



Projeto

Mestrado em Engenharia Informática - Computação Móvel

*Soluções de Controlo de Dispositivos para
Automação Residencial*

José Manuel Palma Brites

Leiria, Março de 2016



Projeto

Mestrado em Engenharia Informática - Computação Móvel

*Soluções de Controlo de Dispositivos para
Automação Residencial*

José Manuel Palma Brites

Projeto de Mestrado realizado sob a orientação do Doutor Luís Filipe Fernandes Silva
Marcelino, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de
Leiria.

Leiria, Março de 2016

Agradecimentos

O autor deste projeto, gostaria de agradecer, em primeiro lugar, ao Professor Doutor Luís Marcelino pelo seu contributo e apoio prestado no seu desenvolvimento.

Em segundo lugar gostaria de agradecer a toda a equipa envolvida no projeto IVIT, desde os seus promotores aos professores e colegas do departamento de Engenharia Eletrotécnica, em especial ao colega Gilberto Jorge pelo seu contributo vital no desenvolvimento de todo o hardware aqui utilizado.

Ficam também os agradecimentos a todos aqueles que, de forma direta ou indireta apoiaram todo este trabalho de desenvolvimento.

Resumo

Nos últimos anos temos assistido ao aparecimento de novos dispositivos inteligentes e sistemas de automação residencial. Com o objetivo de conectar e controlar equipamentos eletrônicos, dentro de uma habitação, existe a necessidade de desenvolver soluções integradas de controlo remoto. Os smartphones, apresentam todas as características, poder computacional e portabilidade, ideais para controlo e monitorização deste tipo de dispositivos. Quando se considera o desenvolvimento de um produto novo e dependendo do problema, escolher a abordagem e tecnologias mais adequadas nem sempre é fácil.

Este projeto apresenta um conjunto de tecnologias e soluções de controlo baseadas em arquiteturas móveis. Para prova de conceito, e baseado num sistema real (IVIT), é proposta uma solução inovadora usando um smartphone. O IVIT, desenvolve uma tecnologia para aplicar num reservatório de aquecimento de água de inércia variável, com funcionalidades de monitorização, controlo e operação colaborativa do sistema de aquecimento.

Os resultados obtidos comprovaram que o desempenho da aplicação ultrapassou as expectativas e que a solução proposta é uma alternativa viável para o controlo de dispositivos para automação residencial usando smartphones.

Palavras-chave: Automação Residencial, Internet das Coisas, Dispositivos Inteligentes, Dispositivos Móveis, Wi-Fi, Bluetooth.

Abstract

In recent years we have witnessed an emergence of new smart devices and home automation systems. Connecting and controlling all electronics within a home, has created a real necessity for integrated remote control platforms. Smartphones' portability and capability provide the ideal interface to wirelessly monitor and control such smart devices. When considering a new product and depending on the problem, choosing the right approach and technologies can be difficult at times.

This project presents multiple technologies and remote control solutions that can be applied in a mobile-based architecture. For proof of concept and based in a real system (IVIT), we've proposed an innovative mobile solution. IVIT, develops a technology for a variable inertia water heating tank capable of monitoring, control and collaborative operation functions.

Results have shown that application performance exceeded expectations and that the proposed solution is a viable alternative to control devices for home automation using smartphones.

Keywords: Home Automation, Internet of Things, Smart devices, Smartphones, Wi-Fi, Bluetooth

Lista de Figuras

2.1	Alternativas de estratégias de controlo remoto	13
4.1	Arquitetura do Sistema de Controlo Remoto	30
4.2	Visão Geral do Sistema de Controlo Remoto	36
5.1	Placa com módulo Wi-Fi	40
5.2	Processo de Compilação do web server	42
5.3	Vistas principais da aplicação móvel	45
6.1	Comparação do tempo de resposta entre dispositivo real e simulador . .	50
6.2	Comparação dos tempos de decoding de conteúdos JSON	51
6.3	Comparação do tempo de resposta no dispositivo real e simulador simul- taneamente	52

Lista de Tabelas

1.1	Atividades e tarefas IVIT	4
2.1	Protocolos de Comunicação sem fio	9
2.2	Frameworks e APIs para Automação Residencial	23

Lista de Siglas

HVAC	Heating, Ventilation and Air Conditioning
IoT	Internet of Things
WPAN	Wireless Personal Area Networks
API	Application Programming Interface
NFC	Near Field Communication
TX	Transmit
RX	Receive
ISM	Industrial, Scientific and Medical
BLE	Bluetooth Low Energy
IP	Internet Protocol
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
JSON	JavaScript Object Notation
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
HAP	Apple's Home Automation Protocol
WP	Windows Phone
TCP	Transmission Control Protocol
HTTP	Hypertext Transfer Protocol
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
SDP	Service Discovery Protocol
UPnP	Universal Plug and Play
WSDL	Web Services Description Language
RPC	Remote Procedure Call
XML	eXtensible Markup Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security
IDE	Integrated Development Environment
SMTP	Simple Mail Transfer Protocol
URI	Uniform Resource Identifier

Índice

Agradecimentos	III
Resumo	V
Abstract	VII
Lista de Figuras	IX
Lista de Tabelas	XI
Lista de Siglas	XIII
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	3
1.3 Metodologia de trabalho	3
1.4 Estrutura do documento	5
2 Estado da Arte	7
2.1 Protocolos de Comunicação	8
2.1.1 Bluetooth/Bluetooth Smart	8
2.1.2 Wi-Fi - IEEE 802.11x	10
2.1.3 ZigBee - IEEE 802.15.4	11
2.1.4 INSTEON	11
2.1.5 Análise Comparativa	12
2.2 Estratégias de Controlo Remoto	13
2.2.1 Abordagem Local	13
2.2.2 Abordagem Distribuída (Cloud-based)	14
2.2.3 Soluções Híbridas	15
2.3 Ferramentas, Frameworks e APIs	16
2.3.1 Nest / Works with Nest	16
2.3.2 SmartThings	17
2.3.3 Apple HomeKit	18
2.3.4 Wink	20
2.3.5 Outros projetos	21
2.4 Análise Comparativa	22
3 Levantamento de Requisitos	25
3.1 Requisitos do Projeto IVIT	25
3.2 Requisitos Adicionais	26
4 Solução Proposta	29
4.1 Arquitetura do Sistema proposto	29

4.2	Tecnologias	31
4.2.1	Zero-configuration	31
4.2.2	REST	32
4.2.3	JSON	32
4.2.4	HTTPS	33
4.2.5	iOS	33
4.2.6	Microchip TPC/IP Stack Web Server	34
4.3	Visão Geral do Sistema Proposto	36
5	Implementação	39
5.1	Placa de Controlo	39
5.1.1	Micro-controlador/Microprocessador	39
5.1.2	Wi-Fi Transceiver	40
5.1.3	Configurações	40
5.1.4	Web Server	41
5.2	Aplicação móvel	42
5.2.1	Modelo de dados	42
5.2.2	Módulo de Configuração e descoberta de dispositivos	44
5.2.3	Módulo de Comunicação e sincronização de dados	45
5.2.4	Módulo de Apresentação	46
6	Testes	47
6.1	Caraterização dos dispositivos de teste	47
6.2	Experiência preliminar	48
6.3	Testes de conectividade	49
6.4	Testes de performance	51
6.5	Análise de resultados	52
7	Conclusão	55
7.1	Considerações Finais	55
7.2	Trabalho Futuro	56
	Anexos	61

Capítulo 1

Introdução

Este capítulo introduz o contexto deste projeto, descrevendo o seu enquadramento na temática objeto de estudo, apresentando os objetivos principais a cumprir e qual a metodologia de trabalho usada. Por fim, descreve-se a estrutura deste documento.

1.1 Enquadramento

Esta secção apresenta o trabalho realizado no âmbito do Projeto integrado no segundo ano do Mestrado em Engenharia Informática - Computação Móvel da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Inserido no projeto IVIT - Reservatório de Inércia Variável com Controlo Integrado de Várias Fontes de Energia, uma tecnologia que consiste num reservatório de aquecimento de água de inércia variável com funcionalidades de monitorização, controlo e operação colaborativa do sistema de aquecimento. Estas funcionalidades, que incluem comunicação com outros equipamentos e plataformas de gestão, assumem um papel estratégico no desenvolvimento da tecnologia e constituem uma inovação neste tipo de equipamento.

Devido à sua complexidade, esta tecnologia requer o desenvolvimento de interfaces de controlo e monitorização inovadoras que permitam ao consumidor ou ao operador fazer uma gestão eficiente dos recursos, de forma fácil e consistente.

Enquadrado na temática da Automação Residencial, surge a necessidade de aplicar conceitos inovadores e que sejam facilmente integrados em sistemas já implementados. Esta concordância pode ser obtida através do estudo das tecnologias e das plataformas

que podem, previsivelmente, contribuir para a criação do sistema de controlo ideal.

O conceito de Automatização Residencial representa a introdução de tecnologia dentro de uma habitação [1]. Destina-se a melhorar a qualidade de vida dos seus ocupantes, melhorando o conforto e segurança, aumentando a eficiência energética e reduzindo custos. Esta tecnologia é o que nós descrevemos como dispositivos inteligentes, e pode incluir, desde sistemas de entretenimento, a lâmpadas, fechaduras, persianas e sistemas de aquecimento, ventilação e ar condicionado, também designados pela sigla inglesa HVAC (*Heating, Ventilation and Air Conditioning*). O que os torna inteligentes, são as funcionalidades adicionais que trazem, e toda a orquestração que estes dispositivos digitais permitem, de forma a criar o ambiente desejado.

Certificar-se que estes possam funcionar corretamente e alcançar pleno potencial, sendo confiáveis, fáceis de usar e de baixo custo, não é tarefa fácil. Além disso, controlar estes dispositivos deve ser uma experiência agradável, que deve ser conseguida através do uso de uma única plataforma ou controlador centralizado, e não com dezenas de dispositivos de controlo remoto que encontramos na maioria das nossas habitações.

A popularização e o crescente uso de dispositivos móveis, como smartphones e tablets, no dia-a-dia do cidadão comum, e o seu poder computacional, fazem com que esta seja a plataforma ideal para controlo remoto deste tipo de tecnologia. Pequenos mas poderosos, são dispositivos que cabem na palma de uma mão, capazes de processamento rápido, armazenamento de dados e, mais importante, que permitem ligações sem fios através de Bluetooth ou Wi-Fi. Além disso, o ecrã tátil e a familiaridade com o dispositivo, tornam a experiência de utilizador mais natural e intuitiva.

O conceito de Internet das Coisas (IoT), integra todos os tipos de deteção, identificação, comunicação, trabalho em rede, dispositivos e sistemas de informação. Ligando todas as pessoas e coisas de acordo com os seus interesses, de modo a que qualquer pessoa, a qualquer hora e em qualquer lugar, através de qualquer dispositivo, possa aceder de forma mais eficiente às informações de qualquer objeto ou serviço. Desde que surgiu este conceito, tem existido um interesse em desenvolver tecnologias para o tornar uma realidade [2].

1.2 Objetivos

De acordo com o que foi descrito anteriormente e em linha com a proposta deste projeto, foram definidos um conjunto de objetivos essenciais para o seu desenvolvimento e conseqüente sucesso.

Na primeira fase, e após, o estudo das várias tecnologias relacionadas com automação residencial, nomeadamente, que protocolos de comunicação são utilizados, em que situações se aplicam e quais as suas vantagens e desvantagens. As diferentes arquiteturas, estratégias de controlo remoto, ferramentas e *frameworks* disponíveis, que potenciem e facilitem a comunicação, gestão e controlo de dispositivos inteligentes, são também objetos de estudo. O primeiro objetivo, resultado deste estudo, será a definição de um protocolo de comunicação com o reservatório de aquecimento.

O segundo objetivo, passa pela criação e avaliação de uma prova de conceito, que implemente algumas das tecnologias estudadas e que resulte num protótipo funcional, sob a forma de uma aplicação móvel, que permita a monitorização e controlo do reservatório de aquecimento.

A avaliação do sistema implementado passa pela realização de testes de conectividade e performance que verifiquem e qualifiquem a viabilidade da arquitetura definida avaliando o sistema em várias situações de funcionamento. Por fim, os resultados obtidos devem ser analisados e discutidos.

Resumidamente, os objetivos são:

- Definição do protocolo de comunicação com o dispositivo;
- Desenho e implementação de uma aplicação móvel para monitorização e controlo;
- Realização de testes de conectividade e performance, e análise de resultados;

1.3 Metodologia de trabalho

Enquadrado no projeto IVIT, que envolve várias componentes foi necessária uma coordenação mais próxima com o departamento de Engenharia Eletrotécnica da Escola

Tabela 1.1: Atividades e tarefas IVIT

A.9	Investigação sobre as interfaces externas de gestão do sistema
T.9.1	Interface de gestão de dispositivos.
T.9.2	Definição dos protocolos de comunicação com o dispositivos.
T.9.3	Definição das especificações dos serviços web para fornecedores de serviço.
T.9.4	Definição das especificações de um portal para clientes finais.
A.10	Desenvolvimento das interfaces externas de gestão do sistema
T.10.1	Implementação de drivers para permitir ligar o dispositivo via IP.
T.10.2	Implementação de um serviço web para fornecedores de serviços.
T.10.3	Implementação de um portal para clientes finais.
T.10.4	Implementação de uma API para interligar o dispositivos com outras plataformas.
A.11	Teste das interfaces externas de gestão do sistema.
T.11.1	Testes funcionais dos sistemas implementados.
T.11.2	Testes de usabilidade dos sistemas implementados.
T.11.3	Testes de performance em carga dos sistemas implementados.

Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria (IPL). Encarregue do desenvolvimento do hardware para o módulo de controlo do reservatório de aquecimento, a colaboração desta equipa revelou-se fundamental em praticamente todas as fases de desenvolvimento, desde a definição do protocolo de comunicação aos testes de campo efetuados.

Organizado por atividades, a coordenação deste projeto atribuiu ao IPL, nomeadamente aos departamentos de Engenharia Eletrotécnica e Engenharia Informática, um conjunto de atividades, das quais se destacam, a Atividade 9 (A.9), Atividade 10 (A.10) e Atividade (A.11), apresentadas na tabela 1.1. Relacionadas com este trabalho em concreto, as tarefas a realizar são: a Tarefa 9.2 (T.9.2) e Tarefa 9.4 (T.9.4), Tarefa 10.3 (T.10.3) e Tarefa 10.4 (T.10.4), e Tarefa 11.3 (T.11.3). Algumas das tarefas atribuídas inicialmente não foram contempladas neste projeto, acabando por ser atribuídas a outra equipa de trabalho exterior ao IPL.

A metodologia de desenvolvimento, teve portanto de ser enquadrada neste esforço entre equipas.

As restantes tarefas podem ser consultadas, na listagem de atividades e *milestones* do projeto IVIT, que se encontra no Anexo 1.

1.4 Estrutura do documento

Este documento encontra-se estruturado em sete capítulos, sendo o primeiro a introdução. No capítulo segundo é feito um estudo acerca do estado da arte da temática de automatização residencial. Serão apresentadas as tecnologias e protocolos de comunicação, abordagens de controlo remoto e ferramentas, frameworks e/ou APIs.

Seguidamente, o capítulo terceiro apresenta um conjunto de requisitos que devem ser aplicados na solução proposta e no desenvolvimento do projeto, requisitos estes que se baseiam nas conclusões retiradas do capítulo segundo.

No capítulo quarto, será apresentada a solução proposta para a implementação do protótipo funcional, onde são descritas as tecnologias usadas e a forma como estas se enquadram arquitetura proposta.

O capítulo quinto está relacionado com a implementação da solução proposta, descrevendo o hardware utilizado pela equipa de eletrotecnia e desenvolvido única e exclusivamente para o projeto IVIT. Este capítulo apresenta também algumas das funcionalidades e limitações do Web Server disponibilizado pela plataforma de desenvolvimento de hardware. Por fim, e ainda neste capítulo será a apresentado o protótipo de aplicação móvel que servirá de prova de conceito.

No capítulo sexto serão apresentados os testes de conectividade e performance efetuados ao protótipo.

Por fim, no capítulo sétimo é feita uma conclusão de todo o trabalho desenvolvido no âmbito deste projeto e do que poderá vir a ser desenvolvido num trabalho futuro.

Capítulo 2

Estado da Arte

Com a popularização dos sistemas de automação residencial, o conceito de Internet of Things (IoT) pode ser aplicado aos dispositivos integrantes destes sistemas. Este conceito pressupõe que todos estes dispositivos são identificados inequivocamente, para que possam comunicar entre si usando a infra-estrutura de rede/Internet existente.

A ligação destes dispositivos à rede/Internet é só o primeiro passo. Conseguir comunicar com estes dispositivos, enviando-lhes comandos a partir de um dispositivo de controlo à distancia ou conseguir que se adaptem às alterações do ambiente em que se encontram, é o que torna estes sistemas ainda mais interessantes.

Com o aparecimento de todos estes novos dispositivos “inteligentes” criados por um vasto leque de fabricantes, cada um com a sua abordagem, surge a necessidade de criar soluções para unificar e compatibilizar estes dispositivos. Protocolos de comunicação, standards e frameworks, de forma a que estes possam facilmente, comunicar entre si, permitindo assim a integração de novos dispositivos em sistemas de automação já existentes e nas plataformas de controlo remoto disponíveis.

Dos diferentes protocolos de comunicação criados especificamente para a automação residencial sob a forma de WPANs (Wireless Personal Area Networks), destacam-se o ZigBee, Z-wave, X10, Insteon, EnOcean e algumas variantes com propósitos específicos. Estes protocolos oferecem, em alguns casos, uma solução mais barata e eficiente em termos de consumo de energia em comparação com outras WPANs mais usuais tais como o Bluetooth ou Wi-Fi [3].

Associado à comunicação com estes dispositivos, o controlo dos mesmos através de aplicações cliente, é também um problema que só a criação de APIs e frameworks, pode resolver.

Esta necessidade despertou a atenção de grandes marcas do setor tecnológico como a Google, Samsung, Apple e GE, que começaram a procurar soluções para criar ou disponibilizar frameworks de automação que possam ser usadas pelos dispositivos móveis atualmente existentes.

Este capítulo apresenta, descreve e compara as principais tecnologias e protocolos de comunicação utilizados nesta temática e, quais os princípios arquiteturais e as abordagens utilizadas pelos sistemas de automatização residencial atualmente existentes. São também apresentadas e descritas algumas ferramentas como, APIs e frameworks, que permitem acrescentar funcionalidade aos dispositivos, potenciando a sua orquestração.

2.1 Protocolos de Comunicação

Desenvolvido em 1975, o protocolo X10, está identificado como sendo o primeiro standard criado para sistemas de automatização residencial [4]. Esta tecnologia usa a instalação elétrica de uma habitação para a comunicação com os dispositivos.

Em abordagens mais recentes, é promovido o uso de standards de comunicação compatíveis com smartphones, quer por comunicação direta, ou usando algum tipo de hub como ponto de acesso. Das várias tecnologias de comunicação que equipam os smartphones atualmente no mercado, para além das tradicionais, existem outras como NFC e infravermelhos, normalmente usadas em comunicações de curta distancia, pelo que não serão mencionadas neste estudo.

A ligação com o smartphone é então, na maioria dos casos, garantida pelo uso de protocolos como Wi-Fi ou Bluetooth/Bluetooth Smart. Alguns fabricantes já começaram a incluir nos seus produtos, mecanismos de controlo usando um smartphone.

Esta secção apresenta as principais características das tecnologias de comunicação mais utilizadas pela maioria dos atuais sistemas de automação residencial.

2.1.1 Bluetooth/Bluetooth Smart

Bluetooth é uma tecnologia de comunicação sem fios, criada pela Ericsson em 1994, que permite que dispositivos comuniquem sem fios através de redes ad-hoc de curto

Tabela 2.1: Protocolos de Comunicação sem fio

	ZigBee IEEE 802.15.4	Wi-Fi IEEE 802.11b/g/n/ac	Bluetooth Smart	INSTEON
Data Rate	250 kbps	100 Mbps-1000 Mbps	1 Mbps	38.4 kbps
Range	10 - 100m	20 - 100m	10 - 100m	50m
Frequency	915 MHz US 868 MHz EUR 2.4 GHz World	2.4 and 5 GHz	2,4 GHz	915 MHz US 868 MHz EUR 921 MHz AUS
Interference	dynamic freq. selection	dynamic freq. selection	adapting freq. hopping	-
Power	90mW TX 72mW RX 3uA sleep	1.28W TX 0,94W RX 46mW sleep	84mW TX 66mW RX 0,2mW sleep	-
Cost	médio	alto	baixo	baixo

alcance, conhecidas como piconets. O Bluetooth foi criado com a intenção de substituir os cabos que conectam dispositivos pessoais, mantendo elevado o nível de segurança. As principais características desta tecnologia são a sua ubiquidade, baixo consumo de energia e baixo custo. Bluetooth opera na banda industrial, científica e médica (ISM) a 2,4 GHz, disponível e não licenciada na maioria dos países. Porque opera na mesma banda que outras tecnologias, a interferência deve ser tida em conta. O Bluetooth, usa o conceito de Adaptive Frequency Hopping, que trabalha dentro do espectro tirando partido da frequência disponível. Ou seja, deteta outros dispositivos no espectro e tenta evitar as frequências que estão a usar.

Em termos de alcance, depende da implementação e classe de rádio usada pelos fabricantes. Na maioria dos casos o alcance é de cerca de 10 metros, sendo o limite teórico até 100 metros. A taxa de transferência de dados máxima é de cerca de 1 Mbps e o consumo de energia, em rádios mais utilizados é de cerca de 2,5 mW de potência. Devido à aceitação global da tecnologia Bluetooth e da grande procura de dispositivos de controlo remoto sem fios, o consumo de energia torna-se uma preocupação importante.

Bluetooth Smart ou Bluetooth Low Energy (BLE) é descrito como a versão inteligente e de baixo consumo da tecnologia Bluetooth, direcionada para dispositivos que funcionam com pequenas baterias por longos períodos de tempo e é compatível com aplicações para smartphones e tablets já existentes no mercado. Projetado para maximizar o tempo de vida da bateria, utiliza cerca de 1/2 da potência da tecnologia Bluetooth clássica mas pode baixar para cerca de 1/10 dependendo do caso de uso [5]. Siekkinen et al. [6] estudaram o consumo de energia real da tecnologia BLE, medindo dispositivos reais com um medidor de potência. Os resultados mostraram que o BLE

consome de facto muito pouca energia, apresentando um rácio de energia por bit transmitido bastante interessante. No entanto, pode ser ainda mais otimizado permitindo o envio de mais pacotes ao longo de uma ligação e implementando mecanismos de combate à interferência.

2.1.2 Wi-Fi - IEEE 802.11x

Wi-Fi é o nome de uma popular tecnologia de rede sem fios que usa ondas de rádio para fornecer ligações de rede de alta velocidade. A Wi-Fi alliance define Wi-Fi como qualquer produto de rede local sem fios baseado nos padrões IEEE 802.11x. Wi-Fi é usado em muitos dispositivos, como computadores pessoais, smartphones e tablets, o que lhes permite ligar-se a qualquer recurso na rede, normalmente a Internet, através de um ponto de acesso de rede sem fios. Wi-Fi também permite comunicação direta de um dispositivo para outro, sem ter um ponto de acesso como intermediário, criando-se uma rede ad-hoc. Com a capacidade de se ligar a qualquer recurso de rede, o controlo de dispositivos pode ir para além do controlo remoto local. Permite o acesso remoto a partir da Cloud fornecendo a capacidade de controlar estes dispositivos em qualquer lugar do mundo.

Tal como o Bluetooth, também o Wi-Fi usa a banda ISM de 2,4 GHz, mas devido ao seu uso massificado, os padrões de implementação mais recentes como o 802.11n e o mais recente 802.11ac, são dual-band, operando em ambos os 2,4 e 5 GHz. Dual-band duplica a capacidade, permitindo que os dispositivos usem, os menos usados 5 GHz para aplicações de alto desempenho, e 2,4 GHz para o uso tradicional. Com estas implementações mais recentes, verificou-se um aumento nas taxas de transferência de dados para 100 Mbps, mas que podem teoricamente chegar a 1000 Mbps. Em termos de alcance pode variar entre 20 metros em espaços fechados e pode ir até 100 metros em espaços abertos [7].

Halperin et al. [8] desmistificaram o consumo de energia do Wi-Fi. Usando um adaptador de rede 802.11n com diferentes configurações, chegaram à conclusão que para configurações, como dobrar a largura de banda, apresentam um impacto muito baixo no consumo, ou outras, como adicionar uma cadeia de transmissão, que pelo contrário, aumentam significativamente o consumo. Em todo o caso, comparado com o Bluetooth, o Wi-Fi apresenta um consumo bastante elevado.

2.1.3 ZigBee - IEEE 802.15.4

Baseado no padrão IEEE 802.15.4, o ZigBee é uma tecnologia de rede sem fios, de topologia mesh, de baixo custo e baixo consumo de energia, usada em automação residencial para controlo remoto e monitorização de aplicações. ZigBee é uma tecnologia criada para ser uma alternativa mais simples e mais barata do que outras soluções de redes de área pessoal sem fios, como Bluetooth e Wi-Fi.

Tal como as anteriores também opera na banda ISM de 2,4 GHz em todo o mundo. Opcionalmente, pode ser utilizado a 915 MHz, nos EUA e Austrália, 868 MHz na Europa e a 784 MHz em China. As taxas de transferência de dados podem variar entre 20 Kbps a 250 Kbps. O seu baixo consumo de energia e especificação aberta, tornam o ZigBee a tecnologia de comunicação perfeita para pequenos dispositivos que dependam de baterias por longos períodos de tempo. Apesar de limitar o alcance da transmissão entre 10 e 100 metros, dependendo da potência e características ambientais, devido à sua topologia de rede mesh, estes podem comunicar com outros que atuam como repetidores, permitindo a transmissão de dados através de longas distâncias [9]. O consumo de energia destes equipamentos é muito semelhante ao BLE, como podemos observar na tabela 2.1.

A interoperabilidade do ZigBee consegue-se através da padronização a todos os níveis da rede, especialmente ao nível da aplicação. Por essa razão o ZigBee disponibiliza normas de aplicação para fins específicos, garantindo que os dispositivos de diferentes fabricantes possam trabalhar em conjunto. Por exemplo, a Phillips e outras marcas contribuíram com a sua experiência para o desenvolvimento do ZigBee Light Link, destinado ao controlo remoto de produtos de iluminação. [10]

2.1.4 INSTEON

INSTEON é uma tecnologia proprietária que combina a instalação elétrica de uma habitação, com um sistema sem fios. Os dispositivos comunicam entre si utilizando o protocolo INSTEON através de rádio frequência a 915MHz, nos EUA, 868 MHz na Europa e 921 MHz na Austrália e sobre a instalação elétrica, numa topologia de rede full mesh. As taxas de transferência de dados podem variar entre 13.1 Kbps e 38.4 Kbps. Apesar de limitar o alcance da transmissão a 50 metros em espaços abertos, devido à sua topologia de rede, é possível transmitir em distâncias maiores usando repetidores [11].

Esta tecnologia foi desenvolvida para substituir o X10, um dos padrões de automação residencial mais antigo.

2.1.5 Análise Comparativa

Após descrever as tecnologias e protocolos de comunicação sem fio, mais utilizadas para controle remoto de dispositivos, podemos fazer uma breve análise destacando o potencial das tecnologias apresentadas e relatar as vantagens e desvantagens de cada uma, quando aplicadas no controle remoto usando um Smartphone.

Como é bem conhecido, os smartphones atualmente no mercado estão equipados com algumas das mais recentes tecnologias de comunicação, incluindo Wi-Fi e Bluetooth.

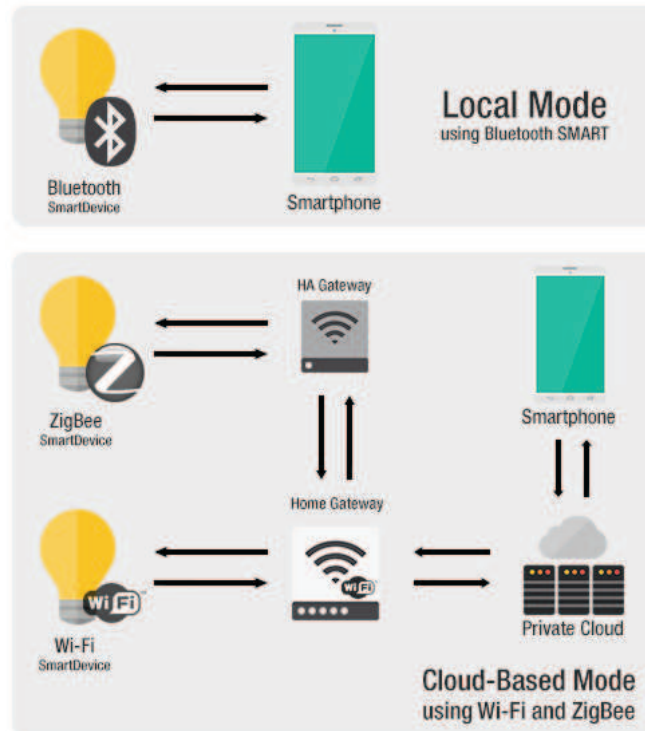
Cada uma das tecnologias descritas anteriormente apresenta as suas vantagens quando comparada com outras. Como indica a Tabela 2.1, o ZigBee é uma das tecnologias mais eficientes energeticamente e com baixo custo, mas como tal como o INSTEON, requer hardware adicional para traduzir dados transmitidos usando ZigBee ou o protocolo INSTEON para um padrão de comunicação compatível com smartphones.

Bluetooth e Wi-Fi são as únicas tecnologias de comunicação que permitem isso mesmo, seja diretamente ou facilitadas por hardware adicional. Para além disso são as únicas que permitem a transmissão de maiores volumes de dados. A secção seguinte, descreve diferentes abordagens usando Bluetooth ou Wi-Fi. O Zigbee será apenas mencionado para efeitos de ilustração.

Os dados apresentado na Tabela 2.1 foram compilados de várias fontes, motivo pelo qual não se encontra referenciada diretamente.

O Anexo 3 apresenta o artigo científico redigido em conjunto com este capítulo. Intitulado "Remote Control of Smart Devices using Smartphones", este artigo apresenta uma comparação mais detalhada de tecnologias de comunicação sem fios.

Figura 2.1: Alternativas de estratégias de controlo remoto



2.2 Estratégias de Controlo Remoto

Após descrever os principais protocolos de comunicação e as suas características, é importante entender como estes podem ser usados numa arquitetura de comunicação para controlo remoto de dispositivos inteligentes sem fio, usando um smartphone, e quais os desafios técnicos envolvidos. Das várias arquiteturas estudadas destacam-se as abordagens locais, onde os dispositivos se encontram interligados diretamente ou através de um gateway local. Abordagens distribuídas (Cloud-based) que pressupõem a existência de um servidor Cloud que permite controlar dispositivos remotamente. A figura 2.1 ilustra os dois tipos de arquitetura usando 3 protocolos de comunicação diferentes, Bluetooth, ZigBee e Wi-Fi.

2.2.1 Abordagem Local

Como o nome sugere, uma abordagem local elimina a necessidade de um servidor, permitindo que os dispositivos comuniquem diretamente entre si. Livrar-se da dependência da Internet pode reduzir significativamente alguns problemas de segurança e privaci-

dade eliminando qualquer tentativa de controle remoto malicioso fora da rede local. No entanto, eliminado qualquer ligação com o exterior pode tornar-se um problema. A configuração e manutenção do sistema tem de ser feita localmente, tornando mais difícil para fornecedores de sistemas e provedores de serviços monitorizar, diagnosticar e atualizar todos os seus produtos, quando são detetados problemas.

Esta abordagem, mantém os fornecedores de sistemas de automação residencial, atualmente estabelecidos, com controlo dos seus dispositivos, preservando assim, os seus papéis essenciais na cadeia de valor da automação residencial. Não estando dependente frameworks ou APIs de terceiros, emergentes no mercado de automação residencial, elimina quaisquer problemas que possam fazer o produto funcionar mal ou induzir em erro o consumidor [12].

O Bluetooth é geralmente a tecnologia escolhida nesta abordagem, devido ao seu baixo consumo de energia, baixo custo e ao mesmo tempo à sua arquitetura de rede segura. Isto permite a criação de dispositivos baratos e ao mesmo tempo confiáveis, e que podem ser perfeitamente integrados com um smartphone num processo de emparelhamento simples.

Produtos como lâmpadas inteligentes, por exemplo a Elgato que criou a AVEA, uma lâmpada inteligente que pode ser controlada usando um dispositivo iOS. O emparelhamento é um processo fácil, típico do Bluetooth, que não necessita da configuração de equipamentos adicionais. Um único equipamento iOS, pode controlar até 10 lâmpadas independentemente ou em simultâneo, criando o ambiente desejado [13]. Usando a mesma tecnologia, a MISFIT criou a BOLT, outra lâmpada inteligente que pode ser controlada por qualquer dispositivo iOS ou Android equipado com bluetooth 4.0 ou superior. Apresenta as mesmas características da anterior, mas sem a limitação do número de lâmpadas [14].

2.2.2 Abordagem Distribuída (Cloud-based)

A necessidade de uma representação digital do mundo real, onde todos os tipos de deteção, controlo e comunicação de rede estejam perfeitamente conectados com os utilizadores, de modo que qualquer um, a qualquer hora e em qualquer lugar, através de qualquer dispositivo, possa aceder, controlar e monitorizar qualquer objeto, tem impulsionado o mercado do consumidor para um crescente interesse em soluções de automação residencial distribuídas [2].

A Cloud permite criar esta ubiquidade, ou seja, permite que os dispositivos estejam presentes em toda a parte ao mesmo tempo. Ligar estes dispositivos a um servidor na Cloud facilita o controle remoto e gestão eficiente a partir de qualquer lugar, minimizando também o esforço dos consumidores durante o processo de instalação e configuração. Permite também que, fabricantes e prestadores de serviços possam comunicar remotamente com os seus dispositivos, para manutenção remota ou simplesmente para monitorização e recolha de dados.

Por outro lado, este tipo de solução trás preocupações acrescidas no que diz respeito à segurança e privacidade dos utilizadores, sendo um desafio acrescido para fabricantes e que deve ser tido em conta. Soluções distribuídas que implicam comunicação IP, acrescem todos os desafios de segurança e configuração próprios do IP. A comunicação segura deve, por isso, ser garantida.

2.2.3 Soluções Híbridas

Apresentados os prós e contras de cada abordagem, local e distribuída, é fácil entender o porquê da criação de soluções híbridas que explorem o melhor de cada abordagem. Simplificando a configuração e manutenção e minimizando as questões de segurança e privacidade.

Pang et al. [12] propuseram uma arquitetura híbrida baseada em IP capaz de tirar partido do melhor de cada uma das abordagens apresentadas anteriormente, criando uma combinação flexível. Com uma arquitetura escalável e tirando partido de um gateway, o protótipo criado baseou-se no conceito 6LoWPAN (IPv6 sobre redes de área pessoal sem fio de baixa potência), trazendo as vantagens IP para dispositivos de baixo custo e baixo consumo de energia.

Mais recentemente, alguns fabricantes e empresas do setor tecnológico, que disponibilizam produtos, ferramentas e frameworks para fins de automação residencial, têm envidado esforços para a criação de arquiteturas híbridas, facilmente escaláveis e de fácil configuração. Na secção seguinte serão apresentadas e descritas algumas destas frameworks, quais as suas características arquiteturais e que opções oferecem aos consumidores finais.

2.3 Ferramentas, Frameworks e APIs

Após descrever e comparar, os vários protocolos de comunicação e as várias abordagens de controlo remoto, esta secção apresenta um conjunto de frameworks e APIs que oferecem soluções arquiteturais integradas que permitem tirar o máximo partido dos dispositivos e do que a sua orquestração pode proporcionar. No final é feita uma comparação entre as ferramentas apresentadas.

2.3.1 Nest / Works with Nest

Os laboratórios Nest disponibilizam uma API de dados em tempo real que oferece um modelo de subscrição para aceder aos dados partilhados pelos dispositivos Nest (Termóstato). Esta API permite criar aplicações que podem aceder aos dados partilhados pelos dispositivos, ler o modificar os seus valores.

As aplicações que usam o serviço Nest permitem que os utilizadores possam estar em contacto com a sua casa em qualquer lugar. Todos os dispositivos Nest (Termóstatos, detetores de fumo e alarmes CO₂) e aplicações cliente (iOS, Android e Web) estão ligadas ao serviço Nest. Este serviço guarda todo o estado do sistema num documento JSON, e cada vez que existe uma alteração, o serviço notifica todos os clientes que subscreveram um determinado objeto que ocorreram alterações de estado. Esta sincronização é garantida pelo serviço Firebase, uma API REST bastante usada para sincronização em tempo real ou pela API REST Nest. [15]

Por exemplo, podemos criar uma aplicação cliente para controlar o termóstato Nest e controlar os atributos de “temperatura atual” e “temperatura pretendida”. Para controlar mudanças nestes 2 atributos, só temos de subscrever o objeto termóstato e quando o utilizador ajustar a temperatura do termóstato, o atributo “temperatura pretendida” altera, e o serviço nest atualiza o documento JSON, que será sincronizado em tempo real. A aplicação cliente fica a espera de alterações e mostra os valores reais de temperatura atuais.

O potencial desta plataforma captou a atenção da gigante Google, empresa esta que anunciou, em 2011 na Google I/O uma framework para home automation, denominado por Android@Home, um projeto abandonado, mas que foi agora revitalizado pela aquisição dos laboratórios Nest por parte da Google em Janeiro de 2014.[16]

2.3.2 SmartThings

SmartThings, uma start-up criada em 2012, oferece uma plataforma para o chamado “Open Physical Graph”, ou seja a representação virtual, online do mundo físico. Ao interagir com o “physical graph”, estamos automaticamente a interagir com o mundo físico e vice-versa. É uma plataforma aberta, facilmente acessível por consumidores, developers e fabricantes de dispositivos.

Permite de forma fácil e “out of the box”, transformar uma casa comum, numa casa inteligente sem necessidade de recorrer a técnicos especializados. Para isso, esta plataforma oferece uma solução ponto a ponto, de hardware, software, experiência de utilizador e suporte.

Esta plataforma disponibiliza um hub, ou seja um dispositivo físico que serve de intermediário entre as várias aplicações cliente e os dispositivos físicos. Seguindo uma abordagem “Cloud First”, esta plataforma requer, para já, que este hub esteja sempre online e ligado à cloud.

Em relação aos protocolos suportados por este hub, e de forma a poder ser emparelhado com uma vasta gama de dispositivos de outras marcas atualmente existentes no mercado, o hub suporta protocolos como ZigBee, Z-Wave, IP(cabo ou Wi-Fi) ou dispositivos que requerem ligação a outros serviços cloud.

Em termos de arquitetura esta plataforma foi desenhada para criar uma abstração dos detalhes específicos dos dispositivos (ZigBee, Z-Wave, Wi-Fi/IP) permitindo aos developers focarem apenas nas capacidades e ações suportadas pelo dispositivo (ligar/desligar, ligado/desligado, etc). [17]

Conceptualmente, a arquitetura é composta por 5 camadas: os dispositivos físicos, que representam a capacidade de execução de ações, a camada da conectividade que representa a ligação entre a cloud e os dispositivos, a camada de abstração dos detalhes e capacidades dos dispositivos, a camada de inteligência representada pelas SmartApps ou SmartServices que são aplicações e serviços que controlam e processam os dados ou ações dos dispositivos e, por fim, a camada de apresentação representada pelas aplicações cliente. Como developer, esta plataforma permite que sejam criados vários tipos de componentes, usando uma linguagem própria denominada de Groovy.

As SmartApps são pequenos programas, que correm normalmente em background,

e que permitem criar funcionalidade entre os vários dispositivos, permitindo controlar hardware com software simples. Estas SmartApps podem ser tipicamente caracterizadas pela sua função. Um exemplo de SmartApp pode ser: "Desligar as luzes a uma determinada hora quando não for detetado movimento ". Claro que estas SmartApps permitem muito mais que simples ações. As SmartApps conseguem comunicar com serviços externos, enviar notificações push e SMS, etc.

Os Device-Type Handlers integram novos dispositivos com a plataforma, permitindo também personalizar a forma de apresentação do dispositivo na aplicação cliente. Devido á camada de abstração podemos preocupar-nos apenas com as capacidades genéricas dos dispositivos. Estes handlers podem interagir com vários tipos de dispositivos físicos. Dispositivos que comuniquem com o hub, através de ZigBee e Z-Wave, dispositivos geridos por serviços cloud externos ou dispositivos ligados á rede local, tipicamente por Wi-Fi. Uma vez conectado á rede, o hub, podemos comunicar com os dispositivos ao nível da rede, usando REST ou SOAP.

As Dashboard Solution Module SmartApps, permitem criar módulos de interface personalizados, dentro da aplicação disponibilizada, criando uma experiência de utilizador unificada para aplicações específicas, como por exemplo Casa e Família.

Integration ou Service Manager SmartApps, permitem chamar Webservices externos e disponibilizar Webservices que possam ser chamados por sistemas externos. Podem também ser aplicações com o objetivo de permitir a integração com a plataforma Smartthings.

Tal como a plataforma anterior, o potencial da SmartThings foi reconhecido por uma das grandes marcas do setor tecnológico, a Samsung, que acabou por adquirir esta empresa em Agosto de 2014. Apesar de adquirida por outra empresa, a SmartThings vai continuar a existir como uma empresa independente, mantendo a sua estratégia de desenvolvimento, fazendo agora parte do grupo Samsung's Open Innovation Center.[18]

2.3.3 Apple HomeKit

HomeKit é uma framework para iOS 8 que permite comunicar e controlar dispositivos inteligentes, presentes numa casa, que suportem o protocolo HAP (Apple's Home Automation Protocol), de forma transparente e totalmente integrada. Esta abordagem, promove o uso de um protocolo comum para dispositivos de automação residencial e

de uma API pública que permite configurar e comunicar com estes dispositivos, possibilitando assim a criação de apps que não têm necessariamente de ser disponibilizadas pelos fabricantes de dispositivos. Permite que todos os dispositivos possam ser integrados de forma coerente sem necessidade de coordenação entre si.

O HomeKit permite 3 funções principais nas suas aplicações:

- Descobrir acessórios (dispositivos), usando Bonjour, e adiciona-los a uma base de dados de configuração da casa.
- Mostrar, editar e agir sobre os dados presentes nessa base de dados.
- Comunicar com acessórios configurados e serviços, através de uma API RESTful, permitindo desencadear ações, como por exemplo aceder as luzes da sala.

Esta base de dados de configuração não está apenas disponível às aplicações, é também disponibilizada ao sistema de controlo por voz, Siri, o que permite aos utilizadores executar ações sobre os dispositivos de forma simples.[19]

Sendo esta uma framework recente que promove o uso de um protocolo comum, apenas os dispositivos que adotam este protocolo são compatíveis "out-of-the-box".

A framework disponibiliza um Home Manager que permite gerir, adicionar e remover Casas (Homes). Cada Casa contém várias Divisões (Rooms) que contém por sua vez vários Acessórios (Accessories). Estes Acessórios correspondem a dispositivos físicos atribuídos a uma divisão onde cada um deles apresenta uma série de Serviços (Services). Estes serviços representam a funcionalidade de um acessório que contém por sua vez um conjunto de características (Characteristics).

Por exemplo um acessório para controlar o portão da garagem, contém:

1. serviço que disponibiliza informação sobre o acessório;
2. serviço que controla o mecanismo de abrir a garagem;
3. serviço que controla a luz do mecanismo;

Cada um destes serviços apresenta um conjunto de características que podem ser lidas e/ou alteradas. O serviço que controla o mecanismo poderá conter as características:

1. “estado actual” -> “aberto”, “fechado”, “a abrir”, “a fechar”;
2. “estado pretendido” -> “aberto”, “fechado”;
3. “portão trancado” -> “sim”, “não”;
4. “portão obstruído” -> “sim”, “não”

Para além destas funcionalidades, permite outras para cenários mais avançados, tais como: permitir a integração de dispositivos já existentes, usando o conceito de bridges, que fazem a tradução entre protocolos. Permite também criar Zonas, ou grupos de divisões que constituam por exemplo, um segundo andar. Ou ainda a definição de grupos de serviços, por exemplo "luzes noturnas". Por fim a definição de Action sets ou conjuntos de ações que podem ser executadas simultaneamente e até agendadas para executar numa data específica, é também possível. [20]

Devido à necessidade de tornar a framework HomeKit uma solução distribuída existe a intenção de tornar esta plataforma numa solução híbrida facilmente configurável. Usando a atual Apple TV (equipada com Bluetooth e Wi-Fi) como um gateway, os utilizadores podem controlar os seus dispositivos HomeKit em qualquer lugar [21].

2.3.4 Wink

Wink é essencialmente uma aplicação móvel criada inicialmente para controlar dispositivos da iniciativa Quirky+GE (General Electric), no entanto, graças a um Hub criado pela Wink, este sistema permite agora integração com maioria dos dispositivos já existentes no mercado, numa só plataforma. Este Hub foi criado para funcionar com os protocolos de comunicação mais utilizados, e apresenta por isso suporte a ZigBee, Z-wave, Lutron ClearConnect, Kidde, Bluetooth Low Energy e Wi-Fi.

Assim, a Wink oferece uma solução completamente aberta, acessível, simples e de fácil utilização. A aplicação está disponível para iOS e Android.

Em relação á comunidade de developer, a Wink disponibiliza uma API RESTful que permite ligar os dispositivos aos utilizadores e aplicações. A documentação desta API apenas especifica os pedidos que podem ser feitos à API e os recursos que disponibiliza. Não existe informação pública de como integrar um novo dispositivo com API.[22]

2.3.5 Outros projetos

Existem também outros projetos interessantes que se encaixam nesta problemática. Destacam-se os projetos, Open Source Automation, Freedomotic e OpenHAB.

Através de uma arquitetura de plugins sobre uma licença open source, o projeto Open Source Automation (OSA), permite que qualquer um possa controlar dispositivos inteligentes à distância, criando plugins específicos e aplicações cliente capazes de controlar os vários dispositivos configurados. Esta plataforma suporta alguns dos protocolos mais utilizados em automação tais como, Wi-Fi, Z-wave, Bluetooth, Insteon X-10, entre outros. Baseado em .NET, o OSA apenas corre em Windows e não apresenta uma solução para Linux. Através da sua API REST permite que sejam criadas aplicações cliente noutras plataformas, como por exemplo Android.

No projeto Freedomotic, tal como o anterior a sua arquitetura é também baseada em plugins. Este projeto tem como objetivo quebrar as barreiras entre as necessidades humanas, tecnologias de automação, inteligência artificial e aplicações. É uma framework Java pelo que corre em qualquer sistema operativo capaz de correr um JVM. É escalável, pode ser usado para controlar desde apartamentos pequenos até grandes edifícios. Por ser uma arquitetura de plugins OSGi permite adicionar novas funcionalidades sempre que necessário. Permite a criação de interfaces para qualquer dispositivo móvel iOS, Android, WP e Web. É completamente agnóstico em relação ao hardware e apresenta um simulador o que permite que a plataforma corra sem necessidade de nenhum sensor ou atuador ligado.

Por último o projeto Open Automation Bus (openHAB) tem como objetivo disponibilizar uma plataforma de integração universal de todos os dispositivos relacionado com automação residencial. Uma solução Java baseada na arquitetura OSGi. É completamente agnóstico em relação ao hardware e disponibiliza interfaces unificadas para os vários dispositivos móveis, iOS, Android e Web.

Estes projetos, apesar de serem soluções legítimas, apresentam algumas limitações, não só a nível de arquitetura, mas também pela falta de versatilidade das suas plataformas de desenvolvimento e dos interfaces que disponibilizam. Não são, por isso, considerados na secção seguinte.

2.4 Análise Comparativa

Fazendo uma análise comparativa das frameworks acima descritas, é possível verificar que existem diferenças significativas, quer na abordagem em si, quer nas funcionalidades que disponibiliza. A tabela 2.2 apresenta um resumo das principais características de cada abordagem. Começando pelo HomeKit da Apple, é um dos mais completos em termos de arquitetura e segurança. Seguindo uma abordagem protocolar, com o objetivo de integrar tudo numa única solução, requer que os dispositivos sejam desenvolvidos com base num protocolo específico. No entanto é uma framework proprietária pelo que apenas é compatível com dispositivos iOS.

Ao contrário da anterior, a Smartthings apresenta uma solução baseada num hub, isto é, necessita de hardware adicional para comunicar com os dispositivos. Isto permite a integração com outro tipo de produtos já existentes para além de compatibilizar outros protocolos com um smartphone, por exemplo. Já o Nest apenas disponibiliza um serviço que permite interagir com os seus produtos. Pode em alguns casos não ser considerada uma framework. Em todo o caso, a sua arquitetura constitui uma base bastante sólida para a criação de uma framework baseada nos seus conceitos inovadores. Por fim a Wink, apresenta também uma API sob a forma de um serviço RESTful que facilita a comunicação dos dispositivos Wink com os utilizadores, aplicações e a web.

Analisando as características das frameworks e serviços descritos, é possível obter uma visão geral das tecnologias mais usadas nesta problemática da automação residencial. No capítulo seguinte, para além de se apresentarem os requisitos do projeto IVIT, serão também apresentados alguns requisitos adicionais que, com base no que foi estudado neste capítulo, são relevantes para o produto final.

Tabela 2.2: Frameworks e APIs para Automação Residencial

	HomeKit	Smartthings	Nest	Wink
Plataformas	iOS 8 e seguintes	iOS, Android, WP 8.1	iOS, Android, Web, OSX, Windows, Linux, Rails, Django	iOS, Android
Linguagens	Objective-C, SWIFT	Groovy	Java, Objective-C, Javascript	
Descoberta de Dispositivos	Bonjour	SSDP, mDNS/DNS-SD/Bonjour		
Protocolos de Comunicação	Wi-Fi, Bluetooth LE	Wi-Fi, ZigBee, Z-Wave	Wi-Fi	Wi-Fi, Bluetooth LE, ZigBee, Z-wave, Lutron, ClearConnect, Kidde
Acesso a Dados	REST	REST, SOAP	REST, FireBase	REST
Representação de Dados	JSON	JSON	JSON	JSON
Abordagem Distribuída	Opcionalmente	Sim	Sim	Sim
Segurança	End-to-end encryption, Autenticação bi-direcional, Sessões com chaves únicas	OAuth2	OAuth2, SSL	Autenticação de dois passos, SSL
Integração com outros serviços e/ou protocolos	Sim	Sim	Sim, apenas para integração o serviço Nest	Sim
Permite novos dispositivos?	Sim, dentro do programa MFi	Sim	Não	Sim, formando uma parceria
Simulador	Sim	Sim	Sim	Não
App Cliente Própria	Não	Sim	Sim	Sim
Hub	Não	Sim	Não	Sim
Controlo por Voz	Siri	Google Now	Não	Não

Capítulo 3

Levantamento de Requisitos

Neste capítulo são apresentados os requisitos do projeto IVIT, bem como alguns requisitos de desenvolvimento adicionais, que com base no que foi estudado no capítulo anterior constituem uma mais-valia para o projeto e para o produto final.

3.1 Requisitos do Projeto IVIT

Tal como referido no capítulo primeiro, os objetivos principais deste projeto passam pelo desenvolvimento de um protocolo de comunicação com o reservatório e desenho e implementação de uma aplicação móvel para monitorização em controlo do sistema.

Sendo este um dispositivo que se encontra normalmente instalado numa divisão à parte, deverá ser adoptado um protocolo de comunicação com um alcance médio ou longo, e se for o caso que suporte uma ligação por cabo.

A criação de uma aplicação cliente, intuitiva, que permita monitorizar e controlar o reservatório deverá apresentar apenas informação considerada útil, ou seja, não deverá conter demasiados dados técnicos. Por exemplo, sendo o dispositivo em causa, um reservatório de águas quentes, uma informação considerada útil será o tempo restante de água quente ou simplesmente a temperatura atual.

Tal como referido anteriormente na definição de objetivos, os requisitos do projeto IVIT são definidos pelas atividades inerentes a este trabalho (Atividade 9, Atividade 10 e Atividade 11). As atividades podem ser consultadas no Anexo 1.

3.2 Requisitos Adicionais

Após o estudo do estado da arte, é possível definir alguns requisitos que se consideram importantes para a solução e que devem ser tidos em conta no desenvolvimento deste projeto.

Devido à popularização dos dispositivos e aplicações móveis, a aplicação cliente deverá também seguir esta tendência adotando o uso de dispositivos móveis facilmente acessíveis à maioria dos utilizadores, isto é, os equipamentos mais populares iOS ou Android.

Seguidamente, e por se pretender usar uma infraestrutura de rede existente, a comunicação deverá ser feita usando uma das tecnologias compatíveis com um smartphone, Wi-Fi ou Bluetooth. Este último, como já foi referido anteriormente, apresenta algumas limitações de proximidade, logo deverá ser adotado o protocolo IP através de Wi-Fi.

Facilitar e minimizar o esforço de configuração deste tipo de sistemas por parte dos utilizadores é outro dos requisitos adotado pela maioria das soluções já estudadas. Deverá ser possível reconhecer o reservatório na rede e efetuar a sua configuração automaticamente usando tecnologias como o Zero-configuration, por exemplo.

A monitorização do sistema deverá ser feita em tempo real. Neste caso, pode-se aplicar o termo *soft real-time*, ou seja, para sistemas de tempo real classificados pela consequência da falha de um prazo ou medição, é tolerável o desvio de alguns segundos ou até mesmo a falha de uma das medições. Não comprometendo o sistema, a utilidade de um resultado, degrada após uma falha, degradando apenas a qualidade de serviço do sistema [23].

Por fim, e uma vez que a compatibilização e centralização dos dispositivos de automação residencial tem cada vez mais um impacto na sua usabilidade, deverá ser feito um esforço para a integração da solução encontrada numa das frameworks estudadas anteriormente.

Em resumo, os requisitos de projeto são:

- Promover o uso de dispositivos móveis (iOS e/ou Android);

- Comunicação por Wi-Fi;
- Reconhecimento automático do reservatório na rede;
- Monitorização em tempo real;
- Integração com uma das Frameworks estudadas;

Capítulo 4

Solução Proposta

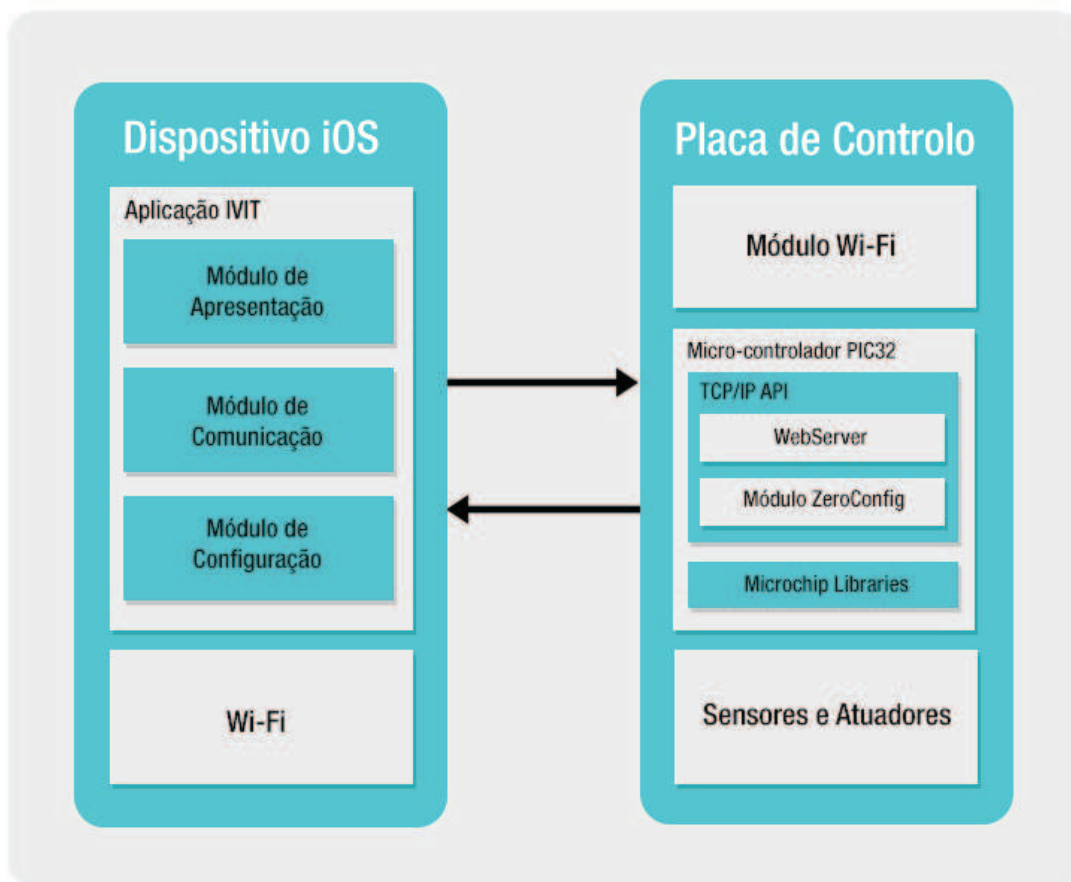
Este capítulo começa por apresentar um diagrama com a arquitetura proposta para monitorização e controlo do reservatório. Seguidamente descrevem-se um conjunto de tecnologias escolhidas para arquitetura proposta. No final, será apresentada uma visão geral do sistema, incorporando todas as tecnologias escolhidas.

4.1 Arquitetura do Sistema proposto

Conceptualmente a arquitetura deste sistema não se prevê ser complexa, tanto que sendo esta uma solução relativamente barata, pretende-se que seja simples, segura e eficaz. Com uma arquitetura baseada na comunicação sem fio, este sistema é composto, essencialmente, por uma aplicação móvel e um simples web server com funcionalidades limitadas. O protocolo de comunicação escolhido para este sistema é o protocolo IP através de uma ligação sem fios Wi-Fi. Sendo esta uma tecnologia de comunicação bastante comum, é possível fazer uso de uma infraestrutura de rede já existente.

Em relação ao módulo web server, este será disponibilizado pela placa de controlo, desenvolvida para este projeto, que controla as funções associadas ao reservatório, e que inclui, entre outros componentes, um micro-controlador PIC32, responsável pelo processamento de dados e um módulo Wi-Fi. Este módulo, associado à pilha protocolar TCP/IP permite que qualquer outro dispositivo, ligado na mesma rede, possa comunicar com o serviço através das várias camadas aplicacionais.

Figura 4.1: Arquitetura do Sistema de Controlo Remoto



A aplicação móvel, que acaba por ser o ponto fulcral do projeto, tem como objetivo, apresentar ao utilizador os dados de monitorização que são recolhidos, em tempo real, pelos sensores da placa de controlo. Esta aplicação está idealizada para smartphones, mais concretamente para equipamentos com o sistema operativo iOS 8 e seguintes. Este sistema operativo móvel, proporciona uma grande facilidade no desenvolvimento de aplicações, uma vez que disponibiliza um conjunto de ferramentas úteis.

Esta aplicação está dividida num conjunto de módulos: um módulo de configuração baseado em Zero-configuration/Bonjour, que permite efetuar uma ligação entre dois dispositivos de forma fácil; um módulo de comunicação para comunicar com o web server, permitindo efetuar pedidos HTTP, e sendo também responsável pelo decoding de conteúdos JSON devolvidos pelo web server; um módulo de apresentação que mostra os dados, devidamente tratados, ao utilizador.

A figura 4.1 permite ilustrar melhor a arquitetura descrita. Apresentam-se de seguida um conjunto de tecnologias, algumas delas já mencionadas, que se enquadram

na arquitetura proposta e que foram escolhidas para o desenvolvimento e prova de conceito.

4.2 Tecnologias

4.2.1 Zero-configuration

A capacidade de identificar unicamente um dispositivo numa rede, atribuindo-lhe um endereço, neste caso um endereço IP, é um dos principais problemas da comunicação em rede. Tipicamente, nas redes modernas, estes dispositivos são identificados através de um servidor com o protocolo DHCP (Dynamic Host Configuration Protocol), que lhes atribui um endereço automaticamente.

Outro dos problemas prende-se pela resolução de nomes, ou seja, apesar de ser atribuído um endereço IP a um dispositivo, este nem sempre é facilmente identificável por um humano. Para resolver este problema passaram a ser usados servidores de DNS (Domain Name Service) que permitem a associação de nomes legíveis aos endereços IP. A resolução de nomes requer que o endereço do servidor DNS seja conhecido. Embora esta seja uma solução conhecida para a identificação de dispositivos na rede, nem sempre é possível e/ou conveniente a utilização deste tipo de servidores.

Assim surgiu o conceito de Zero-configuration networking que não é mais que um conjunto de tecnologias que permitem criar uma rede funcional quando dispositivos de rede são interligados sem a necessidade de configurações manuais ou de servidores. O Zero-configuration é composto por tecnologias que permitem: atribuir endereços IP, automaticamente, aos dispositivos integrantes da rede, atribuir a si mesmos nomes legíveis e facilmente compreendidos, e identificação de serviços de rede. O Multicast DNS (mDNS) permite que a rede local aja como um servidor DNS permitindo o uso de nomes facilmente compreendidos para aceder a dispositivos na rede. Caso os dispositivos escolham usar o mesmo nome, cada um destes irá automaticamente negociar novos nomes. Apesar de atribuir nomes, o mDNS não disponibiliza informação sobre o tipo de dispositivo ou sobre o seu estado. Para isso existem os protocolos de descoberta de serviços (SDP) que permitem aos dispositivos anunciar que serviços disponibilizam, por exemplo uma impressora pode anunciar que tem serviços de impressão disponíveis [24]. Existem várias implementações para a descoberta de serviços, destacando-se o mDNS/DNS-SD, a abordagem da Apple e o UPnP/SSDP, a abordagem da Microsoft.

Do conjunto das tecnologias inerentes ao zero-configuration, resultaram algumas implementações específicas, das quais se deve destacar a implementação da Apple denominada por Bonjour.

4.2.2 REST

REST (Representational State Transfer) é um estilo arquitetural usado em aplicações de rede ou serviços web que permite que quaisquer dispositivos ligados numa rede possam comunicar entre si, na maioria dos casos, através do protocolo de comunicação HTTP.

As aplicações que usam REST são chamadas de RESTful e fazem uso do protocolo HTTP para enviar, ler e apagar dados, ou seja para efetuar operações CRUD (Create, Read, Update e Delete). Tal como os browsers usam HTTP verbs (GET, POST, PUT, DELETE) para mostrar páginas web ou enviar dados para um servidor, o mesmo acontece com as aplicações RESTful.

Podemos por exemplo afirmar que a própria World Wide Web pode ser vista como uma arquitetura baseada em REST.

Do ponto de vista da programação, REST é uma alternativa mais leve e mais rápida que os tradicionais Webservices, SOAP, WSDL, etc ou RPC. Apresenta um conjunto de restrições arquiteturais que, quando aplicadas como um todo, destacam a escalabilidade na interação de componentes, simplicidade das interfaces, implementação independente de componentes e componentes intermediários que reduzem a latência, reforçam a segurança, e encapsulam sistemas legados [25].

4.2.3 JSON

JSON (JavaScript Object Notation) é um formato leve de representação e troca de dados que é facilmente compreendido por humanos e facilmente gerado e interpretado por máquinas. JSON é um formato de texto que é completamente independente de linguagem de programação. É usado principalmente para a troca de dados entre um servidor e uma aplicação e considera-se como uma alternativa ao formato XML. JSON é constituído por duas estruturas: uma coleção de pares chave/valor, normalmente designado pelas linguagens de programação como um objeto, e por listas ordenadas de

valores, normalmente caracterizadas por um array.

Estas estrutura de dados são universais e são suportadas, virtualmente, por todas as linguagens de programação modernas, tornando se assim um formato bastante utilizado para representação de dados [26].

4.2.4 HTTPS

O HTTPS (HyperText Transfer Protocol Secure) é um protocolo para comunicação segura sobre uma rede informática, que resulta da aplicação do protocolo SSL/TLS sobre o HTTP, adicionando assim as capacidades de segurança do SSL/TLS à comunicação HTTP.

O SSL/TLS usa um conjunto de chaves públicas e privadas para a troca de chaves simétricas temporárias que permitem cifrar o fluxo de dados, incluindo parâmetros aplicativos, cabeçalhos, cookies e outros, garantindo confidencialidade e integridade ao HTTPS. Para além disso usando certificados digitais e a norma X.509, podemos garantir a autenticidade do servidor/serviço protegendo contra ataques do tipo man-in-the-middle.

Dado que o HTTPS fornece uma série de funcionalidades de segurança, a sua utilização é fortemente aconselhada, no entanto não é vital ao funcionamento de uma aplicação.

4.2.5 iOS

O iOS [27] é um sistema operativo, desenvolvido pela Apple, que surgiu com o lançamento do primeiro iPhone em 2007, estendendo-se agora aos iPads, iPod Touch e ainda Apple TV. Este sistema operativo é proprietário, ou seja, só funciona em equipamentos que sejam desenvolvidos pela própria Apple.

Desenvolvido para interfaces touchscreen que permitem multi-toque, o iOS é um dos mais populares sistemas operativos para smartphones existentes no mercado.

Apesar de a linguagem recomendada pela Apple seja o novo SWIFT, a utilizada neste projeto é o Objective-C. Esta linguagem é orientada a objetos e baseada em C,

adicionando os conceitos da linguagem Smalltalk.

Por fim, o IDE escolhido para o desenvolvimento do projeto foi o Xcode. Esta plataforma foi criada e desenvolvida pela Apple para criação de aplicações para as suas plataformas: iOS e OS X. Uma das grandes vantagens deste IDE é disponibilizar um grande conjunto de ferramentas que facilitam o desenvolvimento de alguns componentes, como o grafismo da aplicação. Para complementar tem ainda um vasto conjunto de frameworks integradas.

4.2.6 Microchip TPC/IP Stack Web Server

A família de micro-controladores da Microchip associada a um módulo Wi-Fi, disponibiliza funcionalidades de um web service através da pilha protocolar TCP/IP da Microchip e de um Web Server limitado, que permite que, qualquer outro dispositivo ligado na mesma rede possa comunicar com o serviço.

Esta pilha protocolar TCP/IP disponibiliza uma base para aplicações de rede embebidas, gerindo a maioria das interações necessárias entre a camada física e a camada aplicacional. Inclui diferentes módulos para as camadas aplicacionais mais utilizadas tais como HTTP, para páginas web, SMTP para envio de e-mails ou Zero-configuration para configuração e descoberta de dispositivos [24].

O módulo HTTP apresenta várias capacidades, algumas delas genéricas a qualquer servidor web, outras específicas de aplicações embebidas deste tipo.

Variáveis dinâmicas

Uma das necessidades mais básicas é a de fornecer aos utilizadores das aplicações cliente informações de estado. O servidor HTTP disponibiliza esta funcionalidade através de callbacks de substituição de variáveis dinâmicas. Estes comandos quando presentes no código, alertam o servidor para a execução de uma função que faz com que os dados das variáveis sejam substituídos pelo seu valor de output. As variáveis dinâmicas estão identificadas por `~variável(parametros)~`. Por exemplo num ficheiro JSON com a estrutura presente em 4.1 origina o output presente em 4.2:

Listing 4.1: raw JSON

```
{
"led0" : "~led(0)~",
"led1" : "~led(1)~",
...
"btn1" : "~btn(1)~",
"btn2" : "~btn(2)~",
...
"temp0" : "~temp(0)~",
"temp1" : "~temp(1)~"
}
```

Listing 4.2: output JSON

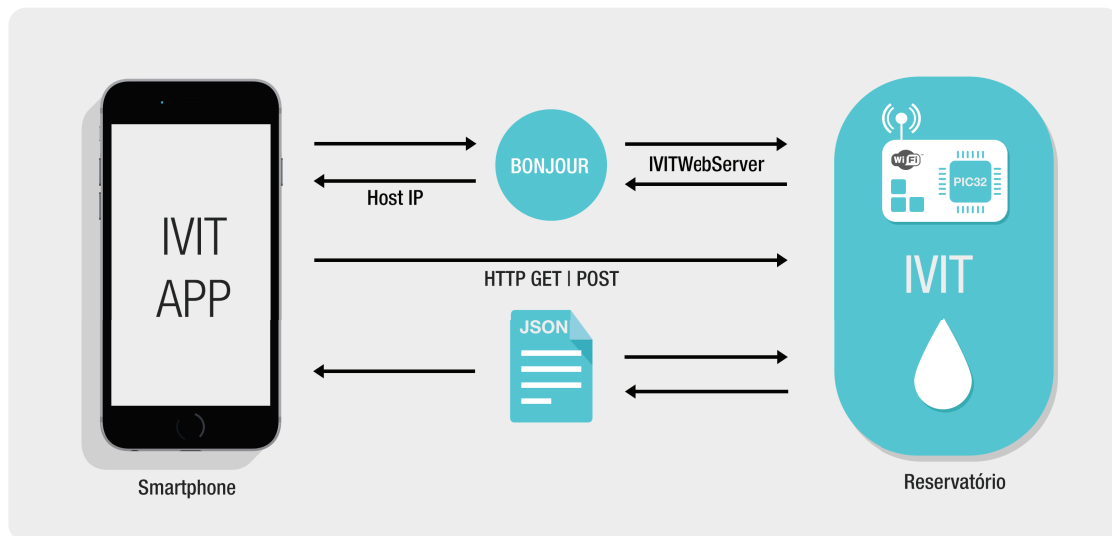
```
{
"led0" : "1",
"led1" : "1",
...
"btn1" : "up",
"btn2" : "down",
...
"temp0" : "25C",
"temp1" : "20C"
}
```

Método GET

O protocolo HTTP tem como objetivo possibilitar a comunicação cliente-servidor. O HTTP funciona como um protocolo de pedido-resposta entre um cliente e servidor. Um browser será o cliente, e a uma aplicação num computador que aloja um website será o servidor. O cliente submete um pedido HTTP ao servidor, que devolve uma resposta ao cliente.

O método GET, permite fazer pedidos ao recurso especificado. Este método acrescenta os dados no final do URI. (ex: `http://servidor/form.html?led1=1&led2=0`) Os dados são enviados via GET e guardados no array `curHTTP.data`. Uma vez que fica guardado em memória, o seu tamanho está limitado, por defeito, a 100 bytes. Ao executar este pedido a função `"HTTPExecuteGet"` é usada para processar os dados e

Figura 4.2: Visão Geral do Sistema de Controlo Remoto



executar as ações necessárias.

Método POST

O método POST permite submeter dados para serem processados num recurso específico. Os dados enviados não são visíveis no endereço. No entanto, usa o mesmo método de encoding de URL. Todos os dados POST ficam no buffer TCP, por isso a aplicação terá de aceder diretamente ao buffer TCP.

O capítulo seguinte apresenta os componentes envolvidos na programação da placa de controlo.

4.3 Visão Geral do Sistema Proposto

Após descrever as principais tecnologias que constituem a solução encontrada, é necessário perceber como cada uma delas se aplica na implementação de uma prova de conceito. A figura 4.2 apresenta uma visão geral do sistema proposto e de como os vários componentes interagem entre si. Podemos também perceber a partir da figura o funcionamento ou *workflow* do sistema.

Após configurar os dispositivos na mesma rede, a comunicação é iniciada pela apli-

cação móvel. Através desta aplicação e usando a tecnologia zero-configuration/Bonjour é possível reconhecer o reservatório na rede e efetuar a ligação automaticamente em poucos passos. De seguida a aplicação fica pronta a comunicar com o reservatório usando pedidos HTTP GET ou POST. O reservatório simplesmente responde com um ficheiro JSON que contém o estado atual de cada um dos sensores e atuadores. Novamente, a aplicação móvel será responsável por tratar esses dados, apresentando ao utilizador apenas a informação relevante, devidamente tratada.

Capítulo 5

Implementação

Este capítulo descreve todo o processo de implementação e prova de conceito inerente à solução proposta, desde as características da placa de controlo até às funcionalidades de cada módulo da aplicação móvel, capaz de monitorizar e controlar em tempo real alterações de estado.

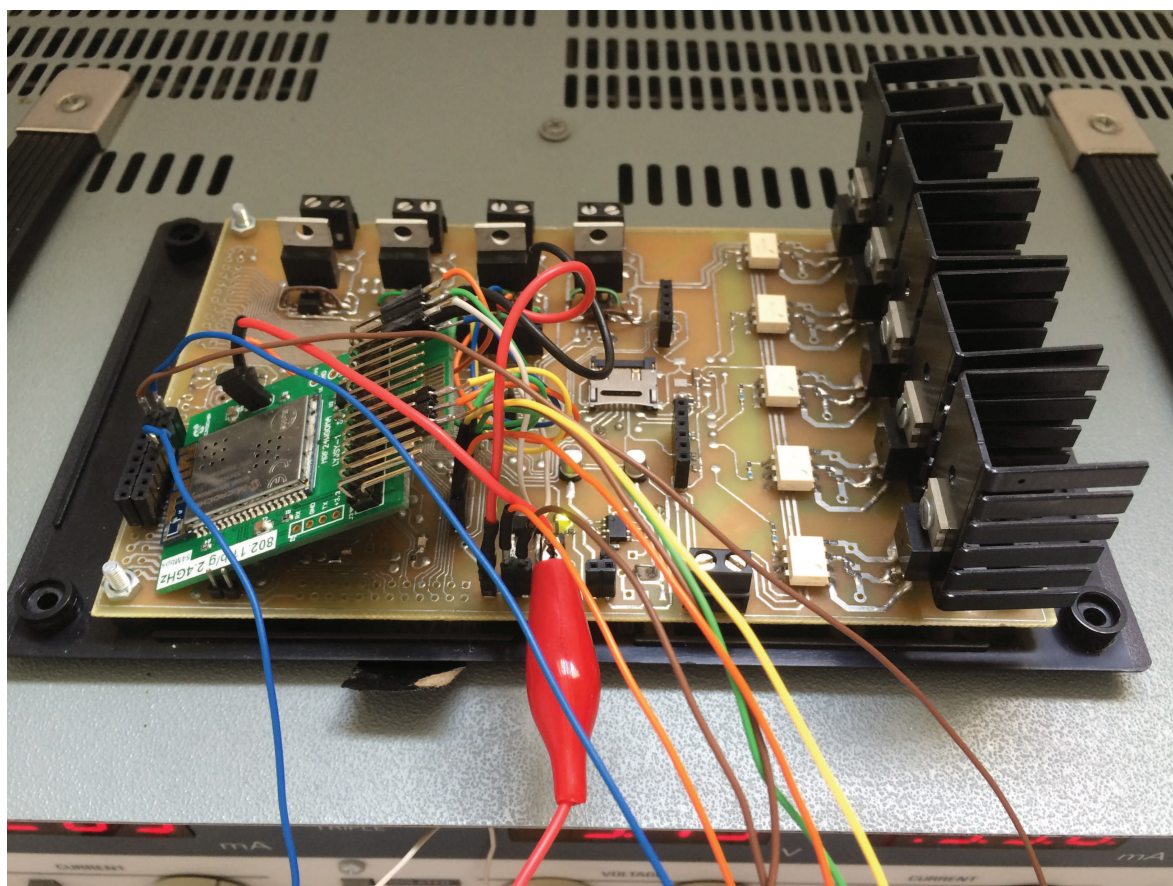
5.1 Placa de Controlo

Para compreender melhor o funcionamento do sistema, segue-se a descrição dos principais componentes usados na placa de controlo, que são importantes para se entenderem algumas limitações do sistema. São também descritas algumas das configurações efetuadas. Na figura 5.1 podemos observar a placa de controlo desenhada para este projeto.

5.1.1 Micro-controlador/Microprocessador

O micro-controlador utilizado pertence à família de micro-controladores de baixo custo da Microchip, ideais para aplicações embebidas. O modelo utilizado para este projeto é o PIC32MX795F512L. Este modelo é um micro-controlador PIC de 32-bit com uma frequência de relógio de 80 MHz, 128K de memória RAM e 512K de memória Flash. Destes 512k, apenas 120k estão reservados para o web server e respetivos ficheiros de apoio. No entanto, a placa de controlo apresenta um slot de expansão de memória para cartões SD.

Figura 5.1: Placa com módulo Wi-Fi



5.1.2 Wi-Fi Transceiver

Para a comunicação de rede sem fios, foi utilizado o módulo transceiver Wi-Fi IEEE 802.11 b/g, MRF24WG0MA, facilmente integrado com micro-controladores PIC e ideal para soluções de redes sensoriais Wi-Fi de baixo consumo, automação residencial e aplicações para o consumidor. Este módulo tem a antena completamente integrada na PCB e permite velocidades de transferência de dados até 54 Mbps.

5.1.3 Configurações

Sendo este um projeto com várias componentes, parte das configurações efetuadas foram feitas em conjunto com o departamento encarregue do desenvolvimento de hardware. As configurações dos módulos aplicativos são feitas num único ficheiro .h que se encontra em Anexo no código fonte do servidor na pasta TCPIP/demo_app_ivit_io/Configs com o nome "TCPIP MRF24W PIC32_SK.h". Já as configurações do módulo Wi-fi, são feitas no ficheiro "WF-Config.h" da pasta TCPIP/demo_app_ivit_io. Para a apli-

cação em causa, destacam-se os módulos HTTP2 Server para o webserver e ZeroConf para descoberta de dispositivos.

Por fim, foram aplicadas algumas configurações para comunicação segura, ativando o módulo SSL. Este módulo, na sua versão atual usa o protocolo SSLv3, considerado obsoleto em junho de 2015 pelo RFC 7568. Esta situação levou a que este protocolo não pudesse ser aplicado na aplicação desenvolvida.

5.1.4 Web Server

O web server utilizado, é disponibilizado pelo módulo HTTP2 web server da API TCP/IP da Microchip. Para entender como funciona este web server, é necessário descrever três componentes principais envolvidos no processo: as páginas web, o utilitário MPFS2 e os ficheiros CustomHTTPApp.c e HTTPPrint.h.

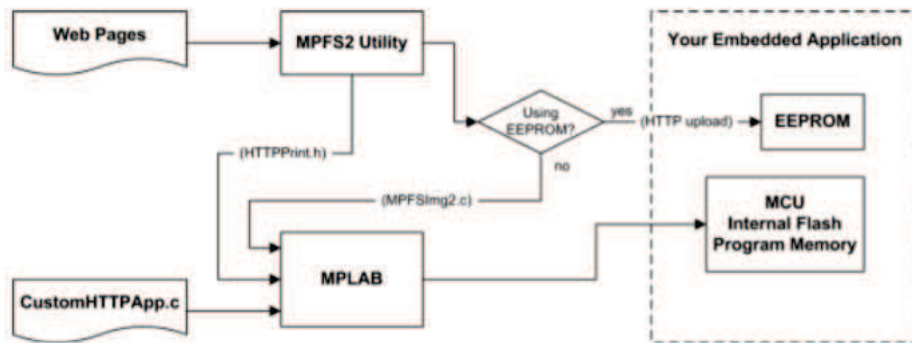
As páginas web incluem todo o HTML e imagens, folhas de estilo CSS, ficheiros Javascript e outros ficheiros auxiliares, necessários para a apresentação do website. O módulo apresenta uma aplicação demonstrativa, que serviu de base para este projeto. Esta aplicação consiste num conjunto de páginas web que permitem ver o estado dos vários sensores e efetuar configurações básicas.

O utilitário MPFS2 é disponibilizado pela Microchip e permite compactar e fazer o upload das páginas web, num formato que possa ser guardado mais eficientemente, tanto na memória flash interna como em armazenamento externo. Esta ferramenta também indexa as variáveis dinâmicas presentes nas páginas web, atualizando o ficheiro HTTPPrint.h com esses índices. Ao adicionar ou remover variáveis dinâmicas é necessário voltar a compilar o projeto no MPLAB IDE.

O ficheiro CustomHTTPApp.c, implementa a aplicação web. Descreve o output das variáveis dinâmicas, efetua o parsing dos dados submetidos através dos métodos GET e/ou POST e valida credenciais de acesso. Por fim, o ficheiro HTTPPrint.h é gerado automaticamente pelo utilitário MPFS2 e, tal como referido, indexa todas as variáveis dinâmicas [24]. Este processo encontra-se ilustrado pela figura 5.2.

A monitorização em tempo real, é um dos pontos principais deste projeto. Para que isso possa ser possível através de aplicações externas foi necessário criar um ficheiro de controlo, ou modelo de dados para guardar estas informações vitais. Este modelo

Figura 5.2: Processo de Compilação do web server



de dados usa um documento estruturado do tipo JSON, que será descrito com mais detalhe na secção seguinte. Este ficheiro, denominado por "status.json" encontra-se na pasta TCPIP/demo_app_ivit_io/WebPages2.

5.2 Aplicação móvel

A aplicação móvel desenvolvida tem como objetivo provar o funcionamento de todos os elementos descritos na solução proposta. Esta aplicação foi desenvolvida e testada para a plataforma iOS na versão 8.0 e superiores. A escolha desta plataforma prende-se no facto de na versão 8.0 ter sido introduzida framework Homekit, descrita anteriormente e que assim se pudessem efetuar testes à mesma. A aplicação permite monitorizar em tempo real todas as leituras dos sensores e atuadores presentes na placa de controlo desenvolvida para este projeto.

5.2.1 Modelo de dados

Devido às limitações de hardware e software da placa de controlo desenvolvida para este projeto, o modelo de dados utilizado é uma solução NoSQL, pelo que os dados são guardados num documento estruturado do tipo JSON. Não existe nenhuma estrutura pré-definida para este modelo, segue apenas uma ordem lógica e as regras JSON. Como podemos verificar o modelo utilizado está dividido pelos diferentes tipo de sensores e atuadores presentes no módulo de controlo:

- "led(id)": representa o estado dos LEDs do módulo de controlo. "0" para desligado e "1" para ligado.

- "btn(id)": representa o estado dos botões presentes no módulo de controlo. "down" para premido e "up" para não premido.
- "temp(id)": representa a medição dos vários sensores de temperatura. Medido em graus °C.
- "dcout(id)": representa o estado de saída de potência DC. "ON" para ligado e "OFF" para desligado.
- "acout(id)": representa o estado de saída de potência AC. "ON" para ligado e "OFF" para desligado.

Listing 5.1: Modelo de dados JSON

```

{
    "led0" : "~led(0)~",
    "led1" : "~led(1)~",
    ...
    "btn1" : "~btn(1)~",
    "btn2" : "~btn(2)~",
    ...
    "temp0" : "~temp(0)~",
    "temp1" : "~temp(1)~",
    "temp2" : "~temp(2)~",
    ...
    "dcout0" : "~dcout(0)~",
    "dcout1" : "~dcout(1)~",
    "dcout2" : "~dcout(2)~",
    ...
    "acout0" : "~acout(0)~",
    "acout1" : "~acout(1)~",
    "acout2" : "~acout(2)~"
}

```

O Anexo 2, apresenta um excerto de modelo com uma estrutura mais complexa. Os ficheiros completos podem ser consultados no formato digital. Este modelo foi usado na fase de testes, para verificar o comportamento da aplicação.

5.2.2 Módulo de Configuração e descoberta de dispositivos

De modo a facilitar a configuração e ligação à placa de controlo, utilizou-se o conceito de zero-configuration networking, descrito no capítulo anterior. Esta tecnologia permite criar uma rede funcional quando os dispositivos de rede são interligados sem necessidade de configurações manuais. Fornece também uma convenção de nomes mais familiar em vez de usar apenas o IP.

Após ativar este módulo na placa de controlo, todas as configurações iniciais foram mantidas, apenas o nome foi alterado. O servidor é anunciado como um serviço do tipo `http` com o nome de `IVIT-WebServer`.

Sendo esta uma aplicação em iOS, a implementação desta tecnologia por parte da Apple é ligeiramente diferente. Denominada por Bonjour, permite a publicação e descoberta de serviços TCP/IP numa rede local o que facilita a configuração de rede para os utilizadores.

O iOS disponibiliza vários conjuntos de APIs para aplicações que usem Bonjour. Para esta aplicação foram utilizadas as classes `NSNetService` e `NSNetServiceBrowser` presentes na Foundation Framework. Estas classes disponibilizam abstrações orientadas a objetos para descoberta e publicação de serviços.

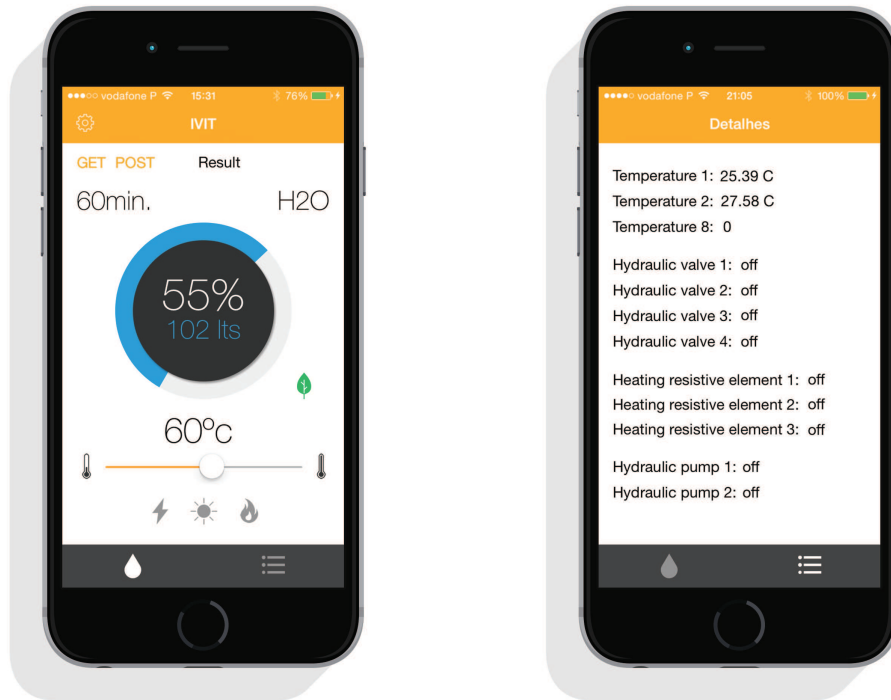
Os objetos do tipo `NSNetService` representam instâncias de serviços Bonjour, tanto para publicação como para descoberta, por parte de um cliente. `NSNetServiceBrowser` representa um browser para um tipo de serviço particular. Os serviços Bonjour são denominados de acordo com o Internet standard para serviços IP existente (Descritos no RFC 2782) [28]. Neste caso o tipo de serviço será `_http._tcp`.

Em termos de apresentação, a aplicação disponibiliza uma vista com opções e configurações. Antes de realizar esta configuração, devemos ter a certeza que estamos ligados na mesma rede que o módulo de controlo. Nesta vista temos a possibilidade de procurar os serviços disponíveis e ligando-nos a um deles apenas com um toque.

Internamente e após a descoberta dos serviços, a aplicação irá proceder à resolução de nomes do serviço, ou seja usar a informação da instância do serviço para obter os endereços IP, portos e host names.

Após esta configuração estamos prontos a usar a aplicação com todas as suas fun-

Figura 5.3: Vistas principais da aplicação móvel



cionalidades. Esta configuração é apenas necessária executar uma vez.

5.2.3 Módulo de Comunicação e sincronização de dados

No seguimento do que foi mencionado no módulo de configuração, a aplicação já se encontra pronta a comunicar com o web server. Toda a informação necessária ao seu funcionamento estará guardada num documento estruturado do tipo JSON, atualizado em tempo real, pela placa de controlo, usando o conceito de variáveis dinâmicas, descrito anteriormente.

Este módulo de comunicação tem como objetivo recolher, periodicamente, toda a informação acerca do estado atual do sistema e prepará-la para ser mostrada ao utilizador. Usando as classes `NSURLRequest` e `NSURLConnection` é possível efetuar pedidos HTTP assíncronos que devolvem os conteúdos do URL. O próximo passo será converter os conteúdos obtidos para um objeto JSON usando a classe `NSJSONSerialization`. Por fim é apenas necessário efetuar o decoding deste objeto e retirar os dados que interessam ser apresentados ao utilizador. Este processo será repetido periodicamente, e neste caso, repete-se a cada 5 segundos. Este tempo foi o que se entendeu ser razoável para esta aplicação.

5.2.4 Módulo de Apresentação

O módulo de apresentação representa aquilo que o utilizador pode ver na aplicação. Como podemos observar na figura 5.3, são apresentadas duas vistas principais: uma vista com um resumo do estado atual do sistema, com os dados devidamente tratados; e uma vista com a leitura atual de cada um dos sensores e atuadores da placa de controlo. Na primeira vista, são apenas apresentadas as informações que se consideram ser úteis para o utilizador, tais como, a autonomia e capacidade atual de água quente, a temperatura atual da água e quais as fontes de calor a serem usadas no momento. Para efeitos de teste, criou-se uma segunda vista que apresenta as leituras de cada um dos sensores, neste caso, apenas dois sensores de temperatura estavam a ser usados. Para além destas duas vistas, existe uma terceira que permite aceder às configurações e onde podemos procurar e configurar a ligação o serviço disponibilizado pela placa de controlo.

Em Anexo, formato digital, encontra-se um vídeo com a demonstração da aplicação em funcionamento. Este vídeo regista a monitorização em tempo real dos valores obtidos por três sensores de temperatura.

Capítulo 6

Testes

Este capítulo descreve um conjunto de testes efetuados que validam a solução proposta. Depois de caracterizar os dispositivos de teste utilizados, serão apresentados os resultados de testes de conectividade e performance, nos vários dispositivos. Será descrita uma experiência preliminar que não obteve resultados satisfatórios, e uma experiência final que compara o desempenho de cada um dos dispositivos utilizados.

6.1 Caracterização dos dispositivos de teste

Na comunicação de redes sem fios, os dispositivos utilizados podem condicionar o desempenho da rede, e por isso torna-se necessário caracterizar os equipamentos usados. Neste projeto, e de modo a facilitar o desenvolvimento da aplicação, a maioria dos testes foram executados no simulador iOS disponibilizado pelo XCode. O simulador assume as condições de rede presentes no computador usado para fins de desenvolvimento. No entanto foram também efetuadas comparações com um dispositivo real.

O computador utilizado é um Apple MacBook Pro de final de 2013 com o sistema operativo OSX 10.10 Yosemite, equipado com um processador Intel Core i5 2,4 GHz, 8 GB de memória RAM e disco SSD. Mais importante ainda, apresenta um adaptador de rede sem fios AirPort Extreme compatível com as normas IEEE 802.11 a/b/g/n/ac. Em relação ao dispositivo real, utilizou-se um iPhone 5s com iOS 9.0, equipado com um processador Apple A7 1,3 GHz, 1 GB de memória RAM e um adaptador de rede sem fios compatível com as normas IEEE 802.11 a/b/g/n.

As características do dispositivo servidor, desenvolvido para este projeto, encontram-se descritas na secção 5.1.

6.2 Experiência preliminar

Sendo esta uma aplicação altamente dependente da sua ligação à rede, é imperativo garantir o bom funcionamento da aplicação perante todas as condições de rede.

Para esta experiência foram realizados testes de conectividade e performance, simulando diferentes condições de rede. Usando a ferramenta Network Link Conditioner, instalada opcionalmente no XCode, é possível simular várias condições de rede. O perfil utilizado, para uma rede com perdas, simula uma ligação com uma largura de banda de 1mbps, um atraso de cerca de 500ms e uma taxa de perda de pacotes de 10 %. Ao efetuar um simples teste de ping, foi fácil perceber o impacto de uma ligação deste tipo. O equipamento de rede utilizado é um Linksys WRT54GC Compact Wireless-G BroadBand Router compatível com as normas IEEE 802.11 a/b/g.

Assim, efetuaram-se uma série de testes que permitem avaliar o desempenho da aplicação perante as várias condições de rede, tipos de ligação (segura e não segura) e volume de dados. Utilizando ficheiros JSON de diferentes volumes, perante perfis de condições normais de rede (NORMAL), más condições de rede (LOSS) e comunicação segura (HTTPS), foi possível obter a comparação da média e mediana dos tempos de resposta.

Analisando os resultados, observaram-se algumas flutuações nos tempos de resposta, em todos os perfis de rede testados. Os gráficos gerados não apresentavam uma progressão que se conseguisse explicar.

Por isso, e por não se considerarem satisfatórios, os resultados dos testes acima descritos, optou-se pela realização de um conjunto de novos testes em ambiente controlado, desta vez numa rede local. O equipamento utilizado inicialmente estaria a ser utilizado por mais utilizadores em simultâneo, pelo que não se poderia prever, à partida, qual a sua influência nos testes efetuados.

A secção seguinte descreve as novas condições de rede criadas, bem com os dispositivos utilizados e os resultados obtidos.

6.3 Testes de conectividade

Os testes realizados nesta secção, para além de reafirmarem a influência das condições de rede, têm também como objetivo verificar o comportamento da aplicação perante condições de rede normais utilizando vários volumes de dados em diferentes dispositivos, permitindo assim tirar conclusões relativamente à influência das características dos dispositivos utilizados.

Para a realização destes testes foram criadas novas condições de rede. Para evitar qualquer influência, que uma ligação ao exterior, possa provocar, decidiu-se que os testes deveriam ser realizados numa rede local. Para isso, utilizou-se um router wireless que distribui o sinal aos equipamentos ligados à sua rede. O modelo do equipamento utilizado é um TP-Link TL-MR3020, Wireless N Router, compatível com a norma IEEE 802.11n, com velocidades até 150 Mbps.

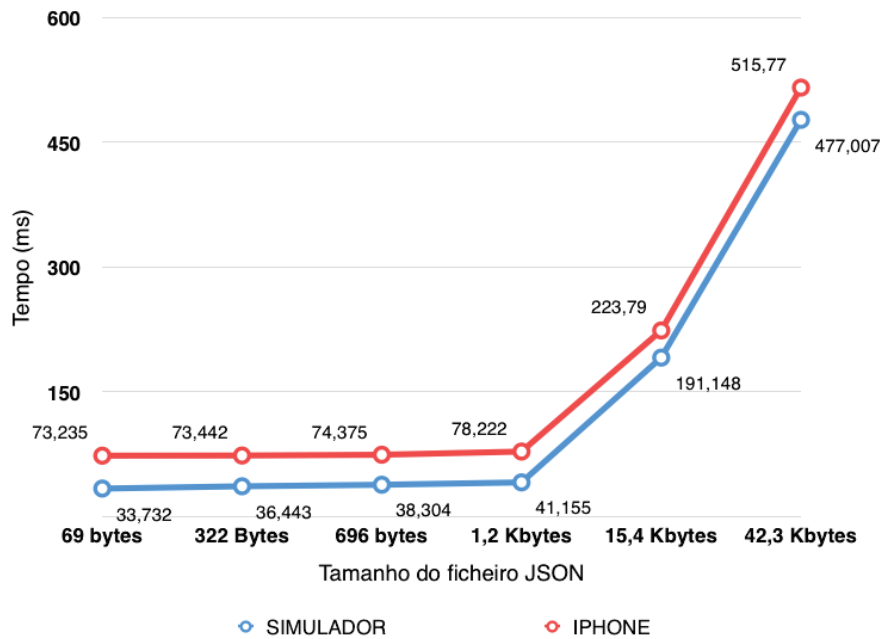
Os dispositivos de teste utilizados mantiveram-se inalterados, pelo que se encontram descritos na secção 6.1. Todos os dispositivos de teste estão configurados para aceder à mesma rede. Esta rede está protegida por WPA/WPA2, permitindo que apenas dispositivos autorizados possam aceder à mesma.

Para fins de análise, há que ter em conta a instabilidade da rede, pelo que as métricas estatísticas utilizadas podem não ser representativas. Apesar de média e mediana serem muito semelhantes, o valor obtido é diferente quando existem valores extremos. Em estatística, este fenómeno denomina-se por obliquidade. Para salvaguardar esta situação, típica de flutuações na rede, optou-se por usar a mediana em vez da média.

A mediana é o valor numérico que separa a metade superior de uma amostra de dados, população ou distribuição de probabilidade, em conjunto ordenado de forma crescente ou decrescente, a partir da metade inferior. A mediana de uma lista finita de números pode ser encontrada ordenando todas as observações do valor mais baixo para o valor mais elevado identificando o valor do meio. Para conjuntos de dados ordenados de amostras de tamanho n , se n for ímpar, a mediana será o elemento central. Se n for par, a mediana será o resultado da média simples entre os dois elementos centrais.

$$\text{Se } n \text{ for ímpar, } \frac{(n+1)}{2} \quad \text{Se } n \text{ for par } \frac{\frac{n}{2} + (\frac{n}{2} + 1)}{2} \quad (6.1)$$

Figura 6.1: Comparação do tempo de resposta entre dispositivo real e simulador



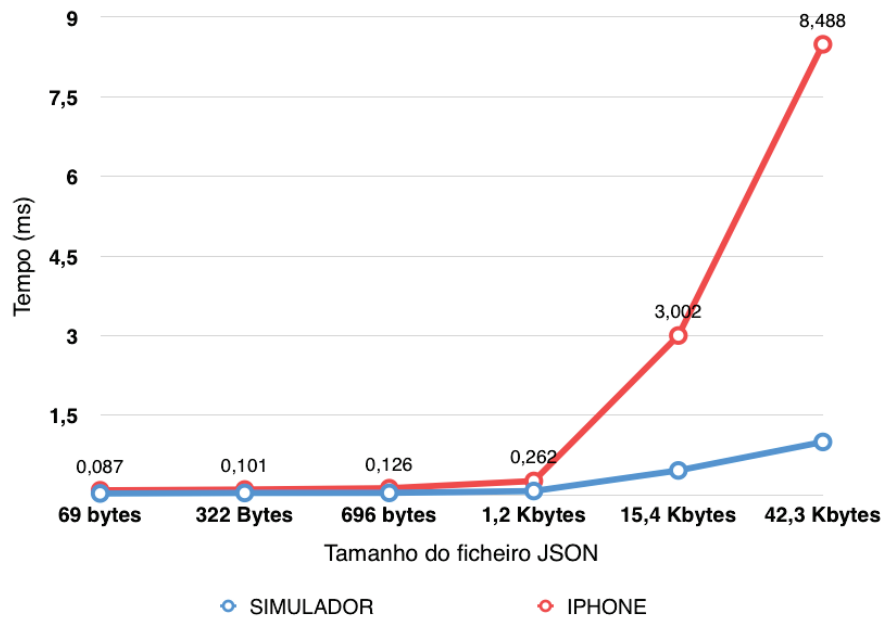
Numa primeira fase, para verificar o tempo de resposta do sistema, foram efetuados um conjunto de pedidos ao servidor para obter a medição de 3 sensores de temperatura. O tempo de resposta foi aproximadamente 33 milissegundos (ms) no simulador, e 73 milissegundos (ms) no dispositivo real.

De notar que todos os valores obtidos, foram calculados usando a aplicação e com base em pelo menos 20 observações, efetuadas para cada um dos casos num intervalo de 5 horas. Os ficheiros resultantes destas medições encontram-se em Anexo na pasta Anexos/JSON_and_Datasets.

Como podemos observar no gráfico da figura 6.1, no eixo das abcissas estão representados ficheiros de teste com tamanhos diferentes. Estes tamanhos refletem o número de parâmetros guardados no ficheiro, ou seja o estado dos vários sensores e atuadores presentes no módulo de controlo. De notar que este eixo não está à escala, e cada valor representa um ficheiro de testes. O eixo das ordenadas apresenta o tempo de resposta do sistema nos vários dispositivos. A linha azul representa o simulador e a linha vermelha o dispositivo real.

Como podemos verificar, o tempo de resposta manteve-se dentro dos mesmos valores para os quatro primeiros casos. Entre 30 a 40 milissegundos (ms), no simulador e entre 70 a 80 milissegundos (ms) no dispositivo real. Para os restantes, o tempo de resposta aumenta para valores na casa dos 500 milissegundos (ms) em ambos os dispositivos.

Figura 6.2: Comparação dos tempos de decoding de conteúdos JSON



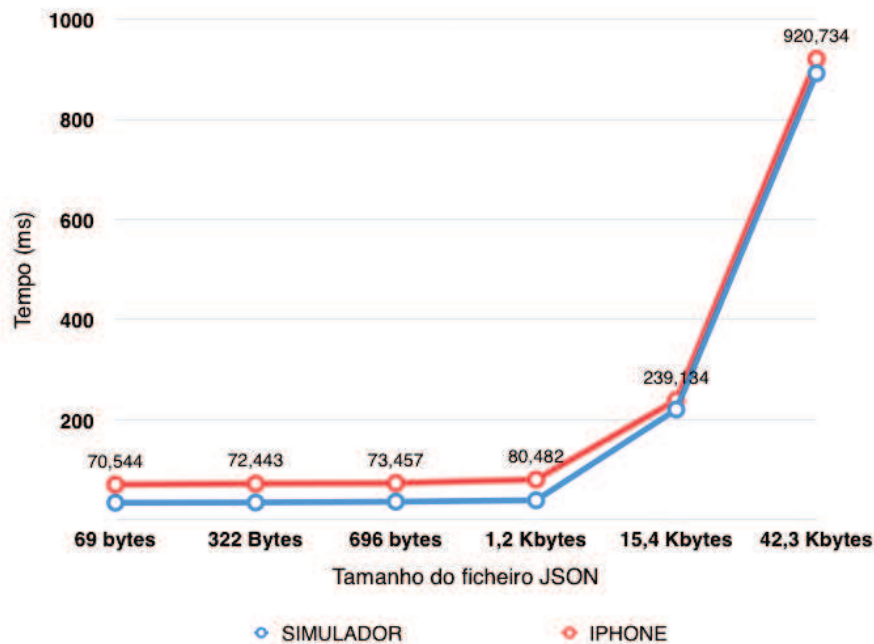
Consideram-se cenários realistas, informações de estado com um número razoável de parâmetros que descrevem os sensores e atuadores. Entendam-se os quatro primeiros casos.

6.4 Testes de performance

Ao recolher os dados, foi tido em conta o decoding de conteúdos JSON, pelo que as observações do gráfico da figura 6.1 não incluem o tempo de decoding, apenas o tempo de resposta. No entanto, e achando-se ser importante ter um termo de comparação de decoding de conteúdos JSON nos vários dispositivos. Essas medições foram registadas durante observações anteriores. Os ficheiros JSON utilizados encontram-se em Anexo na pasta Anexos/JSON_and_Datasets.

De acordo com o gráfico da figura 6.2, verificamos que o tempo de decoding de conteúdos é inferior a 1 milissegundo (ms) para todos os cenários, no caso do simulador. No dispositivos real, é também inferior a 1 milissegundo (ms), mas neste caso, apenas para cenários realistas. Isto sugere, uma quebra de performance da aplicação, que em cenários menos prováveis e dependendo do poder computacional do dispositivo usado, poderá originar um atraso no tempo de resposta da aplicação. No entanto, e como seria de esperar o tempo de decoding é proporcional ao volume de dados.

Figura 6.3: Comparação do tempo de resposta no dispositivo real e simulador simultaneamente



Por fim, foi também possível efetuar testes de concorrência, desta vez utilizando os dois dispositivos em simultâneo. Este teste permite aferir o bom funcionamento da aplicação mesmo com vários clientes a acederem aos mesmos dados simultaneamente. O uso de pedidos assíncronos por parte da aplicação permite um tempo de resposta mais rápido.

Como se pode ver pelo gráfico da figura 6.3 não se verificam diferenças significativas em relação ao caso anterior. Consegue-se apenas observar, um tempo de resposta superior, para o maior volume de dados.

6.5 Análise de resultados

Analisando todos os dados obtidos, podemos afirmar que a aplicação apresenta um comportamento muito semelhante para cenários realistas, ou seja, independentemente do volume de dados, o tempo de resposta mantém-se dentro dos mesmos valores. De acordo com os princípios básicos, determinados pelas capacidades de perceção humanas, o desempenho de uma aplicação pode ser classificado segundo o seu tempo de resposta. 0,1 segundo é o limite para que um utilizador tenha a sensação de reação instantânea [29]. Com resultados abaixo de 0,1 segundo, podemos reafirmar a viabilidade da solução

encontrada.

Ainda assim, e em situações extremas, com maior volume de dados, a aplicação continua a dar resposta em tempo considerado tolerável para o utilizador. Para além disso, os testes de concorrência, validam o seu bom funcionamento, mesmo lidando com vários clientes.

Em relação aos dispositivos utilizados, podemos verificar que existe uma diferença evidente no tempo de resposta, cerca de 30 a 40 milissegundos (ms) para todos os casos. Mesmo não sendo perceptível pelo utilizador, podemos afirmar que dependendo das características do adaptador de rede de cada dispositivo, o tempo de resposta da aplicação pode ser afetado.

Capítulo 7

Conclusão

Este último capítulo, faz um resumo de todo trabalho desenvolvido para este projeto e dos seus resultados. Apresenta algumas considerações sobre o que correu menos bem e quais os problemas encontrados durante a fase de estudo e desenvolvimento. Por fim seguem-se algumas sugestões para trabalho futuro, relacionadas principalmente com a aplicação desenvolvida.

7.1 Considerações Finais

Relacionado com o projeto IVIT, e integrado numa equipa multidisciplinar de I&D, foi realizado um trabalho de investigação, que visa o desenvolvimento de interfaces de controlo e monitorização de um sistema inovador de aquecimento de água. Este trabalho teve como objetivos principais, a definição de um protocolo de comunicação com o reservatório e implementação de uma aplicação móvel.

Em resposta aos objetivos do projeto e após implementar a prova de conceito, foi possível desenvolver uma aplicação móvel funcional, capaz de monitorizar e controlar o reservatório. Os resultados positivos dos testes efetuados, indicaram o excelente desempenho da aplicação, provando ser, de facto, possível controlar remotamente um dispositivo inteligente usando esta solução. Uma solução simples, eficaz e de custo reduzido que pode ser aplicada para o controlo de outros dispositivos.

A oportunidade de integrar uma equipa de investigação e de desenvolvimento constituiu um desafio interessante durante o decorrer do projeto, apesar dos contratemplos que, por vezes, originou. Encontrar pontos de convergência entre o hardware disponível e as necessidades da aplicação provou ser outro dos desafios durante a definição do

protocolo e abordagens de comunicação.

Fazendo um balanço de tudo o que foi conseguido no desenvolvimento deste projeto podemos constatar que, os objetivos principais foram alcançados, apesar de se verificar uma ausência na integração deste sistema, com qualquer uma das frameworks de automação residencial estudadas. Estas frameworks apesar de serem públicas, não são totalmente abertas, isto é, para o desenvolvimento de hardware compatível é necessária uma licença que não pode ser obtida para este projeto por diversas razões. No entanto, nada disso impediu que o projeto obtivesse resultados positivos, como se confirmou na fase de testes.

Este projeto, em toda a sua envolvimento, permitiu desenvolver, competências de trabalho em equipas multidisciplinares e aprofundar conhecimentos no desenvolvimento de aplicações embebidas e aplicações móveis. No fundo, permitiu consolidar todos os conhecimentos adquiridos ao longo do primeiro ano do Mestrado em Engenharia Informática - Computação Móvel.

7.2 Trabalho Futuro

No desenvolvimento de aplicações existe sempre algo que corre menos bem ou que não pode ser implementado. Neste projeto, a aplicação criada é apenas uma prova de conceito, e não o produto final. Não sendo o foco deste projeto, para o caso de estudo apresentado, e de acordo com as potencialidades da tecnologia IVIT, poderiam ser desenvolvidas interfaces mais completas segundo os requisitos do cliente. Aplicar métricas de segurança mais recentes associadas a uma nova iteração no desenvolvimento de hardware, seria mais um passo para o sucesso desta aplicação.

Numa outra perspetiva, e aplicando todos os conceitos adquiridos, poderia ser criada uma aplicação móvel universal que pudesse controlar vários dispositivos compatíveis com as tecnologias usadas nesta solução. Assim seria possível orquestrar todo o tipo de interações entre dispositivos, tal como acontece em algumas das frameworks apresentadas.

Bibliografia

- [1] A. Gurek and I. Korkmaz, “An android based home automation system,” *High Capacity Optical Networks and Enabling Technologies*, pp. 121–125, 2013.
- [2] Z. Pang, “Technologies and architectures of the internet-of-things (iot) for health and well-being,” PhD Thesis, Royal Institute of Technology (KTH), 2013.
- [3] C. Withanage, R. Ashok, C. Yuen, and K. Otto, “A comparison of the popular home automation technologies,” *ISGT ASIA*, pp. 600–605, 2014.
- [4] G. Khusvinder, S.-H. Yang, F. Yao, and X. Lu, “A zigbee-based home automation system,” *Consumer Electronics, IEEE Transactions*, pp. 422–430, 2009.
- [5] Bluetooth, [Consultado em 25 de Maio de 2015]. [Online]. Available: <http://www.bluetooth.com/>
- [6] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, “How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4,” *WCNC 2012 Workshop on Internet of Things Enabling Technologies, Embracing Machine-to-Machine Communications and Beyond*, pp. 232–237, 2012.
- [7] Wi-Fi, [Consultado em 25 de Maio de 2015]. [Online]. Available: <http://www.wi-fi.org/>
- [8] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, “Demystifying 802.11n power consumption,” *2010 international conference on Power aware computing and systems*, pp. 1–5, 2010.
- [9] A.-C. Olteanu, G.-D. Oprina, N. Tapus, and S. Zeisberg, “Enabling mobile devices for home automation using zigbee,” *19th International Conference on Control Systems and Computer Science*, pp. 189–195, 2013.
- [10] Z. Alliance, “Zigbee light link,” [Consultado em 25 de Maio de 2015]. [Online]. Available: <http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbee-light-link/>
- [11] INSTEON, “Insteon whitepaper: Compared,” 2013.

- [12] Z. Pang, Y. Cheng, M. E. Johansson, and G. Bag, “Preliminary study on wireless home automation systems with both cloud-based mode and stand- alone mode,” *IEEE 17th International Conference on Computational Science and Engineering*, pp. 970–975, 2014.
- [13] Elgato, “Avea, dynamic mood light,” [Consultado em 25 de Maio de 2015]. [Online]. Available: <http://www.elgato.com/smart/avea>
- [14] Misfit, “Bolt wirelessly connected smart bulb,” [Consultado em 25 de Maio de 2015]. [Online]. Available: <http://misfit.com/products/bolt>
- [15] Nest Labs, “Works with nest,” 2014. [Online]. Available: <https://nest.com/works-with-nest/>
- [16] J. Levi, “Google’s acquisition of nest could revitalize android @home,” 2014. [Online]. Available: <http://pocketnow.com/2014/01/21/google-smart-home-powered-by-nest>
- [17] SmartThings, “What is smartthings?” 2014. [Online]. Available: <http://docs.smartthings.com>
- [18] A. Hawkinson, “Smartthings, samsung and the open platform,” 2014. [Online]. Available: <http://blog.smartthings.com/news/smartthings-updates/smartthings-samsung-open-platform/>
- [19] Apple, “The homekit framework,” 2014. [Online]. Available: https://developer.apple.com/library/ios/documentation/HomeKit/Reference/HomeKit_Framework/index.html
- [20] K. McLaughlin, “Session 213 introducing homekit,” WWDC14, 2014.
- [21] L. Painter, “Apple homekit release date rumours: First homekit enabled devices on sale next month,” [Consultado em 25 de Maio de 2015]. [Online]. Available: <http://www.macworld.co.uk/feature/apple/apple-homekit-release-date-rumours-3585269/>
- [22] Wink, “Build your connected home,” 2014. [Online]. Available: <http://www.wink.com/about/>
- [23] A. Menychtas, D. Kyriazis, and K. Tserpes, “Real-time reconfiguration for guaranteeing qos provisioning levels in grid environments,” *Future Generation Computer Systems Volume 25*, pp. 779–784, 2009.
- [24] I. Microchip Technology, “Microchip tcp/ip stack help v2013-06-15,” 2013.

- [25] L. Richardson and S. Ruby, *RESTful Web Service*. O'Reilly Media, 2007.
- [26] ECMA-404, "The json data interchange format," Standard, 2013.
- [27] Apple, "What is ios?" [Consultado em 25 de Maio de 2015]. [Online]. Available: <http://www.apple.com/ios/what-is/>
- [28] Apple "Bonjour overview." [Online]. Available: <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/NetServices/Articles/domainnames.html>
- [29] J. Nielsen, "Response times: The 3 important limits," 1993.

Anexos

Anexo 1 - Atividades e Milestones IVIT

Resp.	Dep.	A/T	Descrição (Atividades/Tarefas/Milestones)	Início	Fim	Estado	Meio de verific.	Doc.	Observações
CD		A.1	Gestão técnica do projecto	14-01-01	15-05-31				
CD		T.1.1	Coordenação e controlo de ID&T. - Não aplicável	14-01-01	15-05-31		Não aplicável		
CD		T.1.2	Gestão de custos indirectos/Certificação das despesas por TOC e ROC. - Não aplicável	14-01-01	15-05-31		Não aplicável		
IPL		T.1.3	Gestão de custos indirectos do IPL. - Não aplicável	14-01-01	15-05-31		Não aplicável		
CD		A.2	Estudos	14-01-01	14-11-30				
CD		T.2.1	Análise das necessidades térmicas e de controlo do cliente - Relatório técnico com as necessidades térmicas e de controlo do cliente.	14-01-01	14-03-31		Relatório		
CD		T.2.2	Reavaliação das necessidades térmicas e de controlo do cliente - Relatório técnico onde as inovações desenvolvidas (termodinâmicas, mecânicas e de controlo) são analisadas e confrontadas com as necessidades reais do cliente padrão.	14-07-01	14-11-30		Relatório		
CD		T.2.3	Identificação dos requisitos e necessidades de comunicação (M2M, M2U) - Relatório com as especificações e requisitos de comunicação do equipamento (M2M e M2U), incluindo diagramas das interações M2M e M2U, bem como modelos do interface de controlo do dispositivo ("mockup" da "GUI" de controlo).	14-03-01	14-06-30		Relatório		
CD		A.3	Investigação Termodinâmica e Mecânica	14-01-01	14-06-30				
CD		T.3.1	Estudo do controlo termodinâmico do sistema utilizando perfis de utilização de referência - Instalação computacional de simulação termodinâmica do sistema. - Instalação física de simulação termodinâmica do sistema.	14-01-01	14-06-30		Software		
CD		T.3.2	Introdução ao projecto de design industrial - Documento directriz, onde as principais características de usabilidade do equipamento sob o ponto de vista mecânico e de interacção com o utilizador são analisadas e propostas.	14-03-01	14-05-31		Relatório		
CD		A.4	Desenvolvimento Termodinâmico e Mecânico	14-03-01	14-12-31				
CD		T.4.1	Projecto termodinâmico do sistema - Documento contendo as características de projecto termodinâmico do equipamento que será utilizado na primeira versão do projecto mecânico do sistema.	14-04-01	14-06-30		Relatório		
CD		T.4.2	Optimização do funcionamento termodinâmico do sistema - Documento com as características de projecto termodinâmico final do equipamento, incluindo os paradigmas de controlo termodinâmico propostos.	14-07-01	14-10-31		Relatório		
CD		T.4.3	Projecto mecânico de acordo com normas europeias - Documento com o projecto mecânico do equipamento.	14-03-01	14-05-31		Relatório		
CD		T.4.4	Optimização do projecto mecânico - Documento com o projecto mecânico final do equipamento.	14-09-01	14-12-31		Relatório		
CD		A.5	Testes Termodinâmicos e Mecânicos	14-06-01	15-02-28				
CD		T.5.1	Teste do funcionamento termodinâmico do sistema - Relatórios com os resultados obtidos nos testes do funcionamento termodinâmico do sistema e dos diferentes algoritmos de controlo.	14-07-01	14-10-31		Relatório		
CD		T.5.2	Teste do reservatório de acordo com os processos de certificação mecânica e termodinâmica previstos - Relatórios com as metodologias de ensaio e os resultados obtidos nos testes realizados.	14-06-01	15-02-28		Relatório		
CD		A.6	Investigação do Controlo Local	14-01-01	14-03-31				
CD		T.6.1	Estudo e especificação do conceito inicial - Documento com as especificações detalhadas do conceito inicial do controlo local.	14-01-01	14-01-31		Relatório		
CD		T.6.2	Implementação e prova de conceito - Prova de conceito do controlo local. - Código fonte utilizado na prova de conceito do controlo local. - Documentação técnica que descreve e detalha a implementação da prova de conceito do controlo local.	14-02-01	14-03-31		Equipamento físico Software Relatório		
IPL		A.7	Desenvolvimento do controlo local.	14-03-01	14-11-30				
IPL	EE	T.7.1	Dimensionamento do módulo de controlo local. - Projeto eletrónico do circuito esquema e PCB do módulo de controlo - Placa electrónica que materializa o módulo de controlo local	14-03-01	14-06-30		Relatório Equipamento físico		Realizado. Realizado.
IPL	EE	T.7.2	Desenvolvimento do software de controlo local. - Código fonte do sistema eletrónico do controlo local desenvolvido - Documentação técnica que descreve e detalha a implementação do software do controlo local desenvolvido	14-06-01	14-08-31		Software Relatório		Falta atualização remota e MQTT. Falta completar documentação.
IPL	EE	T.7.3	Desenvolvimento e implementação de mecanismos de segurança, monitorização e recuperação de erros. - Código fonte das funções de segurança, monitorização e recuperação de erros	14-08-01	14-10-31		Software Relatório		Realizado recuperação do sistema em caso de bloqueio. Falta completar documentação.
IPL	EE	T.7.4	Desenvolvimento da interface com o utilizador. - Projecto eletrónico do circuito esquemático e PCB do módulo de interface com o utilizador - Placa electrónica que materializa o interface com o utilizador - Código fonte das funções necessárias ao funcionamento do interface com o utilizador - Documentação técnica que descreve e detalha a implementação das funções necessárias ao funcionamento do interface com o utilizador.	14-09-01	14-11-30		Relatório Equipamento físico Software Relatório		Webserver. Webserver. Webserver. Falta completar documentação.
IPL		A.8	Testes ao controlo local.	14-02-01	15-02-28				
CD		T.8.1	Validação e análise do conceito inicial de controlo local. - Documento com os resultados obtidos nos testes realizados à prova de conceito.	14-02-01	14-05-31		Relatório		
IPL	EE	T.8.2	Testes funcionais ao controlo local. - Documento com os resultados obtidos nos testes realizados sobre a funcionalidade dos diversos periféricos.	14-09-01	14-10-31		Relatório		Falta completar documentação.
IPL	EE	T.8.3	Testes de falhas e operação em situações de erro. - Documento com os resultados obtidos nos testes realizados sobre a reacção do sistema a falhas e operação em situação de erro.	14-11-01	14-12-31		Relatório		Falta completar documentação.
IPL	EE	T.8.4	Testes de usabilidade. - Documento com os resultados obtidos nos testes realizados sobre a usabilidade e receptividade do interface homem-máquina (M2U).	15-01-01	15-02-28		Relatório		Falta completar documentação.

Resp.	Dep.	A/T	Descrição (Atividades/Tarefas/Milestones)	Início	Fim	Estado	Meio de verific.	Doc.	Observações
IPL		A.9	Investigação sobre as interfaces externas de gestão do sistema.	14-07-01	14-11-30				
		T.9.1	Interface de gestão de dispositivos.	14-07-01	14-08-31				
IPL	EE		- Projecto electrónico do circuito esquemático e PCB da interface externa de comunicação do módulo de controlo.				Relatório		Falta completar documentação.
	EE		- Placa electrónica que materializa o circuito esquemático e PCB da interface externa de comunicação do módulo de controlo.				Equipamento físico		Realizado.
IPL		T.9.2	Definição dos protocolos de comunicação com o dispositivo.	14-09-01	14-10-31				
	EI		- Documento com as especificações e detalhes técnicos do protocolo de comunicação máquina-máquina (M2M) adoptado.				Relatório		
IPL		T.9.3	Definição das especificações dos serviços web para fornecedores de serviços.	14-09-01	14-11-30				
	EI		- Documento com as especificações e requisitos do serviço web (web service), detalhando o mais possível a sua futura implementação.				Relatório		
IPL		T.9.4	Definição das especificações de um portal para clientes finais.	14-09-01	14-11-30				
	EI		- Documento com as especificações e requisitos do portal para clientes finais, incluindo modelos de interface de controlo do dispositivo.				Relatório		
CD		T.9.5	Definição das especificações do módulo de análise de dados.	14-09-01	14-11-30				
			- Documento com as especificações e requisitos do módulo de análise de dados, incluindo os detalhes técnicos dos algoritmos de processamento eficiente de dados em bases de dados de grande dimensão que se prevejam implementar.				Relatório		
IPL		A.10	Desenvolvimento das interfaces externas de gestão do sistema.	14-11-01	15-03-31				
		T.10.1	Implementação de drivers para permitir ligar o dispositivo via IP.	14-11-01	15-01-31				
IPL	EE		- Código fonte da programação do sistema electrónico desenvolvido na tarefa T9.1;				Software		Realizado.
	EE		- Documentação técnica que descreve e detalha a integração da interface externa de comunicação com o módulo de controlo local, e as características da pilha de comunicação IP.				Relatório		Falta completar documentação.
IPL		T.10.2	Implementação de um serviço web para fornecedores de serviços que permita gerir os dispositivos	15-01-01	15-02-28				
	EI		- Código fonte da implementação do serviço web para fornecedores de serviços.				Software		
	EI		- Documentação técnica que descreve e detalha a implementação do serviço web para fornecedores de serviços.				Relatório		
IPL		T.10.3	Implementação de um portal para clientes finais.	15-02-01	15-03-31				
	EI		- Código fonte da implementação do portal para clientes finais.				Software		
	EI		- Documentação técnica que descreve e detalha a implementação do portal para clientes finais.				Relatório		
IPL		T.10.4	Implementação de uma API para interligar o dispositivo com outras plataformas.	14-12-01	15-03-31				
	EI		- Código fonte da implementação de uma API genérica que permita aceder à plataforma de gestão dos dispositivos por entidades externas.				Software		
	EI		- Documentação técnica que descreve e detalha a implementação de uma API genérica que permita aceder à plataforma de gestão dos dispositivos por entidades externas.				Relatório		
CD		T.10.5	Implementação de um módulo de análise de dados.	14-12-01	15-03-31				
			- Código fonte da implementação do módulo de análise de dados.				Software		
			- Documentação técnica que descreve e detalha a implementação do módulo de análise de dados.				Relatório		
IPL		A.11	Teste das interfaces externas de gestão do sistema.	15-04-01	15-05-31				
		T.11.1	Testes funcionais dos sistemas implementados.	15-04-01	15-05-31				
IPL	?		- Documento com os resultados obtidos nos testes funcionais, de usabilidade e de performance em carga dos sistemas desenvolvidos.				Relatório		
IPL		T.11.2	Testes de usabilidade dos sistemas implementados.	15-04-01	15-04-30				
	?		- Documento com os resultados obtidos nos testes de usabilidade das interfaces.				Relatório		
IPL		T.11.3	Testes de performance em carga dos sistemas implementados.	15-04-01	15-05-31				
	?		- Documento com os resultados obtidos nos testes de performance em carga dos módulos e do sistema desenvolvido.				Relatório		
CD		T.11.4	Testes funcionais do módulo de análise de dados.	15-04-01	15-05-31				
			- Documento com os resultados obtidos nos testes funcionais do módulo de análise de dados.				Relatório		
CD		T.11.5	Testes de performance em carga do módulo de análise de dados.	15-04-01	15-05-31				
			- Documento com os resultados obtidos nos testes de performance em carga do módulo de análise de dados.				Relatório		
CD		A.12	Integração de sistemas	14-09-01	15-05-31				
		T.12.1	Projecto de design industrial	14-09-01	14-12-31				
CD			- Documento com o projecto de design industrial.				Relatório		
CD		T.12.2	Integração dos projectos de especialidade e projecto do produto final	14-12-01	15-02-28				
			- Documento com o projecto detalhado de todas as componentes físicas, mecânicas e electrónicas, do produto demonstrador da tecnologia IVIT.				Relatório		
CD		T.12.3	Construção do equipamento final	15-02-01	15-04-30				
			- Unidades físicas do produto demonstrador da tecnologia.				Equipamento físico		
CD		T.12.4	Processo de certificação mecânica, informática e de controlo das unidades finais	15-03-01	15-05-31				
			- Processos de certificação da tecnologia IVIT.				Documentação		
CD		A.13	Testes do produto	15-04-01	15-05-31				
		T.13.1	Testes laboratoriais das unidades construídas	15-04-01	15-05-31				
CD			- Documento com os resultados obtidos nos testes laboratoriais do equipamento final.				Relatório		
CD		T.13.2	Testes de campo das unidades construídas	15-04-01	15-05-31				
			- Documento com os resultados obtidos nos testes em campo do equipamento final.				Relatório		
CD		A.14	Divulgação	14-07-01	15-05-31				
		T.14.1	Promoção e divulgação de resultados	14-07-01	15-05-31				
CD			- Publicação de matéria inovadora resultante do desenvolvimento da tecnologia IVIT pelo promotor do projecto.				Particip. eventos		
IPL		T.14.2	Promoção e divulgação de resultados do IPL	14-07-01	15-05-31				
			- Publicação de matéria inovadora resultante do desenvolvimento da tecnologia IVIT pelo co-promotor do projecto.				Particip. eventos		
CD		A.15	Protecção da propriedade industrial	14-01-01	15-05-31				
		T.15.1	Actos relativos ao Pedido de Patente Internacional	14-01-01	14-07-31				
CD			- Actos relativos ao Pedido de Patente Internacional.				Documentação		
CD		T.15.2	Estudo de posicionamento tecnológico em países EPO	14-01-01	14-12-31				
			- Estratégia de desenvolvimento industrial da tecnologia, estudos de concorrência e posicionamento actual e provisional da tecnologia em países europeus.				Relatório		
CD		T.15.3	Vigilância tecnológica em países EPO e elaboração de contractos de licença	15-01-01	15-05-31				
			- Vigilância tecnológica em países EPO.				Relatório		
CD		T.15.4	Estudo de posicionamento tecnológico em países não EPO	14-01-01	14-12-31				
			- Estratégia de desenvolvimento industrial da tecnologia, estudos de concorrência e posicionamento actual e provisional da tecnologia em países não europeus.				Relatório		
CD		T.15.5	Vigilância tecnológica em países não EPO e elaboração de contractos de licença	15-01-01	15-05-31				
			- Vigilância tecnológica em países não EPO.				Relatório		

Legenda:

- Realizado.
- Em execução.
- Não realizado.
- Indefinido, a preencher por cada entidade responsável.

IPL Instituto Politécnico de Leiria
CD ComplexDetails, Lda.
EE Eng. Eletrotécnica
EI Eng. Informática

Anexo 2 - Exemplo de JSON estruturado (JSON4.json)

```
{
  "home_name" : "ipl",
  "device_id" : "IVIT",
  "is_online" : "true",

  "sensors" : {
    "led" :{
      "description" : "Control LEDs",
      "sensor" : {
        "led0" : "~led(0)~",
        "led1" : "~led(1)~",
        "led2" : "~led(2)~",
        ...
      }
    },
    "btn" :{
      "description" : "Control Buttons",
      "sensor" : {
        "btn0" : "~btn(0)~",
        "btn1" : "~btn(1)~",
        "btn2" : "~btn(2)~",
        ...
      }
    },
    "temp" :{
      "description" : "Temperature Sensors",
      "sensor" : {
        "temp0" : "~temp(0)~",
        "temp1" : "~temp(1)~",
        "temp2" : "~temp(2)~",
        ...
      }
    },
    "dcout" :{
      "description" : "DC Power",
      "sensor" : {
```

```

        "dcout0" : "~dcout(0)~",
        "dcout1" : "~dcout(1)~",
        "dcout2" : "~dcout(2)~",
        "dcout3" : "~dcout(3)~"
    }
},
"acout" :{
"description" : "AC Power",
"sensor" : {
    "acout0" : "~acout(0)~",
    "acout1" : "~acout(1)~",
    "acout2" : "~acout(2)~",
    "acout3" : "~acout(3)~",
    "acout4" : "~acout(4)~"
}
}
}
}

```

Anexo 3 - "Remote Control of Smart Devices using Smartphones"

Remote Control of Smart Devices using Smartphones

Jose Brites
Polytechnic Institute of Leiria
Email: jose.brites@ipleiria.pt
Leiria, Portugal

Abstract—A recent growth in smart devices and home automation systems, that connect all kinds of electronic devices within a home, providing control over various functions, has created a real necessity for integrated remote control platforms. Mobile devices provide the ideal interface to wirelessly monitor and control such smart devices due to their portability and wide range of capabilities. Choosing the right approach and technologies can be difficult at times when considering a new product, and depending on the problem they aim to solve, different approaches might be more or less suitable. This paper provides a comparison of popular wireless communication technologies, such as Bluetooth Smart, Wi-Fi and others, comparing their main features and evaluating the referred systems according to some of the users concerns. It also compares different remote control approaches when using a mobile device as the controller. The comparison presented in this paper can benefit product and application engineers in selecting the appropriate technologies and to determine what are the consumer and market tendencies.

Keywords—Home Automation, Smart devices, Smartphones, Bluetooth, Wi-Fi.

I. INTRODUCTION

In recent years we've been witnessing a significant growth of technology and devices that are part of our daily lives. Home Automation represents the introduction of technology within the home [1]. It aims to enhance the quality of life of its occupants, improving comfort and security and increasing energy efficiency while reducing costs. This technology is what we describe as smart devices, and can range from home entertainment systems, to light bulbs, locks, blinds and HVAC (Heating, Ventilation and Air Conditioning). What makes them smart is the added functionality they bring to the orchestration of digital devices creating the desired ambiance. Making sure they can function properly and achieve full potential while being reliable, easy to use and cost effective, is no small task. Moreover, controlling this devices must be a user friendly experience that should be achieved using a single platform or a centralized controller, and not the dozens of remote control devices we find in most homes. As smartphones are now a part of our lives, it is natural to consider their use for this purpose. Small but powerful devices, that fit in the palm of a hand, capable of fast processing, data storage and most importantly network enabled devices. As smartphones are already equipped with Bluetooth and fast Wi-Fi connection, they are the perfect wireless remote controller for smart devices. Furthermore, a smartphone touch screen makes the user experience feel more natural and intuitive.

The concept of Internet of Things, integrates all kinds of sensing, identification, communication, networking and

informatics devices and systems. Connecting seamlessly all the people and things upon interests, so that anybody, at any time and any place, through any device or media, can more efficiently access the information of any object or service. Since this concept appeared there has been an interest in developing technologies to make it a reality [2].

This paper compares popular wireless communication technologies, including Bluetooth, Wi-Fi, ZigBee, Insteon and some variants. We also compare the different approaches used by companies to control their smart devices using a smartphone. The following sections describe some of the existing home automation technologies, remote control apps, the kind of devices they control and how they handle communication between the devices. After listing and comparing systems, this paper describes how new market entrants are making a difference in this area, compared to more traditional approaches and is concluded with possible future work comments.

II. BACKGROUND AND POPULAR TECHNOLOGIES

There are several technologies competing for market share in the home automation system segment. Some are standards like ZigBee, Wi-Fi, Bluetooth and others proprietary technology, including X10, Z-Wave, INSTEON and EnOcean [3]. There are some variants of these communication protocols applied to specific purposes. This paper describes some of them. Most of these technologies were designed for the single purpose of Home Automation. X10 an industry standard, developed in 1975, is identified as the oldest standard in home automation systems [4] and uses the home's power lines for communication with household devices. Most recent approaches try to make use of communication standards compatible with smartphones, through direct communication or using a hub as an access point. In the end, the connectivity with the smartphone is granted by the use of common protocols like Wi-Fi or Bluetooth/Bluetooth Smart. Device manufacturers are now including in their products mechanisms that allow users to control them from a smartphone.

This section describes different technologies and introduces the approaches used to control household items using smartphones, according to different brands.

A. Bluetooth/Bluetooth Smart

Bluetooth is a wireless communications technology, created by Ericsson in 1994, that allows devices to communicate wirelessly through short-range, ad-hoc networks known as piconets. Bluetooth was created with the intent of replacing the cables connecting personal devices while maintaining high

levels of security. The main features of this technology are ubiquitousness, low power and low cost.

Bluetooth operates in the industrial, scientific and medical (ISM) band at 2.4 GHz available and unlicensed in most countries. Because it operates in the same band as other technologies, interference must be dealt with. Bluetooth uses adaptive frequency hopping, working within the spectrum to take advantage of the available frequency. It detects other devices in the spectrum avoiding the frequencies they are using. In terms of range it depends on the manufacturers' implementation and class of radio. In most cases its about 10 meters but, theoretically, it can go up to 100 meters. Power consumption on most commonly used radios uses about 2.5mW of power.

Due to the global acceptance of Bluetooth technology and the wide demand for wireless enabled devices that can be controlled remotely at close distance, power consumption becomes a major concern. Bluetooth Smart is described as the intelligent, power-friendly version of Bluetooth wireless technology, targeted at devices running on small batteries for long periods and is backward compatible with applications for smartphones and tablets already in the market. Designed for maximum battery life it uses 1/2 of the power of classic Bluetooth technology and can go as low as 1/10 depending on use case [5].

Bluetooth Smart is now being used in some popular smart devices. Elgato, has created AVEA, a smart light bulb that can be controlled using an iOS device. Pairing the light bulb with the controlling device its a fairly easy process not requiring any gateway or bridge due to Bluetooth simple pairing process. A single iOS device can control up to 10 light-bulbs independently or orchestrated to work together creating the desired light mood [6]. Using the same technology, Misfit created BOLT, is another smart light bulb controlled by any iOS or Android smartphone equipped with Bluetooth 4.0 and above. It has almost the same features as AVEA, but without the limitation on the number of light-bulbs [7].

B. Wi-Fi

Wi-Fi is the name of a popular wireless networking technology that uses radio waves to provide high speed network connections. The Wi-Fi alliance defines Wi-Fi as any wireless local area network product based on the IEEE 802.11x standards. Wi-Fi is used in many devices like, personal computers, smartphones and tablets, enabling them to connect to any network resource, typically the Internet, through a wireless network access point. Wi-Fi also allows communication directly from one device to another without an access point as an intermediate creating an ad-hoc network. With the ability to connect to any network resource, controlling devices can go beyond local remote control. It enables remote access from the Cloud providing the ability to control this devices from anywhere in the world.

Like Bluetooth, Wi-Fi also uses the 2.4 GHz ISM band, but because of its high demand, most recent implementation standards like 802.11n and the latest 802.11ac are dual-band, operating in both 2.4 and 5 GHz bands. Dual-band doubles the capacity allowing devices to use the less crowded 5 GHz band for high performance applications and the 2.4 GHz for basic

needs. In range it as about 20 meters of indoor range and can go up to 100 meters outdoors [8].

This technology is featured in popular smartdevices, such as the NEST Learning Thermostat and the LIFX LED Light bulbs. Both have a smartphone app that uses Wi-Fi to communicate with the device. The pairing process is fairly straight forward. Nest connects directly with the home network while with LIFX needs a connection to the light bulb first [9], [10].

C. ZigBee

Based on the IEEE 802.15.4 standard, ZigBee is a low-cost, low-power, wireless mesh network technology used in home automation for wireless control and application monitoring. ZigBee is a technology intended to be simpler and less expensive than other wireless personal area networks like Bluetooth and Wi-Fi.

It also operates on the ISM band at 2.4GHz worldwide. Optionally it can be used at 915 MHz in the USA and Australia, 868 MHz in Europe and at 784 MHz in China. Data rates vary from 20 kbit/s to 250 kbit/s. Its low power consumption and open specification makes ZigBee perfect for long battery life operated devices. While limiting transmission range from 10 meters to 100 meters depending on power output and environmental characteristics, due to its mesh network devices are able to communicate with each other and can act as repeaters allowing data transmission over long distances [11].

ZigBee's interoperability comes from standardization at all levels of the network, especially the application level which is what the user is more familiar with. For this reason ZigBee provides application standards for specific uses to insure that devices from different vendors can work together. For instances, Phillips and other brands contributed their expertise to the development of ZigBee Light Link [12].

Phillips HUE, uses this standard on their LED light-bulbs. It allows consumers to gain wireless control over all their LED fixtures, light bulbs and other switches. To allow wireless access from a smartphones, HUE uses an Ethernet central hub, connected to the home network router. The light fixtures communicate with each other and with the hub using the ZigBee Light Link standard and then using Wi-Fi to connect to a smartphone enabling remote control [13]. The hub approach is described in section III.

D. INSTEON

INSTEON is a proprietary technology that combines the home's power line system with a wireless system. Devices communicate with each other using the INSTEON protocol over the air via radio frequency at 915MHz and over the power line in a full mesh network configuration. This technology was developed to replace the X10 standard, on of the oldest available home automation standards.

INSTEON offers a wide range of low cost home automation products, from wall switches and wall outlets, to thermostats and LED light bulbs. It can interoperate with other devices using different networking protocols, including TCP/IP and ZigBee and has also a solution for remote control using a smartphone App. Once again using a hub approach. This App is available for iOS, Android, Windows and Windows Phone.

TABLE I. WIRELESS COMMUNICATION TECHNOLOGIES

	ZigBee IEEE 802.15.4	Wi-Fi IEEE 802.11b/g/n/ac	Bluetooth Smart	INSTEON
Data Rate	250 kbps	100 Mbps- 1000 Mbps	1 Mbps	38.4 kbps
Range	10 - 100m	20 - 100m	10 - 100m	50m
Frequency	915 MHz US 868 MHz EUR 2.4 GHz World	2.4 and 5 GHz	2.4 GHz	915 MHz US 868 MHz EUR 921 MHz AUS
Interference	dynamic freq. selection	dynamic freq. selection	adapting freq. hopping	-
Power	90mW TX 72mW RX 3uA sleep	1.28W TX 0.94W RX 46mW sleep	84mW TX 66mW RX 0.2mW sleep	-
Cost	\$\$	\$\$\$	\$	\$

E. Technology Compared

After detailing popular home automation technologies for wireless communication and remote control within a home, a brief analysis can highlight the potential of the presented technologies while reporting the advantages and disadvantages of each one when applied to the problem referred by this paper, remote control using a smartphone.

As it is well known smartphones currently in the market are all equipped with some of the latest communication technologies, some of them mentioned previously in this paper like Wi-Fi and Bluetooth, and others that may or may not be useful for wireless remote control, including NFC and infrared. Infrared has been used for short range communication and remote control in many applications throughout the years. Because today's smartdevices are becoming more complex and as consumers demand more features, infrared is being discarded as a viable option, and thus not mentioned previously.

Each of the previously described technologies have its own advantages when compared to others. As Table I shows, ZigBee seems to be one of the most power efficient at the lowest cost, but like INSTEON it requires additional hardware to translate data transmitted using ZigBee's or INSTEON protocol to a smartphone compatible communication standard.

Bluetooth and Wi-Fi are the only communication technologies that are used for wireless remote control using smartphones, either by direct communication or facilitated by additional hardware. This paper discards other technologies and focus on this two communication standards, discussing which one is more suitable for different remote control approaches.

The next section, describes different approaches using Bluetooth or Wi-Fi.

III. REMOTE CONTROL APPROACHES

After highlighting the technologies and their characteristics, it is important to understand how they can be used in a communication architecture to remote control smart devices wirelessly, using a smartphone, and the technical challenges involved. This paper describes in this section, the two main approaches used to address this problem, the local mode, where devices are interconnected to each other directly or through a local gateway, and the cloud based mode where devices are connected to a cloud server (Figure 1).

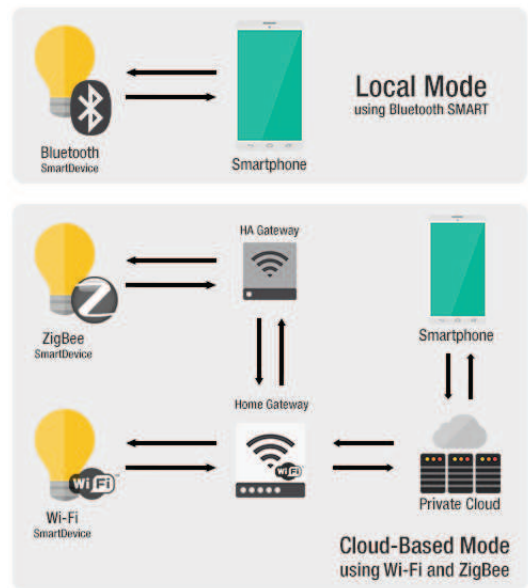


Fig. 1. Remote control approaches architecture

A. Local Mode

As the name states local mode eliminates the need for a remote server, and devices are able to communicate to each other directly or through some kind of local gateway. Getting rid of internet dependency can reduce significantly some security and privacy issues cutting out any malicious remote control attempt from outside the local network.

This can be a issue when system's configuration and maintenance are in order, making it harder for systems suppliers and service providers to reach, monitor and update all their products when problems are detected.

This approach keeps the currently established home automation systems suppliers in control of their devices while maintaining their essential roles in the home automation value chain. Not relying on third party frameworks or APIs that are trying to emerge in the home automation market, it eliminates any problems that may cause the product to malfunction or mislead the consumer.

Bluetooth is usually the technology chosen in this approach due to its low power, low cost and at the same time secure network architecture. This enables the creation of cheap, but reliable devices that can seamlessly be integrated with a smartphone in a simple pairing process.

B. Cloud-Based Mode

The need for a digital representation of the real world, where all kinds of sensing, control and networking are seamlessly connected to people, so that anybody, at any time and any place, thought any device, can access, control and monitor any object, has driven the consumer market to an increasing interest in cloud-based home automation solution.

The Cloud creates this ubiquitousness, connecting devices to a server in the cloud enables remote control and efficient

management from anywhere reducing effort to end consumers during the installation and configuration process. Manufacturers and service providers are also able to communicate remotely with the devices enabling remote maintenance or simple monitoring and data collection.

At the same time security and privacy concerns from end consumers become an important challenge for suppliers that needs to be carefully addressed. Cloud-based brings IP-based communication and all its known problems in security and configuration. Communication must be secured and configuration should automatically create a usable network, using technologies like zero-configuration networking.

C. Hybrid Solutions

Given the pros and cons of both local mode and cloud-based mode, it is easy to understand why an hybrid solution is in order. Simplifying configuration and maintenance while reducing security and privacy issues.

Pang et al. [14] proposed an IP-based hybrid architecture which can take the advantages of the both modes with minimized cost to switch between the two modes. With a scalable gateway architecture, the prototype created was based on the 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks), bringing IP compatibility to low power and low cost wireless devices.

Apple is also making efforts on the HomeKit platform, to make it an hybrid solution easily set up. Using the current Apple TV (equipped with Bluetooth and Wi-Fi) as a gateway, users are able to control their HomeKit ready devices away from home [15].

IV. FRAMEWORKS, APIS AND TOOLKITS

Frameworks and APIs are now making their debut in home automation. With the appearance of this new smart devices created by a wide range of vendors, each one with its own approach, there's a increasing necessity to create solutions able to bring together all this devices. Standards and frameworks allow new devices to be easily integrated with the existing automation systems, remote control platforms and applications. This need has come to the attention of large technological companies pushing efforts to create home automation frameworks able to provide a complete solution of hardware and client applications.

Services like Apple's HomeKit are forcing a new paradigm of integration, where all home devices should be connected to a single control platform making it user friendly and secured. To do so Apple promotes the use of a protocol driven approach alongside vendors and a public API for communicating with and controlling connected devices. Apps can be created by the developer community and not necessarily by vendors. Other Frameworks and APIs like SmartThings and Wink promote the use of a gateway able to communicate with other protocols, including, ZigBee and Z-wave, thus offering a complete solution of hardware, software, user experience and support and allowing existing products to interact with each other.

The use of protocol driven frameworks makes development so much easier for application developers. Apps can be developed at a higher level and do not require specific knowledge

TABLE II. BLUETOOTH SMART AND WI-FI COMPARISON

	Bluetooth Smart	Wi-Fi
Number of RF Channels	79	14(2.4 GHz)
Number of Connections	Not defined; implementation dependant	255, theoretically
Authentication	Shared Secret	WPA2
Encryption	AES block cipher	RC4 stream cipher (WEP); AES block cipher
Data Protection	24-bit CRC	32-bit CRC
Application Areas	Mobile Phones; Tablets; PCs; gaming; watches; sports and fitness; health-care; security and proximity; automotive; home; automation	Mobile Phones; Tablets; PCs; gaming; automotive; home; automation

of communication standards or protocols, it just works. It can also benefit from other features available on the client's device such as voice control. From a vendors perspective, in order to have a compatible product, a specific set of technical requirements and specifications must be followed when developing the product. This ensures consistency but can often be seen as an inconvenience for manufacturers. Other frameworks that offer hardware solutions allow existing smart devices to communicate with each other and to be remotely controlled using the same App. To achieve that, protocol translations are in order, requiring developers to create custom code for each new device.

V. SYSTEMS COMPARED

After describing the technologies and studying the different remote control approaches, this paper compares the referred systems according to some of the users concerns, things like security and privacy, energy efficiency, performance and cost.

A. Security and Privacy

When exposing objects to its digital representation, security threats become a reality. To protect users and insure their privacy, communications must be made secured. Depending on the communication technology used, different threats require different security measures. Bluetooth for instance, has defined powerful security features to protect the communication of a user's data and identity, avoiding common attacks like man-in-the-middle (MITN), passive eavesdropping and identity tracking. A MITM attacker has the ability to monitor and control a conversation within a communication channel. To protect against it Bluetooth uses a passkey entry pairing method, giving the user control over communications. Passive eavesdropping is another threat can happen when an attacker is able to listen to a private conversation without consent. Cryptography techniques like the ECDH public key method ensure protection against this kind of attack. Bluetooth also applies an unique identifier to each device, that can make devices traceable if its not properly masked. By changing private addresses frequently, only trusted parties could resolve them [16].

As for Wi-Fi, unsecured wireless networks allow attackers within the same geographical network range to "sniff", capture

and record private information or even gain unauthorized access to private network resources compromising security. To protect against this threats Wi-Fi uses methods like WEP or WPA. The current standard is WPA2, encrypting the network with a 256-bit key. A longer key improves security over WEP. To make it even more secured if an attacker has gained access to the wired network somehow, encryption and authorization can be applied in the application layer, using technologies like SSL, SSH and others.

B. Energy Efficiency

Energy efficiency has become a real concern since the concept of internet of things first appeared. Turning every day objects into network enabled devices while keeping them energy efficient for long periods of time has been a challenge for the more recent standards of communication. Bluetooth Low Energy (Smart) was specifically designed to work at a low power state. Siekkinen et al. [17] studied the energy consumption of BLE by measuring real devices with a power monitor. Results have shown that BLE consumes indeed extremely little energy and as a very attractive ratio of energy per bit transmitted. However it can be even more optimized by allowing more packets to be sent within a connection event and by implementing AFH to combat interference.

Halperin et al. [18] have demystified the power consumption of Wi-Fi. Using an 802.11n NIC across multiple configurations it has been found that some configuration changes, like doubling the bandwidth to double the bit rate have very little impact while others like adding a transmit chain, cost a lot, and that some changes apparently related to power, for instance, transmit power control, have very little effect on the power actually consumed. Despite this, compared to Bluetooth, Wi-Fi is for sure much more power hungry.

C. Performance

In terms of performance, the difference is clear between this two technologies. On one hand, Bluetooth Smart was design for maximum power efficiency, having an effect on data rate and range. It is ideal for small battery operated devices that process small amounts of data. On the other hand, most recent implementations of Wi-Fi focus on providing higher speeds and not improving range or energy efficiency. Although low power versions of Wi-Fi are often used, its main purpose in home automation is to provide remote access across the cloud, and its usually applied to devices requiring a constant source of power.

The effects of wireless communication technologies coexistence, or interference are also directly related to performance. In [17] results demonstrated a 60% packet loss when no mechanisms to combat interference were applied. Due to its popular adoption is the past years, more recent implementations of Wi-Fi use dual band mechanisms. Using the less crowded 5 GHz it allows for high performance applications when needed.

D. Cost

Cost is also a concern, not only from consumers, but for developers and manufacturers as well. This analysis is based on the cost required to purchase a basic module, not including installation or labour costs. A quick research shown that, as

expected, the cost of Bluetooth, even Bluetooth 4.0 modules, is considerably smaller than Wi-Fi. Bluetooth 4.0 modules can cost as low as 5\$ depending on implementation while Wi-fi modules can cost 30\$ or more.

VI. CONCLUSION

This paper has presented an overview of popular wireless communication technologies used in home automation systems including Bluetooth Smart and Wi-Fi with an evaluation in terms of performance, security, energy efficiency and cost. It has also compared different remote control approaches when using mobile devices as the controller. This paper cannot draw any conclusion regarding which technology is best and which approach is more suitable, since different problems may require different solutions. We can only state an obvious market tendency for using Bluetooth Smart for local solutions and Wi-Fi to provide the ubiquitousness that only the Internet can bring.

REFERENCES

- [1] A. Gurek and I. Korkmaz, "An android based home automation system," *High Capacity Optical Networks and Enabling Technologies*, pp. 121–125, 2013.
- [2] Z. Pang, "Technologies and architectures of the internet-of-things (iot) for health and well-being," PhD Thesis, Royal Institute of Technology (KTH), 2013.
- [3] C. Withanage, R. Ashok, C. Yuen, and K. Otto, "A comparison of the popular home automation technologies," *ISGT ASIA*, pp. 600–605, 2014.
- [4] G. Khushvinder, S.-H. Yang, F. Yao, and X. Lu, "A zigbee-based home automation system," *Consumer Electronics, IEEE Transactions*, pp. 422–430, 2009.
- [5] Bluetooth, [Consulted May 25th, 2015]. [Online]. Available: <http://www.bluetooth.com/>
- [6] Elgato, "Avea, dynamic mood light," [Consulted May 25th, 2015]. [Online]. Available: <http://www.elgato.com/smart/avea>
- [7] Misfit, "Bolt wirelessly connected smart bulb," [Consulted May 25th, 2015]. [Online]. Available: <http://misfit.com/products/bolt>
- [8] Wi-Fi, [Consulted May 25th, 2015]. [Online]. Available: <http://www.wi-fi.org/>
- [9] Nest, "Nest learning thermostat," [Consulted May 25th, 2015]. [Online]. Available: <http://nest.com/thermostat/life-with-nest-thermostat/>
- [10] LIFX, "Lifx smart bulb," [Consulted May 25th, 2015]. [Online]. Available: <http://www.lifx.com/>
- [11] A.-C. Olteanu, G.-D. Oprina, N. Tapus, and S. Zeisberg, "Enabling mobile devices for home automation using zigbee," *2013 19th International Conference on Control Systems and Computer Science*, pp. 189–195, 2013.
- [12] Z. Alliance, "Zigbee light link," [Consulted May 25th, 2015]. [Online]. Available: <http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbee-light-link/>
- [13] Phillips, "Hue personal wireless lighting," [Consulted May 25th, 2015]. [Online]. Available: <http://www2.meethue.com/>
- [14] Z. Pang, Y. Cheng, M. E. Johansson, and G. Bag, "Preliminary study on wireless home automation systems with both cloud-based mode and stand-alone mode," *IEEE 17th International Conference on Computational Science and Engineering*, pp. 970–975, 2014.
- [15] L. Painter, "Apple homekit release date rumours: First homekit enabled devices on sale next month," [Consulted May 25th, 2015]. [Online]. Available: <http://www.macworld.co.uk/feature/apple/apple-homekit-release-date-rumours-3585269/>
- [16] S. Sandhya and D. K. A. Sumithra, "Analysis of bluetooth threats and v4.0 security features," *2012 International Conference on Computing, Communication and Applications (ICCCA)*, pp. 1–4, 2012.

- [17] M. Siekkinen, M. Hienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4," *WCNC 2012 Workshop on Internet of Things Enabling Technologies, Embracing Machine-to-Machine Communications and Beyond*, pp. 232–237, 2012.
- [18] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n power consumption," *2010 international conference on Power aware computing and systems*, pp. 1–5, 2010.