

INTERFACE IP-KNX PARA UM SISTEMA DE CONTROLO DOMÓTICO

Catarina Adelaide Dias Contente



Mestrado em Engenharia Electrotécnica e de Computadores

Área de especialização de Automação e Controlo

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2013

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, ramo de Automação e Sistemas.

Candidata: Catarina Adelaide Dias Contente, N° 1070207, 1070207@isep.ipp.pt

Orientação científica: Filipe Miguel Tavares De Azevedo, fta@isep.ipp.pt

Co-orientação: Domingos Salvador Dos Santos, dss@isep.ipp.pt



Mestrado em Engenharia Electrotécnica e de Computadores

Área de especialização de Automação e Controlo

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2013

Agradecimentos

Numa época em que são cada vez menos os que se aventuram numa rota universitária, não poderia deixar de agradecer a todos aqueles que, de uma forma ou de outra, me guiaram até esta etapa, me encorajando no início, me apoiando em momentos de maior fadiga, não me deixaram esmorecer. Para todos esses o meu profundo e sincero “obrigada”.

Começo por deixar o meu agradecimento aos meus pais, ao meu irmão e à minha avó Alice, por toda a influência que exercem na minha vida. Obrigada por existirem, pelos ensinamentos, pelos princípios que me passaram, que fizeram de mim a pessoa que hoje aqui escreve.

Como não poderia deixar de ser, agradeço ao Pedro, meu namorado e amigo, por todo o amor e companheirismo, pela forma como faz parte da minha vida e a meu lado caminha em todas as decisões.

Agradeço também aos meus amigos, em particular à Catarina, à Cátia e à Eli por me acompanharem, e por, numa perspectiva mais cómica, marcarem o meu mundo feminino com as roupas e vernizes a que ele tem direito!

À Diana agradeço-lhe a amizade e também a paciência para dar uma mãozinha na revisão deste documento. Obrigada Didi pela disponibilidade.

Agradeço também aos meus colegas do ISEP que marcaram o meu percurso universitário, em particular ao João Costa, Sérgio Ramos, Jonathan Garcia, e ao Ricardo Gomes pela integração num mundo marcadamente masculino mas que, afinal, não se pauta por tantas diferenças como se faz pensar por vezes.

A todos os professores que participaram no meu percurso escolar, em particular àquela que foi a minha primeira formadora, a professora Madalena Matos, que preencheu todas as fichas lectivas delatando as minhas distrações e travessuras, que exigiu que eu explorasse as minhas capacidades cognitivas desde tenra idade, moldando-me desde cedo para aqui chegar.

Ao Carlos Paiva, meu Mestre, colega e amigo, obrigada pela vasta partilha de conhecimentos que, sem dúvida, marcou para sempre a minha vida profissional.

Agradeço também ao técnico do laboratório de máquinas eléctricas, Armando Herculano, pela disponibilidade durante o desenvolvimento deste trabalho.

Por fim mas não em último lugar, deixo o meu agradecimento aos Engenheiros Filipe Azevedo e Domingos Santos pela disponibilidade para me orientarem nesta dissertação.

Resumo

São vários os factores sociais e económicos que valorizam a aplicação de tecnologias de domótica em edifícios. No caso particular dos edifícios residenciais, a tendência dos seus utilizadores é a instalação de sistemas de controlo da segurança, do ambiente, de mecanismos de rega e de alarmes. Assim, seguindo a premissa do marketing, que identifica como uma boa prática a projecção de produtos / serviços que satisfaçam as necessidades inventariadas pelos seus utilizadores, este trabalho assenta na criação de um sistema domótico, controlado remotamente através de uma aplicação Android, que pretende, numa primeira instância, o controlo das lâmpadas de uma habitação.

Neste trabalho é utilizado o protocolo KNX.TP para a comunicação dos dispositivos de domótica existentes no ISEP, que constituem o ambiente domótico deste trabalho. De forma a implementar o controlo remoto destes dispositivos via internet, este trabalho foca-se no desenvolvimento de uma interface IP-KNX, usando como *hardware* de controlo, um Arduino Mega 2560, uma placa de interface Ethernet para Arduino, a placa de integração KNX, e um servidor web com a linguagem PHP instalada. Para efeitos de demonstração, foi criada uma aplicação para o SO Android que controla as lâmpadas da rede KNX.

Neste trabalho foram utilizadas várias linguagens de programação: C++ no *firmware* do Arduino, PHP no servidor web e JAVA + XML na aplicação Android.

Palavras-Chave

Domótica, Automação Residencial, Edifício Inteligente, KNX, *TP-UART*, Interface, Android, Cliente, Servidor, *Smartphone*, Controlo Remoto, Interoperação, Arduino.

Abstract

There are many social and economic factors that value the application of home automation technologies in buildings. In the particular case of residential buildings, their users' tendency is to install security, environment, alarms and irrigation mechanisms control systems. Therefore, following the marketing premise, which identifies as a good manner the projection of products/services that satisfy their users' needs, this task is based on the creation of an Android application which controls remotely a home automation system, that want to control the lighting in a house, in the first stage.

This paper uses the KNX.TP protocol for the home automation devices communication currently on use in ISEP that compose this work home automation environment. In order to implement these devices remote control through internet, this paper focuses in the development of a IP-KNX interface, using as his control hardware a Arduino Mega 2560, an Ethernet interface plate for Arduino, the KNX integrating plate and a web server with PHP language installed. For demonstration, it has been created an application for the SO Android that controls the lightings of the KNX network.

In the presented task, many programming languages have been used: C++ in the Arduino firmware, PHP in the web server and JAVA + XML in the Android application.

Keywords: Home automation; building home automation; smart building; KNX, TP-UART, Interface, Android, Client, Server, Smartphone, Remote Control, Interoperation; Arduino.

Résumé

Il y a beaucoup de facteurs sociaux et économiques qui valorisent l'application des technologies d'automatisation dans les bâtiments. Dans le cas particulier des immeubles d'habitation, la tendance de ses utilisateurs est l'installation de systèmes de contrôle de la sécurité, de l'environnement, des mécanismes d'irrigation et des alarmes. Ainsi, suite à la prémisses de marketing, qui s'identifie comme une bonne pratique de la projection des produits / services qui répondent aux besoins recensés par ses utilisateurs, ce travail est basé sur la création d'un système domotique, contrôlé à distance par le biais d'une application Android, qui cherche en premier lieu, le contrôle des lampes.

Ce travail utilise le protocole KNX.TP pour la communication des dispositifs domotiques existants dans ISEP, qui constituent l'environnement domotique de ce travail. Pour mettre en œuvre le contrôle à distance de ces appareils par le biais de l'internet, ce travail se concentre sur le développement d'une interface IP KNX, en utilisant comme *hardware* de commande, un Arduino Mega 2560, une interface Ethernet, une intégration pour le réseau KNX, et un serveur web avec PHP installé. Pour des fins de démonstration, une application a été créée pour le système Android qui contrôle les lampes du réseau KNX.

Dans cette étude, nous avons utilisé plusieurs langages de programmation : C ++ dans de *firmware* de l'Arduino, PHP sur le serveur Web et Java + XML dans l'application Android.

Mots clef:

Domotique, Automation, Bâtiments intelligents, Arduino, KNX TP-UART, Interface, Android, Client, Serveur, *Smartphone*, Télécommande, Interopération.

Nota ao Leitor

O presente documento está redigido em português europeu, segundo o acordo ortográfico de 1990. Ao longo da sua estrutura, é possível encontrar, simultaneamente, descrições em português e em inglês, bem como abreviaturas. Tal opção está relacionada com a utilização e descrição de alguns conceitos que são mais reconhecidos usando a literatura anglo-saxónica, utilizando uma formatação em *itálico* nestes estrangeirismos.

Índice

1. INTRODUÇÃO	1
1.1. MOTIVAÇÃO	2
1.2. OBJECTIVOS.....	4
1.3. GESTÃO DO TRABALHO.....	5
1.4. ORGANIZAÇÃO DO MANUSCRITO	5
2. SISTEMA DOMÓTICO	7
2.1. SISTEMA DOMÓTICO	8
2.1.1. <i>Funcionalidades</i>	8
2.1.2. <i>Protocolos de comunicação</i>	9
2.1.3. <i>Soluções comercializadas no mercado nacional</i>	12
2.2. CONTROLO DO AMBIENTE DOMÓTICO	14
2.2.1. <i>Funcionalidades de um sistema de controlo doméstico</i>	15
2.2.2. <i>Requisitos de qualidade de um sistema de controlo</i>	16
2.3. SÍNTESE	18
3. DEFINIÇÕES TÉCNICAS DO SISTEMA.....	19
3.1. REQUISITOS DA IMPLEMENTAÇÃO DO SISTEMA	19
3.2. <i>NORMA KNX</i>	21
3.2.1. <i>Arquitectura</i>	22
3.2.2. <i>Telegrama KNX</i>	24
3.2.3. <i>Endereçamento</i>	26
3.2.4. <i>Interoperação</i>	27
3.2.5. <i>Engeneering Tool Software - ETS</i>	27
3.3. REDE KNX	28
3.4. INTERFACE IP-KNX	29
3.4.1. <i>Unidade de Processamento</i>	30
3.4.2. <i>Integração na rede KNX</i>	31
3.4.3. <i>Integração na rede IP</i>	32
3.5. <i>INTERFACE GRÁFICA ANDROID</i>	33
3.5.1. <i>Android-Client</i>	33
3.5.2. <i>Dados Estatísticos</i>	34
3.6. BASES DE DADOS	35
3.7. FERRAMENTAS DE DESENVOLVIMENTO	36
3.8. SÍNTESE	37
4. IMPLEMENTAÇÃO DO SISTEMA.....	39

4.1.	DESCRIÇÃO GERAL DA IMPLEMENTAÇÃO	39
4.2.	HARDWARE DA INTERFACE IP/KNX.....	42
4.2.1.	Diagrama de blocos.....	43
4.2.2.	Esquema electrónico.....	43
4.2.3.	Circuito Impresso (PCB)	45
4.3.	FIRMWARE DO ARDUINO.....	45
4.3.1.	Interface de controlo KNX.....	46
4.3.2.	Servidor TCP-IP	47
4.4.	CONTROLO DE ALTO NÍVEL	48
4.4.1.	Cliente TCP-IP	48
4.4.2.	Aplicação móvel.....	50
4.5.	SÍNTESE.....	53
5.	VALIDAÇÃO DO SISTEMA	55
5.1.	DEFINIÇÃO DE TESTES	55
5.2.	RESULTADOS.....	56
5.2.1.	Comunicação entre a Rede KNX e a interface IP-KNX.....	56
5.2.2.	Comunicação entre Interface IP-KNX e o servidor web.....	57
5.2.3.	Comunicação entre o servidor web e o cliente android.....	60
5.3.	SÍNTESE.....	62
6.	CONCLUSÃO	63
6.1.	APRECIÇÃO CRÍTICA DO SISTEMA DESENVOLVIDO	63
6.2.	DESENVOLVIMENTOS FUTUROS	64
6.2.1.	Interface IP-KNX.....	65
6.2.2.	API Android.....	65
6.2.3.	Validação.....	66
	REFERÊNCIAS DOCUMENTAIS.....	67
	ANEXO	71

Índice de Ilustrações

Figura 1	Representação metafórica de uma residência controlada remotamente [9].	4
Figura 2	Representação de algumas funcionalidades de uma casa automatizada [9].	9
Figura 3	Diagrama do sistema – relação dos requisitos de implementação.	21
Figura 4	Instalação KNX (adaptado de [18]).	21
Figura 5	Estrutura de uma instalação KNX [18].	22
Figura 6	Topologias possíveis da rede KNX (adaptado de [25]).	23
Figura 7	Topologia hierarquizada da rede KNX [18].	23
Figura 8	Telegrama KNX (adaptado de [27]).	24
Figura 9	Estrutura de uma Palavra [28].	25
Figura 10	Bits do telegrama KNX e do endereço de origem (adaptado de [28]).	25
Figura 11	Estrutura do arquivo de um projecto do ETS [31].	28
Figura 12	Placa de testes de rede KNX existente no laboratório do ISEP.	28
Figura 13	Rede KNX com ligação à internet através de uma <i>interface</i> IP/KNX.	29
Figura 14	Arduino Mega 2560 [36].	31
Figura 15	CI <i>Transceiver</i> KNX: Siemens TP-UART-IC.	31
Figura 16	Arduino Ethernet Shield W5100 [41].	32
Figura 17	Dados estatísticos da utilização das versões da plataforma Android [49].	34
Figura 18	Dados estatísticos das configurações de ecrã actualmente mais utilizadas [49].	35
Figura 19	Diagrama geral da implementação.	40
Figura 20	Diagrama geral do fluxo de comunicação.	41
Figura 21	Diagrama de blocos da interface IP-KNX.	43
Figura 22	Esquema eléctrico da placa de interface KNX – Bloco de isolamento galvânico.	44
Figura 23	Esquema eléctrico da placa de interface KNX – Ligações do TP-UART.	44
Figura 24	<i>Interface</i> IP-KNX – Imagens do PCB.	45
Figura 25	Fluxograma geral do <i>firmware</i> do Arduino.	46
Figura 26	Fluxograma da função Parse.	47
Figura 27	Diagrama de comunicação entre o servidor web e a interface IP-KNX.	48
Figura 28	Fluxograma do servidor <i>web</i> – <i>Socket</i> TCP-IP (cliente).	49
Figura 29	Fluxograma do ficheiro <i>index.php</i> do servidor <i>web</i> .	50
Figura 30	Cliente Android – Páginas de (i) Login, (ii) Registo e (iii) Dashboard.	53
Figura 31	Porta série 0 – Telegrama de Reset.	56
Figura 32	Envio de comandos para a rede KNX e recepção das respostas dessa rede.	57
Figura 33	Servidor web – envio de comando inválido.	58
Figura 34	Comunicação da interface IP-KNX com o servidor web – Comando inválido.	58

Figura 35	Servidor web – envio de comando válido.	59
Figura 36	Comunicação da interface IP-KNX com o servidor web – <i>Timeout</i>	59
Figura 37	Teste da comunicação do interface IP-KNX com o servidor web – Comando válido.	60
Figura 38	API Android – Janela de registo e excerto da respectiva entrada na BD remota.	61
Figura 39	API Android – Validação do Login.....	61
Figura 40	API Android – Comutação do estado da lâmpada 1.....	62

Índice de Tabelas

Tabela 1	Diagrama de Gantt do projecto.	5
Tabela 2	Protocolos de rede utilizados na domótica.	11
Tabela 3	Soluções de sistemas domóticos mais relevantes no mercado nacional.....	13
Tabela 4	Funcionalidades explícitas dos sistemas domóticos.....	15
Tabela 5	Funcionalidades implícitas dos sistemas de controlo domótico.....	16
Tabela 6	Requisitos de qualidade de um sistema de controlo domótico.....	17
Tabela 7	Comandos DML.....	35
Tabela 8	Ferramentas e elementos de desenvolvimento.	36
Tabela 9	Comandos implementados na interface IP-KNX.	42
Tabela 10	Tabela “users” da BD remota.....	51

Acrónimos

- ADT – *Android Development Tools* (Ferramentas de Desenvolvimento Android)
- API – *Application Programming Interface* (Aplicação de Interface de Programação)
- AVAC – Aquecimento, Ventilação e Ar condicionado
- AVD – *Android Virtual Device* (Dispositivo Android Virtual)
- BAS – *Building Automation Systems* (Sistemas de Automação de Edifícios)
- BCI – *BatiBus Club International* (Clube Internacional BatiBus)
- BCU – *Bus Coupling Unit* (Unidade de Acoplamento ao Barramento)
- BD – Base de Dados
- CI – Circuito Integrado
- DEE – Departamento de Engenharia Electrotécnica
- DIP – *Dual In-Line Package*
- DMI – Dispositivos Móveis de Internet
- DML – *Data Manipulation Language* (Linguagem de Manipulação de Dados)
- DTP – *Datapoints*
- DVM – *Dalvik Virtual Machine* (Máquina Virtual Dalvik)
- EEPROM – *Electrically-Erasable Programmable Read Only Memory* (Memória Programável de Leitura Apagável Electricamente)
- EHSA – *European Home Systems Association* (Associação Europeia de Sistemas)

Domésticos)

- EIB – *European Installation Bus* (Barramento de Instalação Europeu)
- EIS – *EIB Interworking Standards* (Standards de interoperabilidade EIB)
- EIBA – *European Installation Bus Association* (Associação EIB)
- ETS – *EIB Tool Software* (Ferramenta de *Software* EIB)
- GUI – *Graphical User Interface* (Interface Gráfica do Utilizador)
- IDE – *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado)
- IP – *Internet Protocol* (Protocolo de Internet)
- ISEP – Instituto Superior De Engenharia Do Porto
- JASON – *Java Script Object Notation*
- KNX – Konnex
- MCU – *Microcontroller* (Microcontrolador)
- PC – *Personal Computer* (Computador Pessoal)
- PCB – *Printed Circuit Board* (Placa de Circuito Impresso)
- SDK – *Software Development Kit* (Kit de Desenvolvimento de *Software*)
- SMD – *Surface Mounted Device* (Componente de Montagem em Superfície)
- SO – Sistema Operativo
- SOIC – *Small Outline Integrated Circuit*
- SPI – *Serial Peripheral Interface*
- SQL – *Structured Query Language* (Linguagem de Pergunta Estruturada)
- SRAM – *Static Random Access Memory* (Memória Estática de Acesso Aleatório)

- ROM – *Read Only Memory* (Memória Apenas de Leitura)
- TP – *Twisted Pair* (Par trançado)
- TCP – *Transmission Control Protocol* (Protocolo de Controlo da Transmissão)
- UART – *Universal Asynchronous Receiver Transmitter* (Emissor Receptor Assíncrono Universal)
- UDP – *User Datagram Protocol*
- UI – *User Interface* (Interface de Utilizador)
- USB – *Universal Serial Bus* (Barramento Série Universal)

1. INTRODUÇÃO

Dadas as consequências da actual crise energética, provocada pelo aumento da procura dos combustíveis fósseis, a gestão energética reveste-se da maior relevância, traduzindo-se numa mais-valia no que toca à eficiência [1]. Tal facto veio impulsionar a aplicação da automação a edifícios, tecnologia normalmente designada por domótica. Assim, pode afirmar-se que o termo “domótica” designa a utilização da electrónica e da informática para melhorar a funcionalidade e o conforto de um edifício, tanto ao nível empresarial como em aplicações domésticas [2].

A utilização da domótica está em ascensão, aplicando-se cada vez mais em edifícios de serviços, industriais, hospitalares, etc., construídos de raiz ou requalificados, uma vez que o seu *payback* é de um curto espaço de tempo, focando-se principalmente na eficiência energética e nos sistemas de segurança, comumente designados de *Building Automation Systems* (BAS) [1].

Nas habitações, a domótica é implementada com especial ênfase no conforto dos seus utilizadores, o que se traduz num incremento do seu valor devido ao luxo e ostentação que exibem. Deste modo, dotar a casa de “inteligência” pode ser um meio de, não só controlar as variáveis da sua habitação, mas também, sentir o reconhecimento e valorização da sociedade. Este facto faz da domótica um trunfo diferenciador no mercado imobiliário, também em dificuldades actualmente [3].

É comum denominar “edifício inteligente” a um edifício com domótica. De acordo com as suas funcionalidades, distinguem-se cinco tipos de edifícios inteligentes:

- *Contains intelligent objects* (na literatura anglo-saxónica) - Edifício que contém dispositivos e electrodomésticos que funcionam de um modo inteligente;
- *Contains intelligent, communicating object* (na literatura anglo-saxónica) - Edifício que contém dispositivos inteligentes que comunicam entre si, trocando informações e aumentando assim a sua funcionalidade;
- *Connected home* (na literatura anglo-saxónica) - Edifício que tem uma rede interna interligada com a rede externa, permitindo o controlo interactivo dos sistemas, e o acesso aos serviços e à informação, quer de dentro, quer do exterior;
- *Learning home* (na literatura anglo-saxónica) - Neste tipo de edifício inteligente os padrões de utilização são armazenados, e os dados acumulados são utilizados na antecipação das necessidades dos utilizadores. Por exemplo, o edifício adquire e ajusta os padrões da utilização do aquecimento e da iluminação (“*the adaptive home*”);
- *Alert home* (na literatura anglo-saxónica) - As actividades das pessoas e dos objectos são constantemente monitorizadas alertando e antecipando acções a tomar (“*the aware home*”) [4].

1.1. MOTIVAÇÃO

As necessidades do mercado são uma componente crítica para orientar a Organização e a sua gestão, dado que actualmente, a definição do produto não pode ser independente das necessidades dos clientes. Os fabricantes têm então que estudar as necessidades dos clientes para, a partir daí, desenharem um produto que os satisfaça. É a era do *marketing* [5].

O *marketing* defende que a identificação e selecção de conjuntos heterogéneos de consumidores (mercado-alvo), e a sua análise quantitativa e qualitativa, é essencial para planear e executar a concepção de produtos capazes de satisfazer as suas necessidades, de forma a que o grau de satisfação seja superior ao obtido pelos produtos concorrentes, permitindo assim que o relacionamento entre a Organização e os seus clientes seja duradouro [6].

Segundo um estudo realizado por António Flores, baseado nos resultados de um inquérito a profissionais de classe média e alta, para o consumidor, uma casa inteligente é “... *uma casa que apresenta automatismos, que gere da melhor forma os seus recursos energéticos e ecológicos, que é programada, que pode ser programada à distância, que tem componentes electrónicos que auxiliam a gestão de tarefas domésticas e tudo isto, para melhorar a qualidade de vida dos seus utilizadores*” [4].

Os resultados desse estudo indicam a satisfação dos utilizadores de habitações inteligentes e que os seus principais interesses relativamente aos sistemas a controlar se centram na segurança, no controlo de ambiente, no sistema de rega e nos alarmes. Conclui-se ainda que os consumidores têm a percepção de que é muito dispendioso automatizar uma casa e que o seu controlo poderá ser complexo, necessitando de um apoio técnico permanente [4].

De acordo com o *Information Economy Report 2011* da Organização das Nações Unidas (ONU), estima-se que no final de 2010 mais de 75% da população dos países desenvolvidos utilizava a internet, o que levou à proliferação dos dispositivos de acesso à internet, nomeadamente os *smartphones* e os *tablets*, em detrimento dos computadores pessoais (PC) [7].

Focado no mercado dos *smartphones*, o *Gartner Group* realizou um estudo sobre a utilização de sistemas operativos móveis, baseado nas vendas dos dispositivos móveis com acesso à internet. Este estudo revelou que em 2012 o Sistema Operativo (SO) Android foi o mais popular com 49,2% do mercado [8].

Os resultados destes estudos, cuja Figura 1 pretende, de uma forma metafórica, representar, foram determinantes na motivação do presente trabalho, o desenvolvimento de um sistema doméstico controlado remotamente por uma aplicação Android.



Figura 1 Representação metafórica de uma residência controlada remotamente [9].

1.2. OBJECTIVOS

A iluminação é, possivelmente, a variável mais significativa no gasto energético global de utilização, sendo também a mais ajustável ou a mais susceptível de ser controlada em domótica. Desta forma, o controlo da iluminação contribui para o aumento da eficiência energética, sendo para tal, de vital importância que se evitem situações de desperdício e se maximize o aproveitamento da luz natural, respeitando os níveis de iluminação adequados. Como consequência indirecta, regular a luminosidade permite a redução da carga térmica da iluminação e, conseqüentemente, poupanças energéticas relacionadas com os sistemas de Aquecimento, Ventilação e Ar-Condicionado (AVAC) [1][8][9].

O controlo de estores permite regular a entrada de iluminação natural e também a entrada da luz directa, de onde provém a radiação infravermelha como uma das formas de aquecimento da divisão sempre que a temperatura interior seja baixa [1].

Dada a importância da iluminação e dos estores para o controlo do ambiente na habitação, e tendo também em consideração que são os sistemas preferenciais para os consumidores, serão estes que integrarão o projecto domótico.

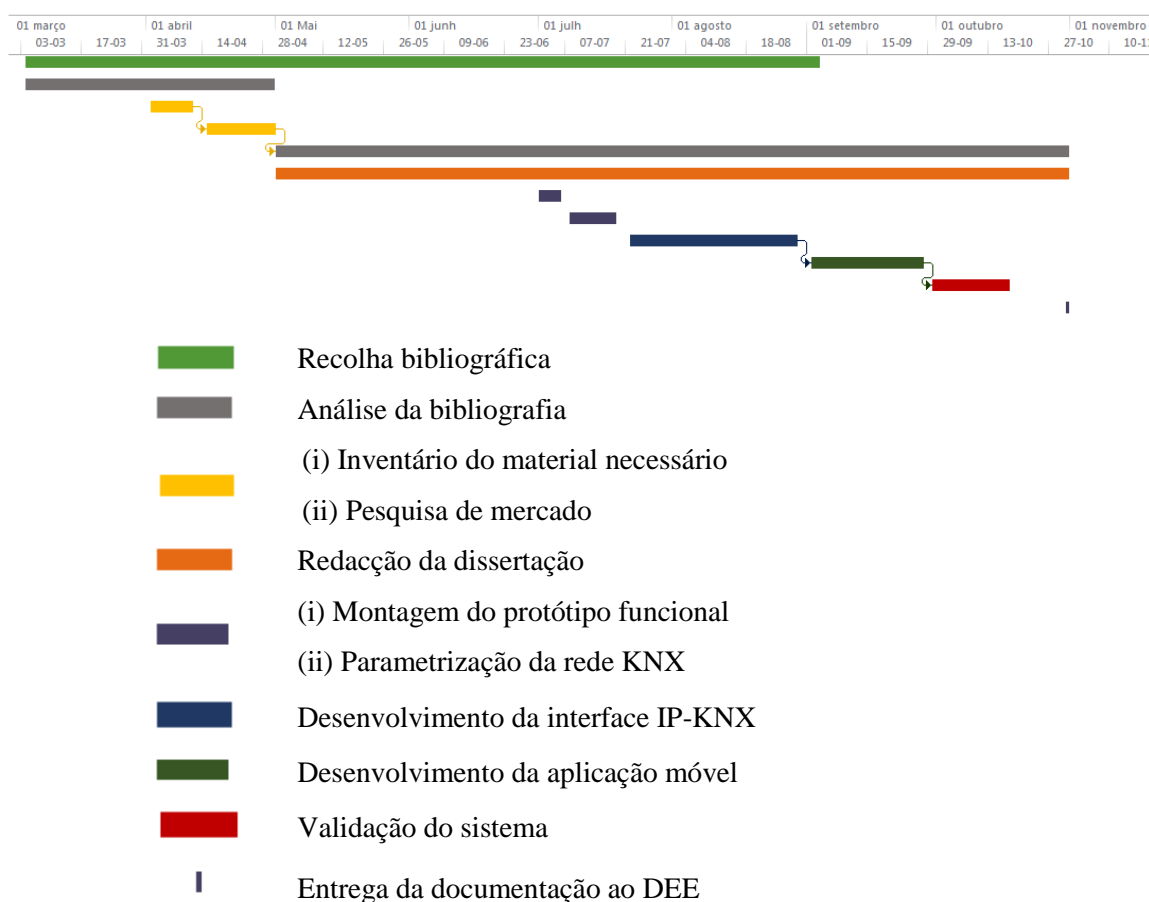
Tomando em consideração o perfil dos clientes deste mercado, as suas pretensões e preferências, e o domínio do SO Android no mercado das telecomunicações por *smartphone*, este trabalho pretende convergir essas tendências na criação de um sistema de acesso remoto a uma infra-estrutura de automação residencial, composta por actuadores de

lâmpadas e de estores, via internet, por uma aplicação Android, para apresentar um protótipo de uma *connected home*.

1.3. GESTÃO DO TRABALHO

Visando projectar um produto de forma eficiente e eficaz, para além da definição clara dos objectivos, é imperativa a definição de metas e o escalonamento de tarefas. O diagrama de Gantt da Tabela 1 apresenta as actividades de desenvolvimento do projecto e as suas interdependências, pretendendo também representar o avanço das suas diferentes etapas.

Tabela 1 Diagrama de Gantt do projecto.



1.4. ORGANIZAÇÃO DO MANUSCRITO

Esta dissertação está organizada em 6 capítulos e pretende guiar o leitor pelas etapas de desenvolvimento do trabalho.

Este capítulo enquadra a automação em edifícios residenciais, normalmente conhecida como domótica, apresentando as motivações e objectivos do presente trabalho. No Capítulo 2 são apresentadas as funcionalidades de um sistema domótico, apresentando alguns dos protocolos de comunicação mais utilizados nesta área e algumas soluções comercializadas no mercado nacional. De seguida é feito um levantamento das funcionalidades e requisitos de qualidade que um sistema domótico moderno pode oferecer ao seu utilizador. O Capítulo 3 define a arquitectura do projecto, os seus requisitos de implementação, e as ferramentas necessárias ao seu desenvolvimento. O Capítulo 4 apresenta a implementação do trabalho, descrevendo o projecto electrónico e de controlo, recorrendo a diagramas, fluxogramas e excertos de código. O capítulo 5, “Verificação e validação”, apresenta a análise dos resultados bem como as metodologias utilizadas. Finalmente, o capítulo 6 apresenta as conclusões deste trabalho enumerando as funcionalidades e melhorias podem acrescentar valor ao sistema criado, que não foram implementados.

2. SISTEMA DOMÓTICO

O capítulo anterior introduziu a domótica, enquadrando-a nos propósitos do presente projecto. Este capítulo aborda: *(i)* o estado da arte na área da domótica e *(ii)* o controlo de um ambiente domótico. Subdivide-se em duas secções de forma a abordar separadamente as funcionalidades de um sistema domótico e alguns dos protocolos utilizados nessa área, mas também os sistemas de controlo do ambiente domótico, focando-se, nesta última secção, na definição das características que um sistema desta natureza deve reunir.

No final da década de 1970, iniciou-se, nos Estados Unidos, uma grande onda de soluções voltadas para a automação residencial, em especial para atender as necessidades de deficientes físicos, proporcionando-lhes um maior conforto, mobilidade e segurança.

Com o advento dos computadores pessoais, da internet e com a proliferação de telemóveis e outras tecnologias nos anos 90, a procura dos consumidores de sistemas de domótica aumentou. Desta forma, com o despertar do interesse nesta área, também devido ao aumento da faixa etária da população, apareceram várias tecnologias com o intuito de trazer conforto à habitação e de reduzir barreiras que dificultassem as actividades das pessoas dessa faixa cada vez mais numerosa, impulsionando-se, assim, o desenvolvimento deste ramo da automação [10].

2.1. SISTEMA DOMÓTICO

“Um sistema é uma combinação de componentes que actuam conjuntamente e realizam um certo objectivo”; *“Um sistema de controlo realimentado é aquele que tende a manter uma relação prescrita entre a saída e a entrada de referência comparando-as e utilizando a diferença como um meio de controlo”*, Katsuhiko Ogata [11].

Um sistema domótico é composto por diversas funcionalidades que dotam a residência de grande conforto e comodidade. Tipicamente é composto por uma rede de comunicação que permite a interligação de um conjunto de equipamentos e outros subsistemas, com o objectivo de obter informações sobre o ambiente em que está inserido e efectuar o respectivo controlo [12].

As subsecções seguintes procuram, de uma forma genérica, revelar as funcionalidades de um sistema domótico e alguns dos protocolos de comunicação nele utilizados, finalizando com a apresentação de alguns produtos comerciais utilizados nesta área.

2.1.1. FUNCIONALIDADES

A nível doméstico destacam-se as seguintes funcionalidades [2] de um sistema domótico, algumas representadas na Figura 2:

- Controlo do ambiente:
 - Iluminação (interna e externa);
 - Persianas, cortinados e toldos;
 - AVAC;
- Controlo energético;
- Sistemas de segurança:
 - Detecção de inundação;
 - Detecção de incêndio;
 - Detecção de gases tóxicos;
 - Detecção de intrusos (videovigilância; sensor de presença);
 - Simulação de presença;
- Sistemas de som e imagem;
- Controlo de fechaduras;
- Controlo de tomadas;

- Controlo de electrodomésticos;
- Controlo do sistema de rega.

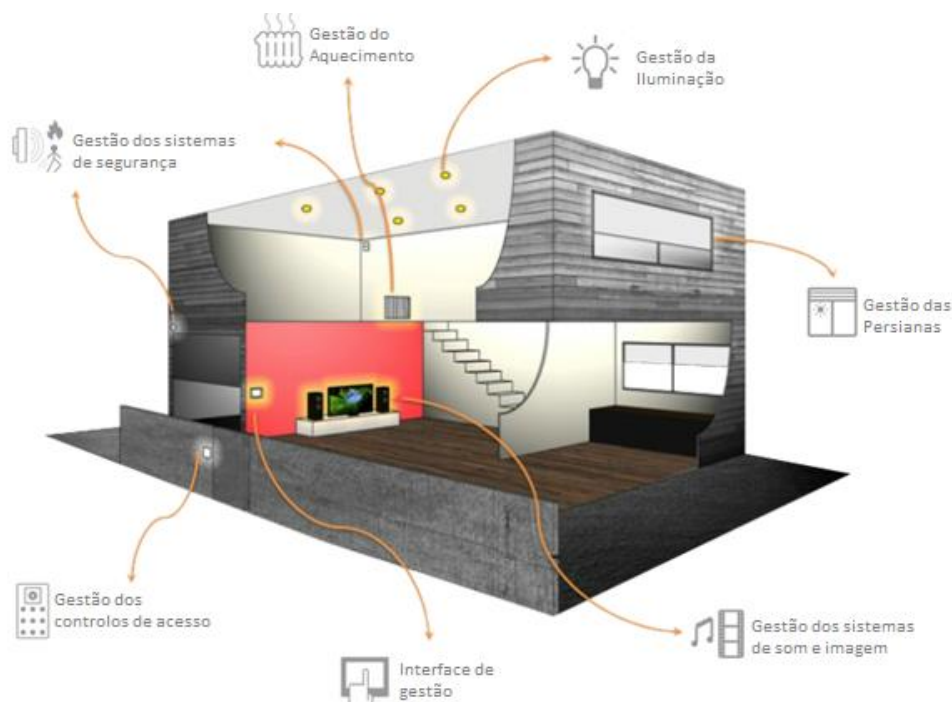


Figura 2 Representação de algumas funcionalidades de uma casa automatizada [9].

2.1.2. PROTOCOLOS DE COMUNICAÇÃO

Como referido anteriormente, um sistema domótico é constituído por uma rede de comunicação (com ou sem fios) que interliga os elementos distribuídos no edifício, por meio de um comando ou por informações capturadas no ambiente, conduzindo-as aos dispositivos que podem executar acções, sinalizar ou fornecer dados de leitura. Essa rede serve ainda para a transmissão de sinais de dados, telefonia, áudio e vídeo. Para tal, é necessária a instalação de sistemas e protocolos apropriados que permitam garantir a conectividade e integração entre as múltiplas funcionalidades dos dispositivos da instalação doméstica [12]. Em automação essa capacidade designa-se de interoperação.

A domótica despertou grande interesse em diversos grupos económicos, levando ao aparecimento de vários sistemas que se constituíram como normas¹. Contudo, a grande diversidade de sistemas existentes, cada um com o seu protocolo de comunicação, levou à

¹ Uma norma é um documento estabelecido por consenso e aprovado por um organismo reconhecido, que fornece para uso comum e repetitivo, regras, directrizes ou características para actividades ou seus resultados, visando a obtenção de um óptimo grau de coordenação num dado contexto [13].

existência de problemas quanto à compatibilidade ao nível da interoperabilidade [14], verificando-se que uma norma uniforme para o controlo dos diversos dispositivos existentes num edifício facilitaria a implementação de funcionalidades inovadoras e complexas. Assim, um dos factores de selecção da tecnologia prende-se com o nível de interoperação [4].

O objectivo principal da interoperação é oferecer aos utilizadores uma ampliação de resultados. Um exemplo da interoperação poderá verificar-se quando os sensores de vigilância de uma residência, conjuntamente com outros elementos da instalação, captam uma ameaça de intrusão e accionam, imediatamente, a iluminação de alguns ambientes e/ou o sistema de irrigação do jardim com o propósito de dissuadir o intruso. Ao mesmo tempo, o sistema de vídeo pode accionar a gravação e apresentar a imagem do invasor, em tempo real, num canto do ecrã da televisão, e, em simultâneo o sistema de áudio pode emitir um alerta sonoro. Persistindo a invasão, o sistema poderá providenciar a abertura da porta do canil para libertar os cães, travar as portas de acesso da residência e executar ainda uma ligação telefónica para a polícia ou para outro contacto previamente programado [12].

Um outro factor a ter em conta é a utilização (ou não) de protocolos² abertos, em que o fabricante é livre de desenvolver e comercializar novos produtos, desde que sejam cumpridos os requisitos tecnológicos de domótica. Esse facto mostra uma outra particularidade dessas tecnologias, ou seja, para uma qualquer funcionalidade ou requisito necessário, haverá sempre um ou mais produtos que conseguirão corresponder às expectativas [4].

A Tabela 2 apresenta, de forma sucinta, alguns dos protocolos de rede utilizados em aplicações de domótica: o BatiBUS, o *European Installation Bus* (EIB), o *European Home System* (EHS), o *Consumer Electronics Bus* (CEBus) e o *Home Radio Frequency Working Group* (Home RF), e os meios físicos utilizados por cada protocolo.

² H. Merz, T. Hansemann e C. Hübner [26] definem protocolo como “um conjunto de regras que gere como dois dispositivos comunicam entre si”.

Tabela 2 Protocolos de rede utilizados na domótica.

<i>Standard</i>	Meio Físico	Descrição
Protocolos de Rede – Alianças e Grupos de trabalhos		
<i>BatiBUS</i>	TP0 – 4800 bps ^[12]	Sensores de união e actuadores para construção de sistemas que controlem AVAC, sistemas de segurança e acesso.
<i>EIB</i>	TP1 – 9600 bps ^[12]	Sensores e actuadores para construir sistemas que controlem AVAC, sistemas de segurança e acesso.
<i>EHS</i>	Todos	Uma colaboração entre indústrias e governos europeus sobre Domótica. Um dos objectivos é ser uma norma na europa de um barramento comum (EHS).
<i>CEBus</i>	Todos	Protocolo criado pela <i>Electronic Industries Association</i> (EIA) para ser possível a interligação e comunicação entre dispositivos electrónicos da casa.
<i>HomeRF</i>	RF	A missão deste grupo de trabalho é tornar possível uma vasta gama de produtos electrónicos de consumo que operem entre si, estabelecendo uma especificação aberta para comunicações digitais de RF (sem licença), para PC's e produtos electrónicos de consumo em qualquer sítio e nos arredores da casa.
<i>LonMark Interoperability Association</i>	Todos	Missão: integrar facilmente dispositivos baseados em redes <i>LonWorks</i> , fazendo uso de ferramentas e componentes normalizados.
<i>ZIGBEE</i>	RF	É uma versão melhorada do <i>HomeRF</i> e destinada a ambientes industriais. Pensa-se que esta pode ser uma das normas que irá ser bastante utilizada no mundo da domótica.
Protocolos de Rede – Proprietários		
<i>Lonworks Echelon Corp.</i>	Todos	Redes de controlo comerciais e para a casa. Uma rede <i>LonWorks</i> é um grupo de dispositivos que trabalham juntos para sensoriar, monitorizar, comunicar e, de alguma forma controlar.
<i>X-10</i>	Corrente eléctrica/ RF	Utiliza a rede eléctrica e facilita o controlo de dispositivos domóticos sem instalação de qualquer fio em casa, uma vez que utiliza a instalação eléctrica já existente.

Nota: Adaptado de [15].

Entre os protocolos aqui apresentados os que têm uma maior disseminação são o X10 e o EIB. O X10 é predominantemente utilizado na América do Norte, enquanto o EIB tem maior implantação na Europa [16]. O protocolo X10 oferece uma elevada flexibilidade a baixo custo, parecendo dar resposta ao segmento de mercado de menor investimento nesta área, ao passo que os sistemas baseados no protocolo EIB se enquadram no segmento mais exigente e de maior investimento [4].

O Konnex (KNX) resulta da união de três associações europeias: *BatiBus Club International* (BCI), *European Installation Bus Association* (EIBA) e *European Home Systems Association* (EHSA) com o intuito de promover um padrão único e aberto para o mercado de redes de domótica na Europa. Esta associação foi fundada em 1999 por nove empresas do sector de tecnologia, dentre elas a *Bosch Telecom*, a *Electrolux* e a *Siemens* [17], contando hoje com cerca de 4200 fabricantes afiliados, o que contribui para o seu nível de interoperabilidade [1].

O KNX é a única norma internacional aberta para o controlo de edifícios que cumpre com as normas europeias CENELEC EN5009 e CEN EN 13321-1, e está também aprovado como norma internacional ISO/IEC 14543-3. Esta norma, além de oferecer especificações para a automação de equipamentos de instalação eléctrica, oferece também soluções para aplicação AVAC, o que, aliado à sua óptima flexibilização e capacidade de interoperabilidade, lhe confere um elevado nível funcional [17].

Apesar de ter como base o padrão EIB, que usa apenas um par trançado como meio físico, o KNX especifica no seu modelo vários componentes e mecanismos de rede, permitindo uma ampla escolha de configuração dependendo das necessidades e independentemente do modo de configuração [19]. Este grande conjunto de meios físicos de comunicação entre equipamentos confere ao sistema uma grande flexibilidade de utilização: este é o sistema com mais potencialidades e que mais tem evoluído a nível mundial [20].

2.1.3. SOLUÇÕES COMERCIALIZADAS NO MERCADO NACIONAL

Existem várias soluções de sistemas domóticos disponíveis no mercado nacional. A Tabela 3 procura resumir as funcionalidades dos sistemas domóticos, comercializados em Portugal, mais relevantes nas feiras internacionais, organizando as suas características nas áreas: Campo de aplicação do sistema, Expansibilidade, Capacidade de interligação com outros sistemas e interfaces de controlo.

Tabela 3 Soluções de sistemas domóticos mais relevantes no mercado nacional.

Sistema		VIVIMAT	DOMUS-INT	SIMON	OMTRONIC	CARDIO	
Aplicação	Apartamentos	S	S	S	S	S	
	Vivendas	S	S	S	S	S	
	Edifícios	N	S	S	N	N	
	Indústria	N	S	S	N	N	
Construção	Nova	S	S	S	S	S	
	Existente	N	N	N	S	N	
Interligação	EIB	N	N	N	N	N	
	X10	N	N	N	S	S	
Expansibilidade	Nº Máximo de endereços	48 IN + 56 OUT	Ilimitado	128 IN + 128 OUT	100	200	
	Expansibilidade Futura	N	Difícil	Difícil	S	N	
Controlo	Telefone	S	S	S	N	N	
	PC	S	Internet	Local	N	S	
	Telecomando	N	IV	IV	RF	IV	
	Wi-Fi	S	S	N	N	N	
	Iluminação	Regulação	S	N	S	S	S
		Cenários	S	S	S	S	S
S = Sim; N = Não; IV = Infravermelhos; RF = Radiofrequência.							

Nota: Adaptado de [4].

Analisando a Tabela 3 verifica-se que apenas o *Domus-Int* e o *Simon* podem ser aplicados a todos os tipos de edifícios, não havendo nenhum sistema capaz de se interligar a uma rede EIB. Quanto ao número máximo de endereços, apenas o *Domus-Int* é ilimitado, o que lhe confere vantagem em edifícios com maior número de componentes. No que diz respeito à expansibilidade, o sistema *Omtronic* destaca-se pois é um único que a permite,

sem dificuldade. Em última análise, nenhum dos sistemas permite o controlo em todas as tecnologias.

2.2. CONTROLO DO AMBIENTE DOMÓTICO

Um ambiente domótico consiste num conjunto de equipamentos domésticos interligados por uma rede, disponível a um grupo de utilizadores.

Os ambientes domóticos podem ser compostos por diferentes contextos naturais. Os contextos são caracterizados pela sua variabilidade e mudança. A variabilidade corresponde às diferenças entre os elementos de contextos diferentes, por exemplo, a cozinha e a sala de um apartamento, ou as salas de dois apartamentos, não terão o mesmo número e tipo de equipamentos, da mesma forma que o radiador da sala não terá o mesmo funcionamento que o da cozinha. A mudança corresponde às alterações de elementos que compõem um dado contexto, isto porque os equipamentos são aparelhos eléctricos que podem ser adicionados ou suprimidos, estando também sujeitos, em qualquer momento, a avarias.

Consideremos que o ambiente está subdividido num conjunto de contextos que definem a sua organização: cada contexto delimita um lugar no qual se encontram os equipamentos e os utilizadores, podendo ser composto por outros contextos. Assim, um utilizador é uma pessoa física que interage com os equipamentos disponíveis no ambiente domótico, em função dos seus direitos de acesso, para satisfazer as suas necessidades. Nesse âmbito, um cenário é a execução combinada de um conjunto de serviços num dado momento, para satisfazer uma necessidade complexa do utilizador.

Assim, os ambientes domóticos são compostos principalmente por três elementos: os utilizadores, os equipamentos e o sistema domótico. De forma a dar resposta às necessidades dos utilizadores, um sistema domótico deve estar à altura de gerir as características dos ambientes domóticos no que toca à variabilidade e às alterações dos contextos [21].

As próximas subsecções apresentam: *(i)* as funcionalidades de um sistema de controlo domótico e *(ii)* os requisitos de qualidade que devem pautar um sistema desta natureza.

2.2.1. FUNCIONALIDADES DE UM SISTEMA DE CONTROLO DOMÓTICO

Podemos definir dois tipos de funcionalidade que um sistema de controlo domótico deve oferecer. As funcionalidades explícitas, que permitem aos utilizadores gerir as suas necessidades, e as funcionalidades implícitas, que permitem aos sistemas realizar as necessidades dos utilizadores em resposta às alterações ligadas ao ambiente e ao sistema [21].

i. FUNCIONALIDADES EXPLÍCITAS DOS SISTEMAS DE CONTROLO DOMÓTICO

A Tabela 4 apresenta quatro funcionalidades explícitas que os sistemas domóticos devem oferecer aos utilizadores. Estas funcionalidades são fornecidas através de interfaces gráficas que permitem aos utilizadores parametrizar o comportamento do sistema em função das suas necessidades.

Tabela 4 Funcionalidades explícitas dos sistemas domóticos.

Funcionalidades explícitas dos sistemas domóticos	
Gestão de Serviços	Permitem que os utilizadores finais possam invocar os serviços fornecidos pelos equipamentos disponíveis no ambiente, em função dos seus direitos de acesso. É o sistema quem se encarrega da execução dos serviços.
Gestão de cenários	Os utilizadores finais devem poder gerir (definir, modificar ou eliminar) cenários personalizados em função dos seus direitos de acesso e dos equipamentos disponíveis no ambiente. O sistema encarrega-se da automatização da execução dos cenários e da suspensão dos cenários eliminados.
Gestão dos perfis dos utilizadores	Um perfil permite que o sistema identifique o utilizador e personalize a interacção com ele em função do seu papel e dos seus direitos de acesso aos serviços fornecidos pelos equipamentos. Os utilizadores privilegiados devem ser capazes de gerir (definir, modificar ou eliminar) os perfis de utilizadores.
Gestão da parametrização do sistema	Os administradores do sistema devem poder parametrizar a arquitectura do sistema por razões que podem estar ligadas à <i>performance</i> ou à fiabilidade deste. A modificação da arquitectura do sistema não deve perturbar nem interromper os cenários em execução.

Nota: Adaptado de [21].

ii. FUNCIONALIDADES IMPLÍCITAS DOS SISTEMAS DE CONTROLO DOMÓTICO

Para além das funcionalidades explícitas, definem-se duas funcionalidades implícitas que os sistemas domóticos devem implementar, abordadas na Tabela 5. Essas funcionalidades são transparentes para o utilizador e permitem ao sistema manter a sua melhor coerência, a sua *performance* e a sua fiabilidade [22], possibilitando, assim, a gestão automática da sua adaptação às alterações relacionadas com equipamentos ou à sua própria arquitectura. As suas funcionalidades deverão permitir que o sistema seja tolerante a avarias de equipamentos dos seus próprios elementos. Os cenários em curso de execução não devem ser interrompidos nem perturbados e os utilizadores devem ser capazes de interagir com o sistema a qualquer momento [21].

Tabela 5 Funcionalidades implícitas dos sistemas de controlo domótico.

Funcionalidades implícitas dos sistemas de controlo domótico	
Gestão do ambiente	<p>Gestão da adição e da supressão/avaria de equipamentos:</p> <p>Após a supressão/avaria de um equipamento, os utilizadores não deverão ter acesso ao seu controlo, devendo os serviços fornecidos pelo equipamento ser suprimidos das interfaces gráficas; os cenários que não possam ser executados devido a essa supressão deverão ser suspensos.</p> <p>Após a adição de um equipamento, o sistema deverá ser capaz de o controlar e deverá verificar se os cenários suspensos devido à avaria de um equipamento podem ser reactivados.</p>
Gestão do sistema	<p>Após a adição ou a supressão/avaria de um elemento do sistema, este deverá ser capaz de reconfigurar a sua arquitectura. Isto significa que deverá ser capaz de adaptar dinamicamente a repartição de controlo dos equipamentos e de execução de cenários para explorar melhor os seus recursos.</p>

Nota: Adaptado de [21].

2.2.2. REQUISITOS DE QUALIDADE DE UM SISTEMA DE CONTROLO

Nesta secção pretende-se proceder à apresentação dos requisitos de qualidade indispensáveis que um sistema de controlo domótico deve possuir para realizar as suas funcionalidades. São eles: (i) a configurabilidade, (ii) a capacidade de adaptação dinâmica,

(iii) a capacidade de adaptação autónoma e (iv) a sua sensibilidade ao contexto, todas elas discriminadas na Tabela 6.

Tabela 6 Requisitos de qualidade de um sistema de controlo doméstico.

Requisitos de qualidade de um sistema de controlo doméstico	
Configurabilidade	Os sistemas domésticos devem oferecer funcionalidades aos utilizadores através de interfaces gráficas. As interfaces e seus conteúdos devem ser adaptados em função do perfil dos utilizadores, do ambiente e do sistema num dado momento. Os utilizadores devem ser capazes de configurar a arquitectura do sistema (gestão da parametrização do sistema), de personalizar o seu comportamento em função do utilizador (gestão dos perfis de utilizador) e em função dos equipamentos (gestão de serviços e de cenários).
Adaptação dinâmica	O sistema deve permitir a adaptação dinâmica da sua arquitectura para o controlo dos equipamentos, a implementação de cenários e a aplicação da parametrização. A adaptação dinâmica consiste em efectuar modificações na arquitectura do sistema durante o seu funcionamento sem o perturbar.
Adaptação autónoma	O sistema deve ser capaz de se adaptar automaticamente aos equipamentos disponíveis e aos cenários definidos pelos utilizadores. Deve também gerir autonomamente as funcionalidades implícitas (adaptação às mudanças que se produzem no ambiente do sistema). A adaptação autónoma consiste na integração de mecanismos que permitem ao sistema uma adaptação automática da sua arquitectura sem intervenção humana. Isso permite ao sistema gerir o ambiente sem conhecimento <i>a priori</i> dos equipamentos disponíveis e das necessidades dos utilizadores.
Sensibilidade ao contexto	A fim de ter qualidades precedentes, um sistema doméstico deve ser capaz de identificar os utilizadores, os equipamentos disponíveis no ambiente e os diferentes elementos que compõem o sistema. Deverá igualmente ser capaz de detectar as alterações que se produzem tanto ao nível do ambiente como do sistema. A sensibilidade ao contexto permite ao sistema detectar alterações e reagir adaptando o seu comportamento.

Nota: Adaptado de [21][22][23].

A evolução destes sistemas caminha em direcção às tecnologias baseadas nos modelos de redes neurais. Estas redes são compostas de dispositivos artificiais que se baseiam nos mecanismos de aprendizagem, inspirados no cérebro humano. Por exemplo, o sistema de automação poderá, após repetidos comandos, “aprender” a regular a temperatura de

refrigeração do ambiente e ligar uma cafeteira, antecipando o retorno de um morador à sua residência permitindo, assim, que ele encontre a temperatura da sua residência confortável e o seu café pronto no momento de ser tomado [12]. Seguindo a distinção dos tipos de casas apresentada no Capítulo 1, podemos afirmar que essa evolução se enquadra em casas do tipo *Learning e Alert Home*.

2.3. SÍNTESE

Este capítulo mostrou que um sistema domótico oferece inúmeras funcionalidades ao seu utilizador, potenciando o conforto da sua habitação. Para tal, os sistemas domóticos recorrem a protocolos de comunicação para que os seus componentes possam interagir. Na selecção de um protocolo de comunicação é importante ter em consideração a interoperabilidade e a sua abertura, havendo, nos dias de hoje, um vasto leque de opções para instalação de sistemas domóticos, contemplando vários meios físicos, acessíveis a todos os bolsos.

Para responder às necessidades dos utilizadores, um sistema de controlo domótico deverá fornecer-lhes funcionalidades explícitas (gestão de serviços, gestão de cenários, gestão de perfis de utilizadores e gestão da parametrização do sistema) e deverá, também, operar automaticamente as funcionalidades implícitas (gestão do ambiente e do sistema). De forma a realizar as funcionalidades precedentes, um sistema domótico deverá ser configurável, de adaptação automática e dinâmica e sensível ao contexto.

O próximo capítulo apresenta a arquitectura geral de um sistema domótico, controlado via internet por um *smartphone*.

3. DEFINIÇÕES TÉCNICAS DO SISTEMA

No capítulo anterior foram apresentadas as funcionalidades que um sistema de controlo domótico moderno pode oferecer. O presente capítulo apresenta as definições técnicas do sistema a desenvolver, indispensáveis para a sua posterior implementação, guiando o leitor desde a definição dos requisitos do sistema até ao esboço da rede domótica, da arquitectura do sistema e consequente selecção de componentes, bem como as definições da interface gráfica e as ferramentas necessárias para a sua implementação (apresentada no capítulo seguinte). Assim, este capítulo divide-se em sete secções, das quais a primeira é destinada à definição de requisitos, pretendendo as seguintes abordar, em separado, os requisitos de implementação entretanto inventariados.

3.1. REQUISITOS DA IMPLEMENTAÇÃO DO SISTEMA

A finalidade deste sistema é permitir que um ou vários utilizadores de um edifício inteligente o controlem remotamente via internet através de um *smartphone* com SO Android. Para tal, o sistema de controlo a desenvolver deve funcionar em tempo real, ser robusto, seguro e intuitivo.

Tal como já foi referido no capítulo 2, um sistema domótico é composto por uma rede de comunicação que permite a interligação dos equipamentos que a constituem. Deste modo, é necessária a selecção de um protocolo de comunicação, que possibilite a interoperação de todos os participantes da rede e que preveja uma possível expansão, pois a implementação de um sistema com um elevado nível de flexibilidade permite que o barramento seja reprogramável a baixo custo [4].

A tecnologia domótica KNX é uma norma aberta que, para além de permitir o uso de todos os meios físicos existentes, é sempre ampliável com a integração de novos módulos e/ou com a parametrização de novas funções [24].

Adoptando as considerações anteriores e uma vez que o Instituto Superior de Engenharia do Porto (ISEP) possui, num dos seus laboratórios, componentes que fazem uso do protocolo KNX, tomou-se este protocolo como melhor opção para a comunicação da rede doméstica.

“Boas tecnologias de comunicação não garantem automaticamente uma boa comunicação. (...) É necessário um tradutor para garantir que duas partes comuniquem directamente”, H. Merz, T. Hansemann e C. Hübner [26].

Nestas condições, sendo o objectivo final o controlo da rede doméstica via internet, e adoptando o KNX, é imperativo o uso de uma interface IP-KNX que estabeleça a ligação entre a rede KNX e a internet. Este trabalho centra-se na construção de uma interface IP-KNX, criando os meios necessários à sua validação (aplicação móvel constituída por um servidor web e um cliente Android).

Quanto à interface gráfica num dispositivo móvel com o SO Android, esta deve ter como funções mínimas a ligação ao endereço IP do servidor WEB da rede domótica a controlar, a validação dos dados de acesso dos utilizadores e o envio de comandos de controlo para a rede KNX. Dada a necessidade de assegurar a segurança de acesso à rede KNX, e possíveis expansões da aplicação, é necessária a utilização de uma base-de-dados (BD) que armazene, entre outras, as informações de cada utilizador.

A Figura 3 ilustra a relação destes requisitos de implementação que são alvo de atenção nas secções seguintes.

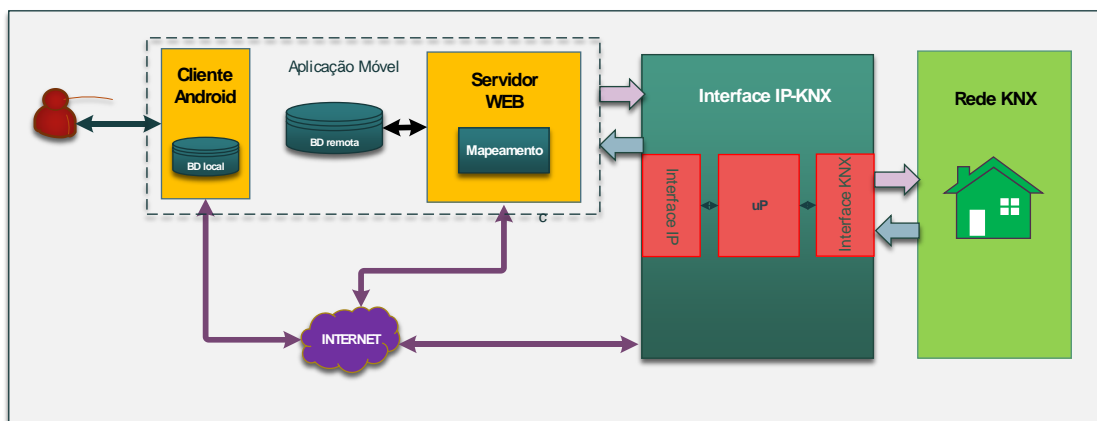


Figura 3 Diagrama do sistema – relação dos requisitos de implementação.

3.2. NORMA KNX

Tal como referido no capítulo 2, o KNX suporta vários meios físicos. Este documento refere-se apenas ao meio TP1 (9600 bps).

Uma instalação KNX.TP faz uso de um único barramento para interligar todos os equipamentos, tal como representado na Figura 4. Este barramento de dados é maioritariamente implementado de forma descentralizada, no entanto, se necessário, é possível a sua implementação em aplicações centralizadas [25].

Esta distribuição descentralizada é implementada nos dispositivos transmissores e receptores que possuem o seu próprio microcontrolador [26], podendo assim comunicar directamente uns com os outros sem recorrer a nenhuma hierarquia ou a um dispositivo de supervisão. Desta forma, a organização descentralizada traz mais flexibilidade ao sistema [25].

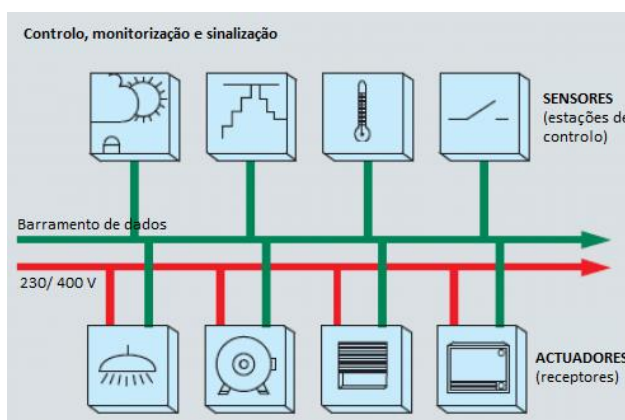


Figura 4 Instalação KNX (adaptado de [18]).

Os dispositivos do barramento de dados são compostos por uma *bus coupling unit* (BCU) e um módulo de aplicação/terminal. A informação a ser processada é transferida do barramento para a BCU. A BCU recebe e transmite dados, garante a alimentação eléctrica ao sistema electrónico e guarda informações importantes tais como o endereço físico actual, um ou mais endereços de grupo e também o *software* de aplicação com os parâmetros do dispositivo [27].

A Figura 5 resume a estrutura de uma instalação KNX. Como se pode verificar, a instalação precisa de um meio de transmissão (linha verde) para a comunicação dos seus participantes. Cada participante possui uma BCU que gere a interface entre o módulo de aplicação e o barramento de dados. Por sua vez os módulos de aplicação dependem do seu *software*, desenvolvido e fornecido pelo fabricante, para que com a ferramenta *EIB Tool Software* (ETS), o dispositivo possa ser integrado e parametrizado na rede KNX.

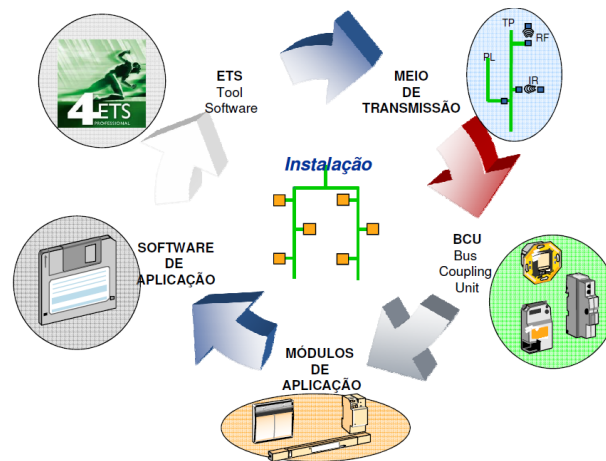


Figura 5 Estrutura de uma instalação KNX [18].

Esta secção não tem a pretensão de ser um manual sobre especificações KNX, mas apenas de dar ao leitor uma visão geral das características deste protocolo.

3.2.1. ARQUITECTURA

Os dispositivos (ou nós) que trocam dados através de um barramento ou rede estão geometricamente dispostos e ligados uns com os outros de uma forma específica. Esta disposição é conhecida como topologia do barramento ou da rede [26].

A Figura 6 apresenta as várias topologias possíveis para uma rede KNX, são elas linear, em estrela e em árvore, ou uma combinação de várias topologias, e neste último caso denomina-se de topologia mista.

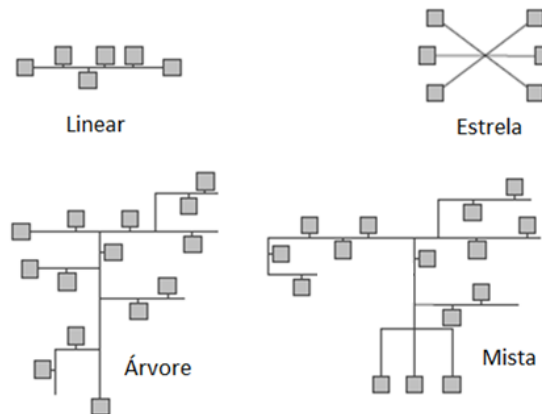


Figura 6 Topologias possíveis da rede KNX (adaptado de [25]).

A herança do meio físico EIB deu ao KNX uma divisão hierárquica da rede em áreas, linhas e nós, bem como uma facilidade para fins de comissionamento, diagnóstico e manutenção [27].

Uma rede KNX é capaz de abarcar 14400 dispositivos (64 x 15 x 15). Isto porque a sua topologia lógica permite 64 dispositivos em cada linha, podendo haver um máximo de 15 linhas agrupadas numa linha principal, constituindo assim uma área. A rede KNX prevê um máximo de 15 áreas interligadas por uma linha de *backbone* (Figura 7) [25][26][27]. No entanto, o protocolo KNX permite o uso de repetidores para a ligação de dois segmentos. Assim, permitindo um máximo de 4 repetidores, faz com que cada linha passe a ter a capacidade de ligar até 256 dispositivos [25].

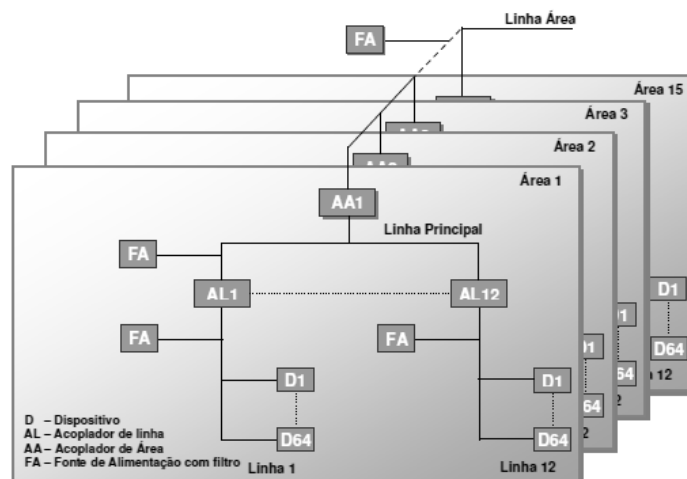


Figura 7 Topologia hierarquizada da rede KNX [18].

3.2.2. TELEGRAMA KNX

As informações, *e.g.* comandos de comutação e mensagens, são trocadas entre os dispositivos na forma de telegramas. Um telegrama consiste numa série de caracteres combinados em campos [27]. A Figura 8 representa um telegrama KNX formado pelos campos de controlo, endereçamento, dados e *checksum*.

Os campos controlo e *checksum* são necessários para garantir o tráfego sem erros do telegrama e são analisados pelo dispositivo receptor. O campo de endereçamento contém o endereço de origem e de destino. O endereço de origem é um endereço físico. O endereço de destino determina os dispositivos com quem a origem quer comunicar.

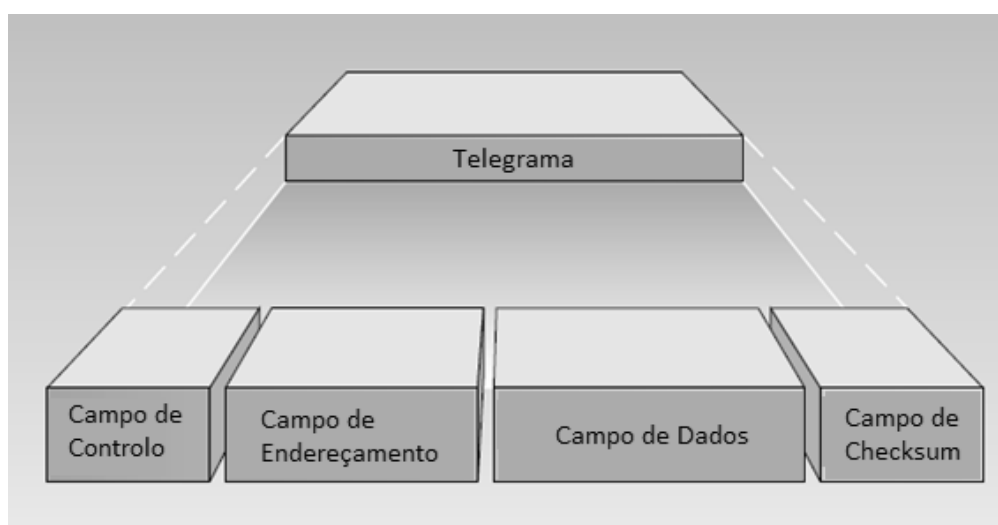


Figura 8 Telegrama KNX (adaptado de [27]).

De um modo geral, o sensor envia um telegrama para o barramento KNX. Esse telegrama é lido pelos actuadores pertencentes ao grupo de comunicação e estes actuam de acordo com a informação nele contida.

O telegrama contém informações relativas ao dispositivo que estabelece a comunicação e informações úteis à acção. Essa informação é transmitida de forma agrupada em “palavras” de onze bits cada uma. Na Figura 9 está representada a estrutura de uma palavra. Como se pode verificar, esta começa por um bit inicial (ST), seguida de oito bits de dados (D0 até D7) e do bit de paridade (Pz), terminando com um bit final (SP). Uma vez passado o tempo de dois bits, chamado de pausa, segue-se uma nova palavra [28].

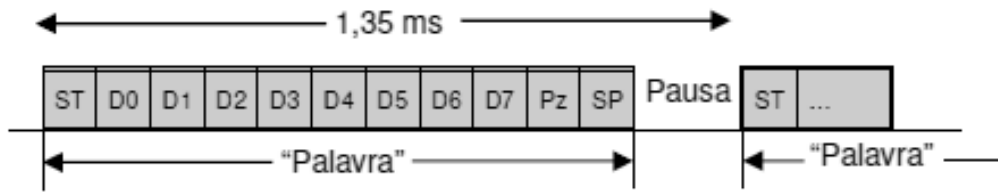


Figura 9 Estrutura de uma Palavra [28].

O telegrama é emitido com uma velocidade de 9600 bps, *i.e.*, um bit ocupa o barramento durante 1/9600 s ou 104 ms. Uma vez que a palavra é composta, no total, por 11 bits, incluindo os dois bits de pausa para a palavra seguinte, o tempo de transmissão de uma palavra é de 1,35 ms. O tempo de ocupação do barramento por uma informação varia entre 20 a 40 ms.

Quando vários dispositivos começam a emitir em simultâneo entra em acção um escalonamento de prioridades de transmissão [25].

Tal como referido anteriormente, o campo de endereçamento contém os endereços de origem e de destino do telegrama (Figura 10). O endereço de origem é composto por 16 bits, dos quais 4 definem a área em que se encontra o dispositivo emissor, os seguintes 4 definem a linha e os últimos 8 bits definem o próprio dispositivo. O endereço de destino é o endereço do (s) dispositivo (s) receptor. Este endereço contém 17 bits. Caso o bit 17 seja igual a 0, então o telegrama é dirigido exclusivamente a um dispositivo; caso seja igual a 1, o telegrama é dirigido a todos os dispositivos do grupo [28].

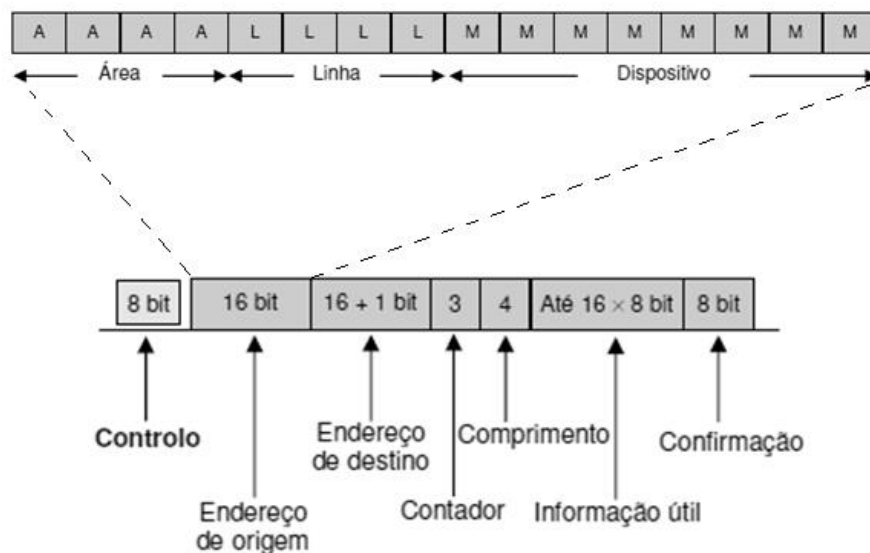


Figura 10 Bits do teleograma KNX e do endereço de origem (adaptado de [28]).

3.2.3. ENDEREÇAMENTO

A comunicação entre um sensor (*e.g.* um interruptor) e um actuador (*e.g.* um comutador de uma lâmpada) é uma sequência de operações. Neste caso, usando o protocolo KNX, um interruptor, definido unicamente pelo seu endereço físico, pode comunicar com várias lâmpadas usando um endereço de grupo.

O endereçamento de grupo baseia-se na troca de informação codificada com regras comuns entre os participantes da comunicação. Um emissor é capaz de transmitir um telegrama apenas para um único endereço de grupo. No lado oposto, um receptor pode ser assinante de vários endereços de grupo, sendo possível que faça a recepção de telegramas provenientes de emissores diferentes [25].

Os seja, os dispositivos instalados no barramento podem conter dois tipos de endereçamento:

- **Endereço Físico (modo de acesso ao meio)** - Cada dispositivo do barramento é identificado por um endereço físico único. Assim, não podem coexistir dois dispositivos instalados no barramento com o mesmo endereço físico [28]. O endereço físico de um dispositivo identifica-o e, ao mesmo tempo, revela a sua posição na topologia, pois é constituído pelos três campos da hierarquia separados por um ponto (Área.Linha.Nó) [25].
- **Endereço de Grupo (modo de acesso funcional)** – Os sensores e actuadores, para executarem comandos/acções centralizadas, reúnem-se através de endereços de grupo. Ou seja, os endereços de grupo são funções de dispositivos KNX pertencentes ao mesmo grupo que podem ser controladas por uma mensagem enviada pelo dispositivo fonte. No entanto as funções podem pertencer a vários grupos e podem ser activadas independentemente por cada dispositivo do grupo [28].

O KNX distingue dois tipos de endereços de grupo, constituídos pela hierarquia de grupo separada por uma barra:

- Endereçamento de grupo a dois níveis – Grupo Principal/Subgrupo
- Endereçamento de grupo a três níveis – Grupo Principal/Grupo Intermédio/Subgrupo.

Em ambos os endereçamentos de grupo, o grupo principal “0” está normalmente reservado para funções de alarme, enquanto os grupos principais de 1 a 13 estão reservados para sistemas como iluminação e estores [25].

3.2.4. INTEROPERAÇÃO

Os dispositivos KNX têm que conseguir comunicar entre si mesmo que sejam de fabricantes diferentes. Para tal, para além da utilização do mesmo protocolo, precisam de conseguir interpretar correctamente as mensagens enviadas por outros dispositivos.

A interoperação assenta na troca de dados com o mesmo formato. Para isso todos os intervenientes têm de utilizar o mesmo formato de mensagens.

Para garantir a interoperação entre equipamentos foram definidas normas *EIB Interworking Standards* (EIS) entretanto renomeadas *KNX Standardized Data Types*. Estas normas definem regras de interoperação de funções utilizadas pelos dispositivos, sendo seu objectivo a definição de valores e a interpretação de dados incluídos nos objectos de comunicação utilizados nas funções [25]. Os tipos de dados mais comuns estão normalizados a nível europeu e integrados na norma EN 50090.

Assim, existindo um formato normalizado para determinada função, de modo a obter a certificação do produto, o fabricante KNX é obrigado a usá-lo.

Este conceito de interoperação veio facilitar o desenvolvimento de *gateways* entre KNX e outros protocolos uma vez que o código proprietário pode ser facilmente mapeado para os formatos de dados comuns descritos na norma KNX [29].

3.2.5. ENGINEERING TOOL SOFTWARE - ETS

A associação KNX desenvolveu o ETS para configurar a infra-estrutura KNX. Este *software*, que é actualmente o mais usado para configuração de instalações, permite a administradores e engenheiros a possibilidade de criar a hierarquia do edifício, a topologia da rede e, finalmente, criar os grupos de objectos que representarão as funcionalidades entre os dispositivos [31].

O ETS faz uso de uma base de dados com a descrição de todas as informações necessárias sobre os dispositivos para a sua correcta configuração. Essa base de dados é criada pelo

fabricante de cada dispositivo utilizando uma ferramenta própria, o *manufacturer tool*, e só pode ser lida pelo ETS [32].

O ETS exporta projectos num arquivo composto por múltiplos ficheiros XML, tal como representa a Figura 11.

O ficheiro *knx_master.xml* contém a descrição de todos os tipos de *Datapoint (DTP)*. A topologia da rede, organização do edifício e endereços de grupo são armazenados no ficheiro *0.xml*. Finalmente, há uma pasta para cada fabricante, *Manufacturer ID*, que contem um ficheiro XML para cada tipo de dispositivo presente na rede. Esse ficheiro informa quais os DTPs disponíveis no dispositivo [31].

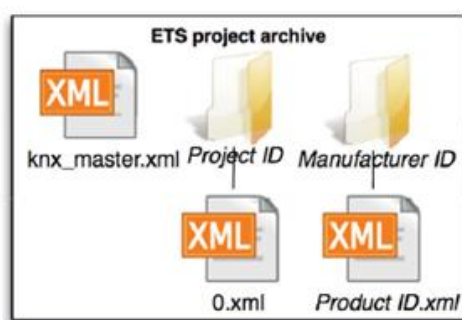


Figura 11 Estrutura do arquivo de um projeto do ETS [31].

Ao exportar um projecto obtém-se um ficheiro *.knxproj* que, na verdade, está num arquivo comprimido [33].

3.3. REDE KNX

A Figura 12 apresenta a fotografia da placa existente no ISEP com os componentes KNX instalados.

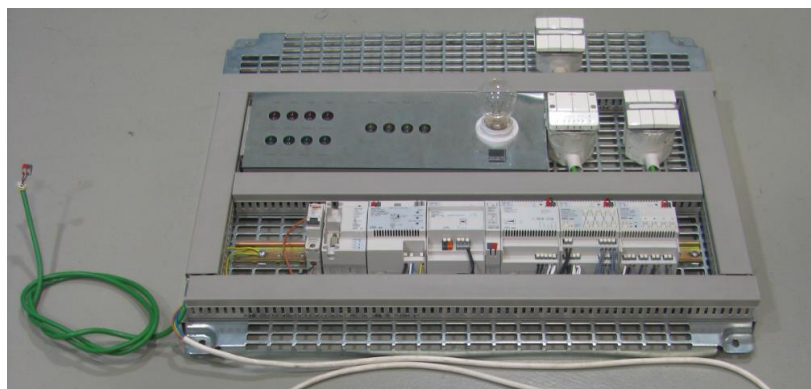


Figura 12 Placa de testes de rede KNX existente no laboratório do ISEP.

De forma a dar resposta aos requisitos do sistema, em função dos componentes a controlar, foram usados um actuador de comutação com quatro canais de saída (para quatro luminárias de dois estados), um *dimmer* universal (para uma luminária de intensidade variável), e um comutador de estores, também com quatro canais de saída.

Desta feita, atendendo ao material disponível no DEE, foram usados os actuadores N567/01, N528/02 e N523/03.

Uma vez que se pretende possibilitar um controlo remoto a partir da internet, é necessária a instalação de uma interface que permita a comunicação da rede KNX com o exterior. Para tal função, foi construída uma *interface* IP-KNX, cuja arquitectura será apresentada mais à frente.

Assim, a rede KNX terá a topologia apresentada na Figura 13.

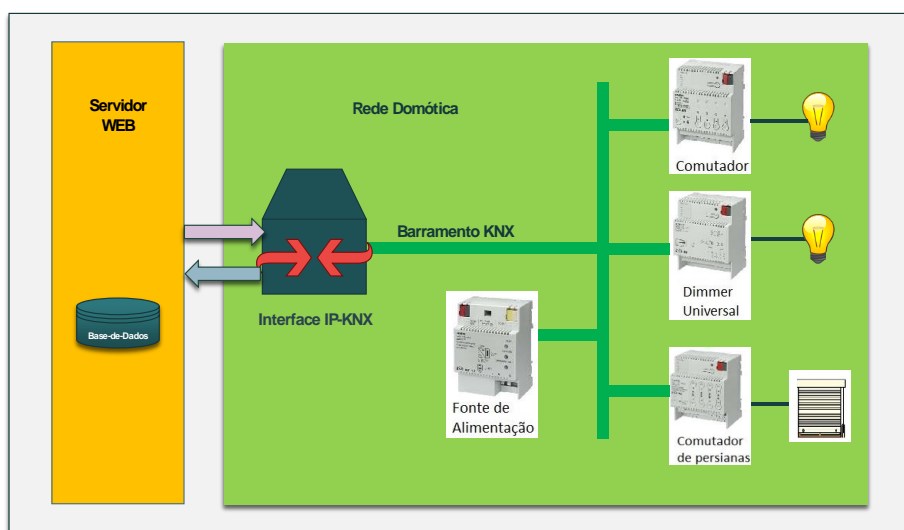


Figura 13 Rede KNX com ligação à internet através de uma *interface* IP/KNX.

3.4. INTERFACE IP-KNX

Para interligar a infra-estrutura web com a infra-estrutura domótica é necessária uma interface IP-KNX. Em termos funcionais, essa interface traduz os pedidos enviados pelo cliente Android ao servidor web em telegramas KNX que serão enviados para a rede KNX.

Para dar resposta aos objectivos traçados, o dispositivo precisa de um bloco de controlo capaz de processar dados e de comunicar com as redes KNX e Ethernet. Para tal, é necessário um microcontrolador (MCU) com interfaces às redes de comunicação mencionadas.

Tomando em consideração factores económicos e de simplicidade de desenvolvimento, aliado à necessidade da unidade de processamento comunicar com duas redes, optou-se por um MCU com mais do que uma *Universal Asynchronous Receiver Transmitter* (UART). Para tal, foi seleccionado o MCU ATmega 2560. Uma vez que já existe no mercado uma placa de prototipagem para este chip, e revelando-se esta ser mais económica do que a construção de raiz de um circuito para o referido MCU, optou-se pela adopção da placa de desenvolvimento Arduino Mega 2560. Para além disto, tal como é referido mais à frente, é-nos permitido ligar o Arduino directamente ao PC, via *Universal Serial Bus* (USB), o que facilita a depuração. Há também que considerar que a plataforma Arduino³ oferece também um conjunto de bibliotecas disponíveis gratuitamente e em código aberto.

Para estabelecer uma interface com a rede externa, optou-se pelo uso de um *Ethernet Shield*, que é um acessório de baixo custo do Arduino para o ligar à internet, bastando para tal, anexa-la ao Arduino e usar um cabo RJ45 para a ligar à rede, sendo facilmente programado usando a biblioteca *Ethernet Library* [41].

Para a interface série-KNX, foi seleccionado o CI TP-UART. Este CI tem como vantagem, comparativamente com outros CIs semelhantes, uma biblioteca (*KnxTpUart*) para Arduino.

3.4.1. UNIDADE DE PROCESSAMENTO

O Arduino Mega 2560, representado na Figura 14, é uma placa de prototipagem microcontrolada, baseada no ATmega 2560, que contém tudo o que é necessário para suportar o MCU. Entre outras características, tem cinquenta e quatro pinos de entrada/saída digital, dezasseis entradas analógicas, quatro UART, um cristal oscilador de 16 MHz, uma ligação USB, um conector de alimentação e um botão de *reset*.

³ Acessível em <http://arduino.cc/>

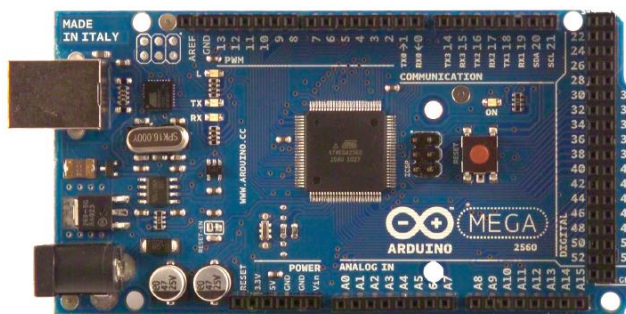


Figura 14 Arduino Mega 2560 [36].

Este Arduino pode ser alimentado via ligação USB (5 V) ou por uma fonte de alimentação externa. Esta última pode ser um adaptador AC/DC ou uma bateria, num intervalo recomendado de 7 a 12 Volt.

Esta placa pode ser programada com o *software* Arduino. Para tal, traz previamente carregado um *bootloader* que permite ao programa monitor efectuar o *upload* do código, evitando a utilização de um programador externo. Para esse efeito possui um CI ATmega U8 integrado na placa que disponibiliza uma das portas UART via USB e fornece uma porta COM ao *software* no PC.

O ATmega 2560 tem 256 KB de memória *flash* para armazenar código, dos quais 8 KB são usados pelo *bootloader*, 8 KB de SRAM e 4 KB de EEPROM [36].

3.4.2. INTEGRAÇÃO NA REDE KNX

O TP-UART da Siemens (Figura 15) é um CI para acesso ao barramento que fornece a camada física e uma parte da camada de enlace da pilha EIB/KNX para par trançado (TP), convertendo o sinal KNX para um sinal série, compatível com as portas UART do MCU. As restantes camadas devem ser realizadas pelo *firmware* do MCU [44].



Figura 15 CI *Transceiver* KNX: Siemens TP-UART-IC.

Este circuito integrado suporta todas as funções de recepção e transmissão. Para além disso gera uma tensão estável de 3.3 V ou 5 V para ser usada pela interface ao MCU.

O acoplamento pode ser realizado directamente ou por via de acopladores ópticos para garantir um isolamento galvânico⁴. Uma vez que o Arduino, durante a fase de desenvolvimento, esteve ligado, via USB, ao PC, optou-se pelo uso de dois acopladores ópticos (6N137) de alta velocidade (10 Mbit/s) [20][40].

O TP-UART já tem vários serviços definidos, ou seja, um protocolo próprio para várias acções. Cada byte transmitido para o TP-UART é iniciado com um *Control Byte* do KNX. As informações adicionais são transmitidas com um código 'ESC' no campo de controlo do telegrama KNX. O controlador ligado ao TP-UART precisa de receber um *timeout* de 2 a 2,5 ms para detectar o fim do pacote [44].

3.4.3. INTEGRAÇÃO NA REDE IP

O Arduino *Ethernet Shield* (Figura 16) baseia-se no chip *WIZnet Ethernet W5100* que fornece acesso à rede IP nos protocolos *Transmission Control Protocol* (TCP) ou *User Datagram Protocol* (UDP), suportando ligações com uma velocidade de 10/100 Mb. Este chip requer uma tensão de funcionamento de 5 V, fornecida pelo Arduino, e possui um *buffer* interno de 16KB. Possui também um *slot* para cartão micro-SD que pode ser usado para armazenar arquivos que vão servir na rede. De forma a utilizar mais facilmente este *slot*, existe a livraria *SD Library* [41]. Estes dois periféricos utilizam o protocolo *Serial Peripheral Interface* (SPI), controlados pela UART do ATmega 2560.



Figura 16 Arduino Ethernet Shield W5100 [41].

A biblioteca *Ethernet* permite que uma placa Arduino se ligue à internet, podendo servir como servidor, aceitando ligações de entrada, ou cliente, realizando as ligações de saída,

⁴ Isolamento galvânico é um isolamento eléctrico.

com os protocolos TCP e UDP. Segundo a documentação técnica, esta biblioteca suporta até quatro conexões simultâneas (de entrada, saída, ou uma combinação de ambas) [41].

3.5. INTERFACE GRÁFICA ANDROID

O SO Android foi desenvolvido inicialmente pela *Android Inc.*, uma empresa entretanto adquirida pela *Google*, e mais tarde pela *Open Handset*, um consórcio de 47 empresas de *hardware*, *software* e telecomunicações, apostadas em desenvolver normas abertas para dispositivos móveis [45][46]. A constituição de uma aplicação Android está disponível para consulta no Anexo A.

3.5.1. ANDROID-CLIENT

A linguagem Java, uma das mais populares, tem várias vantagens, das quais se destaca o facto de ser uma plataforma independente, *multi-threading*, orientada a objectos e segura, que a tornam desejável para o desenvolvimento de aplicações em ambiente Internet [46].

A internet impulsiona as capacidades dos dispositivos Android, pois a informação em tempo-real torna-se fácil de obter, o que facilita a actualização de dados.

Tomando em consideração as limitações de processamento dos Dispositivos Móveis de Internet (DMI), torna-se aconselhável correr os processos mais intensos num servidor *Web*. Tal estratégia permite diminuir o tempo de processamento e, em alguns casos, ajuda a manter o fluxo de execução da aplicação. Esta disposição é chamada de computação cliente-servidor, uma arquitectura onde o cliente faz pedidos ao servidor, que está à escuta e pronto para os executar [47].

Um *socket* é uma interface de comunicação entre dois computadores que é muito usada por aplicações para enviar pedidos ou respostas para a rede. Em Java, um *socket* pode ser visto como um objecto especial de um cliente ou servidor cuja comunicação se baseia em TCP-IP. Independentemente da complexidade da comunicação do *socket*, esta deverá incluir as seguintes etapas [46]:

1. A criação de um objecto '*socket*';
2. Estabelecimento da ligação;
3. Abertura do *iostream* (fluxo de entrada/ saída) que está ligado ao *socket*;
4. Operação de leitura e escrita no *socket*;
5. Fecho do *socket*.

3.5.2. DADOS ESTATÍSTICOS

Ao programador de uma aplicação interessa conhecer o seu dispositivo alvo. Embora seja possível desenvolver aplicações para diferentes versões da plataforma Android, este é um dos parâmetros que interessa saber.

Existem várias versões da plataforma Android: *Cupcake* (Android 1.5), *Donut* (Android 1.6), *Eclair* (Android 2.0), *Froyo* (Android 2.2), *Gingerbread* (Android 2.3), *Honeycomb* (Android 3.0), *Ice Cream Sandwich* (Android 4.0) e *Jelly Bean* (Android 4.1) [48].

Baseado nas características dos dispositivos que recentemente recorreram à *Google Play Store*, o site *developer.android.com* anuncia a *Jelly Bean* como a versão da plataforma mais utilizada na actualidade com um peso de 40,5%, seguida da *Gingerbread* com 33,1% [49]. Tal estatística pode ser verificada na Figura 17.

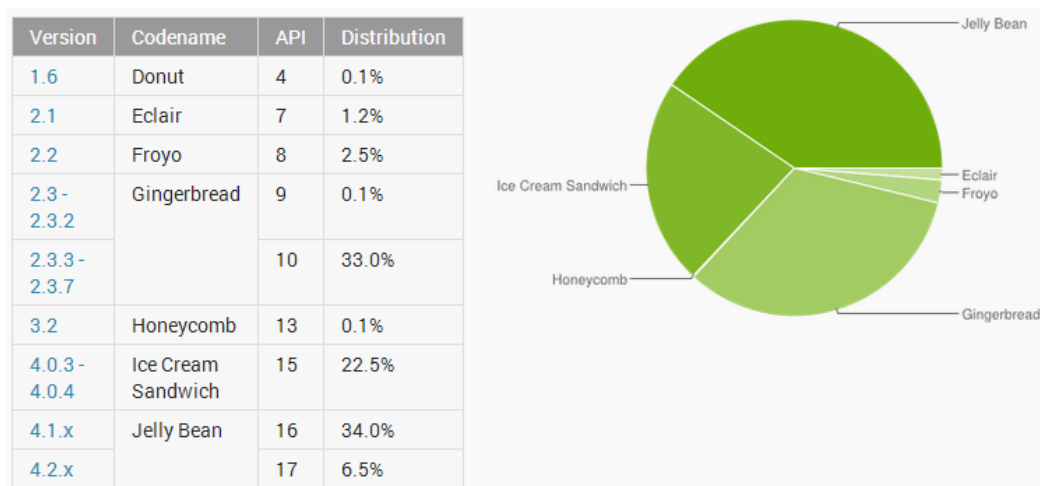


Figura 17 Dados estatísticos da utilização das versões da plataforma Android [49].

Outro parâmetro importante é a configuração do ecrã, definido pela combinação do tamanho e da densidade. A Figura 18, em que estão representados estes dados, conclui-se que a esmagadora maioria dos utilizadores da plataforma Android (80,1%) possui um dispositivo com um ecrã de tamanho médio, e, no que toca à densidade, as mais utilizadas são mdpi (23,2%), xhdpi (23,9%) e hdpi (34,5%) [49].

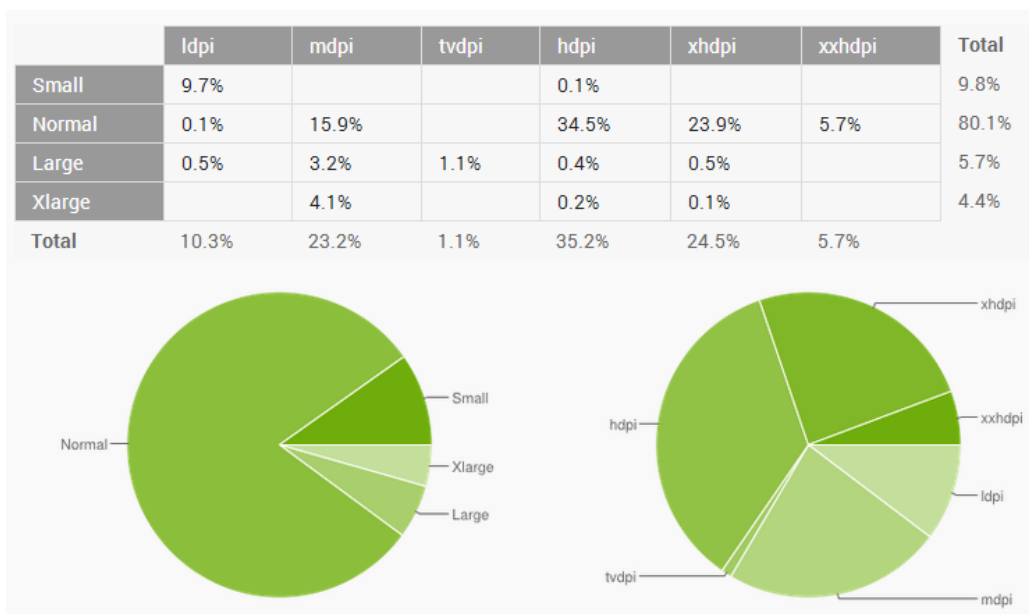


Figura 18 Dados estatísticos das configurações de ecrã actualmente mais utilizadas [49].

3.6. BASES DE DADOS

Para dar resposta às necessidades mencionadas foi usado o sistema de gestão de BDs MySQL, que usa a *Structured Query Language* (SQL) como interface. A linguagem SQL é dividida em vários subconjuntos, de acordo com as operações que queremos efectuar sobre uma BD. Um desses conjuntos é o *Data Manipulation Language* (DML). O DML é utilizado para adicionar, editar, seleccionar e apagar registos da BD. Para tal, são utilizados os comandos apresentados na Tabela 7.

Tabela 7 Comandos DML.

Função	Comando SQL	Exemplo
Adicionar registo	INSERT	INSERT INTO pessoa (id, nome, sexo) VALUE
Consultar registo	SELECT	SELECT * FROM pessoa
Editar registo	UPDATE	UPDATE pessoa SET data_nascimento = '11/09/1985' WHERE id_pessoa = 7
Eliminar registo	DELETE	DELETE FROM pessoa WHERE id_pessoa = 7

Além das aplicações escritas em Java, o Android possui um conjunto de bibliotecas C/C++, usadas por diversos componentes, que permitem trabalhar com arquivos de vídeo,

exibição de conteúdo em 2D e 3D, inclusive bibliotecas implementadas utilizando OpenGL e uma BD relacional denominada SQLite [32].










O *SQLite* é uma biblioteca livre e multiplataforma que implementa um motor de BD SQL transaccional, sem necessitar de servidor pois ela própria é um servidor. Assim, lê e escreve directamente no arquivo de BD armazenado em disco. Esta ferramenta foi utilizada para a máquina de estados da aplicação Android [50].

Assim é possível criar um BD no ambiente de desenvolvimento e instalá-lo no dispositivo, o que acelera o processo de desenvolvimento, visto que os DMI possuem um poder de processamento limitado [32].

3.7. FERRAMENTAS DE DESENVOLVIMENTO

Uma vez definida a arquitectura do sistema, interessa inventariar as ferramentas de desenvolvimento necessárias à sua implementação, tendo em vista o cumprimento dos objectivos traçados da forma mais simples para o programador. Essas ferramentas encontram-se reunidas na Tabela 8.

Tabela 8 Ferramentas e elementos de desenvolvimento.

Rede KNX			25 A4 Binary, Flash Before Off 980303 21 A1 Universal dimmer 906701 25 A4 Roller Shutter switch 980181	
Firmware Arduino	 Visual Studio 2012	 arduino for visual studio www.visualmicro.com	Bibliotecas Arduino: <ul style="list-style-type: none"> • KnxTpUart • Ethernet • SPI • String 	
Servidor web				
Cliente Android	 Java SE Development Kit	 Eclipse JDK	 ADT Pluggin	

3.8. SÍNTESE

Este capítulo definiu os vários constituintes do sistema. Seleccionou-se o KNX como protocolo de comunicação da rede domótica e para a sua constituição utilizaram-se dispositivos KNX disponíveis no ISEP. De forma a dotar o sistema de um controlo remoto por internet definiu-se o projecto de uma interface IP-KNX, que usa um Arduino Mega 2560, uma expansão Ethernet para o Arduino e o CI TP-UART. Como interface de utilizador foi escolhido o SO Android, na sua versão Jelly Bean, com um ecrã de tamanho médio (mdpi).

Este capítulo terminou com a enumeração das ferramentas de desenvolvimento a utilizar. O capítulo seguinte apresenta a implementação do sistema em todos os seus constituintes.

4. IMPLEMENTAÇÃO DO SISTEMA

O capítulo anterior abordou as definições técnicas do sistema e opções relativas ao trabalho a desenvolver. Este capítulo descreve a sua implementação, começando por apresentar uma descrição geral da mesma e, posteriormente, para a descrição do desenvolvimento da interface IP-KNX, ao nível do *hardware*, *firmware* e do seu controlo de alto nível, incluindo neste último a aplicação móvel criada.

4.1. DESCRIÇÃO GERAL DA IMPLEMENTAÇÃO

A implementação deste trabalho divide-se conceptualmente em duas partes: (i) Criação da interface IP-KNX e (ii) Criação de uma aplicação móvel para o SO Android que permite controlar remotamente dispositivos da rede KNX, de forma a validar a interface IP-KNX.

A Figura 19 ilustra o diagrama geral da implementação. A interface IP-KNX é composta por um Arduino Mega 2560 e duas placas de integração nas redes Ethernet e KNX, cujo conjunto designamos de “bloco de interface”. A integração do Arduino com a rede Ethernet é estabelecida pela placa *Ethernet Shield* e a integração com a rede KNX é assegurada pelo módulo KNX desenvolvido. Uma vez que se pretende que a interface IP-KNX assegure a comunicação entre a aplicação Android, presente numa rede IP, e os

dispositivos da rede KNX, é usado um servidor web para responder às solicitações do cliente Android e para gerir a base de dados “remota”, que tem como missão armazenar dados relacionados com a gestão de utilizadores.

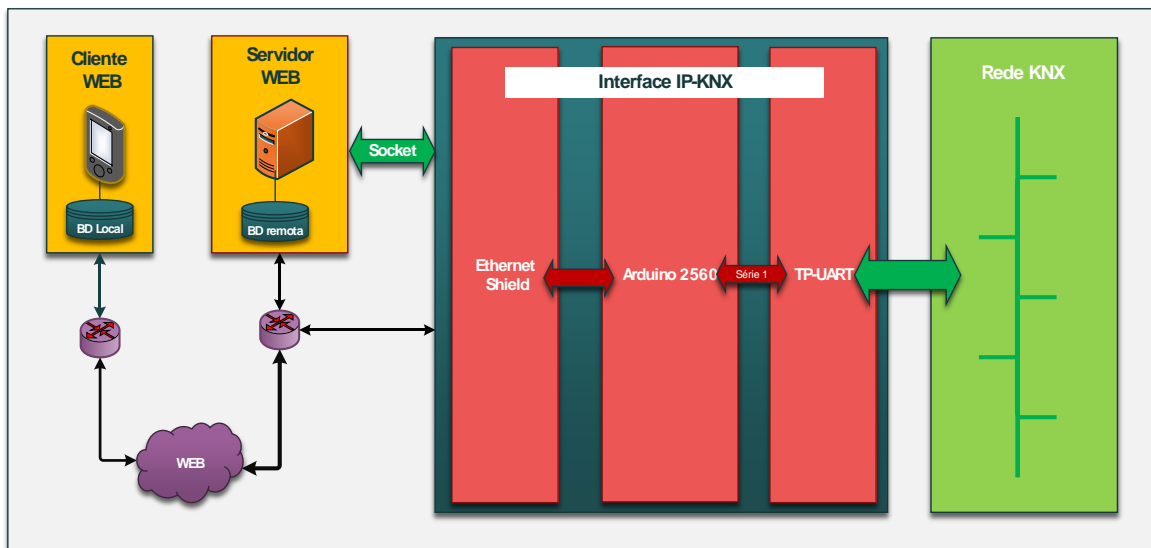


Figura 19 Diagrama geral da implementação.

A Figura 20 apresenta o diagrama geral do fluxo de comunicação. Consideremos como exemplo o controlo de uma lâmpada actuada na rede KNX com o endereço de grupo x/y/z. Quando o utilizador prime o botão “*btnX*” a aplicação envia, através da internet (pelo método POST), uma mensagem ao servidor web identificando o botão que deu origem à mensagem e a acção associada (*toggle*⁵). Ao receber a mensagem, o servidor identifica a acção solicitada, consulta o seu mapeamento, constrói uma mensagem e envia-a à interface IP-KNX, que a traduz num telegrama que é enviado para a rede KNX. Para que a aplicação gráfica Android seja funcional, necessita de saber qual o estado actual do componente a controlar, neste caso a lâmpada X. Assim, a aplicação envia comandos solicitando ao servidor web o estado da lâmpada. Este constrói uma mensagem, enviada para a interface IP-KNX, e é esta interface quem envia um telegrama para a rede KNX solicitando ao actuador o estado da lâmpada. Por sua vez, o actuador responde num telegrama KNX que é traduzido pela interface IP-KNX e transmitido para o servidor web como resposta ao seu pedido. Este último envia o estado da lâmpada X à aplicação do dispositivo que a solicitou e esta actualiza visualmente esse resultado.

⁵ *Toggle* é uma comutação de estado.

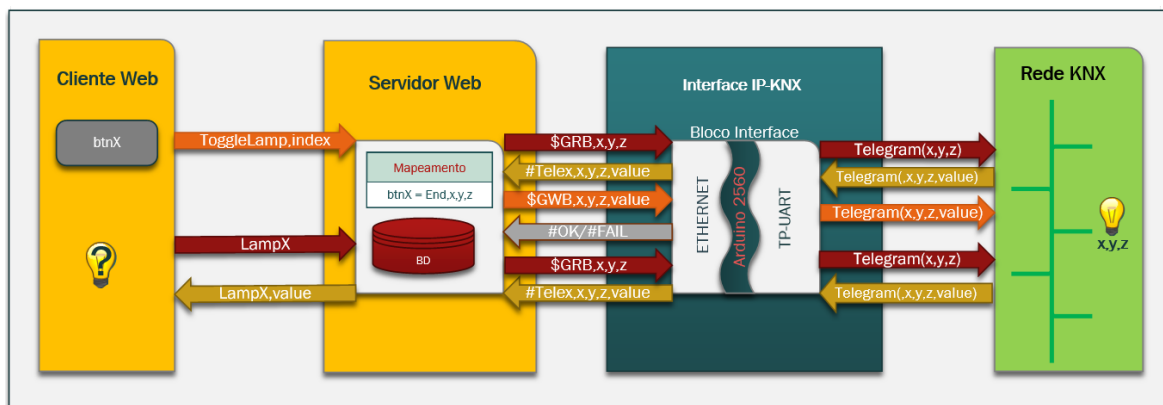


Figura 20 Diagrama geral do fluxo de comunicação.

O fluxo de comunicação entre o servidor web e a interface IP-KNX assenta num protocolo que adopta uma mensagem num formato inspirado no protocolo NMEA⁶. O protocolo implementado define que uma mensagem de solicitação seja iniciada pelo caractere ‘\$’, e que uma mensagem de resposta a uma solicitação seja iniciada por um ‘#’. Em ambos os tipos de mensagens os vários parâmetros são separados por uma vírgula ‘,’ e finalizadas por um *carriage return* (‘\r’).

A estrutura de pedidos do servidor web à interface IP-KNX inicia-se então com o ‘\$’ e é seguida por um comando identificador da função pretendida (Tabela 9), as coordenadas do endereço do dispositivo de destino, separadas por uma vírgula, o valor que se pretende atribuir ao dispositivo, seguido de um ‘*’ e do valor do campo de *checksum* da mensagem, terminando com o *carriage return*. A esta mensagem o bloco Arduino responde com uma *string* iniciada pelo carácter ‘#’ indicando que o comando foi executado com sucesso (#ACK) ou insucesso (#NACK), ou ainda que existiu um problema na comunicação com a rede KNX (#NACK,TIMEOUT). Exemplo:

- \$GWB,1,4,3,1*53\r – Esta mensagem solicita que o dispositivo com o endereço de grupo 1/4/3 adopte o valor de estado 1. Tomando o exemplo anterior de uma lâmpada, esta mensagem ordena que ela fique ligada.

⁶ O protocolo NMEA define mensagens neste formato [51]: \$Endereço,{<valor>}*<checksum><CR><LF>

- #ACK,GWB – O bloco interface responde positivamente ao servidor Web, informando-o que a directiva foi enviada com sucesso num telegrama para a rede KNX.
- #NACK,XPTO – O bloco interface responde ao servidor web dizendo que o comando recebido, nesta caso ‘XPTO,’ lhe é desconhecido;
- #NACK,TIMEOUT – Erro de comunicação com a rede KNX.

Este protocolo define regras também para a solicitação do estado dos participantes na rede KNX. Assim, mantendo a coerência do formato anteriormente apresentado, o servidor web envia uma mensagem ao Arduino iniciada por ‘\$’, seguida pelo comando e pelos três parâmetros do endereço do dispositivo cujo estado se pretende saber. A essa mensagem, e após o envio do pedido à rede e da recepção da respectiva resposta, o bloco interface responde ao servidor web com uma mensagem iniciada pelo caracter ‘#’, seguido da *string* ‘ACK’ e o comando em causa (*i.e.* GRB), seguido do telegrama obtido.

Tabela 9 Comandos implementados na interface IP-KNX.

Comando	Função associada	Descrição
GW2BF	groupWrite2ByteFloat()	Escreve o valor numérico da directiva associada ao endereço de grupo indicado
GWB	groupWriteBool()	Escreve o valor booleano da directiva associada ao endereço de grupo indicado
GRB	groupReadBool()	Solicita a leitura do valor booleano associado ao dispositivo com o endereço de grupo indicado
GR2BF	groupRead2ByteFloat()	Solicita a leitura do valor numérico associado ao dispositivo com o endereço de grupo indicado

4.2. *HARDWARE DA INTERFACE IP/KNX*

Esta secção apresenta a implementação física da interface IP-KNX, detalhando os blocos constituintes, o seu esquema eléctrico e finalizando com a placa de circuito impresso resultante.

4.2.1. DIAGRAMA DE BLOCOS

Como representa a Figura 21, a interface IP-KNX desenvolvida neste trabalho é composta por um Arduino Mega 2560, uma placa Ethernet *Shield* e uma placa de interface RS232 com a rede KNX, baseada no CI TP-UART. Todas as placas da interface IP-KNX suportam uma tensão de funcionamento de 5 V. A placa de interface RS232-KNX contém um isolamento galvânico de forma a proteger a interface IP-KNX da rede KNX.

A interface IP-KNX é alimentada a +5 V fornecidos pela ligação USB utilizada na depuração. Numa aplicação final, a porta de depuração USB é dispensável, pelo que será necessário um alimentador DC ou uma bateria com a tensão no intervalo recomendado [7 - 12] V.

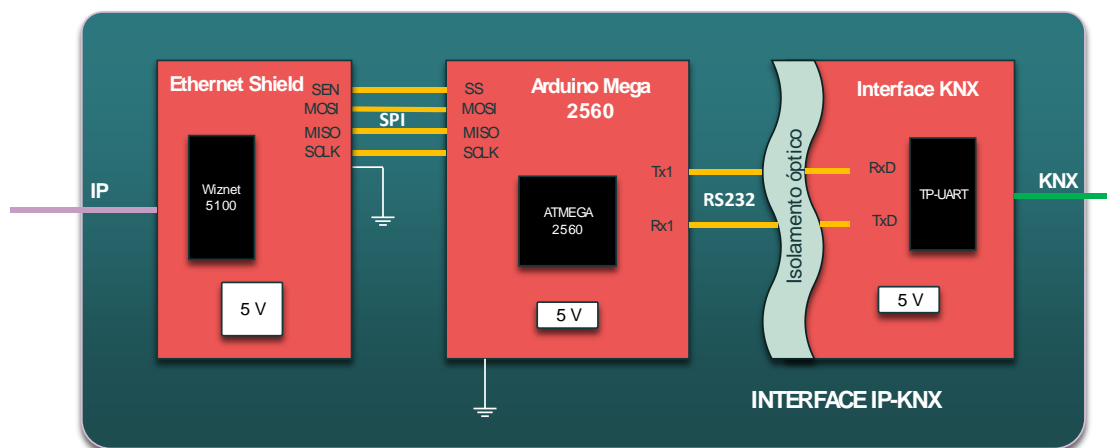


Figura 21 Diagrama de blocos da interface IP-KNX.

4.2.2. ESQUEMA ELECTRÓNICO

Dado o baixo custo do Arduino e do Ethernet *Shield* optou-se pela sua compra, em vez de se criar de raiz todo o dispositivo. Assim, procedeu-se à construção da placa de interface RS232-KNX.

Este interface usa o transmissor TP-UART, que se liga ao Arduino através da porta série 1 (Rx1 e Tx1). O TP-UART é alimentado pelo barramento KNX (VB+), controla a transmissão (TxO) e recepção (RxIn) de dados nesse barramento e gera a tensão de alimentação dos acopladores ópticos (+VCC).

A ligação série ao Arduino, é disponibilizada nos portos TxD e RxD, cujos sinais são isolados opticamente pelos CIs ISO1 e ISO2, respectivamente (Figura 22). As resistências

R2 e R5, limitam a corrente dos díodos emissores dos acopladores ópticos, e as resistências R4 e R3 polarizam os colectores do andar de saída (respectivamente +VCC e +5V).

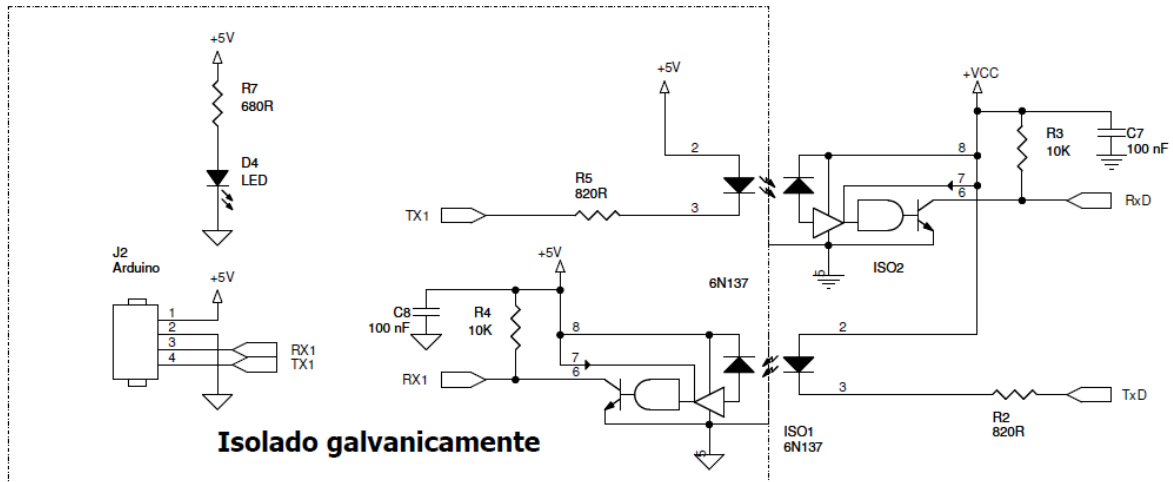


Figura 22 Esquema eléctrico da placa de interface KNX – Bloco de isolamento galvânico.

O circuito do TP-UART corresponde ao desenho de referência do fabricante (Figura 23). Foram implementados indicadores luminosos (Led D3 e D4) da presença de alimentação (Rede KNX e +5 V Arduino, respectivamente).

O esquema eléctrico das ligações do CI TP-UART está ilustrado na Figura 23. Como se pode ver no esquema, o TP-UART comunica com o Arduino através dos pinos TxD e RxD e com o barramento KNX através dos pinos RxIN e TxO.

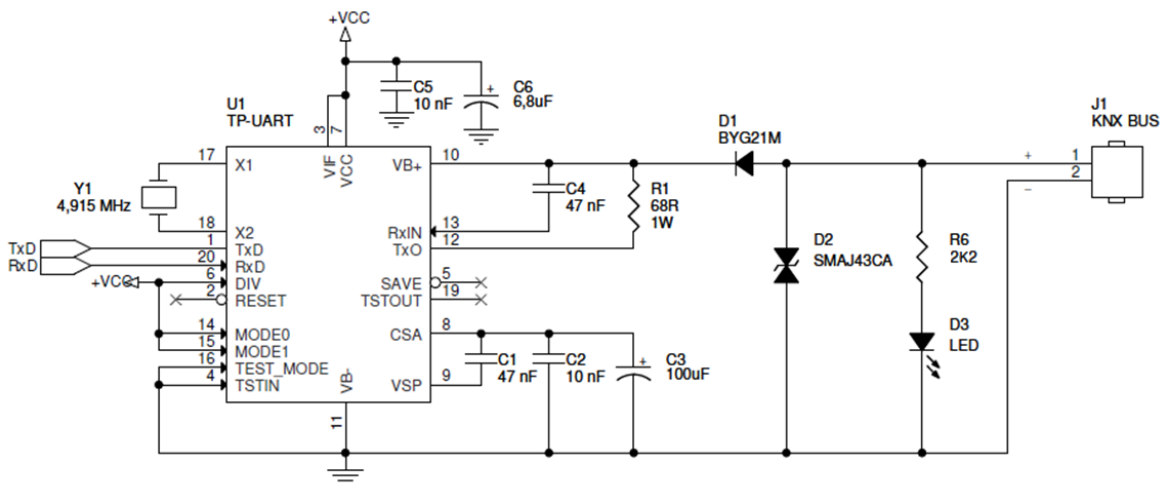


Figura 23 Esquema eléctrico da placa de interface KNX – Ligações do TP-UART.

4.2.3. CIRCUITO IMPRESSO (PCB)

Tendo em vista a posterior produção em massa do circuito da interface RS232 - KNX, a montagem deste está feita num *Printed Circuit Board* (PCB) de, aproximadamente, 6x3cm com face única, usando maioritariamente componentes com o encapsulamento *Surface Mounted Device* (SMD), devido ao seu baixo custo, à sua fiabilidade e ao seu formato compacto. As exceções são o CI TP-UART que usa um encapsulamento *Small Outline Integrated Circuit* (SOIC), os acopladores ópticos e o cristal com um encapsulamento *Dual In-Line Package* (DIP). O desenho do PCB criado está disponível para consulta no Anexo B.

A Figura 24 apresenta os desenhos das camadas de cobre (à esquerda, em cima), das respectivas implantações de componentes (à esquerda, em baixo), e a montagem final (à direita, em baixo). À direita, em cima, é apresentada uma imagem global das várias camadas.

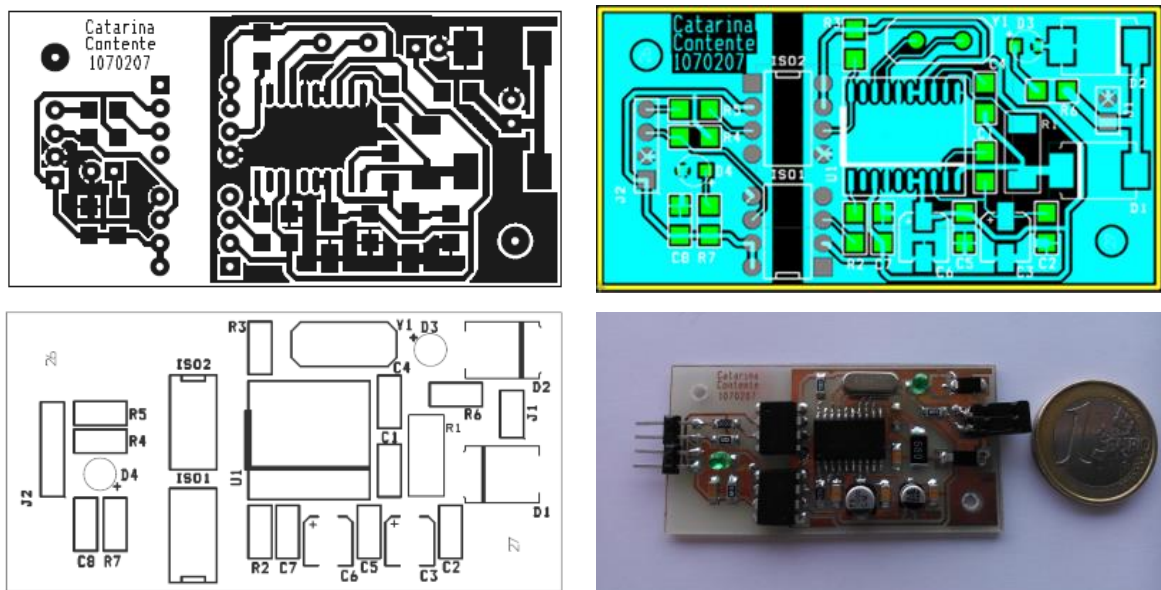


Figura 24 Interface IP-KNX – Imagens do PCB.

4.3. FIRMWARE DO ARDUINO

O Arduino é a unidade de processamento da interface IP-KNX. Como tal, cabe a si o tratamento de toda a informação entregue pelas placas de integração IP e KNX. Para esse efeito, tal como ilustrado na Figura 25, o seu *firmware* estabelece ligação a duas portas série do Arduino (0 – debug, e 1– KNX), às velocidades de 9600 e 19200 bps

respectivamente, inicia a escuta do *socket* de comunicação com o servidor web, e, após estas definições, entra na função *loop()* que chama duas funções: (i) A “*ethernetParse()*”, recebe e trata a solicitação do cliente web, e (ii) A “*debugParse()*” que, tal como o nome sugere, foi criada para auxiliar o desenvolvimento do *firmware*, imprimindo na porta série 0 as mensagens que *ethernetParse()* envia, e recebe o comando com o mesmo formato do recebido via ethernet.

A função *serialEvent1()* é uma rotina executada após o *loop()* sempre que hajam dados na porta série 1. Nesta rotina, o MCU processa o evento e determina qual o seu tipo. Caso se trate de um telegrama KNX, a sub-rotina guarda-o numa variável para enviar, posteriormente, o telegrama como resposta a um pedido da aplicação móvel.

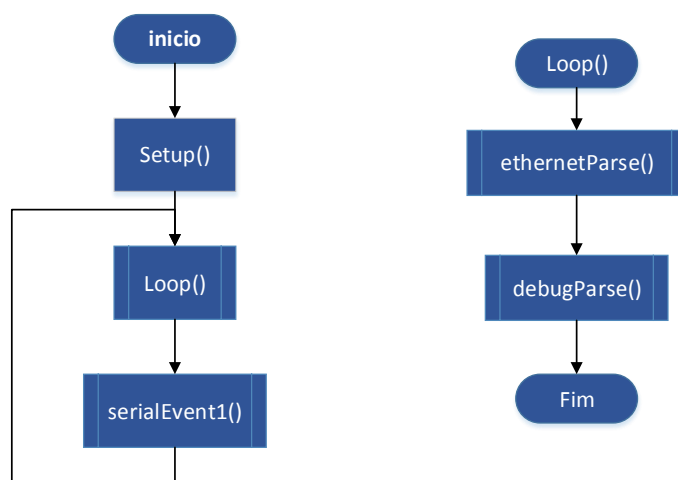


Figura 25 Fluxograma geral do *firmware* do Arduino.

4.3.1. INTERFACE DE CONTROLO KNX

O Arduino está continuamente a executar a função *loop()*, onde é chamada a função *parse()* com duas implementações semelhantes, em que só a origem/destino das mensagens trocadas é alterado (porta série ou cliente web). A função *parse*, cujo fluxograma é apresentado na Figura 26, chama a função *getCommandLine()* para obter uma linha de comandos com o formato genérico “ $\$ \langle \text{CMD} \rangle, \langle \text{PARAM}_1 \rangle, \langle \text{PARAM}_N \rangle * \langle \text{CHK} \rangle$ ” terminado por *carriage return* (‘r’). Se o comando for reconhecido, é chamada a função *ExecuteCMD()*, que vai extrair os parâmetros necessários e criar o telegrama a enviar para a rede KNX utilizando a biblioteca “*KnxTpUart*”.

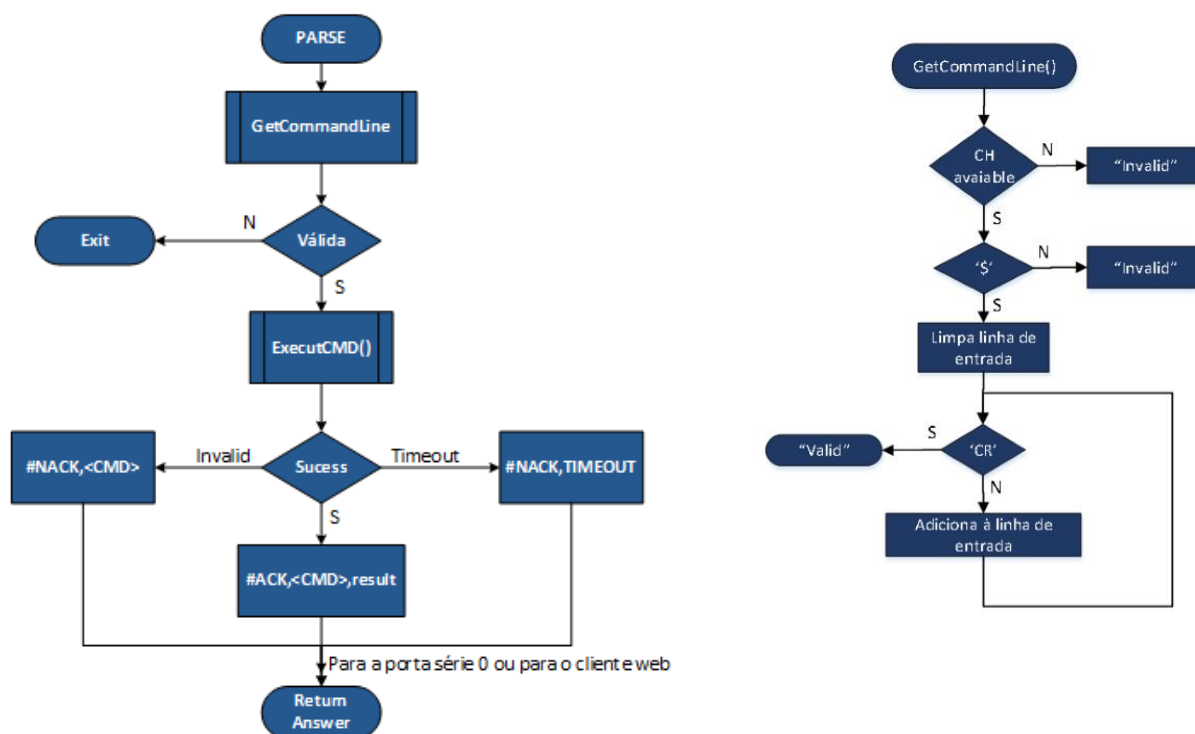


Figura 26 Fluxograma da função Parse.

No caso de existir um telegrama de resposta, este é enviado na mensagem #ACK, após a recepção da mesma pela função *SerialEvent1()*; Todas as funções de envio de um telegrama para a rede KNX retornam “true” se forem bem-sucedidas. Desta forma, no caso de insucesso, é retornada a mensagem “#NACK,TIMEOUT”.

No caso do comando não ser reconhecido, é retornada a mensagem “#NACK,<CMD>”.

4.3.2. SERVIDOR TCP-IP

O arduino não se mostrou funcionalmente capaz de ter um servidor em simultâneo com um cliente, o que implicaria dois *sockets*. Como alternativa, foi implementado um sistema de *polling*⁷ em que o arduino funciona como servidor e responde a pedidos da aplicação cliente, residente no servidor web (Apache). Ou seja, a comunicação entre o bloco interface e o servidor web está assente num *socket* TCP-IP, como ilustra a Figura 27.

⁷ *Polling* é um método de comunicação entre duas estações, primária e secundária, onde a máquina primária (cliente) envia um *Poll* (pedido) para a secundária, e esta responde com dados.

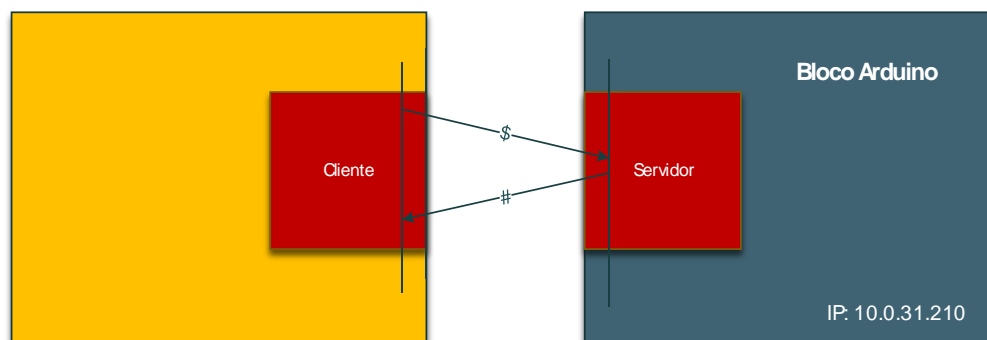


Figura 27 Diagrama de comunicação entre o servidor web e a interface IP-KNX.

O excerto seguinte apresenta a inicialização da biblioteca Ethernet com os endereços físicos (MAC e IP) (função *Ethernet.begin()*), e do servidor em escuta, à espera de pedidos enviados pelo cliente (função *server.begin()*).

```
Void setup()
{
  (...)
  Ethernet.begin(mac, 10.0.31.210);
  server.begin();
  (...)
}
```

4.4. CONTROLO DE ALTO NÍVEL

Esta secção detalha a implementação do código da aplicação móvel, constituída por um servidor *web* e um cliente Android. O servidor *web* usa a linguagem PHP para implementar um cliente do *socket* TCP-IP (comunicação com o bloco interface), o mapeamento de índices e endereços dos dispositivos, e a gestão da BD remota, enviando as respostas ao seu cliente Android num formato JSON⁸. Por sua vez, o cliente Android, programado em Java, converte as respostas recebidas no formato da sua linguagem e actualiza a BD local.

4.4.1. CLIENTE TCP-IP

O servidor web, desenvolvido para implementar a comutação de estado de um dispositivo e retornar ao cliente o seu estado actual, tem embebida uma aplicação cliente, responsável pela criação e iniciação do *socket* com o arduino, acções citadas no excerto seguinte.

```
$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
socket_connect($sock, "10.0.31.210", 80);
```

⁸ JSON (*JavaScript Object Notation*) é um formato de intercâmbio de dados, completamente independente da linguagem, mas usa convenções familiares aos programadores da família de linguagens C [52].

O fluxograma da Figura 28 apresenta o fluxo de informação na ligação ao bloco interface. Este cliente, após a inicialização do *socket*, envia à rede KNX, através da interface IP-KNX, a solicitação da leitura do estado do componente (comando GRB). Após a recepção da resposta com o valor solicitado, a aplicação envia uma mensagem a solicitar a definição do estado para o valor booleano oposto (comando GWB) e, de seguida, indaga novamente a rede sobre o estado actual do dispositivo. Desta forma, o servidor web permite que o cliente Android apresente graficamente o estado real do dispositivo, neste caso, uma lâmpada.

O excerto seguinte documenta as funções utilizadas para enviar e receber mensagens.

```
socket_write($sock, $msg, strlen($msg));
$result = socket_read($sock, 100, PHP_NORMAL_READ );
(...)
```

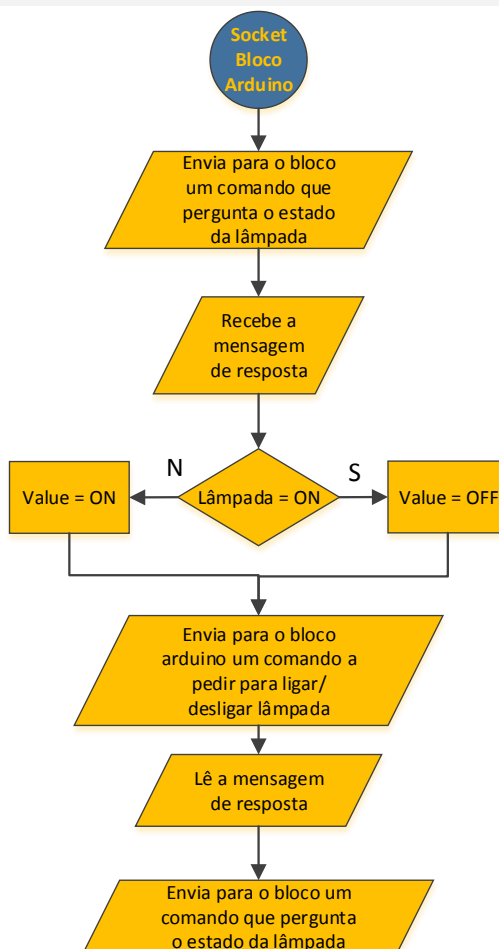


Figura 28 Fluxograma do servidor *web* – *Socket TCP-IP* (cliente).

4.4.2. APLICAÇÃO MÓVEL

Tal como foi referido anteriormente, a aplicação móvel é composta pelo cliente Android e pelo servidor web. Este último está implementado em vários ficheiros PHP:

- *config.php* – Ficheiro que contem as variáveis de ligação à BD remota.
- *DB_connect.php* – Ficheiro usado para activar/desactivar a ligação à BD remota.
- *DB_functions.php* – Este ficheiro contém as funções necessárias para editar as tabelas da BD remota.
- *index.php* → Ficheiro que determina o fluxo de acções do servidor *web*. Dá respostas no formato JSON. A Figura 29 apresenta o fluxo de acções deste ficheiro.

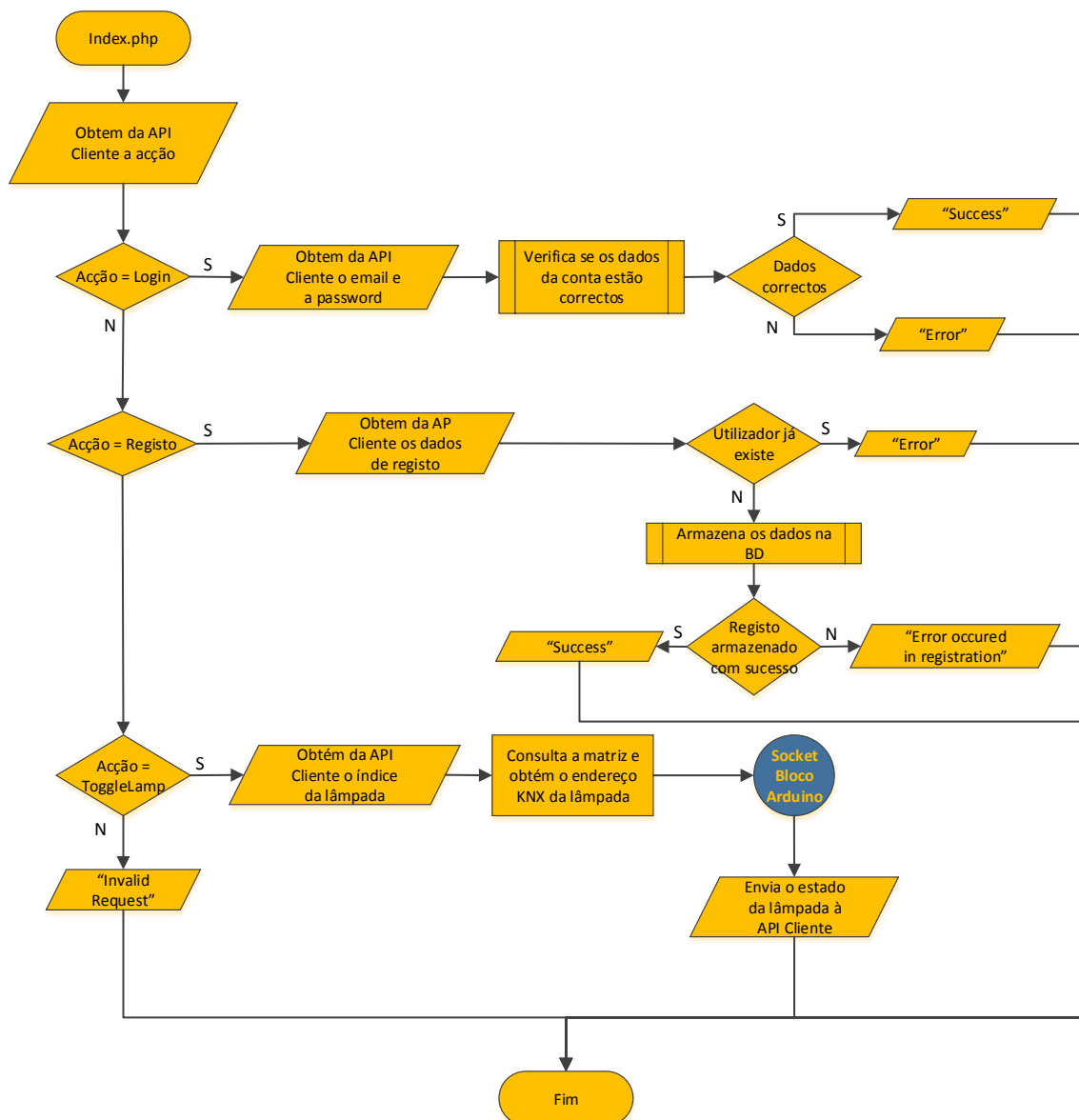


Figura 29 Fluxograma do ficheiro *index.php* do servidor *web*.

i. SERVIDOR WEB

Tal como ilustrado na Figura 29, o servidor *web* obtém do cliente Android a acção que este solicita (com o método POST) e, conforme essa solicitação, processa os dados e envia as respostas. De momento, este servidor está preparado para validar os dados de acesso do utilizador, registar um novo utilizador e permitir o controlo de lâmpadas (ON/OFF).

– BD REMOTA

A BD remota de momento é composta pela tabela “*users*”. Esta, apresentada na Tabela 10, foi concebida para permitir métodos de segurança à aplicação.

Tabela 10 Tabela “*users*” da BD remota.

Coluna	Tipo
<u>uid</u>	int(11)
unique_id	varchar(23)
name	varchar(50)
email	varchar(100)
encrypted_password	varchar(80)
salt	varchar(10)
created_at	datetime
updated_at	datetime
id_profile	int(3)

Ao fazer o registo, é solicitado ao utilizador a introdução de um nome, um *email* e uma palavra-passe. O servidor *web* recebe estes dados e implementa mecanismos de segurança:

- A partir do nome introduzido gera um identificador único que armazena no campo “*unique_id*”;
- A palavra-passe é encriptada, usando a função de codificação *base64_encode()*, e armazenada no formato encriptado. Para ser lida, num login, por exemplo, o servidor web recorre a uma função de descriptação.

– MAPEAMENTO

O cliente Android está implementado de forma a atribuir, de forma automática, um índice a uma lâmpada. Caso receba um pedido de *toggle* o servidor web obtém do cliente Android o índice da lâmpada a controlar, consulta uma matriz de mapeamento, citada no excerto seguinte, e obtém o endereço KNX correspondente. Após essa tradução, inicia a comunicação, via *socket*, com a interface IP-KNX, já descrita anteriormente.

```
$LampAddr[0]["Group"] = 0;  
$LampAddr[0]["Middle"] = 4;  
$LampAddr[0]["SubGroup"] = 1;  
$LampAddr[1]["Group"] = 0;  
$LampAddr[1]["Middle"] = 5;  
$LampAddr[1]["SubGroup"] = 2;  
$LampAddr[2]["Group"] = 0;  
$LampAddr[2]["Middle"] = 6;  
$LampAddr[2]["SubGroup"] = 3;  
$LampAddr[3]["Group"] = 0;  
$LampAddr[3]["Middle"] = 1;  
$LampAddr[3]["SubGroup"] = 0;
```

ii. CLIENTE ANDROID

Foi criada uma biblioteca, com vários ficheiros, para a implementação da aplicação gráfica:

- *JSONParser.java* – Ficheiro que faz a interpretação do formato JSON.
- *DatabaseHandler.java* – Este ficheiro é uma classe com funções para lidar com operações da BD [SQLite].
- *UserFunctions.java* – Este ficheiro contém as funções necessárias para as várias *Activities*.

Por limitações de tempo, foram implementadas apenas três *Activities*, que correspondem às páginas de controlo da aplicação gráfica: (i) *LoginActivity.java*, (ii) *RegisterActivity.java* e (iii) *DashboardActivity.java*. O design destas páginas está nos respectivos ficheiros XML.

A Figura 30 apresenta a imagem das *activities* implementadas.

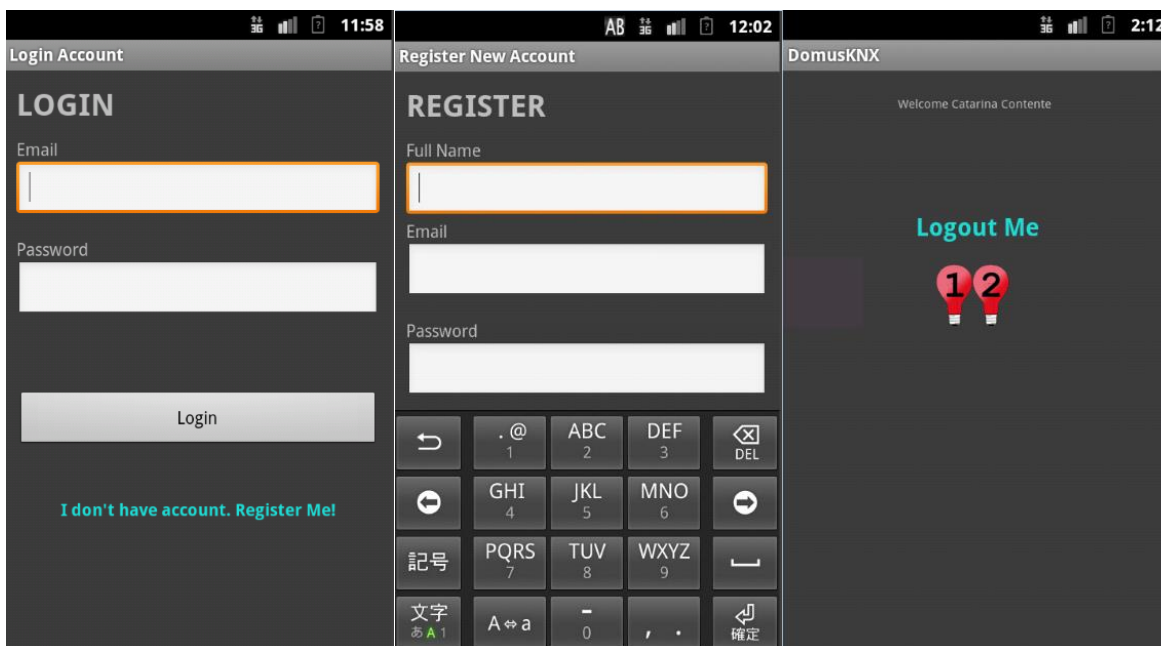


Figura 30 Cliente Android – Páginas de (i) Login, (ii) Registo e (iii) Dashboard.

No Anexo C estão os esboços da aplicação, desenvolvidos numa fase inicial de projecto mas que não chegaram a ser desenhados em XML.

4.5. SÍNTESE

Foi desenvolvida uma placa de interface Série-KNX para interligar o Arduino Mega 2560 à rede KNX, e uma aplicação Android, baseada na filosofia “*Slim Client – Fat Server*”, que utiliza duas BDs (local e remota).

O *firmware* do Arduino implementa um servidor, que recebe pedidos do cliente, residente no servidor Apache (PHP). Este servidor recebe os pedidos provenientes do cliente Android (usando o formato de intercâmbio de dados JSON). O Arduino converte e transmite estes pedidos para a rede KNX, via interface Série-KNX, sob a forma de telegramas. Para tal a interface IP-KNX tem implementadas funções de escrita e leitura, nos formatos booleano e numérico, estando assim apta para permitir o controlo de dispositivos que recebem como parâmetro de controlo este tipo de formato.

A aplicação Android criada solicita ao seu utilizador a introdução de dados de acesso e, após validação dos mesmos, permite-lhe o controlo remoto, via internet, da rede KNX.

O capítulo seguinte apresenta a validação do sistema.

5. VALIDAÇÃO DO SISTEMA

O capítulo anterior apresentou a filosofia de implementação do sistema. O presente capítulo trata a validação do sistema criado, do ponto de vista qualitativo. Esta validação foi realizada em laboratório reproduzindo um sistema real de automação residencial com o protocolo KNX.

5.1. DEFINIÇÃO DE TESTES

A validação da interface IP-KNX passa pela interligação da API Android com a rede KNX, pois é através desta, por operações desencadeadas pelo utilizador, que conseguimos testar a capacidade de tradução da interacção da interface IP-KNX.

Uma vez que a rede *ethernet* do DEE está bloqueada a acessos a partir do exterior, não foi possível uma ligação directa a partir do Android (Wi-Fi) até à rede interna do DEE. Para colmatar esta adversidade, o sistema foi validado, de forma faseada, em três partes:

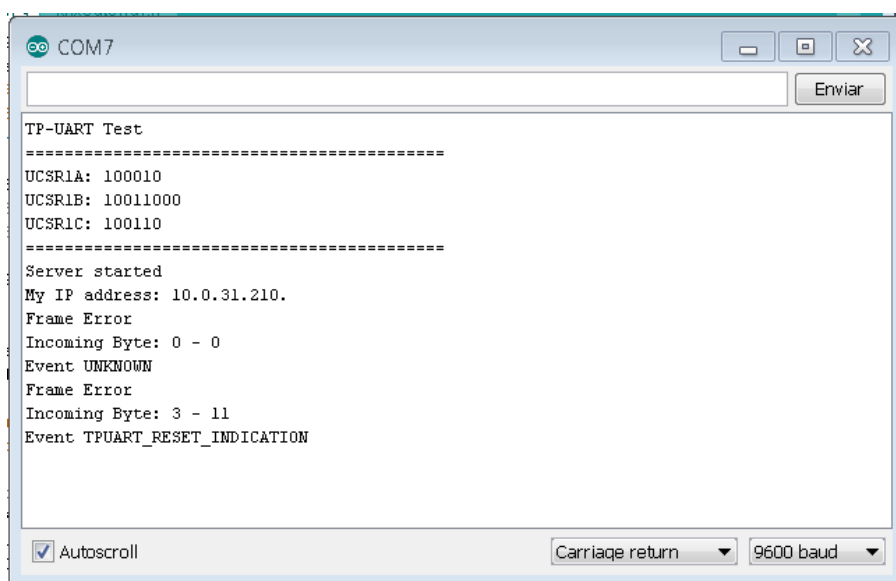
- Comunicação entre a rede KNX e o Arduino;
- Comunicação entre o Arduino e o servidor web;
- Comunicação entre o servidor web e o cliente Android.

5.2. RESULTADOS

Esta secção apresenta os resultados obtidos nos testes parciais realizados. A demonstração está feita recorrendo a imagens capturadas do terminal da porta série 0 do Arduino, do *browser* de internet e da aplicação gráfica Android.

5.2.1. COMUNICAÇÃO ENTRE A REDE KNX E A INTERFACE IP-KNX

A comunicação entre a rede KNX e a interface IP-KNX foi testada através da monitorização da porta série 0. Ao iniciar o sistema, o *firmware* da interface IP-KNX solicita uma reinicialização da UART da placa de integração KNX (Figura 31).



```
COM7
TP-UART Test
=====
UCSR1A: 100010
UCSR1B: 10011000
UCSR1C: 100110
=====
Server started
My IP address: 10.0.31.210.
Frame Error
Incoming Byte: 0 - 0
Event UNKNOWN
Frame Error
Incoming Byte: 3 - 11
Event TPUART_RESET_INDICATION

Autoscroll [checked] Carriage return 9600 baud
```

Figura 31 Porta série 0 – Telegrama de Reset.

Foram introduzidos manualmente, no terminal da porta série 0, comandos de definição (GWB) e de leitura do estado (GRB) de uma lâmpada (indicados a vermelho).

Como é visível na Figura 32, ao enviar o comando “*\$GWB,0,4,1,1*34*”, solicitou-se à rede KNX que ligasse a lâmpada com o endereço de grupo 0/4/1. A lâmpada ligou. De seguida, foi enviado um pedido de leitura do estado da mesma lâmpada (“*\$GRB,0,4,1*23*”), e a rede KNX enviou o telegrama de resposta que foi impresso no terminal (indicado a verde). Essa impressão mostra que a interface IP-KNX leu o estado da lâmpada, que corresponde ao primeiro parâmetro impresso, neste caso, o valor 1.

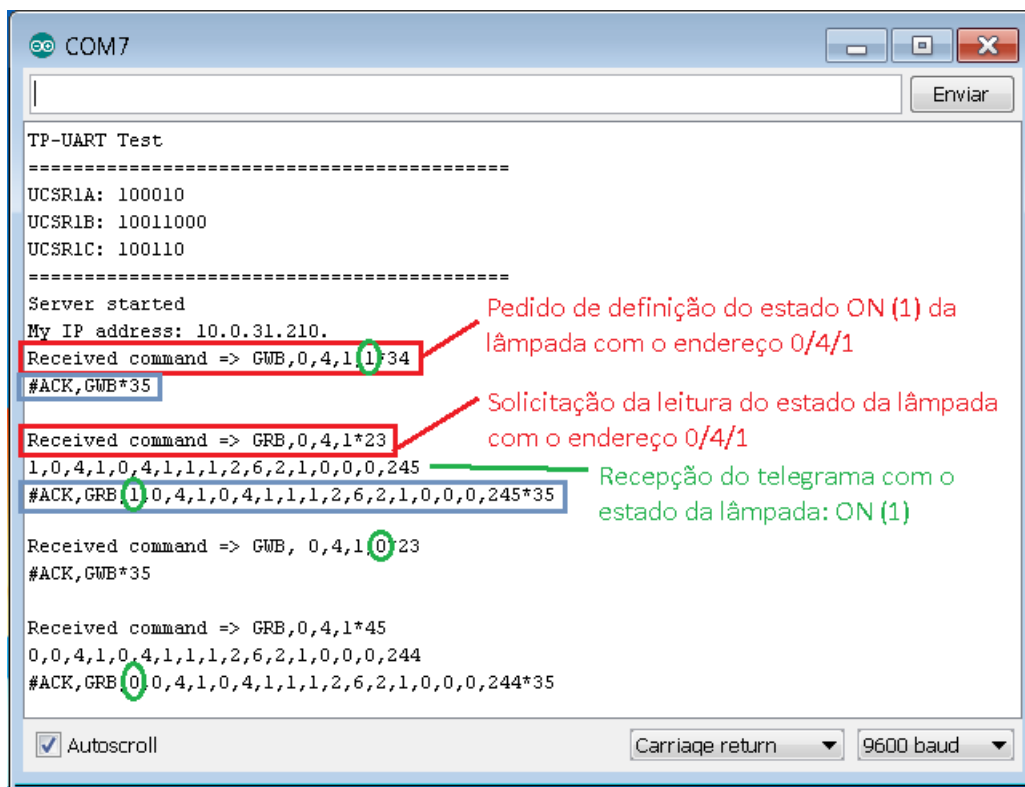


Figura 32 Envio de comandos para a rede KNX e recepção das respostas dessa rede.

5.2.2. COMUNICAÇÃO ENTRE A INTERFACE IP-KNX E O SERVIDOR WEB

Seguiu-se a validação da comunicação entre interface IP-KNX e o servidor web. Foi simulado o envio de dados do cliente Android através da sua introdução manual na barra de endereços. Para que tal fosse possível, o ficheiro *index.php* foi alterado para usar o método GET (ao invés de POST), criando-se para este teste o ficheiro *index1.php*.

i. COMANDO INVÁLIDO

Inicialmente foi simulada a introdução de um comando que não é reconhecido pelo servidor web. Para tal, foi introduzido na barra o endereço: *localhost/Arduino/index1.php?tag=cmd&cmd=Invalid&index=0*. A resposta a estes parâmetros está na Figura 33, onde é visível que, ao ser recepcionado o comando \$XPTO (inexistente) com o índice 0, o servidor consultou a sua matriz de mapeamento e traduziu o índice 0 no endereço 0/4/1, compondo a mensagem “\$XPTO0,4,1*3F”, enviando-a posteriormente à interface IP-KNX.



Figura 33 Servidor web – envio de comando inválido.

Como mostra a Figura 34, a interface IP-KNX recebeu o comando enviado pelo servidor web. Na Figura 33 é visível a resposta que o servidor web recebeu da interface IP-KNX, “#NACK*35”, determinando que o comando XPTO é inválido.

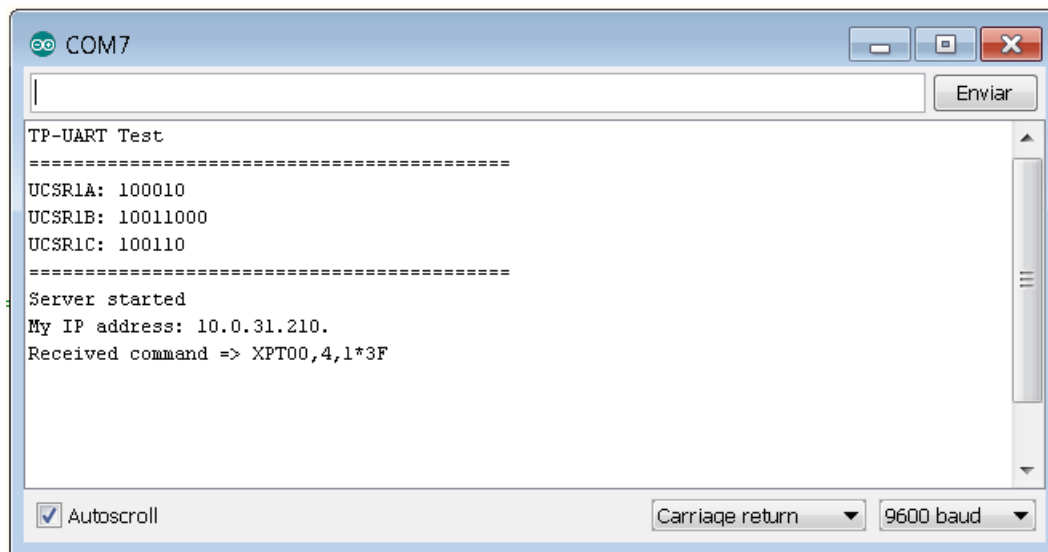


Figura 34 Comunicação da interface IP-KNX com o servidor web – Comando inválido.

ii. COMANDO VÁLIDO

Procedeu-se à introdução de um comando válido (*ToggleLamp*) e confirmou-se novamente o sucesso da comunicação do servidor web com a interface IP-KNX, através do envio de um pedido de leitura do estado da lâmpada 0/4/1, recebendo no entanto uma mensagem de erro, *#NACK,TIMEOUT*35*, pois a interface não estava ligada à rede KNX, não obtendo dela qualquer resposta (Figura 35).



Figura 35 Servidor web – envio de comando válido.

A Figura 36 demonstra que a interface IP-KNX recebeu o comando de solicitação de leitura do estado do dispositivo, mas que, por não ter ligação à rede KNX, foi retornada a mensagem de erro “*Timeout while receiving message*”.

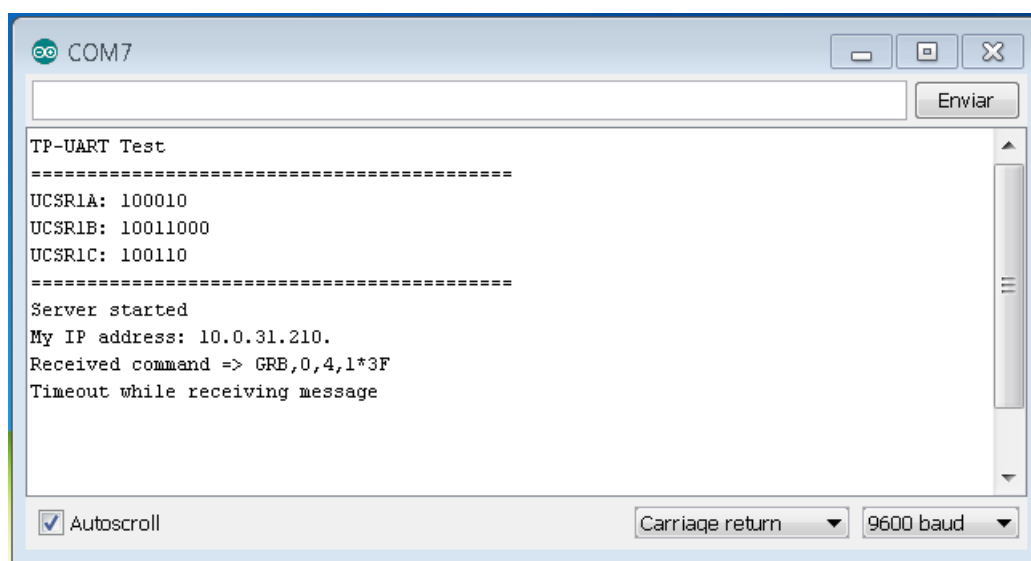
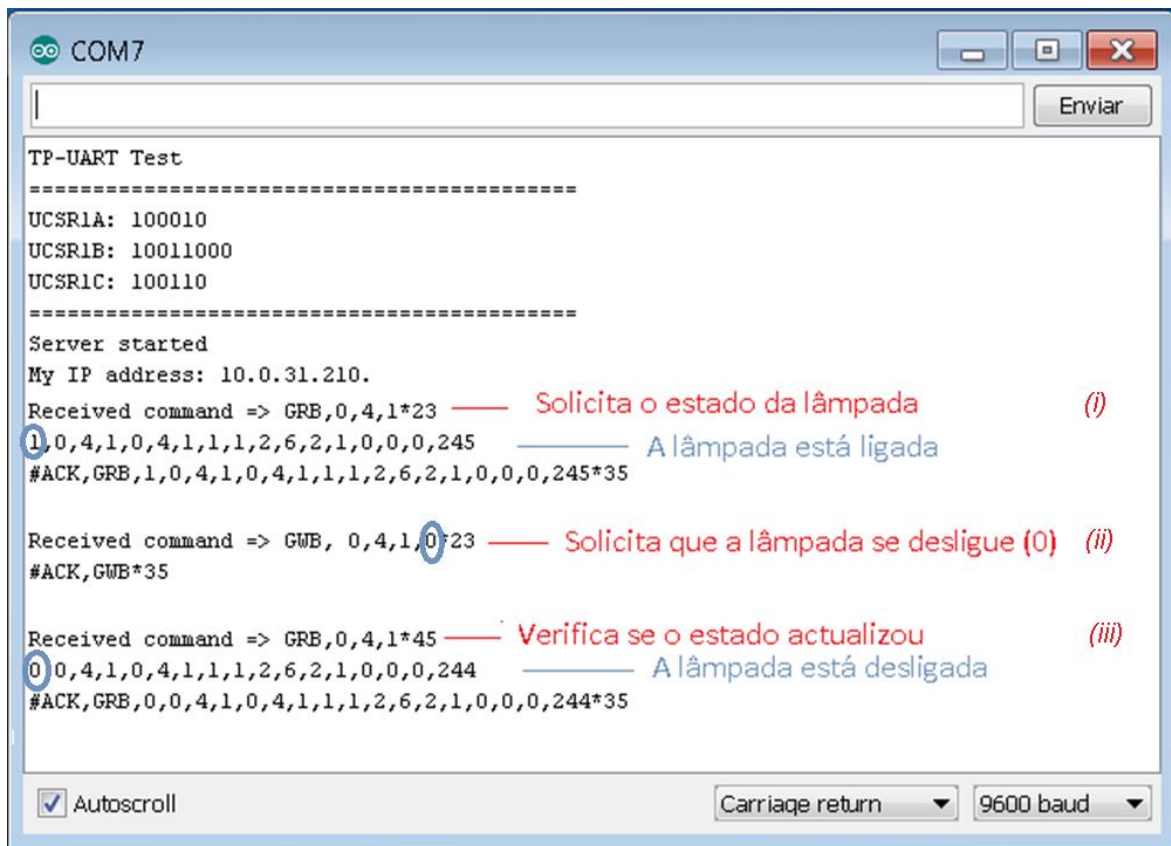


Figura 36 Comunicação da interface IP-KNX com o servidor web – *Timeout*.

Por fim, foi testado o envio do comando *ToggleLamp*, com o índice 0, mas com a interface IP-KNX ligada à rede KNX. A Figura 37 apresenta o fluxo de comandos enviado pelo servidor web à interface IP-KNX, e por esta à rede KNX. Nesta figura é visível a sequência de comandos enviados pelo servidor web para uma acção de comutação (*toggle*) da lâmpada com o endereço de grupo 0/4/1: (i) Começa por solicitar o estado da lâmpada, ao

receber a resposta (ligada) (ii) Solicita a sua definição no estado oposto (desligada) e, por fim, (iii) Solicita novamente o estado da lâmpada para confirmar se este foi actualizado.



```

COM7
TP-UART Test
=====
UCSRIA: 100010
UCSRIB: 10011000
UCSRIC: 100110
=====
Server started
My IP address: 10.0.31.210.
Received command => GRB,0,4,1*23 — Solicita o estado da lâmpada (i)
1,0,4,1,0,4,1,1,1,2,6,2,1,0,0,0,245 — A lâmpada está ligada
#ACK,GRB,1,0,4,1,0,4,1,1,1,2,6,2,1,0,0,0,245*35

Received command => GWB,0,4,1*0 — Solicita que a lâmpada se desligue (0) (ii)
#ACK,GWB*35

Received command => GRB,0,4,1*45 — Verifica se o estado actualizou (iii)
0,0,4,1,0,4,1,1,1,2,6,2,1,0,0,0,244 — A lâmpada está desligada
#ACK,GRB,0,0,4,1,0,4,1,1,1,2,6,2,1,0,0,0,244*35

Autoscroll Carriage return 9600 baud
  
```

Figura 37 Teste da comunicação do interface IP-KNX com o servidor web – Comando válido.

5.2.3. COMUNICAÇÃO ENTRE O SERVIDOR WEB E O CLIENTE ANDROID

Tal como mencionado no Capítulo 4, o cliente Android limita-se a enviar comandos e parâmetros para o servidor web processar e devolver a resposta.

i. REGISTO

Começou-se por testar a função de registo de utilizadores, testando assim a ligação entre o servidor web e a BD remota. Conforme ilustrado na Figura 38, foram introduzidos os dados:

- Catarina Contente (Nome)
- 1070207@isep.ipp.pt (email)
- pass (palavra-passe)

Como se pode ver no fundo da figura, este registo foi adicionado à base de dados, com o identificador único “5271d7809357b6.38183573” e a palavra-passe encriptada “IHpWD2v6kvR6L8ImUUXxRYGNtjk1Njc3MDE3Mjlx”.

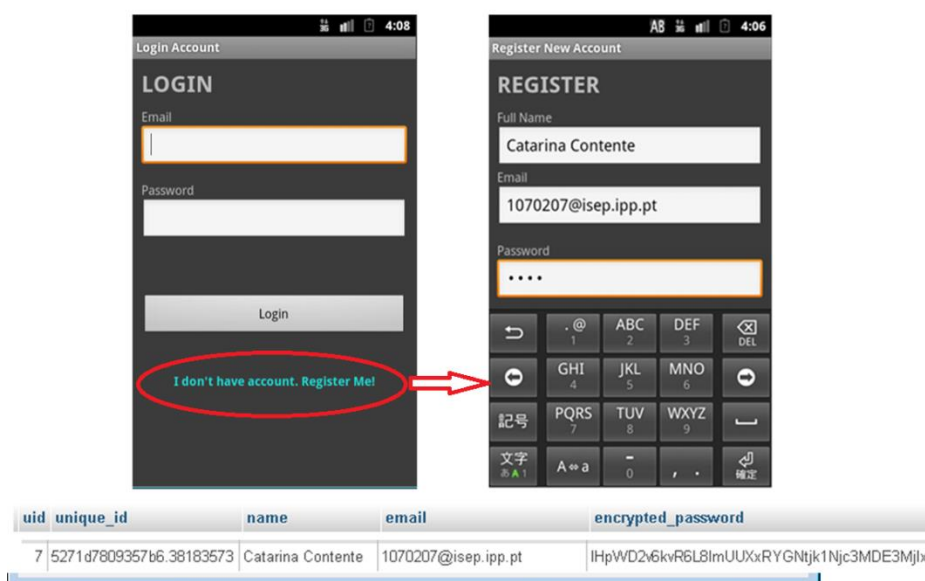


Figura 38 API Android – Janela de registo e excerto da respectiva entrada na BD remota.

ii. LOGIN E LOGOUT

Fora da rede do ISEP, em casa, foi testada a função de validação dos dados de entrada (Login). Introduziram-se os dados do registo relatado anteriormente e a aplicação transitou para a ‘Activity’ Dashboard, conforme ilustrado na Figura 39.

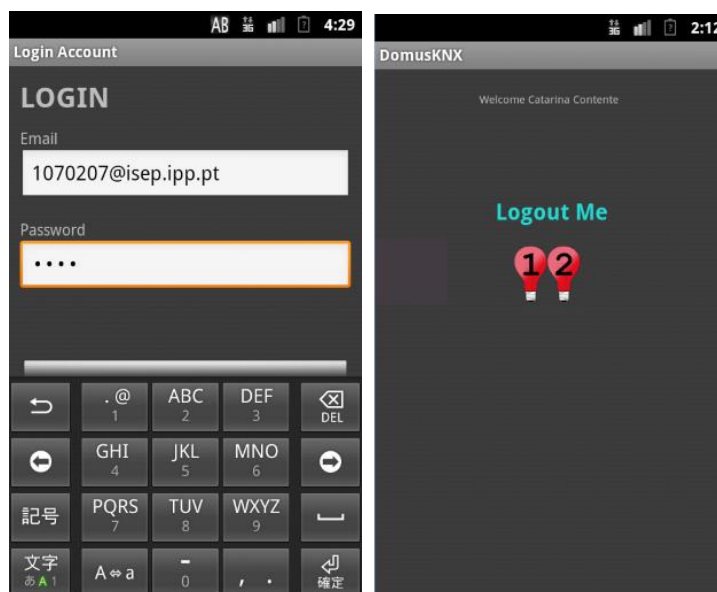


Figura 39 API Android – Validação do Login.

Uma vez dentro do *Dashboard*, ao clicar em “Logout Me” é encerrada a sessão do utilizador, voltando a ser apresentada a *Activity* de Login.

iii. TOGGLE

Para testar a actualização do estado do ícone, foi alterado o ficheiro PHP para enviar manualmente o estado (1 ou 0), independentemente da comunicação com a rede KNX.

A Figura 40 apresenta a actualização do ícone de estado da lâmpada 1.

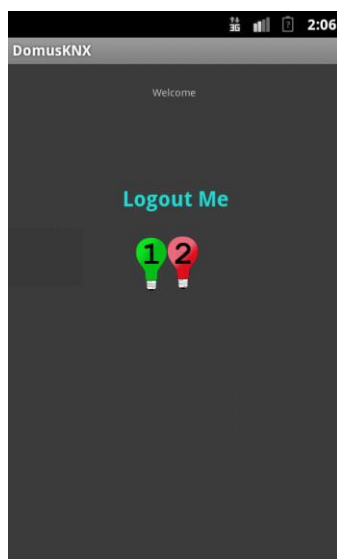


Figura 40 API Android – Comutação do estado da lâmpada 1.

5.3. SÍNTESE

Este capítulo apresentou a fase de testes do sistema desenvolvido. Dadas as limitações, a validação foi realizada de forma faseada, dividindo-se em três etapas que testaram a comunicação entre os blocos do sistema. Os resultados obtidos comprovaram a existência desta comunicação, validando com sucesso o trabalho realizado.

O capítulo seguinte apresenta a conclusão deste trabalho.

6. CONCLUSÃO

Um sistema domótico pode oferecer inúmeras funcionalidades que dotam uma habitação de um elevado nível de conforto. Essas funcionalidades dão à residência uma maior versatilidade, quando comparada com um edifício tradicional (sem tecnologia domótica). Este capítulo apresenta as conclusões resultantes da convergência da análise da literatura com a avaliação dos resultados obtidos, dividindo-se em duas secções para, separadamente, apresentar a apreciação crítica do sistema desenvolvido e possíveis desenvolvimentos a implementar no futuro.

6.1. APRECIÇÃO CRÍTICA DO SISTEMA DESENVOLVIDO

No capítulo anterior, foi validado o sistema desenvolvido, de uma forma faseada. Dos resultados obtidos, conclui-se que a interface IP-KNX cumpre com os seus objectivos, uma vez que está apta para receber comandos provenientes do servidor web, convertê-los e proceder ao seu encaminhamento para a rede KNX. Posteriormente, devolve as respostas da rede (sob a forma de telegrama) ao servidor web. Por seu lado, o servidor web devolve esta informação aos clientes Android autenticados que a solicitaram.

Ao nível da análise temporal de resposta do sistema, esta não pode ser medida devido à inexistência de uma ligação completa e simultânea de todos os blocos envolvidos na comunicação.

O protocolo NMEA, que inspirou o protocolo implementado, pressupõe a utilização de um campo de *checksum* no final de cada mensagem. Este, contudo, não foi implementado dada a necessidade de se testar faseadamente, e, para tal, as mensagens serem introduzidas manualmente, sem a capacidade de cálculo automático de *checksum* de forma efectiva. Contudo, a sua implementação é muito simples, bastando para tal executar a função de cálculo de *checksum* no momento apropriado, antes do envio de cada mensagem, e testar o seu valor no receptor.

Ao nível da segurança, o sistema autentifica os dados do utilizador antes de lhe permitir qualquer acesso. Estes dados são armazenados de forma encriptada, limitando o acesso aos mesmos.

De notar que o desenvolvimento efectuado ao nível do SO Android e no servidor web serviu unicamente o propósito de provar o funcionamento da interface IP-KNX. Assim, não foi implementado uma aplicação completa, ao nível da aplicação móvel.

No que diz respeito ao custo da implementação da interface IP-KNX, teve-se o cuidado tentar obter o melhor desempenho ao menor custo. A nível informativo é apresentado no Anexo D a tabela com a lista de materiais utilizados e os respectivos custos. Nestes, estão incluídos os portes de envio e os respectivos impostos. No entanto, não estão contabilizados o custo do circuito impresso (feito no ISEP), nem o custo da mão-de-obra da montagem.

6.2. DESENVOLVIMENTOS FUTUROS

Validado o conceito, encontram-se várias sugestões a desenvolver futuramente para que este vá de encontro às exigências dos consumidores, oferecendo, entre outras coisas, um nível mais elevado de funcionalidades.

Sendo a tecnologia domótica um factor diferenciador no mercado imobiliário, e dado o sucesso comercial dos *smartphones* a nível mundial, justifica-se que este sistema venha a ser aperfeiçoado, tanto ao nível da interface IP-KNX como ao da API Android. Nessa sequência, será pertinente rentabilizar este sistema certificando a interface na Associação

KNX e hospedando a API no *Android Market*. Poderá também ser desenvolvida uma aplicação gráfica para o SO iOS.

6.2.1. INTERFACE IP-KNX

Relativamente à interface IP-KNX, listam-se vários aperfeiçoamentos necessários. Do ponto de vista funcional, a melhoria mais óbvia prende-se com a implementação do mapeamento dos endereços por uma aplicação específica para esse efeito. Outros aperfeiçoamentos possíveis estão relacionados com a expansão da interface a outros tipos de actuadores e respectivos DTPs (*e.g.* controladores de intensidade luminosa (*dimmers*), elementos multimédia, controladores de estores, etc.).

Será também interessante desenvolver uma aplicação gráfica dedicada à configuração de alguns parâmetros da interface IP-KNX:

- Endereço KNX;
- IP;
- Porto;
- Aplicação de mapeamento.

Essa aplicação deverá automatizar o mapeamento permitindo a importação do ficheiro de parametrização da rede KNX, exportado do ETS. Para tal, a BD remota deverá ser desenvolvida, implementando as tabelas necessárias para esse efeito.

6.2.2. API ANDROID

Quanto à aplicação de controlo remoto, entre outras coisas, o seu aspecto gráfico poderá ser melhorado. No que diz respeito a funcionalidades, são vários os aperfeiçoamentos sugeridos:

- Interface de configuração do IP do servidor web;
- Remover a opção de registo de novos utilizadores da página de login e incluí-la no interior da aplicação, apenas acessível após a validação dos dados de acesso, de forma que, apenas o administrador possa, por exemplo, registar novos utilizadores;

- Expansão das funcionalidades:
 - Gestão de utilizadores;
 - Monitorização energética;
 - Gestão de cenários;
 - Definição de perfis de utilizador (Perfis de inclusão que permitam facilitar a interação com o edifício de pessoas de mobilidade reduzida; Perfis orientados à gestão de energia que assegurem economias que tornem a utilização dos edifícios mais sustentável);
- O sistema poderá, de forma automática e sem requerer conhecimentos do utilizador, permitir:
 - Obter a configuração do edifício, incluindo a sua estrutura em andares e divisões, assim como outras áreas de interesse;
 - Disponibilizar todas as funcionalidades da instalação KNX do edifício;
 - Monitorizar parâmetros relevantes da instalação;
 - Alterar parâmetros da instalação que resultem numa utilização mais eficaz da energia consumida no edifício.

6.2.3. VALIDAÇÃO

Realizar uma validação, comparando o desempenho do sistema, quando ligado a vários dispositivos móveis heterogéneos, de forma a avaliar melhor a sua flexibilidade e escalabilidade. Simultaneamente, obter os dados estatísticos de utilização.

Referências Documentais

- [1] FARIA, Luis; “*Domótica e a Requalificação de Edifícios*”; Touchdomo, Lda, Porto, Portugal; Artigo Publicado na Revista Neutro à Terra, Nº6, Dezembro de 2010.
- [2] REIS, Jorge; “*Controlador de domótica de custo reduzido no Instituto de Sistemas e Robotica*”; Dissertação de Licenciatura e Engenharia Electrotécnica e de Computadores orientada por Mário Jorge Rodrigues de Sousa e apresentada à Faculdade de Engenharia da Universidade do Porto, em Julho de 2006.
- [3] MOWEN, John C.; MINOR, Michael S., “*Comportamento do Consumidor*”, Pearson Education; 2003.
- [4] FLORES, António; “*A Criação de Valor no Binómio ‘Casa Inteligente / Consumidor’*”; Instituto Superior de Engenharia do Porto; Artigo Publicado na Revista Neutro à Terra, Nº5, Junho de 2010.
- [5] COSTA, João; “*O comportamento do consumidor: necessidades, motivações e personalidade*”; Artigo publicado em <http://www.portal-gestao.com>, consultado a 17/05/2013.
- [6] PINTO, Carlos; “*Fundamentos de Gestão*”; Instituto Superior de Engenharia do Porto; 2005.
- [7] ONU; “*Information Economy Report 2011*”; United Nations Conference on Trade and Development; 2011.
- [8] GARTNER GROUP; “*Gartner says Android to command nearly half of Worldwide Smartphone Operating System Market by near-End 2012*”; Artigo publicado em <http://www.gartner.com/newsroom/id/1622614>, consultado a 30/05/2013.
- [9] “*Silences... unveils music beneath noise...*”; Artigo publicado em <http://www.silences.be/fr/domo/>, consultado a 30/05/2013.
- [10] FINDER; “*Pré-Automação Residencial*”; White Paper; Ed. 003; 2011.
- [11] OGATA, Katsuhiko; “*Engenharia de controle moderno*”; Prentice-Hall do Brasil; 1985; ISBN 85-7054-019-1.
- [12] DIAS, César; PIZZOLATO, Nélio; “*Domótica – Aplicabilidade e Sistemas de Automação Residencial*”; Vértices, v. 6, n. 3; 2004.
- [13] “*ISO/IEC Guide 2, Definition 3.2*”; 1996.
- [14] BRANDÃO, Roque; “*A Domótica ao Serviço da Sociedade*”; Instituto Superior de Engenharia do Porto; Artigo Publicado na Revista Neutro à Terra, Nº1, Abril de 2008.
- [15] PALMA, Diana; “*FEUP KNX – Domótica KNX/EIB de Baixo Custo*”; Dissertação de Mestrado em Engenharia Electrotécnica e de Computadores orientada por Mário

- Jorge Rodrigues de Sousa e apresentada à Faculdade de Engenharia da Universidade do Porto, em Março de 2008.
- [16] SILVA, Luís; “*Automação em Ambientes Residenciais*”; Dissertação de Mestrado em Engenharia Mecânica orientada por José Paulo Santos e co-orientada por Rui Moreira, apresentada à Universidade de Aveiro, em 2008.
- [17] “*KNX é agora Standard Internacional: ISSO/IEC 14543-3*”; Artigo publicado em konnex.org; consultado a 05/06/2013.
- [18] SOARES, Miguel; “*GAMMA instabus, Building Management Systems – Standard KNX/EIB*”; Siemens; 2011.
- [19] BOLZANI, Caio; “*Residências Inteligentes*”; São Paulo, Ed. Editora e Livraria da Física; 2004.
- [20] BELEZA, José; “*Sistema Integrado de Segurança e Domótica*”; Dissertação de Mestrado em Engenharia Electrotécnica e de Computadores orientada por Mário Jorge Rodrigues de Sousa e apresentada à Faculdade de Engenharia da Universidade do Porto, em Janeiro de 2009.
- [21] HAMOUI, Mohamad; “*Un Système Multi-Agents à Base de Composands pour l’Adaptation Autonome au Contexte Application à la Domotique*”; Dissertação de Doutoramento de Informática orientada por Marianne Huchard e co-orientada por Christelle Urtado e Sylvain Vauttier e apresentada à Universidade Montpellier II, em 2010.
- [22] DONY, C.; TIBERMACHINE, C; URTADO, C; VAUTTIER, S.; “*Specification of an exception handling system for a replicated agent environment*”; In Proceedings of WEH’08, the 4th international workshop on Exception handling, pages 24-31; Nov 2008.
- [23] NZEKWA, R.; ROOUVOY, R.; SEINTURIER, L.; “*A flexible context stabilization approach for self-adaptive application*”; In Workshop Proceedings of Eighth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), Pages 7-12. IEEE, March 2010.
- [24] QUEIRÓS, Sérgio; “*Como abordar uma instalação de domótica KNX para uma moradia?*”; Schumal – Engenharia e serviços, Lda; Artigo Publicado na Revista Neutro á Terra, Nº10; 2012.
- [25] “*EIBA Handbook Series – Volume 1: Primer*”; V1.1; 1999.
- [26] MERZ, H.; HANSEMANN, T.; HÜBNER, C.; “*Building Automation – communication Systems with EIB/ KNX, LON, and BACnet*”; Springer-Verlag Berlin Heidelberg; 2009; ISBN: 978-3-540-88828-4.
- [27] “*Project Engineering for EIB Installations – Basic Principles*”; European Installation Bus Association; Ed. 4th; 1998.
- [28] “*EIB-ABB*”; ABB; 2001.
- [29] “*Interworking*”; KNX Association.
- [30] “*Serial Data Transmission and KNX Protocol*”; KNX Association

- [31] BOVET, Gerome; HENNEBERT, Jean, "A Web-of-Things Gateway for KNX Networks" *Smart Objects, Systems and Technologies (SmartSysTech), Proceedings of 2013 European Conference on*, vol., no., pp.1,8, 11-12 June 2013.
- [32] FERREIRA, Adriano; "Desenvolvimento de Infra-Estrutura de Comando Multifunções EIB-KNX Para Smartphone"; Dissertação de Mestrado em Engenharia Industrial orientada pelo Prof. Doutor José Carvalho e apresentada na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança, em 2012.
- [33] WIMMER, Thomas; "ETS4 XML to Calimero 2 XML"; Vienna University of Technology; 2011.
- [34] "Building Technologies – Technical product data and descriptions"; disponível em: http://www.hqs.sbt.siemens.com/Lowvoltage/gamma_product_data/data/search_find_en.htm, consultado a 06/06/2013.
- [35] TANENBAUM, Andrew; "Computer Networks"; Prentice Hall Inc.; 2003.
- [36] "Arduino Mega 2560 Datasheet"; Documento publicado em <http://docs-europe.electrocomponents.com/webdocs/0e8b/0900766b80e8ba22.pdf> e consultado a 24/03/2013.
- [37] SIEMENS; "EIB-TP-UART-IC Technical Data"; 2001.
- [38] SIEMENS; "EIB-TP-UART 2-IC Technical Data"; 2012.
- [39] ON SEMICONDUCTOR; "NCN5120: KNX Transceiver for Twisted Pair Networks, Product Overview"; 2013.
- [40] FAIRCHILD; "Single Channel 6N137 Optocouplers Datasheet"; 2011.
- [41] "Arduino Ethernet Shield"; Artigo publicado em <http://arduino.cc/en/Main/ArduinoEthernetShield>, consultado a 24/03/2013.
- [42] "EEPROM Library"; Artigo publicado em <http://arduino.cc/en/Reference/EEPROM>, consultado a 28/05/2013.
- [43] "Ethernet Library"; Artigo publicado em <http://arduino.cc/en/Reference/Ethernet>, consultado a 28/05/2013.
- [44] WEINZIERL, Ing. Thomas; "Development kit fro EIB/KNX devices based on the TP-UART chip"; Weinzierl Engineering GmbH; 2004.
- [45] Chao Wang; Wei Duan; Jianzhang Ma; Chenhui Wang, "The research of Android System architecture and application programming," *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, vol.2, no., pp.785,790, 24-26 Dec. 2011.
- [46] Yan Ling Jie; Zheng Xiao Yi; Cao Da; Zheng Siting, "Development and implementation of Eclipse-based file transfer for Android Smartphone" *Computer Science & Education (ICCSE), 2012 7th International Conference on*, vol., no., pp.568,571, 14-17 July 2012.
- [47] FELKER, Donn; "Android Application Development for Dummies"; Wiley Publishing, Inc.; ISBN: 978-0-470-77018-4; 2011.

- [48] “*Android’s versions are named in alphabetic order*”; artigo publicado em: <http://www.knowahead.in/dyk/androids-versions-are-named-in-alphabetical-order/>, consultado a 02/08/2013.
- [49] Android Developer; “*Dashboards*”; publicado em: <http://developer.android.com/about/dashboards/index.html>, consultado a 10/08/2013.
- [50] “*About SQLite*”; consultado em <http://www.sqlite.org/about.html>, a 05/08/2013.
- [51] “*NMEA Protocol Specification*”; Versão 1.4; 2013.
- [52] “*Introducing JSON*”; publicado em <http://www.json.org/>, consultado a 27/08/2013.

Anexo

O anexo desta dissertação está dividido em dois conjuntos:

- Impresso, nas páginas seguintes com o seguinte índice:

ANEXO A. Constituição de uma aplicação Android

ANEXO B. PCB da interface RS232-KNX

ANEXO C. Desenho da API Android

ANEXO D. Lista de Materiais

- Em formato electrónico, com a seguinte estrutura de ficheiros:

\Datasheets	– Folhas de dados dos componentes utilizados.
\Esquemas Eléctricos	– Ficheiros em PDF de todos os esquemas eléctricos.
\Fotos	– Fotografias do trabalho.
\PCB	– Ficheiros em PDF do circuito impresso.
\MEEC_1070207_VFINAL.pdf	– Ficheiro em PDF deste documento.
\MEEC_1070207_ANEXOS_VFINAL.pdf	– Ficheiros em PDF dos anexos deste documento.