João Pedro Gomes Rosa

Licenciado em Ciências de Engenharia Eletrotécnica e de Computadores

Mecanismos de Segurança IoT

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador: João Almeida das Rosas, Professor Auxiliar, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Júri

Presidente:	Doutor João Miguel Murta Pina - FCT/UNL
Arguente:	Doutor José Manuel Matos Ribeiro da Fonseca - FCT/UNL
Vogal:	Doutor João Almeida das Rosas - FCT/UNL

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2021

IoT Security Mechanisms

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Para a Tana, para o Zé, para a Trindade e para a Maria João

Agradecimentos

Gostaria de utilizar esta secção para agradecer a todos os que, de alguma maneira, contribuíram para a realização desta dissertação.

Em primeiro lugar, quero agradecer ao professor João Rosas por me ter indicado este tema de dissertação, assim como por todo o tempo e apoio que disponibilizou para me ajudar na elaboração desta dissertação.

Gostaria de agradecer também a todos os meus colegas de curso que, de alguma forma, estiveram presentes no meu percurso enquanto estudante.

Um obrigado também a todas as pessoas que estão ou estiveram ligadas à anTUNiA, por tudo aquilo que me ensinaram e por todos os bons e maus momentos que passámos.

Finalmente, gostaria de exprimir um agradecimento especial aos meus amigos, aos meus pais e à minha irmã por todo o apoio incondicional que me deram.

Resumo

Hoje em dia, temos todo o tipo de informação na ponta dos nossos dedos. Acesso à Internet é característica essencial de quase todos os dispositivos tecnológicos e já se começa a ver esta capacidade em certos aparelhos que tradicionalmente não são associados com esta tecnologia.

Aliado ao desenvolvimento desses dispositivos e das suas habilidades de se interligarem, surgiu o conceito de Internet das Coisas. Isto é, todos os dispositivos, serviços e recursos conectados através da Internet, resultando num ambiente inteligente, rápido e autónomo.

No entanto, a implementação deste conceito está associada a um problema importante. Ter um conjunto grande e diversificado de dispositivos e tecnologias suscita preocupações sobre segurança, privacidade e integridade.

Esta dissertação pretende encontrar soluções aos problemas presentes nos aspetos de segurança e privacidade de redes IoT causados pelas vulnerabilidades presentes nestes sistemas que soluções tradicionais não conseguem responder. Através da identificação das áreas mais problemáticas destes sistemas assim como os tipos de ataques que incidem sobre os mesmos, este trabalho procura responder às questões de segurança e privacidade através da implementação de um protótipo de uma rede IoT em duas fases distintas.

Numa primeira instância, o protótipo tem como objetivo elucidar as principais falhas de um sistema IoT típico através da construção de uma rede mais simples, para que, numa segunda abordagem, seja possível responder ao maior número de problemas de segurança e privacidade numa rede mais complexa.

Palavras-chave: *Internet*, IoT, Segurança, Rede, Dispositivos, *Smart-Home*

Abstract

Nowadays, information is at the tip of our fingers. Access to the Internet is now a standard feature of almost every mobile technological device and it is also starting to appear in gadgets that usually are not associated with networking technologies.

With the current development of these devices and their interconnection capabilities, the concept of Internet of Things emerged. Accordingly, every device, service and resource are all connected through the Internet in order to present smart, fast and autonomous environment of multiple technologies.

However, the implementation of such concept is associated with a main barrier. Having a big and diverse cast of systems raises concerns about security, privacy and integrity.

This dissertation aims to find solutions to the problems of the security and privacy aspects of IoT networks caused by the vulnerabilities present in these systems that traditional solutions cannot respond to. Through the identification of the most problematic areas of these systems as well as the types of attacks that affect them, this work seeks to answer security and privacy issues by implementing a prototype of an IoT network in two distinct phases.

In the first instance, the prototype aims to clarify the main flaws of a typical IoT system by building a simpler network, so that, in a second approach, it is possible to respond to the greatest number of security and privacy problems in a more complex network.

Keywords: Internet, IoT, Security, Network, Devices, Smart-Home

Índice

Lista de Figuras

Lista de Tabelas

Listagens

Glossário

Bluetooth	Bluetooth é uma tecnologia de comunicação utilizada para a troca de dados entre dispositivos em curtas distâncias de modo a construir redes de área pessoal sem fios. Através da utilização de ondas de rádio, esta tecnologia permite o envio e receção de dados entre dois dispositivos emparelhados sem a necessidade de estarem em linha de visão um do outro.
DoS	DoS significa *Denial of Service* (negação de serviço em português) e representa um ataque a um sistema com o objectivo de tornar um certo recurso indisponível de aceder pelos utilizadores legítimos.
HTTP	Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto em português) é um protocolo de comunicação que serve como base de todo o sistema da comunicação de dados da Internet. Através de ligações lógicas como *links* (em português, hiperligações), este protocolo visa intermediar a troca de informação entre cliente (como um *web browser*, por exemplo) e um servidor.
ISP	Internet Service Provider (em português, Fornecedor de Acesso à Internet) é a entidade responsável por distribuir serviços ligados à utilização da Internet. São principalmente responsáveis pelo acesso de utilizadores à Internet mas também podem oferecer outros serviços como registo de nome de domínios e hospedagem de *sites*.
Servidor *Cloud*	Um servidor *cloud* é o principal recurso de computação por nuvem, que consiste na disponibilidade de recursos, como armazenamento de informação ou poder de processamento, sem gestão ativa direta do utilizador. Através de um acesso pela Internet, um servidor em nuvem possibilita a disponibilização de diversos serviços e recursos a qualquer hora e em qualquer lugar.

URL — Uniform Resource Locator é o termo técnico de um endereço *web* e representa uma maneira de aceder a um recurso numa determinada rede. O exemplo mais comum de um URL é o de um endereço de uma *web page*.

Wi-Fi — Wi-Fi é uma marca registada da organização sem fins lucrativos "Wi-Fi Alliance" e representa um conjunto de protocolos de redes sem fios. Estes protocolos permitem a ligação de dispositivos como computadores ou *smartphones* entre si e à Internet através da conexão a um ponto de acesso sem fios.

Siglas

CSV	Comma-Separated Values
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
IT	Information Technology
LDR	Light Dependent Resistor
LED	Light-Emitting Diode
MAC	Media Access Control
PIN	Personal Identification Number
PIR	Passive Infrared
PTC	Positive Temperature Coefficient
PWM	Pulse-Width Modulation
RAM	Random Access Memory
RFID	Radio Frequency Identification
ROM	Read-Only Memory
USB	Universal Serial Bus

Capítulo 1

Introdução

1.1 Motivação

Hoje em dia, vivemos na Era da Informação. Em apenas uns curtos anos, a popularização da Internet, ao colocar uma enorme quantidade de informação à distância de uns cliques, mudou o nosso modo de vida, a maneira como trabalhamos, como fazemos negócios e como comunicamos [9].

Atualmente pensa-se que o número de utilizadores da Internet seja aproximadamente de **4.536** milhões de pessoas, ou seja, sensivelmente **58.8%** da população mundial, a qual está distribuída, por continentes, de acordo com a Figura 1.1 [9].

Este número enorme de pessoas está a aumentar de dia para dia e, de maneira a responder a este crescimento, a própria Internet está-se a adaptar a novos meios de comunicação. Cada vez mais se verifica um crescimento de dispositivos móveis e serviços para os mesmos [7], o que, aliado ao sempre presente desenvolvimento das tecnologias ligadas à Internet, fez com que nascesse o conceito de *Internet of Things* (em português, Internet das Coisas), onde todos os dispositivos dentro duma determinada rede, estão ligados entre si através da Internet, dando assim acesso instantâneo a um utilizador a qualquer recurso do sistema ligado à dita rede [19].

Este conceito, que já está a começar a ser designado pela "terceira revolução industrial", estima-se que, ainda antes do fim de *2020*, se chegue aos *200* milhões de dispositivos focados em IoT e para diversos propósitos, como *home automation*, educação, distribuição de energia, transportes ou até mesmo finanças [5].

Por definição, os sistemas IoT vão formar grandes e diversas redes de dispositivos, aplicações e serviços, sempre conectados entre si, o que resulta na vantagem tremenda de ter acesso perto de instantâneo a qualquer parte da rede. Porém, existem algumas preocupações com este conceito [5].

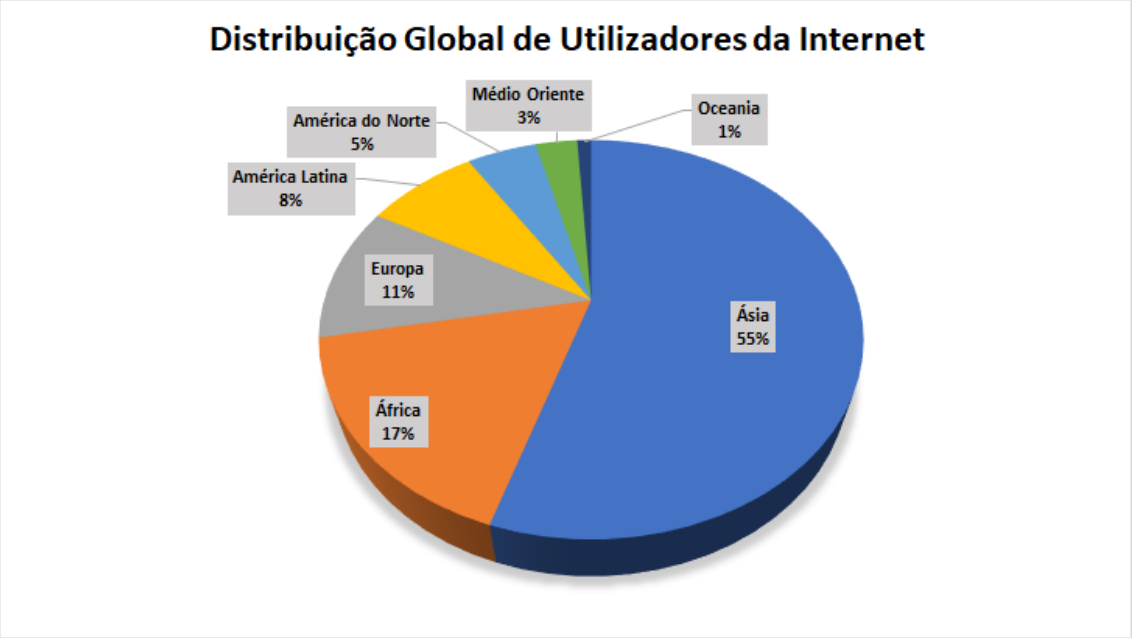

Figura 1.1: Gráfico das percentagens de Utilizadores da Internet por Continente, como visto em [9] em Janeiro de 2020

1.2 Descrição do Problema

Devido à sua natureza inerente, uma rede IoT típica irá invariavelmente ter um número considerável de dispositivos e serviços diferentes, o que também significa que irá haver uma grande vulnerabilidade a diversos ciberataques, que terão como alvos esses dispositivos e serviços, pondo em causa a segurança, integridade, privacidade e confidencialidade da sua própria rede [7].

Qualquer serviço, aplicação ou dispositivo que que faça parte de uma rede IoT vai sempre utilizar vários e diferentes recursos dessa rede, o que resulta numa enorme quantidade de informação transmitida entre diferentes pontos da rede. Esta heterogeneidade de diversos mecanismos e tecnologias significa que métodos tradicionais, tanto de recolha de dados, como de processamento de informação, podem não ser ideais, introduzindo assim possíveis vulnerabilidades de segurança nas diversas trocas de informação entre os diferentes serviços de uma rede [8].

Em suma, um sistema IoT é composto por um número grande e diverso de dispositivos e tecnologias, o que causa problemas, tanto de segurança como de privacidade, que se distinguem dos desafios tradicionais presentes no mudo da segurança em IT [1]. Problemas como abuso de privilégios e métodos fracos ou mesmo inexistentes de autenticação entre os vários dispositivos da rede são assuntos comuns no tema de segurança em redes IoT, assim como os diferentes atributos de um ataque que incida sobre as mesmas [23].

Relativamente a trabalhos de pesquisa e investigação, o assunto de segurança, privacidade e confidencialidade ainda se encontra numa fase inicial [15]. Embora já existam alguns trabalhos que reforçam partes de um sistema IoT já estabelecidas [24], existem

ainda questões sérias por responder, sendo talvez a maior a falta de um protocolo *standard* de uma rede IoT como um conjunto num todo, em vez de uma junção de diferentes tecnologias e serviços [10].

São estas questões que este trabalho tem como objetivo responder, através da elaboração de um protótipo de uma rede IoT utilizando duas iterações diferentes. Numa primeira abordagem, empregando o uso de serviços mais generalizados, procurou-se construir uma rede IoT simples, contendo apenas uma aplicação para telemóvel, um serviço *web* e um dispositivo de medição de temperatura, de modo a realçar os principais problemas de segurança para que uma segunda iteração da rede, mais complexa, pudesse ter uma maior facilidade em responder a esses desafios.

Nesta segunda abordagem, através de tecnologias especialmente direcionadas para o uso em redes IoT, aumentou-se o número dos dispositivos, resultando assim num aumento da complexidade da rede que agora não só tem de registar valores lidos por sensores, como também atuar no ambiente em que está inserida, procurando sempre alcançar um estado de segurança máximo respondendo a todas as questões existentes levantadas pela primeira iteração da rede.

1.3 Estrutura da Dissertação

Esta dissertação é constituída pelos seguintes 8 capítulos: Introdução, O Conceito de IoT, Problemas de Segurança, Falhas Existentes e Trabalhos Relacionados, Descrição Informal dos Sistemas Implementados, Estrutura dos Sistemas Implementados, Validação e Conclusão.

- No presente Capítulo 1, **Introdução**, pode-se encontrar uma contextualização do conceito de IoT (Subcapítulo 1.1) e aos desafios que motivaram esta dissertação (Subcapítulo 1.2).

- O Capítulo 2, **O Conceito de IoT**, representa a primeira parte do Estado da Arte e esclarece como nasceu o conceito de IoT e como pode ser aplicado a uma *smart home* (Subcapítulo 2.1). Explica também a arquitetura de uma rede, as suas camadas de arquitetura e as diferentes tecnologias usadas (Subcapítulo 2.2).

- O Capítulo 3, **Problemas de Segurança**, é a segunda parte do Estado da Arte e explora os diferentes tipos de ameaças e vulnerabilidades de um sistema IoT através de um estudo das vulnerabilidades das redes IoT (Subcapitulo 3.1) e dos atributos de possíveis ataques orientados contra as mesmas (Subcapitulo 3.2), assim como alguns exemplos de possíveis ataques direcionados contra as diversas camadas de uma rede (Subcapitulo 3.3).

- O Capítulo 4, **Falhas Existentes e Trabalhos Relacionados**, conclui o Estado da Arte descrevendo as condições atuais de segurança em redes IoT, as suas principais

lacunas (Subcapítulo 4.1) e alguns trabalhos relacionados com o tema (Subcapítulo 4.2).

- O Capítulo 5, **Descrição Informal dos Sistemas Implementados**, contextualiza o trabalho realizado através de uma pequena explicação do desenvolvimento do mesmo (5.1) e explicita os requisitos de cada uma das redes implementadas (5.2).
- O Capítulo 6, **Estrutura dos Sistemas Implementados**, descreve quais os recursos, serviços e dispositivos utilizados na implementação das redes, observando *hardware* comum a ambas (6.1), e como foram implementadas tanto a primeira rede (6.2) como a segunda (6.3).
- O Capítulo 7, **Validação**, explicita as condições do cenário de teste considerado para as duas redes (7.1) e verifica se estas cumprem os requisitos implementados (7.2).
- O Capítulo 8, **Conclusão**, realiza um pequeno resumo de todo o trabalho realizado (8.1) e indica algum desenvolvimento futuro (8.2).

Capítulo 2

O Conceito de IoT

Este capítulo serve de introdução ao Estado da Arte da redes IoT e está dividido em duas partes, sendo que a primeira secção serve para esclarecer como surgiu o conceito de **IoT** e quais a suas aplicações, especialmente no ramo da domótica, onde se irá inserir tema central desta dissertação, que é a segurança deste tipo de redes. A segunda parte deste capítulo explica a arquitetura de um sistema típico, isto é, o que são as suas camadas estruturais e como funcionam, referindo sempre que tipo de dispositivos, tecnologias e serviços constituem as ditas camadas.

2.1 O que é IoT?

Acredita-se que o termo "Internet of Things", em português "Internet das Coisas", ou abreviando "IoT" tenha sido usado pela primeira vez por Kevin Ashton como título de uma apresentação que tinha como tema a ligação de RFID ao, na altura emergente, tópico da Internet [3] [18]. Atualmente, IoT representa conjuntos de objetos, serviços e tecnologias todos conectados à Internet [20] e tem o objetivo principal de formar um sistema inteligente de dispositivos autónomos que satisfaçam as todas as ligações dispositivo-dispositivo e utilizador-dispositivo (ver Figura 2.1) [24].

Um ambiente IoT típico tem vários atributos que o distinguem de sistemas tradicionais de *networking*, como por exemplo WSN. As redes IoT são constituídas por uma grande quantidade de dispositivos e tecnologias, tendo cada um capacidades e protocolos de comunicação únicos. Utilizando a Internet como estrutura de base, todos estes dispositivos e tecnologias conseguem manter uma comunicação viável e de baixa latência, estando tanto nas proximidades uns dos outros, como separados por grandes distâncias, procurando operar com o mínimo de consumos energéticos possíveis através de protocolos de comunicação eficientes e sendo os mais seguros e autónomos possíveis [8].

Figura 2.1: Esquema de uma versão simplificada de uma rede IoT

Um bom exemplo de um sistema IoT é o de uma *smart home*, como vista em [4], que consiste numa casa com uma rede autónoma de dispositivos que comunicam entre si de forma a melhorar o conforto do utilizador. Como se pode verificar no exemplo da Figura 2.2, a casa poderá ter uma variedade de dispositivos como:

- Sensores de temperatura e humidade, que podem indicar aos atuadores da unidade de ar condicionada da casa como deve proceder;
- Sensores de movimento e câmaras de videovigilância ligados ao sistema de vigilância da casa, que permitem observar os movimentos de residentes ou vigiar possíveis intrusos;
- Eletrodomésticos inteligentes, que através da partilha de informação como sua as taxas de utilização, podem servir para reduzir o consumo energético dos mesmos, reduzindo assim o impacto ecológico da casa.

Todos estes recursos, através da sua respetiva conexão à Internet, tornam-se facilmente observáveis, podendo ser modificados em qualquer lugar pelo residente através do uso de um telemóvel.

2.2 Arquitetura de uma Rede IoT

É do consenso de vários autores que o principal modelo utilizado para definir uma rede IoT consistem em três camadas distintas: **Application Layer** (em português, Camada de Aplicação), **Network Layer** (em português, Camada de Rede) **Perception Layer** (em português, Camada de Perceção). Cada camada representa um diferente conjunto tanto de tecnologias, serviços e dispositivos como de falhas e problemas [24] [12] [13] [5].

A Figura 2.3 representa um pequeno exemplo de uma rede IoT, de forma a mostrar quais podem ser algumas das tecnologias e dispositivos envolvidos em cada camada.

2.2.1 Camada de Aplicação

Esta camada é caracterizada por ter o nível arquitetural mais alto da rede e como tal, acomoda todas as interações com o utilizador final do sistema, servindo de ponto de

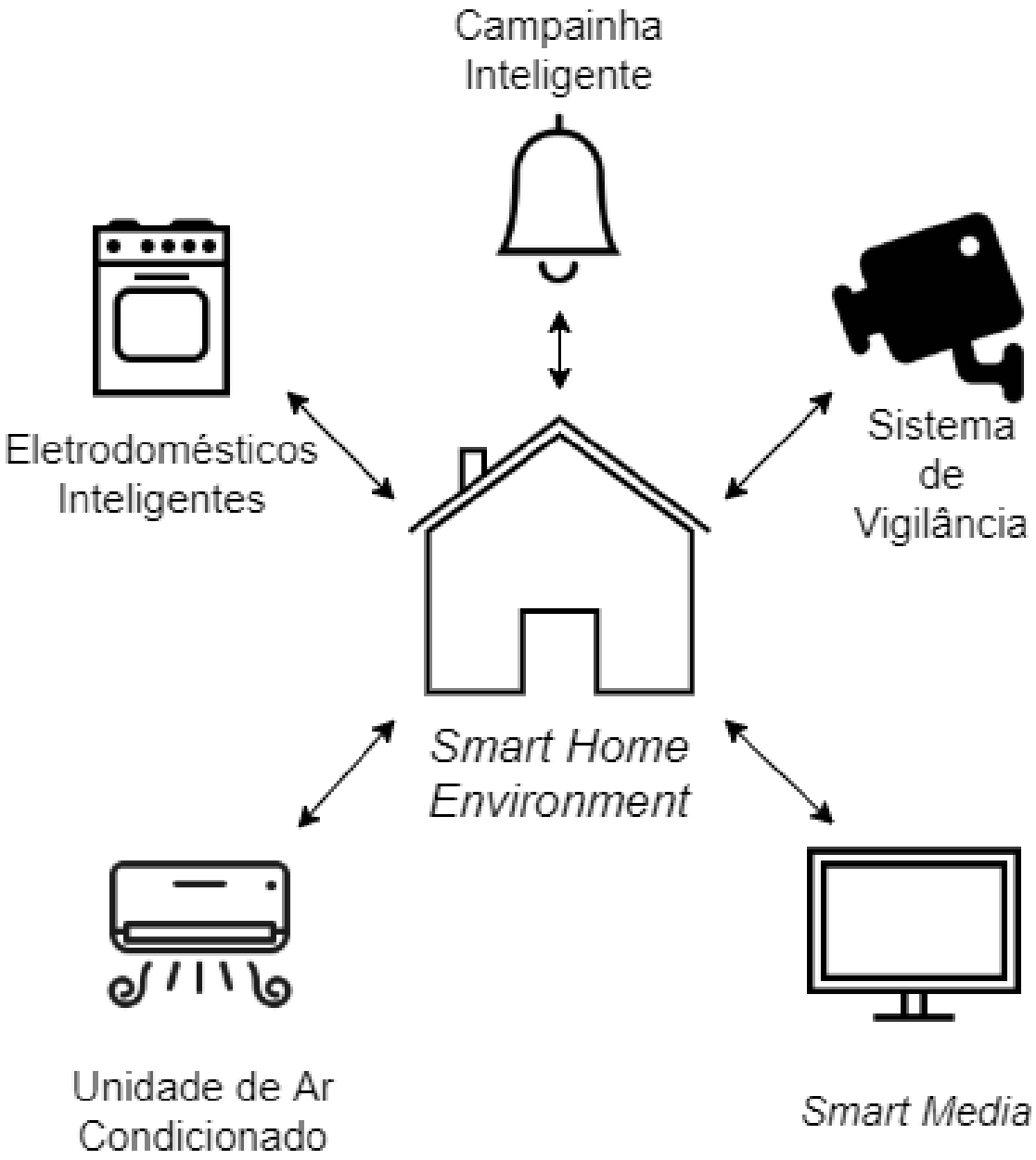

Figura 2.2: Exemplo de alguns possíveis dispositivos de uma *Smart Home*

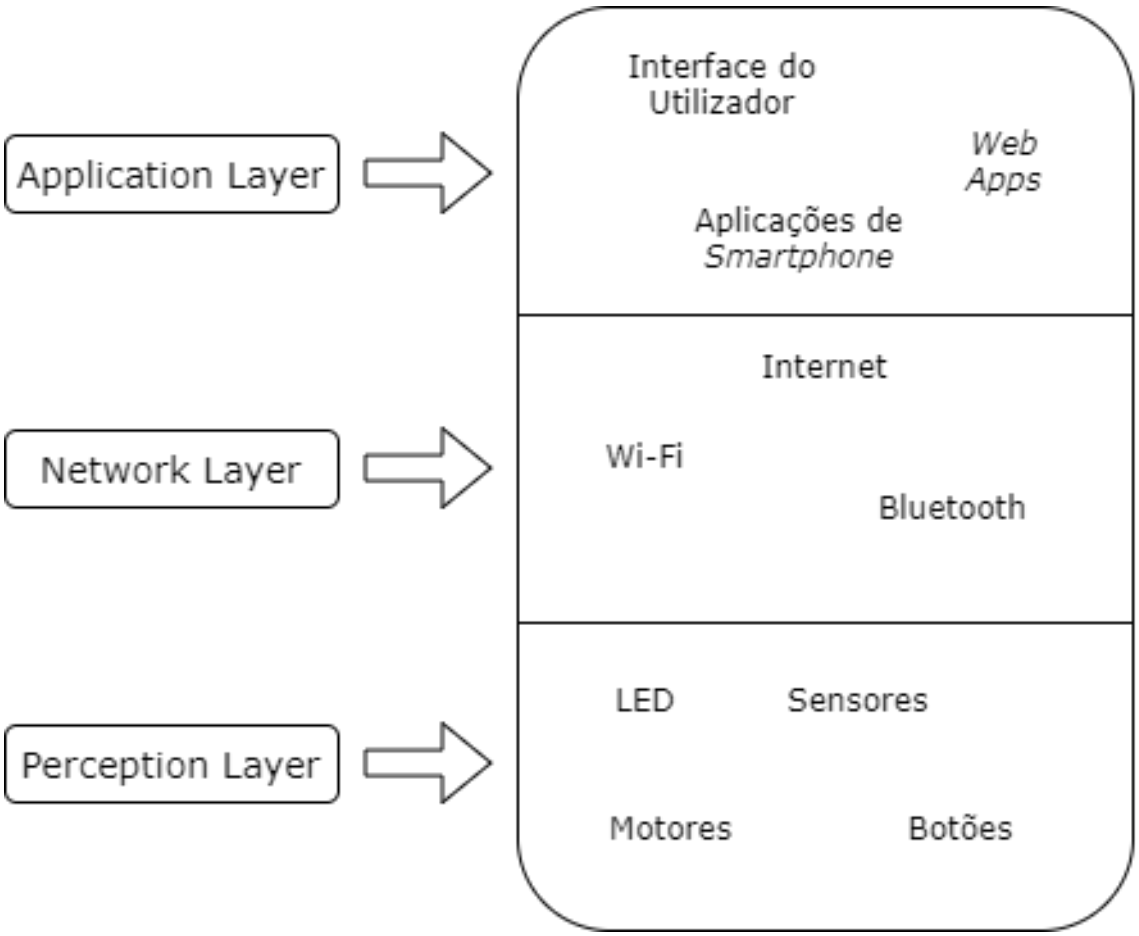

Figura 2.3: Exemplo de uma arquitetura IoT de três camadas [5]

acesso aos diversos serviços que a rede providencia, geralmente através de *web apps* ou mesmo aplicações para telefone [6].

Em termos de segurança, o principal foco desta camada é a partilha de informação, e como tal, a camada deve ter em especial atenção assuntos como a sua privacidade, controlo de acessos e fugas de informação [16].

2.2.2 Camada de Rede

Esta camada é responsável pelas conexões e o processamento entre as camadas do sistema através de tecnologias como Wi-Fi ou Bluetooth [6] [12].

Alguns autores, como [16] e [6] por exemplo, preferem separar esta camada em dois assuntos completamente distintos (ver Figura 2.4) :

1. O primeiro assunto é relativo aos processos de comunicação e *networking* principalmente usados com o nível arquitetural mais baixo da rede através da Internet e outras tecnologias de comunicação móvel de forma a recolher os dados gerados pela Camada de Perceção. O foco principal em termos de segurança deste tema é a

gestão de segurança da rede, ou seja, deve haver preocupação em ter mecanismos para evitar congestionamentos e protocolos de autenticação de dispositivos [16].

2. O segundo assunto é relativo ao processamento em si dos dados recolhidos pela primeira parte da Camada de Rede, o que vai permitir o funcionamento da atividade de alto-nível da rede presente na Camada de Aplicação [6]. Este serviço é geralmente providenciado por serviços em *cloud* e o seu principal foco de segurança é a proteção dos dados processados através de encriptação, autenticação, deteção de intrusos e possíveis distorções dos dados recolhidos e/ou processados [16].

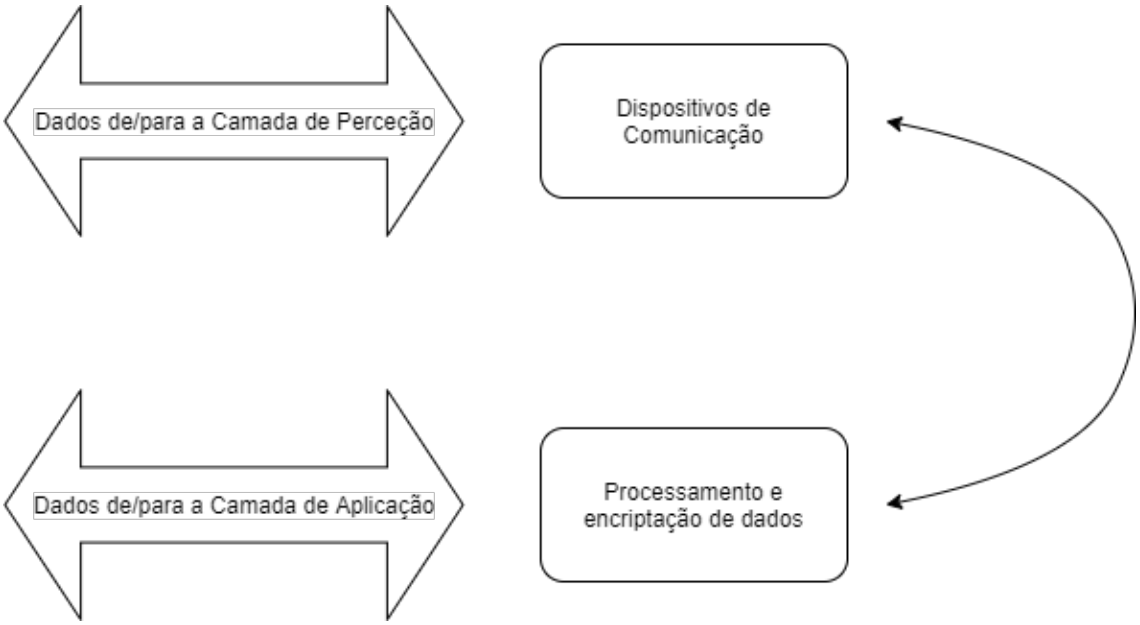

Figura 2.4: Diagrama do Método Operacional da Camada de Rede

2.2.3 Perception Layer

Finalmente, esta a terceira camada representa os sistemas que compõem o nível arquitetural mais baixo da rede, isto é, é caracterizada por dispositivos de pouca memória que observam ou provocam mudanças no ambiente em seu redor, como por exemplo sensores, motores ou botões [6].

O principal foco em termos de segurança nesta camada recai sobre os controladores locais de cada dispositivo e deve-se abordar assuntos como autenticação de sensores e análise comportamental [16].

Capítulo 3

Problemas de Segurança

Este Capítulo continua o Estado da Arte e aborda o assunto das vulnerabilidades de uma rede IoT relativamente à sua segurança e privacidade. Numa primeira parte é explorado as diferentes fraquezas dos sistemas IoT. De seguida é esclarecido as propriedades que um possível ataque que incida neste tipo de redes possa tomar, concluindo o capítulo com uma secção com alguns exemplos de ataques de diferentes níveis arquitecturais.

3.1 Tipos de Vulnerabilidades

Uma típica rede IoT é geralmente caracterizada pela monitorização e gestão de uma grande quantidade de diferentes serviços e dispositivos, que podendo estar geograficamente separados, têm de adquirir, processar e atuar de acordo com os dados recolhidos pelos seus nós finais [21]. Devido à sua heterogeneidade de recursos, os sistemas IoT devem ter uma preocupação significativa que não consegue ser respondida por soluções mais tradicionais, visto que estas não estão preparadas para um número grande tanto de dispositivos distintos como de diferentes plataformas computacionais, cada um com os seus problemas distintos relativamente a memória e limitações de processamento. Isto significa que uma rede IoT vai ter um grande número de pontos de acesso vulneráveis a acesso indesejado ou *exploits* [16].

Como visto em [23], as redes IoT têm cinco grandes tipos de problemas (ver Figura 3.1) que representam ameaças a um ambiente estável em termos segurança, privacidade e confidencialidade:

1. **Confiança do Ambiente** - Este tipo de vulnerabilidades refere-se a ameaças causadas no próprio *hardware* ou mesmo no ambiente físico circundante deste. Insere-se aqui ataques que podem variar desde alterar o funcionamento do dispositivo (apontar uma câmara apara um ângulo diferente, por exemplo), a forçar o dispositivo a

tomar ou registar ações através de movimentos ou sons, ou até mesmo a remoção/-vandalização do dito dispositivo.

2. **Autenticações** - Este assunto aborda métodos fracos ou mesmo inexistentes métodos de autenticação. Por exemplo, um grande número de acessórios Bluetooth utiliza passwords como *0-0-0-0* or *1-2-3-4* durante emparelhamentos, sem qualquer maneira fácil do utilizador mudar a sua palavra-passe, o que torna estes acessórios numa grande vulnerabilidade. Métodos fracos de autenticação podem também ser mais facilmente vítimas de ataques onde se tenta adivinhar a passe, quer por um número gigante de tentativas (*brute forcing*), quer por passes relacionadas (ataques de dicionário).

3. **Confiança da Rede** - Este tipo de problemas ocorre quando um dispositivo deposita confiança na rede a que está associado e não toma medidas de precaução como por exemplo métodos de autenticação. Isto significa que o dispositivo assume que o que quer que seja que esteja conectada à mesma rede é seguro também, o que resulta numa possível vulnerabilidade a possíveis intrusões que tenham conseguido acesso à rede.

4. **Privilégios** - Esta categoria de problemas alude ao abuso de aplicações e serviços que acedem a certos canais de informação para recolher mais dados do que os necessários para operar. Isto pode ser observado, por exemplo, em aplicações de *smartphones* que acedem a dispositivos e informação que, em uso regular, não deveriam ter acesso.

5. **Falhas de Implementação** - Finalmente, este tipo de problemas representa vulnerabilidades causadas por implementações defeituosas de *software*, como acessos garantidos através de *bugs*, *exploits* ou fugas de informação (*leaks* de credenciais, por exemplo).

Figura 3.1: Diagrama das Áreas Vulneráveis da Segurança de Redes IoT

3.2 Propriedades de Ataques

Devido à sua natureza complexa e heterogénea, uma rede IoT pode ser vulnerável a um grande número de diferentes ataques [16], pelo que é importante definir o que os definem e em que consistem.

De acordo com [23], um ataque que incide numa rede IoT pode ser classificado através de 6 atributos distintos (ver Figura 3.2) :

Figura 3.2: Diagrama das Características de um ataque a uma rede IoT

1. **Camada Alvo**. Como já foi visto na secção 2.2, uma rede IoT é composta por três diferentes *layers*, o que significa que um ataque irá sempre incidir sobre um dispositivo ou serviço presente numa ou mais camadas da rede. Portanto, a primeira propriedade de um ataque que se consegue identificar é a camada cujo ataque terá como alvo.

2. **Dispositivo Alvo**. O segundo atributo que é possível identificar é o dispositivo ou o conjunto destes que uma intervenção maliciosa tentará tirar partido. Um ataque pode ter como alvo tanto um único sensor como todo um grupo de dispositivos diferentes que, em conjunto, representam um sistema de uma rede, pelo que é importante definir qual o dispositivo vulnerável e não apenas a camada a que este pertence.

3. **Canal de Transmissão**. Outra propriedade de um ataque que se consegue observar é o meio de comunicação da invasão, ou seja, o canal usado para se ligar à rede alvo. Este canal pode assumir várias formas, quer seja como canais remotos como WiFi ou Bluetooth, como canais mais próximos de um dispositivo como toques físicos, gestos ou sinais acústicos.

4. **Consequências**. O quarto atributo observável é a consequência direta do ataque. Uma invasão de um sistema pode provocar vários resultados na rede que visa incidir, que podem variar desde *leaks* de informação sensível, a impedimento de acesso a

certos serviços da rede (DoS) ou até mesmo a controlo total ou parcial de certos elementos da rede.

5. **Tamanho**. A quinta propriedade que é possível identificar é o número de dispositivos ameaçados. Por exemplo, um intruso pode focar os seus recursos num único dispositivo como forma de prejudicar a rede, limitando o ataque a esse único dispositivo. Porém o mesmo intruso pode também utilizar esse dispositivo para influenciar outros dispositivos na rede a que estes estão ligados, aumentando assim a escala da invasão.

6. **Furtividade**. Finalmente, a sexta e última propriedade de uma invasão é a sua capacidade de permanecer em segredo antes, durante e depois da intervenção na rede. O utilizador da rede pode ficar completamente alheio ao ataque dirigido, ter conhecimento parcial ou mesmo total sobre a intrusão e as suas consequências, geralmente devido às mudanças (ou falta destas) no ambiente regular da rede causadas pela intrusão.

3.3 Exemplos de Ataques

Uma intrusão numa rede IoT pode tomar várias formas e utilizar múltiplos métodos para atingir os diferentes serviços do sistema alvo. Esta secção serve para enumerar alguns exemplos de ataques, elucidando o tipo de ataques que incidem nos diferentes níveis da arquitetura de uma rede IoT, como visto em [10].

Exemplos de Ataques

Alto Nível	Médio Nível	Baixo Nível
Software Exploits	Interferência de Conexões Ataques *Sinkhole*	Ataques de Interferências Ataques Sybil de Baixo Nível Deprivação de Sono

Figura 3.3: Lista de Exemplos de Ataques a uma rede IoT

3.3.1 Alto Nível

- ***Software Exploits***. De forma a ganhar acesso ao sistema, um intruso pode tentar aproveitar-se de qualquer tipo de falhas de *software* presente em aplicações de telemóvel ou nos serviços *web* que a rede possa utilizar, como a *cloud*. Este tipo de

ataque, em norma, aproveita-se de vulnerabilidades causadas por atualizações defeituosas.

3.3.2 Médio Nível

- **Interferência de Conexões**. Devido à necessidade de identificação única de cada dispositivo de uma rede IoT, o processo de comunicação para a atribuição dessa dita identificação é atacado. Isto pode não só fornecer a um intruso acesso a múltiplos tipos de informação sobre os dispositivos da rede como também impedir que circulem dados entre partes do sistema, dificultando assim a utilização de certos serviços na rede afetada.

- **Ataques *Sinkhole***. Este tipo de intrusão distingue-se pelo desvio de informação que circule na rede para um nó externo a esta, permitindo assim que um intruso aceda a informação confidencial e tome ações contra a rede baseadas nessa informação. Este tipo de ataques geralmente resultam em violações de privacidade e confidencialidade mas podem ser também usados para problemas mais severos como o bloqueio dos nós afetados.

3.3.3 Baixo Nível

- **Ataque de Interferência**. O objetivo deste tipo de ataque é a disrupção de certos dispositivos através da emissão de certos sinais (como certas radiofrequências, por exemplo) sem sentido, ordem ou protocolo percetíveis de maneira que este se confunda com os sinais que o dispositivo seria suposto de receber. Isto causa confusão na coleção e análise de informação nos dispositivos afetados, afetando o fluxo de informação destes, culminado assim num comportamento defeituoso do equipamento.

- **Ataque *Sybil* de Baixo Nível**. Este tipo de ataque pretende danificar os processos normais de funcionamento da rede através da conexão de nós maliciosos com falsas identidades. Ao gerar endereços MAC aleatórios, o intruso tenta disfarçar os seus nós como parte da rede de maneira a extinguir recursos desta, o que leva aos dispositivos legítimos terem o seu acesso à rede negado.

- **Depravação de Sono**. Este tipo de ataque procura a disrupção do funcionamento normal de dispositivos de energia limitada. Ao forçar certos dispositivos que estariam desenhados para funcionar de forma intermitente a correr procedimentos de forma interruptamente, um intruso pode forçar um dispositivo a desligar-se temporariamente, negando assim a disponibilização do serviço do dispositivo afetado.

Capítulo 4

Falhas Existentes e Trabalhos Relacionados

Este capítulo serve de conclusão ao Estado da Arte das redes IoT e serve para elucidar quais os falhas atuais em termos de segurança em redes IoT e que questões ainda precisam ser respondidas. É aqui também que se encontra uma pequena análise de alguns trabalhos relacionados com o tema da segurança e privacidade em redes IoT.

4.1 Problemas Atuais

Embora se baseie bastante em várias tecnologias estabelecidas [24], todo o conceito de IoT, e mais precisamente o seu aspeto da segurança, ainda está num estado infantil de investigação e pesquisa. Já existem alguns mecanismos de segurança implementados que reforçam segurança das redes IoT, mas ainda há um longo caminho a percorrer, pois o tema ainda tem alguns problemas operacionais para resolver [15].

Os maiores problemas que todo o conceito de segurança de redes IoT são:

- **Restrições de Recursos**. Uma das maiores dificuldades que os sistemas de segurança de ambientes IoT atualmente atravessam é a limitação de recursos de dispositivos e tecnologias envolvidos nas próprias redes, o que torna soluções de privacidade e segurança tradicionais bastante difíceis de implementar [10]. Por exemplo, a camada arquitetural mais baixa das redes, que tipicamente é formada por vários dispositivos de poucos recursos e baixa memória, impede a implementação de soluções de segurança existentes mas sofisticadas, como criptografia altamente otimizada por exemplo [8].

- **Múltiplos Pontos Únicos de Falha**. Um sistema grande e diverso como uma rede IoT infelizmente significa também uma lista grande e diversa de *Single Points Of Failure*. [10]. Um único dispositivo ou serviço vulnerável pode ser a causa de uma

tremenda ameaça a todo o sistema [7], como por exemplo uma campainha inteligente integrada num sistema IoT aplicado a uma *smart home* pode ser explorada de maneira a expor as credenciais de autenticação WiFi da rede colocando assim todo o sistema em perigo, como visto em [11].

- **Falta de Protocolos**. Ter um conjunto diverso e heterogéneo de dispositivos, serviços e tecnologias é um problema em si quando não existe qualquer tipo de protocolo para o conjunto num todo. Nota-se uma grande falta de um *standard* que englobe todas as camadas da arquitetura de uma rede IoT, desde aos já referidos dispositivos com restrições de recursos até servidores de alta capacidade [10].

Em suma, existe uma falta de investigação relativamente tanto a *hardware* e *software* dedicados a IoT seguros e nota-se a ausência de protocolos orientados para IoT [15]. É necessário encontrar métodos para proteger os dispositivos da Camada de Perceção de forma a evitar falhas dos mesmos através de algoritmos de confiança, encriptação e/ou autenticação tendo sempre o objectivo de melhorar a segurança, a privacidade, a integridade e a confidencialidade da rede [19].

4.2 Trabalhos Relacionados

Atualmente já existe uma grande quantidade de trabalhos relacionados com o conceito de IoT. Mais especificamente, IoT aplicado a *Smart Home Environments* facilmente se encontra tanto trabalhos mais simples como pequenos tutoriais em páginas da Internet, como projetos científicos que procuram levar o conceito de IoT mais longe.

4.2.1 Diversas Aplicações de Redes IoT

No trabalho visto em [2], os autores relacionam o conceito de redes IoT com o conceito de análise de *Big Data*, sendo o foco do trabalho o processamento eficientemente de uma enorme quantidade de dados gerados por unidades de aquecimento, ventilação e ar condicionado de modo a apresentar estes dados de uma maneira simples e legível ao consumidor final.

Também é possível observar o potencial de redes IoT noutras indústrias como a agricultura. Como visto em [17], uma rede IoT pode servir para facilitar a monitorização de colheitas, fazendo com que baste algo como um *smartphone* para conhecer instantaneamente parâmetros como a humidade do solo, a temperatura ambiente ou até mesmo se a planta em questão se encontra doente.

4.2.2 Segurança em Redes IoT

Direcionado para o tema de segurança em IoT, existem também trabalhos mais complexos como o trabalho visto em [14], que procura reforçar os processos de autenticação na *Network Layer* através da adaptação de algoritmos complexos e pesados como *BlockChain* aos dispositivos de recursos reduzidos de uma rede IoT, aplicada também ao exemplo de um *Smart Home Environment*.

Outra abordagem às questões de segurança e privacidade que rodeiam o conceito de redes IoT é a utilização de algoritmos de *Machine Learning*, como visto em [22]. Através de métodos como, por exemplo, deteção de *malware* baseada em aprendizagem ou autenticação *learning-based*, os autores exploram as vantagens que algoritmos deste tipo têm quando aplicados a um conceito de uma rede IoT, assim como os principais desafios que estes ainda precisam de responder.

Capítulo 5

Descrição Informal dos Sistemas Implementados

Este capítulo serve para enquadrar o trabalho prático realizado nesta dissertação, descrevendo os objectivos pretendidos para o desenvolvimento dos sistemas de redes IoT implementados e explicitando os requisitos necessários para cada um deles.

5.1 Contexto de Desenvolvimento

Este trabalho teve como objetivo a análise de condições de segurança, privacidade e confidencialidade presente nos recursos atualmente disponíveis para redes IoT. Para tal construiu-se um protótipo de uma rede IoT aplicada a um ambiente de *Smart Home*, isto é, um ambiente de domótica onde a rede visa automatizar os recursos e dispositivos de uma casa de maneira a melhorar as condições de vida do seu habitante.

De modo a chegar a um protótipo robusto, fez-se uma abordagem ao tema em duas partes:

- Na abordagem inicial criou-se uma rede o mais simples possível através de recursos e *software* de uso mais geral (em oposição a recursos e software dedicados a IoT), de modo a facilitar a observação de fraquezas e vulnerabilidades de um sistema IoT para que mais tarde, numa segunda rede mais complexa, se pudesse ter um foco total em questões de privacidade e segurança. Esta primeira rede é composta por um dispositivo, que é responsável pela transmissão constante da temperatura recolhida no local onde se encontram; por um serviço de *cloud* que regista os dados recolhidos e por uma aplicação de *smartphone* que mostra ao utilizador a temperatura atualmente lida pelo dispositivo e outras temperaturas lidas anteriormente, como é possível ver na Figura 5.1.

- Numa segunda abordagem, após uma análise das condições da primeira rede, procurou-se construir uma rede mais complexa, tanto em dispositivos como em recursos e *software*, tendo sempre em vista a máxima segurança do sistema. Para tal optou-se não só por usar recursos e software especialmente direcionados para IoT, mas também como mais dispositivos com propósitos diferentes, tentando responder às questões de segurança e privacidade levantadas pela primeira rede. Esta segunda rede é, à semelhança também da primeira, composta por uma aplicação que se liga ao utilizador aos seus dispositivos através de um serviço de *cloud*, porém conta com 5 dispositivos de propósitos diferentes (medir temperatura, deteção de intrusos interior e exterior, controlo de sistema de iluminação e acessório Bluetooth), como é possível observar na Figura 5.2.

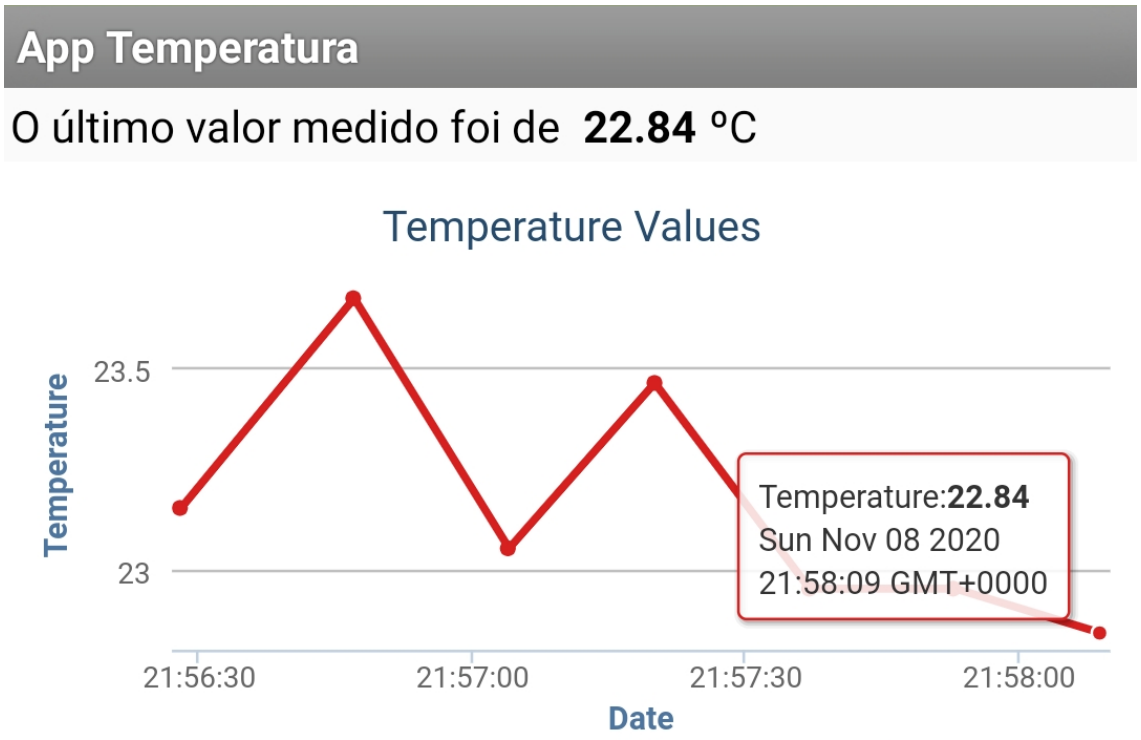

Figura 5.1: Aplicação da Primeira Rede Implementada em Funcionamento

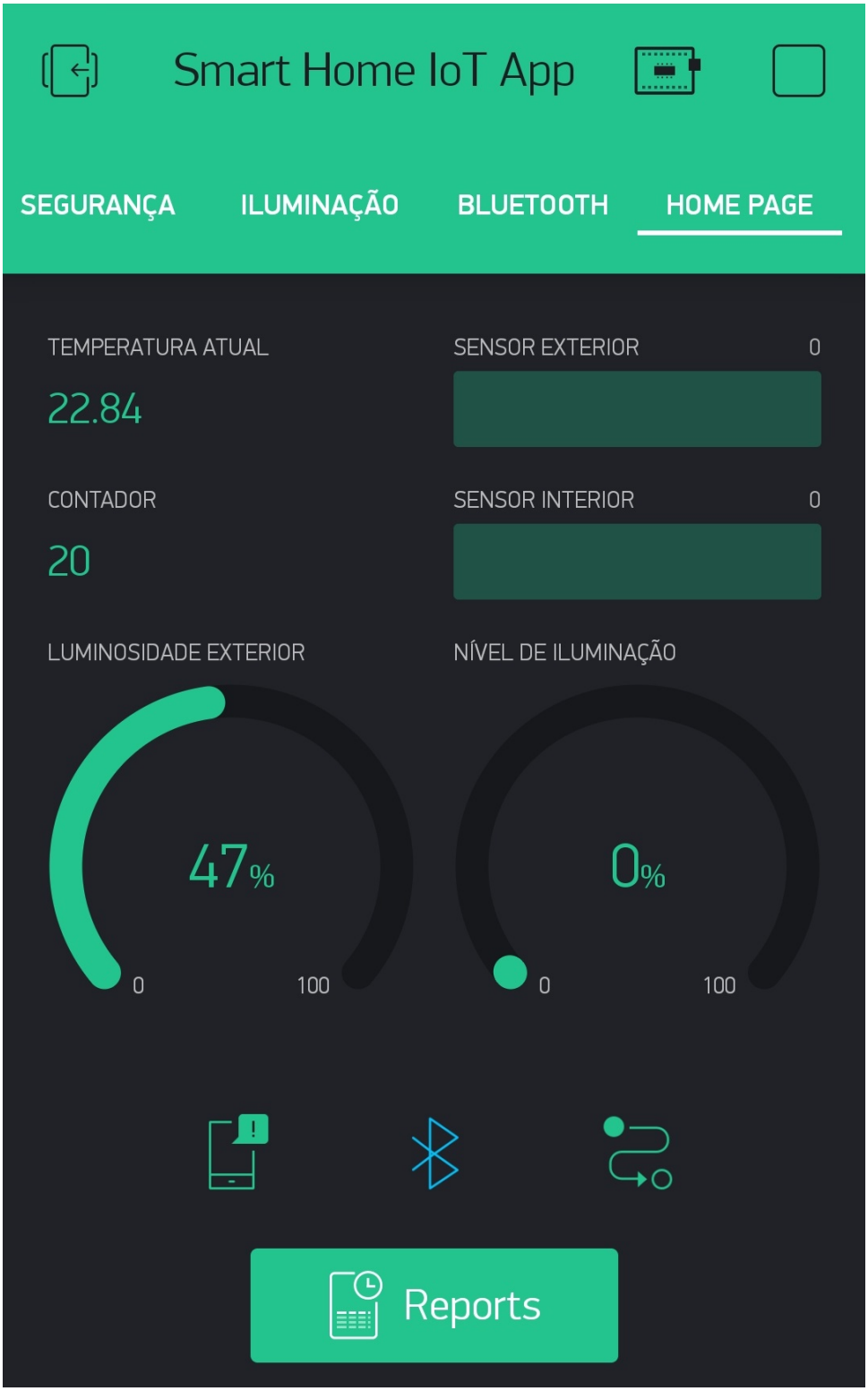

Figura 5.2: Página Inicial da Aplicação da Segunda Rede Implementada

5.2 Requisitos Funcionais e Não Funcionais

Tendo em conta a descrição das redes feita na secção anterior, serve a presente secção para descrever os requisitos funcionais e não funcionais de cada um dos sistemas desenvolvidos.

5.2.1 Primeira Rede

Como já foi referido, a primeira rede IoT implementada conta com uma aplicação de *smartphone* que simplifica o acesso do utilizador à rede, um dispositivo de leitura de temperatura que vai enviar os seus dados constantemente e um *Cloud Service* que é responsável pelo processamento e transmissão de informação entre o dispositivo previamente referido e a aplicação do utilizador.

Como tal, os requisitos funcionais desta primeira rede estão descritos na Tabela 5.1.

Tabela 5.1: Requisitos Funcionais da Primeira Rede Implementada

Nome	Descrição
Leitura de Temperatura	Capacidade do dispositivo da rede registar a temperatura ambiente do local onde se encontra num dado instante.
Conexão WiFi	Capacidade do dispositivo ligar-se a uma rede WiFi, nomeadamente através de um *router*.
Envio de Informação	Capacidade do dispositivo enviar a informação recolhida pelo sensor de temperatura para o serviço de *cloud*.
Recepção de Informação	Capacidade do serviço *cloud* receber a informação enviada pelo dispositivo.
Processamento de Informação	Capacidade do serviço *cloud* processar a informação recebida, nomeadamente todas as leituras de temperatura registadas pelo dispositivo.
Conexão Móvel	Capacidade da aplicação aceder aos recursos da rede através de qualquer conexão à Internet.
Consulta da Temperatura Atual	Capacidade da aplicação mostrar ao utilizador qual a temperatura mais recente medida pelo dispositivo.
Consulta de Temperaturas Passadas	Capacidade da aplicação mostrar ao utilizador quais foram as temperaturas medidas pelo dispositivo desde o início do funcionamento deste.

Como é possível observar, os requisitos funcionais descrevem as funcionalidades e serviços base da rede. De modo a descrever os atributos e restrições do sistema, assim como as funções para garantir segurança e privacidade de modo a responder a problemas como os identificados no Estado da Arte desta dissertação, temos os requisitos não funcionais desta rede na Tabela 5.2.

Tabela 5.2: Requisitos Não Funcionais da Primeira Rede Implementada

Nome	Descrição
Autenticação do Utilizador	Capacidade do sistema autenticar o telefone do utilizador de maneira a garantir que este é um agente de confiança da rede e não um intruso.
Autenticação de Dispositivo	Capacidade da rede autenticar o dispositivo de registo de temperaturas de maneira a garantir que este é um agente de confiança e não um intruso.
Taxa de Recepção de Informação	Capacidade da rede garantir que o utilizador consegue receber informações vindas do dispositivo a uma taxa de uma leitura de temperatura a cada 30 segundos no máximo.
Taxa de Envio de Informação	Capacidade do dispositivo garantir o envio de informação a uma taxa de uma leitura de temperatura a cada 30 segundos no mínimo.
Verificação de Estado do Dispositivo	Capacidade da rede conseguir identificar se o dispositivo se encontra conectado ou se desconectou da rede e se essa informação é transmitida ao utilizador.
Análise de Informação Anómala	Capacidade da rede analisar e identificar valores fora do funcionamento regular do dispositivo e se a existência dessas anomalias é transmitida ao utilizador.

5.2.2 Segunda Rede

Como também já foi referido a segunda rede resultou da evolução natural de uma análise feita à primeira rede. Este segundo ambiente conta na mesma com uma aplicação de *smartphone* para o utilizador interagir com a rede e um servidor para registar e processar dados. Porém, esta segunda rede conta com mais do que apenas um sensor de temperatura, pois é constituída também por dois sensores de movimento, um sensor de luz aliado a um LED e um acessório Bluetooth.

Por ser uma evolução da primeira rede, a segunda rede vai naturalmente ter alguns requisitos funcionais semelhantes, pois o modo de funcionamento de ambas é idêntico. Porém como é mais complexa, por ter mais dispositivos, também terá mais requisitos para acomodar as funções destes. Estes requisitos funcionais foram divididos, de acordo com a camada da rede que abordam, nas três tabelas seguintes.

Tabela 5.3: Requisitos Funcionais dos Dispositivos da Segunda Rede Implementada

Nome	Descrição
Leitura de Temperatura	Capacidade de um dispositivo da rede registar a temperatura ambiente do local onde se encontra num dado instante.
Deteção de Movimento Exterior	Capacidade de um dispositivo da rede detetar movimentos no espaço exterior da habitação num dado instante.
Deteção de Movimento Interior	Capacidade de um dispositivo da rede detetar movimentos no espaço interior da habitação num dado instante.
Deteção de Luminosidade	Capacidade de um dispositivo da rede detetar o nível de iluminação do espaço exterior.
Iluminação Automática	Capacidade de um dispositivo da rede ligar ou desligar iluminações da habitação de acordo com o nível de iluminação exterior.
Iluminação Manual	Capacidade de um dispositivo da rede ligar ou desligar iluminações da habitação de acordo com a vontade do utilizador.
Atualização de Contador	Capacidade de um dispositivo da rede mostrar o número atual de um contador presente na aplicação da rede.

Como se pode observar, a Tabela 5.3, representa os requisitos funcionais relacionados com os dispositivos da rede, que nesta segunda implementação conta com mais funcionalidades para além de apenas registar temperaturas.

A Tabela 5.4 descreve os requisitos funcionais relacionados com servidores e outros serviços de Internet da rede, como os diferentes tipos de conexões entre os múltiplos agentes do sistema assim também como o devido processamento de informação recebida e enviada pelos mesmos.

Tabela 5.4: Requisitos Funcionais dos Serviços de Internet da Segunda Rede Implementada

Nome	Descrição
Conexão Bluetooth	Capacidade de um dispositivo se ligar e transmitir informação por uma ligação Bluetooth ao telemóvel do utilizador.
Conexão WiFi	Capacidade de um dispositivo da rede se ligar a uma rede WiFi, nomeadamente através de um *router*.
Envio de Informação	Capacidade de um dispositivo da rede enviar a informação recolhida e/ou gerado pelos seus sensores e atuadores para os servidores de Internet.
Recepção de Informação	Capacidade dos servidores de Internet receberem as informações enviadas pelos dispositivos.
Processamento de Informação	Capacidade dos servidores de Internet processarem a informação recebida tanto dos dispositivos como da *app* do utilizador.
Conexão Móvel	Capacidade da aplicação aceder aos recursos da rede através de qualquer conexão à Internet.

Os requisitos funcionais relacionados com a aplicação de *smartphone* da rede, que é o serviço da rede que permite ao utilizador dispor dos recursos da mesma, podem ser consultados na Tabela 5.5.

Tabela 5.5: Requisitos Funcionais da *App* da Segunda Rede Implementada

Nome	Descrição
Consulta da Temperatura Atual	Capacidade da aplicação mostrar ao utilizador qual a temperatura mais recente medida pelo dispositivo.
Consulta de Temperaturas Passadas	Capacidade da aplicação mostrar ao utilizador quais foram as temperaturas medidas pelo dispositivo desde o início do funcionamento deste.
Consulta de Segurança Atual	Capacidade da aplicação mostrar ao utilizador o estado dos sensores de movimento exterior ou interior, ou seja se alguns destes deteta algum movimento, em tempo real.
Consulta de Segurança Passada	Capacidade da aplicação mostrar ao utilizador valores passados dos sensores de movimento exterior ou interior, ou seja se alguns destes detetou algum movimento, desde o início do seu funcionamento.
Consulta de Iluminação	Capacidade da aplicação mostrar ao utilizador as condições de luminosidade exterior atual e o estado atual da iluminação da habitação, isto é se as luzes se encontram ativas ou não.
Controlo de Iluminação	Capacidade da aplicação alternar o modo de iluminação de automático para manual e vice-versa.
Controlo de Contador	Capacidade da aplicação incrementar ou decrementar um número de maneira que este número seja constantemente enviado por uma ligação Bluetooth.

Tendo em conta que a preocupação em termos de segurança e privacidade é a mesma para as duas redes, os requisitos não funcionais também são bastante semelhantes entre as redes, como é possível observar na Tabela 5.6.

Tabela 5.6: Requisitos Não Funcionais da Segunda Rede Implementada

Nome	Descrição
Autenticação do Utilizador	Capacidade do sistema autenticar o telefone do utilizador de maneira a garantir que este é um agente de confiança da rede e não um intruso.
Autenticação de Dispositivo	Capacidade da rede autenticar cada um dos dispositivos da Camada de Aplicação de maneira a garantir que este são agentes de confiança e não um intrusos.
Trocas de Informação Periódica da Aplicação	Capacidade da rede garantir que o utilizador consegue aceder aos recursos disponibilizados pelos diversos dispositivos da rede que se caracterizam por enviar informação periodicamente, com um máximo de 1 segundo de intervalo entre envio ou receção de informação.
Trocas de Informação Espontânea da Aplicação	Capacidade da rede garantir que o utilizador consegue aceder aos recursos disponibilizados pelos diversos dispositivos da rede que se caracterizam por enviar informação de forma espontânea, sem nenhuma ordem em particular.
Trocas de Informação Periódica de um Dispositivo	Capacidade de um dispositivo garantir o envio e receção de informação periodicamente, com um máximo de 1 segundo de intervalo entre envio ou receção de informação.
Troca de Informação Espontânea de um Dispositivo	Capacidade de um dispositivo garantir o envio e receção de informação de de forma espontânea, sem nenhuma ordem em particular.
Verificação de Estado do Dispositivo	Capacidade da rede conseguir identificar se o dispositivo se encontra conectado ou se desconectou da rede e se essa informação é transmitida ao utilizador.
Análise de Informação Anómala	Capacidade do sistema analisar e identificar valores fora do funcionamento regular do dispositivo e se a existência dessas anomalias é transmitida ao utilizador.
Atuação devido a Informação Anómala	Capacidade do sistema realizar algum tipo de resposta à identificação de valores fora do funcionamento regular do dispositivo.

Capítulo 6

Estrutura dos Sistemas Implementados

Neste capítulo é possível observar uma descrição detalhada da implementação das redes IoT, isto é, quais os recursos, serviços e *hardware* utilizados para a sua construção, assim como uma explicação do funcionamento normal dos sistemas desenvolvidos.

6.1 Tecnologia Comum a Ambas Implementações

Como já foi referido, de maneira a chegar a um protótipo de rede IoT o mais robusto possível em termos de segurança e privacidade, foi realizada uma abordagem que consistia, numa primeira fase, na implementação de uma pequena rede IoT aplicada a uma *smart home*.

Esta primeira iteração, através de recursos de uso mais variado em vez de tecnologias e *software* dedicado a ambientes IoT, tinha como objectivo evidenciar possíveis problemas de segurança para que, numa segunda fase, se construísse uma rede com atributos de segurança o mais forte possíveis, através de tecnologias e *software* especialmente dedicados a redes IoT assim como das lições retiradas da primeira implementação.

Isto significa que a segunda rede resulta dum desenvolvimento natural das questões levantadas pela primeira rede, e como tal, ambas as redes utilizam alguns dos mesmos recursos, como cabos, *smartphones* e *routers*.

Porém, o principal recurso comum nas duas implementações é a placa NodeMCU, que é o *hardware* que possibilitou estabelecer uma ligação dos diversos sensores e atuadores das duas redes à Internet. Esta placa, cujo *pinout* é possível observar na Figura 6.1, tem o microcontrolador ESP8266 como peça central, o que permite assim o suporte nativo de ligações a redes WiFi através de uma antena integrada na própria placa. O *firmware* desta placa é suportado pela linguagem LUA, podendo ser também programada através do IDE Arduino, que foi o método utilizado neste trabalho. As principais características desta

placa são:

- Alimentação de 5 V via porta micro-USB;
- Processador que opera até 160 MHz;
- Botões de Reset e Flash;
- Memória RAM de 96 kb;
- Memória ROM de 64 kb;
- Memória flash de 4Mb;
- Pin analógico com uma resolução de 10 bits;
- 11 Pins digitais de GPIO com funções de PWM e I2C;
- Dimensões de 49 mm de comprimento, 24.5 mm de largura e 13 mm de altura.

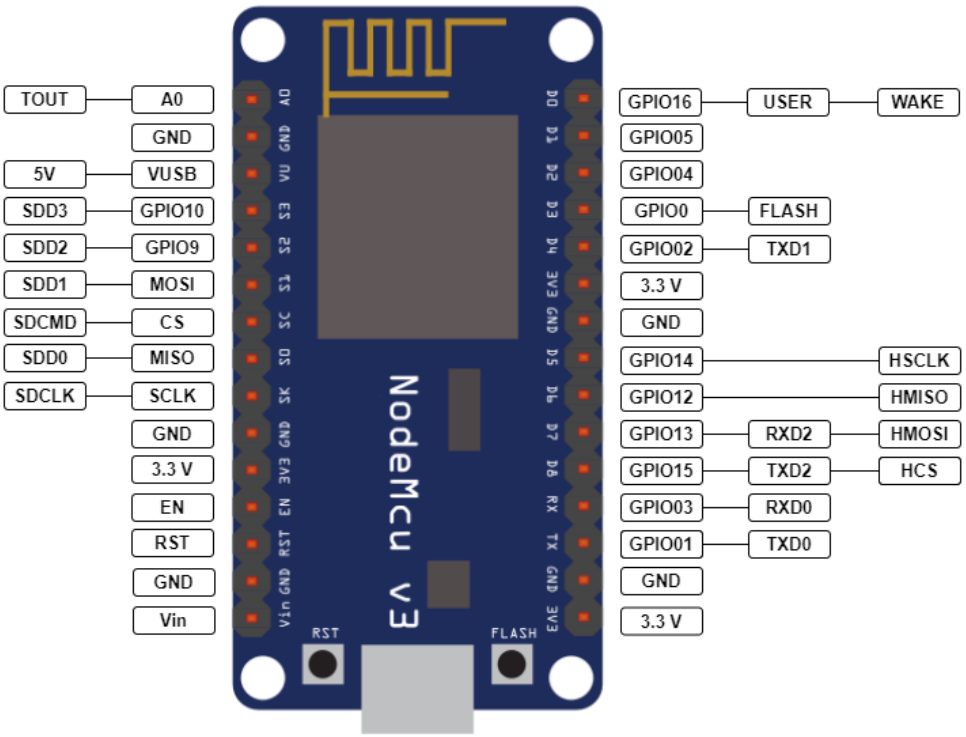

Figura 6.1: *Pinout* da placa NodeMCU

6.2 Funcionamento do Primeiro Sistema

Esta secção serve para explicar o funcionamento da cada componente da primeira rede construída, elucidando quais os tipos de recursos e serviços utilizados em cada camada do sistema.

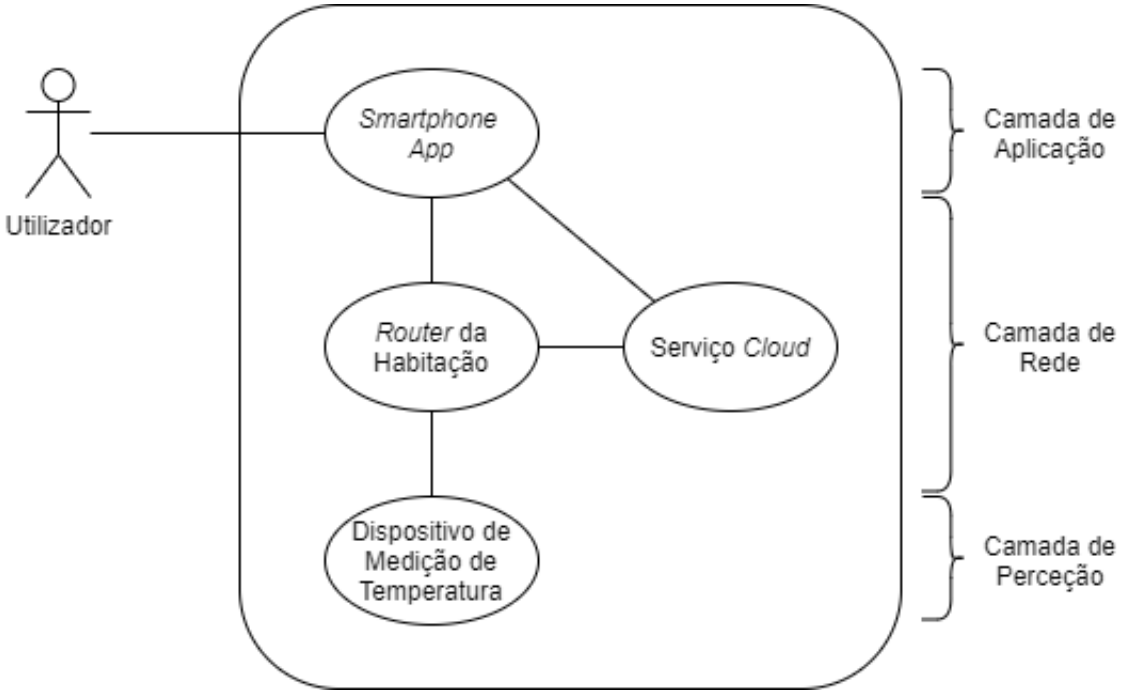

Figura 6.2: Diagrama de Caso de Uso da Primeira Rede Implementada

Como se pode observar na Figura 6.2, o funcionamento deste sistema é simples. Na Camada de Aplicação podemos encontrar uma *app* que, ligando-se à rede através de qualquer ligação à Internet, tem o propósito simples de mostrar a temperatura mais recente medida e providenciar uma consulta do gráfico com o registo das leituras de temperatura anteriormente obtidas.

A Camada de Rede divide-se em duas partes. A primeira é composta pelas ligações relacionadas com transmissão de dados entre camadas e pelas conexões dos serviços das mesmas com o serviço *cloud* através de ligações *Web*. A segunda parte consiste no serviço *cloud* que recolhe os dados enviados, processa-os de maneira a apresentar a temperatura mais recente e um gráfico com valores anteriores e procede ao envio desses dados para a aplicação.

Finalmente a Camada de Percepção conta com um dispositivo responsável apenas pela leitura da temperatura do local onde se encontra e pela transmissão dessa informação para o serviço *cloud*.

6.2.1 Implementação da Camada de Aplicação

A Camada de Aplicação deste primeiro sistema é composta por uma aplicação cujo funcionamento, como já foi referido, consiste em mostrar ao utilizador a medição de temperatura mais recente assim como os valores previamente registados através de um gráfico dos valores medidos em função do tempo da medição.

Esta aplicação foi construída através do software "MIT App Inventor", que é um ambiente de desenvolvimento de *smartphone apps* integrado numa aplicação *web* que prima pela simplicidade e na construção das ditas *apps*, através de uma abordagem de *drag-and-drop* de elementos como botões ou caixas de texto por exemplo, e de blocos de funções pertinentes aos mesmos de maneira a evitar o uso de código no processo de desenvolvimento da *app*. Este serviço conta também com uma aplicação de suporte para telemóvel que permite ao utilizador, em segundos, correr, testar e programar aplicação no seu próprio telefone.

Como se pode ver na Figura 5.1, é possível identificar dois objetos dos quais a aplicação é composta:

- O primeiro objeto é um conjunto de caixas de texto que vão, periodicamente, fazer solicitações HTTP (*Hypertext Transfer Protocol*, ou em português, Protocolo de Transferência de Hipertexto) aos serviços *cloud* da rede, de modo a receberem o valor da medição de temperatura mais recente. Estes pedidos funcionam na base de uma hiperligação e têm como objetivo solicitar informações presentes numa página com o registo dos valores de temperatura, do qual é retirado o mais recente, sendo este apresentado ao utilizador. A estrutura da hiperligação utilizada no pedido será explicada de uma maneira mais aprofundada na secção seguinte devido à hiperligação conter elementos que fazem parte dos processos de autenticação da rede.

- O segundo objeto que é possível identificar é o gráfico dos valores medidos em função da sua data de medição. Este gráfico consiste apenas numa pequena visualização de uma página *web* adaptada à estrutura da aplicação através de um URL (*Uniform Resource Locator, ou em português, Localizador Uniforme de Recursos*) disponibilizada pelos serviços *cloud*. Isto é, este gráfico é resultante do processamento de dados que ocorre no serviço *cloud* e a aplicação é apenas responsável pela preparação de uma visualização deste.

Em ambos estes casos, a aplicação, independentemente da periodicidade do registo de valores de temperaturas, tem a capacidade de atualizar a sua informação a uma taxa mínima de um segundo por pedido.

6.2.2 Implementação da Camada de Rede

Como foi visto nos capítulos iniciais, uma Camada de Rede de um sistema IoT tem duas grandes responsabilidades: os processos de comunicação entre os dispositivos e serviços da rede e o processamento da informação transmitida por estes. Como os sistemas desenvolvidos não são exceção, a Camada de Rede deste primeiro sistema também se encarrega das ligações entre dispositivos, serviços e aplicações ao servidor *cloud* que vai efetuar o processamento das informações recolhidas.

Portanto, de forma a que os recursos das outras duas camadas consigam aceder aos serviços *cloud* da rede através de ligações pela Internet, a rede impõe duas situações diferentes:

- A primeira diz respeito à ligação do dispositivo de leitura da temperatura. Esta distingue-se por ser exclusivamente com o *router* da habitação, que através do ISP (*Internet Service Provider*, ou em português Fornecedor de Acesso à Internet) realiza a conexão do dito dispositivo à Internet, permitindo assim o envio de informação para o servidor *cloud*.

- A segunda diz respeito à conexão da aplicação do utilizador, que pode ser tanto uma ligação ao mesmo *router* da rede ao qual o dispositivo de leitura da temperatura se conecta , como a qualquer outra rede exterior à habitação que providencie uma ligação à *Web* como até qualquer outra capacidade de Internet Móvel que o telefone possua.

Em relação à recolha e processamento de dados, utilizou-se o serviço ThingSpeak, que é uma plataforma analítica para ambientes IoT e especializa-se na agregação, visualização e análise de dados num serviço *cloud*, sendo que também oferece visualizações numa página *web*, podendo ser facilmente consultada, como por exemplo através de um *browser*.

Como se vê na Figura 6.3, a plataforma divide-se em canais (ou *channels*, em inglês), que representam o conjunto de sistemas associados a uma rede total, que por sua vez de dividem em campos (ou *fields*, em inglês) que são responsáveis pela representação da informação recolhida por um dispositivo ou pela apresentação do mesmo. Em suma, um canal engloba a rede e os campos englobam os dispositivos presentes na rede. A versão da licença utilizada foi a licença grátis do *software*, que disponibiliza um máximo de 4 canais com 10 campos cada um e um limite de intervalo de envio/recepção de mensagens de 15 segundos por canal.

No caso desta rede, é utilizado um canal dedicado às operações relacionadas com a medição de temperatura, que utiliza apenas um campo para medição da temperatura mais recente. A a página *web* que se pode ver na Figura6.3 representa o canal utilizado para esta rede e apresenta duas visualizações, um gráfico das temperaturas medidas em função da hora que o servidor as recebeu e o valor da temperatura mais recente recebida.

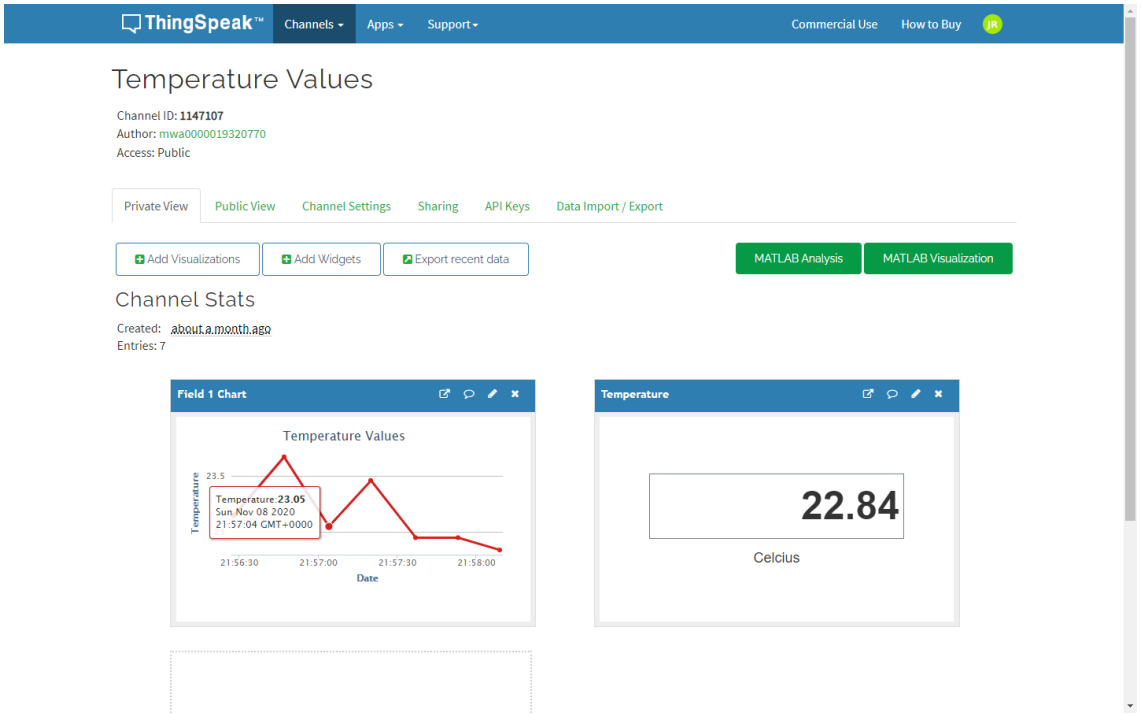

Figura 6.3: Visualização da Página *Web* do Serviço *Cloud* da Primeira Rede

Um canal deste serviço possui um identificador único ("Channel ID") de 7 algarismos e pode ter visualizações públicas, que qualquer pessoa ou recurso pode aceder através do URL da visualização disponibilizado pelo serviço, ou visualizações privadas, que apenas podem ser acedidas através de chaves únicas de 16 números e letras cada, para pedidos de escrita ou de leitura dos campos de cada canal. No caso desta rede, a visualização do gráfico é pública, para que seja possível apresentá-la na aplicação e a visualização da temperatura mais recente é privada.

Este servidor *cloud* vai servir essencialmente três pedidos: o pedido de apresentação do gráfico feito pela *app*, que é realizado através de um *URL* disponibilizado pelo serviço ao tornar a visualização do gráfico pública; o pedido de requisição da temperatura realizado pela aplicação e o pedido de envio de informação realizado pelo dispositivo de medição de temperatura.

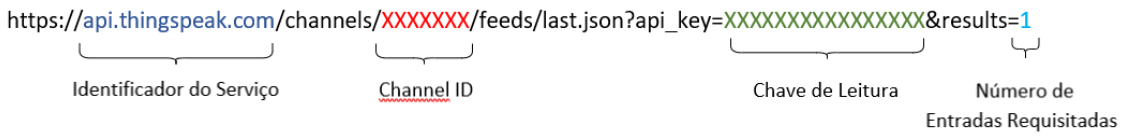

Figura 6.4: Estrutura da Hiperligação Utilizada pela Aplicação para Requisitar a Temperatura mais Recente ao Servidor *Cloud*

A hiperligação utilizada para pedido de recepção de informação efetuado pela aplicação ao servidor tem a estrutura que se pode consultar através do exemplo da Figura 6.4, onde a azul escuro está o identificador do serviço ao qual o pedido é direcionado, que é o serviço ThingSpeak, a vermelho o identificador único do canal, a verde a chave de

leitura para que a aplicação consiga obter informação da visualização privada e a azul claro está o número de entradas que se pretende obter, que no caso desta rede é uma (a mais recente).

A resposta do servidor à aplicação consiste numa estrutura de 3 parâmetros para cada entrada pedida, começando pela mais recente até à mais antiga, consoante o número de entradas pedidas. Os parâmetros das estruturas das entradas são: data e hora a que o servidor recebeu a leitura; número da entrada, que funciona como um contador do número total de leituras efetuadas; e valor do campo, que é onde se transmite o valor da medição da temperatura. É este último campo que a aplicação vai retirar de forma a apresentar ao utilizador a temperatura mais recente.

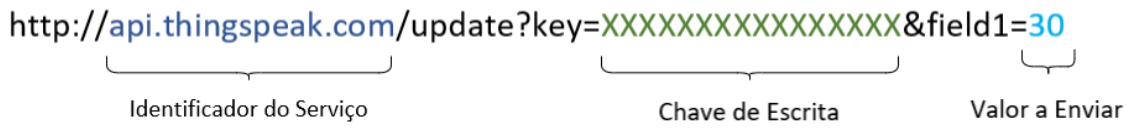

Figura 6.5: Estrutura da Hiperligação Utilizada pelo Dispositivo para Enviar as Medições de Temperatura ao Servidor *Cloud*

A hiperligação utilizada para pedido de envio de informação efetuado pelo dispositivo de medição de temperatura ao servidor tem a estrutura que se pode consultar através do exemplo da Figura 6.5, onde a azul escuro está o identificador de serviço, a verde está a chave de escrita necessária para o dispositivo se autenticar na rede como uma fonte de informação segura e a azul claro está o valor da medição que o dispositivo pretende enviar.

6.2.3 Implementação da Camada de Aplicação

Como é possível observar pela Figura 6.6, o dispositivo responsável pela medição e envio da temperatura para o servidor *cloud* da rede é composto pela ligação de uma placa NodeMCU com um sensor do tipo Ks0022.

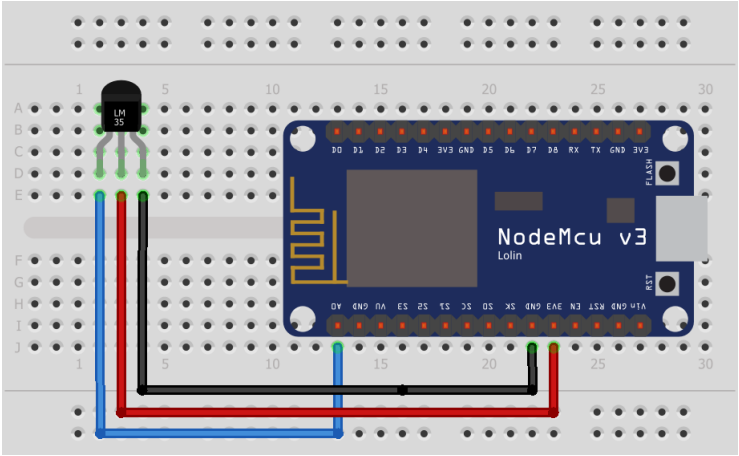

Figura 6.6: Ilustração do *Hardware* Utilizado no Dispositivo de Medição de Temperatura

É importante notar que, para efeitos de ilustração, optou-se pela utilização de um termístor LM35, que é um elemento equivalente ao sensor utilizado no dispositivo, pois não existe nenhum tipo de ilustração gratuita deste no *software* que foi usado para fazer as ilustrações do *hardware*.

```
//setup da placa NodeMCU
void setup()
{
  Serial.begin(9600);//inicio do monitor serie

  //método para garantir que a placa nao se encontra conectada a nenhuma rede
  WiFi.disconnect();
  delay(2000);

  //Ligacao ao hotspot WiFi da rede
  Serial.println("Connecting to the network");
  WiFi.begin("Nome_da_Rede","Password_da_Rede");
  while ((!(WiFi.status() == WL_CONNECTED))){
    delay(300);
  }
  Serial.println("Connected");
}
```

Listagem 6.1: Excerto de Código Utilizado Para Conectar o Dispositivo a uma Rede WiFi

O sensor Ks0022 baseia-se no termístor LM35 para detetar a temperatura ambiente do local onde se encontra, tendo uma gama de entrada de 0 a 100 graus Celsius. Este termístor é um semicondutor do tipo PTC (*Positive Temperature Coefficient*, ou em português, Coeficiente de Temperatura Positivo), isto é, consoante a variação da temperatura ambiente, verifica-se uma variação proporcional da resistência do termístor, e por consequente uma variação da tensão aos polos deste. É esta tensão que o sensor transmite para a placa NodeMCU, que por sua vez vai converter esse número num valor de temperatura, como é possível ver no excerto de código programado na placa presente na Listagem 6.2.

```
//Medicao da tensao do termistor e conversao para Celsius

  unsigned int sensorInput = analogRead(A0);//leitura do valor registado pelo termístor
  unsigned int temperatura = (500 * val) /1024;//conversao para temperatura

  //Impressao no Monitor Serie
  Serial.print("A temperatura é: ");
  Serial.println(temperatura);
```

Listagem 6.2: Excerto de Código Utilizado Para Registar a Temperatura Ambiente

Tendo um valor de uma medição da temperatura, o dispositivo deve proceder ao envio desta aos recursos *cloud* da rede. Para tal, o dispositivo, através das capacidades da placa

NodeMCU começa por se conectar ao *router* da habitação de forma a estabelecer uma ligação à Internet. Esta ligação, cujo excerto de código se pode observar na Listagem 6.1, é realizado no ciclo de *setup* da placa de forma a garantir que esta é a primeira ação que a placa toma ao iniciar o seu funcionamento.

```
1  //envio do valor da variavel "temperatura" para a cloud
2    if (client.connect("api.thingspeak.com",80))
3    {
4      request_string = "http://api.thingspeak.com/update?";
5      request_string += "key=";
6      request_string += "XXXXXXXXXXXXXXXX";
7      request_string += "&";
8      request_string += "field1";
9      request_string += "=";
10     request_string += temperatura;
11
12     http.begin(request_string);
13     http.GET();
14     http.end();
15     }
```

Listagem 6.3: Excerto de Código Utilizado Para Enviar um Medição de Temperatura para a Camada de Rede

De seguida, a placa vai proceder ao envio da informação para o serviço *cloud* da rede. Como se pode ver no excerto de código da Listagem 6.3, o dispositivo, caso o servidor esteja disponível para receber informação, vai formar uma *string* com a hiperligação com a estrutura observada na Figura 6.5, que se utiliza para o envio de informação para o serviço ThingSpeak.

É importante notar que, embora a capacidade do dispositivo transmitir leituras de temperatura seja de uma leitura por centenas de milissegundos, como já foi referido, o servidor ThingSpeak, devido ao tipo de licença adquirida, apenas suporta a troca de uma mensagem cada 15 segundos. Por este facto, foi esta a cadência de leitura implementada neste dispositivo.

6.3 Implementação e Funcionamento do Segundo Sistema

Esta secção serve para explicar o funcionamento de cada componente da segunda rede construída, indicando para cada camada, os recursos, serviços e *hardware* que a constituem.

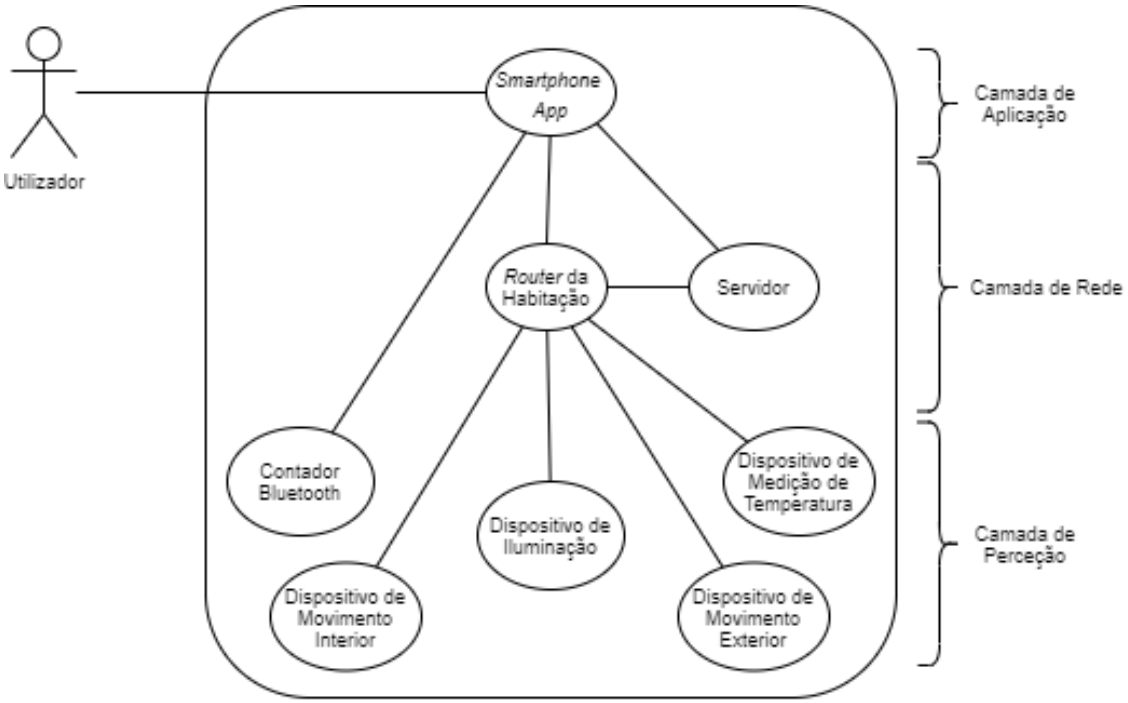

Figura 6.7: Diagrama de Caso de Uso da Segunda Rede Implementada

Como é possível verificar observando a Figura 6.7, o funcionamento desta segunda iteração do protótipo de rede IoT é muito semelhante ao funcionamento da rede anterior. Na *Application Layer* continua-se a ter uma *app* que, tanto por uma ligação ao *hotspot* WiFi presente na habitação, como por qualquer outra ligação à Internet, tem o propósito de apresentar os recursos da rede ao utilizador, que nesta iteração são mais do que apenas um dispositivo de medição de temperatura.

A Camada de Rede deste segundo sistema é igualmente composta por duas partes que desempenham funções semelhantes às da primeira rede, sendo a primeira parte também responsável pela comunicação entre dispositivos e serviços presentes nas diversas camadas da rede, maioritariamente através da Internet e segunda parte pela recolha, processamento e envio de informação relevante tanto ao utilizador como aos dispositivos da rede. As diferenças da *Network Layer* desta segunda rede é que as ligações entre camadas não são feitas exclusivamente pela Internet pois existe não só um dispositivo com capacidades de ligação Bluetooth como o servidor pode ser local em vez de em *cloud*.

A *Perception Layer* desta segunda rede é a camada mais diferente em comparação com o sistema anterior, pois possui 5 dispositivos que compõem 4 módulos diferentes. O primeiro módulo é composto por um dispositivo de medição da temperatura ambiente,

em muito semelhante ao dispositivo da rede anterior. O segundo módulo é composto por dois dispositivos que são responsáveis por deteção de movimento, sendo um dispositivo dedicado a movimentos no exterior da habitação e o outro a movimento interior. O terceiro módulo representa um sistema de controlo de iluminação que é responsável por registar o nível de luminosidade exterior e ligar a iluminação da habitação consoante os valores de iluminação observados. Finalmente, o quarto módulo da Camada de Percepção é o dispositivo de ligação Bluetooth, que é responsável apenas por mostrar um contador que o utilizador pode incrementar ou decrementar ao seu desejo através da *app*.

6.3.1 Implementação da Camada de Aplicação

A *Application Layer* deste segundo sistema construído é, tal como na primeira rede, composto por uma aplicação de smartphone. Porém esta rede é composta por mais dispositivos, portanto a aplicação tem que, obrigatoriamente, mostrar mais recursos da rede ao utilizador.

A *app* foi desenvolvida através do software "Blynk", que é um ambiente de recursos especialmente focado em redes IoT que fornece um construtor de aplicações numa aplicação para telemóveis e todo um suporte de recolha, processamento e envio de informação através de um servidor dedicado à rede, que pode ser local (no computador do utilizador da rede) ou em *cloud*.

O construtor de aplicações, que está disponível para os sistemas operativos Android e iOS, permite o desenvolvimento de aplicações direcionadas ao controlo e gestão de recursos de redes IoT e funciona à base da introdução de blocos e *widgets*, cada um com a sua função específica. Esta aplicação utiliza também o conceito de pinos virtuais, que funcionam como unidades de registo de modo a facilitar o trânsito de informação entre as variáveis do código programado no *hardware* e os ditos *widgets* aplicação.

A versão da licença utilizada foi a licença grátis do *software*, que permite a ligação de um máximo de 5 dispositivos de *hardware* ao servidor *cloud* e disponibiliza 256 pinos virtuais para o uso da rede. Importante referir também que não existe qualquer tipo de restrição, tanto na aplicação como no servidor, à cadência de trocas de informação nem ao número de mensagens enviadas ou recebidas, quer sejam periódicas como no caso do módulo de registo de temperatura, por exemplo, quer sejam esporádicas como no caso do módulo de deteção de movimentos.

De modo a implementar uma maneira do utilizador poder aceder todos os recursos que os dispositivos da rede oferecem, a aplicação é composta por cinco separadores.

O primeiro separador, **Home Page**, que pode ser observado na Figura 5.2, é a página inicial da aplicação e é responsável por resumir os recursos essenciais da rede de modo a que o utilizador possa fazer uma consulta rápida dos valores medidos pelos diversos sensores da rede sem que seja necessário utilizar separadores respetivos. Para isso, conta com dois *widgets* de *display* de valores numéricos (para as medições de temperatura e para o contador), *widgets* de *display* binário (para os sensores de movimento) e dois *widgets*

de percentagem (para os níveis de luminosidade exterior e iluminação interior). Existem também três blocos neste separador que, embora não apresentem nenhum valor vindo de sensores, complementam a rede de outra maneira. Estes blocos servem para possibilitar notificações de estado dos dispositivos mesmo se a aplicação estiver a funcionar em segundo plano, conectar dispositivos Bluetooth à aplicação, criar eventos a partir de valores lidos pelos sensores (um exemplo de evento é enviar notificação ao utilizador se o sensor de temperatura registar um valor demasiado alto) e enviar um ficheiro CSV (*Comma-Separated Values*) com um registo dos valores de todos os pins virtuais do sistema ao longo do período de funcionamento da rede.

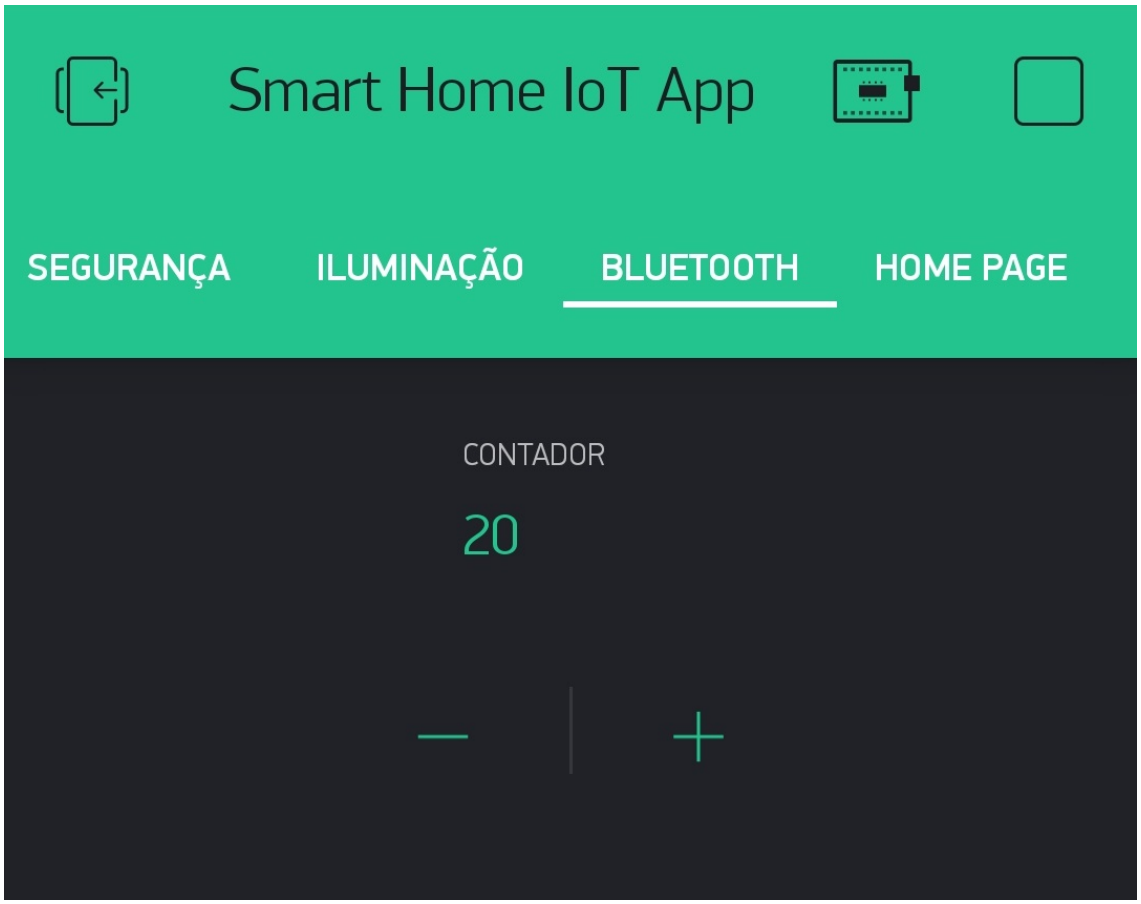

Figura 6.8: Separador "Bluetooth"da Aplicação da Segunda Rede Implementada

O segundo separador, **Bluetooth**, que pode ser observado na Figura 6.8 e representa a interação do utilizador com o módulo Bluetooth através de dois *widgets*. O primeiro *widget* mostra o valor corrente do pino virtual que representa o número que irá ser incrementado ou decrementado, mediante a vontade do utilizador, através do segundo *widget*, que é composto por dois botões com os símbolos de adição e subtração. Ao ser alterado o valor do pino virtual que corresponde ao contador, a aplicação envia o valor deste para o dispositivo Bluetooth para este cumprir a sua função e para o servidor da rede para efeitos de registo de valores, como o ficheiro CSV.

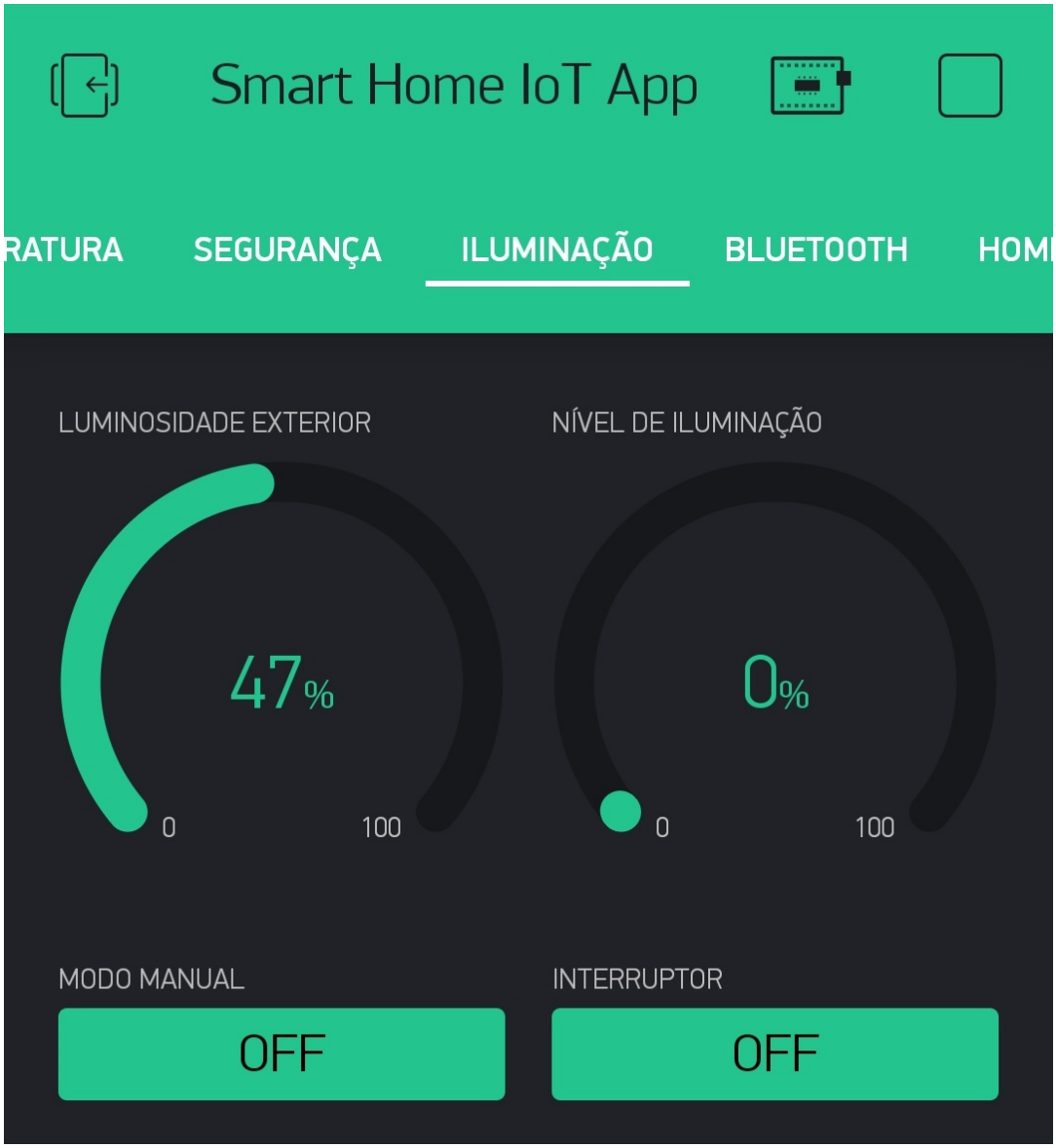

Figura 6.9: Separador "Iluminação"da Aplicação da Segunda Rede Implementada

O terceiro separador da aplicação, **Iluminação**, que pode ser observado na Figura 6.9, é a página que representa a interação com o módulo de controlo de iluminação da rede. Este separador conta com quatro *widgets*, sendo que dois mostram o nível de intensidade da luminosidade medida no exterior da habitação e o nível da iluminação da habitação e os outros dois servem para o utilizador desativar o modo automático e controlar a iluminação manualmente. Esta secção da *app* utiliza um pino virtual para cada bloco, sendo que vai ler os níveis de luminosidade e de iluminação através do seu pino virtual respetivo e vai enviar para os outros dois os valores necessários para alterar o estado dos dois botões (0 e 1 para representarem OFF e ON, respetivamente).

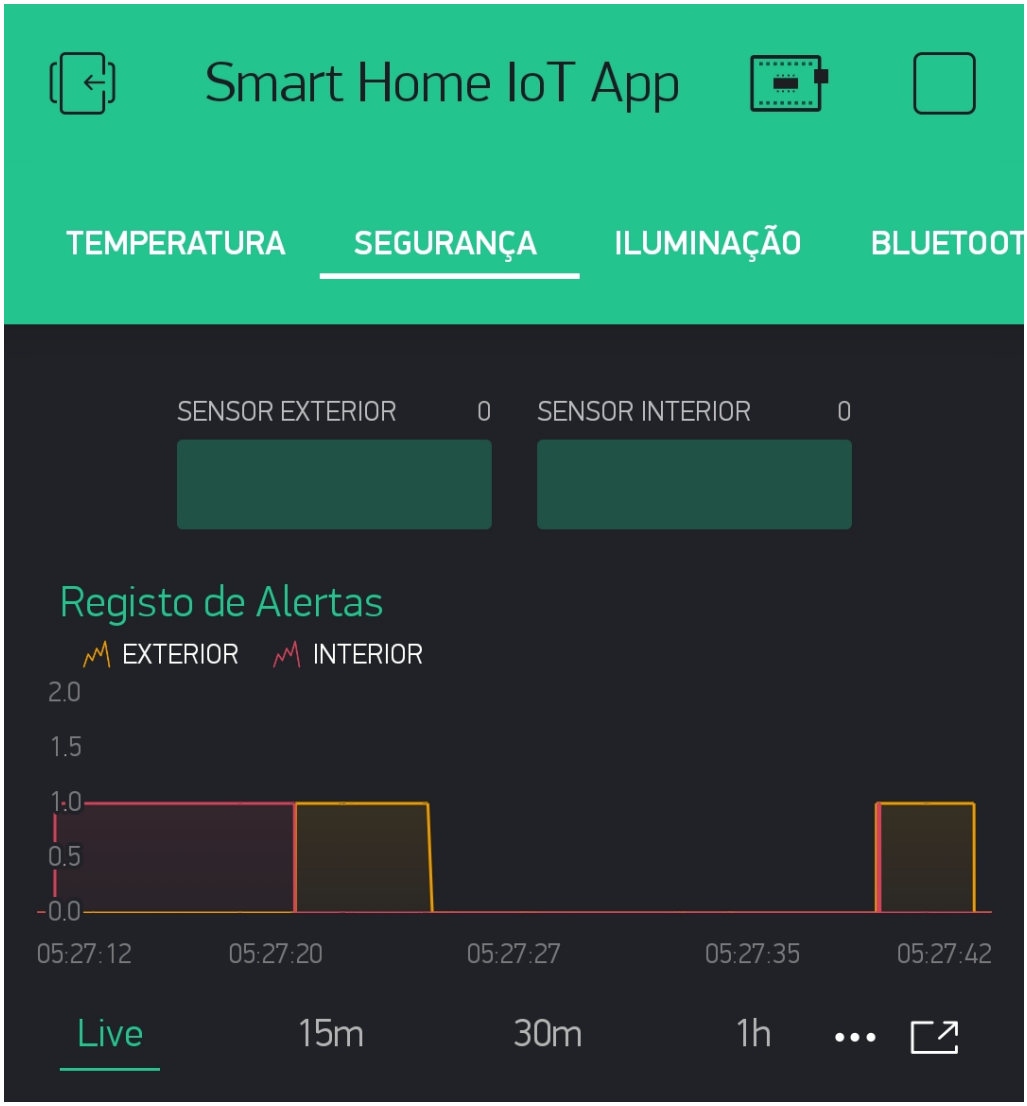

Figura 6.10: Separador "Segurança"da Aplicação da Segunda Rede Implementada

O quarto separador da aplicação, **Segurança**, que pode ser observado na Figura 6.10, é responsável pela interface gráfica do módulo de segurança e é composto por três blocos. Os dois primeiros blocos são simples blocos de *display* que se iluminam quando um dos dispositivos, interior ou exterior, deteta movimento e envia para o respetivo pino virtual. Já o terceiro bloco é um gráfico do historial de ambos os sensores em função do tempo decorrido desde o inicio de serviço dos dispositivos e pode adaptar-se a uma visualização dos valores em direto ou em diferentes períodos de tempo (na figura estão representados três períodos apenas mas o gráfico tem um alcance possível de minutos, horas, dias, semanas e até mesmo meses). É importante observar que os sinais dos dois dispositivos são diferentes devido à natureza do módulo de segurança, que como tem de obter sinais imediatos dos dispositivos de deteção de movimentos, requer uma visualização o mais aproximada possível dos sinais lidos pelos sensores utilizados nos dispositivos (a razão da natureza dos sinais lidos pelos sensores é explicada em mais detalhe na Subsecção 6.3.4).

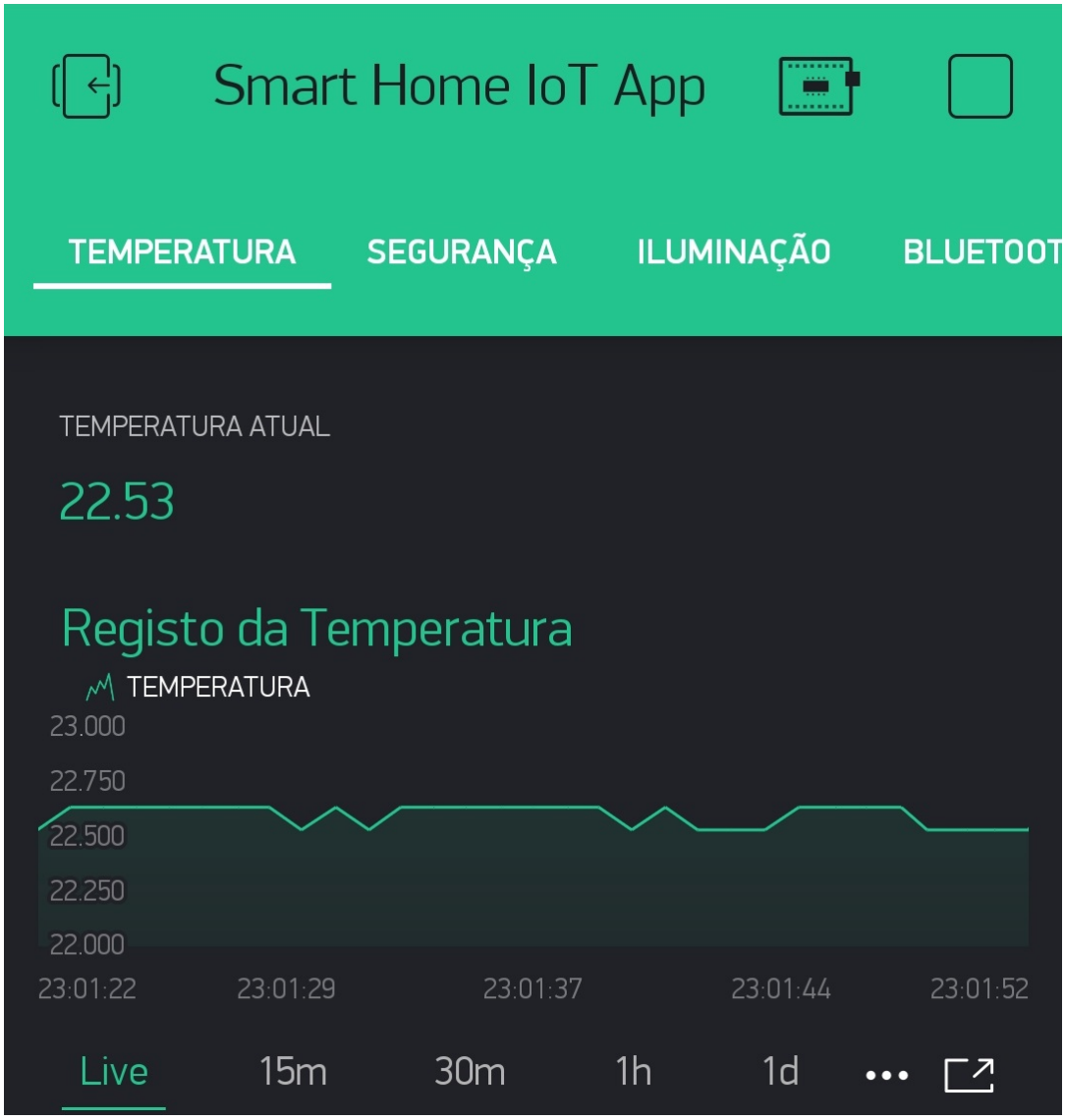

Figura 6.11: Separador "Temperatura"da Aplicação da Segunda Rede Implementada

Finalmente, o quinto separador da aplicação, **Temperatura**, que pode ser observado na Figura 6.11, é responsável pela apresentação dos valores observados pelo dispositivo de medição de temperatura. Este módulo tem um funcionamento bastante semelhante à aplicação do primeiro sistema, no sentido em que apresenta o valor mais recentemente medido pelo dispositivo, assim como os valores anteriormente medidos, na forma de um gráfico. Este separador utiliza apenas a informação do pino virtual dedicado ao valor da temperatura medida pelo dispositivo não só para apresentar ao utilizador a temperatura atual através de um bloco de *display* de valores, como também para fornecer informação ao bloco do gráfico, que vai fazer o desenho destes valores em relação ao tempo que foram enviados para o pino virtual. De notar que o dispositivo envia medições de temperatura para o pino virtual a cada segundo, portanto estes blocos irão ser atualizados a cada segundo também.

6.3.2 Implementação da Camada de Rede

Tal como na primeira iteração, a *Network Layer* deste segundo sistema segue o modelo tradicional de redes IoT, ou seja, esta camada é responsável pelo trânsito de informação entre dispositivos, serviços e outros recursos da rede e pelos métodos de autenticação e processamento dos diversos tipos de dados que a rede irá gerar.

O serviço utilizado para responder a estes assuntos foi o servidor que está integrado com os serviços Blynk, que se caracteriza pelo uso de pinos virtuais, que são nada mais do que variáveis localizadas no servidor direcionadas para tomar valores de informação como se fossem pinos presentes nos sensores dos diversos dispositivos de *hardware*. Este serviço, que está intrinsecamente ligado ao criador de aplicações, é responsável por toda a recolha, processamento e envio de informação entre serviços da rede, por todos os processos de autenticação do utilizador e dos dispositivos assim como de estabelecer três tipos de ligações:

- O primeiro tipo diz respeito às ligações entre os dispositivos da Camada de Perceção e o *hotspot* WiFi oferecido pelo *router* da habitação, que liga os dispositivos de *hardware* à Internet através do serviço realizado pelo ISP, possibilitando assim a ligação dos dispositivos com o servidor.

- O segundo tipo diz respeito à ligação da aplicação com o servidor, que à semelhança da primeira rede, permita tanto ser através de uma ligação direta ao *router* da rede, como através de qualquer rede com capacidade de ligação à Internet.

- O terceiro tipo de ligação diz respeito à ligação de acessórios sem capacidade de ligação à Internet, como é o caso do módulo Bluetooth, ao servidor da rede. Esta ligação tem de ser indireta, ou seja, o dispositivo comunica apenas com o telemóvel que por sua vez se liga ao servidor de modo a partilhar a informação relacionada com o dispositivo, como métodos de autenticação ou troca de valores de e para pinos virtuais.

Em termos de autenticação, o servidor utiliza dois métodos para garantir que, tanto a aplicação, como os dispositivos são seguros. No caso do utilizador, o servidor requer a inscrição de uma conta de utilizador para iniciar sessão na rede e posteriormente, aceder aos seus recursos. Esta conta necessita de um *email* válido e de uma palavra-passe de 8 a 16 caracteres, podendo estes serem letras ou números. No caso dos dispositivos, a sua autenticação é realizada através de um *token* de 32 caracteres, podendo estes ser letras caracteres ou símbolos. Este *token* é enviado para o *email* da conta associada para depois ser programado no *hardware* e deve ser único para cada dispositivo da rede.

Tal como a primeira rede, este servidor também disponibiliza informação através de pedidos HTTP, tanto para leitura e escrita de pinos virtuais como para verificar se um dispositivo se encontra conectado ao servidor. A estrutura das hiperligações utilizadas nestes pedidos encontram-se exemplificadas na Figura 6.12.

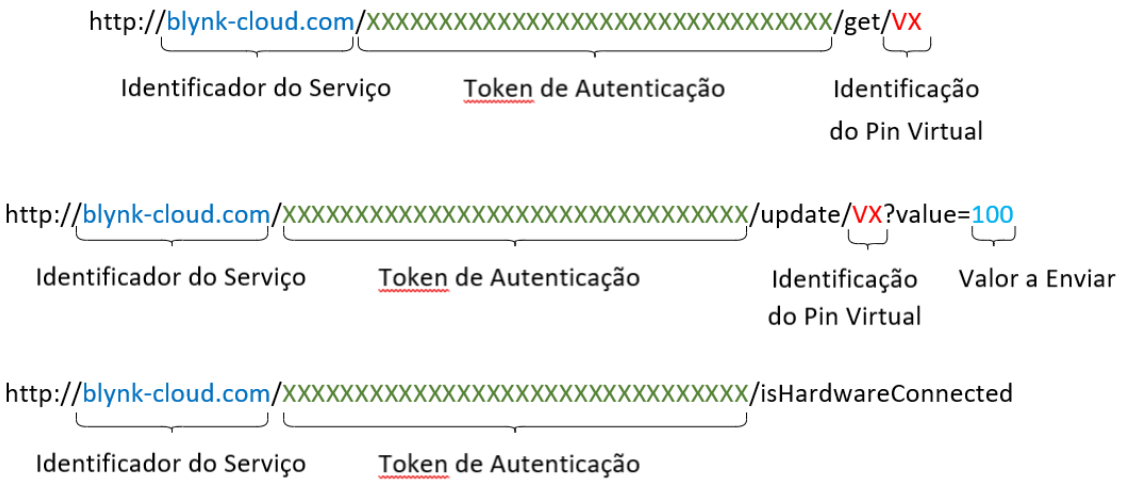

Figura 6.12: Estrutura das Hiperligações Utilizadas para Leitura e para Escrita do Pinos Virtuais e Verificação de Conexão dos Dispositivos, respetivamente

Importante referir que, tal como a aplicação, o servidor não restringe o número nem frequência de trocas de informação entre os dispositivos da rede, quer esta seja periódica ou esporádica. Portanto a cadência e a quantidade de envio e receção de dados são aquelas que os dispositivos necessitarem de impor devido ao seu funcionamento.

Finalmente, o serviço Blynk oferece a possibilidade de o utilizador montar um servidor local, com um funcionamento exatamente igual ao servidor *cloud*, num computador com capacidade de se tornar num *hotspot* WiFi e usar este em vez do servidor *cloud* disponibilizado. Porém este servidor local não possui qualquer tipo de ligação à internet. De modo a ligar o servidor local à Internet (pois a ligação à Internet é o tema central de uma rede IoT) teria de se utilizar um *software* de terceiros para fazer redireccionamento de portas, o que iria fazer com que qualquer entidade conseguisse aceder à rede através do IP da mesma, que se consegue descobrir facilmente, e da porta utilizada, que é o porta padrão para serviços *web* (80). Esta grande vulnerabilidade implica a necessidade de *software* de terceiros para proteção da rede, como serviços de *firewall*, por exemplo. Visto não haver disponibilidade para acarretar com os custos da inclusão de ambos estes pacotes de *software*, a opção de usar o servidor local para o funcionamento normal da rede não foi considerada. No entanto, o servidor local mostrou-se útil para testar algumas questões relativas a proteções contra ataques do sistema, mais especificamente, ataques que envolvessem ligações aplicação-servidor e dispositivo-servidor.

6.3.3 Implementação do Módulo de Medição de Temperatura

O módulo de temperatura é composto por um dispositivo de medição de temperatura muito semelhante ao dispositivo da primeira rede implementada não só em termos de *hardware*, que é igual para os dois dispositivos (Figura 6.6), como na maneira como regista o valor da temperatura através do valor lido pelo termístor.

As diferenças deste dispositivo verificam-se na existência do *token* único deste dispositivo para autenticação na rede, e de a temperatura ser medida e enviada para o servidor em intervalos de um segundo, como é possível observar pelo excerto de código presente na Listagem 6.4, que mostra a placa a preparar função de registo da temperatura para funcionar em intervalos de um segundo e a ligar-se ao *router* utilizando o seu *token* de autenticação e as credenciais da rede.

```
// Token de Autenticacao unico para cada dispositivo
char auth[] = "OrfAuyb68amIZ4dGmKxXKGkVI6xb_ysf";

// Credenciais de Acesso da Rede
char ssid[] = "Nome_da_Rede";
char pass[] = "Password_da_Rede";

//setup do timer
BlynkTimer timer;

void setup()
{
  //ligacao ao hotspot da rede e ao servidor cloud
  Serial.begin(9600);
  WiFi.disconnect();
  delay(3000);
  Blynk.begin(auth, ssid, pass);

  //Timer que vai chamar a funcao "temperaturaTimer" de um em um segundo
  timer.setInterval(1000L, temperaturaTimer);
  //1000 ms=1 s ; L para indicar que é uma variavel do tipo long
}
```

Listagem 6.4: Excerto de Código Utilizado no Dispositivo de Temperatura da Segunda Rede

```
//funcao para enviar dados para um pin virtual
 Blynk.virtualWrite(VX, value);
```

Listagem 6.5: Método para Enviar Informação para um Pino Virtual

A função "temperaturaTimer", irá ser chamada em intervalos de um segundo e é composta pelas mesmas linhas de código que o excerto de código apresentado na Listagem

6.2 mas com a adição de um método para enviar o valor registado para o servidor da rede. Este método que se encontra descrito na Listagem 6.5, passa como argumentos o pin virtual para qual se deseja enviar informação e o valor que se pretende enviar.

6.3.4 Implementação do Módulo de Segurança

O módulo de Segurança é o único módulo do sistema que utiliza dois dispositivos distintos, um para registar movimentos realizados no exterior e outro para fazer a mesma tarefa no interior da habitação.

Começando pelo dispositivo de deteção de movimento exterior, cujo esquema se encontra na Figura 6.13, este é composto por um sensor PIR (Passive Infrared sensor, em português, sensor de infravermelhos passivo) e por uma placa NodeMCU. A placa alimenta o sensor com uma tensão de 3.3 V e este transmite o seu sinal para o pino D1 da placa.

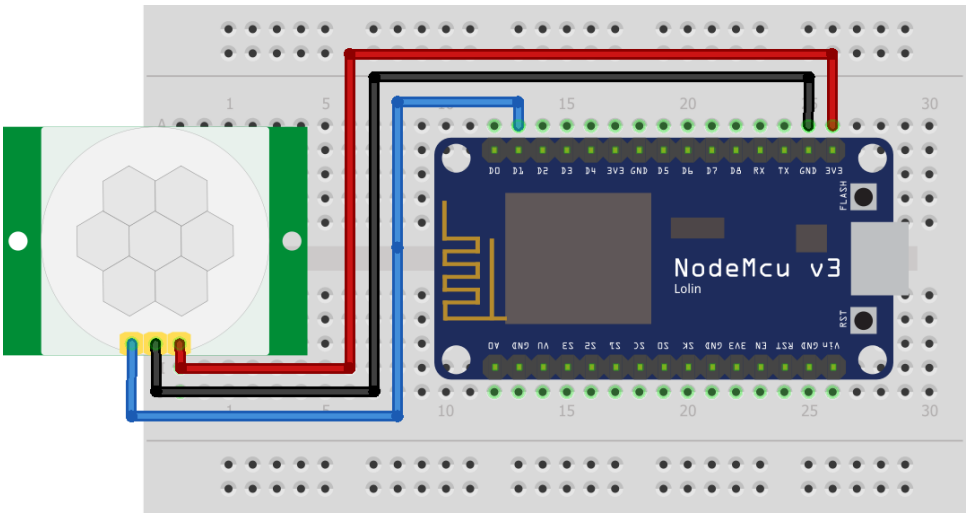

Figura 6.13: Ilustração do *Hardware* do Dispositivo de Deteção de Movimento Exterior

```
/funcao para detetar movimentos
void getPIRData()
{
  byte state = digitalRead(sensorPin);

  //sensor PIR tem output 1 quando deteta movimento
  if(state == 1)
  {
    Serial.println("Somebody is in this area!");
  }

  Blynk.virtualWrite(V2, state);
}
```

Listagem 6.6: Método Utilizado para Deteção de Movimento Exterior

O sensor PIR, é um sensor piroelétrico, isto é, este sensor é sensível a diferenças de calor radiante causado por radiação infravermelha e utiliza essa sensibilidade para detetar movimentos em seu redor a uma distâcia de cerca de 3 a 4 metros. Como se pode observar no excerto de código da Listagem 6.6, este sensor, que normalmente envia o sinal "0" para o pino D1 da placa, ao sentir algum tipo de movimento ao seu redor envia o sinal "1" para a placa, que por sua vez vai enviar o valor para o pino virtual dedicado a este sensor.

Passando agora para o dispositivo de deteção de movimento interior, que se encontra representado na Figura 6.14, este dispositivo é composto pelo sensor de deteção de obstáculos ST1081 e por uma placa NodeMCU.

É importante notar que, tal como aconteceu com o sensor de temperatura, não existe nenhum tipo de ilustração gratuita do sensor ST1081 no *software* que foi usado para fazer as ilustrações do *hardware*. Por esta razão e exclusivamente para efeitos de ilustração optou-se pela utilização de um elemento equivalente, como é o sensor FC51. A única diferença entre ambos é que o sensor ST1081 possui dois potenciómetros para controlo da intensidade do emissor e da sensibilidade do receptor, enquanto o sensor FC51 apenas tem um potenciómetro para controlo da intensidade do emissor.

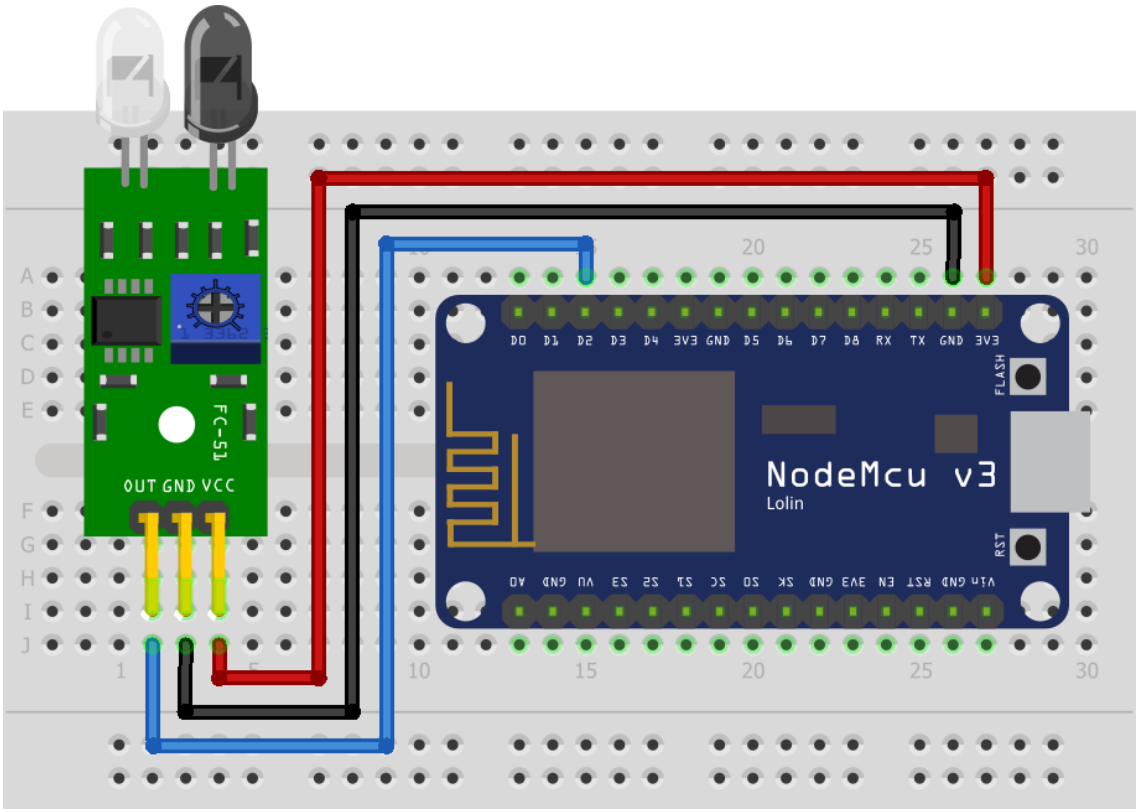

Figura 6.14: Ilustração do *Hardware* do Dispositivo de Deteção de Movimento Interior

O sensor ST1081 tem um potencial alcance de 2 a 40 cm e em estado normal transmite o valor "1" ao pino D2 da placa NodeMCU, estando sempre a emitir um feixe de radiação infravermelha. Ao haver a reflexão deste feixe numa dada superfície, o recetor do sensor vai detetar o feixe emitido pelo emissor do sensor pelo que o sensor enviará o valor "0"

ao pino D2 da placa, detetando assim uma ocorrência de movimento na sua área de influência. Como há o envio de um zero na deteção de movimento, o código necessário para enviar este sinal para o pin virtual da rede dedicado ao dispositivo de movimento interior é ligeiramente diferente do código programado no dispositivo anterior, como se pode observar na Listagem 6.7.

```
void getIRData()
{
  //leitura do sensor
  byte sensorValue = digitalRead(sensorPin);

  //sensor IR tem output 0 quando deteta movimento portanto é necessário enviar 1 para o
    ↪ pino virtual
  if(sensorValue == 0)
  {
    Serial.println("Somebody is in this area!");
    Blynk.virtualWrite(V3, 1);
  }else
  {
   Blynk.virtualWrite(V3, 0);
  }
}
```

Listagem 6.7: Método Utilizado para Deteção de Movimento Interior

É importante referir que, embora componham um módulo, cada um destes dois dispositivos possui um *token* de autenticação único e que, devido à natureza do tipo de mensagens que têm de efetuar, estes sensores enviam informação para a rede de forma esporádica (quando são detetados movimentos) e sem qualquer padrão aparente ou intervalo de tempo.

6.3.5 Implementação do Módulo do Contador Bluetooth

O módulo Bluetooth é composto por um dispositivo apenas que contém um módulo Bluetooth HC-05 ligado a uma placa Arduino Uno, como é possível ver na Figura 6.15.

O módulo HC-05 é um modulo que utiliza um PIN (Personal Identification Number) de 4 algarismos de modo a emparelhar-se com um dispositivo que, no caso desta rede, se pretende que seja o *smarphone* do utilizador. Este módulo vai-se encarregar de todas as comunicações do dispositivo com a aplicação, que por sua vez, vai indicar à rede os valores do número que o utilizador pretende contar. O código que suporta esta ligação é algo diferente do código utilizado pelas placas NodeMCU e pode ser observado na Listagem 6.8. A ligação com o telefone é estabelecida através do emparelhamento que por sua vez permite o envio do *token* de autenticação, de modo a que o dispositivo consiga ir buscar informação do pino virtual ao qual o contador está associado.

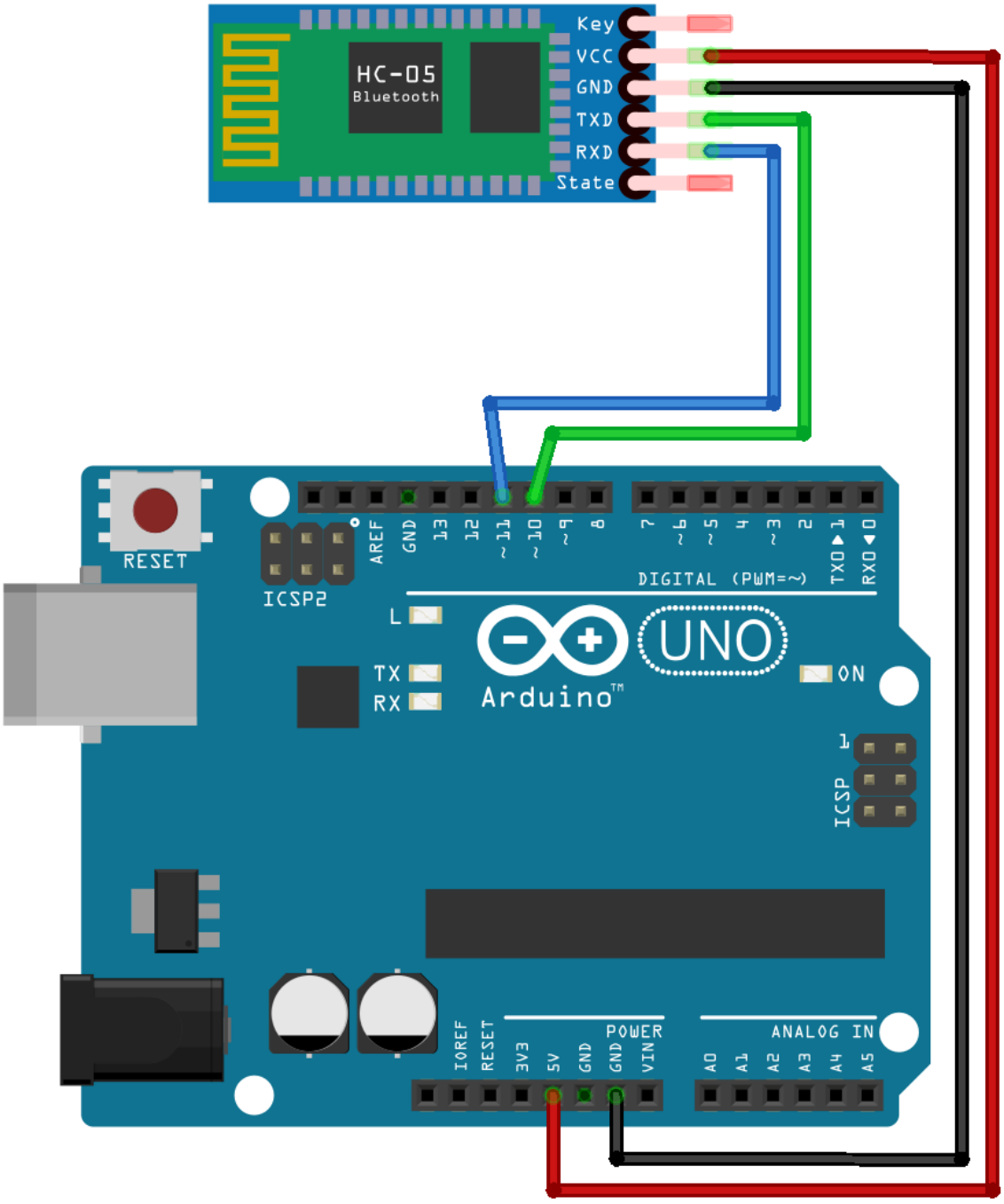

Figura 6.15: Ilustração do *Hardware* do Dispositivo Bluetooth

```
#define BLYNK_PRINT Serial

#include <BlynkSimpleSerialBLE.h>
#include <SoftwareSerial.h>

SoftwareSerial SwSerial(10, 11); // RX, TX
SoftwareSerial SerialBLE(10, 11); // RX, TX (pin 10->tx do modulo ; pin 11->rx do modulo

// Token de Autenticacao unico para cada dispositivo
char auth[] = "M76-XXJOKpS3qRzyDLd\_i8JsNosrDZhh";

void setup()
{
  Serial.begin(9600);

  SerialBLE.begin(9600);
  Blynk.begin(SerialBLE, auth);
}
```

Listagem 6.8: Excerto de Código Utilizado para Conectar o Dispositivo Bluetooth à Aplicação e ao Servidor

A placa Arduino Uno serve apenas de interface programável de modo a que o dispositivo consiga ir buscar o valor do contador que se encontra no seu pino virtual dedicado. A placa serve também como um modo do dispositivo ter output do valor do contador à medida que é alterado pelo utilizador através da sua aplicação. O excerto de código programado na placa utilizado para consultar o valor do pino virtual à medida que este é alterado e imprimir esse valor no monitor série do dispositivo encontra-se na Listagem 6.9

```
// Recebe como parametro o pino virtual V0 quando este regista alteracoes
BLYNK_WRITE(V0)
{
  int pinValue = param.asInt(); // mete em pinValue o valor que vem de V0
  Serial.print("O contador vai em: ");
  Serial.println(pinValue);//mostra o valor de V0, que é o valor do contador
}
```

Listagem 6.9: Excerto de Código Utilizado para Consultar o Valor do Contador

6.3.6 Implementação do Módulo de Controlo de Iluminação

O módulo de controlo de iluminação é composto por um dispositivo que contem um sensor LDR (Light Dependent Resistor, em português, Resistor Dependente de Luz), uma resistência de 10 kΩ, um LED e uma placa NodeMCU, estando montado de acordo com a ilustração da Figura 6.16

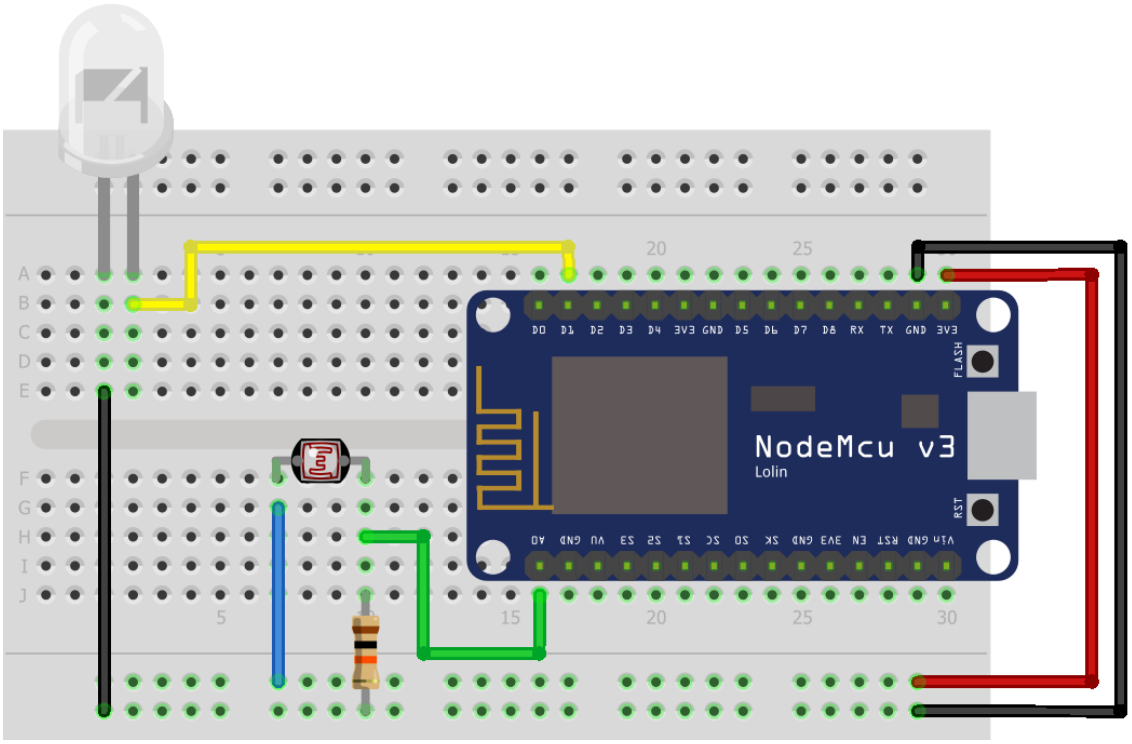

Figura 6.16: Ilustração do *Hardware* do Dispositivo de Controlo de Iluminação

Visto que o sensor LDR altera a sua resistência elétrica consoante a luz que sobre si incide, sendo que quanto mais luz, menor resistência irá impor no circuito, é possível determinar a tensão aos polos deste através de um divisor de tensão através da tensão de entrada imposta pela placa NodeMCU (que será de 3.3 V) e da resistência de 10 kkΩ. Este valor é transmitido ao pino analógico da placa (A0) e pode tomar um valor entre "0" e "255".

De seguida a placa vai converter este valor para uma percentagem e , de acordo com a leitura dos pinos virtuais que estão dedicados aos botões de controlo do sistema de iluminação (que se podem observar na Figura 6.9), controlar o LED ligado ao pino D1 da placa, que representa as iluminações da habitação. Como se pode observar no Excerto de código da Listagem 6.10, o circuito funciona no modo automático por defeito, ligando o LED caso o nível a iluminação seja inferior a 20% e desligando o mesmo quando o valor da iluminação registada é superior a esse limite. Caso o botão para ligar o modo manual da rede seja pressionado pelo utilizador na aplicação, o circuito passa a funcionar de forma manual, ligando e desligando a luz, consoante o outro botão da aplicação, que funciona como um interruptor de iluminação.

```
// Recebe como parametro o pino virtual V0 quando este regista alteracoes
void getLDR()
{
  byte ldrValue = analogRead(A0);//consulta o valor da tensao do sensor LDR que mostra 0
    ↪ no escuro e até 255 quando iluminado
  int lumin = (ldrValue*100) / 255;//conversao da tensao para uma percentagem

  //Funcionamento do Modo Automatico
  if(lumin <= 20 && manualState == 0)
  {
    digitalWrite(ledPin, HIGH);
    Blynk.virtualWrite(V5,100);
  }else if( lumin > 20 && manualState == 0)
  {
    digitalWrite(ledPin, LOW);
    Blynk.virtualWrite(V5,0);
  }

  //Funcionamento do Modo Manual
  if(lightSwitch == 0 && manualState == 1)
  {
    digitalWrite(ledPin, LOW);
    Blynk.virtualWrite(V5,0);
  }else if( lightSwitch == 1 && manualState == 1)
  {
    digitalWrite(ledPin, HIGH);
    Blynk.virtualWrite(V5,100);
  }

  Blynk.virtualWrite(V4, lumin);
}
```

Listagem 6.10: Excerto de Código Utilizado para Controlo Automático e Manual da Iluminação

É Importante referir que, embora a leitura do nível de iluminação exterior seja enviado para o servidor de uma maneira periódica em intervalos de 1 segundo (tal como o sensor de temperatura), o dispositivo recebe os valores dos pinos virtuais dos botões de controlo quando estes são alterados, tal como o dispositivo do contador Bluetooth (o método é idêntico ao excerto de Código visto na Listagem 6.9, variando apenas o valor de cada pino virtual). Isto significa que o dispositivo de controlo de iluminação contacta o servidor tanto de forma periódica, para enviar valores para pinos virtuais, como de forma esporádica, para registar as alterações de outros pinos virtuais.

Capítulo 7

Validação

Este capítulo serve para verificar o valor dos protótipos desenvolvidos no âmbito desta dissertação, analisando os sistemas de acordo com os parâmetros de seguranças e privacidade estabelecidos pelos requisitos funcionais e não funcionais.

7.1 Condições de Teste

De modo a testar os fatores de segurança e privacidade de ambas as redes implementadas, foram consideradas as seguintes condições de um cenário de teste:

- Ambas as redes têm como funcionamento normal todos os seus dispositivos ligados aos respetivos serviços de servidores de recolha e processamento de informação, operando da maneira que foi descrita nos capítulos anteriores.

- O utilizador é um mero cliente da rede sendo que esta funciona sem a sua aplicação, ou seja, todos os dispositivos realizam trocas e registos de informação com a Camada de Rede, mesmo que o utilizador não tenha a aplicação ativa num dado momento ou período.

- Para efeitos de teste de conexões e autenticações com o servidor da segunda rede, devido às capacidades dos recursos disponíveis, utilizou-se um servidor local, em detrimento do servidor *cloud* disponibilizado pelo serviço oferecido pela aplicação Blynk. Estes testes, que serão descritos com mais pormenor numa das secções seguintes, foram a única instância para qual o servidor local foi utilizado, tendo-se considerado a utilização do servidor *cloud* para o funcionamento normal da rede devido à incapacidade de o servidor local se ligar à Internet, como foi descrito na Subsecção 6.3.2.

- Existe uma entidade que possui recursos semelhantes aos implementados nas duas redes e que as pretende invadir através de ataques que têm como objetivo, escutar, mudar ou modificar informação confidencial da rede, introduzir dispositivos intrusos na rede, provocar DoS de alguns ou todos os serviços ou mesmo danificar hardware.

É importante referir também que não se considerou que serviços fora da esfera de influência da rede tivessem alguma situação de interrupção do serviço prestado à rede, como por exemplo, a empresa dona do ISP da rede realizar melhorias na sua infraestrutura, causando um *downtime* temporário da ligação da rede à Internet.

7.2 Análise da Segurança das Redes Implementadas

Esta secção serve para explicitar as condições de segurança e privacidade das redes implementadas, tendo especial foco na verificação dos requisitos funcionais e não funcionais observados na Secção 5.2 de maneira a observar a existência de valor dos protótipos de redes IoT implementados.

7.2.1 Primeiro Sistema

Como já foi referido anteriormente, esta rede foi desenhada com o objectivo principal de facilitar a identificação dos problemas de segurança do mundo que é uma rede IoT, tendo-se usado recursos que se mostraram bons para a construção da rede em si, mas menos que ideais no que diz respeito ao aspeto de privacidade e confidencialidade de informação. Optou-se por esta escolha de *software* e serviços menos ideais, para que na construção de uma segunda maior e melhor iteração de uma rede IoT fosse possível mitigar ao máximo as questões deixadas em aberto pela primeira iteração, tornando assim o protótipo final o mais seguro possível.

O facto de que esta primeira rede não ser ideal do ponto de vista de segurança é evidenciado por uma análise dos requisitos funcionais e não funcionais do sistema.

Embora cumpra sem qualquer tipo de discussão os requisitos funcionais, esta rede tem algumas dificuldades em cumprir os requisitos não funcionais na sua totalidade.

O primeiro problema surge ao abrir a aplicação. Não existe qualquer rotina para autenticar o utilizador, o que significa que um intruso, se conseguisse obter acesso à aplicação, teria acesso imediato ao gráfico de registo de temperaturas, pois este é embebido na aplicação através de uma hiperligação a uma vista pública do canal do serviço Thingspeak. Este serviço, de modo a possibilitar o envio da visualização do gráfico da temperatura, impõe que a visualização seja pública, o que é considerado uma falha grave de privacidade por si só, o que por sua vez irá permitir que o intruso conhecer o número de identificação do canal (“Channel ID”) que faz parte da hiperligação do gráfico. Em suma, apenas por

conseguir acesso à aplicação, um potencial intruso poderia ter acesso aos registos de temperaturas medidas e ao identificador único do canal dedicado à rede pelo servidor, isto tudo sem que o utilizador pudesse vir a tomar conhecimento.

Em relação à autenticação do dispositivo, esta é feita através de duas chaves. Uma utiliza-se quando se quer enviar informação para o servidor *cloud* e a outra quando se deseja receber informação do mesmo. Estas chaves, que são constituídas por 16 carateres, podendo estes serem letras maiúsculas ou algarismos, não causa nenhum entrave propriamente dito à segurança da rede, pois o número possível de combinações que estas podem tomar é grande o suficiente para se considerarem seguras. Porém o método de utilização destas chaves por parte do serviço *cloud* levanta algumas questões de implementação. O serviço ThingSpeak não permite que o mesmo dispositivo utilize ambas as chaves, ou seja, o dispositivo de *hardware* não consegue enviar e receber informação de e para o servidor, o que, não levantando questões de segurança, restringe bastante a possível implementação futura de outros dispositivos nesta rede, como por exemplo um dispositivo de controlo de iluminação como o da segunda rede (que está descrito na Subsecção 6.3.6), que iria necessitar tanto de receber dados do servidor como enviar valores para o mesmo.

Relativamente às taxas de envio e receção de informação, o sistema cumpriu com os limites impostos pelo requisito, porém também levantou questões que teriam de ser respondidas na segunda implementação. O limite do período de 15 segundos por mensagem para o servidor imposta pela licença grátis do software impossibilita a implementação de dispositivos que utilizassem um tipo de comunicação esporádica. Tenha-se como exemplo um dos dispositivos de deteção de movimento da segunda rede (Como descrito na subsecção 6.3.4). Se este módulo detetasse movimento fora do intervalo dos 15 segundos, a informação não seria enviada para o servidor, o que por sua vez iria resultar no utilizador não saber que teria sido detetado movimento.

Finalmente, esta primeira rede também não possui métodos para saber se o dispositivo de medição de temperatura se encontra conectado à rede ou não, como não tem nenhuma maneira de analisar os valores recebidos de forma a detetar possíveis valores estranhos. Se um intruso decidisse levar uma pequena fonte de calor como um isqueiro ao pé do dispositivo para incutir valores falsos, o que o utilizador iria ver seria apenas um valor bastante alto de temperatura. Se o mesmo individuo decidisse danificar ou mesmo remover o dispositivo, apenas se verificaria que a informação iria para de ser atualizada na aplicação e nada relativamente ao estado da conexão do dispositivo à rede.

7.2.2 Segundo Sistema

A segunda iteração da rede IoT implementada foi construída de modo a poder responder a todas as questões levantadas pela análise da primeira iteração da rede IoT e pelos problemas identificados devido ao trabalho de pesquisa presente nos capítulos de Estado da Arte desta dissertação, tendo sempre o conceito de segurança em mente. Esta rede tentou atingir o máximo de condições de segurança, privacidade e confidencialidade

possível através da utilização de software especialmente desenhado para dispositivos IoT e do desenho de diferentes tipos de dispositivos de forma a poder variar ao máximo o tipo de informação que iria transitar pela rede.

Esta segunda rede cumpre todos os requisitos funcionais impostos e cobre praticamente todas as vulnerabilidades identificadas no primeiro sistema, tendo uma resposta positiva a quase todos os requisitos não funcionais.

Começando pela autenticação do utilizador, que mesmo tendo a *app*, necessita de criar uma conta no servidor da rede, sendo necessário um email válido e uma palavra-passe de 8 a 16 caracteres, podendo estes serem letras ou números. De modo a testar este processo de autenticação, foi utilizada a opção do servidor local, a qual permitiu descobrir que é possível limitar o número de vezes que um utilizador pode tentar fazer *login*, após o qual o utilizador tem de esperar um determinado tempo, o que protege o sistema de intrusos que tentem adivinhar uma palavra passe através de tentativas repetidas. Tanto o número de tentativas de autenticação como o tempo que o utilizador fica impedido de se tentar ligar à rede são customizáveis à vontade do administrador do servidor.

Em relação ao processo de autenticação dos dispositivos da Camada de Perceção, este ocorre sempre que um dispositivo se conecta à rede e efetua um envio ou receção de informação através de *tokens* de autenticação únicos para cada dispositivo, que são nada mais que uma *string* de 32 caracteres, podendo estes serem letras, algarismos ou mesmo símbolos como pontuação, por exemplo. É graças a estes *tokens* que o servidor consegue inferir o estado da conexão de cada dispositivo individualmente, sendo que a aplicação desenvolvida é capaz, mesmo funcionando em segundo plano, de enviar uma notificação ao utilizador que um dispositivo específico se desligou da rede, o que, por exemplo, pode servir para avisar o utilizador de um ataque que perturbe um determinado dispositivo ao ponto de este se desconectar da rede.

Tanto a aplicação, como os dispositivos e assim como o servidor suportam qualquer tipo de cadência de troca de mensagens entre todos os elementos da rede, seja esta periódica ou esporádica. As condições de envio e receção não são impostas pelo servidor mas sim por quem faz os pedidos de leitura ou escrita nos pinos virtuais, ou seja, é a aplicação e os dispositivos quem dita quando e com que frequência vão trocar informação com a rede.

Relativamente aos processos de verificação de conexões, como já foi referido, a rede está sempre ao corrente da conexão de cada um dos dispositivos graças aos seus *tokens* de autenticação. Porém, no caso do dispositivo de conexão Bluetooth observaram-se algumas incoerências, pois este dispositivo ligava-se à rede, tanto através do servidor *cloud*, como do servidor local, sendo que neste último era possível observar que dispositivo realizava pedidos de consulta como seria de esperar, mas no entanto, na aplicação nunca apareceu que este dispositivo se tinha conectado ao servidor apesar de este não ter problemas em ir buscar o valor do pino virtual ao dito servidor e mostrar o valor do contador na porta série do dispositivo. Devido a isto, considerou-se a incapacidade da aplicação mostrar o estado da conexão do dispositivo Bluetooth tivesse a ser causada por algum tipo de *bug*

visual do *software* ou devido a algum possível defeito que o módulo HC-05 adquirido tivesse.

Ainda relativo ao estado de conexões, o uso de um servidor local permitiu observar também que é feito um registo das credenciais do utilizador sempre que a aplicação realiza um pedido de receção a transmissão de dados para os pinos virtuais, o que significa que o servidor regista também as conexões da aplicação do utilizador

Finalmente, os requisitos relativos à análise de valores estranhos ao funcionamento regular da rede são os únicos que o sistema deixa algo a desejar, cumprindo parcialmente apena o requisito de análise de informação anómala, pois embora seja capaz de atuar devido ao registo de valores estranhos (através do uso de notificações) no momento que estes são processados pelo sistema, se o utilizador não tiver a aplicação ativa não é capaz de perceber que se passa alguma coisa. No entanto estes valores não são esquecidos pois o servidor é capaz de gerar e enviar um ficheiro CSV para o *email* do utilizador com os valores registados de todos os pinos virtuais do sistema com a respetiva data e hora de registo. Porém, dados anómalos presentes neste ficheiro não causam nenhuma ação na rede, pois o servidor não analisa os dados registados, o que seria necessário para efetivamente cumprir o último requisito não funcional da rede.

7.3 Considerações Finais

A implementação da primeira rede tinha o propósito de ajudar a perceber quais as principais questões relativamente à segurança dos sistemas IoT e como seria de esperar, esta rede levantou bastantes, pois não cumpriu com grande parte dos requisitos não funcionais necessários para considerar esta rede como segura.

O protótipo de rede IoT final, que resulta da segunda implementação da rede, cumpriu praticamente todos requisitos funcionais e não funcionais e ofereceu uma resposta a grande parte das questões levantadas tanto pelo trabalho de pesquisa sobre o tema como também devido aos problemas levantados pela primeira implementação, pelo que se considera que os resultados obtidos tenham sido bastante positivos.

Capítulo 8

Conclusão

Este capítulo conclui o trabalho desenvolvido no âmbito desta dissertação, através de um resumo e as devidas ilações retiradas dos resultados obtidos e sugere algumas ideias para desenvolvimentos futuros.

8.1 Síntese

O objectivo desta dissertação era aferir o estado atual de segurança em redes IoT através do desenvolvimento de um protótipo de uma rede IoT inserida num ambiente de uma *Smart Home*.

Em primeira instância foi construída uma rede mais simples, através de recursos não tão orientados para desenvolvimento de redes IoT para que fosse possível evidenciar as principais preocupações de segurança e privacidade que pairam sobre o mundo das redes IoT. De seguida, com o conhecimento adquirido da primeira implementação, construiu-se uma segunda rede, mais complexa e através de recursos especialmente focados em desenvolvimento de redes IoT.

Como seria de esperar, observaram-se alguns problemas relativamente à segurança da primeira rede, porém a segunda implementação da rede colmatou quase todos os problemas levantados pela primeira rede, o que se resultou num protótipo de uma rede modelo de um sistema IoT que tem boas condições de segurança e privacidade.

Finalizando, o protótipo de rede IoT construído contribui para o estudo do tema de segurança em redes IoT, pois evidencia as condições atuais de segurança, privacidade e confidencialidade destes sistemas, explicitando quais são os seus pontos fortes e por onde é possível melhorar.

8.2 Trabalho Futuro

O protótipo de rede IoT construído, embora possa ser considerada bastante seguro, tem algumas áreas que poderão beneficiar de algumas otimizações de modo a que a rede possa responder a futuros requisitos

De modo a melhor o protótipo de rede IoT implementado, ficam alguns tópicos como sugestões para futuros desenvolvimentos:

- Optimização do código dos dispositivos, de modo a usarem o mínimo de memória possível;
- Aplicação que analise a informação do ficheiro CSV de registo de valores de modo a permitir a atuação sobre todos os valores anómalos registados;
- *Software* de redireccionamento de portas e *firewall*, de modo a que seja possível usar o servidor local em segurança.

Bibliografia

[1] M. Abomhara e K. Geir M. "Security and Privacy in the Internet of Things: Current Status and Open Issues". Em: *Computer* (). DOI: https://doi.org/10.1109/PRISMS.2014.6970594.

[2] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta e M. Alikarar. "A smart home energy management system using IoT and big data analytics approach". Em: *IEEE Transactions on Consumer Electronics* 63.4 (2017), pp. 426–434. ISSN: 15584127. DOI: 10.1109/TCE.2017.015014.

[3] K. Asthon. "That 'Internet of Things' Thing". Em: *RFID Journal* (2009), pp. 97–114. URL: https://www.rfidjournal.com/articles/view?4986.

[4] J. Branger e Z. Pang. "From automated home to sustainable, healthy and manufacturing home: a new story enabled by the Internet-of-Things and Industry 4.0". Em: *Journal of Management Analytics* 2.4 (2015), pp. 314–332. ISSN: 23270039. DOI: 10.1080/23270012.2015.1115379.

[5] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac e P. Faruki. "Network Intrusion Detection for IoT Security Based on Learning Techniques". Em: *IEEE Communications Surveys and Tutorials* 21.3 (2019), pp. 2671–2701. ISSN: 1553877X. DOI: 10.1109/COMST.2019.2896380.

[6] E. Fernandes, A. Rahmati e N. Feamster. "New Problems and Solutions in IoT Security and Privacy". Em: Table I (2019), pp. 1–5. arXiv: 1910.03686. URL: http://arxiv.org/abs/1910.03686.

[7] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal e B. Sikdar. "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures". Em: *IEEE Access* 7 (2019), pp. 82721–82743. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2924045.

[8] F. Hussain, R. Hussain, S. A. Hassan e E. Hossain. "Machine Learning in IoT Security: Current Solutions and Future Challenges". Em: (2019), pp. 1–23. arXiv: 1904.05735. URL: http://arxiv.org/abs/1904.05735.

[9] Internet World Stats. "Internet Growth Statistics". Em: (2019). URL: https://www.internetworldstats.com.

[10] M. A. Khan e K. Salah. "IoT security: Review, blockchain solutions, and open challenges". Em: *Future Generation Computer Systems* 82 (2018), pp. 395–411. ISSN: 0167739X. DOI: 10.1016/j.future.2017.11.022. URL: https://doi.org/10.1016/j.future.2017.11.022.

[11] M. Kumar. *How to Hack WiFi Password from Smart Doorbells*. 2016. URL: https://thehackernews.com/2016/01/doorbell-hacking-wifi-pasword.html.

[12] M. Leo, F. Battisti, M. Carli e A. Neri. "A federated architecture approach for Internet of Things security". Em: *2014 Euro Med Telco Conference - From Network Infrastructures to Network Fabric: Revolution at the Edges, EMTC 2014* (2014), pp. 1–5. DOI: 10.1109/EMTC.2014.6996632.

[13] R. Mahmoud, T. Yousuf, F. Aloul e I. Zualkernan. "Internet of things (IoT) security: Current status, challenges and prospective measures". Em: *2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015* (2016), pp. 336–341. DOI: 10.1109/ICITST.2015.7412116.

[14] S. N. Mohanty, K. C. Ramya, S. S. Rani, D. Gupta, K. Shankar, S. K. Lakshmanaprabu e A. Khanna. "An efficient Lightweight integrated Blockchain (ELIB) model for IoT security and privacy". Em: *Future Generation Computer Systems* 102 (2020), pp. 1027–1037. ISSN: 0167739X. DOI: 10.1016/j.future.2019.09.050. URL: https://doi.org/10.1016/j.future.2019.09.050.

[15] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum e N. Ghani. "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations". Em: *IEEE Communications Surveys and Tutorials* 21.3 (2019), pp. 2702–2733. ISSN: 1553877X. DOI: 10.1109/COMST.2019.2910750.

[16] J. Pacheco e S. Hariri. "IoT security framework for smart cyber infrastructures". Em: *Proceedings - IEEE 1st International Workshops on Foundations and Applications of Self-Systems, FAS-W 2016* (2016), pp. 242–247. DOI: 10.1109/FAS-W.2016.58.

[17] M. Patil, A. Adamuthe e A. Umbarkar. "Smartphone and IoT Based System for Integrated Farm Monitoring". da. Em: *Techno-Societal 2018*. Ed. por P. P., R. B., B. R., V. A. e A. S. Cham: Springer, 2020. DOI: 10.1007/978-3-030-16848-3_43.

[18] V. Sachdeva e L. Chung. "Handling non-functional requirements for big data and IOT projects in Scrum". Em: *Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering* (2017), pp. 216–221. DOI: 10.1109/CONFLUENCE.2017.7943152.

[19] J. Sengupta, S. Ruj e S. Das Bit. "A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT". Em: *Journal of Network and Computer Applications* 149.October 2019 (2020), p. 102481. ISSN: 10958592. DOI: 10.1016/j.jnca.2019.102481. URL: https://doi.org/10.1016/j.jnca.2019.102481.

[20] S. Singh e N. Singh. "Business Opportunities & Reference Architecture for E-commerce". Em: *Ieee* (2015), pp. 1577–1581.

[21] H. Suo, J. Wan, C. Zou e J. Liu. "Security in the internet of things: A review". Em: *Proceedings - 2012 International Conference on Computer Science and Electronics Engineering*, *ICCSEE 2012* 3 (2012), pp. 648–651. DOI: 10.1109/ICCSEE.2012.373.

[22] L. Xiao, X. Wan, X. Lu, Y. Zhang e D. Wu. "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?" Em: *IEEE Signal Processing Magazine* 35.5 (2018), pp. 41–49. ISSN: 15580792. DOI: 10.1109/MSP.2018.2825478.

[23] N. Zhang, S. Demetriou, X. Mi, W. Diao, K. Yuan, P. Zong, F. Qian, X. Wang, K. Chen, Y. Tian, C. A. Gunter, K. Zhang, P. Tague e Y.-H. Lin. "Understanding IoT Security Through the Data Crystal Ball: Where We Are Now and Where We Are Going to Be". Em: (2017). arXiv: 1703.09809. URL: http://arxiv.org/abs/1703.09809.

[24] K. Zhao e L. Ge. "A survey on the internet of things security". Em: *Proceedings - 9th International Conference on Computational Intelligence and Security*, *CIS 2013* (2013), pp. 663–667. DOI: 10.1109/CIS.2013.145.