançados com protecção (*Shielded*), com uma taxa de transferência de 78,125 kbps. A codificação é efectuada em *Manchester* diferencial. Este possibilita a interligação de 64 dispositivos num segmento da rede ou de 128 dispositivos utilizando uma fonte de alimentação DC [4]. A interligação dos equipamentos pode ser efectuada segundo as topologias indicadas na Figura 8.

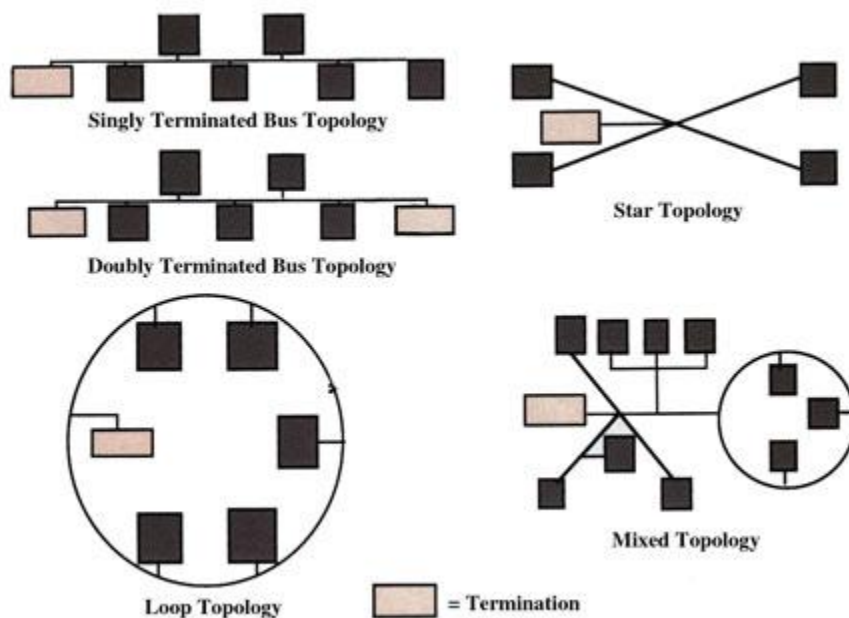


Figura 8: Topologias suportadas no meio TP/FP-10 [4]

Outro meio suportado é através da rede eléctrica com o canal PL-20 (*Power Line*), sendo este o meio de mais fácil instalação numa casa, edifício ou veículo. São utilizados sinais em banda estreita com frequências de 115 kHz a 132 kHz, com uma taxa 5,4 kbps, e 75 kHz a 86

kHz com uma taxa de 3,6 kbps. A fiabilidade da comunicação é garantida através de algoritmos de cancelamento de ruído e correcção de distorção [4].

Na camada de ligação de dados é utilizado um algoritmo da família CSMA. É um algoritmo único que permite que um canal sobrecarregado funcione próximos da sua capacidade máxima, ao contrário dos outros que têm a tendência a degradar quando estão próximo da capacidade máxima. O algoritmo CSMA é uma variante do algoritmo *p-persistent* designada *predictive p-persistent CSMA*. Este mantém os benefícios dos algoritmos CSMA, melhorando algumas falhas que surgiam em aplicações de controlo. Cada dispositivo da rede é identificado através de uma sequência de 48 *bits* única [4].

A camada de rede permite o encaminhamento de mensagens para um único dispositivo (ponto-a-ponto), para múltiplos dispositivos (*Multicast*) ou para todos os elementos (*Broadcast*).

A camada de transporte define a entrega de pacotes de resposta (*Acknowledge*) dos pacotes enviados e reenvio em caso de falha no recebimento da resposta. É possível também o envio de pacotes sem resposta ou o envio de vários pacotes antes de receber uma resposta. Também é definido como são detectadas e rejeitadas mensagens duplicadas na rede que podem ter sido enviadas devido à não recepção de uma resposta [4].

A camada de sessão suporta serviços de autenticação, pedido/resposta [4].

A camada de apresentação define as variáveis da rede disponíveis, formato da mensagem de e para a aplicação [4].

A camada de aplicação define muitos serviços para configuração, diagnóstico, transferência de ficheiros e gestão da aplicação, alarmes, agendamento, etc. [4].

O *LonTalk* é um protocolo extremamente complexo o que torna a implementação deste muito dispendiosa. Para facilitar o desenvolvimento e redução de custos a *Echelon* disponibiliza o *Neuron Core* num componente discreto denominado *Neuron Chip*. Este contém a implementação das camadas protocolares 2 a 6 do protocolo *LonTalk*. Para evitar a sobrecarga de um CPU com o tratamento das várias camadas protocolares, cada *Neuron Chip* tem 3 CPUs integrados (um para processamento da aplicação, um para processamento de rede e outro para processamento de ligação) [4], [21].

São também disponibilizados vários transmissores/receptores (*transceivers*) que implementam as várias camadas físicas suportadas pelo protocolo, já preparados para associação com um *Neuron Chip*. Os componentes podem ser comprados separadamente mas também

podem ser comprados em conjuntos chamados *Smart Transceivers* que é a combinação de um ou mais *transceivers* e um *Neuron Chip* [21].

## 2.2 Conversores de protocolos

O desenvolvimento de várias arquitecturas de redes e protocolos de comunicação distintos tem dificultado a integração de sistemas [22].

Um dos grandes desafios da interligação de protocolos é garantir a interoperabilidade num sistema com diferentes protocolos, sendo normalmente utilizados nestas situações equipamentos que realizam a conversão entre os mesmos [22], [23].

Uma possível solução para garantir a interoperabilidade dos equipamentos seria a definição de um protocolo de comunicação comum, ou utilização de um já existente (Exemplo: TCP/IP), que todos os equipamentos deverão utilizar para comunicar [22].

A conversão de protocolos é uma prática muito utilizada pois não é simples desenvolver ou adoptar um protocolo comum visto, já existirem no mercado, e instalados em edifícios inúmeros sistemas que utilizam protocolos distintos. A solução de adopção de um protocolo comum iria desvalorizar todo o trabalho realizado para implementar os sistemas e protocolos já existentes, tornando essa solução indesejável para muitos fabricantes [22].

No entanto, a conversão de protocolos é uma acção complexa sendo necessário um levantamento das características de ambos os protocolos e a análise de métodos para converter as informações de um protocolo para informações que o outro protocolo entenda [23]. Para dificultar ainda mais a tarefa, não existe uma teoria geral de como efectuar uma conversão e verificar se a conversão é a mais correcta ou útil [22].

Os conversores de protocolos, normalmente conhecidos como *gateways*, são utilizados para fazer a interligação entre duas ou mais redes que utilizam protocolos distintos. A *gateway*, de um modo geral, implementa as 7 camadas do modelo OSI, de forma a realizar com sucesso uma conversão entre protocolos, como pode ser visto na Figura 9. A implementação das 7 camadas garante a correcta interpretação e construção de pacotes nas várias camadas dos protocolos [23].

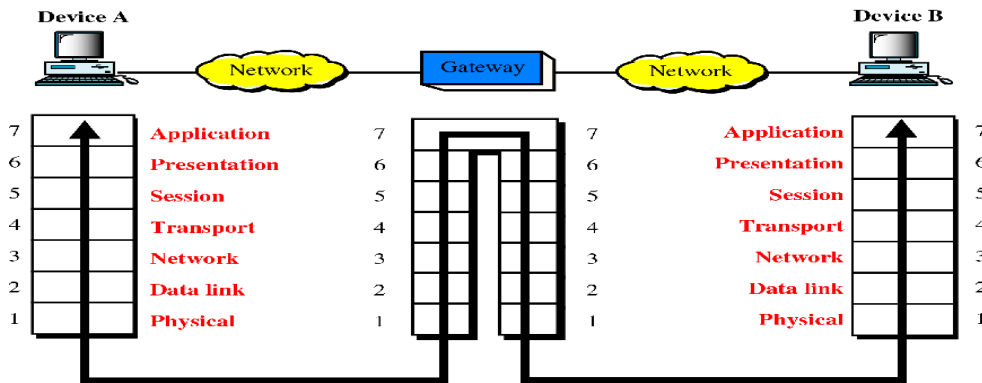


Figura 9: Arquitectura protocolar de uma gateway [24]

Existem no mercado conversores de protocolos que podem ser utilizados para fazer conversões de 1 para 1 como, por exemplo, o ZG-2400E que realiza a conversão entre TCP/IP e *ZigBee* [25]. Naturalmente é mais simples a conversão entre dois protocolos, mas torna-se mais dispendiosa e complexa a tarefa quando se pretende converter entre vários protocolos, sendo necessário um conversor por cada par de protocolos a interligar.

Existem também alguns conversores de protocolos que realizam a conversão de 1 protocolo para múltiplos, mas estes são relativamente dispendiosos quando comparados com soluções sem conversão de protocolos (utilização de equipamentos do mesmo protocolo). Um exemplo de um conversor de protocolos com capacidade de conversão de múltiplos protocolos é o *Harmony Multi-Protocol HAN (Home Automation Network) Gateway/Controller* (Figura 10).



Figura 10: Gateway Multiprotocolo Harmony [26]

O equipamento representado na Figura 10 permite efectuar a interligação entre uma rede *Ethernet* e uma rede sem fios ou um sistema de automação doméstica [26].

A gateway multiprotocolo *Harmony* possui um rádio compatível com o protocolo IEEE802.15.4 que pode ser configurado para suporte *ZigBee Pro*, 6LoWPAN e *JenNet*.

Também possui um interface para interligação com um módulo de comunicação pela rede eléctrica com suporte aos protocolos *Insteon* e X10 [26].

O equipamento da Figura 10 inclui software de automação doméstica e de gestão de energia, que corre sobre um sistema operativo Linux, tornando mais fácil o desenvolvimento de novas aplicações [26].

### 2.3 *Smart grids* (Redes eléctricas inteligentes)

Uma rede eléctrica inteligente pode ser vista como uma rede eléctrica normal na qual são introduzidos sistemas de medição e controlo inteligentes, tais como [12]:

- Tecnologias de informação;
- Sensores;
- Comunicação de alta velocidade;
- Armazenamento de energia;
- Geração distribuída de energia;
- Controladores de energia domésticos;
- Gestão automática do consumo de energia.

A rede inteligente permite a integração de geradores de energia distribuídos, possibilitando, por exemplo, aos consumidores a instalação de geradores eólicos e fotovoltaicos domésticos, tornando-se assim microprodutores de energia eléctrica. Os geradores domésticos permitem aos consumidores reduzirem a sua conta de energia eléctrica e ainda a possibilidade de vender o excesso para a rede [12].

As *smartGrids* também permitem que veículos eléctricos (híbridos ou eléctricos normais) armazenem energia em alturas que o preço é baixo e que descarreguem essa energia em alturas em que o preço é elevado. Tudo isto só é possível se a infra-estrutura inteligente permitir a actualização do preço da energia e fazer com que esta seja disponibilizada a um controlador doméstico, através de ligações de alta velocidade [12].

O controlador doméstico necessário para as acções descritas poderá ser dedicado ou fazer parte do sistema de automação doméstico, permitindo assim acesso a muitas mais opções e soluções de gestão de energia.

As características principais de uma rede eléctrica inteligente são [12]:

- Variação do preço da energia em tempo real;
- Capacidade para acomodar as tecnologias de armazenamento e geração de energia distribuídas;
- Elevado nível de participação do utilizador;
- Optimização da rede para operação eficiente;
- Previsão e resposta a distúrbios na rede;
- Sistema mais económico e ecológico.

Já se encontram implementadas algumas redes eléctricas inteligentes sendo um dos exemplos, uma das primeiras e maiores redes inteligentes, instalada em Itália pela Enel S.p.A [27]. A rede foi desenvolvida dentro do projecto Telegestore, através de utilização de medidores e *software* de gestão desenvolvidos pela própria companhia [12].

Em Évora o projecto InovGrid é referido como a primeira implementação de uma rede de energia inteligente em Portugal [28]. Está a ser desenvolvido pela EDP Distribuição. A rede inteligente contém diversos sensores e actuadores instalados que permitem a monitorização e controlo do estado da rede através do balanceamento de cargas, prevenção e detecção de avarias.

Esta rede reage a situações como a injeção de energia na rede e o aumento do consumo, através de um sistema de controlo automático. Este sistema de controlo possibilita o redireccionamento da energia, em casos de avaria, evitando falhas no fornecimento de energia eléctrica.

### **2.3.1 Sistemas de despacho automático (*Automatic Demand Response*)**

Um sistema de despacho automático é um sistema que permite aos consumidores finais dar a sua contribuição no funcionamento da rede eléctrica inteligente, através da redução do consumo em resposta ao estado da rede eléctrica [29].

Numa rede tradicional o consumidor final não possui informação suficiente para dar a sua contribuição para o bom funcionamento da rede eléctrica. Com a introdução de novas tecnologias e da comunicação de alta velocidade na rede eléctrica (tecnologias de *Smart Grids*), através de aparelhos como contadores inteligentes, o utilizador dispõe da informação necessária para ajudar a gerir o seu consumo e contribuir para um funcionamento mais eficiente e fiável da rede eléctrica [30].

O despacho de energia refere-se a modificações no consumo de energia em resposta a mudanças do preço ao longo do tempo ou a incentivos para a redução do consumo (custo) em alturas de maior consumo ou sobrecarga [30]. O despacho automático de energia implica que as modificações do consumo são efectuadas sem intervenção do utilizador.

Estes sistemas permitem, além da melhoria da fiabilidade da rede eléctrica, reduzir a longo prazo o consumo, removendo a necessidade imediata da construção de novas estações geradoras. Permitem também reagir em situações de redução de produção bruscas, muito frequentes com a integração de geradores distribuídos com base em energias renováveis [30].

Em alturas de picos de consumo, o sistema de despacho consegue responder disponibilizando energia para as novas cargas através do redireccionamento da energia de outras cargas (desactivação de cargas). Este redireccionamento de energia é um processo mais rápido (apesar de levar 5 minutos a responder) do que activar uma estação geradora para alturas de picos (o que pode levar até 30 minutos) [30].

Em termos de custo, é também mais barato reencaminhar a energia durante curtos períodos de pico, do que activar uma estação geradora para responder ao pico de consumo [30].

Os sistemas de despacho podem ser divididos em duas categorias [31]:

- Baseado em incentivos
- Baseado no tempo.

Nos sistemas de despacho baseados em incentivos os consumidores são pagos para reduzir o consumo em alturas de pico de consumo ou de consumo elevado, segundo informações indicadas pelo fornecedor de energia eléctrica [31].

Nos sistemas de despacho baseados no tempo, os consumidores têm acesso em tempo real aos preços da electricidade que reflectem os níveis de consumo e produção [31].

Os sistemas baseados no tempo dependem muito da escolha dos clientes consumirem mais em alturas em que o preço da electricidade é menor.

A variação dos preços ao longo do dia pode dar origem a dois tipos de reacções [12]:

- Reduções do consumo com o aumento de preço, para cargas que não podem ser agendadas para outras alturas (iluminação, ar condicionados, etc.);
- Agendamento da activação de cargas para outras alturas de menor custo (maquinas de lavar, veículos eléctricos, carregadores de baterias, etc.).

A primeira reacção exige um equilíbrio entre o conforto e a poupança de energia e a segunda reacção exige uma mudança dos hábitos de consumo. Ambas exigem sempre uma redução do conforto do utilizador, daí a adesão a estes sistemas ainda não ser muito atractiva, pois os consumidores não querem reduzir o conforto. Com um baixo nível de adesão aos sistemas de despacho, os efeitos destes na rede são muito menores que os pretendidos, contribuindo pouco para a melhoria do desempenho da rede eléctrica. Incentivar mais consumidores a aderir a programas de despacho irá fazer com que os sistemas baseados no tempo apresentem melhores resultados permitindo o controlo do preço da energia e da forma dos diagramas de carga [12].

Na Figura 11 pode-se verificar o efeito no consumo, obtido através do uso de sistemas de despacho baseados no tempo.

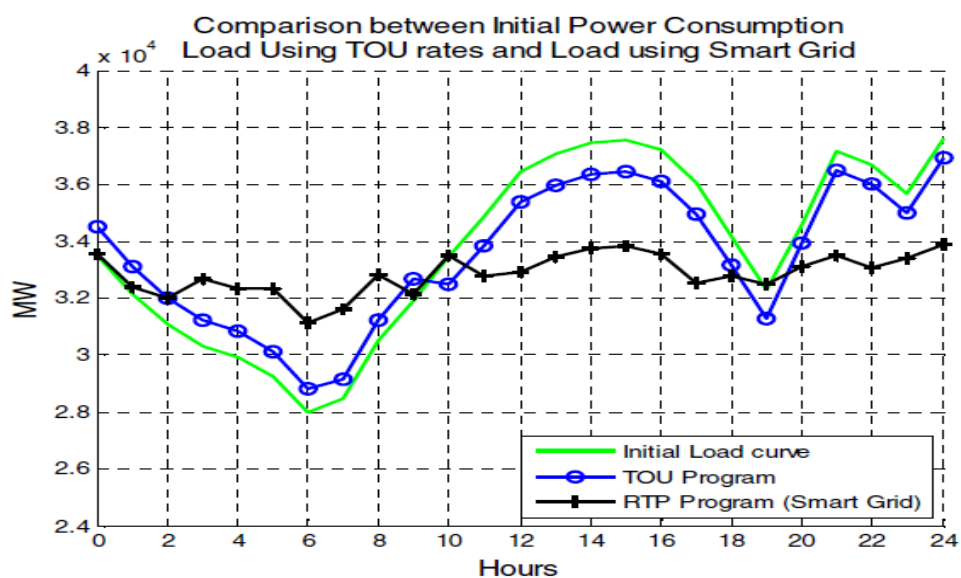


Figura 11: Comparação do diagrama de carga sem sistema de despacho, com sistema RTP e com sistema TOU [31]

O sistema TOU (*Time of Use*) baseia-se na definição de preços fixos para determinadas alturas do dia (como as tarifas bi-horária e tri-horária).

O sistema RTP (*Real Time pricing*), apresenta uma variação em tempo real do preço da energia que reflecte os níveis de produção e de consumo ao longo do dia, possibilitando um melhor controlo do consumo ao longo do dia.

Em ambos os sistemas verifica-se um aumento do consumo nas alturas de menor consumo e redução nas alturas de maior consumo, ou seja como o comportamento que um

sistema de despacho deverá apresentar, ajudando no bom funcionamento da rede e reduzindo o custo mensal da electricidade do consumidor.

Nota-se claramente que o sistema RTP provocou um “alisamento” do diagrama de carga devido à monitorização em tempo real do preço da energia e à gestão das cargas, com base nesse preço. Este “alisamento” facilita a previsão dos níveis de consumo.

Um exemplo de um sistema de despacho implementado é o do edifício da sede da companhia *Echelon* em San Jose, Califórnia. Este sistema de despacho utiliza a rede de automação do edifício para efectuar a gestão de energia, tendo acesso aos sistemas de HVAC e iluminação.

O sistema permite a monitorização e o controlo remoto através da Internet (Figura 12) [32].

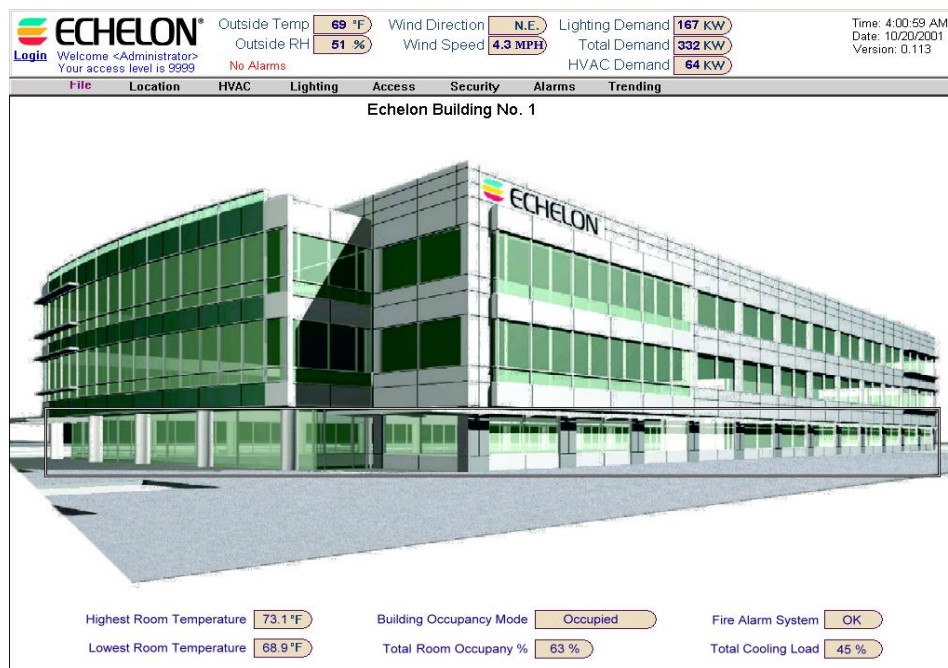


Figura 12: Interface remoto para gestão do edifício [32]

No interface da Figura 12 é possível visualizar-se a informação de todo o edifício desde os consumos de energia até aos níveis de ocupação do edifício.

Cada funcionário possui um interface onde selecciona as opções pessoais que são utilizadas pelo sistema de despacho (Figura 13). Quando o sistema necessita de responder a um evento serão utilizadas as opções indicadas para reduzir o consumo.

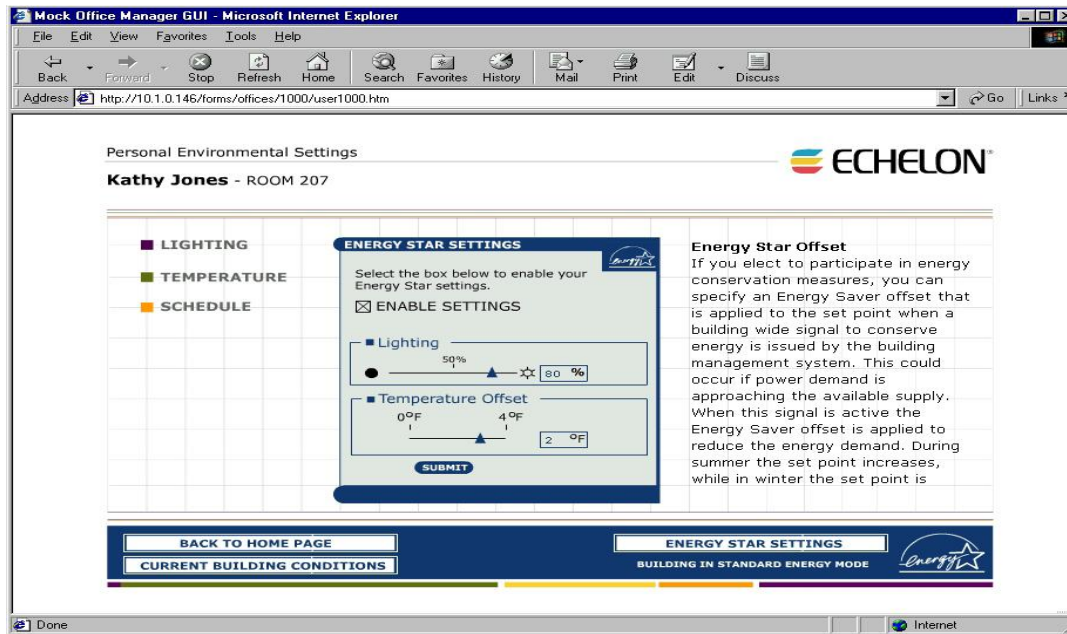


Figura 13: Interface de opções de configuração do sistema de despacho [32]

As opções disponíveis permitem reduzir a intensidade luminosa para uma determinada percentagem e a redução do nível de temperatura em alguns graus. A definição da percentagem de luminosidade a reduzir permite efectuar um balanço entre o conforto e a redução de consumo.

O sistema de despacho reage à ocorrência de picos de consumos elevados, indicados pela companhia que fornece a energia eléctrica. Uma mensagem de evento é enviada pela Internet, na altura de um pico, e recebida pelo servidor que encaminha a informação do evento para o sistema de automação. O sistema de automação responde actuando nos dispositivos, utilizando as opções pessoais de cada funcionário (Exemplo: reduzir intensidade luminosa para 50%).

Durante um estudo realizado em 2006, o sistema de despacho demonstrou que consegue reduzir rapidamente (em menos de 3 minutos) 30% do consumo de energia do edifício em resposta a um evento, sem afectar os ocupantes do edifício [32].

### 3 Sistema de automação

Para se atingir os objectivos propostos necessita-se de um sistema de automação com capacidade de monitorização, processamento da informação e capacidade de actuação sobre equipamentos domésticos.

Foi proposto um sistema centralizado, semelhante ao proposto em [16], que comunica com equipamentos de automação baseados em diferentes protocolos (Figura 14).

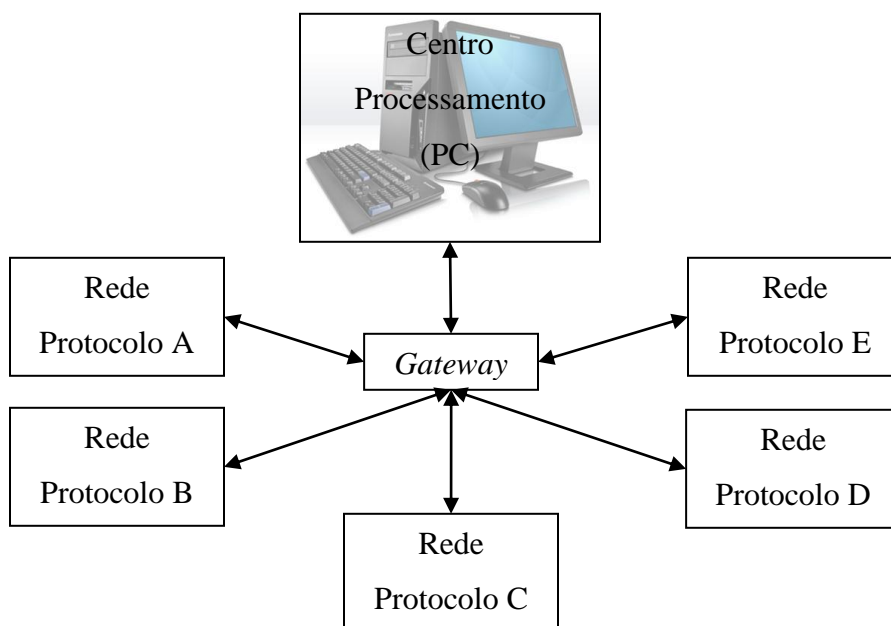


Figura 14: Arquitectura do sistema proposto

A principal dificuldade surge na interligação entre os diferentes protocolos, sendo proposta a utilização de uma *gateway* para realizar a conversão de protocolos.

O centro de processamento do sistema foi baseado num computador, onde iriam correr as aplicações de configuração e gestão da *gateway* e o sistema de despacho de energia.

O computador comunica com a rede de automação através da *gateway*, que realiza através das conversões uma abstracção entre o computador (Aplicação) e os equipamentos da rede.

A rede de automação é construída por equipamento que suportam diferentes protocolos, sendo a rede composta por diversas sub-redes, uma para cada tipo de protocolos.

A *gateway* realiza a abstracção fazendo com que os diversos equipamentos, com diferentes protocolos, sejam vistos como pertencentes ao mesmo protocolo, isto é, um

equipamento do protocolo A vê os outros equipamentos como pertencentes ao protocolo A, mesmo que na realidade pertençam a outros protocolos.

Uma das vantagens deste sistema em relação ao sistema proposto em [16] é que não existe a necessidade de um computador para realizar a interligação dos protocolos, podendo estes comunicar entre si, através da *gateway* sem a necessidade do computador para realizar a interligação.

Esta solução permite também uma maior facilidade de desenvolvimento de aplicações pois essas aplicações não necessitam implementar a comunicação com diferentes protocolos, sendo somente necessário saber comunicar com um único protocolo.

O sistema apresenta uma grande facilidade de instalação, baixo custo e simplicidade. Estes factores estão muito dependentes dos equipamentos e protocolos escolhidos, sendo de seguida efectuada uma análise das características dos diferentes protocolos escolhidos e equipamentos que os suportam.

Refira-se que a *gateway* encontra-se preparada para a implementação de um sistema distribuído, em que a inteligência do sistema se encontra em cada equipamento de automação.

### 3.1 Protocolos escolhidos para o protótipo

Como um dos objectivos era a implementação de um sistema de gestão de energia doméstico, escolheram-se os protocolos utilizados em equipamentos que apresentam baixos custos e simplicidade de instalação, de forma a ser mais atractivo à adesão.

O protocolo X10 é um protocolo simples e os equipamentos apresentam facilidade de instalação e baixo custo. Por estas razões foram escolhidos equipamentos de actuação de cargas (lâmpadas e electrodomésticos) que proporcionam facilidade de instalação em residências e com baixo custo por actuador (aproximadamente 17,13 € por cada interruptor de lâmpadas e 29,37 € por cada interruptor de electrodomésticos, preço de venda em [33]).

Foram escolhidos dois actuadores X10 para lâmpadas e um para electrodomésticos, com custo total dos equipamentos de aproximadamente 63,63 €. Para controlar estes equipamentos foi necessário a adquirir de um controlador XM10E, cujo custo foi de 43,21 € (preço de venda em [33]), obtendo-se um total de 106,84 €.

Para construir uma rede de sensores numa residência sem o custo adicional de instalação de cablagem, pode ser utilizado o protocolo *ZigBee*. Uma das razões para a escolha deste protocolo deveu-se à disponibilidade de alguns módulos em laboratório. O baixo custo desses

módulos permite reduzir o custo total do sistema. Um módulo *XBee* (módulo que implementa protocolo *ZigBee*) pode ser adquirido a um preço de 26,31 € (preço de venda em [34]). A partir de um módulo *XBee* podem ser desenvolvidos equipamentos como: sensores, actuadores ou interfaces com o utilizador. A característica de criação automática da rede de sensores *ZigBee*, que possibilita a adição de novos elementos e a flexibilidade na escolha da topologia da rede, trás para o sistema a vantagem de facilmente e rapidamente ser possível implementar uma rede de automação. Na implementação do protótipo foram utilizados dois módulos *XBee* (1 coordenador para a *gateway* e um nó folha (*End Device*) para transmitir pacotes), apresentando um custo de 52,62 €.

O GSM (*Global System for Mobile communications*) não é um protocolo de automação, mas foi introduzido para facilitar a transmissão de mensagens e avisos para o utilizador e também para possibilitar a interacção através do envio de comandos básicos (como acender ou apagar uma lâmpada). A troca de mensagens é efectuada através do serviço SMS (*Short Message Service*) que está disponível em todos os telemóveis e que pode ser utilizado em qualquer lugar com cobertura da rede telefónica móvel (alcance elevado).

Um protocolo intermédio foi desenvolvido para facilitar a conversão entre os protocolos indicados anteriormente. É um protocolo pequeno e simples, que permite maior facilidade na conversão de protocolos, gestão da *gateway* e acesso à rede constituída pelos vários protocolos. O protocolo *ZigBee*, ou até outro protocolo existente, poderia ter sido utilizado como protocolo intermédio, mas devido à sua complexidade este tornaria mais difícil a implementação de algoritmos de conversão. O protocolo intermédio apresenta funções básicas, que servem para fazer a conversão entre os protocolos, visto ser mais fácil a implementação de algoritmos de conversão que dão uso a funções básicas para a implementação de funções mais complexas ou a fragmentação de funções complexas em várias funções básicas.

A *gateway* apresenta suporte para a integração de mais protocolos, mas para não aumentar demasiado a complexidade do sistema, e não sobrecarregar o *Firmware* e a ocupação dos recursos da *gateway* optou-se por limitar o sistema a estes 4 protocolos (3 existentes e 1 implementado).

O sistema herda muitas das suas características dos protocolos escolhidos, como as características de baixo custo dos equipamentos e a facilidade de instalação. Isto possibilita a implementação de um sistema adaptável a diferentes aplicações, pela escolha dos protocolos.

Devido à flexibilidade na escolha dos protocolos utilizados, foi possível implementar um sistema de automação de menor custo quando comparado com um sistema limitado a um só protocolo. Um exemplo da redução de custos pode ser visto quando comparando um sistema baseado somente no protocolo KNX, onde cada equipamento de actuação de duas lâmpadas custa 75,30 € (preço de venda pela DOSAPAC), e o sistema protótipo implementado, em que os dois equipamentos de actuação X10 utilizados custam um total de 34,6 € (45,9% do preço do equipamento KNX).

### 3.2 Protótipo da *gateway*

Na implementação do protótipo da *gateway* optou-se por utilizar ao máximo soluções integradas para a comunicação através de cada protocolo. Isto permitiu uma redução do tempo de desenvolvimento e da dimensão da *gateway*, através da utilização de componentes integrados em vez de discretos.

A arquitectura adoptada para a interligação de *hardware* é centralizada, sendo o elemento central o microcontrolador (Figura 15).

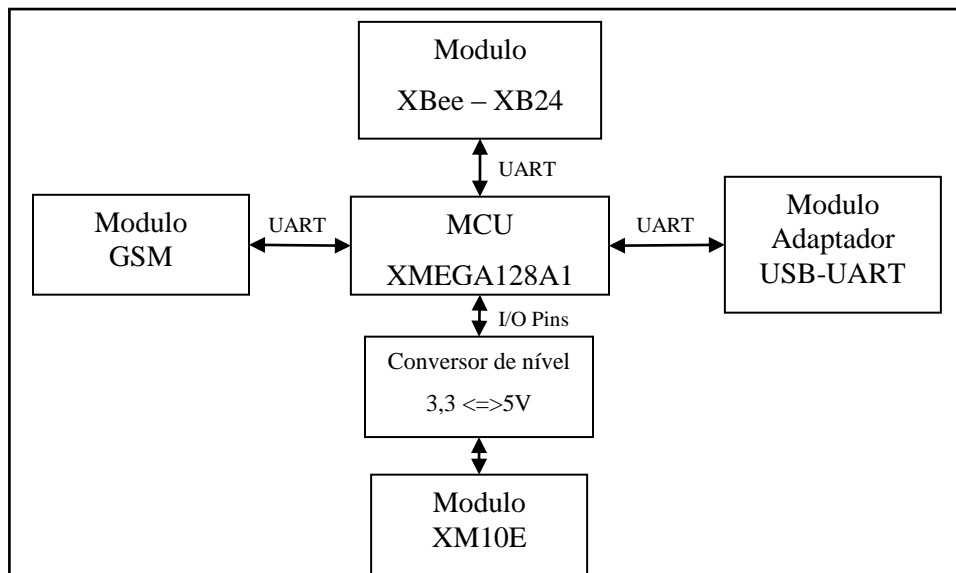


Figura 15: Arquitectura do *hardware* da *gateway*

Muitos dos módulos interligam-se directamente com o microcontrolador devido aos níveis de tensão serem os mesmos (3,3 V). A excepção é a interligação com o módulo X10 (XM10E) que necessita de um conversor de níveis, pois o módulo X10 funciona com 5 V enquanto o microcontrolador funciona com 3,3 V.

A comunicação com o módulo X10 é realizada através de um conversor de nível, utilizando um *buffer* com saída em colector aberto. O esquema de ligações do protótipo e desenho da placa PCB (*Printed Circuit Board*) da *gateway* podem ser encontrados nos Anexo B e C, respectivamente.

O conversor de nível é alimentado por um regulador *Step Down* de 5 V na placa PCB da *gateway*, cuja saída também é encaminhada para *headers* que podem ser utilizadas para alimentar outros módulos externos. O mesmo é feito com a saída de 3,3 V.

O módulo GSM apresenta uma entrada de alimentação própria que é fornecida por um transformador externo de 5 V, capaz de fornecer 2,5 A de corrente. O módulo requer uma fonte de alimentação capaz de fornecer picos de corrente com 2 A. Optou-se por utilizar o transformador externo em vez da alimentação da *gateway*, para evitar a sobrecarga do regulador de 1 A, e possibilitar a utilização do módulo GSM de forma individual, permitindo a instalação a uma maior distância da *gateway*.

### 3.2.1 MCU (MicroController Unit) XMEGA128A1

O microcontrolador escolhido para a *gateway* pertence à família XMEGA da Atmel, e utiliza um CPU com arquitectura AVR.

De entre as inúmeras características do microcontrolador as consideradas mais importantes para este tipo de aplicação são:

- Controlador DMA (*Direct Memory Access*) com 4 canais;
- Sistema de eventos com 8 canais;
- 8 Contadores de 16 *bits*;
- 8 portas USART (*Universal Synchronous Asynchronous Receive Transmit*) com capacidade de modulação e desmodulação de sinais IrDA (*Infrared Data Association*) numa das portas;
- 4 portas de comunicação I<sup>2</sup>C;
- 4 portas de comunicação SPI;
- Motor de encriptação AES e DES;
- 2 conversores Analógico-digital de 12 *bits* com 8 canais;
- 2 conversores Digital-Analógico de 12 *bits* com 2 canais;
- Controlador de interrupções programável com vários níveis de prioridade.

As capacidades deste microcontrolador permitem a interligação do mesmo com vários dispositivos externos, como sensores e *transceivers*, devido à quantidade elevada de periféricos de comunicação (16 ou mais dispositivos utilizando os periféricos USART, I<sup>2</sup>C e SPI, dependendo das topologias adoptadas).

Embora as características pareçam exageradas para a implementação deste protótipo, este microcontrolador foi escolhido pensando numa futura expansão das capacidades da *gateway*.

O MCU utilizado encontrava-se montado numa placa PCB (AVR-HX128A1) com os componentes mínimos para o funcionamento do mesmo, disponibilizado para venda pela OLIMEX, a um preço de 21,95 € (Figura 16) [35].

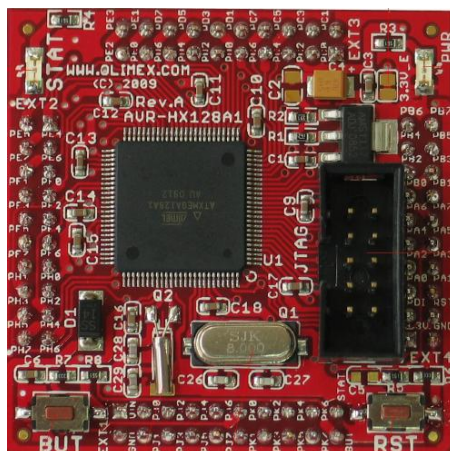


Figura 16: Placa PCB do microcontrolador XMEGA128A1

Na placa está incluído um cristal de 8 MHz, que pode ser utilizado como relógio principal do sistema, e um cristal de 32 kHz, para ser utilizado num temporizador em tempo real. O microcontrolador também possui relógios internos de 2 MHz e 32 MHz. De momento é utilizado o relógio interno de 2 MHz, mas tal pode ser modificado através de programação para 8 MHz externo ou 32 MHz interno.

A placa inclui também um regulador LM1117 *Step Down* de 3,3 V que recebe como entrada 5 V a 9 V, e dá uma saída regulada e estável de 3,3 V. Este regulador alimenta o microcontrolador e o módulo XBee.

Na placa PCB também está incluído um botão de RESET do microcontrolador e dois LEDs, um para a alimentação e outro para sinalização do estado da *gateway*.

### 3.2.2 Módulo XBee

A solução para integração com o protocolo *ZigBee* foi um módulo de rádio frequência *XBee Series 2* produzido pela MaxStream [36].

Estes foram fabricados para implementar o protocolo *ZigBee* operando na banda ISM (*Industrial Scientific and Medical*) de 2,4 GHz. No módulo estão implementadas as camadas física, de ligação de dados e de rede do protocolo *ZigBee*, reduzindo o tempo de desenvolvimento necessário [36].

Estes módulos apresentam as seguintes características [36]:

- Baixo consumo (40 mA em transmissão/recepção e < 1  $\mu$ A adormecido);
- 250 kbps de taxa de transmissão em rádio frequência;
- Potência de transmissão de 2 mW;
- Alcance de 40 m (interior) a 120 m (exterior);
- Interface com módulo *XBee* por comunicação série com suporte dos modos de comando AT e API.

O módulo *XBee* interligado ao MCU foi programado com o *firmware ZigBee Coordinator API*. Este *firmware* permite a configuração do módulo como coordenador da rede *ZigBee* e o envio e recepção de dados da rede *ZigBee*, por meio do envio de tramas API específicas pela porta série do módulo.

Nesta implementação focou-se a utilização das tramas de envio e recepção de mensagens (com API\_ID 0x10 e 0x90) podendo as outras tramas também ser utilizadas em futuras expansões das capacidades do módulo *gateway*. As restantes tramas API disponíveis são indicadas no Anexo D.

Para o funcionamento básico do módulo foram efectuadas apenas as ligações essenciais dos pinos RX (Pino 3) e TX (Pino 2), o RST (Pino 5), 3,3V (Pino 1) e GND (Pino 10), tal como se poder ver na Figura 17.

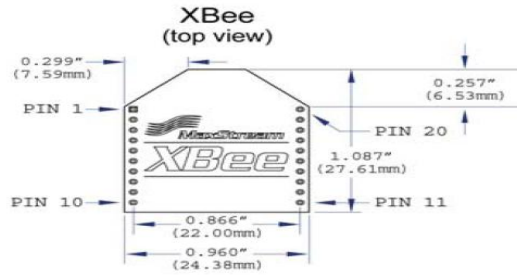


Figura 17: Dimensões e numeração dos pinos do módulo XBee [36]

Os pinos TX e RX da UART do XBee foram ligados directamente aos pinos RX e TX, respectivamente, de uma porta USART do microcontrolador.

### 3.2.3 Módulo Adaptador USB – UART

O módulo MM232R é utilizado para comunicação de dados com um computador. Este realiza a conversão da comunicação série UART para USB (*Universal Serial Bus*). O dispositivo é detectado no computador como uma porta COM virtual.

Este módulo disponibiliza uma saída de alimentação de 5 V, fornecida pelo USB, que foi aproveitada para alimentar a *gateway*.

Para o módulo funcionar com a alimentação do barramento USB foi necessário ligar o pino USBPWR (pino 15) ao VCC50 (pino 14) e o pino VCCIO (pino 3) ao VCC30 (pino 4) segundo é especificado no manual do componente (Figura 18) [37].

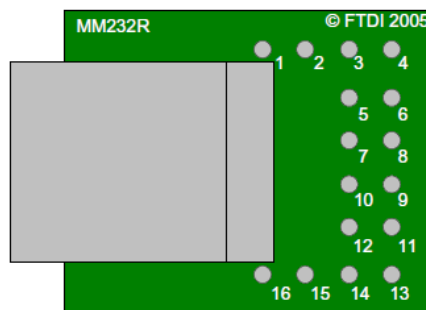


Figura 18: Esquema de pinos do adaptador UART-USB [37]

Os pinos TXD (pino 5) e RXD (pino 7) foram ligados directamente aos pinos RX e TX, respectivamente, de uma porta USART do microcontrolador.

### 3.2.4 Módulo X10

O módulo XM10E implementa a camada física do protocolo X10, permitindo o envio de sinais através da rede eléctrica. Este módulo é ligado a uma tomada normal no mesmo segmento da rede eléctrica que os dispositivos a controlar.

Este módulo tem uma saída opto-acoplada de um sinal em onda quadrada (Pino 1, ver Figura 19) cujas transições são sincronizadas com a passagem pelo “zero” da rede eléctrica. Esse sinal é utilizado para sincronizar o microcontrolador no envio dos códigos X10.

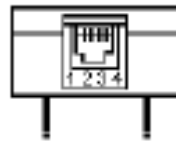


Figura 19: Pinout do módulo XM10E

Este módulo trabalha com níveis de tensão de 5 V, sendo neste caso necessário utilizar um conversor de nível pois o microcontrolador trabalha com tensões de 3,3 V. O circuito deste conversor é indicado no esquema de ligações da *gateway* (Anexo B). A alimentação deste módulo é obtida através da rede eléctrica onde este se liga através de uma tomada macho.

### 3.2.5 Módulo GSM

O módulo GSM utiliza o módulo TC35 da Siemens, inserido numa placa PCB de desenvolvimento (Figura 20).



Figura 20: Placa PCB do módulo GSM

Na placa PCB encontra-se um regulador de 5 V para 3,3 V (LM1117) que fornece a alimentação ao módulo TC35.

O módulo TC35 está conectado à placa PCB por um *socket* ZIF (*Zero Insertion Force*) e os pinos do mesmo são encaminhados para *headers*.

A comunicação com o módulo TC35 é efectuada através de uma porta UART, na qual os pinos RXD e TXD são conectados aos pinos RX e TX, respectivamente, de uma porta USART do microcontrolador.

Como a placa PCB do módulo GSM é separada da placa da *gateway* os pinos do módulo GSM são conectados à placa da *gateway* por meio de fios flexíveis.

Uma fotografia do protótipo implementado é apresentada no Anexo E.

### 3.3 *Firmware de uniformização de protocolos*

O *firmware* implementado no microcontrolador foi baseado na linguagem de programação C. Para desenvolvimento e compilação do código foi utilizado o AVR Studio versão 4.18, da Atmel em conjunto com o compilador WinAVR.

O código do *firmware* foi distribuído por vários ficheiros “.c” e “.h”, para uma maior organização e facilidade de modificação e expansão do código (o código desenvolvido é listado no Anexo F).

No ficheiro “comum.h” encontram-se as definições das estruturas e constantes utilizadas juntamente com os principais protótipos das funções comuns a todos os outros ficheiros. As estruturas definidas mais importantes são as estruturas de dados dos dispositivos, a estrutura dos pacotes internos, as rotas e as estruturas das tabelas e *buffers*.

É o ficheiro “main.c” que contém o código principal encarregue de chamar as funções para a conversão e encaminhamento dos pacotes que chegam ao módulo. Este código também se encarrega de modificar as tabelas de dispositivos e rotas.

Cada protocolo integrado é implementado em 2 ficheiros. O ficheiro “.h” que contém a definição de todas as estruturas e constantes necessárias para trabalhar e converter o protocolo e o ficheiro “.c” que contem as funções utilizadas para receber, enviar e converter o protocolo. Para facilitar a identificação o nome dos ficheiros indica qual o protocolo a que estão associados (Exemplo: “x10.h” e “x10.c” ou “GSM.h” e “GSM.c”).

No desenvolvimento do programa optou-se por utilizar os recursos do microcontrolador, nomeadamente o DMA e o tratamento de interrupções, de forma a reduzir os atrasos de conversão, através da realização de algumas tarefas de forma concorrente.

### 3.3.1 Método de conversão dos protocolos

A *gateway* efectua a conversão entre protocolos utilizando um protocolo intermédio, isto é, para converter de um protocolo para outro é realizada primeiro uma conversão para um protocolo intermédio e depois este é convertido para o protocolo desejado (Figura 21).

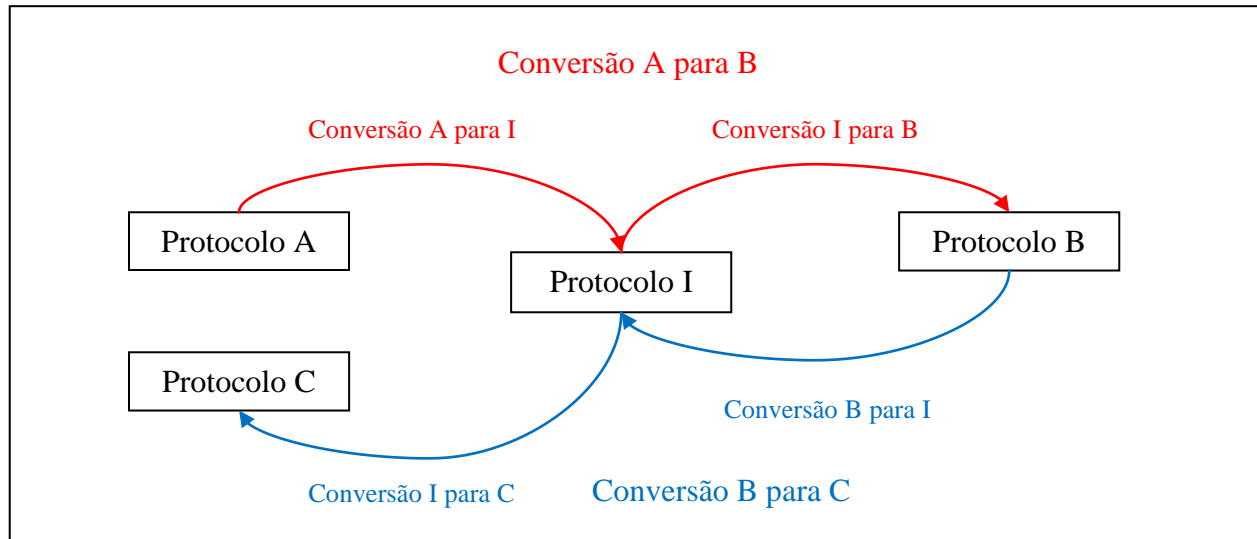


Figura 21: Diagrama do processo de conversão proposto

Ao longo deste projecto esta técnica será denominada de conversão indirecta e a conversão sem protocolo intermédio será denominada conversão directa.

Optou-se por utilizar a conversão indirecta a pensar na futura expansão do número de protocolos pois esta permite facilmente adicionar novos protocolo sem a necessidade de implementar algoritmos de conversão entre todos eles. O número de algoritmos que são necessários implementar para um dado número de protocolos podem ser calculados através das expressões (3.1) e (3.2). Os resultados até 10 protocolos podem ser visualizados na Figura 22.

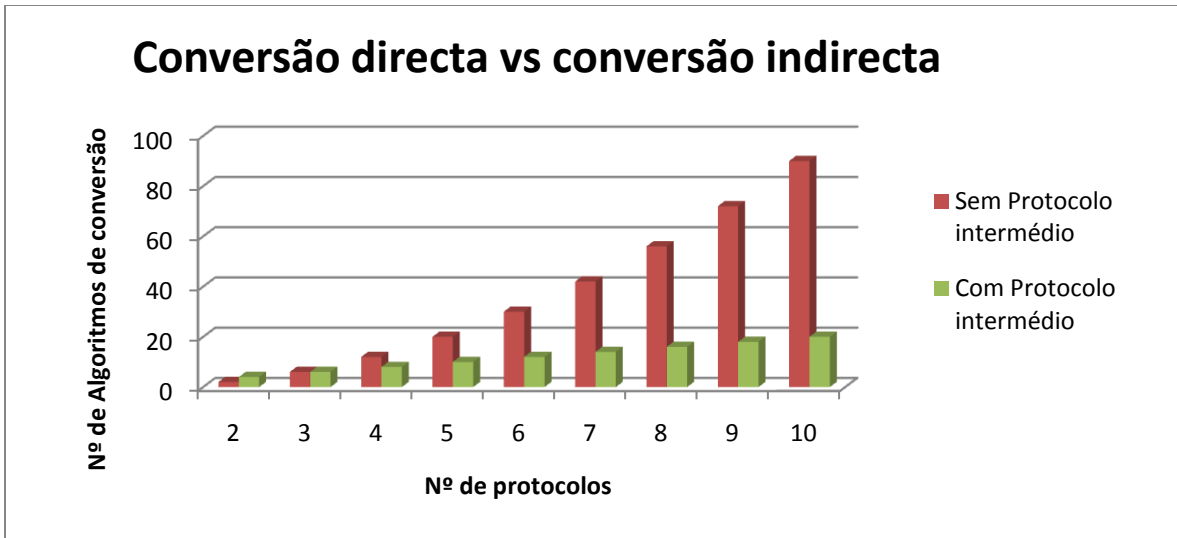


Figura 22: Comparação do número de algoritmos para conversão da forma directa e indirecta, calculados com as equações (3.1) e (3.2)

$$N^{\circ} \text{ Algoritmos} = P_2^n \text{ (Conversão directa)} \quad (3.1)$$

$$N^{\circ} \text{ Algoritmos} = 2 \times n^{\circ} \text{ Protocolos (Conversão indirecta)} \quad (3.2)$$

É importante realçar que o protocolo intermédio não é contabilizado no número de protocolos. O número de algoritmos da conversão directa,  $P_2^n$ , é o cálculo das permutações “ $P$ ” possíveis dos “ $n$ ” protocolos em grupos de 2 protocolos.

Analisando o gráfico da Figura 22, nota-se claramente que não é vantajosa a utilização de conversão indirecta entre dois ou três protocolos, mas para um número de protocolos superior a três notam-se claramente as vantagens de implementação, visto que o número de algoritmos que é necessário implementar é inferior ao da conversão directa.

A conversão directa tem a vantagem de ser mais rápida (uma só conversão) enquanto que a conversão indirecta será mais lenta, devido ao facto de serem realizadas duas conversões, mas generaliza a integração de novos protocolos. Nas aplicações de gestão de energia eléctrica o atraso da conversão indirecta não afecta o desempenho do sistema.

### 3.3.2 Protocolo intermédio

Para realizar a conversão indirecta de protocolos é necessário escolher um protocolo que irá agir como intermediário. Embora existam muitos protocolos, como o KNX, ZigBee, LonTalk, etc., que poderiam ser utilizados, optou-se por implementar um pequeno e simples protocolo com

ideias baseadas em alguns protocolos existentes, que será denominado PIntern (Protocolo Interno).

No PIntern definiu-se de uma forma básica as camadas de ligação de dados, rede e aplicação indicadas pelo modelo OSI. As outras camadas não são definidas de forma a manter o protocolo simples. A camada física utilizada é definida pelo protocolo UART, já existente para a comunicação entre equipamentos como sensores, microcontroladores, etc.

### 3.3.2.1 Camada de ligação de dados

A camada de ligação de dados do PIntern define a forma como é delimitado um pacote de dados. Também define a verificação da integridade do pacote através do cálculo do seu *checksum* (ver Figura 23).

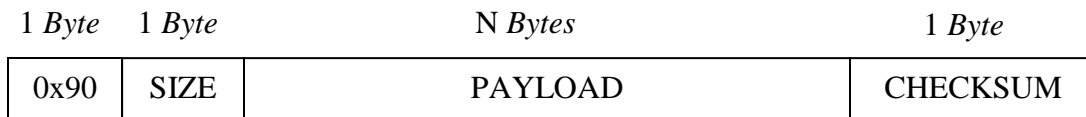


Figura 23: Formato da trama de ligação de dados do protocolo PIntern

O início do pacote é sempre representado pelo valor 0x90 seguido da dimensão do *payload* (SIZE). O pacote termina com o *byte* de *checksum* (semelhante à API dos módulos XBee) para verificação da integridade do pacote e detecção de erros. A dimensão corresponde ao número de *bytes* do *payload* sem incluir o *checksum*.

O *checksum* é calculado através do somatório dos valores hexadecimais dos *bytes* do *payload*, que depois é subtraído a 0xFF. Caso o resultado da subtração exceda 0xFF o *checksum* é considerado como sendo o valor do *byte* menos significativo.

### 3.3.2.2 Camada de rede

A camada de rede do PIntern especifica que a identificação dos dispositivos é efectuada por um endereço de 8 *bits* único para cada elemento, e com suporte para *unicast*, *multicast* e *broadcast* de pacotes. Também define um conjunto de endereços reservados.

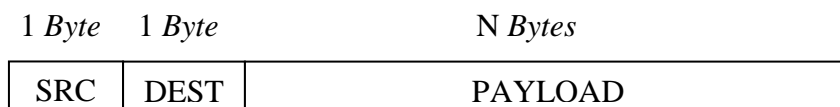


Figura 24: Formato da trama da camada de rede do protocolo PIntern

Dos 256 ( $2^8$  *bits*) endereços disponíveis 4 são reservados, ficando 252 endereços para dispositivos. Poderiam ter sido utilizados mais *bits* para o endereço, mas de forma a manter a simplicidade e para reduzir o cabeçalho, optou-se por limitar o número a 8 *bits*. Os quatro endereços reservados servem para configuração e encaminhamento dos pacotes (ver Tabela 1).

Cada pacote tem um endereço de origem (SRC), para permitir o envio de respostas. Também possui um endereço de destino (DEST), que permite realizar *unicast*, *broadcast* (semelhante ao protocolo IP) ou encaminhamento através da tabela de rotas (semelhante às *binding tables* do KNX).

**Tabela 1: Tabela dos endereços do protocolo PIntern**

Endereço	Utilização
0	Endereço da <i>Gateway</i> (para envio de comandos de configuração)
1	Endereço do dispositivo predefinido (normalmente atribuído à aplicação de configuração)
2-253	Endereço de dispositivos (disponíveis para atribuição)
254	Endereço de encaminhamento de pacotes pela tabela de rotas
255	Endereço de <i>broadcast</i> de pacotes

A tabela de rotas contém o registo dos endereços de origem e de destino a utilizar para o reencaminhamento das mensagens. Quando é recebida uma mensagem cujo endereço de destino (DEST) seja 254, são procuradas na tabela todas as rotas cuja origem é indicada por SRC, obtendo-se uma lista de destinos para onde será enviada a mensagem (*multicast*). Cada rota é composta por um endereço de origem e um de destino.

### 3.3.2.3 Camada de aplicação

A camada de aplicação do PIntern define os formatos dos dados dos pacotes, comandos e atributos suportados (ver Figura 25).

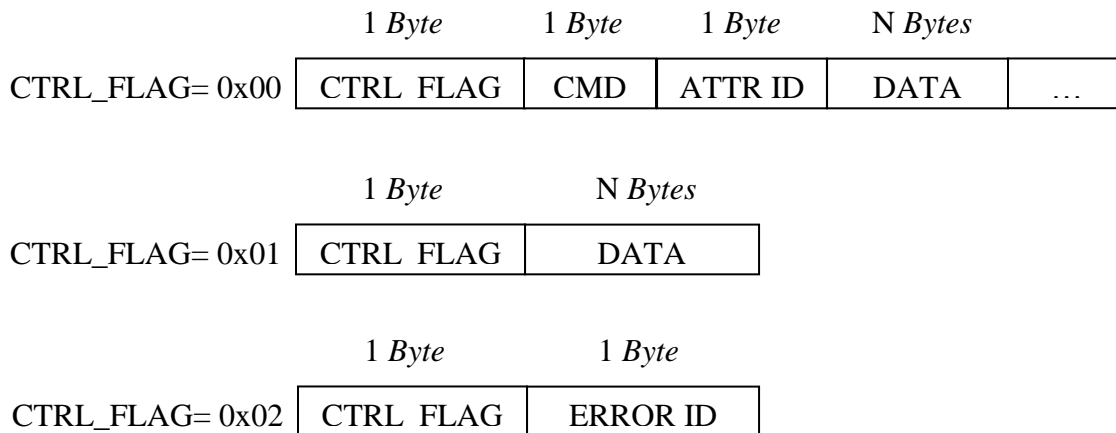


Figura 25: Formato da trama da camada de aplicação do protocolo PIntern

Na versão actual o *byte* CTRL\_FLAG indica somente o formato do segmento de dados, mas em versões futuras poderá ser utilizado para indicar a prioridade do pacote e/ou algumas opções para a conversão e encaminhamento.

Os tipos de formato suportados (CTRL\_FLAG) actualmente são:

- 0x00: Formato de comando, no qual é indicado o comando a executar e os parâmetros.
- 0x01: Formato de mensagem, no qual os dados enviados são vistos como uma mensagem e não são interpretados.
- 0x02: Formato de notificação, no qual é indicado o identificador do erro ocorrido.

A configuração da *gateway* é efectuada através do envio de pacotes de comando. Os vários comandos suportados são indicados na tabela 2 do anexo G.

O PIntern representa os dispositivos da rede sob a forma de objectos genéricos identificados pelo seu endereço. Cada objecto contém também o atributo “estado”, que indica o estado de funcionamento do dispositivo (em funcionamento, parado, ocupado). A escrita para este atributo é interpretada numa conversão como sendo um comando para mudar o estado de funcionamento do dispositivo.

A função de cada objecto é definida por um conjunto de atributos que podem ser modificados ou lidos com os comandos WRITE e READ, respectivamente (ver Figura 26).

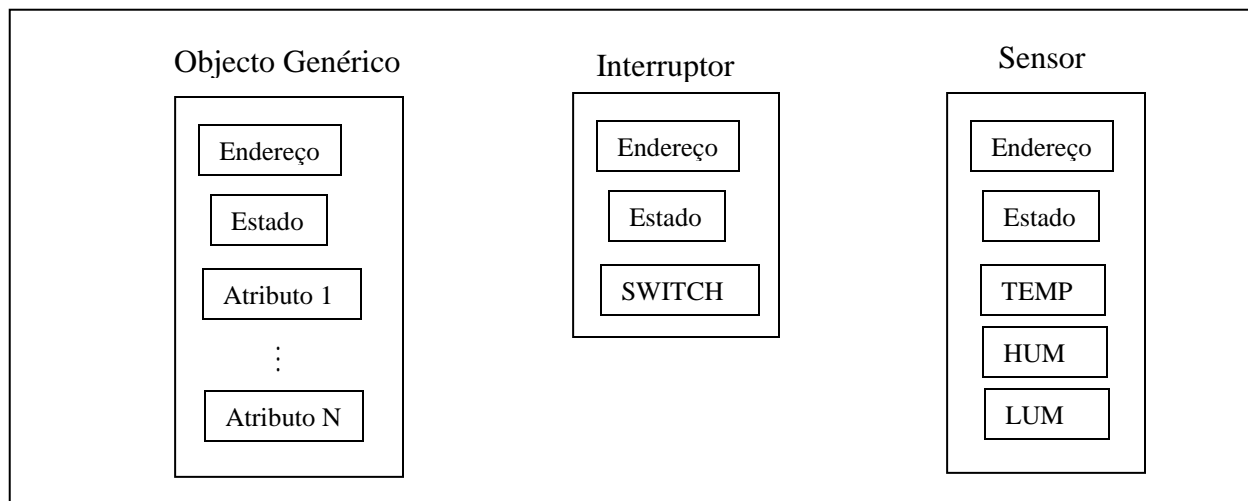


Figura 26: Representação de objectos no protocolo PIntern

Como exemplo, a representação de um interruptor é realizada por um objecto com um endereço único e um atributo do tipo SWITCH cujos valores são “1” ou “0”, representando os estados ON e OFF, respectivamente. Para modificar o estado do interruptor basta enviar o comando WRITE do atributo SWITCH com o valor correspondente ao estado pretendido.

Um pedido de leitura de qualquer atributo é respondido sobre a forma de um comando de escrita desse atributo. O dispositivo PIntern que efectua o pedido de leitura de um atributo a outro dispositivo PIntern, receberá do destinatário do pedido de leitura um comando de escrita do atributo. Isto permite que os dispositivos possam interagir entre si através da troca de comandos básicos de leitura e escrita.

Os atributos e comandos suportados nesta versão do protocolo PIntern são indicados em tabelas no Anexo G.

No caso em que não se pretenda efectuar a conversão da camada de aplicação os dados podem ser enviados sob a forma de mensagem. Nesta situação é necessário garantir que o dispositivo destinatário conseguirá interpretar os dados enviados. Por predefinição os pacotes cujos dados são enviados como mensagens devem ser encaminhados através da tabela de rotas sendo fácil garantir que o pacote chegará a um dispositivo que conseguirá interpretar os dados.

### 3.3.3 Arquitectura interna dos buffers

O *firmware* baseia-se na conversão e passagem de pacotes entre *buffers* até estes serem enviados ou descartados.

Optou-se pela utilização de *buffers* FIFO (*First In First Out*) circulares para armazenar os pacotes recebidos, por encaminhar ou por enviar (ver Figura 27).

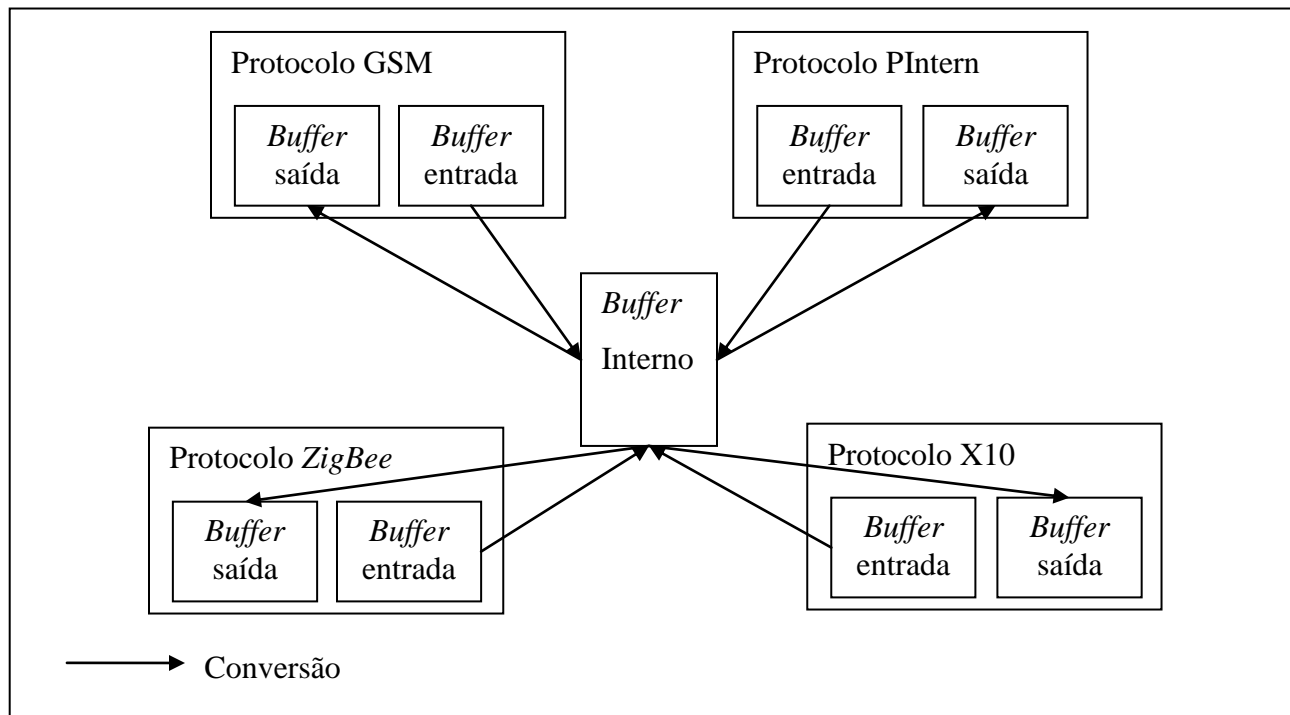


Figura 27: Diagrama de interligação dos *buffers*

Cada protocolo contém dois *buffers* FIFO circulares, um para os pacotes recebidos (entrada) que necessitam ser convertidos e outro para os pacotes convertidos que necessitam de ser enviados (saída). A estrutura dos pacotes contidos nestes *buffers* difere de protocolo para protocolo. Estes *buffers* são essenciais pois para realizar a conversão de pacotes é necessário recebê-los na sua totalidade, para só depois se realizar a conversão.

Para armazenar os pacotes convertidos ou por converter (pacotes PIntern) é utilizado um único *buffer* FIFO circular de maior dimensão.

A utilização dos *buffers* permite adaptar as taxas de envio e recepção de pacotes.

A implementação dos *buffers* podia ter sido efectuada utilizando listas ligadas, cujo tamanho é variável, tal como a sua ocupação de memória. Cada elemento da lista contém um ponteiro para o elemento seguinte e, opcionalmente, um ponteiro para o elemento anterior.

As listas ligadas apresentam a vantagem de utilizar eficientemente a memória, sendo esta alocada dinamicamente consoante é necessária. Outra vantagem é a facilidade de inserção,

remoção e reordenação da lista, pois é somente necessária a modificação dos ponteiros dos elementos sem a necessidade de ciclos para realizar a deslocação dos mesmos.

Por outro lado as desvantagens implicam uma implementação mais complexa, a necessidade de percorrer a lista até chegar ao elemento pretendido, cuidados adicionais ao alocar e desalocar dinamicamente a memória, a alocação de posições dispersas para cada elemento da lista (fragmentação), o que pode dificultar futuras alocações.

Assim, de forma a tornar o *firmware* mais simples, optou-se pela utilização de *arrays* de tamanho fixo, devido à facilidade de implementação, facilidade de acesso aos elementos por meio dos seus índices e melhor organização da memória, visto o espaço ser alocado pelo compilador em posições sequenciais de memória. A escolha da utilização de *arrays* também traz algumas desvantagens, por exemplo, em termos de utilização eficiente de memória, pois a memória alocada na compilação não pode ser libertada em *runtime* (durante a execução do programa). Existe também uma maior dificuldade de inserção, remoção e reordenação dos elementos da lista (especialmente dos elementos intermédios).

Cada *buffer* apresenta dois índices, um para inserir mensagens (IN) e outro para remover mensagens (OUT), como pode ser visto na Figura 28.

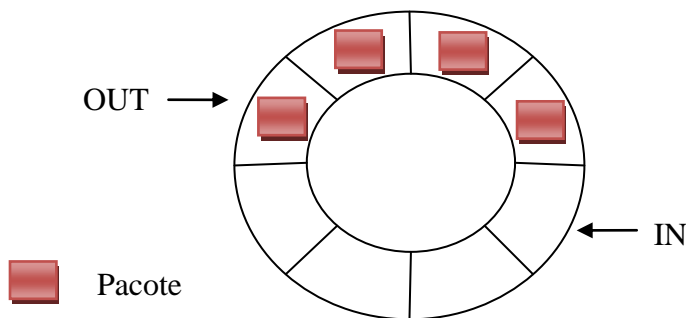


Figura 28: Representação do buffer FIFO circular

Estes índices só tomam o mesmo valor quando a lista se encontra vazia. Se um incremento num dos índices implicar que ambos se tornam iguais a lista é considerada cheia.

Quando um dos índices indica a posição final e necessita ser incrementado o resultado do incremento retorna o índice para a posição inicial do *buffer* através da aplicação de uma máscara (Figura 29).

```
ibuff.in=((ibuff.in+1)& (INT_BUF_MASK));
```

Figura 29: Segmento de código que efectua incremento de um índice do *buffer*

Ao longo do código do *firmware* o acesso aos elementos dos *buffers* é efectuado através de ponteiros. Esta solução foi adoptada porque permite reduzir a memória alocada pelo compilador, obtendo-se mais espaço para a alocação dinâmica, isto é, o espaço alocado para um ponteiro é menor que o espaço alocado para a estrutura contida no *buffer*. Claro que devido a esta escolha foi necessário ter um cuidado extra no desenvolvimento do código, devido à complexidade adicional de se trabalhar com ponteiros em C.

### 3.3.4 Protocolo XBee

O módulo XBee implementa a maior parte do protocolo *ZigBee* internamente, sendo somente necessário comunicar pela porta UART através de pacotes XBee API.

Nesta secção será descrito o procedimento de configuração da porta UART que irá receber e enviar os pacotes XBee API. Também será descrito o procedimento da conversão de pacotes XBee para PIntern e de PIntern para XBee.

O algoritmo de envio e recepção de pacotes XBee API é também descrito.

#### 3.3.4.1 Configuração inicial

Os *buffers* de entrada e saída de pacotes são inicializados como vazios.

É configurado o pino utilizado para o *reset* do módulo XBee como saída e inicializado com o nível lógico “1”.

Em seguida o *reset* do módulo é efectuado de forma a reinicializar a rede *ZigBee*. O *reset* é efectuado colocando o pino respectivo a “0” durante 1  $\mu$ s.

Uma das portas UART do microcontrolador é configurada para receber e enviar dados com um *baudrate* de 9600 com 8 *bits* de dados, 1 *bit* de paragem e sem *bit* de paridade. A interrupção de recepção completa (RXC) é configurada com um baixo nível de prioridade (Figura 30).

```

void config_ZIGBEE_USART(void)
{
//configurar os pins RX e TX como entrada e saída respectivamente
ZIGBEE_PORT.DIRSET = ZIGBEE_TX; //colocar pin TXD0 como output
ZIGBEE_PORT.DIRCLR = ZIGBEE_RX; //colocar pin RXD0 como input
// formato de dados da USART, 8 Data bits, No Parity, 1 Stop bit
ZIGBEE_USART.CTRLA = (uint8_t) USART_CHSIZE_8BIT_gc | USART_PMODE_DISABLED_gc | false;
//configuração do baudrate
ZIGBEE_USART.BAUDCTRLA = ZIGBEE_BAUD_SELECT;
//activar a interrupção RXC
ZIGBEE_USART.CTRLA = (ZIGBEE_USART.CTRLA & ~USART_RXCINTLVL_gm) | USART_RXCINTLVL_LO_gc;
}

```

Figura 30: Segmento de código da função de configuração de uma porta UART do microcontrolador

### 3.3.4.2 Recepção de pacote XBee

Com a porta UART configurada a recepção de um *byte* provoca uma interrupção cujo tratamento é realizado pela função “ISR( ZIGBEE\_USART\_RXC\_vect)”. É de realçar que enquanto não é recebido um *byte*, o CPU continua a executar o programa normalmente, sendo este interrompido brevemente para tratamento da interrupção.

De uma forma simplificada, o algoritmo quando recebe o início do pacote XBee (0x7E) dá indicação da recepção de uma nova mensagem. Depois de receber os três primeiros *bytes* (1 *byte* de início e 2 *bytes* de tamanho do pacote) é utilizado o tamanho do pacote para alocar memória para armazenamento do mesmo no *buffer*. É importante lembrar que o tamanho do pacote não inclui o *byte* do *checksum*, sendo necessário acrescentá-lo para alocação da memória.

Nesta altura a interrupção RXC da UART é desactivada impedindo que o algoritmo volte a ser executado até ser terminada a recepção total do pacote.

Depois de alocada a memória é configurado o DMA para recepção dos restantes *bytes* do pacote. O DMA realiza a leitura do registo DATA da UART e coloca o valor na posição de memória alocada sempre que a *flag* RXC da UART é activada, daí a necessidade de desactivar a interrupção RXC para não ser executada a função de tratamento da interrupção. Isto ocorre até ser recebido o número de *bytes* alocados (tamanho do pacote+*checksum*).

Para cada *byte* recebido a posição de memória de destino é incrementada automaticamente pelo módulo DMA, e desta forma a memória alocada previamente é preenchida

com os *bytes* do pacote. O código para configuração do DMA foi baseado em segmentos de código de tutoriais da Atmel [38].

Ao terminar a recepção o DMA provoca uma interrupção. No tratamento dessa interrupção é verificado o *checksum* do pacote recebido. Caso se confirme o *checksum*, o pacote é adicionado ao *buffer* de entrada e caso não se confirme, o pacote é descartado.

Durante todo o processo de recepção utilizando o DMA, o CPU encontra-se livre para executar o programa, com a exceção de momentos em que é necessário aceder á mesma memória que o DMA está a aceder (normalmente a SRAM (*Static Random Access Memory*)). Nesta situação do CPU terá de aguardar até o barramento ser libertado. O acesso a outra memória (Exemplo: memória flash que contém o programa) pode ser realizado através de um segundo barramento.

### 3.3.4.3 Conversão de XBee para PIntern

A secção principal do código, que será apresentada mais adiante, chama a função para converter o pacote XBee para um pacote PIntern quando o *buffer* de entrada correspondente não estiver vazio.

A conversão inicia-se com a obtenção do ponteiro para o pacote que se encontra no *buffer* de entrada XBee, indicado pelo índice OUT (Figura 31).

```
zigbee_msg_t * m=(zigbee_msg_t *) &zbuff_in.buff[zbuff_in.out];
```

Figura 31: Segmento de código para obtenção de um ponteiro para um pacote num buffer

Logo de seguida é verificado o primeiro *byte* do pacote recebido, que corresponde ao ID da trama API recebida. De notar que os *bytes* de inicio de pacote e dimensão já foram descartados por já não terem utilidade nesta fase.

Caso o pacote recebido tenha o API\_ID 0x90 este indica a recepção de uma mensagem proveniente de um nó da rede XBee. Nesse caso, é retirado do pacote o endereço XBee de 64bits que é utilizado para procurar na lista de dispositivos interna da *gateway* (Figura 32).

```

for(j=0;j<XBee_ADDR_SIZE;j++)
{
    tempaddr64[j]=m->msg[j+RECV_MSG_ADDR_POS];
}
//procurar endereço da origem na tabela e obter o endereço interno
gen_dev_t * d=find_dev(tempaddr64,XBee_PROT);

```

**Figura 32: Segmento de código que obtém o endereço de 64 bits do pacote XBee**

Durante a procura o endereço de 64 *bits* obtido do pacote é comparado com o endereço de 64 *bits* dos elementos da lista e caso este seja encontrado é retornado o ponteiro para o mesmo. Caso não seja encontrado é retornado *null*. Através desse ponteiro é possível obter o endereço interno do dispositivo de onde originou a mensagem, e que é utilizado para construir o pacote PIntern. Caso o dispositivo seja encontrado na lista, é analisado o identificador da camada de aplicação do mesmo de forma a descobrir qual a função de conversão da camada de aplicação que deve ser utilizada (Figura 33).

```

switch(d->app_layer)
{
    case XBee_FLORENS_APP_LAYER:
        //preencher com dados da camada de rede
        im->dest=0xFE;
        //chama função para tratar da camada de aplicação
        state=translate_fromflorsens(m,im);
        break;
    case XBee_INT_APP_LAYER:
        //preencher com dados da camada de rede
        im->dest=0xFE;
        //chama função para tratar da camada de aplicação
        state=translate_fromintzbaplayer(m,im);
}

```

**Figura 33: segmento de código que determina camada de aplicação e chama a função apropriada**

São suportadas duas camadas de aplicação: a camada de aplicação definida no projecto de mestrado do colega Tiago Braga, denominada neste projecto como florsens [39], e a camada de aplicação do protocolo PIntern.

Se a camada de aplicação for a florsens, o formato do pacote é indicado pelo primeiro *byte*. De momento só existe suporte para o formato com o ID=0xD0 (Figura 34).

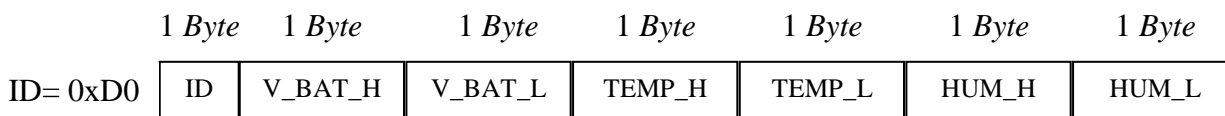


Figura 34: Formato da trama da camada de aplicação florsens

Os valores das medições são associados a atributos de tipo igual ou semelhante. Para este formato de pacote a associação do valor da temperatura, humidade e carga da bateria contido no pacote XBee é efectuada com os atributos do tipo TEMP, HUM e BAT\_LVL, respectivamente.

Por predefinição os valores dos atributos são constituídos por 2 *bytes*, à excepção de alguns cuja dimensão é indicada por um *byte* logo a seguir ao identificador do atributo.

No caso da camada de aplicação ser a do PIntern não é necessário fazer conversão da camada de aplicação pois os dados já apresentam o formato pretendido, sendo necessário simplesmente copiar os dados para o pacote PIntern. Esta conversão produz um pacote PIntern de dimensão mais reduzida, mas com as informações essenciais para o encaminhamento e posterior conversão.

O pacote PIntern depois de construído é adicionado ao *buffer* interno na posição indicada pelo índice IN.

#### 3.3.4.4 Conversão de PIntern para XBee

O procedimento de conversão de PIntern para XBee inicia-se com a obtenção do ponteiro para o pacote do *buffer* interno indicado pelo índice OUT.

Utilizando o endereço de destino procurado na lista de dispositivo é determinado o endereço real do dispositivo, protocolo e camada de aplicação, de forma a serem invocadas as funções de conversão apropriadas. O mesmo processo aplica-se a cada protocolo implementado.

Neste caso o destinatário pertence ao protocolo XBee e a função “translate\_inttoxb” é executada.

A conversão começa com a obtenção de um ponteiro para um pacote vazio do *buffer* de saída XBee indicado pelo índice IN. Depois é determinada qual a camada de aplicação que o dispositivo destinatário utiliza (através da informação indicada na lista de dispositivos).

Nesta fase só existe suporte para conversão para dispositivos com suporte à camada de aplicação PIntern, podendo em futuras versões suportar mais camadas de aplicação.

Um pacote XBee é preenchido para formar uma trama API cujo ID é 0x10 e o formato é indicado no Anexo D.

O `FRAME_ID` do pacote é colocado a 0 para que não seja enviada resposta ao envio do pacote devido ao facto de ainda não ser suportada a recepção de respostas. De seguida é preenchido o endereço de 64 *bits*, que é obtido da lista de dispositivos convertendo assim o endereço interno para o endereço real do dispositivo.

O endereço de 16 *bits* é indicado como 0xFFFE pois o seu valor é desconhecido.

Finalmente, os dados do pacote XBee são inseridos pela função “`translate_tointzbaplayer`”, que limita-se a copiar os dados do pacote PIntern para o pacote XBee. Depois de preenchido o pacote é calculado o *checksum* e o pacote é colocado no *buffer* de saída na posição indicada pelo índice IN, para ser enviado para a rede XBee (Figura 35).

```
bool translate_tointzbaplayer(zigbee_msg_t * m,int_msg_t * im)
{
    //indicar destino
    m->msg[SEND_MSG_DATA_POS]=im->src;
    m->msg[SEND_MSG_DATA_POS+1]=im->dest;
    //indicar as flags de controlo
    m->msg[SEND_MSG_DATA_POS+2]=im->ctrl_flags;
    //dados
    uint8_t j;
    for(j=0; j<im->data_size;j++)
    {
        m->msg[SEND_MSG_DATA_POS+3+j]=im->data[j];
    }
    calc_Zigbee_checksum(m);
    return true;
}
```

Figura 35: Segmento de código para preencher o pacote XBee com os dados do pacote PIntern

### 3.3.4.5 Envio de pacote XBee

O envio do pacote XBee torna-se mais simples devido à utilização do DMA do microcontrolador. Começa-se com a obtenção do ponteiro do pacote a enviar do *buffer* de saída.

Utilizando a dimensão e apontador para os dados do pacote é configurado o DMA para o envio *byte a byte* sempre que a *flag DRE (Data Register Empty)* da UART que comunica com o XBee seja activada. A transmissão termina quando o número de *bytes* enviado é igual à dimensão do pacote.

A cada envio de um *byte* o ponteiro para a origem dos dados é incrementado apontando para o próximo *byte* a ser enviado. O destino dos dados mantém-se fixo no registo DATA da porta UART. O código de configuração do DMA foi baseado em tutoriais da Atmel [38].

Quando terminado o envio do pacote XBee este é retirado do *buffer* de saída.

### 3.3.5 Protocolo PIntern

O protocolo PIntern é utilizado principalmente para a conversão entre os vários protocolos, mas pode ser utilizado para receber e enviar pacotes tal como os outros protocolos.

#### 3.3.5.1 Recepção e envio de pacotes PIntern

A recepção e envio dos pacotes PIntern é efectuada de forma muito semelhante à recepção e envio de pacotes XBee através da utilização do DMA.

#### 3.3.5.2 Conversão

Os dados dos pacotes já se encontram no formato pretendido não sendo necessário realizar uma conversão mas sim uma cópia dos dados (*payload*) do pacote recebido, que se encontram encapsulados dentro da trama de ligação de dados, para um pacote interno. Para o envio, o procedimento encapsula o pacote interno dentro de uma trama de ligação de dados.

### 3.3.6 Protocolo X10

Ao contrário dos outros protocolos que utilizam os periféricos UART para comunicar com o módulo externo, a técnica de comunicação com o módulo X10 implica a utilização dos pinos I/O do microcontrolador.

O Módulo X10 implementa a camada física e de ligação de dados do protocolo X10.

#### 3.3.6.1 Configuração inicial

A comunicar com o módulo o MCU utiliza 3 pinos (2 de entrada e 1 de saída) que são conectados ao conversor de níveis. De forma a sincronizar o envio de *bits* com a travessia pelo “zero” da rede eléctrica é monitorizada a saída zero *crossing* do módulo X10, através do conversor de níveis. Refira-se que é activada a interrupção *pin change* para detectar as transições do sinal (Figura 36).

A transmissão de um impulso com 1 ms de duração é configurada num temporizador para que o *overflow* ocorra dentro desse tempo. A interrupção de *overflow* foi configurada com nível de prioridade baixo (Figura 36).

```
void config_x10(void)
{
    //configurar os pins a utilizar para o x10.
    X10_PORT.DIRCLR = X10_ZEROCROSS; //colocar o zerocross como input
    X10_PORT.DIRCLR = X10_RX; //colocar o pin RX como input;
    X10_PORT.DIRSET = X10_TX; //colocar o pin TX como output;
    X10_PORT.OUTCLR = X10_TX;
    //configurar interrupção de pin change
    X10_PORT.INT0MASK = (X10_PORT.INT0MASK & 0x00) | X10_ZEROCROSS;
    X10_PORT.PIN0CTRL = (X10_PORT.PIN0CTRL & ~PORT_ISC_gm) | PORT_ISC_BOTHEDGES_gc;
    //configurar temporizador com interrupção
    //calculo do valor de PER
    //
    // PER = (tempo (ms) * SYSCLK (Hz))/PRESCALER = SYSCLK (Hz)/(PRESCALER * frecuencia (Hz))
    //
    //definição do registo PER (Periodo) do timer para overflow a 1ms (1KHz) e prescaler=1
    X10TIMER.PER= F_CPU/1000;
    //activação da interrupção de overflow
    X10TIMER.INTCTRLA = ( X10TIMER.INTCTRLA & ~TC0_OVFINTLVL_gm ) | TC_OVFINTLVL_LO_gc;
}
```

Figura 36: Segmento de código que configura os pinos de entrada e saída e temporizador

### 3.3.6.2 Recepção de pacotes X10

De forma a reduzir a complexidade e melhorar a eficiência do algoritmo de envio optou-se por não implementar na *gateway* o suporte à recepção de pacotes X10 nesta versão do *firmware* do protótipo. A estrutura do código encontra-se preparada para a recepção de pacotes X10 sendo somente necessário acrescentar o código do algoritmo. Note-se que nos testes efectuados não são utilizados equipamentos X10 que enviem comandos para a *gateway*, sendo por isso a implementação da recepção desnecessária e contra produtiva.

### 3.3.6.3 Conversão de X10 para PIntern

A recepção de pacotes X10 não foi implementada logo não há necessidade de converter pacotes X10 para pacotes PIntern. Isto permite reduzir a complexidade do *firmware* da *gateway*, reduzindo a dimensão do código e melhorando a utilização de recursos do microcontrolador.

A estrutura do código já se encontra preparada para efectuar a conversão, sendo no entanto necessário implementar o código do algoritmo de conversão.

### 3.3.6.4 Conversão de PIntern para X10

Da mesma forma que nos outros protocolos a conversão começa pela aquisição do apontador para o pacote do PIntern.

Da informação dos dispositivos, presente na tabela de dispositivos, é obtido o *Home code* e o *Unit code* do dispositivo destinatário através do seu endereço interno. Os códigos são complementados na preparação para a inserção dos mesmos no pacote X10 (Figura 37).

```
uint16_t hcode_send=complementedcode(d->addrx10[HOME_ADDR_POS], HOME_ADDR_SIZE);  
//gerar unitcode  
uint16_t ucode_send=complementedcode(d->addrx10[UNIT_ADDR_POS], UNIT_ADDR_SIZE);
```

Figura 37: Segmento de código que chama as funções para complementar o códigos X10

Cada pacote X10 é constituído por duas secções representadas por variáveis de 64 *bits* nas quais serão construídas as sequências de *bits* a enviar pelo pino de transmissão. Na realidade, somente 44 dos 64 *bits* serão utilizados.

Uma das secções (*unit*) irá conter a sequencia de *bits* onde é transmitido o *home code* e *unit code* para indicação de qual o dispositivo que deverá executar o comando que será enviado em seguida. A sequência de *bits* inclui o *starter*, *home code* complementado e *unit code* complementado e a repetição da mesma (44 *bits* no total, ver Figura 5 da página 12). A outra secção (*cmd*) irá conter uma sequência de *bits* semelhante, onde o *unit code* é substituído pelo código do comando a executar.

Depois de construída a sequência de *bits* que determina qual o dispositivo a controlar é chamada a função que interpreta qual o comando que será realizado. De momento só é suportado o comando WRITE no atributo SWITCH de forma a manter os código simples e rápido, mas o *firmware* encontra-se preparado para suportar mais comandos e atributos.

Quando o comando da camada de aplicação do protocolo PIntern indicar a escrita no atributo SWITCH é verificado o valor do atributo de forma a determinar qual o comando X10 a

enviar. Caso o valor do atributo SWITCH seja zero é enviado o comando *OFF*, se o valor for diferente de zero é enviado o comando *ON*.

Depois de terminada a construção das sequências de *bits* a enviar o pacote X10 é adicionado ao *buffer* de saída para posterior envio.

### 3.3.6.5 Envio de pacotes X10

Para o envio do pacote é verificado em primeiro lugar se actualmente não está a ser enviado um. Caso não esteja a ser, é activada a interrupção *pin change* do pino *zero crossing*.

A colocação do *bit* no pino TX é efectuada pela função de tratamento da interrupção que é activada sempre que ocorre uma transição do sinal *zero crossing*.

A função de tratamento de interrupção começa por verificar que segmento da sequência de *bits* está a ser enviada (*unit*, *gap* ou *cmd*).

O valor do *bit* é colocado no pino através do SET ou CLR dependendo no valor resultante da aplicação da máscara na variável que contém a sequência de *bits*. Caso resulte no valor 0 o pino é colocado a “0” e caso seja diferente de zero o pino é colocado a “1” (Figura 38).

```
if((temp_tx->unit & TX_BIT_MASK)==0)
{
    X10_PORT.OUTCLR = X10_TX;
}
else
{
    X10_PORT.OUTSET = X10_TX;
}
```

Figura 38: Segmento de código que coloca no pino de saída a sequência de *bits* do pacote X10

Entre a sequência de endereçamento e a sequência de comando (*cmd*) é introduzido um intervalo de 6 transições pelo zero da rede eléctrica onde não são enviados dados (pino colocado a “0”).

Como o envio *bit a bit* é realizado através de uma função de tratamento de uma interrupção, isto implica que o CPU se encontra a correr o programa, até ser necessário parar a execução para um tratamento rápido da interrupção e depois retornar à execução. Este processo torna a execução do código mais eficiente e rápida.

Quando terminado o processo de envio o pacote X10 é retirado do *buffer* de saída esse pacote e é indicada a disponibilidade para envio do próximo pacote.

### 3.3.7 Protocolo GSM

A comunicação com o módulo GSM externo é efectuada através de uma porta UART.

O interface com o módulo é realizado através de comandos AT (ATtention), o que introduz complexidade adicional pois tratam-se de sequências de *bytes* codificados em ASCII. A comunicação através de comandos AT implica também a necessidade de aguardar respostas, o que dificulta a implementação de um algoritmo que não bloqueie o restante programa à custa da espera por uma resposta. Refira-se que algumas respostas apresentam elevada dimensão e uma estrutura complexa.

#### 3.3.7.1 Configuração inicial

O módulo GSM externo foi previamente configurado com as seguintes opções, que foram guardadas na memória não volátil e são carregadas sempre que o módulo se inicia:

- É desactivado o eco dos comandos através do envio do comando “ATE0”;
- É activada a indicação de nova mensagem com o comando “AT+CNMI=,1”;
- É efectuada a configuração da apresentação de mensagens SMS para modo texto com o comando “AT+CMGF=1”.

A configuração previamente efectuada ao módulo GSM poderá, opcionalmente, ser alterada através do próprio microcontrolador depois de configurada a porta UART.

No microcontrolador é configurada uma porta UART de modo idêntico ao que é efectuada nos protocolos PIntern e XBee. Não é utilizado o DMA para a recepção dos pacotes GSM devido ao desconhecimento da dimensão dos mesmos e à necessidade da leitura de respostas para o envio dos comandos AT.

É também utilizado um pino do microcontrolador para enviar um sinal ao módulo GSM para activá-lo. Este pino é configurado como saída e inicializado com o nível lógico “1”.

O sinal de activação permanece a “0” um tempo mínimo de 100 ms e depois é colocado a “1” para efectuar a activação do módulo GSM. Sem a activação do módulo GSM não são recebidos nem enviados dados através dessa porta UART.

Logo de seguida é enviado o comando “AT^SMSO” que faz com que o módulo se desligue. Isto permite garantir que as configurações iniciais do módulo são carregadas novamente quando o módulo é iniciado. Em seguida é enviado novamente o sinal de activação para iniciar o módulo.

Depois de activado o módulo GSM é necessário enviar o comando “AT+CPIN” de forma a introduzir o PIN (*Personal Identification Number*) do cartão SIM (*Subscriber Identify Module*). Este comando devolverá uma resposta “OK” dentro de 20 s ou menos se o PIN for aceite (tempo para carregar a informação do cartão). Caso contrário é indicado um erro ou não há resposta se o módulo não foi iniciado.

De modo a impedir o bloqueio do programa da *gateway* à espera da resposta da introdução do PIN, o código de inicialização aguarda aproximadamente 30 s (*timeout*). Se até lá não for recebida resposta é enviada uma mensagem ao gestor do módulo (endereço 1) a indicar que o módulo não foi inicializado e prossegue a inicialização da *gateway*.

Opcionalmente, os passos anteriores de introdução do PIN podem ser ignorados se for efectuada anteriormente a desactivação do código PIN do cartão.

Para voltar a efectuar o procedimento de configuração do módulo GSM será necessário fazer *reset* ao microcontrolador.

### 3.3.7.2 *Recepção de pacotes GSM*

A recepção de pacotes GSM é efectuada também através da UART do microcontrolador. A recepção é realizada através do envio de comandos e recepção de notificações AT e de respostas do módulo GSM, através da interrupção RXC da UART.

A função de tratamento da interrupção RXC lê o valor do *byte* recebido para um *buffer* temporário. Os pacotes transmitidos pelo módulo GSM são sequências de caracteres que formam mensagens de notificação de eventos e resposta a comandos. Estas mensagens normalmente iniciam-se pelo símbolo ASCII “+” e terminam com o valor 0x0D (correspondente a “\r”).

Assim, o primeiro *byte* recebido é comparado com o símbolo ASCII “+” e em caso de igualdade é indicada a recepção de um novo pacote. Também é verificado se o *buffer* de entrada do protocolo GSM não se encontra cheio. Caso este esteja cheio a mensagem é desprezada.

Além da indicação de uma nova mensagem é obtido um apontador para uma nova mensagem no *buffer* de entrada. Para armazenar os dados do novo pacote é alocado um espaço para informação do comando ou notificação (Exemplo: índice da mensagem ou número da origem da mensagem) e outro espaço para o conteúdo dos dados que possam vir associados (Exemplo: mensagem de texto da SMS). A dimensão do espaço alocado é predefinida como 100 *bytes* tanto para mensagem como para o comando. Este valor poderá ser alterado, mas deverá ter

em conta que as respostas a comandos AT apresentam dimensão variável que pode atingir dimensões consideráveis.

Enquanto estiver a ser recebido um novo pacote os *bytes* recebidos são colocados na memória alocada. Para cada *byte* recebido é efectuada uma comparação com o valor 0x0D que indica o final do segmento do pacote.

Depois de terminado o primeiro segmento do pacote é verificado se este possui um segundo segmento (segmento de dados) através da comparação dos *bytes* iniciais da sequência de comando com a sequência “CMGR”, que indica que é uma resposta a um pedido de leitura de uma mensagem recebida e possui segmento de dados. Em caso de igualdade, a recepção continua e os novos *bytes* são colocados no espaço de mensagem do novo pacote até se obter novamente o valor 0x0D.

Quando termina a recepção do pacote este é adicionado ao *buffer* de entrada do protocolo GSM.

No caso de se aguardar uma resposta é verificado se no *buffer* temporário existe a sequência de *bytes* esperada ou a sequência “ERROR”. Caso se verifique qualquer uma das sequências no *buffer* temporário é indicada a recepção de uma resposta. Em caso da recepção da sequência de erro é indicada a recepção de uma resposta juntamente com a ocorrência de um erro.

### 3.3.7.3 Conversão de GSM para PIntern

O algoritmo de conversão do protocolo GSM para o protocolo interno não se limita somente a converter um pacote GSM para um pacote interno. Este também realiza a interpretação das respostas e notificações enviadas pelo módulo e constrói pacotes para o envio de comandos AT para o módulo GSM.

Da mesma forma que com os outros algoritmos de conversão, este inicia-se com a obtenção de um pacote contido no *buffer* de entrada.

Sempre que é recebida uma mensagem por SMS no módulo GSM, é enviada para o microcontrolador uma notificação “CMTI” onde é indicado o índice da mensagem recebida. Para se obter a mensagem é necessário o envio do comando “AT+CMGR” com a indicação do índice da mensagem a ler.

O algoritmo de conversão ao interpretar um pacote recebido cujo segmento de comandos inicia-se pela sequência “CMTI”, inicia a construção de um pacote GSM para envio do comando “AT+CMGR”. No comando é adicionado o índice obtido da notificação “CMTI” e a terminação 0x0D no final. O novo pacote construído é adicionado ao *buffer* de saída GSM para ser enviado para o módulo GSM. Da interpretação deste pacote GSM não é construído qualquer pacote PIntern.

A resposta ao comando “AT+CMGR”, quando recebida, é identificada pela sequência inicial “CMGR”. Quando o pacote a converter se inicia com esta sequência o primeiro passo é obter o número de onde originou a mensagem.

Como o número de origem da mensagem, com origem em Portugal, se inicia com a sequência “+351”, o número de origem da mensagem é obtido através de um ponteiro para o símbolo “+”. Como se pretende desprezar a sequência inicial (“+351”) são adicionadas 4 posições ao endereço indicado pelo ponteiro fazendo com que esta aponte para o início do número GSM. O número é copiado para um *array* temporário que é utilizado para procurar o dispositivo na tabela interna da *gateway*.

De modo a reduzir o tempo de tratamento do pacote GSM, muita da informação contida dentro do segmento de comando do mesmo é considerada de pouca importância para a aplicação actual, mas encontra-se disponível no pacote para acesso em futuras versões do *firmware*.

Se o dispositivo for encontrado na tabela de dispositivos da *gateway*, inicia-se a construção do pacote PIntern. Por predefinição o conteúdo da mensagem SMS é convertido para um pacote PIntern com a *CTRL\_FLAG*=0x01 (indicando que o conteúdo é uma mensagem) e o destinatário é indicado pela tabela de rotas (endereço 254).

Outra forma suportada é a inclusão das sequências “CMD:” e “MSG:” no texto da mensagem para indicar que o texto contém o endereço de destino e outras informações a considerar na conversão (Figura 39).

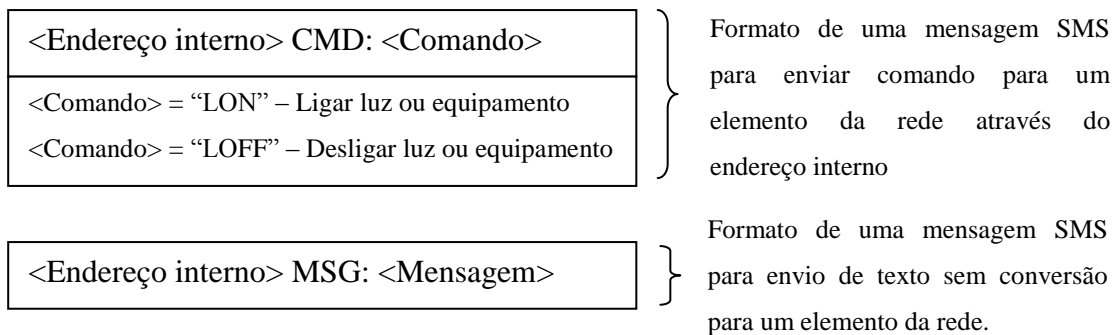


Figura 39: Formato das mensagens SMS para interagir com a rede

Assim, segundo o formato indicado na Figura 39, o envio de um comando para ligar uma luz num dispositivo de rede, cujo endereço é 224, apresenta a seguinte forma: “224 CMD: LON”.

O endereço de destino é o primeiro elemento a ser indicado. Este endereço é representado por caracteres ASCII mas precisa ser convertido para um valor do tipo inteiro.

A conversão é efectuada caractere a caractere sendo obtido o respectivo número inteiro pela subtracção do valor hexadecimal 0x30. Cada valor inteiro é multiplicado pela ordem de grandeza sendo o algarismo das centenas multiplicado por 100, das dezenas por 10 e unidades por 1. A soma destes resultados dá o valor pretendido numa variável do tipo inteiro. Este valor é utilizado como endereço de destino do pacote PIntern a ser construído.

Se for encontrada a sequência “CMD:” na mensagem, o pacote PIntern é construído com a CTRL\_FLAG=0x00 indicando que é um pacote com um comando.

Se a mensagem contiver a sequência “LON” o novo pacote PIntern contém o comando de WRITE do valor 1 no atributo SWITCH. Se a sequência for “LOFF” o pacote contém o comando de WRITE do valor 0. Com a expansão da rede e dos tipos de equipamentos introduzidos, a gateway apresenta a capacidade de expandir facilmente o conjunto de comandos suportados.

Se a sequência presente na mensagem for “MSG:”, o pacote PIntern é construído com a CTRL\_FLAG=0x01, indicando que o conteúdo é uma mensagem. O texto a enviar é simplesmente copiado para o segmento de dados do pacote PIntern, sem ser realizada qualquer conversão ou interpretação.

Depois de construído o pacote PIntern, este é adicionado ao *buffer* interno para posterior encaminhamento e conversão.

#### 3.3.7.4 *Conversão de PIntern para GSM*

Como os receptores GSM utilizados são telemóveis que não fazem a interpretação das mensagens SMS ou a execução de comandos, optou-se por implementar somente o suporte para conversão de pacotes PIntern que contenham mensagens (*CTRL\_FLAG=0x01*), sendo os restantes descartados.

O apontador para o pacote PIntern é adquirido a partir do *buffer* interno, juntamente com a informação do dispositivo de destino.

Com essa informação é construído um pacote GSM para o envio do comando de envio de mensagem SMS. Começando pelo preenchimento do segmento de comando do pacote GSM com o comando “AT+CMGS”, que recebe como argumento o número GSM do destinatário da mensagem. No final do comando é adicionado o terminador 0x0D.

No segmento de mensagem do pacote GSM é colocada a cópia da mensagem contida no pacote PIntern. A mensagem é terminada com o valor 0x1A.

O envio deste comando requer uma resposta por parte do módulo, sendo assim incluído no pacote um apontador para a variável que contém a sequência de resposta esperada. O pacote GSM depois é colocado no *buffer* de saída GSM.

#### 3.3.7.5 *Envio de pacotes GSM*

O envio do pacote é efectuado através do tratamento da interrupção DRE da porta UART. Antes disso o envio da mensagem é preparado através da obtenção do pacote GSM a enviar e da inicialização das variáveis que regem o processo de envio. O envio é iniciado com a activação da interrupção DRE da UART. Caso já esteja a ser enviado um pacote os passos descritos anteriormente não são realizados.

Depois de preparado o envio e activada a interrupção, a função de tratamento é executada sempre que o registo DATA da porta UART estiver vazio (*byte* enviado).

Esta função começa por enviar o comando AT presente no segmento de comando do pacote. O comando é enviado pela porta UART *byte a byte* a cada execução da função.

Quando terminado o envio do comando é verificado se deve esperar alguma resposta. No caso de esperar resposta é desactivada a interrupção DRE da porta UART. Esta interrupção só volta a ser activada depois de recebida a resposta pretendida ou uma indicação de erro.

O passo seguinte é o envio do segmento de mensagem do pacote GSM, caso este exista, utilizando o mesmo procedimento que no envio do segmento de comando.

Terminado o envio do pacote GSM é indicado o final do envio para que o pacote possa ser removido do *buffer* de entrada e desactivada a interrupção DRE da porta UART.

### 3.3.8 Encaminhamento de pacotes

O encaminhamento é efectuado através do despacho dos pacotes contidos nos vários *buffers* da *gateway* (Figura 40).

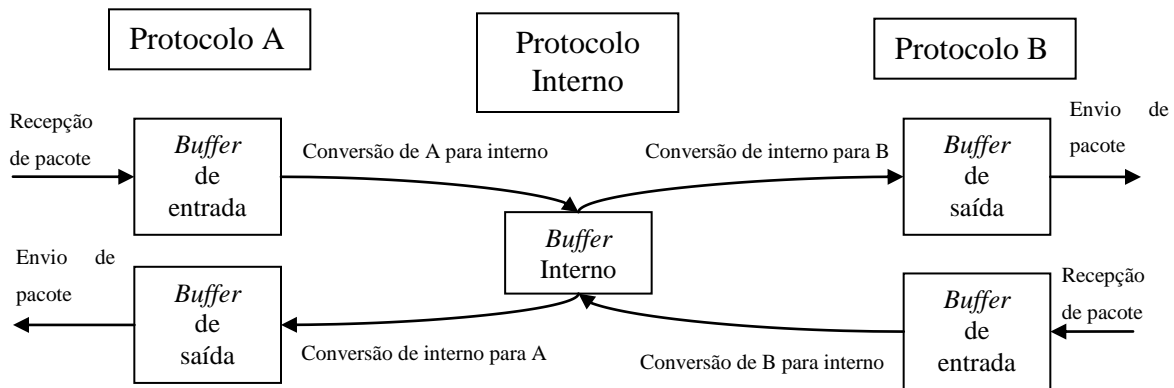


Figura 40: Diagrama do despacho dos pacotes na *gateway*

O algoritmo de recepção do protocolo A recebe e coloca um pacote no *buffer* de entrada. Por sua vez o pacote recebido é removido do *buffer* de entrada pelo algoritmo de conversão de A para o protocolo interno. Caso a conversão seja bem sucedida é adicionado o pacote ao *buffer* interno.

O pacote do *buffer* interno é despachado pelo algoritmo de conversão interno para o protocolo B, que coloca um pacote no *buffer* de saída do protocolo B. Por sua vez, na última fase do percurso de um pacote pela *gateway*, o pacote contido nos *buffers* de saída é transmitido pelo algoritmo de envio do protocolo B.

O mesmo processo de despacho dos pacotes aplica-se em ambos os sentidos de comunicação e para qualquer combinação dos protocolos A e B (onde A e B podem ser qualquer protocolo implementado na *gateway*), mesmo que o protocolo A seja o mesmo que B.

Os pacotes dos vários *buffers* são despachados seguindo um algoritmo de escalonamento *round robin*, onde um pacote de cada *buffer* é despachado em cada passagem (Figura 41). Caso um *buffer* esteja vazio o algoritmo passa para o próximo *buffer*.

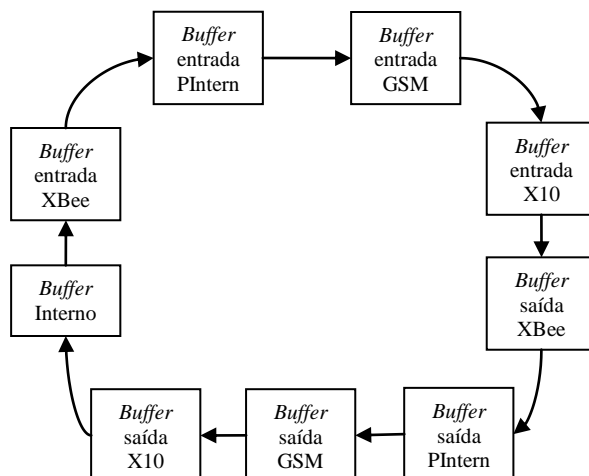


Figura 41: Diagrama do escalonamento de despacho de pacotes da gateway

O algoritmo verifica a existência de pacotes no *buffer* de entrada de cada protocolo. Se houver pacotes é chamada a função de conversão para o protocolo interno apropriada (por exemplo: quando recebido um pacote XBee é chamada a função que converte o pacote XBee para um pacote PIntern).

Depois de verificados os *buffer* de entrada passa-se à verificação da existência de pacotes nos *buffer*s de saída de cada protocolo. Caso existam pacotes, estes são despachados através da chamada da função de envio apropriada.

Se existirem pacotes no *buffer* interno, é analisado o endereço de destino podendo o pacote ser encaminhado por *unicast*, *broadcast* ou tabela de rotas.

### 3.3.8.1 Unicast

Caso o endereço de destino não seja um dos endereços reservados, é procurado o endereço de destino na lista de dispositivos. Caso este seja encontrado é verificado o protocolo do dispositivo e é chamada a função de conversão apropriada.

### 3.3.8.2 BroadCast

Se o endereço de destino é o endereço de *broadcast* (endereço 255), é percorrida a tabela de dispositivos, verificando o protocolo e chamando a função apropriada para a conversão, para cada elemento.

Inicialmente, o algoritmo de *broadcast* construía novos pacotes internos com os endereços dos dispositivos da lista, mas esta técnica sobrecarregava o *buffer* interno e

introduzia um maior atraso no encaminhamento dos pacotes. A solução adotada foi realizar já a conversão para o protocolo de destino apresentando um menor atraso e evitando a sobrecarga do *buffer* interno. Como os dispositivos encontram-se distribuídos pelos vários protocolos evita-se a sobrecarga dos *buffers* de saída.

### 3.3.8.3 Tabela de rotas (Multicast)

Se o endereço de destino é o endereço 254 (encaminhamento pela tabela de rotas), a tabela de rotas é percorrida obtendo-se uma lista de endereços de destino. A lista de destinos é obtida de todas as rotas cujo endereço de origem da rota corresponde ao endereço de origem do pacote a encaminhar.

Para cada endereço de destino da lista é realizada uma conversão para o protocolo externo.

Quer o procedimento de encaminhamento por *broadcast* quer o procedimento pela tabela de rotas introduzem atrasos mais elevados de processamento visto ser necessário construir vários pacotes. Mas para o tipo de aplicações a que a *gateway* se destina o atraso introduzido é aceitável. Em alguns casos, como por exemplo o tratamento de pacotes X10, os algoritmos de conversão e encaminhamento introduzem atrasos relativamente pequenos quando comparados com o atraso de envio ou recepção de um pacote X10 (próximo de 1 segundo).

A técnica de encaminhamento pela tabela de rotas permite a formação de pequenos grupos de equipamentos para envio de pacotes através da definição das rotas adequadas. A formação de grupos de maior dimensão torna-se demasiado complexa com esta técnica.

Todo o processo de encaminhamento está contido num ciclo que se repete indefinidamente, realizando conversões ou envios de pacotes, sempre que estes existam nos *buffers*. Caso não existam pacotes o programa fica continuamente a verificar a existência de novos pacotes.



## 4 Sistema inteligente de gestão de energia

Neste capítulo é descrita a implementação de uma aplicação que comunica através da *gateway* desenvolvida com os equipamentos da rede de automação.

São definidos os objectivos a atingir com o desenvolvimento da aplicação e é efectuada a descrição do funcionamento geral da mesma.

O software foi implementado utilizando a linguagem de programação JAVA, embora a descrição seja efectuada com alguma abstracção à linguagem utilizada. Para a implementação do código e execução da aplicação utilizou-se o programa NetBeans 7.0 e a aplicação Xampp que inclui um servidor Web e uma base de dados MySQL [40], [41].

Ao longo da descrição são indicados os principais elementos utilizados para a implementação, sendo apresentada uma breve descrição da sua funcionalidade.

Os principais objectivos desta aplicação eram:

- Demonstrar as vantagens do desenvolvimento de uma aplicação que recebe/envia dados e comandos de/para equipamentos com diferentes protocolos através de um só formato comum (PIntern);
- Correr um sistema de despacho de energia automático que comunica através da *gateway* para enviar comandos para os actuadores de cargas.

A aplicação principal foi desenvolvida de forma a representar uma rede virtual de dispositivos que possuem endereços próprios, simulando uma rede baseada no protocolo interno (PIntern).

A aplicação foi desenvolvida deste modo para preparar o sistema protótipo para passar a um sistema distribuído, onde as várias aplicações (dispositivos) poderão se encontrar distribuídas nos diversos computadores de uma rede doméstica.

Nesta rede virtual são inseridos 3 dispositivos virtuais com diferentes funcionalidades que serviram para demonstrar as potencialidades da utilização da *gateway*:

- Gestor da *Gateway*;
- Web GUI (Graphical User Interface);
- Despacho de energia.

O gestor da *gateway* é visto como um dispositivo virtual na rede que envia e recebe pacotes da *gateway* permitindo o preenchimento das tabelas de dispositivos e rotas.

Outro dispositivo virtual introduzido funciona como um interface gráfico para o utilizador baseado numa página Web, acessível através de um *browser*.

Para o armazenamento dos dados dos sensores da rede é também introduzido um dispositivo que recebe pacotes e armazena os mesmos numa base de dados. Este objecto JAVA estabelece uma ligação com uma base de dados MySQL onde são armazenados os dados através da execução de *queries*.

Por último, a aplicação de despacho de energia analisa os níveis de consumo e de produção da rede eléctrica, e determina que cargas deverá ligar ou desligar com base nessa situação. Esta aplicação envia comandos para os actuadores da rede de forma a ligar e desligar os equipamentos.

## 4.1 Principais elementos

Para a implementação da aplicação foram definidos vários elementos. Nesta secção serão referidos os elementos de maior importância, indicando as suas principais funcionalidades.

Estes elementos são essenciais para a implementação da aplicação como uma rede virtual e para esta comunicar com a *gateway* podendo ser os mesmos a base para o desenvolvimento de outras aplicações que utilizam a *gateway*.

Os principais elementos utilizados nesta aplicação são:

- Interface *Gateway*;
- Pacote PIntern;
- Encaminhador de pacotes;
- Dispositivo;
- Interface da base de dados.

### 4.1.1 Interface da *gateway*

É através deste elemento que é realizada a comunicação com a *gateway*. Este elemento encarrega-se da construção e envio dos pacotes, criando uma abstracção para a aplicação no envio de pacotes.

Os pacotes são enviados e recebidos através de uma porta série. Os pacotes recebidos e por enviar são armazenados em *buffers* para compensar o afunilamento de pacotes e efectuar a

adaptação das taxas de transmissão. São utilizados dois *buffers* FIFO que armazenam os pacotes pela ordem de chegada.

Este elemento necessita de efectuar duas tarefas:

- Recepção de pacotes e colocação no *buffer* de entrada
- Envio dos pacotes colocados no *buffer* de saída e remoção dos mesmos

A recepção de pacotes é efectuada começando pela detecção do início do pacote PIntern (*byte* 0x90) e obtenção do segundo *byte*. O segundo *byte* é a dimensão do pacote, que é utilizada para saber quando termina a recepção do mesmo. De notar que é necessário receber sempre mais um *byte* do que o indicado, pois o *checksum* do pacote não está incluído na dimensão.

O envio é efectuado sempre que exista um pacote no *buffer* de saída.

O interface *gateway* permite construir pacotes para serem enviados para a *gateway* de forma a realizar as seguintes acções:

- Adicionar/remover dispositivo da lista de dispositivos;
- Adicionar/remover rota da lista de rotas;
- Mostrar lista de dispositivos e lista de rotas;
- Guardar em memória não volátil as listas;
- Encaminhar pacotes.

Como o interface com a *gateway* é um elemento único que pode ser acedido por vários dispositivos em simultâneo existe a necessidade de garantir a exclusão mútua, de forma a prevenir erros.

O interface da *gateway* é implementado pela classe *MP\_module* na aplicação desenvolvida.

#### 4.1.2 Pacote PIntern

Este elemento representa os pacotes PIntern, recebidos ou a enviar para a *gateway*, que são encaminhados para os destinatários finais (dispositivos virtuais).

Este pacote contém os endereços de origem e de destino da informação contida no pacote, permitindo encontrar o destinatário e entregar o pacote.

Este também indica o tipo de conteúdo do pacote podendo ser um comando, uma mensagem ou uma notificação. A integridade do pacote é verificada por este elemento através da verificação e cálculo do *checksum*.

Na aplicação JAVA desenvolvida este elemento é implementado pela classe *Msg*.

### 4.1.3 Encaminhador de pacotes

Este elemento funciona como o *router* da rede virtual implementada pela aplicação.

A sua função é buscar os pacotes recebidos pelo interface da *gateway*, localizados no *buffer* de entrada, e encaminha-los para o dispositivo de destino (indicado pelo endereço de destino do pacote).

O encaminhamento é efectuado através da leitura do endereço de destino do pacote, sendo efectuada uma procura do dispositivo de destino na lista de dispositivos. Caso o dispositivo seja encontrado o pacote é entregue ao mesmo. Caso não seja encontrado o pacote é descartado.

Na rede virtual implementada na aplicação JAVA o encaminhador de pacotes é implementado pela classe *Msg\_dispatch*.

### 4.1.4 Dispositivo virtual

O dispositivo virtual representa um nó da rede virtual de onde surgem ou para onde se destinam os pacotes.

Cada dispositivo é identificado por um endereço único e possui um *buffer* para armazenar os pacotes recebidos.

O envio de comandos e mensagens dos dispositivos virtuais para a rede é efectuado através do interface da *gateway* aproveitando as funcionalidades de construção e envio de pacote. Desta forma é possível realizar o envio de forma abstracta sem saber como é construído um pacote.

Cada dispositivo virtual realiza a interpretação dos dados dos pacotes para efectuar alguma acção, armazenar informação ou enviar novos pacotes. Dependendo do tipo de dispositivo virtual a interpretação dos dados difere.

O processo de interpretação começa pela determinação do tipo de pacote (comando, mensagem ou notificação) sendo a interpretação de cada tipo efectuada de forma diferente.

#### 4.1.5 Interface da base de dados

O interface com a base de dados realiza a ligação a uma base de dados MySQL. Os dados são adicionados, ou obtidos da base de dados, através da execução de *querys* predefinidas.

As *querys* predefinidas estão parcialmente preenchidas e são completadas utilizando informação indicada pelos dispositivos virtuais.

Tal como na interface da *gateway* é necessário garantir a exclusão mútua para prevenir erros.

As *querys* predefinidas permitem efectuar as seguintes acções na base de dados:

- Adicionar, remover e visualizar dispositivos da tabela de dispositivos;
- Adicionar, remover e visualizar rotas da tabela de rotas;
- Adicionar e listar associações entre elemento da página Web com dispositivos da rede;
- Obter informação associada a elementos da página Web;
- Armazenamento e leitura de dados (atributos) enviados pelos dispositivos de rede.

A estrutura da base de dados é constituída por várias tabelas que contêm informação necessária para o funcionamento da aplicação e processamento da informação (Figura 42). Também contém uma tabela utilizada para armazenar a informação recolhida pelo sistema de automação.

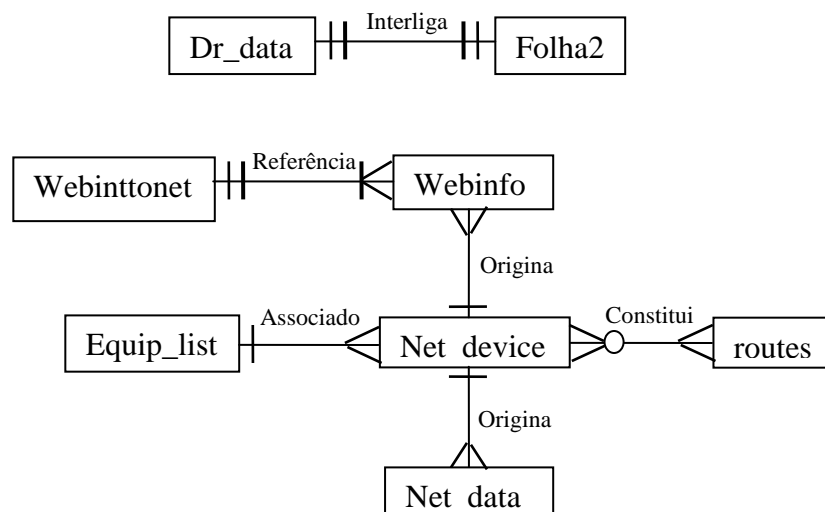


Figura 42: Diagrama Entidade-Associação da base de dados segundo representação Crow's Foot

O diagrama Entidade-Associação estabelece as relações entre as informações da tabela para permitir a construção de *queries* mais complexas para a aquisição e modificação da informação na base de dados. As *queries* predefinidas utilizadas são indicadas no Anexo H na listagem do código da classe *DataBase*.

A tabela *Net\_device* contém a informação sobre cada dispositivo da rede que é registado na *gateway*. Cada elemento da tabela indica o endereço interno, e do protocolo (XBee, X10 e GSM). Também contém informação do protocolo e camada de aplicação do dispositivo.

A tabela de rotas, “*routes*”, armazena as rotas que são utilizadas na configuração da *gateway*.

O gestor da *gateway* interage com as tabelas para armazenar ou carregar a lista de dispositivos e de rotas da *gateway*.

A tabela “*Net\_data*” é utilizada para armazenar os dados recebidos dos vários equipamentos da rede. Cada elemento da lista contém o endereço de origem, identificador do atributo, nome do atributo, valor e *Timestamp* (indicação da data e hora de introdução do valor).

Na tabela “*Equip\_list*” está inserida a informação de cada equipamento incluído no sistema de despacho. Cada elemento contém informação como o identificador do equipamento, descrição, tipo, estado inicial e consumo de potência. Também é incluído o endereço do dispositivo de rede que permite ligar ou desligar o equipamento. Muitas vezes o equipamento está associado a um dispositivo sensor que efectua medições do parâmetro que rege o funcionamento do mesmo. No elemento da tabela “*Equip\_list*” é indicado o endereço do sensor.

Na tabela “*webinttonet*” estão contidas as associações efectuadas entre os elementos do interface Web e do dispositivo da rede controlado. Também é indicado o identificador da informação associada a este.

Na tabela “*webinfo*”, está a listagem da informação associada a um elemento de interface Web. Cada elemento indica a informação ou referência da origem da informação (Exemplo: endereço do sensor de origem).

Nas tabelas “*DR\_data*” e “*folha 2*”, está a informação utilizada pelo sistema de despacho. Em “*DR\_data*” contém os valores da potência de consumo e produção. A tabela “*folha 2*” contém as temperaturas para cada hora do dia.

## 4.2 Gestor da *Gateway*

O gestor da *Gateway* é utilizado para configurar os dispositivos e rotas na *gateway*.

Este é um caso específico de um dispositivo da rede virtual que implementa o seu próprio comportamento e interpretação dos pacotes recebidos. Também apresenta a funcionalidade de envio de mensagens de forma a demonstrar a funcionalidade do módulo e para testar os algoritmos de conversão.

O objectivo do gestor é a construção das tabelas de dispositivos e de rotas de forma coerente, para depois poderem ser sincronizadas as tabelas da aplicação com as tabelas da *gateway*.

O gestor tem a capacidade de gerar endereços internos e endereços X10 de forma aleatória, garantindo que não existe repetição dos mesmos. Garante também que não são atribuídos os endereços reservados. Estes endereços gerados são utilizados na adição de novos dispositivos à lista.

Permite também a definição coerente das rotas garantindo que os dispositivos indicados encontram-se na lista de dispositivos e evitando a definição repetida de rotas.

O gestor é implementado na aplicação pela classe *Module\_manager*.

Antes de avançar com a descrição da implementação do gestor é necessário descrever alguns dos elementos necessários.

### 4.2.1 Dispositivo da rede

É utilizado para representar um dispositivo registado na tabela de dispositivos da *gateway*.

Este elemento contém um endereço para cada protocolo implementado (nomeadamente PIntern, XBee, X10 e GSM).

Com a implementação de novos protocolos na *gateway* será necessário adicionar o endereço deste ao elemento dispositivo da rede. Neste elemento também é indicado o protocolo nativo do dispositivo e a camada de aplicação utilizada. Neste elemento pode também ser incluído um texto descritivo do dispositivo representado. Na aplicação desenvolvida este elemento é implementado pela classe *Net\_device*.

#### 4.2.2 Rota

É utilizada para representar uma rota para o encaminhamento de pacotes. Numa rota é indicado o endereço de origem e o destino dos pacotes. A classe `route_entry` implementa este elemento na aplicação desenvolvida.

#### 4.2.3 Lista de dispositivos e lista de rotas

São as listas que contêm os elementos do tipo dispositivo da rede e de rota.

A lista pode ser estática ou dinâmica (lista ligada), sendo neste caso mais vantajosa a utilização de uma lista dinâmica, pois permite a fácil reordenação, adição e remoção de elementos. Estas listas apresentam funcionalidades que permitem procurar, adicionar e remover elementos. Estas listas são implementadas na aplicação desenvolvida pelas classes `Device_list` e `Route_list`.

#### 4.2.4 Gerador de endereços

O gerador de endereços é utilizado para gerar endereços para os protocolos PIntern e X10.

Quando é efectuado um pedido para gerar um endereço PIntern, é gerado um número aleatório de 0 a 253. É verificado se este endereço já foi atribuído ou se é um dos endereços reservados.

Para gerar um endereço X10, são gerados dois números aleatórios de 0 a 15. Um dos números é utilizado para definir o *homecode* e o outro para definir o *unitcode* de um dispositivo X10.

Este elemento também possui a funcionalidade de converter os endereços XBee e X10 da sua representação em *bytes* para uma *String*. Também permite converter de *String* para a representação em *bytes* dos endereços XBee e X10. Este elemento é implementado na aplicação JAVA pela classe `Addr_generator`.

#### 4.2.5 Interface com o utilizador por linha de comandos

O elemento que permite ao utilizador configurar a *gateway* é o interface pela linha de comandos. Os dados são introduzidos pelo utilizador através da linha de comandos da aplicação.

Na inicialização do interface é pedido ao utilizador que indique onde pretende carregar as tabelas que irá utilizar.

As opções suportadas actualmente são:

- 1 - Carregar da base de dados: as tabelas são preenchidas com o conteúdo presente na base de dados da aplicação;
- 2 - Carregar do módulo (*gateway*): as tabelas são preenchidas com o conteúdo presente na *gateway*;
- 3 - Tabelas novas: inicia tabelas de dispositivos e rotas vazias.

A acção de carregamento seleccionada é efectuada pelo gestor do módulo a pedido do interface. O gestor utiliza as suas ligações com a base de dados ou com a *gateway* para obter os elementos para preencher as listas.

Uma vez carregadas as tabelas de dispositivos e de rotas, o interface entra num ciclo no qual efectua o pedido de um comando e aguarda a resposta.

O texto introduzido pelo utilizador na linha de comandos, é dividido em duas partes que se encontram separadas por um caractere espaço. Na primeira parte está contida a indicação do comando a executar e a segunda parte contém os parâmetros a utilizar para a execução do comando indicado. Quando o comando não é valido é enviada uma notificação ao utilizador. Diferentes comandos requerem diferentes parâmetros que são interpretados pelo interface de linha de comandos. Depois de interpretados os comandos o interface pede ao gestor para executar os comandos, indicando os parâmetros necessários.

O gestor quando necessita de enviar uma resposta ou uma notificação ao utilizador efectua o pedido ao interface que coloca a informação na linha de comando. O interface com o utilizador é implementado na aplicação pela classe *ComandLine\_ui*.

O gestor apresenta duas listas de dispositivos da rede que são utilizadas para a construção da lista de dispositivos a colocar na *gateway*. Uma das listas é modificada pelo utilizador através do envio de comandos para adicionar ou remover elementos e a outra é uma lista temporária utilizada no procedimento de sincronização. O mesmo se aplica para as duas listas de rotas presentes no gestor.

Para cada comando de adicionar ou remover um elemento da lista, introduzido pelo utilizador, é devolvida a listagem dos elementos da lista modificada para que o utilizador consiga acompanhar as modificações realizadas.

O processo de adição de elementos na lista de dispositivos da rede utiliza os dados indicados pelo utilizador no comando inserido para construir o elemento a adicionar. Durante a construção é pedido ao gerador de endereços para gerar um endereço PIntern para associar ao dispositivo de rede. Caso não seja indicado o endereço X10 é também pedido ao gerador de endereços para gerar um endereço X10 para associar ao dispositivo da rede. O dispositivo de rede depois de construído é adicionado à lista.

O procedimento de remoção é bem mais simples. Basta utilizar o endereço do dispositivo de rede, indicado como parâmetro do comando inserido pelo utilizador, para procurar o dispositivo na lista e removê-lo. Para garantir que os endereços indicados nas rotas existem na lista de dispositivos, é removida da lista de rotas qualquer rota que contenha o endereço a remover, quer seja origem ou destino.

O procedimento de adição de uma rota começa com a introdução do comando pelo utilizador. No comando são indicados os endereços de origem e de destino a utilizar na construção da rota. É verificada a existência dos endereços indicados na lista de dispositivo e, caso existam ambos, a rota é adicionada à lista de rotas.

A remoção de uma rota da lista de rotas utiliza também os parâmetros introduzidos pelo utilizador para efectuar a procura e conseqüente remoção da rota.

Depois de construídas as tabelas de rotas e de dispositivos de rede é necessário passar o conteúdo destas para a *gateway*. Uma das soluções seria enviar o comando para a *gateway* efectuado sempre que fosse efectuada uma alteração. Esse método não garante que ambas as listas fiquem idênticas quando são carregadas de locais diferentes, por exemplo, se a lista a modificar foi carregada como vazia e se na lista da *gateway* existem elementos depois das modificações as listas não ficam iguais. Outro problema é o envio de comandos para a *gateway* sempre que é efectuada uma alteração. Assim, optou-se por realizar a modificação das listas da *gateway* através de um procedimento de sincronização por duas razões:

- As modificações das listas são efectuadas de forma sequencial e na mesma altura;
- O número de comandos necessários depende das diferenças existentes entre as listas do gestor e as listas da *gateway*.

A sincronização começa com a obtenção das listas presentes na *gateway*, através do envio do comando para listar os elementos das mesmas, aguardando algum tempo para serem recebidas todas as respostas.

Depois de preenchidas as listas temporárias com os elementos, são verificados quais os elementos da lista do gestor que faltam na lista da *gateway*. Os elementos que faltam são adicionados às listas da *gateway* através do envio dos comandos para adição de elementos na *gateway*.

Depois são verificados os elementos das listas da *gateway* que não existem nas listas do gestor. Os elementos que não existam são removidos da *gateway* através do envio dos comandos de remoção.

Depois de realizada a sincronização é enviado um comando para armazenar as alterações efectuadas nas listas em memória não volátil. Isto permite que as listas possam ser carregadas depois de um *reset* da *gateway* ou de uma falha de energia.

Como o gestor é um caso específico de um dispositivo virtual é necessário que este possa interpretar os pacotes que possam chegar. Sempre que chega um novo pacote é verificado o tipo do conteúdo do mesmo.

No caso de uma mensagem ou notificação é enviada para o interface com o utilizador a notificação ou mensagem recebida.

No caso de um comando é verificado qual o comando recebido e é realizada a acção apropriada. Se o comando recebido for de adição ou remoção de um elemento das listas (rotas ou dispositivo de rede) é construída uma notificação em forma de texto e enviada para o interface com o utilizador. A recepção deste comando implica que a execução do mesmo na *gateway* foi bem sucedido.

Se o comando for uma resposta a um pedido de listagem das listas, os dados do pacote são interpretados para se obterem os diferentes parâmetros. Esses parâmetros são utilizados para construir os elementos que são adicionados às listas temporárias apropriadas.

### 4.3 Web GUI

O dispositivo Web GUI foi utilizado para demonstrar a potencialidade de desenvolvimento de um interface gráfico interactivo baseado numa página Web, acedida através de um *browser*.

O desenvolvimento deste dispositivo foi dividido em dois segmentos:

- Página Web (Figura 43);
- Aplicação no servidor que recebe as acções e encaminha-as para a rede.

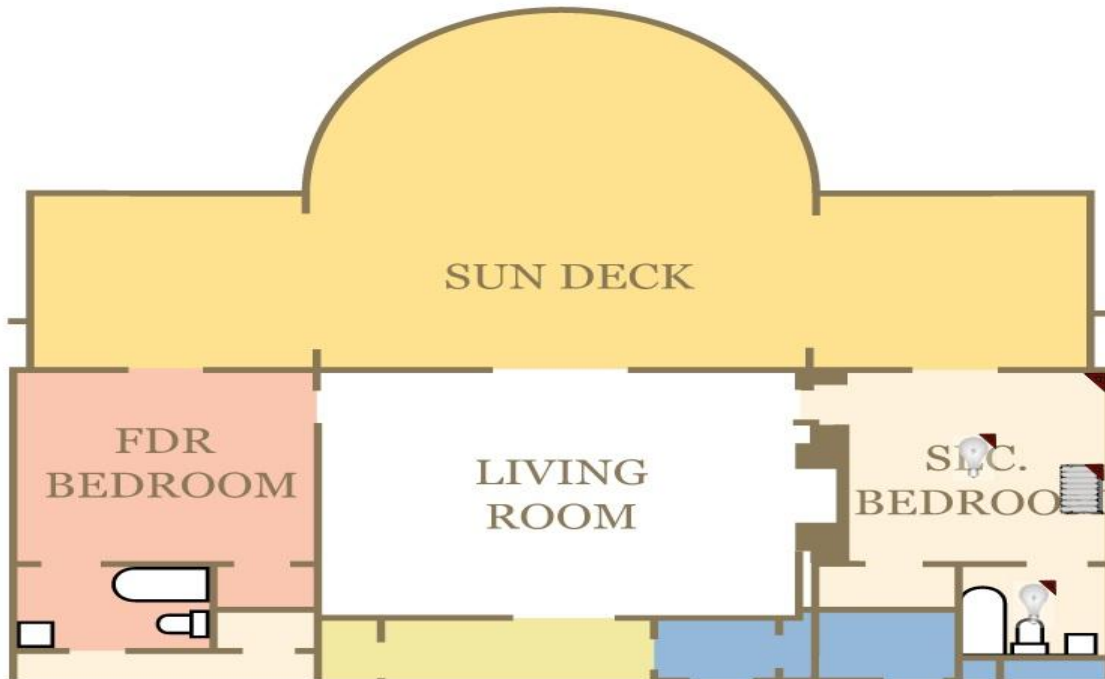


Figura 43: Página Web de interface gráfico

A página Web foi desenvolvida utilizando uma combinação de HTML, SVG, JavaScript e JAVA *applets*. O código HTML utilizado foi constituído por um cabeçalho e o corpo. No corpo são adicionados dois objectos. Um dos objectos foi a *applet* JAVA que é introduzida com a menor dimensão possível, de forma a torná-la invisível ao utilizador. O outro objecto é uma imagem SVG que define o aspecto da página Web.

A página torna-se dinâmica através da utilização do JavaScript, sendo necessário o carregamento dos métodos contidos num ficheiro no servidor. Este carregamento é efectuado no cabeçalho da página.

Para a construção da imagem SVG foi utilizado o programa Inkscape que além das suas capacidades de desenho de imagens SVG permite associar métodos JavaScript a acções efectuadas sobre os elementos SVG (*mouseover*, *mouseout*, *mouseclick*, etc.).

Na realidade a imagem SVG não passa de um ficheiro XML que descreve como é desenhada a imagem. Qualquer modificação pode ser efectuada directamente no código XML, se for necessário. É importante realçar que o ficheiro que contém os métodos JavaScript a serem executados deverá ser carregado pelo código XML da imagem SVG.

O ícone de uma lâmpada é bastante intuitivo e representa um dispositivo da rede de automação, que controla uma lâmpada, e é capaz de alternar entre o estado ligado e desligado.

O outro ícone utilizado representa um aquecedor. Este representa o mesmo tipo de dispositivo da rede que o ícone da lâmpada, mas que controla um aquecedor em vez de uma lâmpada.

Ao clicar sobre os ícones estes mudam de estado e é enviada para a aplicação no servidor a indicação da acção que o utilizador realizou, de forma a propagar o efeito para a rede de automação. Esta mudança é efectuada dinamicamente através da invocação dos métodos JAVAScript implementados.

Os ícones de informação (ícones triangulares colocados nos cantos superiores direitos dos ícones) quando clicados fazem surgir um painel onde é apresentada a informação do elemento. No momento em que o ícone é clicado é efectuado o pedido da informação ao servidor, para preencher o painel.

Para encaminhar as acções e pedidos para o servidor, o método JAVAScript invoca um método da *applet* inserida na página que por sua vez invoca um método do objecto servidor remoto. Caso a invocação do método do objecto remoto retorne alguma resposta, os dados são utilizados pela *applet* para invocar métodos JAVAScript de forma a actualizar pequenos segmentos da página Web.

Para a comunicação com o servidor poderiam ter sido utilizadas outras técnicas como a técnica AJAX (*Asynchronous JAVAScript and XML*). Esta técnica permite a criação de páginas Web dinâmicas, permitindo que sejam carregados somente pequenos segmentos da página, em vez da página toda. A comunicação é efectuada com o servidor através de pedidos HTTP.

De entre as várias técnicas de troca de informação entre uma página Web e o servidor, para a implementação de uma página Web dinâmica, foi proposta e implementada neste projecto a comunicação utilizando a tecnologia JAVA RMI (*Runtime Method Invocation*).

Optou-se por utilizar esta técnica porque demonstra algumas das potencialidades de desenvolvimento de uma aplicação distribuída utilizando a linguagem JAVA.

A página Web contém uma *applet* “invisível” a correr no browser, que comunica com um objecto servidor na aplicação principal através do RMI. A utilização do RMI implica o desenvolvimento de uma aplicação cliente (*Applet* da página Web) e de uma aplicação servidor, a correr num local acessível ao cliente (servidor Web ou outro computador da rede).

Esta solução apresenta algumas vantagens e desvantagens [42], [43]:

Vantagens:

- Podem ser passados como argumentos e retornados objectos JAVA na invocação de métodos remotos;
- Utiliza os mecanismos de segurança disponibilizados pelo JAVA;
- O desenvolvimento da aplicação JAVA distribuída é efectuado da mesma forma que uma aplicação normal, sendo praticamente só necessário incluir 3 linhas de código, para tornar uma aplicação normal numa aplicação servidor;
- Qualquer máquina virtual JAVA suporta aplicações com RMI independentemente da plataforma;
- São utilizadas *threads* JAVA para lidar com os vários pedidos efectuados às aplicações servidoras;
- Distribuição de aplicações JAVA sem a necessidade de as instalar em cada computador.

Desvantagens:

- Nem todos os browsers suportam algumas técnicas de comunicação do RMI como o HTTP *tunneling*;
- O RMI *registry* nem sempre lida bem com ligações através da Internet;
- É necessária a configuração adicional da *firewall* para tornar um objecto servidor RMI acessível.

#### 4.4 Despacho de energia automático

O objectivo principal deste sistema foi o controlo de cargas (equipamentos eléctricos) com base na situação actual e prevista de consumo e produção de energia.

No âmbito de uma *smart grid* uma das características importantes é a capacidade de redireccionar a energia rapidamente, sendo para esse efeito necessário haver um controlo sobre as cargas ligadas à rede. É nesse sentido que um sistema de despacho de energia dá a sua contribuição.

O sistema proposto em seguida não foi definido especificamente para um determinado sistemas de automação, mas sim de uma forma genérica, podendo ser implementado em vários sistemas de automação.

O sistema analisa os níveis de consumo e produção de energia e determina, através de um algoritmo, se o momento é propício para a activação ou desactivação de uma dada carga.

Foi escolhido como critério a diferença de potência, para não fixar o funcionamento do sistema a um determinado sistema de preços. Note-se que o preço da energia depende de muitos factores do mercado, que define como é efectuada a variação do preço em função dos níveis de produção e consumo.

Os níveis de produção e consumo de energia na rede eléctrica podem ser facilmente adquiridos e transmitidos, pela empresa responsável pela rede eléctrica, sem a necessidade de muito processamento, tornado o sistema mais simples.

Este sistema pode facilmente ser modificado para receber o preço da energia, em vez dos níveis de consumo e produção, sem afectar muito o comportamento do sistema, sendo somente necessário alterar a função do factor de incentivo de energia.

O processo é efectuado para cada equipamento (carga) de forma separada, pois cada um tem diferentes consumos e factores que influenciam a decisão.

A primeira fase do processo, que é comum a todos, começa pela obtenção dos níveis de consumo e produção de energia, sendo calculada a diferença entre os níveis (4.1):

$$P_{diff} = P_{consumida} - P_{produzida} \quad (4.1)$$

A diferença ( $P_{diff}$ ) entre a potência consumida ( $P_{consumida}$ ) e produzida ( $P_{produzida}$ ) é utilizada para a decisão em termos de energia. Se a diferença for positiva implica que está a ser consumida mais energia do que produzida, fazendo com que o incentivo a ligar seja o mínimo possível e a desligar o máximo.

O factor de incentivo de energia ( $I_{energy}$ ) é definido por duas funções dependentes de dois parâmetros (4.2), (4.3): a diferença de potência ( $P_{diff}$ ) e a margem de potência ( $P_{margin}$ ). Na Figura 44 é apresentada a representação gráfica das funções.

$$I_{energyON} (P_{diff}, P_{margin}) = \begin{cases} 1 & \text{se } P_{diff} \leq P_{margin} \\ -\frac{P_{diff}}{P_{margin}} & \text{se } P_{margin} < P_{diff} < 0 \\ 0 & \text{se } P_{diff} \geq 0 \end{cases} \quad (4.2)$$

$$I_{energyOFF} (P_{diff}, P_{margin}) = 1 - I_{energyON} (P_{diff}, P_{margin}) \quad (4.3)$$

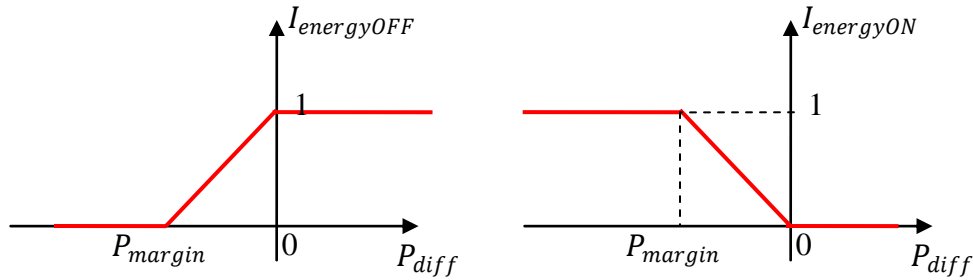


Figura 44: Gráficos do factor de incentivo de energia

Uma função é o incentivo dado para ligar ou manter ligado o equipamento ( $I_{energyON}$ ) (4.2) e a outra é o incentivo para desligar ou manter desligado ( $I_{energyOFF}$ ) (4.3).

A margem de potência indica o nível mínimo de diferença de potências, abaixo do qual o incentivo é sempre máximo para  $I_{energyON}$  e mínimo para  $I_{energyOFF}$ . A margem de potência é indicada pelo utilizador na configuração do sistema, e deverá permanecer constante na maioria dos casos, tornando o resultado da função de incentivo dependente só da diferença de potência.

Para a tomada de decisão o incentivo é calculado para duas situações:

- Manter o estado do equipamento;
- Mudar o estado do equipamento (ligar ou desligar).

Na análise sobre manter ou não o estado do aparelho o incentivo é calculado utilizando directamente a diferença de potência. A diferença de potência já inclui o efeito do estado actual do equipamento.

Caso seja para mudar o estado é necessário inserir o efeito da decisão no valor da diferença de potência antes do cálculo do incentivo. Caso se analise o incentivo de ligar um equipamento, a potência do equipamento deve ser adicionada à diferença de potência. Caso se analise o incentivo de desligar, a potência deve ser subtraída à diferença de potência.

O procedimento de introduzir o efeito da decisão permite distinguir entre o incentivo de manter o estado actual ou de mudar de estado.

O cálculo do incentivo é efectuado para a diferença de potência actual e para uma série de previsões de períodos seguintes. O número de períodos pode ser escolhido pelo utilizador tendo sido utilizados 4 períodos (actual mais 3 previsões) no sistema implementado.

Se o incentivo actual for superior ao incentivo dos outros períodos é retornado o incentivo actual, caso contrário é retornado o incentivo mínimo “0” para indicar que reagir noutro período é mais propício. É claro que esta decisão pode não ser a melhor principalmente

quando o incentivo tende a aumentar durante alguns períodos ocorrendo um adiamento da decisão por vários períodos.

Outro factor que é considerado na decisão é o conforto. Diferentes equipamentos são analisados de forma distinta em termos de conforto sendo este factor específico a cada. Existem equipamentos que não utilizam o factor de conforto.

Como exemplo pode-se trabalhar com o factor de conforto de um aquecedor de um quarto. Tal como o factor de energia, o factor de conforto segue uma função. Existe uma função para o incentivo em termos de conforto para ligar ( $I_{confortON}$ ) o equipamento e uma para desligar ( $I_{confortOFF}$ ).

Esta função depende de três parâmetros (ver Figura 45): temperatura actual ( $T_{actual}$ ), nível de temperatura escolhida pelo utilizador ( $T_{user}$ ) e margem de temperatura ( $T_{margin}$ ).

$$I_{confortON}(T_{actual}, T_{user}, T_{margin}) = \begin{cases} 1 & \text{se } T_{actual} \leq T_{user} - T_{margin} \\ -\frac{T_{actual} - T_{user}}{T_{margin}} & \text{se } T_{user} - T_{margin} < T_{actual} < T_{user} \\ 0 & \text{se } T_{actual} \geq T_{user} \end{cases} \quad (4.4)$$

$$I_{confortOFF}(T_{actual}, T_{user}, T_{margin}) = 1 - I_{confortON}(T_{actual}, T_{user}, T_{margin}) \quad (4.5)$$

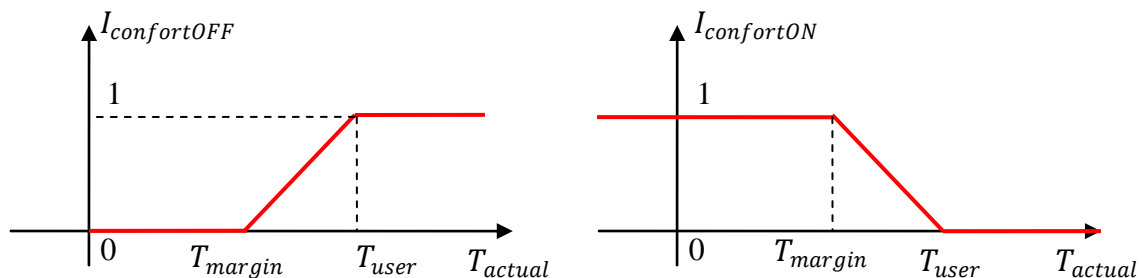


Figura 45: Gráficos do factor de incentivo de conforto

A margem de temperatura indica o valor mínimo da temperatura abaixo da qual é necessário ligar o equipamento para não afectar o conforto. Ultrapassada essa margem o incentivo para ligar o aparelho deverá ser máximo e o incentivo para o desligar deverá ser o mínimo.

O sistema suporta a introdução de mais factores de incentivo que influenciem a decisão final, permitindo melhorar o desempenho do sistema.

No valor do incentivo final, (4.6) e (4.7), cada factor tem o seu peso na decisão da acção. Este peso é definido pelo utilizador de forma a dar mais importância ao conforto ou à gestão da energia.

$$I_{finalON} = I_{energiaON} \times \textit{peso da energia} + I_{confortoON} \times \textit{peso do conforto} \quad (4.6)$$

$$I_{finalOFF} = I_{energiaOFF} \times \textit{peso da energia} + I_{confortoOFF} \times \textit{peso do conforto} \quad (4.7)$$

Os incentivos finais calculados para as duas decisões são comparados e o que apresentar melhor resultado (maior incentivo) é o escolhido.

Depois de a decisão ser tomada, são enviados comandos para a rede de forma a executá-la. Também é introduzido o efeito da decisão nos níveis de consumo.

Este procedimento é efectuado para cada equipamento incluído no sistema de despacho e quantos mais equipamentos forem incluídos melhor será o resultado e flexibilidade do sistema.

Com uma instalação numa só casa o efeito do sistema é insignificante na rede eléctrica, mas se a instalação for efectuada num grande número de casas o efeito será mais significativo.

Este sistema é um sistema de despacho distribuído na rede eléctrica. O processamento da informação e a tomada de decisão é realizado em cada casa individualmente, com as configurações definidas por cada utilizador, fazendo com que o comportamento individual seja distinto.

Como não existe sincronismo no momento de tomada da decisão, evita-se uma variação acentuada do consumo num curto espaço de tempo, sendo as decisões realizadas ao longo de um período. Se a variação dos níveis de consumo forem muito acentuadas num curto espaço de tempo poderá provocar problemas na rede eléctrica.

Para evitar a tomada de decisão simultânea em diferentes casas, provocando uma variação acentuada no consumo, pode-se adoptar a solução em que cada casa, antes de realizar a tomada de decisão, espera um período de tempo aleatório inferior ao período de amostragem dos níveis de potência (ou preço). Por exemplo: com a tomada de decisão efectuada de hora a hora (período de amostragem de 1 hora) o sistema aguarda um período aleatório inferior a 10 minutos antes de efectuar a decisão.

## 5 Testes do sistema de despacho de energia

Para testar o sistema de despacho, foram realizados testes colocando o sistema a correr em diferentes cenários. O resultado do sistema foi comparado com uma simulação em MATLAB do algoritmo do sistema.

Também será analisado o custo da energia de uma casa equipada com o sistema de despacho automático, utilizando três tipos de tarifários: preço constante, tri-horário e preço em tempo real (instantâneo).

Os testes ao sistema de despacho foram realizados em duas fases. Na primeira fase foi efectuada a simulação do algoritmo de despacho utilizando o MATLAB. Nessa fase foram efectuadas simulações em 5 cenários distintos, de forma a testar o funcionamento e os resultados do algoritmo de despacho.

Na segunda fase de testes o sistema de despacho da aplicação foi iniciado com os mesmos parâmetros usados na simulação com o objectivo de verificar-se que o sistema se comporta da mesma forma que na simulação. Para tornar a verificação do comportamento mais rápida utilizou-se uma escala temporal diferente. Nesta escala a análise do sistema de despacho era efectuada de forma acelerada em períodos de 10 segundos que em tempo real corresponderiam a 1 hora.

Para os primeiros testes foram introduzidos dois equipamentos no sistema de despacho. Um dos equipamentos apresenta o estado inicial “ligado”, sendo o efeito deste já incluído nos níveis de consumo da rede eléctrica e quando desligado, o consumo deste será removido dos níveis de consumo da rede. O outro equipamento apresenta um estado inicial “desligado”, sendo um equipamento cujo consumo será adicionado aos níveis de consumo quando ligado.

A decisão do sistema de despacho será efectuada para estes dois equipamentos utilizando os parâmetros de configuração de cada equipamento, que nestes testes são considerados idênticos de forma a facilitar a análise dos resultados (Tabela 2), com a excepção do estado inicial.

**Tabela 2: Equipamentos do sistema de despacho**

ID	Endereço do interruptor principal	Tipo	Nível Decisão	Estado Inicial	Potência Potência equipamento * multiplicador
0	224	Aquecedor	25	<i>OFF</i>	6 MW= (2000 W * 3000)
1	216	Aquecedor	25	<i>ON</i>	6 MW= (2000 W * 3000)

As tabelas de valores de potências e temperaturas utilizadas nos testes para os vários cenários são indicadas no Anexo I.

O sistema foi configurado para o máximo peso na energia de forma a se poder analisar o comportamento do sistema para os vários cenários de produção e consumo de potência. A margem de potência utilizada foi de 20 MW. Isto permite que sempre que a diferença de potência seja superior ou igual a 20 MW, o incentivo a activar, ou manter activado, o equipamento seja máximo.

Devido ao facto de um só sistema instalado numa casa não afectar o diagrama de consumo da rede eléctrica, foi aplicado um multiplicador na potência dos equipamentos para que o efeito fosse significativo no diagrama de consumo. Este efeito deverá ser o mesmo que vários sistemas com a mesma configuração instalados em várias casas. Neste caso o multiplicador utilizado corresponde à instalação de 3000 sistemas de despacho em diferentes casas, o que irá introduzir uma variação considerável nos níveis de consumo.

Nos vários cenários considerou-se que os gráficos de produção são os valores óptimos de referência da produção de energia eléctrica para maximizar a eficiência da produção e distribuição da energia eléctrica.

## **5.1 Cenário 1 – Consumo e produção de referência de energia em regime estacionário**

No cenário 1, o diagrama de cargas utilizado foi obtido com base num diagrama de cargas da Empresa de Electricidade da Madeira para a primavera de 2009 [9]. O diagrama da produção de referência foi construído de forma a simular uma situação de produção sem variações muito acentuadas (Figura 46). Estes níveis de potência são compostos por uma componente constante de 100 MW proveniente de uma estação térmica. As variações na potência são introduzidas por estações hídricas, eólicas, e solares.

Ao longo da execução do algoritmo de despacho, na simulação do programa em MATLAB e na prática com o protótipo laboratorial, foi introduzido o efeito da decisão tomada no diagrama de consumo de potência na rede, obtendo-se assim um diagrama de consumo diferente do diagrama de consumo sem a utilização do sistema de despacho (Figura 46).

Em primeiro lugar foi efectuada a simulação do algoritmo do sistema de despacho em MATLAB. Para esse efeito foi indicado o nome do ficheiro *Excel* onde se encontravam os

valores das potências de consumo e produção e temperaturas ao longo de um dia, e os parâmetros de configuração do sistema (número de períodos a analisar, margem de potência, peso da energia, nível de temperatura desejada e margem de temperatura).

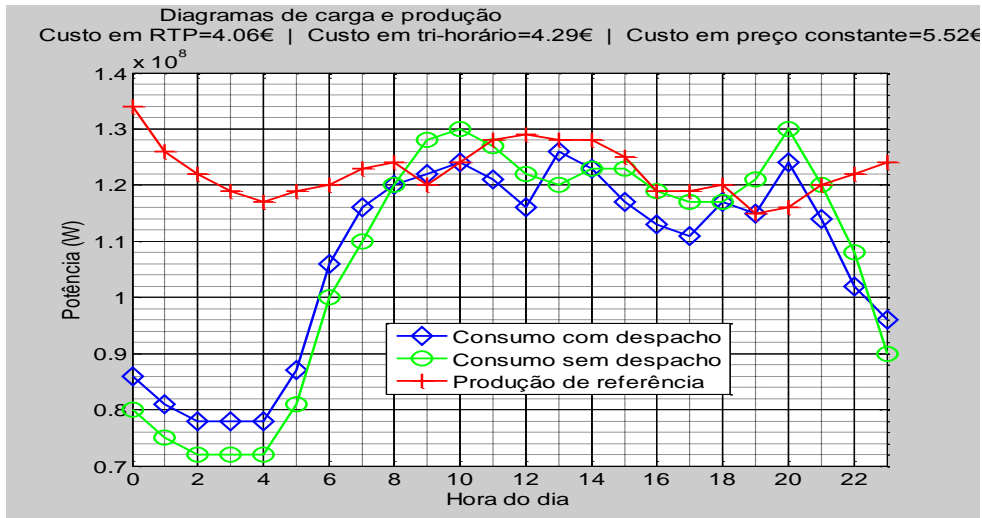


Figura 46: Diagramas de consumo e produção de referência com e sem sistema de despacho, cenário 1

Através do gráfico da Figura 46, pode-se verificar claramente a diferença introduzida no comportamento do consumo pelo sistema de despacho.

Nota-se que por volta das 8 horas o nível de consumo sem despacho ficou muito próximo do nível de produção de referência, o que levou o sistema a decidir desactivar cargas. Às 8 horas foi desactivado o equipamento 0 que reduziu o consumo para um nível inferior ao nível de produção de referência. Às 9 horas foi desactivado o equipamento 1, o que fez reduzir ainda mais o consumo devido à produção de referência ter também sido reduzido (Figura 47). Note-se também que às 13 h foram activados novamente os equipamentos, devido à disponibilidade de energia.

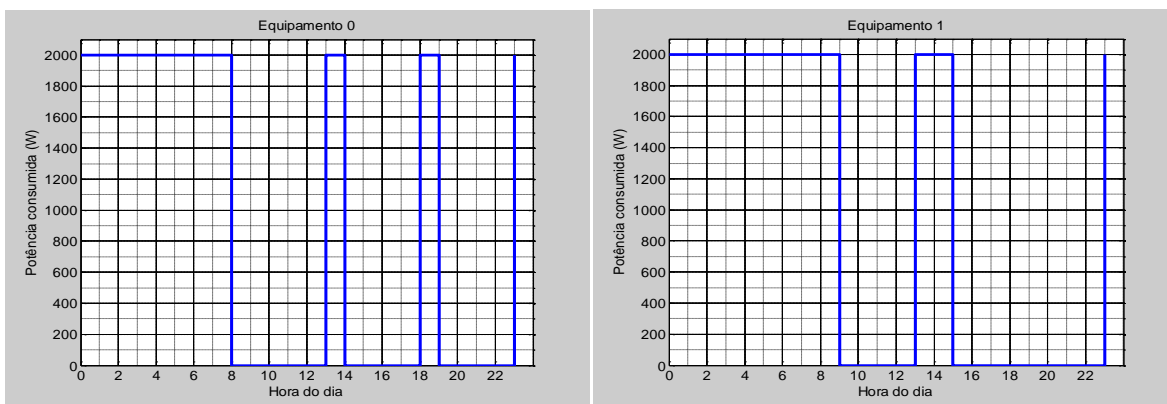


Figura 47: Diagrama de consumo dos equipamentos 0 e 1, cenário 1

As cargas são activadas e desactivadas de forma a manter o consumo abaixo do nível de produção de referência, apesar de existirem algumas alturas em que a quantidade de cargas e potência conjunta controladas não são suficientes (nomeadamente às 9 h e às 20 h, Figura 46).

Às 18 horas o equipamento 0 foi activado, mas o equipamento 1 não foi activado, para não ultrapassar o nível de produção de referência.

Quanto ao custo da energia nota-se claramente que a utilização de um tarifário cujo custo da energia varia em tempo real, dependendo dos níveis de produção de referência e consumo é mais vantajoso. O custo da energia em função dos níveis de produção e consumo é calculado pela seguinte expressão (5.1):

$$\text{Preço da energia} = 0,12\text{€} + \left(\frac{\text{Diff}}{10 \text{ MW}}\right) \times 0,01\text{€} \quad (5.1)$$

O valor 0,12 € representa o custo da energia quando o consumo é igual à produção de referência. Sempre que a diferença é positiva (mais consumo que produção de referência) o custo aumenta em 0,01 € por cada 10 MW de diferença. Com a diferença negativa o custo reduz em 0,01 € por cada 10 MW.

Utilizando a expressão (5.1) as variações dos níveis de consumo e produção podem ser traduzidas para variações do custo, que poderão futuramente ser utilizadas pelo sistema de despacho.

Caso não seja adoptado um tarifário deste género, verifica-se que o tarifário tri-horário também apresenta bons resultados em termos de redução do custo da energia (Figura 46).

Na prática, o protótipo laboratorial (sistema de despacho implementado) apresentou o mesmo comportamento que na simulação, confirmando-se assim, através dos registos apresentados no Anexo J, a correcta implementação do mesmo. Cada vez que o sistema de despacho realizava uma acção era enviada uma mensagem para o utilizador indicando o equipamento e a acção realizada (Figura 48).

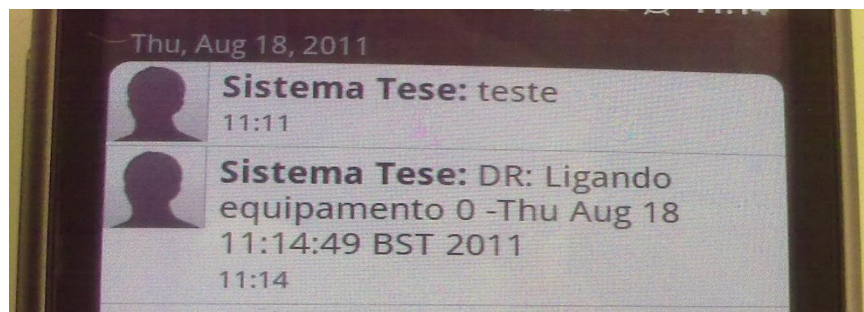


Figura 48: Mensagem enviada pelo sistema de despacho a indicar acção realizada

## 5.2 Cenário 2 – Produção de referência normalmente superior ao consumo de energia

Neste cenário o consumo permanece igual ao cenário 1 e a produção de referência é, na maior parte do tempo, superior ao consumo. Neste cenário o consumo de energia só tende a exceder a produção de referência nas alturas de pico de consumo (Figura 49).

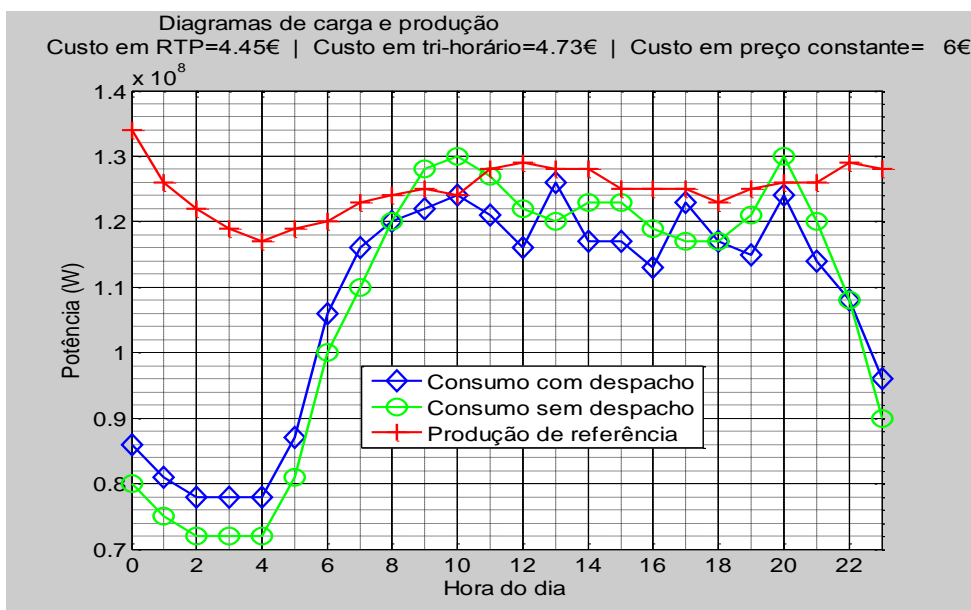


Figura 49: Diagramas de consumo e produção de referência com e sem sistema de despacho, cenário 2

O sistema de despacho em resposta aos níveis de produção e de consumo desactiva os equipamentos 0 e 1 às 8 h e 9 h, respectivamente. Esta acção evita que o consumo ultrapasse a produção de referência. É de realçar que o nível de produção de referência nestes cenários não varia com a variação do consumo como é habitual nas redes eléctricas com gestão tradicional.

Devido ao nível de produção de referência ser mais elevado do que o do cenário 1, nota-se que o consumo em algumas alturas é superior devido à maior disponibilidade de energia eléctrica.

Este aumento de consumo implica um custo adicional para o consumidor, mas para a rede eléctrica representa uma maior eficiência, graças à redução da energia desperdiçada (Figura 50).

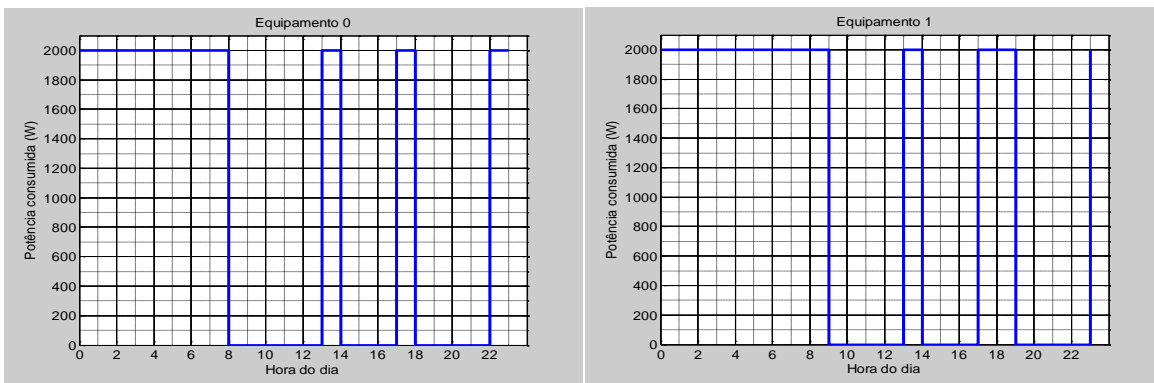


Figura 50: Diagrama de consumo dos equipamentos 0 e 1, cenário 2

Tal como no cenário 1, o sistema com o protótipo laboratorial, apresentou o mesmo comportamento na activação e desactivação dos equipamentos, tal como é apresentado no anexo I

### 5.3 Cenário 3 – Produção de referência normalmente inferior ao consumo de energia

No cenário 3 é representada a situação inversa do cenário 2, mantendo-se o consumo igual ao cenário 1, mas a produção de referência é neste caso, e durante a maior parte do tempo, inferior ao consumo. Isto deve-se à redução da componente de potência constante, introduzida pelas estações térmicas, de 100 MW para 90 MW (Figura 51).

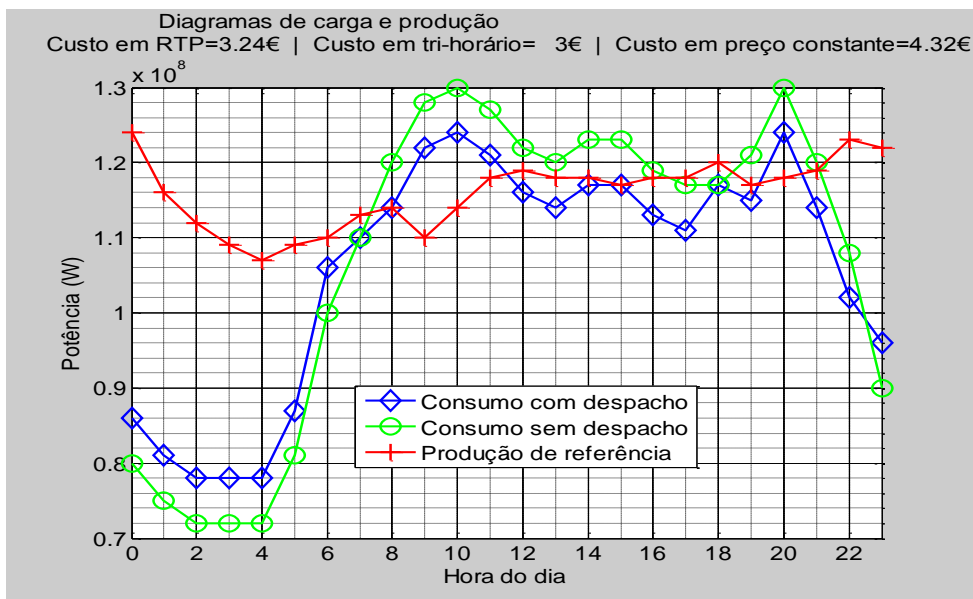


Figura 51: Diagramas de consumo e produção com e sem sistema de despacho, cenário 3

Nota-se, na Figura 51, que em algumas zonas o consumo ultrapassa a produção de referência, devido ao facto do consumo ser tão elevado que nem a desactivação dos equipamentos controlados pelo sistema não reduz suficientemente o consumo (Figura 52).

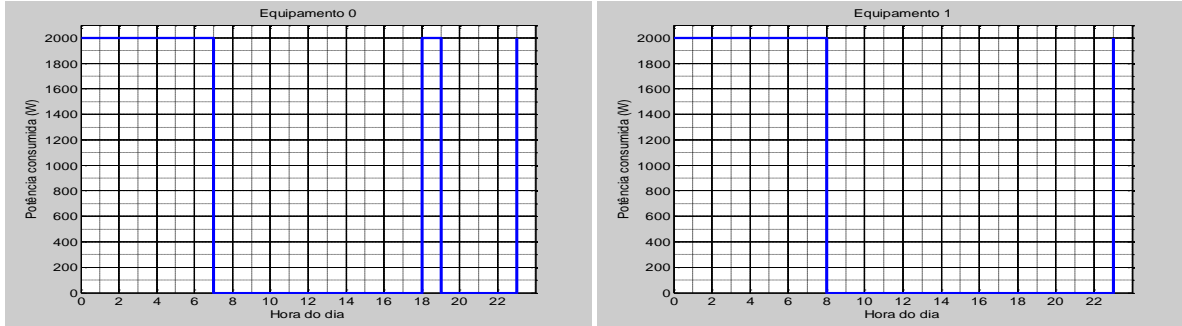


Figura 52: Diagrama de consumo dos equipamentos 0 e 1, cenário 3

Como a produção de referência, neste cenário, é mais reduzida, os equipamentos permanecem mais tempo desligados devido à falta de oportunidades para os activar. Naturalmente, o custo da energia é mais reduzido e contribui, embora não o suficiente, para a fiabilidade da rede eléctrica.

Para reduzir ainda mais o consumo será necessária da integração de mais equipamentos no sistema de despacho ou a instalação do sistema em mais casas.

#### 5.4 Cenário 4 – Aumento e diminuição rápida do consumo

O cenário 4 baseou-se no diagrama de consumo do cenário 1, no qual se introduziu às 13 h um pico de consumo elevado entre as 17 h e as 18 h um nível de consumo reduzido (Figura 53). Este cenário permitiu analisar a resposta do sistema a variações repentinas dos níveis de consumo.

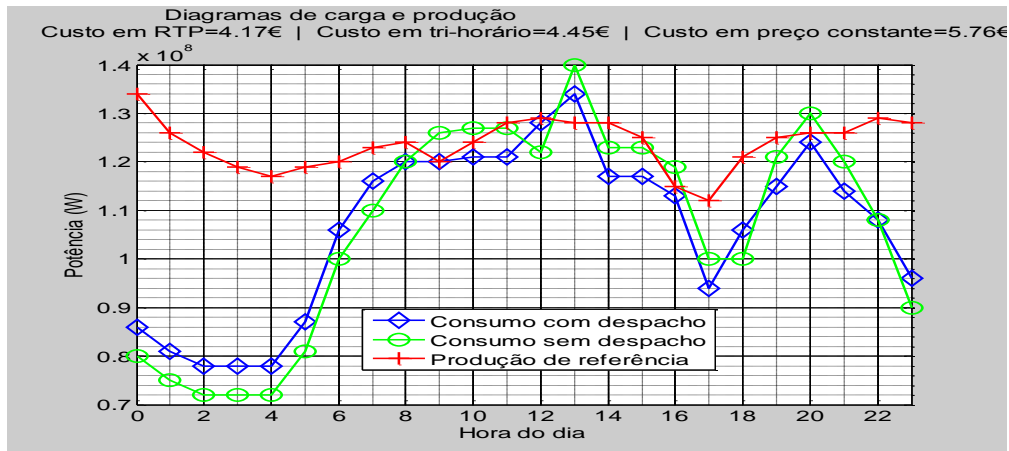


Figura 53: Diagramas de consumo e produção com e sem sistema de despacho, cenário 4

Pode-se verificar que o sistema reagiu às variações do consumo de energia. Às 13 horas os equipamentos 0 e 1 foram desactivados (Figura 54) de forma a reduzir o consumo, embora não tenha sido o suficiente para garantir que o consumo não ultrapassasse a produção de referência.

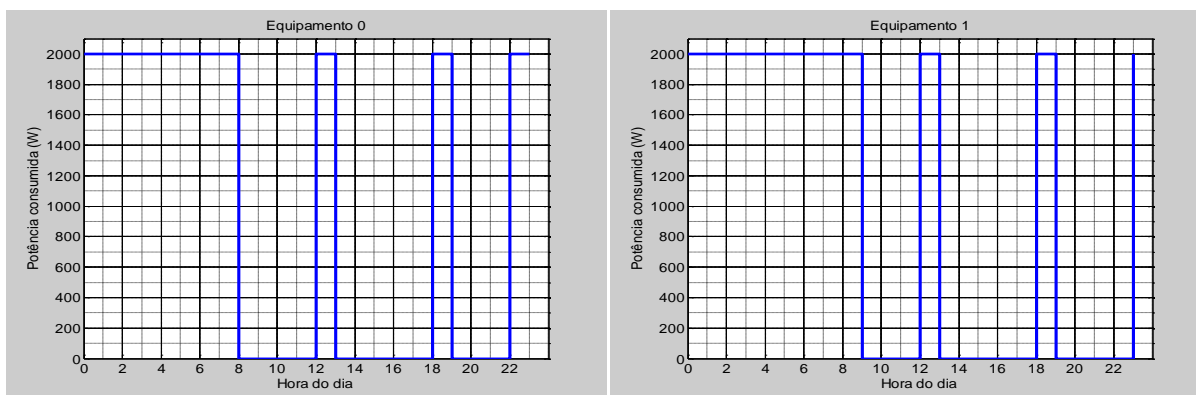


Figura 54: Diagrama de consumo dos equipamentos 0 e 1, cenário 4

Às 17 h o consumo é consideravelmente inferior à produção, mas apesar disso o sistema não activou os equipamentos 0 e 1 devido ao facto de às 18 h o incentivo para activar os equipamentos ser maior, sendo a activação destes equipamentos adiada para o período seguinte.

## 5.5 Cenário 5 – Redução rápida da produção de energia

À semelhança do cenário 4, introduziu-se neste cenário um baixo nível de potência no diagrama de produção de referência (Figura 55). Neste cenário a variação é introduzida no lado da produção onde é mais frequente ocorrer uma situação de grande variação (Por exemplo: devido à obstrução de um parque fotovoltaico com nuvens).

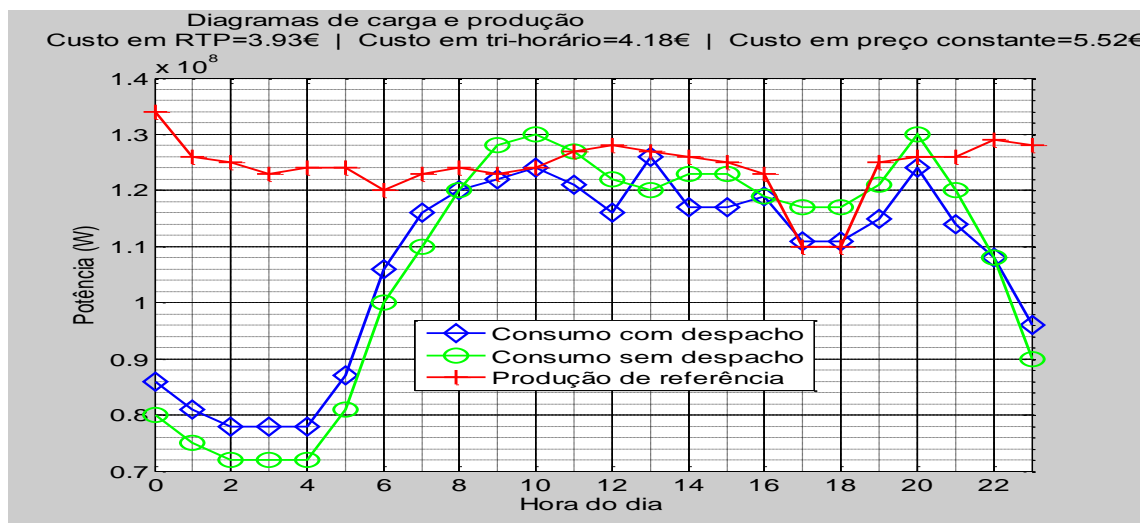


Figura 55: Diagramas de consumo e produção com e sem sistema de despacho, cenário 5

Nas Figura 55 e Figura 56 pode-se verificar que quando ocorre a redução acentuada da produção o equipamento 0, que se encontrava ligado, foi desactivado de forma a reduzir o consumo de energia, para o consumo conseguir acompanhar a produção. O equipamento 1, que já se encontra desactivado no momento da redução de produção, permaneceu desligado.

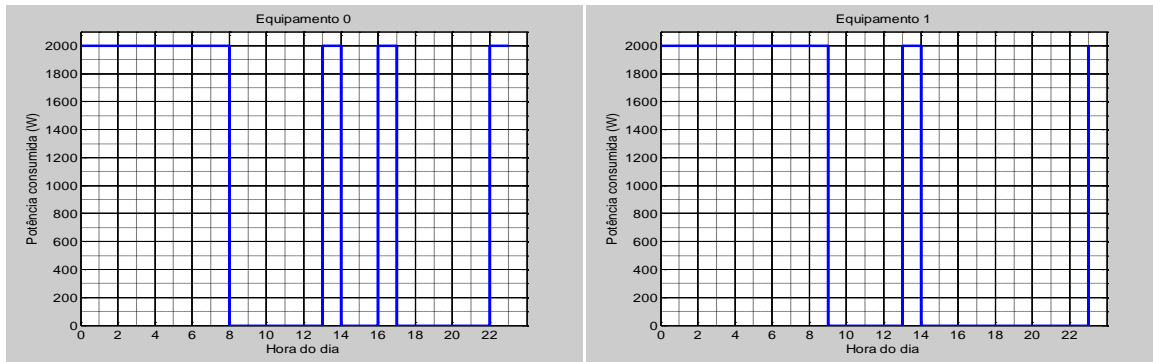


Figura 56: Diagrama de consumo dos equipamentos 0 e 1, cenário 5

## 5.6 Simulação do cenário 1 com variação do peso da energia

Pretende-se mostrar, neste cenário, a influência do peso da energia no comportamento do sistema de despacho. Os níveis de consumo e de produção de referência são iguais aos do cenário 1.

A margem de potência utilizada para esta simulação foi de 50 MW, de forma a impedir que o factor de energia não apresente o seu valor máximo muito frequentemente.

Os resultados são analisados em função do custo da energia consumida para cada variação do peso da energia. A variação do custo de energia depende da forma como as cargas são activadas e desactivadas ao longo do dia.



Figura 57: Gráfico da variação do custo de energia para diferentes valores do peso da energia

Do gráfico da Figura 57 pode-se verificar que o custo da energia, quando o peso é nulo, é superior ao custo da energia quando o peso é máximo. No entanto, o menor custo é obtido quando o peso de energia é de 60%. Isto deve-se ao factor de conforto utilizado, que impede que os equipamentos sejam activados sem necessidade. O factor de conforto depende do valor da temperatura ambiente (Figura 58).

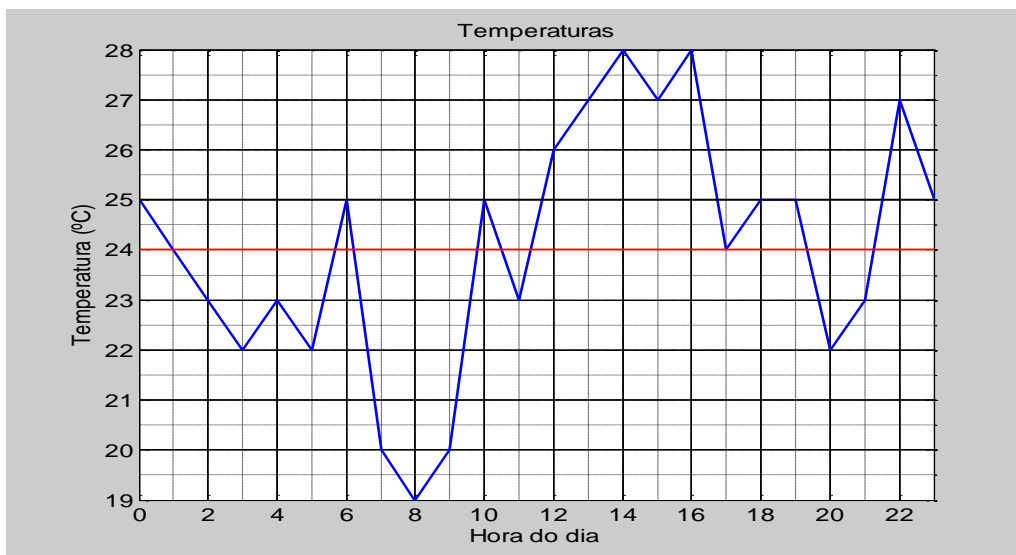


Figura 58: Gráfico da variação da temperatura ao longo do dia utilizado pelo sistema

É de realçar que a variação da temperatura ao longo de um dia não corresponde à variação real. Os valores foram escolhidos de forma a provocar situações que demonstrem o comportamento do sistema de despacho.

Através do gráfico da Figura 57, pode se verificar que o factor de energia começa a influenciar o comportamento quando o peso da energia é superior a 40%.

À primeira vista o custo deveria reduzir-se com o aumento do peso da energia, mas o que acontece é que o custo decresce até aos 60%, mas depois volta a aumentar. O aumento deve-se à activação de equipamentos quando estes não são necessários. A necessidade do equipamento é imposta pelo factor de conforto e quando se reduz a influência deste na decisão o custo tende a aumentar pois o sistema activa cargas baseando-se mais no factor da energia.

No caso específico do peso da energia ser 60% pode-se verificar que em algumas situações em que o equipamento deveria ser activado, em termos de energia, o factor de conforto impede a activação (Figura 59).

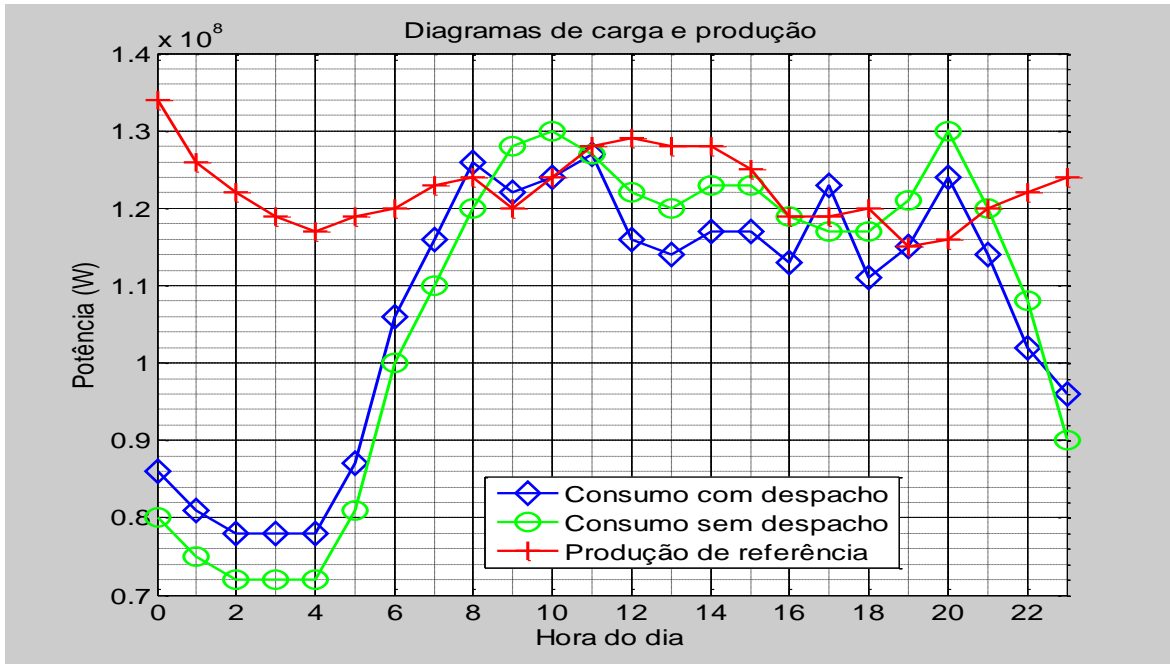


Figura 59: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com peso de energia de 60%

Devido à influência do factor de conforto observou-se uma aparente anomalia às 17 h, quando ambos os equipamentos foram activados (quando somente o equipamento 0 deveria ligar), fazendo com que o consumo ultrapassasse a produção de referência para garantir os critérios de conforto (Figura 60).

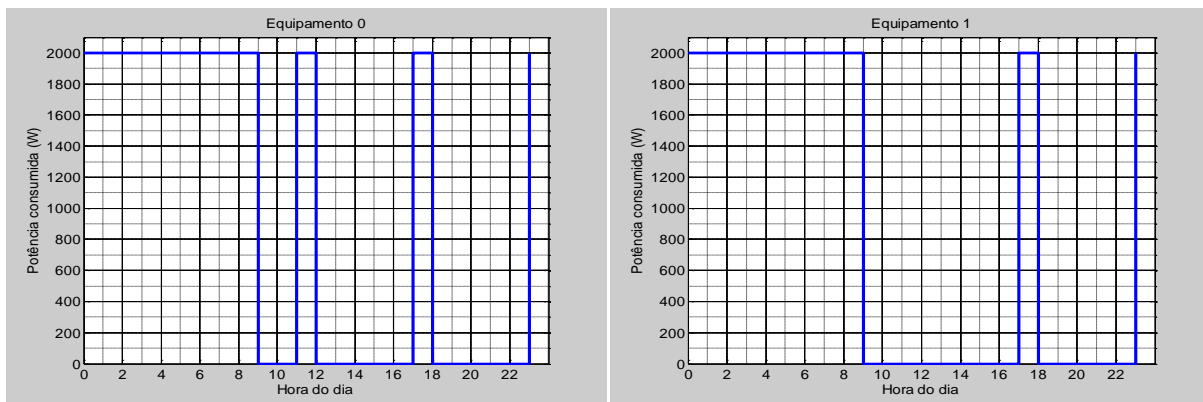


Figura 60: Diagrama de consumo dos equipamentos 0 e 1, cenário 1 com peso de energia de 60%

Às 17 h a temperatura encontrava-se no limite abaixo da temperatura definida pelo utilizador e o factor de conforto apresentava o melhor incentivo para activar o equipamento. A aparente anomalia, referida anteriormente, ocorre porque a análise efectuada às 17 h aos 4 períodos (17 h, 18 h, 19 h e 20 h) resultou nos incentivos de energia devolverem um valor nulo para a decisão de manter ou mudar o estado dos equipamentos, tendo o mesmo efeito que um

peso de energia nulo. Os incentivos calculados deram o valor nulo porque um dos períodos seguintes apresentava melhor incentivo. Se o sistema for configurado somente para considerar um período (período actual) a aparente anomalia deixa de existir, como é possível ver no gráfico da Figura 61.

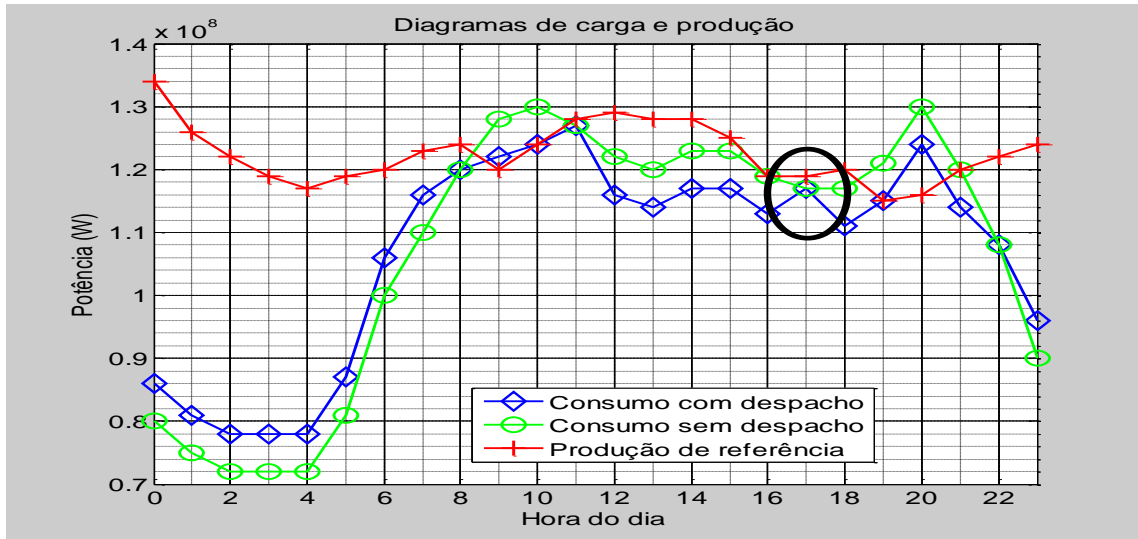


Figura 61: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com peso de energia de 60% analisando somente o período actual

O factor de conforto apresenta maior efeito no período entre as 12 h e 15 h onde, apesar de a produção ser consideravelmente superior ao consumo, os equipamentos permanecem desligados pelo facto da temperatura nesse período ser superior ao nível definido pelo utilizador. É de realçar que o equipamento 0 desligou-se às 12 h exactamente pela razão indicada anteriormente.

Utilizando-se um peso de 75% para a energia notou-se que o equipamento 0 se activou durante o período das 12 h às 15 h, devido à menor influência do factor de conforto na decisão final (Figura 62).

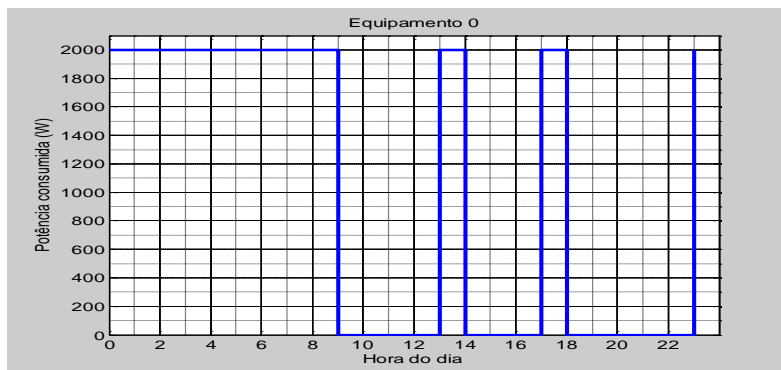


Figura 62: Diagrama de consumo do equipamento 0, cenário 1 com peso de energia de 75%

## 5.7 Simulação do cenário 1 com 6 equipamentos

Nos cenários anteriores foram somente utilizados dois equipamentos, o que limita a flexibilidade do sistema para responder às variações do consumo e da produção. De forma a aumentar a flexibilidade, neste cenário utilizaram-se 6 equipamentos, através da introdução de mais dois pares de equipamentos com as mesmas características que os equipamentos 0 e 1, indicados inicialmente na Tabela 2 (na página 85). O sistema foi analisado com os mesmos parâmetros do primeiro teste efectuado no cenário 1 (Figura 63).

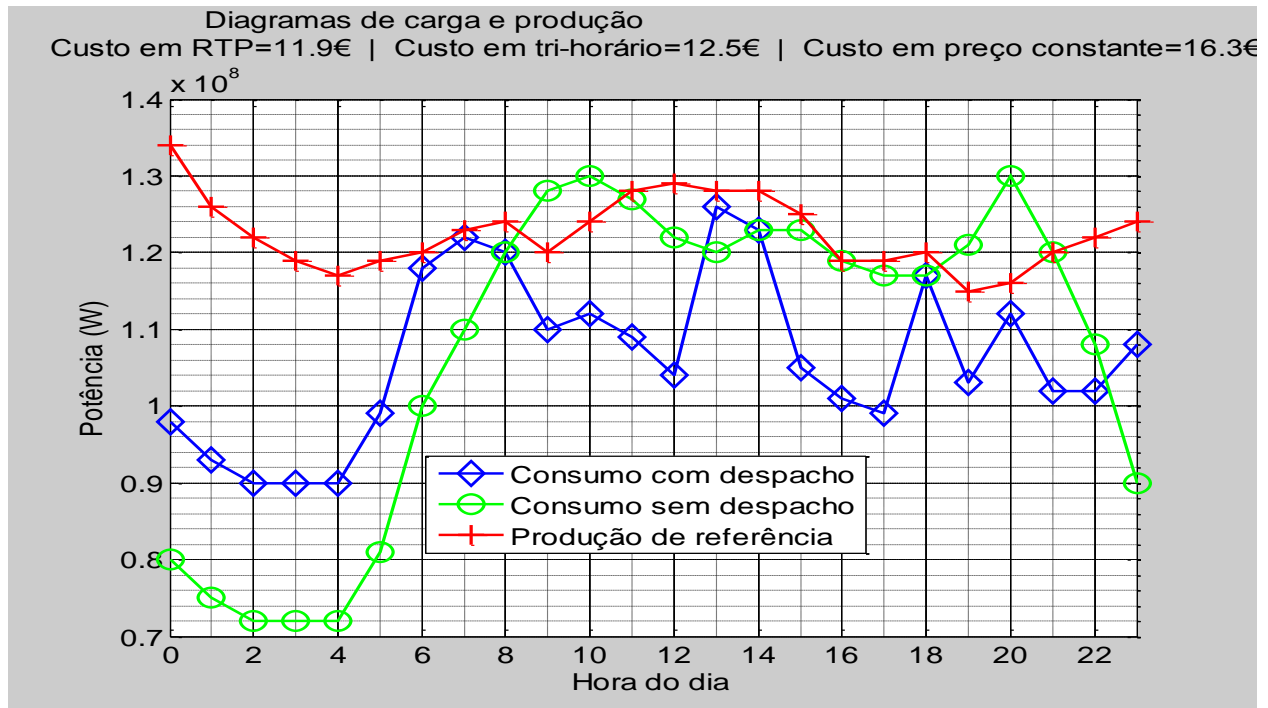


Figura 63: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com 6 equipamentos com margem de 20 MW

Nota-se que a introdução de mais equipamentos no sistema de despacho introduziu uma maior flexibilidade na resposta aos níveis de consumo e produção.

Em seguida ensaiou-se a redução da margem de potência, o que se espera que faça com que o factor de energia apresente o valor máximo de incentivo com mais frequência e permita que o nível de consumo seja mais próximo do nível de produção de referência (Figura 64).

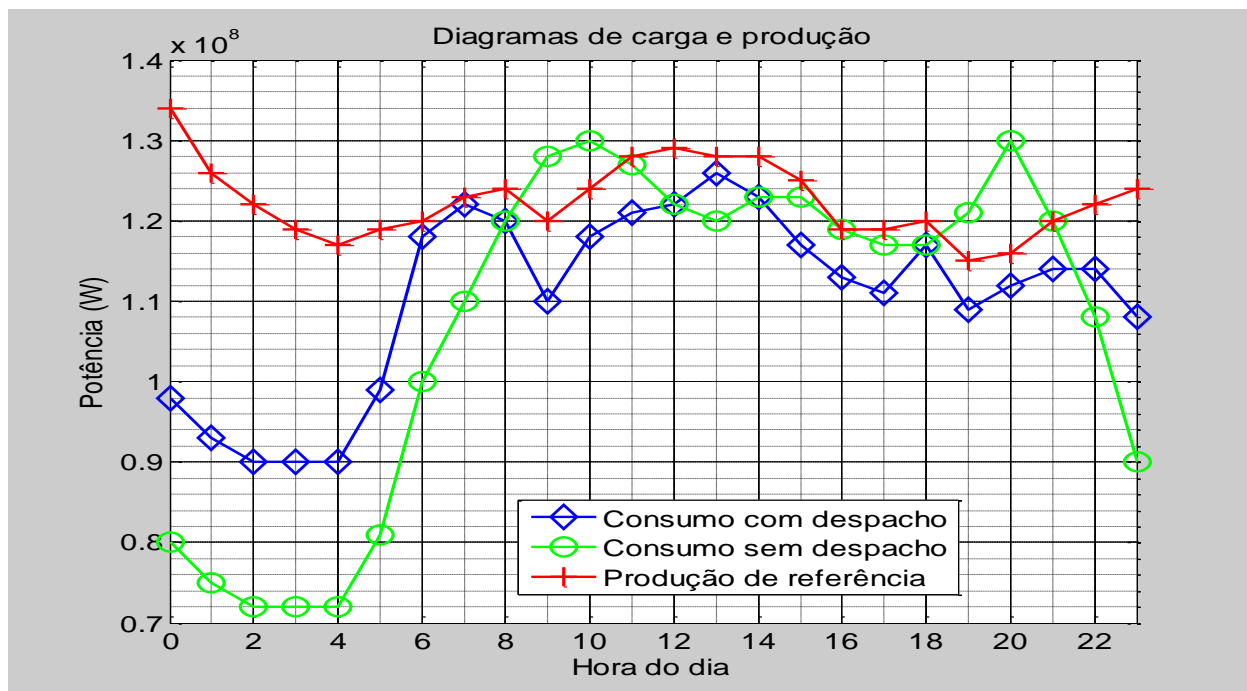


Figura 64: Diagramas de consumo e produção com e sem sistema de despacho, cenário 1 com 6 equipamentos com margem de 5 MW

De facto verificou-se que com a redução da margem de potência o sistema acompanha mais fielmente os níveis de produção de energia, sem os ultrapassar.

Do cenário 1 ao cenário 5 notou-se que o sistema implementado, no protótipo laboratorial, apresentou o mesmo comportamento que a simulação. A simulação permite mais facilmente e rapidamente analisar o comportamento do sistema de automação, podendo ser utilizada para realizar os ajustes de controlo antes de se implementar e configurar o sistema real de gestão distribuída de energia eléctrica.

O sistema conseguiu reagir às rápidas variações introduzidas nos cenários 4 e 5, melhorando a eficiência e a fiabilidade da rede eléctrica.

Nos vários cenários estudados o sistema afectou o consumo de forma positiva, melhorando o desempenho da rede eléctrica. Em nenhum dos cenários a produção de referência foi alterada para satisfazer os níveis de consumo.

A introdução de um maior número de equipamentos no sistema proporciona uma maior flexibilidade na decisão, e o efeito de gestão na rede de energia eléctrica é mais significativo. Isto mostra que a instalação do sistema em utilizadores com hábitos de consumo de energia variáveis, como é o caso dos consumidores domésticos, melhora a gestão da energia da rede com este sistema distribuído de controlo em tempo real.

## 6 Conclusões e trabalhos futuros

### 6.1 Conclusões

O problema do equilíbrio energético está a tornar-se mais difícil de resolver devido ao crescimento da população mundial que implica um aumento do consumo de energia eléctrica. Com uma maior dificuldade de obtenção de combustíveis fósseis, as fontes primárias de energia eléctrica começam a tornar-se mais dispendiosas e o incentivo à utilização de fontes de energia renováveis é cada vez maior.

A utilização de geradores baseados em energia renováveis, como a eólica e a solar, pode introduzir grandes variações nos níveis de produção. Estas variações são difíceis de prever por dependerem essencialmente de factores meteorológicos.

O sistema de despacho automático, implementado nesta dissertação, contribui para a estabilidade e eficiência da rede eléctrica, no lado do consumo, através do controlo de cargas.

Nas alturas em que a produção de referência da energia eléctrica é superior ao consumo o sistema aconselha a activação das cargas, para consumir a energia que está a ser produzida em excesso, evitando assim desperdício de energia (melhora a eficiência).

Quando o nível de consumo é inferior, mas próximo do nível de produção, o sistema aconselha a desactivação das cargas reduzindo assim o consumo energético (melhor estabilidade).

Como o sistema de despacho poderá aconselhar acções que influenciam o conforto, foi necessário introduzir, nas equações de decisão, parâmetros para que a decisão de despacho dependa dos critérios de conforto definidos pelo consumidor de energia eléctrica.

Quanto maior o número de cargas controladas pelo sistema de despacho melhor será o desempenho do mesmo. Isto ocorre pois há uma melhoria da flexibilidade do sistema de gestão de energia eléctrica para responder ao estado da rede eléctrica e respeitar os critérios de conforto dos consumidores de energia.

Naturalmente, a instalação de um só sistema de despacho não é suficiente para afectar o consumo de forma a se obter o comportamento esperado. Quanto maior for o número de sistemas de despacho implementados, maior será o efeito introduzido no consumo, e mais o comportamento se aproximará do pretendido.

O sistema de automação apresenta as funcionalidades principais de controlo de cargas, monitorização e processamento de informação e comunicação com o exterior (Por exemplo com a Internet), sendo esta uma plataforma adequada para a implementação de um sistema de despacho distribuído de energia eléctrica.

O protótipo foi implementado como uma aplicação a correr sobre um sistema de automação, o que implica que pode ser implementado para correr sobre outros sistemas de automação já existentes no mercado.

A implementação de um sistema de automação através da integração de vários protocolos permitiu uma grande flexibilidade na escolha dos equipamentos, de forma a garantir um baixo custo e aumentar a facilidade de instalação do sistema. A interligação dos protocolos, ao ser efectuada através de uma *gateway*, facilitou a implementação da aplicação, visto esta comunicar, com uma rede com vários protocolos, através de um protocolo comum (abstracção dos protocolos).

Para actuação nas cargas foram utilizados equipamentos do protocolo X10, que apresentam baixo custo e facilidade de instalação, visto utilizarem a rede eléctrica para comunicarem. Em comparação com um equipamento KNX para comutação de duas lâmpadas, o custo dos equipamentos X10 para o mesmo efeito, apresentam uma redução de aproximadamente 45%.

A utilização de equipamento com protocolo *ZigBee* permite a fácil construção de uma rede de sensores e actuadores numa residência sem o custo adicional de passagem de cabos, devido ao facto da comunicação ser efectuada por rádio frequência. O protocolo GSM permite a fácil interacção entre o sistema e o utilizador através do envio de mensagens SMS.

As características do sistema implementado tornam este mais atractivo para instalação em casas já construídas sem sistema de automação, que constituem a maior percentagem das casas existentes.

No desenvolvimento da *gateway* optou-se pela implementação da conversão de protocolo através de uma forma indirecta, com a utilização de um protocolo comum. Esta solução permitiu o desenvolvimento mais facilitado de algoritmos de conversão, sendo apenas necessário implementar 2 algoritmos por cada protocolo adicional. Esta estratégia facilita bastante a expansão das capacidades da *gateway*, para a integração de novos protocolos, sem a necessidade de desenvolvimento de muitos algoritmos de conversão.

A *gateway* implementada pode ser utilizada mesmo sem esta se encontrar ligada a um computador, reduzindo ainda mais o custo do sistema. Depois de configuradas as tabelas de equipamentos e de rotas, esta funciona de forma independente, encaminhando pacotes entre os vários protocolos.

## 6.2 Trabalhos futuros

A implementação prática permitiu verificar experimentalmente o sistema proposto. O projecto foi feito de forma genérica com vista a uma futura expansão e melhoria.

Futuramente, são propostas novas ideias e melhoramentos a efectuar ao protótipo implementado e ao sistema de automação proposto, nomeadamente:

- Expansão da capacidade de processamento e memória, através da utilização de um sistema operativo embutido na *gateway*, de forma a expandir as capacidades de processamento;
- De forma a expandir consideravelmente a flexibilidade de implementação de um sistema de automação com uso da *gateway*, é proposta a integração de novos protocolos como bluetooth, KNX, WiFi, Ethernet e IDOM;
- Adicionar a possibilidade de serem utilizadas mais do que uma *gateway* no sistema de automação, sendo estas capazes de comunicar entre si para troca de informações de encaminhamento e de dispositivos registados;
- Modificação do sistema de automação proposto, que utiliza uma arquitectura centralizada para gerir a energia no interior de um edifício, para um sistema de automação distribuído, utilizando sensores e actuadores com capacidade de decisão.
- Melhoria do sistema de despacho implementado através da adição de novos factores que influenciam as decisões de gestão de energia. Estudar mais aprofundadamente os factores que condicionam o conforto do utilizador e que optimizam a redução do consumo de energia;
- Adicionar ao sistema de despacho equipamentos de geração e armazenamento de energia eléctrica, para contribuir para a estabilidade da rede eléctrica.

## Referências

- [1] Peter Palensky, Dietmar Deitrich, “Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads”, IEEE Transactions on Industrial Informatics, Vol. 7, No.3, Agosto 2011.
- [2] Lucas E. P. Mizusaki, “Comparação de Mecanismos de Comunicação para a Casa Inteligente”, Tese de Mestrado, Dep. Informática, Univ. Federal do Rio Grande do sul, Porto Alegre, Brasil, 2009.
- [3] Diana S. C. Palma, “FEUP KNX Domótica KNX/EIB de Baixo Custo”, Tese de Mestrado, Dep. Eng. Electrotécnica e de Computadores, Univ. Porto, Porto, Portugal, 2008.
- [4] “Introduction to the LonWorks Platform”, rev. 2, Echelon Corporation, San José, Califórnia, EUA, 2009.
- [5] Joung-Han Lee, In-Ho Choi, Anwar-Ul-Haq, Seung-Ho Hong, “Design of Demand Response Module for Smart Grid”, IEEE 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Setembro 2010.
- [6] Hanne Saele, Ove S. Grande, “Demand Response From Household Customers: Experiences From a Pilot Study in Norway”, IEEE Transactions on Smart Grid, Vol.2, No 1, Março 2011.
- [7] Godfrey Boyle, Bob Everett, Janet Ramage, “Introductory Overview” in Energy Systems and Sustainability, Oxford, Oxford University press, 2003, ch. 1, pp 3-54.
- [8] Godfrey Boyle, Bob Everett, Janet Ramage, “Primary Energy” in Energy Systems and Sustainability, Oxford, Oxford University press, 2003, ch. 2, pp 57-89.
- [9] Caracterização da Rede de Transporte e Distribuição em AT e MT, Direcção de Estudos e planeamento, Empresa de Electricidade da Madeira, Funchal, Portugal, Março 2011.
- [10] Caracterização da Rede de Transporte e Distribuição em AT e MT, Direcção de Estudos e planeamento, Empresa de Electricidade da Madeira, Funchal, Portugal, Março 2010.
- [11] Godfrey Boyle, “Variable Renewables and the Grid: An Overview” in Renewable Energy and the Grid, London, Earthscan, 2007, ch. 1, pp 1-27.
- [12] Pouyan Khajavi, Hassan Monsef, Hassan Abniki, “Load Profile Reformation through Demand Response Programs Using Smart Grid”, IEEE Modern Electric Power Systems 2010, Wroclaw, Polónia, 2010.
- [13] Paulo Oliveira, Modulo Domótica [Online]. Disponível em: <http://pt.scribd.com/doc/39244263/Sebenta-Domotica-Free#logout>.

- [14] José M. S. Pinheiro (2004, Junho 13). Sistemas de automação [Online]. Disponível em: [http://www.projetederedes.com.br/artigos/artigo\\_sistemas\\_automacao.php](http://www.projetederedes.com.br/artigos/artigo_sistemas_automacao.php).
- [15] Renato J. C. Nunes, “Análise Comparativa de tecnologias para Domótica”, INESC-ID / Instituto Superior Técnico, Lisboa, 2002.
- [16] Luís Silva, “Automação em Ambientes Residenciais”, Tese Mestrado, Dep. Eng. Mecânica, Univ. Aveiro, Aveiro, Portugal, 2008.
- [17] N.D. (N.D.), X10 Powerline Carrier (PLC) Technology [Online]. Disponível em: <http://www.x10.com/support/technology1.htm>.
- [18] KNX System Specifications/Architecture, v3.0, KNX Association, Bruxelas, Bélgica, 2009.
- [19] Yu-Chee Tseng, Meng-Shiuan Pan, Chih-Wei Yi, "Wireless sensor networks," in AccessScience, ©McGraw-Hill Companies, 2008, <http://www.accessscience.com>.
- [20] Ferdinando Monsignore, “Sensoriamento de Ambiente utilizando o padrão ZigBee”, Dissertação Mestrado, N.D, Univ. São Paulo, São Paulo, Brasil, 2007.
- [21] LonTalk Protocol, N.D., Toshiba, N.D, N.D., N.D, pag. 121-126.
- [22] Simon S. LAM, “Protocol Conversion”, IEEE Trans. Software Engineering, vol.14, NO 3, Março 1988.
- [23] Gregor V. Bochmann, Pierre Mondain-Monval, “Design Principles for Communication Gateways”, IEEE journal on selected areas in communications. VOL 8. NO 1, Janeiro 1990.
- [24] Anan Phonphoen, “Networking and Internetworking Devices”, Apresentação doutoramento, Dep. Engenharia de computadores, universidade de Kasetsart, Bangkok, Thailand.
- [25] CIRRONET, “ZG-2400E ZigBee Ethernet Gateway” [PDF] Disponível em: [http://www.cirronet.com/pdf/brochure\\_zg2400e.pdf](http://www.cirronet.com/pdf/brochure_zg2400e.pdf).
- [26] Product brief Harmony Multi-Protocol HAN Gateway, SimpleHomeNet, San Juan Capistrano, Califórnia, EUA, 2009.
- [27] Sergio Rogai, “Telegestore Project: Progress & Results”, IEEE conference Power Line Communication and its applications keynote, Março 2007.
- [28] N.D., InovCity [Online]. Disponível em: <http://www.inovcity.pt/pt/Pages/homepage.aspx>.
- [29] Peter B. Luh, Laurent D. Michel, Peter Friedland, Che Guan, Yuting Wang, “Load Forecasting and Demand Response”, IEEE Power and Energy society Meeting 2010, Setembro 2010.

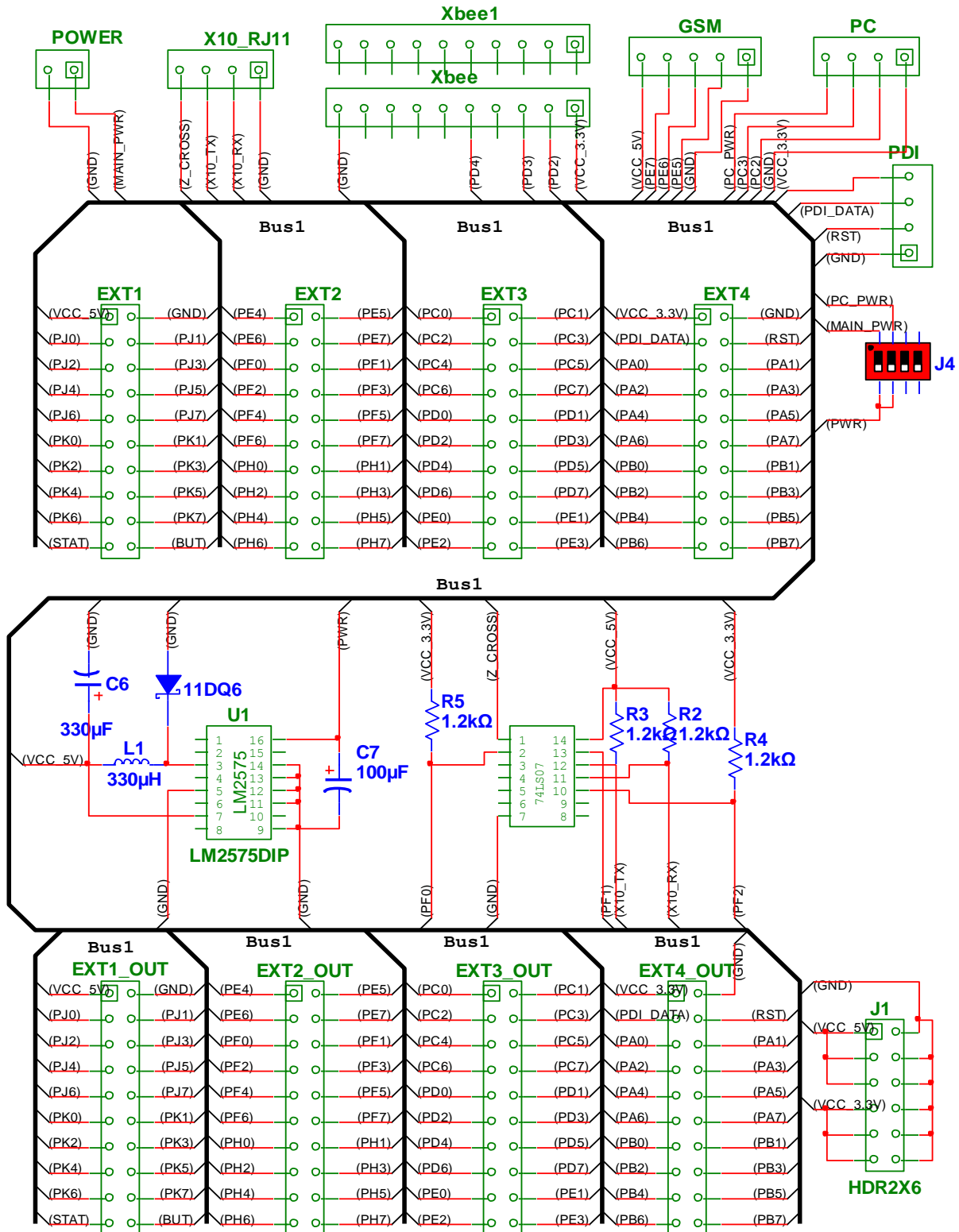
- [30] Salman Mohagheghi, James Stoupis, Zhenyan Wang, Zhao Li, “Demand Response Architecture”, ABB corporate Research Center, Raleigh, NC, EUA, 2010.
- [31] Abigail Anthony, Demand Response [PDF] Disponível em: [http://www.ci.uri.edu/ciip/Publications/Anthony\\_DR%20Overview.pdf](http://www.ci.uri.edu/ciip/Publications/Anthony_DR%20Overview.pdf).
- [32] Steve Nguyen, “Automatic Demand Response: Driving Energy Efficiency for Global climate change Using LonWorks controlo NetWorks”, Echelon Corporation, San Jose, Califórnia, EUA, 2006.
- [33] N.D., Venda de equipamentos X10, [Online] Disponível em: [www.eurox10.com](http://www.eurox10.com).
- [34] Farnell, Venda de componentes electrónicos, [Online] Disponível em: [www.pt.farnell.com](http://www.pt.farnell.com).
- [35] OLIMEX, Venda de placas de desenvolvimento e protótipos; [www.olimex.com](http://www.olimex.com).
- [36] XBee series 2 OEM RF Modules, ver.1.x.2x, MaxStream, Lindon, EUA, 2007.
- [37] MM232R Mini USB-Serial UART Development Module, ver 0.91, FTDI International, Glasgow, Reino Unido, 2005.
- [38] Atmel, AVR1502 [ONLINE] Disponível em: [http://www.atmel.com/dyn/resources/prod\\_documents/AVR1502.zip](http://www.atmel.com/dyn/resources/prod_documents/AVR1502.zip).
- [39] Tiago Braga, “Monitorização Ambiental em Espaços Florestais com rede de sensores sem fio”, Tese de Mestrado, Dep. Matemática e Engenharias, Univ. da Madeira, Funchal, 2010.
- [40] NetBeans, NetBeans 7.0, [Online] Disponível em: [www.netbeans.org](http://www.netbeans.org).
- [41] Apache Friend, Xampp, [Online] Disponível em: [www.apachefriends.org/pt\\_br/xampp.html](http://www.apachefriends.org/pt_br/xampp.html).
- [42] Oracle (N.D.), JAVA Remote Method Invocation – Destributed Computing for JAVA [Online]. Disponível em: <http://www.oracle.com/technetwork/JAVA/JAVase/tech/index-jsp-138781.html>.
- [43] Scott McPherson, RMI [Online]. Disponível em: <http://JAVA.sun.com/developer/technicalArticles/RMI/rmi/>.

## Anexo A: Tabela de códigos X10

	HOUSE CODES					KEY CODES				
	H1	H2	H4	H8		D1	D2	D4	D8	D16
A	0	1	1	0	1	0	1	1	0	0
B	1	1	1	0	2	1	1	1	0	0
C	0	0	1	0	3	0	0	1	0	0
D	1	0	1	0	4	1	0	1	0	0
E	0	0	0	1	5	0	0	0	1	0
F	1	0	0	1	6	1	0	0	1	0
G	0	1	0	1	7	0	1	0	1	0
H	1	1	0	1	8	1	1	0	1	0
I	0	1	1	1	9	0	1	1	1	0
J	1	1	1	1	10	1	1	1	1	0
K	0	0	1	1	11	0	0	1	1	0
L	1	0	1	1	12	1	0	1	1	0
M	0	0	0	0	13	0	0	0	0	0
N	1	0	0	0	14	1	0	0	0	0
O	0	1	0	0	15	0	1	0	0	0
P	1	1	0	0	16	1	1	0	0	0
			All Units Off		0	0	0	0	0	1
			All Lights On		0	0	0	1	1	1
			On		0	0	1	0	1	1
			Off		0	0	1	1	1	1
			Dim		0	1	0	0	1	1
			Bright		0	1	0	1	1	1
			All Lights Off		0	1	1	0	1	1
			Extended Code		0	1	1	1	1	1
			Hail Request		1	0	0	0	1	①
			Hail Acknowledge		1	0	0	1	1	1
			Pre-Set Dim		1	0	1	X	1	②
			Extended Data (analog)		1	1	0	0	1	③
			Status-on		1	1	0	1	1	1
			Status-off		1	1	1	0	1	1
			Status Request		1	1	1	1	1	1

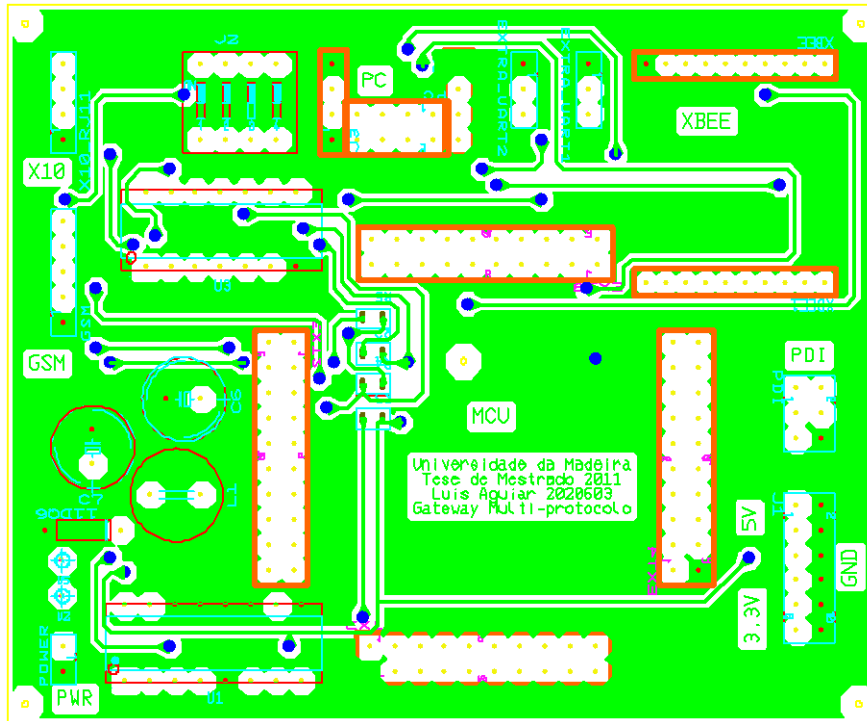
FIGURE 4

## Anexo B: Esquema de ligações da gateway

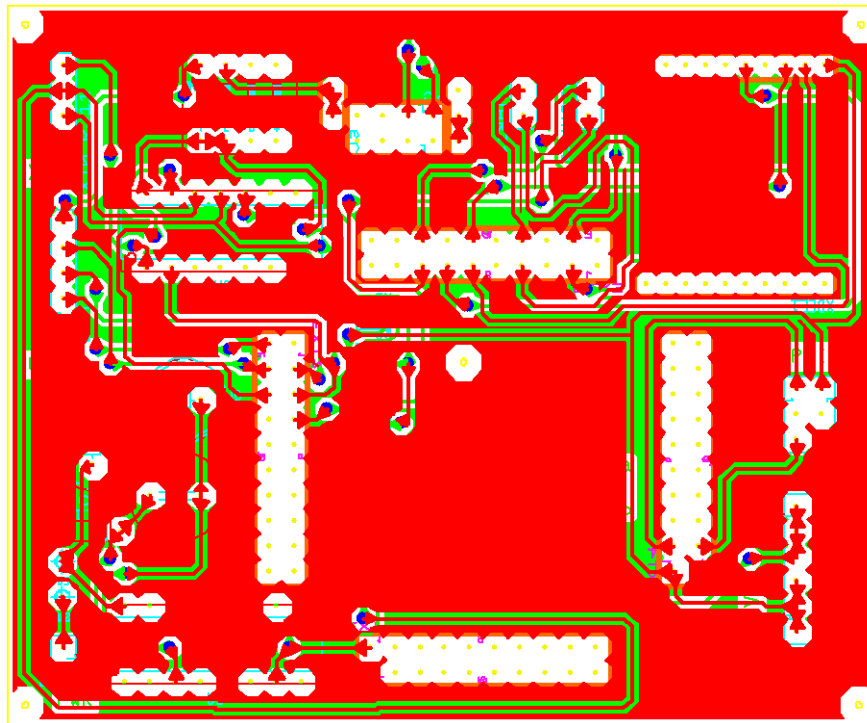


## Anexo C: Desenho PCB da gateway

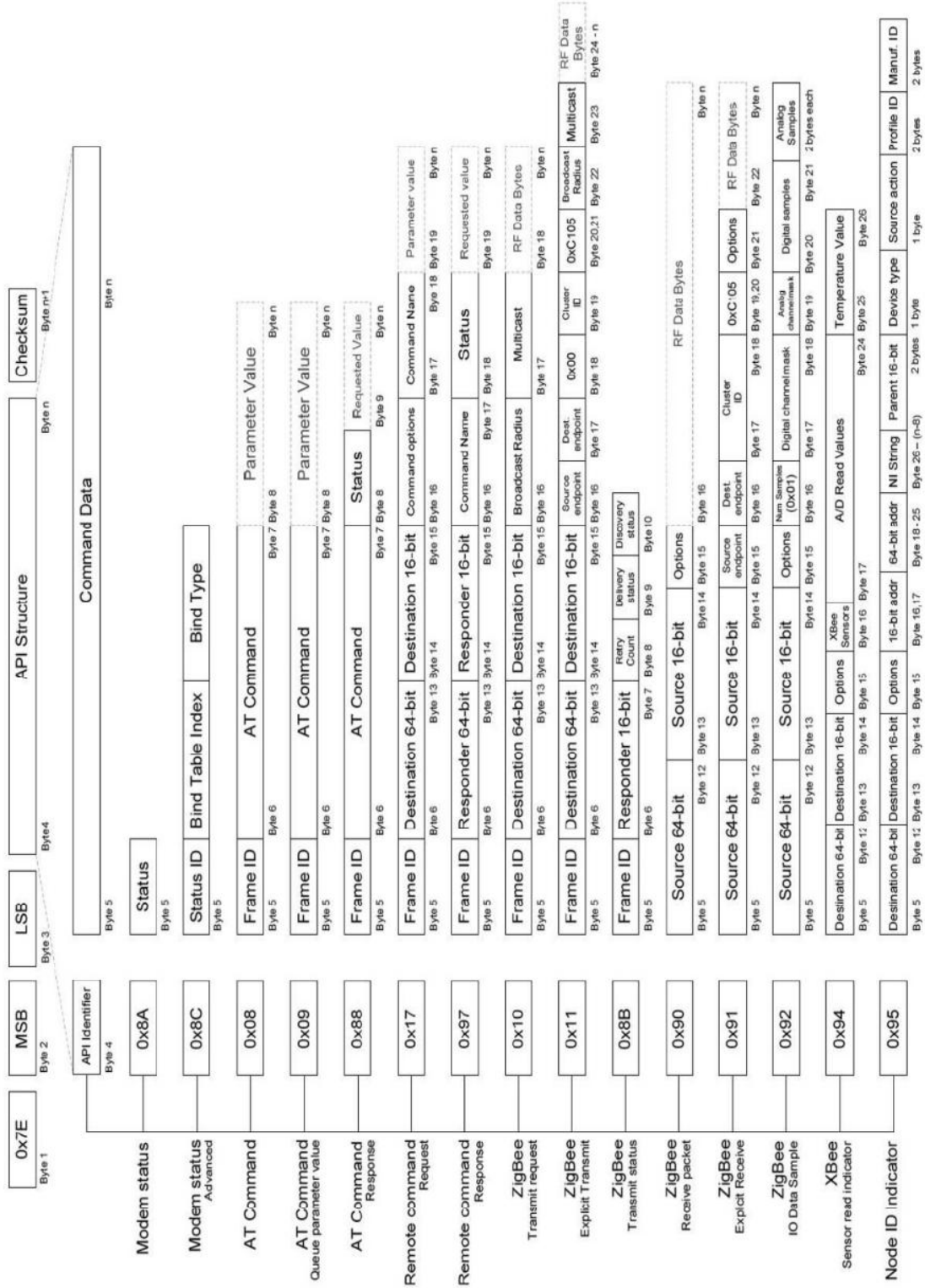
### Parte de cima



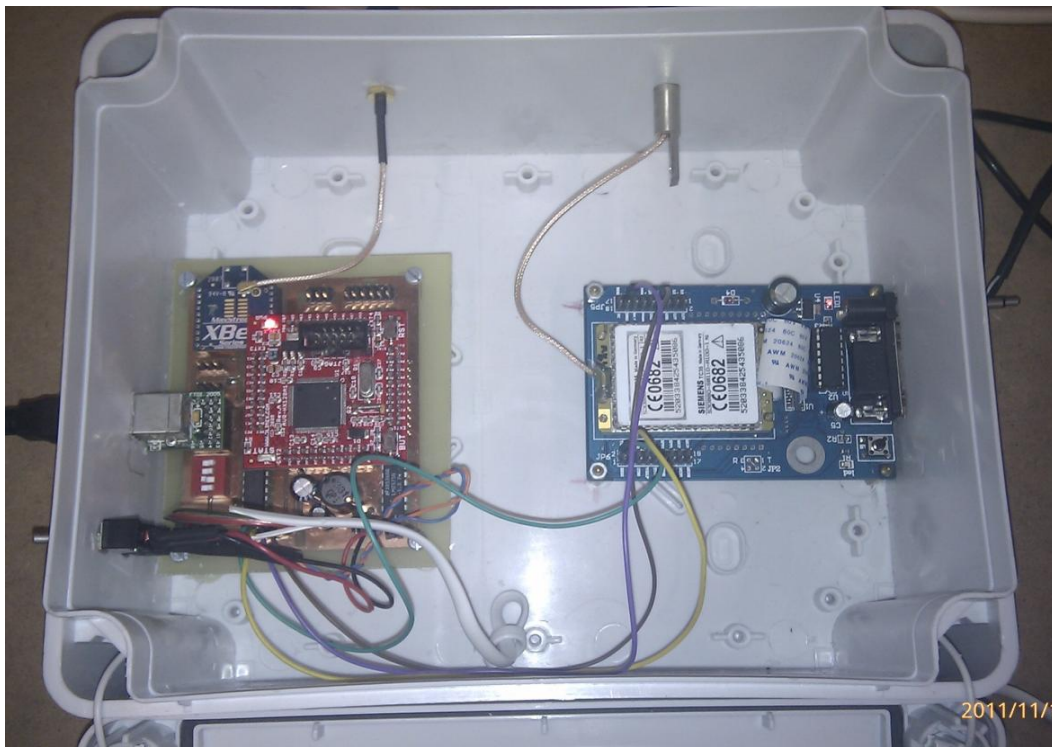
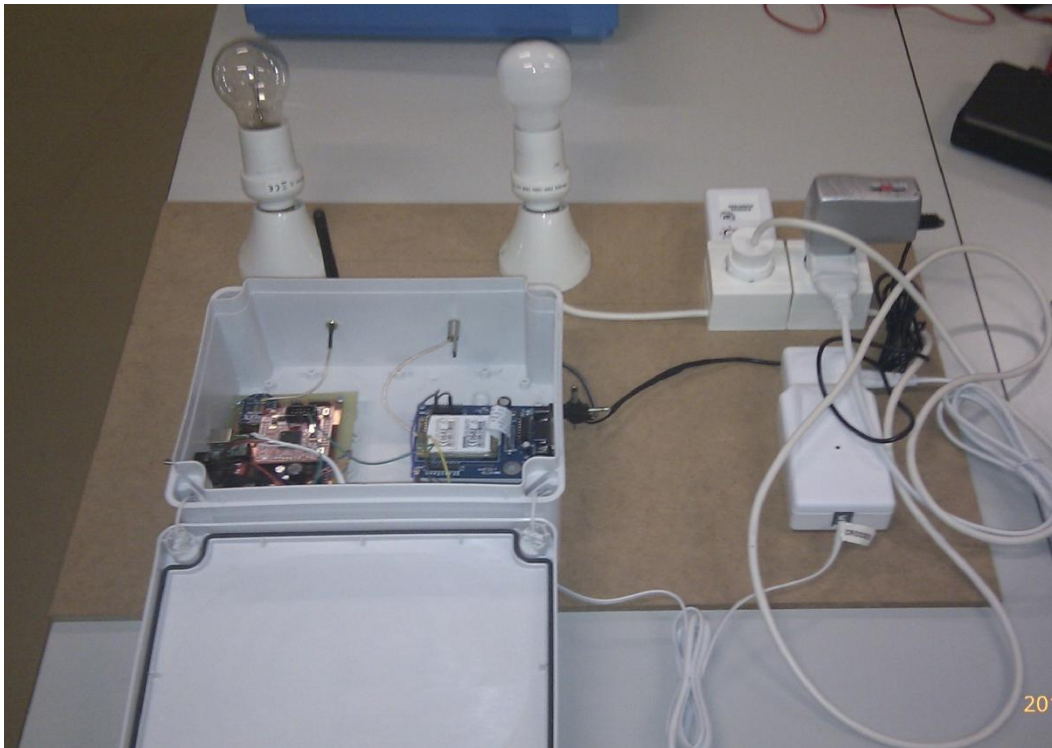
### Parte de baixo



# Anexo D: Lista de APIs XBee



## Anexo E: Fotografia do protótipo implementado



## Anexo F: Código C do *firmware* da *gateway*

A listagem do código encontra-se em formato digital no ficheiro “AnexoF.pdf”.

## Anexo G: Tabela dos atributos e comandos da camada de aplicação PIntern

Tabela G.1 – Atributos

Atributos	ID	NOTA
STATUS	0	Atributo que contém o estado do dispositivo (0 – OK, 1-OFF, 2-ERRO)
DESCR	1	Atributo que contém uma sequencia de caracteres que descreve o dispositivo
SWITCH	2	Atributo que contém o estado de um interruptor (0- OFF, 1- ON)
TEMP	3	Atributo que contém o valor de temperatura
LUM	4	Atributo que contém o valor da Luminosidade
HUM	5	Atributo que contém o valor da Humidade
DISPLAY	6	Atributo que contém uma sequência de caracteres a representar num <i>display</i>
CURRENT	7	Atributo que contém o valor da corrente

Tabela G.2 – Comandos

ID	Comando	Descrição
0	READ	Comando para leitura de um atributo
1	WRITE	Comando para escrita de um atributo
2	ADD_DEVICE	Comando para adicionar dispositivo à lista interna
3	REM_DEVICE	Comando para remover dispositivo da lista interna
4	LIST_DEVICE	Comando para pedir listagem dos dispositivos registados
5	ADD_ROUTE	Comando para adicionar rota à lista interna
6	REM_ROUTE	Comando para remover rota da lista interna
7	LIST_ROUTE	Comando para pedir listagem das rotas registadas
8	SAVE_DATA	Comando para armazenar os dados das tabelas em memória não volátil

## Anexo H: Código JAVA da aplicação

A listagem do código encontra-se em formato digital no ficheiro “AnexoH.pdf”.

## Anexo I: Tabelas dos valores dos cenários de teste do sistema de despacho

**Tabela I.1 – Valores de temperatura utilizados nos testes**

Hora	Temperatura (°C)
0	25
1	24
2	23
3	22
4	23
5	22
6	25
7	20
8	19
9	20
10	25
11	23
12	26
13	27
14	28
15	27
16	28
17	24
18	25
19	25
20	22
21	23
22	27
23	25

**Tabela I.2 – Parâmetros utilizados nos testes do cenário 1**

Hora	Potência Consumida (MW)	Potência Produzida (MW)	Hídrica (MW)	Térmica (MW)	Eólica (MW)	Solar (MW)	preço1 (€)	preço 2 (€)	preço 3 (€)
0	80	134	28	100	6	0	0,066	0,0802	0,12
1	75	126	20	100	6	0	0,069	0,0802	0,12
2	72	122	17	100	5	0	0,07	0,0802	0,12
3	72	119	16	100	3	0	0,073	0,0802	0,12
4	72	117	15	100	2	0	0,075	0,0802	0,12
5	81	119	16	100	3	0	0,082	0,0802	0,12
6	100	120	20	100	0	0	0,1	0,0802	0,12
7	110	123	23	100	0	0	0,107	0,0802	0,12
8	120	124	24	100	0	0	0,116	0,15	0,12
9	128	120	19	100	0	1	0,128	0,15	0,12
10	130	124	22	100	0	2	0,126	0,15	0,12
11	127	128	24	100	0	4	0,119	0,1382	0,12
12	122	129	24	100	0	5	0,113	0,1382	0,12
13	120	128	24	100	0	4	0,112	0,1382	0,12
14	123	128	24	100	0	4	0,115	0,1382	0,12
15	123	125	24	100	0	1	0,118	0,1382	0,12
16	119	119	15	100	0	0	0,12	0,1382	0,12
17	117	119	10	100	2	0	0,118	0,1382	0,12
18	117	120	18	100	3	0	0,117	0,1382	0,12
19	121	115	15	100	0	0	0,126	0,15	0,12
20	130	116	16	100	0	0	0,134	0,15	0,12
21	120	120	20	100	0	0	0,12	0,0802	0,12
22	108	122	22	100	0	0	0,106	0,0802	0,12
23	90	124	24	100	0	0	0,086	0,0802	0,12

**Tabela I.3 – Parâmetros utilizados nos testes do cenário 2**

Hora	Potência Consumida (MW)	Potência Produzida (MW)	Hídrica (MW)	Térmica (MW)	Eólica (MW)	Solar (MW)	preço1 (€)	preço 2 (€)	preço 3 (€)
0	80	134	28	100	6	0	0,066	0,0802	0,12
1	75	126	20	100	6	0	0,069	0,0802	0,12
2	72	122	17	100	5	0	0,07	0,0802	0,12
3	72	119	16	100	3	0	0,073	0,0802	0,12
4	72	117	15	100	2	0	0,075	0,0802	0,12
5	81	119	16	100	3	0	0,082	0,0802	0,12
6	100	120	20	100	0	0	0,1	0,0802	0,12
7	110	123	23	100	0	0	0,107	0,0802	0,12
8	120	124	24	100	0	0	0,116	0,15	0,12
9	128	125	24	100	0	1	0,123	0,15	0,12
10	130	124	22	100	0	2	0,126	0,15	0,12
11	127	128	24	100	0	4	0,119	0,1382	0,12
12	122	129	24	100	0	5	0,113	0,1382	0,12
13	120	128	24	100	0	4	0,112	0,1382	0,12
14	123	128	24	100	0	4	0,115	0,1382	0,12
15	123	125	24	100	0	1	0,118	0,1382	0,12
16	119	125	25	100	0	0	0,114	0,1382	0,12
17	117	125	23	100	2	0	0,112	0,1382	0,12
18	117	123	20	100	3	0	0,114	0,1382	0,12
19	121	125	25	100	0	0	0,116	0,15	0,12
20	130	126	26	100	0	0	0,124	0,15	0,12
21	120	126	26	100	0	0	0,114	0,0802	0,12
22	108	129	29	100	0	0	0,099	0,0802	0,12
23	90	128	28	100	0	0	0,082	0,0802	0,12

**Tabela I.4 – Parâmetros utilizados nos testes do cenário 3**

Hora	Potência Consumida (MW)	Potência Produzida (MW)	Hídrica (MW)	Térmica (MW)	Eólica (MW)	Solar (MW)	preço1 (€)	preço 2 (€)	preço 3 (€)
0	80	124	28	90	6	0	0,076	0,0802	0,12
1	75	116	20	90	6	0	0,079	0,0802	0,12
2	72	112	17	90	5	0	0,08	0,0802	0,12
3	72	109	16	90	3	0	0,083	0,0802	0,12
4	72	107	15	90	2	0	0,085	0,0802	0,12
5	81	109	16	90	3	0	0,092	0,0802	0,12
6	100	110	20	90	0	0	0,11	0,0802	0,12
7	110	113	23	90	0	0	0,117	0,0802	0,12
8	120	114	24	90	0	0	0,126	0,15	0,12
9	128	110	19	90	0	1	0,138	0,15	0,12
10	130	114	22	90	0	2	0,136	0,15	0,12
11	127	118	24	90	0	4	0,129	0,1382	0,12
12	122	119	24	90	0	5	0,123	0,1382	0,12
13	120	118	24	90	0	4	0,122	0,1382	0,12
14	123	118	24	90	0	4	0,125	0,1382	0,12
15	123	117	24	90	0	3	0,126	0,1382	0,12
16	119	118	24	90	0	4	0,121	0,1382	0,12
17	117	118	25	90	2	1	0,119	0,1382	0,12
18	117	120	26	90	3	1	0,117	0,1382	0,12
19	121	117	25	90	2	0	0,124	0,15	0,12
20	130	118	26	90	2	0	0,132	0,15	0,12
21	120	119	26	90	3	0	0,121	0,0802	0,12
22	108	123	29	90	4	0	0,105	0,0802	0,12
23	90	122	28	90	4	0	0,088	0,0802	0,12

**Tabela I.5 – Parâmetros utilizados nos testes do cenário 4**

Hora	Potência Consumida (MW)	Potência Produzida (MW)	Hídrica (MW)	Térmica (MW)	Eólica (MW)	Solar (MW)	preço1 (€)	preço 2 (€)	preço 3 (€)
0	80	134	28	100	6	0	0,066	0,0802	0,12
1	75	126	20	100	6	0	0,069	0,0802	0,12
2	72	122	17	100	5	0	0,07	0,0802	0,12
3	72	119	16	100	3	0	0,073	0,0802	0,12
4	72	117	15	100	2	0	0,075	0,0802	0,12
5	81	119	16	100	3	0	0,082	0,0802	0,12
6	100	120	20	100	0	0	0,1	0,0802	0,12
7	110	123	23	100	0	0	0,107	0,0802	0,12
8	120	124	24	100	0	0	0,116	0,15	0,12
9	126	120	19	100	0	1	0,126	0,15	0,12
10	127	124	22	100	0	2	0,123	0,15	0,12
11	127	128	24	100	0	4	0,119	0,1382	0,12
12	122	129	24	100	0	5	0,113	0,1382	0,12
13	140	128	24	100	0	4	0,132	0,1382	0,12
14	123	128	24	100	0	4	0,115	0,1382	0,12
15	123	125	24	100	0	1	0,118	0,1382	0,12
16	119	115	15	100	0	0	0,124	0,1382	0,12
17	100	112	10	100	2	0	0,108	0,1382	0,12
18	100	121	18	100	3	0	0,099	0,1382	0,12
19	121	125	25	100	0	0	0,116	0,15	0,12
20	130	126	26	100	0	0	0,124	0,15	0,12
21	120	126	26	100	0	0	0,114	0,0802	0,12
22	108	129	29	100	0	0	0,099	0,0802	0,12
23	90	128	28	100	0	0	0,082	0,0802	0,12

**Tabela I.6 – Parâmetros utilizados nos testes do cenário 5**

Hora	Potência Consumida (MW)	Potência Produzida (MW)	Hídrica (MW)	Térmica (MW)	Eólica (MW)	Solar (MW)	preço1 (€)	preço 2 (€)	preço 3 (€)
0	80	134	28	100	6	0	0,066	0,0802	0,12
1	75	126	20	100	6	0	0,069	0,0802	0,12
2	72	125	20	100	5	0	0,067	0,0802	0,12
3	72	123	20	100	3	0	0,069	0,0802	0,12
4	72	124	22	100	2	0	0,068	0,0802	0,12
5	81	124	21	100	3	0	0,077	0,0802	0,12
6	100	120	20	100	0	0	0,1	0,0802	0,12
7	110	123	23	100	0	0	0,107	0,0802	0,12
8	120	124	24	100	0	0	0,116	0,15	0,12
9	128	123	22	100	0	1	0,125	0,15	0,12
10	130	124	22	100	0	2	0,126	0,15	0,12
11	127	127	23	100	0	4	0,12	0,1382	0,12
12	122	128	23	100	0	5	0,114	0,1382	0,12
13	120	127	23	100	0	4	0,113	0,1382	0,12
14	123	126	22	100	0	4	0,117	0,1382	0,12
15	123	125	22	100	0	3	0,118	0,1382	0,12
16	119	123	20	100	0	3	0,116	0,1382	0,12
17	117	110	10	100	0	0	0,127	0,1382	0,12
18	117	110	10	100	0	0	0,127	0,1382	0,12
19	121	125	25	100	0	0	0,116	0,15	0,12
20	130	126	26	100	0	0	0,124	0,15	0,12
21	120	126	26	100	0	0	0,114	0,0802	0,12
22	108	129	29	100	0	0	0,099	0,0802	0,12
23	90	128	28	100	0	0	0,082	0,0802	0,12

**Anexo J: Registos da execução do sistema de despacho de energia**

A listagem os registos encontra-se em formato digital no ficheiro “AnexoJ.pdf”.