



UNIFACS

UNIVERSIDADE SALVADOR

LAUREATE INTERNATIONAL UNIVERSITIES'

**UNIFACS UNIVERSIDADE SALVADOR
MESTRADO ACADÊMICO EM SISTEMAS E COMPUTAÇÃO**

FÁBIO FONSECA BARBOSA GOMES

**REMOTE HOME: UMA PLATAFORMA PARA INTERAÇÕES VIA DISPOSITIVOS
MÓVEIS EM AMBIENTES RESIDENCIAIS**

Salvador
2016

FÁBIO FONSECA BARBOSA GOMES

**REMOTE HOME: UMA PLATAFORMA PARA INTERAÇÕES VIA DISPOSITIVOS
MÓVEIS EM AMBIENTES RESIDENCIAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Computação de UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Artur Henrique Kronbauer.

Salvador
2016

FICHA CATALOGRÁFICA

Elaborada pelo Sistema de Bibliotecas da UNIFACS Universidade Salvador, Laureate International Universities)

Gomes, Fábio Fonseca Barbosa

Remote home: uma plataforma para interações via dispositivos móveis em ambientes residenciais./ Fábio Fonseca Barbosa Gomes.- Salvador: UNIFACS, 2016.

171 f. : il.

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Computação de UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Artur Henrique Kronbauer.

1. Interação homem-máquina. 2. Ambientes Inteligentes. 3. Interações Naturais. I. Kronbauer, Artur Henrique, orient. II. Título.

CDD: 004.019

TERMO DE APROVAÇÃO

FÁBIO FONSECA BARBOSA GOMES

REMOTE HOME: UMA PLATAFORMA PARA INTERAÇÕES VIA DISPOSITIVOS
MÓVEIS EM AMBIENTES RESIDENCIAIS

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, UNIFACS Universidade Salvador, Laureate International Universities, pela seguinte banca examinadora:

Artur Henrique Kronbauer – Orientador _____
Doutor em Ciência da Computação pela Universidade Federal da Bahia (UFBA)
UNIFACS Universidade Salvador, Laureate International Universities

Jorge Alberto Prado de Campos _____
Ph.D. em Spatial Information Science and Engineering pela University of Maine at
Orono (EUA)
UNIFACS Universidade Salvador, Laureate International Universities

Jader Cristiano Magalhães de Albuquerque _____
Doutor em Difusão do Conhecimento pela Universidade Federal da Bahia (UFBA)
Universidade do Estado da Bahia (UNEB)

Salvador, 28 de outubro de 2016.

Dedico este trabalho à minha esposa Laila,
minha filha Beatriz, meus pais, familiares e
amigos.

AGRADECIMENTOS

Agradeço a minha esposa Laila Caroline Oliveira Souza Barbosa Gomes e filha Beatriz Oliveira Barbosa Gomes, além da minha família e amigos pelo apoio e compreensão.

Gostaria de registrar meus sinceros agradecimentos ao meu orientador, Prof. Dr. Artur Henrique Kronbauer, pela orientação e confiança.

Gostaria, também, de registrar meus agradecimentos à equipe de pesquisa do *Framework Para Ambientes Ubíquos*: Igor González Pimenta, Ítalo Ribeiro, Brunno Brito de Araújo e Lívia Sarmiento. Finalmente, agradeço ao meu grande amigo Diego Passos Costa, por participar deste mestrado juntamente comigo.

RESUMO

Atualmente, a sociedade está cada vez mais dependente de uma tecnologia integrada e aprimorada às necessidades do seu cotidiano. Os dispositivos móveis ganharam popularidade e, geralmente, indivíduos os carregam consigo. Desta forma, eles tornaram-se um veículo computacional com diversas aplicabilidades. Assim, o objetivo central deste trabalho é apresentar uma plataforma que possibilite a interação do usuário com diferentes dispositivos eletrônicos por meio de um controle remoto universal, implementado em um aplicativo para *smartphone*. Para alcançar os resultados esperados, foi realizada uma investigação bibliográfica com a finalidade de verificar os trabalhos sobre controle remotos universais para ambientes residenciais nesta pesquisa. Após, foi desenvolvido um modelo, em três camadas, com o propósito de estruturar a arquitetura para integrar comandos executados a partir de um *smartphone* com aparelhos eletrônicos disponíveis em um cenário de interação. Para validar o modelo, foi construída uma plataforma contemplando o aplicativo móvel (controle remoto universal), um *middleware* (infraestrutura de comunicação) e os atuadores (dispositivos eletrônicos). Para verificar as potencialidades da plataforma e a usabilidade do aplicativo, foi realizado um experimento com a participação de 30 pessoas, que responderam a um questionário com o objetivo de contribuir com a avaliação da proposta apresentada. Uma das potenciais contribuições deste trabalho é ter a possibilidade de controlar e identificar os dispositivos eletrônicos no ambiente, inclusive aqueles que não possuem um controle remoto, além de aumentar a qualidade de vida das pessoas, em especial, as que possuem restrições de locomoção. Os resultados observados sugerem que a solução é válida e funciona de forma satisfatória em ambientes residenciais.

Palavras-chave: *Smartphone*, Dispositivo Móvel, Automação Residencial, Domótica e Controle Remoto Universal.

ABSTRACT

Nowadays, society is increasingly dependent on an integrated and enhanced technology to the needs of their daily lives. Mobile devices have gained popularity and usually the individuals carry with them. Thus, they become a computational vehicle in various applicability. Thus, the main objective of this paper is to present a platform that allows user interaction with different electronic devices through a universal remote control, implemented in a smartphone application. To achieve the expected results, a literature search was conducted in order to verify the papers on universal remote control for home environments this search. After a model was developed in three layers, with the purpose of structuring the architecture to integrate commands executed from a smartphone with electronic devices available in an interactive setting. To validate the model, a platform covering the mobile application (universal remote) was built, a middleware (communication infrastructure) and actuators (electronic devices). To check the platform capabilities and application usability, an experiment was conducted with the participation of 30 people who responded to a questionnaire in order to contribute to the evaluation of the proposal. One of the potential contribution of this work is to have the ability to control and identify the electronic devices in the environment, including those that do not have a remote control, and increase the quality of life of people, especially those with mobility restrictions. The results suggest that the solution is valid and works satisfactorily in residential environments.

Keywords: Smartphone, Mobile Device, Smart House, Residential Automation, Domotics and Universal Remote Control.

LISTA DE FIGURAS

Figura 1 - Quadro metodológico	23
Figura 2 - Quanto aos objetivos.....	24
Figura 3 - Quanto aos procedimentos técnicos	25
Figura 4 - Etapas da pesquisa.....	26
Figura 5 - Automação Residencial integrando os subsistemas da residência	30
Figura 6 - Divisão dos setores na Automação Residencial	32
Figura 7 - Ondas da Computação	34
Figura 8 - Áreas que formam a Computação Ubíqua	36
Figura 9 - Simon Personal Communicator.....	38
Figura 10 - Palm Pilot.....	39
Figura 11 - Nokia 9110 Communicator	39
Figura 12 - BlackBerry 850.....	40
Figura 13 - BlackBerry 5810.....	40
Figura 14 - Treo 600.....	41
Figura 15 - iPhone	42
Figura 16 - Motorola Droid / Milestone	43
Figura 17 - Sprint.....	43
Figura 18 - Camadas do Android.....	46
Figura 19 - Camadas do iOS	47
Figura 20 - Irdroid v1.0	52
Figura 21 - Irdroid Wi-Fi v3.0.....	53
Figura 22 - L5 Universal Remote.....	53
Figura 23 - Harmony Smart Control.....	54
Figura 24 - FLPR Universal Remote Control	55
Figura 25 - Gear4 UnityRemote Universal Remote Control.....	55
Figura 26 - iRule	56
Figura 27 - X10 Commander	56
Figura 28 - IR Universal Remote	57
Figura 29 - Samsung AllShare Control	57
Figura 30 - Smart House Controlling System	59
Figura 31 - Mapa dos dispositivos eletrônicos.....	59
Figura 32 - KNX Home Automation	60
Figura 33 - HouseGenie	60

Figura 34 - Home Automation System.....	62
Figura 35 - CPUC.....	63
Figura 36 - UniCon Smart Remote Control.....	64
Figura 37 - Remote Control of a Domestic Equipment	64
Figura 38 - Interface Fixa	65
Figura 39 - <i>Three-tier system</i>	66
Figura 40 - Sistemas Operacionais Utilizados	70
Figura 41 - Quantidade de Trabalhos com <i>Middleware</i>	71
Figura 42 - Identificação de dispositivos no ambiente residencial	72
Figura 43 - Tipo de <i>Renderização</i> do Controle Remoto Universal	72
Figura 44 - Atuadores encontrados na pesquisa	73
Figura 45 - Tecnologias de Rede sem Fio.....	74
Figura 46 - Imagem do Modelo Proposto	79
Figura 47 - Diagrama de Componentes da Camada de Interação	79
Figura 48 - Diagrama de Sequência da Camada de Interação	80
Figura 49 - Diagrama de Componentes da Camada de Gerência.....	81
Figura 50 - Diagrama de Sequência da Camada de Gerência	82
Figura 51 - Diagrama de Componentes da Camada de Atuadores.....	82
Figura 52 - Estrutura da plataforma dividida em camadas	84
Figura 53 - Ações do aplicativo no módulo de inicialização	85
Figura 54 - Lista de dispositivos para interação	85
Figura 55 - Interfaces dinâmicas do controle remoto universal	86
Figura 56 - Módulos do <i>Middleware</i>	87
Figura 57 - Exemplo de ações realizadas pela Camada de Gerência	88
Figura 58 - Funcionamento da Camada de Atuadores.....	89
Figura 59 - Experimento do Controle Remoto Universal - TV desligada	94
Figura 60 - Experimento do Controle Remoto Universal - TV ligada	94
Figura 61 - Apresentação dos Resultados.....	95
Figura 62 - Preferência pela plataforma proposta	101
Figura 63 - Nível de Interação da Plataforma Remote Home	102
Figura 64 - Nível de Velocidade da Plataforma Remote Home	103

LISTA DE QUADROS

Quadro 1 - <i>Framework</i> DECIDE	26
Quadro 2 - Artigos selecionados	51
Quadro 3 - Produtos Comerciais selecionados	52
Quadro 4 - Plataformas Multicamadas	58
Quadro 5 - Plataformas com Duas Camadas	61
Quadro 6 - Principais conceitos relacionados à Automação Residencial e Controle Remoto Universal	67
Quadro 7 - Métrica de Sistema Operacional.....	69
Quadro 8 - Descrição das Métricas	92
Quadro 9 - Questões Utilizadas para o Controle Remoto Universal.....	92
Quadro 10 - Questões Utilizadas para a Plataforma proposta	92
Quadro 11 - Roteiro de Ações	93

LISTA DE ABREVIATURAS E SIGLAS

ACM	Association for Computer
AR	Automação Residencial
CPU	Central Process Unit
DECIDE	Determine Explore Choose Identify Decide Evaluate
IEEE	Institute of Electrical and Electronics Engineers
IHC	Interação Humano-Computador
IP	Internet Protocol
JSON	JavaScript Object Notation
PDA	Personal Digital Assistant
PLC	Power Line Communication
SMD	Multimodal Dialog Systems
UML	Unified Modeling Language
Wi-Fi	Wireless Fidelity
XML	eXtensible Markup Language

SUMÁRIO

INTRODUÇÃO	15
1.1 CONTEXTO.....	17
1.2 PROBLEMAS	18
1.3 JUSTIFICATIVA.....	20
1.4 OBJETIVO.....	21
1.5 METODOLOGIA	22
1.6 ESTRUTURA DA DISSERTAÇÃO.....	26
2 AUTOMAÇÃO EM UM AMBIENTE RESIDENCIAL	28
2.1 INTRODUÇÃO.....	28
2.2 EVOLUÇÃO DA AUTOMAÇÃO RESIDENCIAL.....	28
2.3 A AUTOMAÇÃO GERENCIANDO O AMBIENTE RESIDENCIAL.....	30
2.4 COMPUTAÇÃO UBÍQUA NA AUTOMAÇÃO RESIDENCIAL	34
2.5 A EVOLUÇÃO E IMPORTÂNCIA DOS SMARTPHONES	37
2.6 SISTEMAS OPERACIONAIS MÓVEIS	44
2.6.1 Sistema Operacional Móvel Android	45
2.6.2 Sistema Operacional Móvel iOS	46
2.7 CONSIDERAÇÕES FINAIS	47
3 TRABALHOS SOBRE CONTROLES REMOTOS UNIVERSAIS PARA AMBIENTES RESIDENCIAIS	49
3.1 INTRODUÇÃO.....	49
3.2 PLANEJAMENTO E RESULTADO DO ESTADO	49
3.3 CONTROLES REMOTOS UNIVERSAIS COMERCIAIS.....	52
3.4 CONTROLES REMOTOS UNIVERSAIS ACADÊMICOS	58
3.4.1 Controles Remotos Universais Multicamadas	58
3.4.2 Controles Remotos Universais com Duas Camadas	61
3.4.3 A Importância das Arquiteturas de Camadas	65
3.5 ANÁLISE DAS ABORDAGENS UTILIZADAS	67
3.5.1 Análise referente aos sistemas operacionais utilizados	68
3.5.2 Análise referente á utilização de middleware	70
3.5.3 Análise referente a identificação automática dos dispositivos no ambiente	71

3.5.4	Análise referente à renderização da interface no aplicativo móvel	72
3.5.5	Análise referente à utilização de atuadores no cenário de interação...	73
3.5.6	Análise referente às tecnologias de rede sem fio	74
3.6	CONSIDERAÇÕES FINAIS	74
4	REMOTE HOME: CONTROLE UNIVERSAL PARA UM AMBIENTE RESIDENCIAL	77
4.1	INTRODUÇÃO.....	77
4.2	ESPECIFICAÇÃO DO MODELO	78
4.2.1	Camada de Interação	79
4.2.2	Camada de Gerência	80
4.2.3	Camada de Atuadores.....	82
4.3	A PLATAFORMA REMOTE HOME.....	83
4.3.1	O Controle Remoto Universal.....	84
4.3.2	O Middleware	86
4.3.3	Atuadores acoplados aos Dispositivos Eletrônicos	88
4.4	CONSIDERAÇÕES FINAIS	90
5	REALIZAÇÃO DO EXPERIMENTO, RESULTADOS E ANÁLISE.....	91
5.1	INTRODUÇÃO.....	91
5.2	PLANEJAMENTO DO EXPERIMENTO	91
5.3	ANÁLISE QUALITATIVA DO CONTROLE REMOTO UNIVERSAL	96
5.3.1	Avaliação da Eficiência	96
5.3.2	Avaliação da Eficácia	97
5.3.3	Avaliação da Satisfação.....	97
5.3.4	Avaliação da Aprendizagem	97
5.3.5	Avaliação da Operabilidade	98
5.3.6	Avaliação da Acessibilidade.....	98
5.3.7	Avaliação da Flexibilidade	99
5.3.8	Avaliação da Utilidade.....	99
5.3.9	Avaliação da Facilidade de Uso	100
5.4	ANÁLISE QUALITATIVA DO USO DA PLATAFORMA REMOTE HOME.....	100
5.4.1	Avaliação da Preferência	100
5.4.2	Avaliação da Utilização	101
5.4.3	Avaliação da Interação.....	101
5.4.4	Avaliação da Velocidade de Execução	102

5.5 CONSIDERAÇÕES FINAIS.....	103
6 CONCLUSÕES E TRABALHOS FUTUROS	104
6.1 CONTRIBUIÇÕES.....	105
6.2 LIMITAÇÕES.....	106
6.3 TRABALHOS FUTUROS	107
REFERÊNCIAS.....	109
ANEXO A – DADOS DO PROJETO DE PESQUISA.....	114
APÊNDICE A – QUESTIONÁRIO PROPOSTO AOS PARTICIPANTES DO EXPERIMENTO	117
APÊNDICE B - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....	121
APÊNDICE C - ROTEIRO PARA UTILIZAÇÃO DA PLATAFORMA.....	122

INTRODUÇÃO

Os controles remotos são dispositivos que servem para prover a interação com diversos equipamentos existentes em uma residência, tais como: televisores, *home theaters*, portões de garagens, cortinas, dentre outros. A criação desse utilitário eliminou a necessidade do usuário deslocar-se até os equipamentos eletrônicos para controlá-los.

As funcionalidades dos equipamentos eletrônicos podem ser executadas a distância através do acionamento de uma função disponibilizada no controle remoto, que, por sua vez, converte o comando em ondas eletromagnéticas, que são enviadas ao equipamento eletrônico.

O controle remoto não é uma tecnologia recente, visto que ele foi criado no ano de 1838, por Nikola Tesla, responsável pela patente de um dispositivo que conseguia controlar barcos remotamente através das ondas de rádio. Naquela época, o principal objetivo da tecnologia era solucionar problemas na área militar, tais como: controlar navios, detonar bombas e acionar armas remotamente (SEITER *et al.*, 2013).

Algumas décadas depois, no ano de 1903, foi desenvolvido o Telekino, um projeto do cientista Leonardo Torres y Quevedo, que possibilitava controlar remotamente equipamentos a distância via sinais de rádio. Com o crescimento do uso do rádio, no final de 1930, os fabricantes criaram vários controles remotos, com destaque para a Philco, que lançou o Mystery Control, um controle para rádios à pilha de baixa frequência. Cerca de 10 anos depois, em 1940, foram criados os controles automáticos de garagem (SEITER *et al.*, 2013).

No início da década de 1950, Robert Adler, funcionário da empresa Zenith Electronics Corporation, criou um controle remoto que utilizava raios infravermelhos como meio de comunicação. A partir da década de 1980, essa tecnologia foi amplamente utilizada, resultando na substituição dos controles remotos existentes por uma versão mais atrativa e barata (DESAI *et al.*, 2002).

Apesar de inúmeras vantagens, os controles remotos tradicionais possuem limitações, já que, normalmente, são concebidos para uma única marca e modelo de um determinado fabricante. Desta forma, em uma residência, podem existir diversos controles remotos, um para cada equipamento.

Na tentativa de solucionar o problema, a Apple lançou o Controlador Remoto de Equipamentos (CORE) em 1987. A sua estrutura foi concebida para ser conectada a um computador e carregar uma lista de códigos do *site* do fabricante, contendo os códigos de todos os equipamentos e modelos, o que possibilitava a interação remota com os equipamentos da Apple da época. Apesar das funcionalidades do CORE serem consideradas complexas e de difícil execução, ele foi um dos primeiros controles a ser considerado universal, com a sua configuração controlada por um software de computador (DESAI *et al.*, 2002).

O controle remoto universal é um dispositivo que precisa conhecer diferentes codificações para que possa comunicar-se com os diversos equipamentos eletrônicos disponíveis no ambiente. Segundo Desai *et al.* (2002), ele pode ser dividido em dois tipos: (i) os multimarcas, que já vêm configurados para atuar com determinados aparelhos de uma série de fabricantes, tais como: televisores, DVDs ou ar-condicionados de marcas como Samsung, LG e Sony; e (ii) os programáveis, que podem comunicar-se com diferentes dispositivos de acordo com a programação realizada pelo usuário.

Além dos modelos sugeridos por Desai *et al.* (2002), existe um terceiro tipo de controle universal que é definido pelo uso de um aplicativo que possui as funcionalidades de um controle remoto universal. Ele é instalado em um dispositivo móvel que é fruto da evolução da computação e da eletrônica (que se deu com o aumento da capacidade de processamento em conjunto com a diminuição do hardware) e o surgimento de tecnologias sem fio, tais como: *Wi-Fi*, *Bluetooth* ou *ZigBee*.

A popularização dos dispositivos móveis despertou o interesse dos desenvolvedores de software em criar aplicativos apropriados para essas plataformas. Nos últimos anos, o surgimento de novos aplicativos tem se popularizado e conta com uma gama

cada vez maior de áreas contempladas com inovadoras soluções¹. Desta forma, não é possível pensar na área de Automação Residencial sem idealizar soluções, utilizando os *smartphones* como plataforma de interação (CHUEH; FANJIANG, 2012) (RAJABZADEH *et al.*, 2010).

A Automação Residencial é definida pela integração dos sistemas existentes em uma residência, tais como: hidráulico, elétrico, gás, segurança, multimídia, dentre outros. Eles estão ligados a um módulo controlador que tem como papel integrá-los e gerenciá-los, trazendo mais comodidade e qualidade de vida ao usuário (BOLZANI, 2007). Segundo Bittins *et al.* (2010) e Leitte *et al.* (2013), a Automação Residencial está em grande ascensão no mercado consumidor. Além disso, a tecnologia permite o uso racional e eficiente dos recursos, aliado com o atendimento das necessidades dos usuários (LAMINE; ABID, 2014).

Os conceitos estudados na área de Automação Residencial e a popularização da área de Aplicativos Móveis servem de base para a solução proposta nesta dissertação, que propõe integrar diferentes equipamentos eletrônicos e possibilitar a interação dos usuários a partir de um controle remoto universal, desenvolvido em forma de um aplicativo móvel para *smartphones*.

1.1 CONTEXTO

Esta pesquisa tem, como área de investigação, a interseção de duas grandes áreas: a **Automação Residencial (AR)** e a utilização de **Aplicativos Móveis** para a construção de controles remotos universais.

- a) Automação Residencial – Trata-se da integração entre os sistemas de uma residência e um sistema central. Através dele, o usuário poderá controlar os dispositivos eletrônicos que estão distribuídos em um ambiente residencial (MURATORI; BÓ, 2013).
- b) Aplicativos Móveis – Softwares desenvolvidos para dispositivos móveis, em especial, *smartphones* e *tablets*, que atualmente são muito populares. Os

¹Disponível em: <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

aplicativos móveis possibilitam disponibilizar inúmeras funcionalidades que abrangem diversificadas áreas, trazendo inovadoras soluções para a sociedade.

Observando as definições das duas áreas, percebe-se que a utilização de ambas, para a automação residencial, pode gerar soluções inovadoras. Neste contexto, a proposta desta pesquisa é criar um gerenciador central em software (*middleware*) para atuar como base de integração entre os dispositivos eletrônicos disponíveis em uma residência e um Aplicativo Móvel (controle remoto universal), que deve funcionar como o módulo de interação entre as pessoas e os dispositivos eletrônicos. Desta forma, os comandos executados remotamente pelos usuários devem se materializar em ações dos dispositivos eletrônicos existentes na residência.

Com a solução proposta, pretende-se mitigar os problemas existentes na área de controles remotos universais, que serão descritas na próxima subseção, trazendo uma solução eficiente e de baixo custo aos usuários.

1.2 PROBLEMAS

Nos trabalhos pesquisados, foram encontrados uma série de problemas a serem tratados, tanto para os dispositivos que foram concebidos para serem controlados remotamente, tais como, televisores, DVDs, *home theaters*, como os que não foram concebidos com este recurso, por exemplo, portas, janelas, lâmpadas.

Segundo Mareli *et al.* (2013), existe dificuldade em se estabelecer um protocolo comum de comunicação entre os componentes de um sistema distribuído para ambientes residenciais, por conta da heterogeneidade dos dispositivos envolvidos. Além disso, os autores reforçam a necessidade da especificação de um *middleware* que trate a complexidade dependente da quantidade e variedade de informações de contexto e serviços disponíveis no ambiente.

Autores como Byun *et al.* (2015), Chueh e Fanjiang (2012) e Zhong *et al.* (2011) argumentam que a grande diversidade de controles remotos, proporcional ao número de dispositivos eletrônicos existentes em uma residência, criam dificuldades de

interação. Tais problemas são oriundos da falta de padronização dos teclados, da dificuldade de retenção de funcionalidades e dos problemas de ergonomia.

Outro problema abordado por Koo *et al.* (2010) é a forma de comunicação entre o controle remoto e o dispositivo eletrônico, quando o sinal é transmitido por raios infravermelhos, já que é necessária apontar o controle na direção exata do dispositivo eletrônico. Este problema pode ser solucionado com novas tecnologias de comunicação como, por exemplo, a utilização do *ZigBee* (SOMANI ; PATEL, 2012).

Outro problema descrito na literatura é a interação entre idosos e dispositivos eletrônicos, em função do despreparo dessa faixa etária para a interação com novas tecnologias ou em função da dificuldade de locomoção até o dispositivo eletrônico (BITTINS *et al.*, 2010), (AHSAN *et al.*, 2014). Segundo estudos de Riekkki, *et al.* (2008), Leitte *et al.* (2013) e Pooja *et al.* (2014), a utilização de controles remotos universais pode melhorar a qualidade de vida dos usuários.

As problemáticas apresentadas pelos autores pesquisados podem ser resumidas da seguinte maneira:

- a) Dispositivos eletrônicos que não possuem controle remoto podem causar desconforto ao usuário, pois ele terá que deslocar-se até o dispositivo para realizar uma ação;
- b) Os controles remotos são específicos para cada dispositivo eletrônico, por exemplo, se uma pessoa tiver dois televisores da mesma marca, entretanto de modelos diferentes, provavelmente terá que ter dois controles remotos, um para cada equipamento;
- c) A grande quantidade de controles remotos existentes em uma residência, devido à falta de padronização entre os diferentes fabricantes;
- d) Muitos controles remotos seguem padronizações diferentes em seus teclados e funcionalidades, o que pode confundir os usuários no momento de interagir com um dispositivo eletrônico.

A partir da identificação da problemática em questão, foi possível formular o problema a ser investigado nesta dissertação da seguinte forma:

- **Como controlar diferentes dispositivos eletrônicos remotamente, independentemente de suas marcas, modelos ou de estarem preparados para receber sinais de interação à distância, com baixo custo e com conforto para os usuários?**

1.3 JUSTIFICATIVA

No segundo semestre de 2016, lojas virtuais (Apple Store, Google Play e Windows Store) disponibilizaram mais de 4.200.000 aplicativos para *download*². A evolução da área de dispositivos móveis, o desenvolvimento de novas plataformas e a popularidade dos aplicativos possibilitaram às empresas uma nova maneira de promover produtos, oferecer novos serviços e expandir as formas de alcançar mais usuários.

Através do dispositivo móvel, o cliente tem a possibilidade de controlar diversos eletrodomésticos num ambiente residencial, independente de marca, modelo e configuração prévia (RAJABZADEH *et al.*, 2010). Com isso, implantar um controle remoto universal torna-se um fator preponderante para incrementar a qualidade de vida dos usuários, com um enfoque especial para aqueles que possuem dificuldades de locomoção ou são idosos. Por conseguinte, eles poderão utilizar um controle remoto universal e realizar as ações desejadas remotamente (AHSAN *et al.*, 2014).

Neste contexto, a proposta apresentada nesta pesquisa surge como alternativa para gerenciar os dispositivos eletrônicos existentes no ambiente residencial. Como contribuições deste trabalho, podem ser destacadas: (i) a criação de uma plataforma que prevê a integração de diferentes dispositivos eletrônicos em um ambiente residencial, independente do seu modelo ou marca; (ii) o desenvolvimento de um aplicativo com a finalidade de ser um controle remoto universal, permitindo a interação dos usuários de forma fácil e dinâmica; (iii) a construção de uma interface que se molda automaticamente em função dos dispositivos eletrônicos existentes no cenário de interação e (iv) gerenciamento dos dispositivos que não possuem tecnologia para um controle remoto, como o caso de ventiladores ou lâmpadas.

²Disponível em: <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

1.4 OBJETIVO

O objetivo principal deste trabalho é **criar uma plataforma para a integração de diferentes dispositivos eletrônicos residenciais, independentemente do tipo ou modelo, e possibilitar que esses dispositivos recebam interações a distância por meio de um aplicativo móvel com as propriedades de um controle remoto universal**. Para alcançar este objetivo, foi necessário contemplar os seguintes objetivos específicos:

- a) Compreender os conceitos de Automação Residencial através da literatura;
- b) Investigar os trabalhos relacionados ao uso de controles remotos universais móveis em ambientes residenciais nos diversos repositórios de artigos;
- c) Criar um modelo para contemplar a estrutura do projeto, envolvendo todas as suas particularidades e módulos.
- d) Desenvolver uma aplicação para dispositivos móveis para ser utilizada como controle universal;
- e) Desenvolver uma solução para integrar os diferentes dispositivos eletrônicos (*middleware*), utilizado como gerenciador de configuração, comunicação e controle entre o dispositivo móvel e os atuadores;
- f) Realizar a integração dos atuadores com os dispositivos eletrônicos disponíveis no cenário de interação, disponibilizando uma plataforma para a validação do modelo proposto;
- g) Permitir a comunicação entre diversos atuadores de forma remota;
- h) Comunicar-se com equipamentos que não possuem suporte a um controle remoto tradicional;
- i) Realizar testes com a plataforma para identificar problemas e suas potencialidades;
- j) Analisar os resultados utilizando as métricas de usabilidade propostas por Kronbauer e Santos (2013), realizando possíveis melhorias.

1.5 METODOLOGIA

A metodologia é a forma do pesquisador definir como a investigação deve ser feita, seguindo uma série de etapas. “A elaboração de um projeto de pesquisa e o desenvolvimento da própria pesquisa, seja ela uma dissertação ou tese, necessitam, para que seus resultados sejam satisfatórios, estarem baseadas em um planejamento cuidadoso, reflexões conceituais sólidas e alicerçados em conhecimentos já existentes” (DA SILVA; MENESES, 2005).

Segundo da Silva e Menezes (2005) *apud* Gil (1991), as pesquisas podem ser divididas através de objetivos e procedimentos técnicos. Com relação aos objetivos, este trabalho utiliza os seguintes tipos de pesquisa:

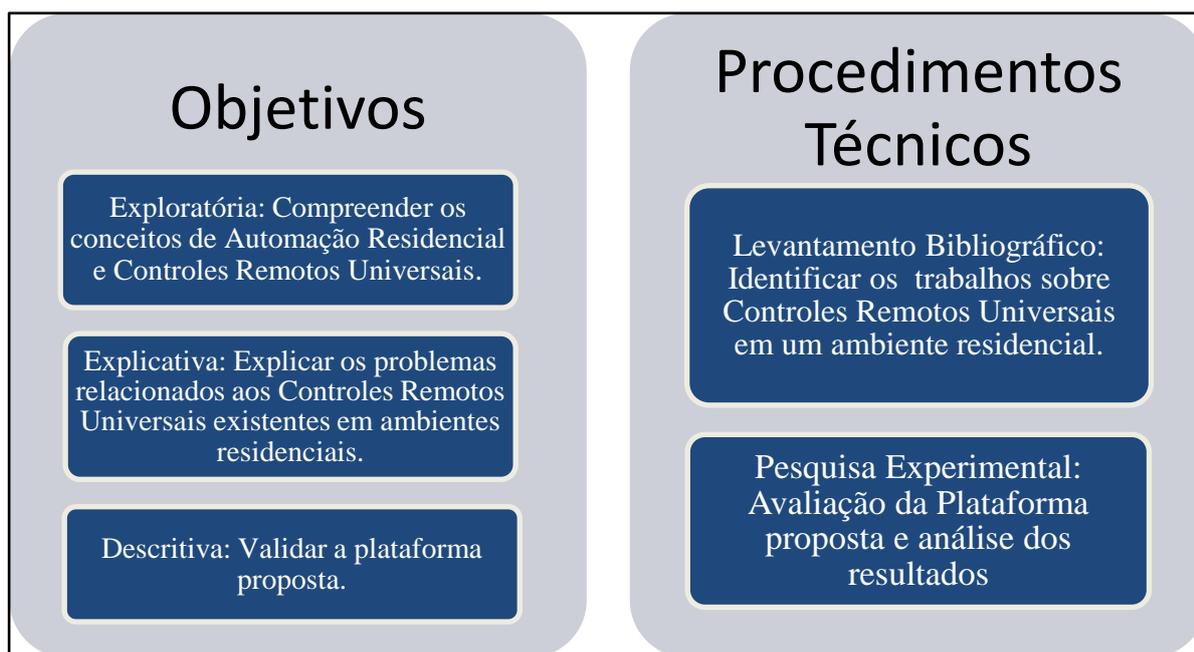
- a) Exploratória – Procura uma maior aproximação com o problema a ser investigado;
- b) Explicativa – Detalha o conhecimento acerca da realidade para explicar os motivos para determinados acontecimentos;
- c) Descritiva – Apresenta as características de diversos tipos de populações e fenômenos, utilizando técnicas padronizadas para coletar dados (questionários ou observação sistemática).

Em relação aos procedimentos técnicos, esta dissertação utiliza:

- a) Pesquisa bibliográfica – Resultado de uma profunda investigação feita através de um material escrito existente (livros, artigos, teses ou dissertações);
- b) Pesquisa experimental – Apresenta os resultados advindos de um experimento científico.

Desse modo, para relacionar os conceitos explicados anteriormente com a linha de pesquisa traçada neste trabalho, a Figura 1 apresenta o quadro metodológico utilizado.

Figura 1 - Quadro metodológico

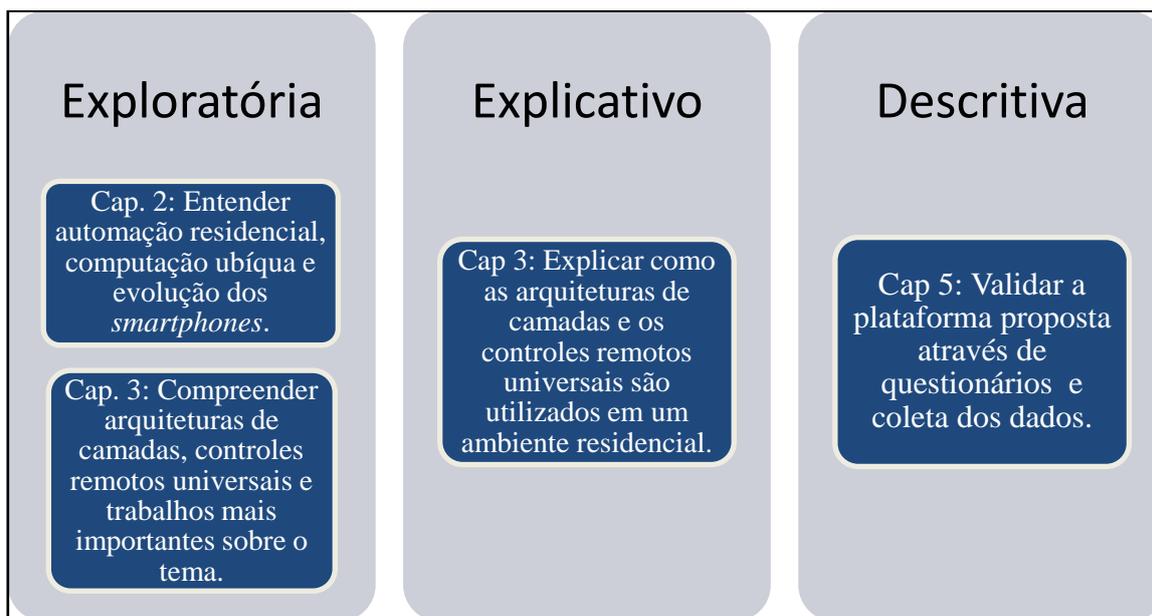


Fonte: Autoria própria do autor desta dissertação (2016).

Como houve a necessidade de compreender os conceitos de automação residencial e controles remotos universais, conclui-se que esta pesquisa é exploratória. Além disso, ela também pode ser considerada como explicativa, visto que apresenta um estudo dos controles remotos universais projetados para operar em ambientes residenciais.

Finalmente, verifica-se que a realização de uma pesquisa experimental foi necessária para testar e compreender o funcionamento do controle remoto universal criado para um ambiente de Automação Residencial, além da plataforma Remote Home. Dessa maneira, conclui-se que a pesquisa apresentada é de natureza descritiva. Por conseguinte, percebe-se que a Figura 2 fornece as informações de forma detalhada em relação à estrutura proposta nesta dissertação, no que diz respeito às classificações apresentadas.

Figura 2 - Quanto aos objetivos



Fonte: Autoria própria do autor desta dissertação (2016).

A pesquisa exploratória foi utilizada no capítulo 2 com o objetivo de entender os principais conceitos de Automação Residencial, incluindo a Computação Ubíqua e a evolução dos *smartphones*. A investigação foi realizada através da análise de artigos científicos e livros que abordavam os temas. No capítulo 3, foi necessário compreender os conceitos sobre a arquitetura de camadas e a evolução dos controles remotos universais em um ambiente residencial, além de identificar os principais trabalhos que relatavam o tema.

O capítulo 3 também foi o responsável pela pesquisa explicativa, visto que ele descreveu os trabalhos acadêmicos e comerciais existentes sobre controles remotos universais em um ambiente residencial. Finalmente, o capítulo 5 tem como principal meta a realização do estudo de caso do projeto proposto, através dele os usuários responderam a um questionário que foi utilizado para coleta de dados fundamentais para o desenvolvimento desta pesquisa.

Conforme foi explicado anteriormente, os procedimentos técnicos utilizados nesta dissertação foram: levantamento bibliográfico e pesquisa experimental, como ilustrado na Figura 3.

Figura 3 - Quanto aos procedimentos técnicos



Fonte: Autoria própria do autor desta dissertação (2016).

No capítulo 3, foram apresentados os trabalhos sobre controle remotos universais para ambientes residenciais com o objetivo de identificar e classificar os principais artigos que relatam o uso de controles remotos universais em ambientes residenciais. O capítulo 5 é responsável pela apresentação da pesquisa experimental, realizada através de um experimento prático com os usuários em um ambiente controlado.

A pesquisa foi realizada a partir das etapas apresentadas na Figura 4. Na primeira, foi feita uma revisão bibliográfica sobre a área de Automação Residencial. A partir daí, foram detectados os trabalhos que abordavam controles remotos universais através do uso de aplicativos móveis.

Desta forma, os resultados encontrados serviram de base para a fundamentação teórica na elaboração desta dissertação. Por conseguinte, uma plataforma foi desenvolvida em uma estrutura de três camadas (capítulo 4) e testada através de um experimento científico em um ambiente controlado e como resultado, obteve-se a coleta das informações para a análise (capítulo 5).

Figura 4 - Etapas da pesquisa



Fonte: Autoria própria do autor desta dissertação (2016).

Para realizar as etapas do experimento proposto foi utilizado o *framework* DECIDE (SHARP *et al.*, 2011). Conforme pode-se ver no Quadro 1, ele contempla todos os aspectos para a realização do experimento, além da coleta e posterior análise de resultados.

Quadro 1 - *Framework* DECIDE

D	Determinar o objetivo da análise.
E	Explorar perguntas a serem respondidas.
C	Escolher o método de avaliação.
I	Identificar e Administrar as questões práticas.
D	Decidir como lidar com as questões éticas.
E	Estabelecer forma de avaliar, interpretar e apresentar os resultados.

Fonte: Sharp *et al.* (2011).

Para a análise dos resultados, foi aplicado um questionário utilizando as métricas de eficácia, eficiência, satisfação, acessibilidade, aprendizagem, operabilidade, utilidade, facilidade de uso e flexibilidade (KRONBAUER; SANTOS, 2013). Além disso, foi adotada a classificação de respostas seguindo o modelo de Likert (1932), sendo de 1, para o muito insatisfeito, até 5, para o muito satisfeito.

1.6 ESTRUTURA DA DISSERTAÇÃO

Para estruturar a dissertação, foi necessária a divisão da mesma em capítulos que estão organizados da seguinte forma:

- a) Capítulo 02 – Aborda o conteúdo referente a Automação Residencial, computação ubíqua e *smartphones*.
- b) Capítulo 03 – Apresenta os trabalhos sobre controle remotos universais para ambientes residenciais de acordo com uma série de critérios, seguido dos resultados por meio da demonstração de tabelas e gráficos estatísticos.
- c) Capítulo 04 – Demonstra o modelo e estrutura da plataforma proposta, além dos passos para o seu funcionamento.
- d) Capítulo 05 – Aborda o experimento utilizado para validar a plataforma, descrevendo como foi a prática com os usuários. Além disso, estão descritas também as avaliações e resultados encontrados.
- e) Capítulo 06 – Discorre sobre as conclusões finais deste trabalho, explicando os resultados finais da plataforma proposta e fornecendo os caminhos para avanços em novas versões da plataforma.

2 AUTOMAÇÃO EM UM AMBIENTE RESIDENCIAL

O objetivo deste capítulo é abordar os principais conceitos e funcionalidades da Automação Residencial. Além disso, serão abordados os fundamentos da computação ubíqua, um breve histórico dos *smartphones* e os conceitos de sistemas operacionais móveis, que servirão para entender como eles podem ser aplicados em conjunto com a Automação Residencial na plataforma proposta. Este capítulo irá apresentar exemplos práticos que podem ser aplicados na rotina diária da sociedade, envolvendo essas áreas.

2.1 INTRODUÇÃO

A Automação Residencial é uma tecnologia que surgiu com o objetivo de unir os diferentes sistemas existentes em uma residência. Inicialmente, a ideia foi a integração entre os sistemas elétricos e hidráulicos. Isso permitiu uma grande otimização de recursos, consequentemente, diminuindo os custos e aumentando a eficiência dos sistemas da residência (BOLZANI, 2007) (MURATORI ; BÓ, 2013).

2.2 EVOLUÇÃO DA AUTOMAÇÃO RESIDENCIAL

Segundo Bolzani (2007), a partir da década de 1920, o termo “casa do futuro” surgiu para auxiliar na venda da nova invenção da época, os eletrodomésticos. A ideia dos fabricantes era que estes dispositivos iriam aumentar a qualidade de vida dos usuários da residência, com o uso de um equipamento munido de um pequeno motor para realizar uma determinada função, evitando tarefas cansativas no dia-a-dia.

Isso foi possível por causa da redução do tamanho dos motores, da eliminação de ruídos e diminuição de fios e engrenagens. Desta forma, equipamentos como ventiladores, liquidificadores, máquinas de lavar, dentre outros, beneficiaram-se dos avanços tecnológicos.

Cerca de 50 anos depois, na década de 1970, nos Estados Unidos, havia uma grande preocupação sobre quais seriam as formas de aumentar a qualidade de vida dos usuários em uma residência (AHSAN *et al.*, 2014). A partir disso, foram planejados os

primeiros módulos com comandos que poderiam ser enviados dentro de uma residência através da rede elétrica, também conhecida como PLC (*Power Line Communication* – Comunicação por Linha de Força). “Tratava-se de soluções simples, praticamente não integradas e que resolviam situações pontuais, como ligar remotamente algum equipamento ou luzes.” (MURATORI ; BÓ, 2013).

A evolução da eletrônica e computadores, a criação da internet e a grande popularidade da tecnologia de dispositivos móveis tornaram possível uma maior aceitação dos conceitos de Automação Residencial, que até então eram vistos como elementos pertencentes a um futuro distante. “Soma-se a isso a oferta abundante e barata de serviços de comunicação, como acesso em banda larga, diversas modalidades de conteúdo digital, *downloads* de músicas e filmes, etc.” (MURATORI ; BÓ, 2013).

Para Shafer *et al.* (1998) *apud* Neto (2014), um exemplo de sistema de Automação Residencial é o Microsoft Easyliving. Ele consiste em construir tecnologias focadas em ambientes residenciais e funciona da seguinte maneira:

- a) O acesso ao sistema é realizado através de um leitor biométrico ou senha, agregando segurança ao usuário;
- b) O sistema possui câmeras que são conectadas a um computador. Este tem a capacidade de calcular a profundidade do ambiente com o objetivo de identificar a presença de um indivíduo;
- c) Isto permite a automação dos dispositivos de forma que eles possam ser ligados na presença de um determinado indivíduo e desligados em sua ausência.

A Automação Residencial é uma área que está em franco crescimento (BITTINS *et al.*, 2010). Ainda assim, no Brasil “se observa uma rápida absorção das novas tecnologias pelos usuários na sua vida diária. No entanto, esta tendência ainda não se transferiu para o mercado de construção civil na mesma intensidade” (BOLZANI, 2007). Isso se torna um nicho ideal para a criação de soluções de baixo custo e rápida implementação, como é o caso da plataforma proposta nesta dissertação.

2.3 A AUTOMAÇÃO GERENCIANDO O AMBIENTE RESIDENCIAL

A Automação Residencial, para Moratori e Bó (2013), é definida como um “conjunto de serviços proporcionados por sistemas tecnológicos integrados como o melhor meio de satisfazer as necessidades básicas de segurança, comunicação, gestão energética e conforto de uma habitação”, como pode ser visualizado na Figura 5. Já os autores Sgarbi e Tonidandel (2004) e Leitte *et al.* (2013) a definem como a utilização de tecnologia embarcada em um ambiente, possibilitando aumento da qualidade de vida e segurança ao usuário comum.

Figura 5 - Automação Residencial integrando os subsistemas da residência



Fonte: WayHouse (2016).

Um fator curioso apontado por Muratori e Bó (2013) é que a Automação Residencial pode ser conhecida também como Domótica. Este termo é o resultado da união da palavra em latim *Domus* (casa) com robótica, porém ele é mais utilizado na Europa. No Brasil, o termo utilizado é Automação Residencial, que surgiu do inglês *home automation*. Para Mainardi (2005), citado por Campos e Cabral (2009), a “Automação Residencial é bastante difundida hoje em dia em todos os países industrializados. Este

tipo de automação é efetivamente a aplicação das técnicas e ferramentas de automação predial em um cenário doméstico”.

A Automação Residencial, como o nome sugere, é voltada para residências, porém seus conceitos também podem ser aplicados a escritórios, visto que seu principal objetivo é a automação das atividades cotidianas em um ambiente. Para Sgarbi e Tonidandel (2004), a Automação Residencial funciona através da integração de diversos sistemas residenciais isolados. Estes são interligados a uma central que pode ser controlada pelo usuário a distância através de algum tipo de interação. No caso desta dissertação, a comunicação é realizada através de uma interface via toques em um dispositivo móvel.

Para que uma solução de Automação Residencial seja implantada em uma residência, é necessária a presença de um responsável pela construção de um módulo controlador para os subsistemas da residência. Esse é o papel do integrador, que pode ser um profissional ou uma empresa. “O integrador pode ser uma empresa de pequeno porte ou um profissional autônomo que atua com parcerias independentes nas atividades em que não tem capacitação específica” (BOLZANI, 2007).

A figura do integrador é importante na área de Automação Residencial, visto que ela ainda não tem uma padronização definida para as tecnologias e protocolos de comunicação disponíveis. Deste modo, torna-se necessário planejar como essas tecnologias poderão ser utilizadas de forma que consigam se comunicar em um único ambiente. Para Bolzani (2007), o planejamento deve ser feito através de uma divisão de setores-chave (Figura 6) e cada um deles irá possuir uma responsabilidade específica, estes setores são: (i) controle, (ii) dados e (iii) multimídia.

Para Bolzani (2007), nos dias atuais, o conceito de “casa do futuro” é padronizar a comunicação entre as diferentes tecnologias existentes em um único sistema. A centralização do gerenciamento dos sistemas existentes na residência sob o gerenciamento de um módulo controlador é o principal conceito da Automação Residencial.

comunicar, tornando a plataforma independente de marcas ou modelos de equipamentos para ser utilizado (BOLZANI, 2007).

Portanto, para o desenvolvimento de uma solução na área de Automação Residencial, o conceito de módulo controlador é necessário. No projeto proposto neste trabalho, ele foi criado através do *middleware*. Este programa permite que os dispositivos eletrônicos disponíveis no ambiente possam ser gerenciados através de uma interface. Assim, o *middleware* poderá receber os comandos desta interface e repassando-os aos dispositivos eletrônicos através dos atuadores, para que possam executar a ação desejada pelo usuário.

Outro fator importante a ser considerado sobre a área de Automação Residencial, é que ela possui as plataformas classificadas de acordo com a seu modo de funcionamento. Desta forma, ela pode funcionar para a separação entre plataformas simples e complexas. Essa classificação é feita da seguinte maneira:

- a) Plataformas Autônomas – Consideradas mais simples, na qual é possível apenas ligar ou desligar um dispositivo eletrônico.
- b) Plataformas Integradas – Possuem diversos subsistemas integrados através de um módulo controlador, porém cada um deles depende, exclusivamente, das definições do fabricante.
- c) Plataformas Complexas – Permitem que os produtos possam ser personalizados de forma que atendam as expectativas do usuário. Elas permitem a comunicação entre os diferentes subsistemas da residência (MURATORI ; BÓ, 2013).

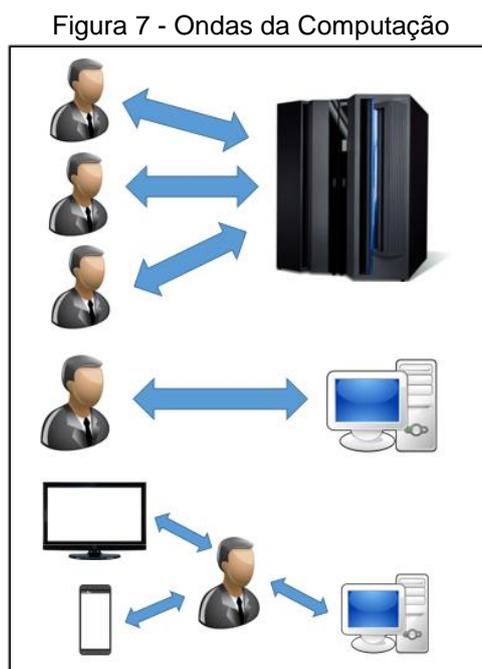
A Automação Residencial é essencial para o funcionamento da plataforma proposta nesta dissertação. Pelo que foi verificado até o momento, sabe-se que ela é um sistema complexo, no qual é possível comunicar os diferentes dispositivos da residência, sem dependência de fabricantes e com mais recursos do que apenas o ligar ou o desligar de um dispositivo. Além disso, o pesquisador fez o papel de integrador, implantando essa estrutura na residência.

2.4 COMPUTAÇÃO UBÍQUA NA AUTOMAÇÃO RESIDENCIAL

Mark Weiser (1991) define a computação ubíqua como a utilização de diversos computadores embarcados e disponíveis em um ambiente. Ela está alicerçada na ideia de que os computadores estão em todos os lugares e momentos, auxiliando o ser humano sem que este tenha consciência disso. Por consequência, é um paradigma ao qual a computação é profundamente integrada, de modo transparente às atividades cotidianas dos usuários.

Os ideais de Mark Weiser (1991) serviram como inspiração para a Automação Residencial, visto que muitos dispositivos estão começando a se comunicar via cabo ou *Wi-Fi*. Esta evolução nos meios de comunicação tornou possível o gerenciamento desses artefatos domésticos através de um único controle. Desta maneira, é permitido que um usuário consiga controlá-los sem saber a sua localização exata na residência.

Para compreender os conceitos da Computação Ubíqua, é necessário conhecer as suas origens. Desta maneira, Mark Weiser (1991) propôs que a história da computação fosse definida em três eras ou ondas computacionais (Figura 7), são elas: (i) primeira onda, (ii) segunda onda e (iii) terceira onda.



Fonte: Cook *et al.* (2009).

- a) Primeira onda (muitos-para-um) - Caracterizada pela utilização de diversos usuários para um computador, essa fase ficou conhecida como a era do *mainframe*.
- b) Segunda onda (um-para-um) - Caracterizada pela utilização de um computador por usuário, o surgimento do computador pessoal foi fundamental para essa nova fase, que ficou conhecida como a era do computador pessoal.
- c) Terceira onda (um-para-muitos) - A relação entre computadores e usuários passou de muitos para um. Desta forma, uma só pessoa tem acesso a uma gama de dispositivos. Ela pode ter um *notebook*, *tablet*, *smartphone* e computador pessoal, utilizando-os simultaneamente. Esta fase é conhecida como era da Computação Ubíqua e Pervasiva.

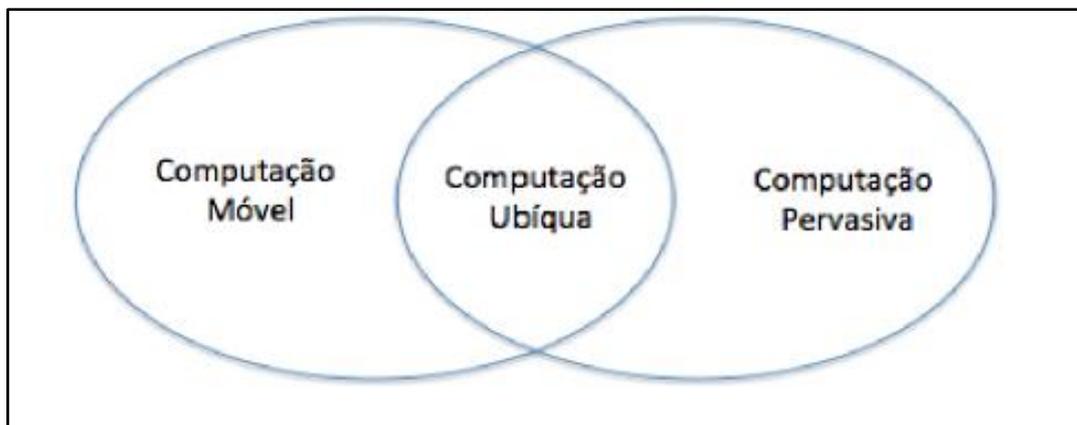
Segundo previsões de Weiser (1991), os indivíduos convivem em uma nova era na computação, da qual uma das principais características é que as interações com dispositivos computacionais ocorram de forma natural, da mesma forma que ocorre quando os seres humanos se comunicam entre si. Para Mareli *et al.* (2013), uma aplicação ubíqua identifica as necessidades de seus usuários coleta, por meio de sensores, as informações do seu contexto de execução e as atende, provendo serviços por meio de atuadores, os quais incluem diversos tipos de interfaces.

Segundo Cook *et al.* (2009), uma das principais características dos sistemas ubíquos são os ambientes altamente dinâmicos, nos quais eles são inseridos. Neles, vários dispositivos interagem entre si para fornecer informações relevantes que contribuam com as atividades diárias de seus usuários de modo imperceptível.

Um dos propósitos da computação ubíqua é facilitar a interação entre usuários e dispositivos computacionais, de modo que cada usuário não perceba que está dando comandos aos computadores espalhados pelo ambiente que o cerca. Para proporcionar esta funcionalidade, os sistemas ubíquos capturam informações sobre o ambiente para dinamicamente adaptar-se ao contexto e automaticamente executar ações apropriadas a cada mudança no cenário de interação (COOK *et al.*, 2009).

A computação ubíqua abrange várias áreas, sendo considerada multidisciplinar, entretanto Cook *et al.* (2009) a definem como sendo a junção da computação pervasiva com a móvel (Figura 8).

Figura 8 - Áreas que formam a Computação Ubíqua



Fonte: Cook *et al.* (2009).

A computação pervasiva prevê que os componentes computacionais estão embarcados no ambiente de forma invisível para o usuário, tendo a capacidade de obter informações acerca do cenário e utilizá-las para controlar, configurar e ajustar as aplicações para melhor adequar-se ao ambiente. Esse deve ser capaz de detectar outros dispositivos que adentrem a ele. A plataforma proposta nesta dissertação utiliza esses conceitos, visto que o *middleware* poderá detectar os dispositivos eletrônicos no ambiente sem a percepção do usuário sobre a sua existência.

A computação móvel permite que seus usuários tenham acesso a serviços independente de sua localização, podendo, inclusive, estar em movimento. Tecnicamente, é um conceito que envolve processamento, mobilidade e comunicação sem fio. A ideia é ter acesso à informação em qualquer lugar e a qualquer momento. Além da mobilidade física dos dispositivos, os softwares deverão ser dotados de mobilidade lógica. A plataforma proposta neste trabalho também utiliza esse conceito, visto que o controle remoto universal é um aplicativo instalado em um dispositivo móvel.

A Computação Ubíqua, que surge como o resultado da união de computação pervasiva com a móvel, irá ampliar a interação entre objetos, dispositivos e serviços

para a criação do ambiente de contexto. A sua principal característica é a invisibilidade. Para Wakkary *et al.* (2005), essa invisibilidade é baseada no uso de sensores e atuadores embarcados no ambiente.

Ainda segundo Wakkary *et al.* (2005), sensores e atuadores são os principais responsáveis nas tomadas de decisão durante as interações com os usuários, fornecendo modalidades de comunicações naturais, de forma que os usuários consigam interagir como se estivessem comunicando-se com seres humanos.

Para Salarian *et al.* (2012), os sensores são dispositivos que possuem a capacidade de detectar ou captar ações ou estímulos do ambiente, respondendo de forma adequada. Por conseguinte, eles conseguem interagir com o usuário e se estiverem associados a outros dispositivos embarcados. Já no caso dos atuadores, eles poderão reagir a um determinado estímulo do indivíduo ou a alguma perturbação no ambiente.

Os atuadores são os dispositivos que convertem a energia elétrica, pneumática ou hidráulica em mecânica. Conseqüentemente, eles estão preparados para realizar alguma ação assim que são acionados (SALARIAN *et al.*, 2012). Eles surgiram com o conceito de hardware aberto, desta forma, o indivíduo tem a possibilidade de integrar um hardware modularizado para diferentes funcionalidades (SATRIA *et al.*, 2015).

2.5 A EVOLUÇÃO E IMPORTÂNCIA DOS SMARTPHONES

Contemplando os conceitos de Automação Residencial e Computação Ubíqua apresentados anteriormente nesta dissertação, é utilizado um aplicativo móvel para *smartphones* com a função de um controle remoto universal para o gerenciamento das ações dos atuadores no cenário de interação. Desta forma, essa seção discute a evolução e importância dos *smartphones*.

Os *smartphones* são aparelhos que, atualmente, são equipados com processadores equivalentes aos *desktops* e *laptops*, apresentam um bom desempenho e conseguem executar os mais variados tipos de aplicativos. Com a sua vasta utilização nas mais diferentes áreas, é um dos principais responsáveis pelo avanço de um conceito

computacional chamado de Tecnologias Convergentes, onde um dispositivo pode concentrar diferentes tipos de mídias e aplicativos (DO *et al.*, 2011) (KRONBAUER; SANTOS, 2011).

Um exemplo da relação dos *smartphones* com as Tecnologias Convergentes pode ser obtido pelo número de utilitários que um *smartphone* disponibiliza atualmente. Com ele é possível ouvir música, assistir filmes, pagar contas, falar ao telefone, navegar na Internet, acessar às redes sociais, entre outros. Há alguns anos atrás, essas funcionalidades dependiam de inúmeros aparelhos diferentes, tais como: rádio, televisão, telefone, computador, etc. Com a evolução do conceito de Tecnologias Convergentes, ocorreu a unificação de diferentes equipamentos nos *smartphones*, tornando-os mais atrativos e úteis para seus usuários.

No ano de 1993, a IBM lançou o Simon Personal Communicator (Figura 9), que tinha uma memória RAM de 1 MB e o processador operava em uma velocidade de 16 MHz. O Simon agregava as funcionalidades de um Assistente Digital Pessoal (PDA – *Personal Digital Assistant*), de um aparelho de fax, possuía ainda a disponibilidade para chamadas telefônicas e tinha uma tela sensível ao toque, o que era uma grande evolução para a época (MARTIN, 2014; REED, 2010).

Figura 9 - Simon Personal Communicator



Fonte: Martin (2014).

No ano de 1996, um novo aparelho surgiu, o Palm Pilot (Figura 10), que possuía um processador Motorola 68328 com velocidade de 16 MHz e 128 KB de memória RAM. Além disso, esse aparelho possuía a tela sensível ao toque e a possibilidade de instalação de aplicativos, porém não fazia ligações, funcionando apenas como um computador de mão (REED, 2010).

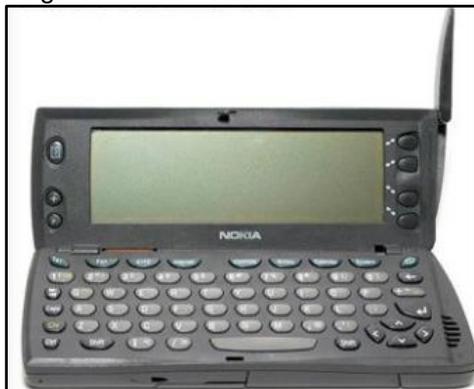
Figura 10 - Palm Pilot



Fonte: Reed (2010).

Nesse mesmo ano, a Nokia lançou o Nokia 9000 Communicator, que possuía algumas novidades que foram utilizadas pelos *smartphones* atuais, tais como: as telas deslizantes, teclado e suporte a diversos protocolos de rede, dentre eles podem ser destacados o SMTP, o POP3 e o MIME1 (MARTIN, 2014). No ano de 1998, foi lançada uma atualização desse aparelho, que foi chamado de Nokia 9110 Communicator, (Figura 11) ele trouxe algumas atualizações no seu sistema operacional e aplicativos (REED, 2010).

Figura 11 - Nokia 9110 Communicator



Fonte: Reed (2010).

No ano de 1999, a empresa Research In Motion, lançou o BlackBerry 850 (Figura 12), que suportava *e-mails*, acesso à internet, além de enviar e receber SMS, ele usava um modem que possibilitava sincronização com computadores pessoais e disponibilizava porta de Barramento Serial Universal (USB - *Universal Serial Bus*) (MARTIN, 2014) e (CRACKBERRY, 2016).

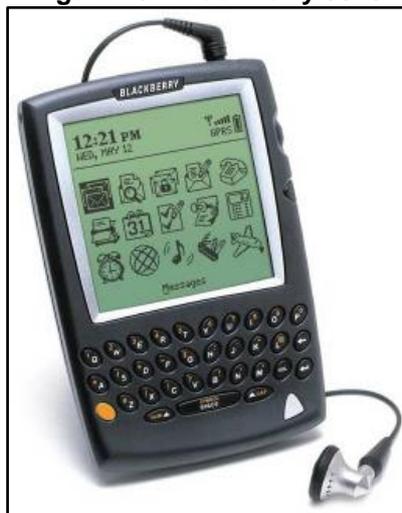
Figura 12 - BlackBerry 850



Fonte: Crackberry (2016).

No início de 2002, foi lançado o primeiro modelo que tinha possibilidade de se comunicar através de chamadas telefônicas, porém exigia o uso do fone de ouvido para realizar essa ação. Esse aparelho chamava-se *BlackBerry 5810*, que pode ser visto na Figura 13 (REED, 2010) (CRACKBERRY, 2016).

Figura 13 - BlackBerry 5810



Fonte: Crackberry (2016).

O ano de 2003 foi marcado pela criação da empresa Handspring, que lançou no mercado o Treo 600, passando a ser mais conhecido quando começou a ser comercializado como PalmOne (Figura 14). Ele possuía um processador ARM de 144 MHz e a sua memória RAM era de 32 MB.

Figura 14 - Treo 600



Fonte: Phonedata (2016).

Este *smartphone* oferecia uma série de recursos integrados, tais como:

- a) Verificar o calendário durante a chamada telefônica;
- b) Discar diretamente a partir da lista de contatos;
- c) Possibilidade de tirar fotos;
- d) Enviar *e-mails*;
- e) Possuía um botão de navegação com cinco direções;
- f) Função que permitia acesso rápido às funções mais usuais do telefone (REED, 2010).

Outro marco importante na área de *smartphones* é o sistema operacional Android, criado em 2005 por Andy Rubin, da empresa Android Inc. O sistema operacional móvel foi desenvolvido a partir de um *kernel* com Linux, por conseguinte, é um sistema operacional de código aberto. A empresa Android Inc. foi comprada pela Google nesse mesmo ano e Rubin passou a ser o Diretor de Plataformas Móveis da Google (JACKSON, 2011).

Apesar do sistema ter surgido em 2005, ele passou a ter notoriedade no mercado apenas a partir de 2009. Segundo Reed (2010), apenas 7% dos usuários americanos de *smartphones* utilizavam-no. Porém, nos dias atuais, esse sistema operacional é utilizado pela grande maioria dos usuários. Isso é reflexo do sistema possuir código aberto, o que determinou que diversos outros fabricantes (Samsung, Sony, LG, Nokia) passassem a desenvolver novos tipos de *smartphones* compatíveis com ele.

Segundo Martin (2014), o ano de 2007 foi marcado pela popularização dos *smartphones*, pois foi nesse ano que surgiu o iPhone (Figura 15) da Apple Inc. Esse dispositivo foi muito superior em relação aos anteriores, pois o aparelho tinha um processador poderoso e uma grande capacidade de memória, permitindo o desenvolvimento de uma grande quantidade de aplicativos móveis, como nunca antes foi visto nessa área³.

Figura 15 - iPhone



Fonte: Reed (2010).

A sua mais nova versão é o *iPhone 6s*, lançada em 2015, que possui um processador de quatro núcleos com velocidade de processamento de 1,84 GHz, memória RAM de 2 GB e câmera de 12 megapixels, que permite gravações de vídeo FULL HD com 1080p.

³Disponível em: <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

No ano de 2009, surgiu o primeiro *smartphone* a utilizar o Android comercialmente. O dispositivo chamava-se Motorola Droid, também conhecido como Motorola Milestone (Figura 16) e foi criado pela empresa Verizon, conseguindo vender cerca de 1 milhão de unidades em apenas 74 dias (REED, 2010). O dispositivo possui um teclado embutido, porém sua tela é *touchscreen*, além disso, ele possui um processador com velocidade de 600 MHz e 256Mb de RAM (PHONEDATA, 2016).

Figura 16 - Motorola Droid / Milestone



Fonte: Phonedata (2016).

Em 2010, foi lançado o *smartphone* Sprint / EVO 4G (Figura 17), criado pela empresa HTC. Entre as novas funcionalidades, o *smartphone* tinha a capacidade de acesso a redes sem fio WiMax e 4G, que é um tipo de rede de alta velocidade muito comum nos Estados Unidos (REED, 2010). O *smartphone* tem um processador Scorpion de 1 GHz e 512Mb de memória RAM (PHONEDATA, 2016)

Figura 17 - Sprint



Fonte: Reed (2010).

Ao mesmo tempo que o hardware utilizado nos *smartphones* se tornou mais poderoso, os softwares embarcados nesses dispositivos também passaram por várias etapas evolutivas. Particularmente, a evolução das plataformas Android e iPhone OS é um dos fatores de maior impacto para o sucesso dos *smartphones*.

A maturidade das plataformas e a popularidade dos aplicativos estão dando às empresas um novo canal para promover produtos, oferecer novos serviços e expandir as formas de alcançar potenciais clientes. Além disso, a capacidade de comprar e baixar aplicativos diretamente para os *smartphones* tem provado ser um serviço eficiente, tanto para os consumidores, como para os desenvolvedores.

Os desenvolvedores são capazes de publicar seus aplicativos rapidamente e os usuários podem navegar em uma biblioteca composta por milhares de aplicações que abrangem, por exemplo, jogos, softwares educacionais e soluções empresariais. Ademais, os sistemas de classificação de produtos possibilitam aos usuários finais opinarem sobre a qualidade e preço das aplicações oferecidas. A facilidade de baixar aplicativos têm motivado as pessoas a buscar, com mais frequência, novos softwares que auxiliem na realização de suas tarefas diárias (HANSEN, 2012).

Com a evolução do hardware e software para *smartphones*, associada à facilidade de obtenção de aplicativos, que são utilizados frequentemente para auxiliar na realização das tarefas diárias das pessoas, e aliando-se o uso desses aplicativos à mobilidade do usuário, torna-se indispensável a sua utilização na área de Automação Residencial.

2.6 SISTEMAS OPERACIONAIS MÓVEIS

Os sistemas operacionais móveis estão intrinsecamente ligados com a evolução dos *smartphones*. Esses programas que funcionam como um elo de ligação entre os componentes existentes em um dispositivo móvel e os aplicativos utilizados pelo usuário (MENDONÇA *et al.*, 2011).

Atualmente, existe uma grande quantidade de sistemas operacionais móveis no mercado, tais como: Android, iOS, Windows Phone, Blackberry OS, Symbian e outros.

Dentre eles, dois sistemas operacionais móveis se destacam, o (i) Android, da Google e o (ii) iOS, da Apple. Nesta subseção serão abordados os conceitos básicos destes dois tipos de sistemas operacionais.

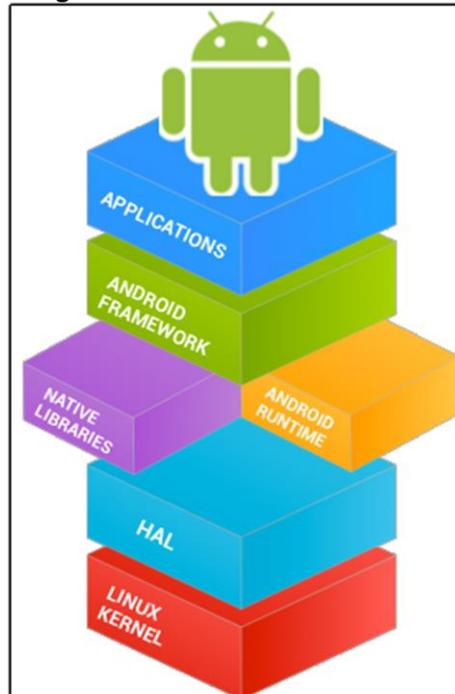
2.6.1 Sistema Operacional Móvel Android

O Sistema operacional móvel open source desenvolvido possui uma quantidade variada de componentes. Ele funciona em diversos modelos e tipos de dispositivos móveis, o que o tornou extremamente popular entre os usuários. Esse conjunto de fatores tornou tal sistema operacional o escolhido para ser utilizado no desenvolvimento do aplicativo do controle remoto universal neste trabalho (MENDONÇA *et al.*, 2011).

O sistema operacional móvel Android possui uma arquitetura dividida em seis camadas, que podem ser visualizadas através da Figura 18, são elas:

- a) *Applications* – Camada de interação entre os aplicativos e os usuários.
- b) *Applications frameworks* – Camada onde estão distribuídos componentes como *Activities*, *Views*, dentre outros.
- c) *Android Runtime* – Responsável pela execução dos aplicativos.
- d) *Native Libraries* – Local onde funciona o armazenamento das bibliotecas.
- e) HAL - Tem a responsabilidade de ativar as funcionalidades do dispositivo móvel, tais como: *bluetooth*, sensores, mídias, dentre outros.
- f) *Linux kernel* – Núcleo do sistema e camada de mais baixo nível, que serve como uma abstração entre o hardware e o software e toma as decisões do sistema operacional (ANDROID, 2016).

Figura 18 - Camadas do Android



Fonte: Android (2016).

2.6.2 Sistema Operacional Móvel iOS

O iOS (*iPhone Operation System* – Sistema Operacional do iPhone) é um sistema operacional móvel proprietário que foi concebido para funcionar em dispositivos móveis da empresa Apple.

A sua arquitetura foi dividida em quatro camadas (Figura 19), são elas:

- a) *Cocoa Touch* – Camada onde estão distribuídos os componentes do sistema;
- b) *Media* – Responsável por gerenciar os dados de vídeos e áudios;
- c) *Core Services* – Camada responsável pela distribuição de recursos com o catálogo de endereços, rede, segurança e banco de dados;
- d) *Core OS* – Onde está localizado o *kernel* do iOS (MENDONÇA *et al.*, 2011).

Figura 19 - Camadas do iOS



Fonte: Mendonça *et al.* (2011).

Uma das vantagens dessa abordagem é que o sistema operacional pode aproveitar todos os recursos existentes do hardware, visto que ele foi desenvolvido para trabalhar com esse hardware específico. Porém, isso pode limitar o uso dos aplicativos desenvolvidos nesse sistema para um grupo restrito de pessoas. Essa foi uma das principais razões que descartaram o uso de tal sistema operacional para ser utilizado como base no desenvolvimento do controle universal remoto utilizado no dispositivo móvel deste projeto.

2.7 CONSIDERAÇÕES FINAIS

No decorrer deste capítulo, ficou evidente a necessidade de aliar a Automação Residencial com a Computação Ubíqua, para a criação de ambientes que podem ser gerenciados através de controles universais embarcados em *smartphones*. Além disso, foi constatado que a Automação Residencial surgiu e baseou a sua evolução na miniaturização da eletrônica e no aumento da capacidade de processamento dos dispositivos computacionais. Outro ponto preponderante, para tornar essa tecnologia viável e acessível para as pessoas de diferentes níveis sociais, foi o barateamento do hardware utilizado para a implementação dos cenários de interação.

Deste modo, foram apresentados os conceitos existentes na área de computação ubíqua, visto que Weiser (1991) estimou como seria a forma de comunicação dos usuários e computadores há mais de 30 anos. Um exemplo disso é a própria solução proposta neste trabalho, em que o usuário se comunica com os dispositivos eletrônicos através de um controle remoto universal construído como um aplicativo

móvel, tomando como base uma forma natural de interação, que é a utilização de toques em um *smartphone*.

Finalmente, foi apresentada a evolução e importância dos dispositivos móveis do seu início até os dias atuais e os principais conceitos dos sistemas operacionais móveis, focando no Android e iOS. Desta maneira, pôde-se perceber que essa área sofreu uma enorme evolução e que, a partir do ano de 2007, ela tornou-se popular, principalmente após o surgimento dos *smartphones*, facilitando o uso de seus dispositivos para inúmeras áreas, incluindo a Automação Residencial. No próximo capítulo, serão apresentados os trabalhos sobre controle remotos universais para ambientes residenciais, além da evolução dos controles remotos universais embarcados em *smartphones*.

3 TRABALHOS SOBRE CONTROLES REMOTOS UNIVERSAIS PARA AMBIENTES RESIDENCIAIS

Neste capítulo é abordado o estudo dos trabalhos sobre controle remotos universais para ambientes residenciais. Além disso, são exibidos os pontos mais importantes de cada um dos trabalhos pesquisados. Na conclusão do capítulo, são apresentados alguns resultados que enfatizam as tendências da área em termos de tecnologias utilizadas para a construção de plataformas que suportem interações por meio de dispositivos móveis.

3.1 INTRODUÇÃO

O desenvolvimento dos dispositivos móveis evoluiu ao longo dos anos. Inicialmente, eles eram conhecidos como celulares e possuíam hardware limitado, além de poucos aplicativos, sendo que o usuário precisava utilizar diversos botões para operá-lo (RIEKKI *et al.*, 2008). A partir de 1993, começaram a ser criados aparelhos com telas deslizantes e aplicativos instaláveis (REED, 2010) (MARTIN, 2014). Entretanto eles só se popularizaram a partir de 2007, com o surgimento do iPhone, que foi o *smartphone* criado pela Apple (DOUGHTERY, 2011 *apud* JOHNSON *et al.*, 2012).

A área de Automação Residencial beneficiou-se dessa popularização, pois tornou possível o surgimento de novos aplicativos com as funcionalidades de controles remotos universais instalados em *smartphones*. Esses aplicativos funcionam como gestores dos dispositivos eletrônicos espalhados no ambiente residencial. Nesse sentido, foi realizada uma pesquisa bibliográfica para identificar as principais características deles.

3.2 PLANEJAMENTO E RESULTADO DO ESTADO

Com o intuito de não ignorar trabalhos científicos possivelmente importantes para a investigação dos trabalhos sobre controle remotos universais para ambientes residenciais, foram adotadas cinco bases científicas:

- a) ACM *Digital Library* – em modo “*advanced search*.” Com filtro: *published since = “2006”, publisher. “ACM” e content formats: “PDF”*.
- b) IEEE *Xplore Digital Library* – em modo “*advanced search*”
- c) Springer – no modo “*advanced search*”.
- d) Scopus – em modo “*advanced search*.” Com o filtro: *subject area=“computer science.”*
- e) Google Acadêmico.

As *strings* de busca utilizadas para a execução das pesquisas podem ser contempladas a seguir:

- a) "*universal remote control*" – "*universal mobile application*" – "*universal control smartphone*" – "*remote system*" – "*touchscreen phone*" – "*universal remote*"
- b) "*smart home*" – "*residential automation*" – "*automated home*" – "*home automation system*" – "*home automation*" – "*home remote system*"

Para realizar a pesquisa, os operadores usados foram o OR e AND. Desta maneira, o OR foi usado para separar as palavras-chaves que possuem o mesmo sentido, por exemplo: *residential automation OR automated home*. O operador AND foi utilizado para separar as palavras com sentidos diferentes, por exemplo, *universal mobile application AND home automation*. Uma das *strings* de busca construída pode ser observada a seguir:

- ("*universal remote control*" OR "*universal mobile application*" OR "*smartphone universal control*" OR "*remote system*" OR "*touchscreen phone*" OR "*universal remote*") AND ("*smart home*" OR "*residential automation*" OR "*automated home*" OR "*home automation system*" OR "*home automation*" OR "*home remote system*").

Após a realização da pesquisa bibliográfica foram selecionados vinte trabalhos acadêmicos, além de dez produtos comerciais, formando trinta soluções que foram consideradas como trabalhos na mesma temática desta dissertação. O Quadro 2 exhibe a relação dos vinte trabalhos acadêmicos encontrados e o Quadro 3 apresenta os dez produtos comerciais selecionados.

Quadro 2 – Artigos selecionados

AUTOR	TÍTULO
Nichols e Mayers (2006)	Controlling a Home and Office Appliances with Smartphones
Riekki <i>et al.</i> (2008)	Universal Remote Control for the Smart World
Bittins <i>et al.</i> (2010)	Supervision and Regulation of Home Automation Systems with Smartphones
Rajabzadeh <i>et al.</i> (2010)	A Mobile Application for Smart House Remote Control System
Koo <i>et al.</i> (2010)	RURC: RF4CE Based Universal Remote Control Framework using Smartphone
Zhong <i>et al.</i> (2011)	Smarthome on Smartphone
Panth e Jivani (2011)	Home Automation System (HAS) using Android Mobile Phone
Chueh e Fanjiang (2012)	Universal Remote Control on Smartphone
Jadhav <i>et al.</i> (2012)	Universal Mobile Application Development (UMAD) on Home Automation
Yan e Shi (2013)	Smart Living Using Bluetooth Based Android Smartphone
Leitte <i>et al.</i> (2013)	Automadroid, Automação Residencial com Dispositivos Móveis
Mareli <i>et al.</i> (2013)	Um Framework de desenvolvimento de aplicações ubíquas em ambientes inteligentes
Das <i>et al.</i> (2013)	Home Automation and Security Mobile Devices
Lamine e Abid (2014)	Remote Control of a domestic equipment from an Android application based on Raspberry Pi Card
Roy e Roy (2014)	Design of Smart Universal Remote Using Mobile for Home Automation
Pooja <i>et al.</i> (2014)	Smart Home Using Android Application
Ahsan <i>et al.</i> (2014)	Unicon Remote Central Model - A Mobile System for Assistive Technology
Shastri (2014)	Home automation system with universally used mobile application platform
Byun <i>et al.</i> (2015)	Development of remote control dongle
Satria <i>et al.</i> (2015)	The framework of Home Remote Automation System Based on Smartphone

Fonte: Autoria própria do autor desta dissertação (2016).

Quadro 3 - Produtos Comerciais selecionados

AUTOR	TÍTULO
Irdroid (2016)	Irdroid v1.0
Irdroid (2016)	Irdroid Wi-Fi v3.0
Mobimax (2016)	L5 Universal Remote
Logitech (2016)	Harmony Smart Control
Ilounge (2016)	FLPR Universal Remote Control
CNET (2016)	Gear4 UnityRemote Universal Remote Control
Irule (2016)	IRule
Melloware (2016)	X10 Commander
Samsung (2016)	Samsung AllShare Control
Wavespark (2016)	IR Universal Remote

Fonte: Autoria própria do autor desta dissertação (2016).

3.3 CONTROLES REMOTOS UNIVERSAIS COMERCIAIS

Os dispositivos móveis possuem aplicativos disponíveis no mercado com diversas funcionalidades, inclusive a capacidade de torná-los controles remotos universais. Nesta seção, são apresentados as principais abordagens referentes aos controles remotos universais embarcados em *smartphones* disponibilizados comercialmente.

O Irdroid v1.0 é um adaptador infravermelho de baixo custo financeiro, para quem não se preocupa com questões estéticas (Figura 20). O dispositivo pode ser interligado ao conector de áudio de aparelhos Android e exige uma pilha AAA para funcionar corretamente.

Figura 20 - Irdroid v1.0



Fonte: Irdroid (2016).

O acessório funciona com o auxílio de um aplicativo *open source* pré-configurado para trabalhar com centenas de aparelhos de fabricantes diferentes, como, por exemplo, Samsung, LG, Motorola, Sony, Panasonic e Phillips. Segundo o fabricante, o adaptador possui alcance máximo de dez metros (IRDROID, 2016).

O Irdroid Wi-Fi v3.0 (Figura 21) é uma versão mais moderna do dispositivo descrito anteriormente. Para que o produto funcione, basta conectá-lo a uma saída USB do aparelho, que será controlado remotamente e estabelecerá uma comunicação através do aplicativo disponível para a plataforma Android (IRDROID, 2016).

Figura 21 - Irdroid Wi-Fi v3.0



Fonte: Irdroid (2016).

O L5 Universal Remote (Figura 22), que foi lançado em 2012 pela Mobimax, tem a capacidade de transformar aparelhos, como o iPod Touch e o iPhone, em controles remotos universais. Basta encaixar o acessório ao conector inferior dos aparelhos para começar a usá-los e controlar televisões, *blu-rays* ou receptores de TV por assinatura (MOBIMAX, 2016).

Figura 22 - L5 Universal Remote



Fonte: Mobimax (2016).

O adaptador vem com um aplicativo configurável, que permite alterar a disposição e o *design* dos botões de acordo com as preferências de seus usuários. Além disso, o produto é compatível com o iPhone 3G, 3GS, 4, 4S, 5S e 6S e com todas as gerações do iPod Touch.

O Harmony Smart Control (Figura 23), da Logitech, é uma opção mais elaborada, pois disponibiliza aplicativos para as plataformas iOS e Android, além de vir com um controle remoto universal independente do *smartphone*. O produto é constituído por um adaptador infravermelho semelhante a um roteador, com um extensor responsável pela captação de sinais e uma aplicação proprietária desenvolvida pelo fabricante (LOGITECH, 2016).

Figura 23 - Harmony Smart Control



Fonte: Logitech (2016).

O FLPR Universal Remote Control (Figura 24), da New Potato Technologies, conecta-se diretamente à saída *lightning* do iPod Touch, iPad e iPhone. Os dispositivos móveis utilizam um aplicativo que disponibiliza sessenta e cinco códigos de controles remotos pré-programados, o que garante a compatibilidade com a maioria dos aparelhos disponíveis atualmente no mercado.

Figura 24 - FLPR Universal Remote Control



Fonte: Ilounge (2016).

O Gear4 UnityRemote Universal Remote Control (Figura 25) permite a configuração de ações pré-determinadas que operam vários dispositivos de forma simultânea. Com essa solução, o indivíduo pode utilizar um botão para manipular um dispositivo eletrônico remotamente. Assim, o produto pode ser configurado para trabalhar com eletrônicos que não constem em seu banco de dados (CNET, 2016).

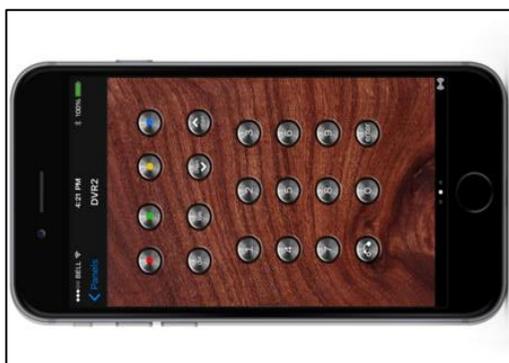
Figura 25 - Gear4 UnityRemote Universal Remote Control



Fonte: CNET (2016).

O iRule (Figura 26), da Apple, é disponibilizado para a plataforma iOS. Ele permite o controle dos dispositivos eletrônicos de uma residência, utilizando a tecnologia *Wi-Fi*. Feita a detecção dos dispositivos sem fio no ambiente, as informações são armazenadas em uma nuvem, o que possibilita que o usuário consiga controlar dispositivos a longas distâncias (IRULE, 2016).

Figura 26 - iRule



Fonte: iRule (2016).

O X10 Commander é um controle remoto universal que funciona para Android ou iOS. Ele utiliza a tecnologia de infravermelho com comandos simples de comunicação, tais como, ligar ou desligar um dispositivo (MELLOWARE, 2016).

A Figura 27 apresenta um exemplo do aplicativo X10 Commander em sua versão para Android. Nela, é possível ver uma lista de lâmpadas espalhadas em vários ambientes de uma determinada residência. A apresentação dos dispositivos eletrônicos é feita de forma que a interação do usuário com o ambiente seja amigável, visto que ele poderá ligar ou desligar uma lâmpada remotamente através de seu controle universal (MELLOWARE, 2016).

Figura 27 - X10 Commander

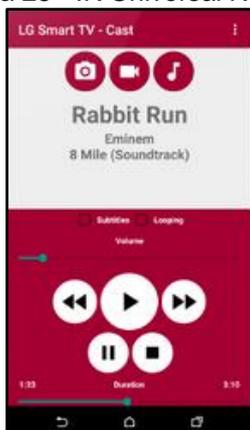


Fonte: Melloware (2016).

O IR Universal Remote é um aplicativo que funciona para Android e foi criado pela empresa WaveSpark. Ele utiliza a tecnologia de rede em infravermelho para se

comunicar diretamente com os dispositivos eletrônicos do ambiente. Sua interface é amigável e intuitiva, como pode ser visualizado na Figura 28. Desta maneira, o usuário tem a possibilidade de gerenciar os dispositivos eletrônicos remotamente, mas apenas com equipamentos que tenham receptores em infravermelho (WAVESPAK, 2016).

Figura 28 - IR Universal Remote



Fonte: Wavespark (2016).

Além dos controles remotos universais identificados neste trabalho, existem outros que funcionam apenas para os dispositivos pertencentes a uma mesma empresa, como é o caso do Samsung AllShare Control (Figura 29). Ele possibilita o controle de todos os aparelhos da Samsung através de um aplicativo instalado em um dispositivo móvel, funcionando em uma rede *Wi-Fi* (SAMSUNG, 2016).

Figura 29 - Samsung AllShare Control



Fonte: Samsung (2016).

3.4 CONTROLES REMOTOS UNIVERSAIS ACADÊMICOS

Esta seção tem como objetivo abordar os controles remotos universais acadêmicos, sendo que os trabalhos estão descritos em subseções de acordo com o número de camadas que a plataforma apresenta.

3.4.1 Controles Remotos Universais Multicamadas

Por meio da pesquisa bibliográfica, foram identificados artigos que utilizaram o conceito de multicamadas proposto por Edwards e Foreword (1998) e Levinson (2003). Esses trabalhos podem ser visualizados através do Quadro 4.

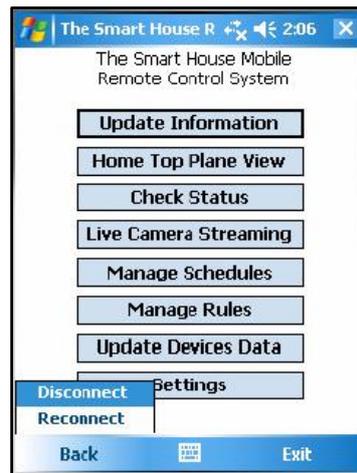
Quadro 4 - Plataformas Multicamadas

PRODUTO	AUTOR
Smart House Controlling System	(Rajabzadeh <i>et al.</i> , 2010)
KNX Home Automation	(Bittins <i>et al.</i> , 2010)
RF4CE-based Universal Remote Control	(Koo <i>et al.</i> , 2010)
HouseGenie	(Zhong <i>et al.</i> , 2011)
SmartAndroid	(Mareli <i>et al.</i> , 2011)
Universal Mobile Application Development	(Jadhav <i>et al.</i> , 2012)
Home Automation and Security for Mobile Devices	(Das <i>et al.</i> , 2013)
Smart Universal Remote Using Mobile	(Roy e Roy, 2014)

Fonte: Autoria própria do autor desta dissertação (2016).

O controle remoto universal Smart House Controlling System (Figura 30) foi criado pelos autores Rajabzadeh *et al.* (2010) e funciona no sistema operacional Windows Phone. Nele, o usuário consegue visualizar e gerenciar os dispositivos eletrônicos que estão distribuídos no ambiente, porém a interface é limitada a quatro padrões de telas.

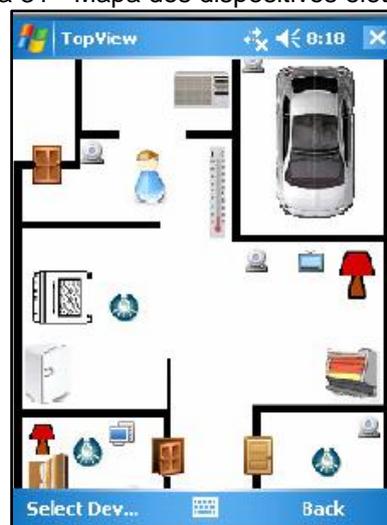
Figura 30 - Smart House Controlling System



Fonte: Rajabzadeh *et al.* (2010).

Um fato interessante sobre o Smart House Controlling é que ele permite a visualização dos dispositivos eletrônicos espalhados no ambiente através dos cômodos da residência, como pode-se ver na Figura 31.

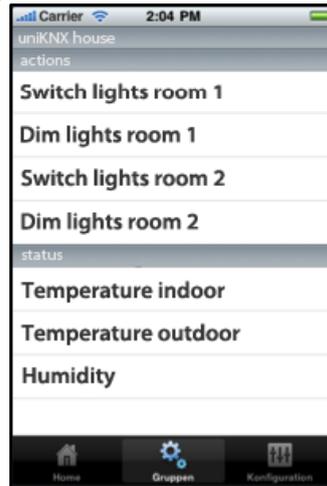
Figura 31 - Mapa dos dispositivos eletrônicos



Fonte: Rajabzadeh *et al.* (2010).

Outro trabalho que apresenta o conceito de multicamadas e merece destaque é o KNX Home Automation (Figura 32), apresentado por Bittins *et al.* (2010). Ele é uma plataforma que funciona para o sistema operacional iOS e permite o controle dos dispositivos eletrônicos de uma residência utilizando a tecnologia *Wi-Fi*. Além disso, ele possibilita a comunicação externa do usuário através da tecnologia em nuvem.

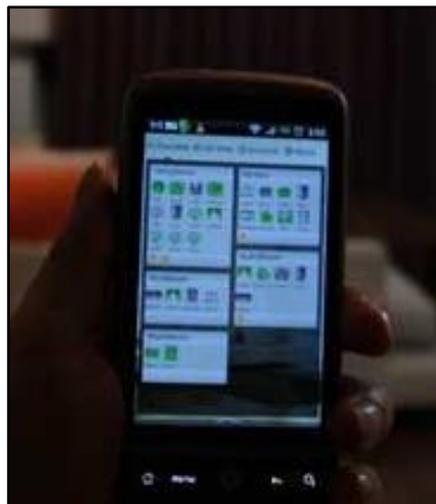
Figura 32 - KNX Home Automation



Fonte: Bittings *et al.* (2010).

A Figura 33 apresenta o HouseGenie, criado pelos autores Zhong *et al.* (2011). Nele, o gerenciamento dos dispositivos eletrônicos do ambiente segue um conjunto de quatro camadas, são elas: (i) Camada de interface, definida pelo controle remoto universal; (ii) camada de serviços, definida pelo *middleware* para o controle e tratamento de aspectos contextuais; (iii) Camada de dispositivos, definida pelos dispositivos eletrônicos do ambiente; e (iv) camada adaptativa, que descreve quais são as tecnologias de rede utilizadas pelos dispositivos eletrônicos.

Figura 33 - HouseGenie



Fonte: Zhong *et al.* (2011).

3.4.2 Controles Remotos Universais com Duas Camadas

A pesquisa bibliográfica realizada para esta dissertação também identificou artigos que não utilizaram o conceito de múltiplas camadas. Na realidade, seus autores optaram por uma solução mais simplificada, com a utilização de apenas duas camadas para a realização do gerenciamento de dispositivos eletrônicos no ambiente através de um controle remoto universal. Esses trabalhos podem ser visualizados através do Quadro 5.

Essas plataformas caracterizam-se pela não utilização da camada intermediária, onde está localizado o *middleware*, sendo assim, o desenvolvimento dessa solução torna-se menos trabalhosa e mais eficiente para o desenvolvedor. Por outro lado, a falta do *middleware* pode limitar a plataforma a uma única modalidade de interação, inviabilizando a utilização da voz, gestos e toques em um mesmo projeto arquitetural.

Quadro 5 - Plataformas com Duas Camadas

PRODUTO	AUTOR
Personal Universal Controller	(Nichols e Myers, 2006)
REACHeS	(Riecki <i>et al.</i> , 2008)
Home Automation System	(Panth e Jivani, 2011)
CPUC	(Chueh e Fanjiang, 2012)
Smart Living	(Yan e Shi, 2013)
HomeTec	(Leitte <i>et al.</i> , 2013)
Home Automation System	(Shastri, 2014)
Smart Home System	(Pooja <i>et al.</i> , 2014)
Remote Control of a Domestic Equipment	(Lamine e Abid, 2014)
UniCon Remote Control Model	(Ahsan <i>et al.</i> , 2014)
Remote Control Dongle	(Byun <i>et al.</i> , 2015)
Smart Automation Home	(Satria <i>et al.</i> , 2015)

Fonte: Autoria própria do autor desta dissertação (2016).

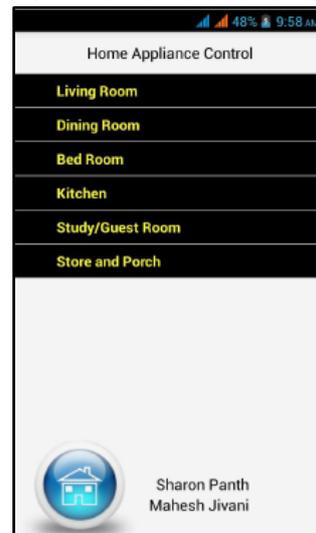
O HomeTec, proposto por Leitte *et al.* (2013), utiliza um controle remoto universal, que funciona em um sistema operacional Android e microcontroladores Arduino. A solução promete um sistema altamente confiável, de baixo custo financeiro e eficiente. Segundo os autores, a plataforma está dividida da seguinte forma:

- a) Android – Responsável pelo envio de comandos e interação do usuário.

- b) Rede – Responsável por entrega de pacotes e conectividade entre o controle universal e os atuadores. Nesta camada, foi definido o uso da *Wi-Fi* como tecnologia padrão para a conectividade da plataforma.
- c) Atuadores – Responsável pela interpretação de comandos e acionamentos dos equipamentos;
- d) Acionamentos – Responsável pelos dispositivos eletrônicos do ambiente, que podem ser motores, lâmpadas, alarmes, eletrodomésticos, entre outros (LEITTE *et al.*, 2013).

O controle remoto universal Home Automation System (Figura 34) foi criado por Panth e Jivani (2011) com o objetivo de gerenciar diretamente os dispositivos eletrônicos através da utilização da tecnologia sem fio, *bluetooth*. O aplicativo funciona através da plataforma Android e sua interface apresenta uma visualização por meio das localidades da residência.

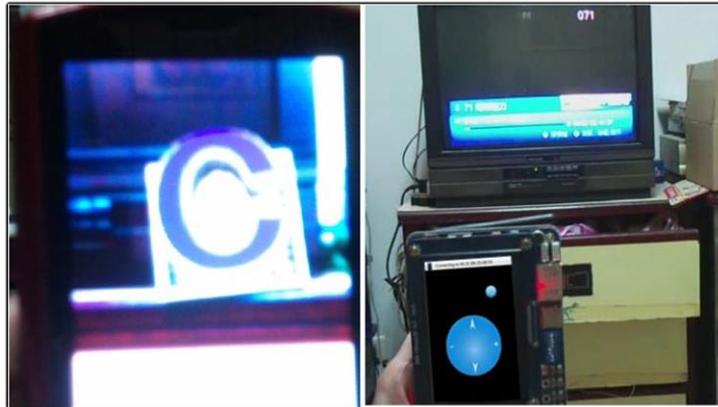
Figura 34 - Home Automation System



Fonte: Panth e Jivani *et al.* (2011).

O CPUC (Context-Aware Personal Interface for Universal Remote Control) é um controle remoto universal, que permite aos usuários o gerenciamento dos dispositivos eletrônicos remotamente, como pode ser visto na Figura 35 (CHUEH ; FANJIANG, 2012).

Figura 35 - CPUC



Fonte: Chueh e Fanjiang (2012).

O CPUC tem sua arquitetura em cinco camadas, são elas:

- a) *Appliance Server* – Função embutida no controle remoto universal, que é responsável por encontrar os dispositivos eletrônicos no ambiente;
- b) *JSON Parser* – Coleta informações dos atuadores do ambiente através do uso de JSON ou XML;
- c) *Priority Manager Service* – Serve para atribuir níveis de prioridade aos dispositivos eletrônicos no ambiente;
- d) *Context Manager Service* – Recebe e envia os comandos do controle remoto universal para os atuadores sem níveis de prioridade;
- e) *Graphical UI Generator* – Responsável pelo funcionamento do controle remoto universal.

Apesar da apresentação em cinco camadas, o controle remoto universal utiliza a camada *Appliance Server* para se comunicar com os atuadores, que são representados graficamente pela camada *Graphical UI Generator*. As camadas *JSON Parser*, *Priority Manager Service* e *Context Manager Service* apenas atribuem mais funcionalidades ao controle remoto universal.

O UniCon Smart Remote Control (Figura 36) é uma plataforma que possui duas camadas, sendo que o controle remoto universal foi desenvolvido através de um conjunto de quatro módulos:

- a) Módulo de comunicação – Responsável pela comunicação física entre o dispositivo eletrônico e o aplicativo móvel;
- b) Módulo de procura – Realiza uma varredura no ambiente em busca de novos dispositivos que possam ser acionados.
- c) Módulo base de dados – Armazena informações contextuais.
- d) Módulo de atualização – Responsável pelo cadastro de novos produtos ou clientes no ambiente (AHSAN *et al.*, 2014).

Figura 36 - UniCon Smart Remote Control



Fonte: Ahsan *et al.* (2014)

O Remote Control of a Domestic Equipment (Figura 37) é um controle remoto universal para a plataforma Android, que utiliza *Wi-Fi* para se comunicar com atuadores Raspberry Pi acoplados aos dispositivos eletrônicos. Esses atuadores têm como função receber os comandos e executar as ações nos dispositivos eletrônicos disponíveis no contexto (LAMINE ; ABID, 2014).

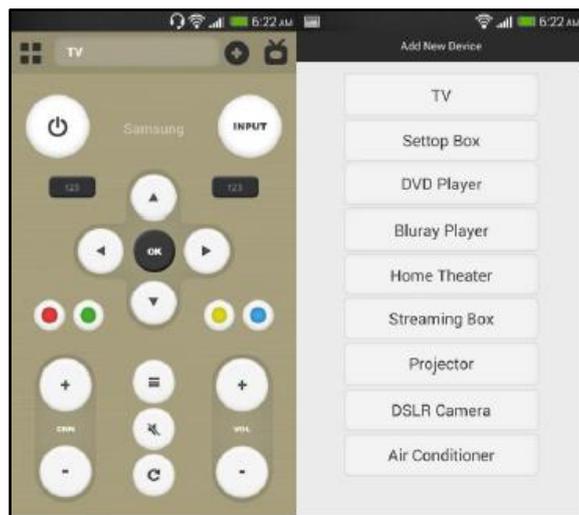
Figura 37 - Remote Control of a Domestic Equipment



Fonte: Lamine e Abid (2014).

O Remote Control Dongle (Figura 38), criado por Byun *et al.* (2015), é um dispositivo acoplado ao *smartphone*, permitindo que ele tenha as mesmas funcionalidades de um controle remoto universal. Para controlar o Remote Control Dongle, o usuário utiliza um aplicativo instalado em seu *smartphone* que permite o gerenciamento direto de todos os dispositivos do ambiente.

Figura 38 - Interface Fixa



Fonte: Byun *et al.* (2015).

O aplicativo comunica-se com o adaptador tornando, assim, possível a seleção de diferentes dispositivos eletrônicos. Entretanto o aplicativo disponibiliza a mesma interface para todos os dispositivos existentes no cenário de interação, o que pode trazer confusão para os usuários no momento de utilizar o controle remoto.

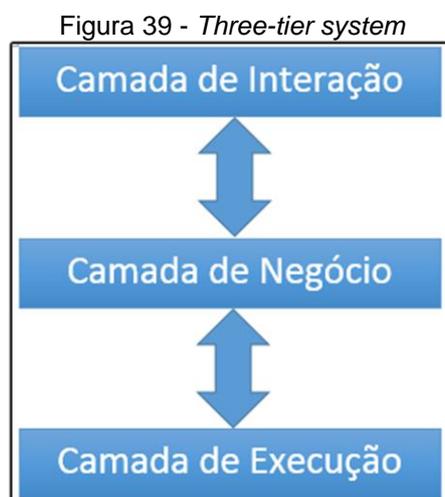
3.4.3 A Importância das Arquiteturas de Camadas

Para desenvolver uma solução na área de computação, é necessário definir a arquitetura que este sistema irá possuir. Para atingir esse objetivo, foram criados diversos tipos de arquiteturas que são classificadas de acordo com a divisão de camadas. A divisão baseada nesse tipo de método é importante, porque podem dividir mais adequadamente as funções de um aplicativo, infraestrutura ou plataforma (LEVINSON, 2003).

As arquiteturas baseadas na classificação por camadas estão divididas da seguinte maneira:

- a) Arquitetura monolítica – São aplicativos que possuem todas as suas funcionalidades definidas em apenas um módulo ou camada;
- b) Arquitetura cliente-servidor ou Duas Camadas – Sistema dividido em duas camadas, sendo que uma será representada pelo cliente (regra de apresentação) e outra pelo servidor (regra de negócio);
- c) Arquitetura de três camadas – Sistema dividido em três camadas independentes e cada uma delas executa uma função específica (interação, negócio e execução) (EDWARDS; FOREWORD, 1998);
- d) Arquitetura de n-camadas⁴ ou Multicamadas – São sistemas que possuem uma quantidade indefinida e ilimitada de camadas, sendo que cada uma delas irá possuir uma função especificada pelo desenvolvedor.

O trabalho proposto nesta dissertação utiliza um modelo de três camadas independentes, baseado no conceito de *Three-tier system* ou sistema de três camadas (EDWARDS; FOREWORD, 1998) (LAMINE; ABID, 2014), que pode ser visualizada na Figura 39.



Fonte: Edwards e Foreword (1998).

Conforme Edwards e Foreword (1998), as camadas possuem as seguintes funcionalidades:

⁴Disponível em: <http://n-tier.com>

- a) Camada de Interação – Possibilita a execução de comandos provenientes dos usuários, sendo responsável por prover mecanismos de captura das interações.
- b) Camada de Negócio – Responsável por traduzir requisições dos usuários em ações a serem executadas no cenário de interação. Também é conhecida como camada de conectividade, servidor de aplicação ou *middleware*.
- c) Camada de Execução – Responsável por receber as requisições da camada de negócio para que possa executar as funcionalidades do sistema.

3.5 ANÁLISE DAS ABORDAGENS UTILIZADAS

Durante a realização da revisão bibliográfica, foram analisadas as diferentes abordagens existentes para a construção de controles remotos universais embarcados em *smartphones*, com o objetivo de buscar as melhores alternativas para a construção do modelo e plataforma idealizados nesta dissertação.

O Quadro 6 apresenta as informações distribuídas em três colunas: (i) conceito, em que é descrito o principal assunto abordado nos trabalhos; (ii) descrição, na qual há um detalhamento do conceito identificado; e (iii) autores, responsáveis pelas publicações. O conjunto dos conceitos estudados forma a inter-relação entre as áreas envolvidas na construção de projetos para a automação residencial.

Quadro 6 - Principais conceitos relacionados à Automação Residencial e Controle Remoto Universal

CONCEITO	DESCRIÇÃO	AUTORES
Automação Residencial	Automação Residencial ou Domótica. Utilização de tecnologia embarcada em um ambiente, servindo para possibilitar ao usuário um aumento na qualidade de vida e segurança, além de tornar possível o uso racional dos recursos existentes.	(Bittins <i>et al.</i> , 2010) (Zhong <i>et al.</i> , 2011) (Das <i>et al.</i> , 2013) (Lamine e Abid, 2014) (Ahsan <i>et al.</i> , 2014)
Computação Ubíqua e Pervasiva	A computação está distribuída em todos os locais e momentos de forma transparente para seus usuários.	(Leitte <i>et al.</i> , 2013) (Mareli <i>et al.</i> , 2013)
Controle Remoto Universal Embarcado em <i>Smartphones</i>	Gerenciamento dos dispositivos internos de um ambiente através da utilização de um dispositivo móvel.	Todos os artigos pesquisados.

CONCEITO	DESCRIÇÃO	AUTORES
<i>Smartphones</i>	Dispositivos móveis computacionais capazes de realizar diversas tarefas através da instalação de aplicativos.	(Leitte <i>et al.</i> , 2013) (Mareli <i>et al.</i> , 2013) (Jadhav <i>et al.</i> , 2012)
Redes sem fio	Tecnologia que permite a comunicação entre dois ou mais dispositivos através de ondas eletromagnéticas.	Todos os artigos pesquisados.
Sistemas Operacionais Móveis	Software que gerencia os recursos do sistema e faz a interface entre o hardware e o usuário. Foi desenvolvido especificamente para dispositivos móveis.	Todos os artigos pesquisados.
JSON	<i>Framework</i> que permite a comunicação de dados primitivos de um sistema para outro em hardwares diferentes.	(Lamine e Abid, 2014) (Chueh e Fanjiang, 2012)
Microcontroladores	Pequeno computador com o formato de um circuito integrado utilizado para aplicações embarcadas.	(Pooja <i>et al.</i> , 2014) (Satria <i>et al.</i> , 2015)
XML	Conceito que funciona para unificar a comunicação entre diferentes sistemas.	(Shastri, 2014)
<i>Architecture at three levels</i> (Arquitetura em três níveis)	Modelo em que a plataforma é dividida em camadas independentes: (i) camada de interação; (ii) camada de negócio; e (iii) camada de acesso ao dado.	Lamine e Amid (2014)

Fonte: Autoria própria do autor desta dissertação (2016).

Após a identificação dos conceitos envolvidos no escopo da pesquisa, foram analisadas as tecnologias mais empregadas nos trinta trabalhos pesquisados para esta dissertação. Esse estudo ajudou na identificação e escolha de possíveis soluções tecnológicas para a construção dos componentes do modelo que será apresentado no próximo capítulo.

3.5.1 Análise referente aos sistemas operacionais utilizados

Quadro 7 apresenta os sistemas operacionais utilizados pelos controles universais existentes (acadêmicos e comerciais). Ele está separado em duas colunas: na

primeira, são listados os autores e o ano de cada trabalho; na segunda, são apresentados os sistemas operacionais.

Quadro 7 - Métrica de Sistema Operacional

AUTOR	SISTEMA OPERACIONAL
Nichols e Myers (2006)	Windows Phone
Riecki <i>et al.</i> (2008)	Symbian
<i>Bittins et al.</i> (2010)	Ios
Rajabzadeh <i>et al.</i> (2010)	Windows Phone
Koo <i>et al.</i> (2010)	Windows Phone
Zhong <i>et al.</i> (2011)	Android
Panth e Jivani <i>et al.</i> (2011)	Android
Chueh e Fanjiang <i>et al.</i> (2012)	Android
Jadhav <i>et al.</i> (2012)	Android, iOS e BlackBerry OS
Yan e Shi (2013)	Android
Leitte <i>et al.</i> (2013)	Android
Mareli <i>et al.</i> (2013)	Android
Das <i>et al.</i> (2013)	iOS
Lamine e Abid (2014)	Android
Roy e Roy (2014)	Android
Pooja <i>et al.</i> (2014)	Android
Ahsan <i>et al.</i> (2014)	Android
Shastri (2014)	Android, iOS e BlackBerry OS
Byun <i>et al.</i> (2015)	Android e iOS
Satria <i>et al.</i> (2015)	Android e iOS
Irdroid (2016)	Android
Irdroid (2016)	Android
Mobimax (2016)	iOS
Logitech (2016)	iOS
Ilounge (2016)	iOS
CNET (2016)	Android
Irule (2016)	iOS
Melloware (2016)	Android ou iOS
Wavespark (2016)	Android
Samsung (2016)	Android

Fonte: Autoria própria do autor desta dissertação (2016).

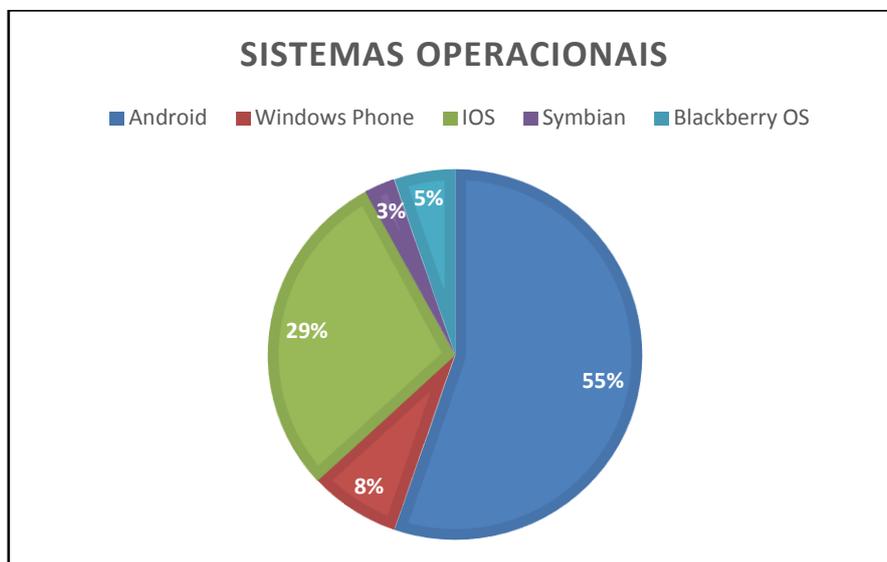
Quadro 7, verificou-se a seguinte situação:

- a) Vinte e uma propostas pesquisadas utilizaram o sistema operacional Android da Google.
- b) Onze propostas utilizaram o sistema operacional iOS da Apple.

- c) Três propostas utilizaram o sistema operacional Windows Phone, da Microsoft.
- d) Duas propostas utilizaram o sistema operacional Blackberry OS.
- e) Uma proposta utilizou o Symbian da Nokia.

Deve-se registrar que algumas das soluções pesquisadas tem a capacidade de funcionar em mais de um sistema operacional. Um exemplo é o sistema X10 Commander, da Melloware (2016). Ele funciona para Android e iOS, portanto o gráfico da Figura 40 irá contemplar mais de um sistema operacional para essa solução.

Figura 40 - Sistemas Operacionais Utilizados

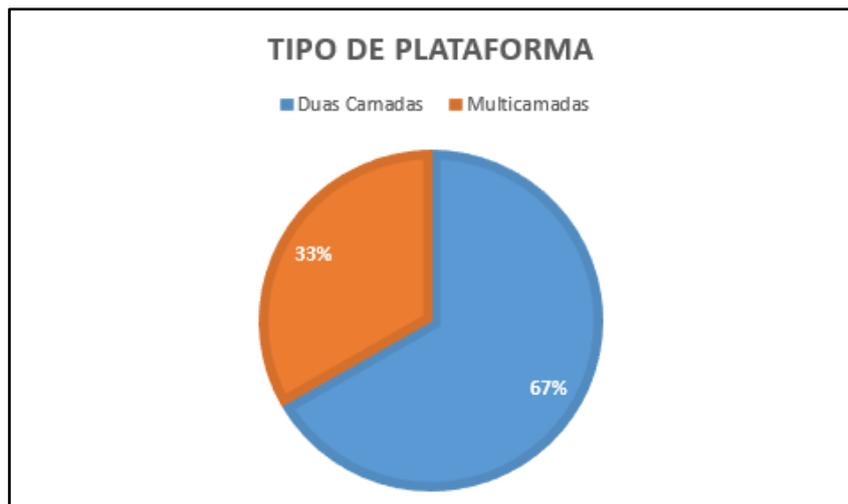


Fonte: Autoria própria do autor desta dissertação (2016).

O sistema operacional Android se tornou a principal escolha dos autores pesquisados, somando 22 projetos no total. Isso reflete a popularidade da plataforma Android, por ser de código aberto, ocasionando a sua adoção por um número maior de fabricantes de *smartphones*. O segundo colocado, foi o sistema operacional iOS da Apple, com 11 soluções existentes nos trabalhos pesquisados para esta dissertação.

3.5.2 Análise referente à utilização de *middleware*

Figura 41, pode ser observado a divisão percentual dos trabalhos pesquisados que utilizam uma infraestrutura em multicamadas ou em duas camadas.

Figura 41 - Quantidade de Trabalhos com *Middleware*

Fonte: Autoria própria do autor desta dissertação (2016).

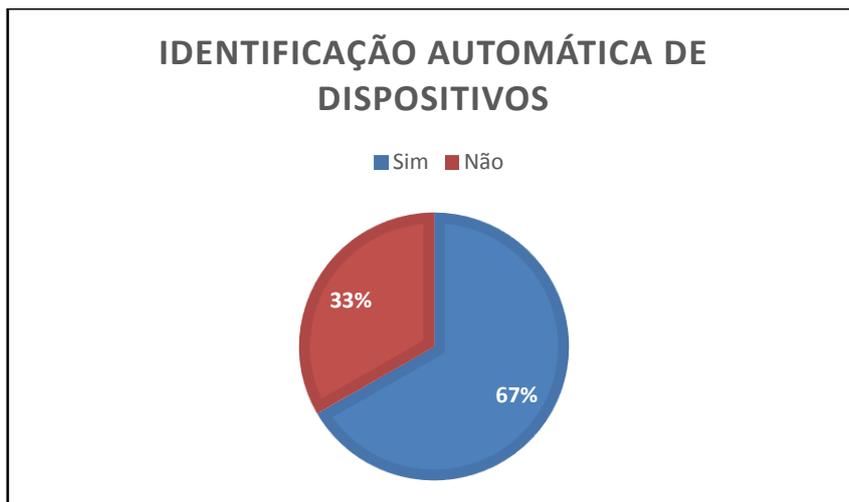
Figura 41, percebe-se que 20 trabalhos pesquisados utilizam uma estrutura em duas camadas, ou seja, não existe um *middleware* entre o controle universal e os dispositivos eletrônicos. Os outros 10 trabalhos possuem uma estrutura em multicamadas, conforme foi discutido nas seções 3.4.1 (Controles Remotos Universais Multicamadas) e 3.4.2 (Controles Remotos Universais de Duas Camadas).

A predominância de projetos com duas camadas pode ser justificada pela simplicidade do seu desenvolvimento, por outro lado, inúmeras limitações de extensibilidade podem ser observadas, tais como: inviabilidade de prover diferentes formas de interação (utilização de outras modalidades de interação) e restrição a tipos específicos de equipamentos, marcas e modelos.

3.5.3 Análise referente a identificação automática dos dispositivos no ambiente

O gráfico da Figura 42 apresenta o resultado em relação à capacidade das soluções propostas identificarem automaticamente os dispositivos eletrônicos disponíveis em um cenário de interação. Observando o gráfico, percebe-se que a maior parte das propostas pesquisadas faz a identificação automática dos dispositivos eletrônicos no ambiente, embora, em muitos casos, a identificação de determinados dispositivos seja restrita a uma determinada marca (fabricante) ou tipo de aparelho.

Figura 42 - Identificação de dispositivos no ambiente residencial



Fonte: Autoria própria do autor desta dissertação (2016).

3.5.4 Análise referente à *renderização* da interface no aplicativo móvel

A Figura 43, apresenta o tipo de construção ou *renderização* de interface do controle remoto universal nos trabalhos pesquisados. Isso é feito de forma a separar os que possuem *renderização* da interface de forma estática daqueles que possuem *renderização* da interface de forma dinâmica.

Figura 43 - Tipo de *Renderização* do Controle Remoto Universal

Fonte: Autoria própria do autor desta dissertação (2016).

Observando a Figura 43, percebe-se que 20 trabalhos pesquisados possuem a construção da interface dinâmica, enquanto que 10 soluções possuem uma interface estática. Esse resultado pode ser fundamentado pela preocupação dos desenvolvedores em criar soluções que sejam de fácil utilização para usuário, fazendo

com que ele tenha uma experiência de uso mais imersiva no contexto das interfaces disponibilizadas para as interações.

3.5.5 Análise referente à utilização de atuadores no cenário de interação

Os atuadores são elementos importantes na plataforma proposta neste trabalho, porém ao analisar a

Figura 44, percebeu-se que 21 soluções não os utilizam, sendo possível concluir que essa tecnologia ainda está começando a ser explorada pela área de controles remotos universais em ambientes residenciais.

Figura 44 - Atuadores encontrados na pesquisa



Fonte: Autoria própria do autor desta dissertação (2016).

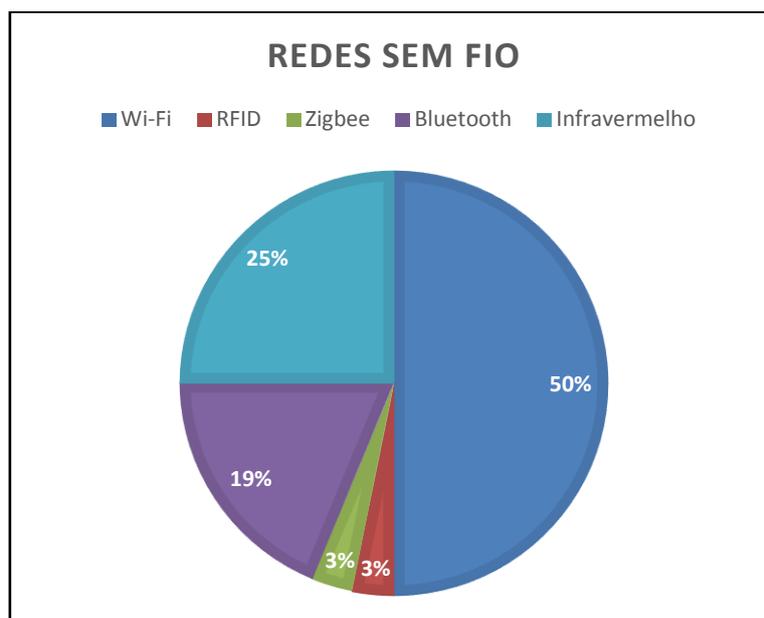
O uso de atuadores é vantajoso para o gerenciamento dos dispositivos eletrônicos, já que aumentam a capacidade de comunicação do controle remoto universal. Assim, os projetos que os utilizam podem abranger uma variedade maior de tecnologias de rede sem fio (*bluetooth*, *Wi-Fi* e infravermelho) para controlar os artefatos eletrônicos (que estão acoplados aos atuadores).

Além da flexibilidade referente a tecnologia de comunicação, a utilização de atuadores permite controlar remotamente dispositivos eletrônicos que não foram projetados com controle remoto, por exemplo, lâmpadas, portas, janelas, dentre outros. Isto ocorre porque os dispositivos eletrônicos estarão acoplados aos atuadores e poderão receber comandos remotamente por diferentes tecnologias de redes sem fio.

3.5.6 Análise referente às tecnologias de rede sem fio

Para um controle universal funcionar adequadamente, é necessário o uso de algum tipo de solução que possa prover a comunicação entre os componentes do sistema sem a utilização de fios. Desta forma, a Figura 45 apresenta o uso das tecnologias de rede utilizadas pelas soluções pesquisadas no estudo bibliográfico realizado durante a elaboração desta dissertação.

Figura 45 - Tecnologias de Rede sem Fio



Fonte: Autoria própria do autor desta dissertação (2016).

Após a realização do estudo, foi identificado que dezesseis trabalhos investigados utilizam a tecnologia *Wi-Fi* como forma de comunicação, beneficiando-se da maior velocidade e alcance se comparada com as demais tecnologias (IEEE, 2016).

A segunda tecnologia de rede mais utilizada foi o infravermelho, com oito propostas, enquanto seis projetos utilizaram a tecnologia de *bluetooth*. As menos utilizadas foram o RFID e o ZigBee com apenas uma implementação. Assim como o caso dos sistemas operacionais, é necessário registrar que algumas das soluções pesquisadas podem funcionar em mais de uma rede (por exemplo, *Wi-Fi* e *bluetooth*).

3.6 CONSIDERAÇÕES FINAIS

Neste capítulo, foi apresentado um levantamento bibliográfico referente à utilização de controles remotos universais embarcados em *smartphones*, com o objetivo de

verificar os trabalhos sobre controle remotos universais para ambientes residenciais e identificar as tecnologias adotadas nos trabalhos correlatos desta dissertação.

Desta maneira, analisando os dados obtidos, pode-se concluir que a maioria dos trabalhos pesquisados utiliza o sistema operacional Android, por ser uma solução *open source* e amplamente adotada por diferentes fabricantes de *smartphones*. Esse fato, cria a possibilidade de aparelhos com diferentes preços utilizarem o mesmo sistema operacional, atendendo a públicos com diferentes faixas sociais.

Outra constatação é que a maioria das soluções não utilizam uma plataforma multicamadas. Isso é decorrente da facilidade em desenvolver uma solução com apenas duas camadas, sendo que o programador só irá se preocupar com a criação da camada de interação, sem a necessidade de desenvolvimento de um *middleware*.

Foi detectado que cerca de vinte trabalhos pesquisados identificam os dispositivos eletrônicos no ambiente de forma automática e suas interfaces podem ser *renderizadas* dinamicamente para permitir que as interações sejam adequadas com as funcionalidades de cada dispositivo.

Após a investigação bibliográfica, foi possível definir as principais características da plataforma Remote Home, levando em consideração os pontos positivos dos projetos analisados, sendo possível destacar os seguintes atributos:

- a) Será uma plataforma multicamada, viabilizando a extensibilidade do projeto.
- b) Irá possuir um *middleware*, possibilitando que outras modalidades de interação possam fazer parte da plataforma.
- c) Os dispositivos no cenário de interação serão detectados automaticamente.
- d) A interface do aplicativo será *renderizada* de acordo com as funcionalidades de cada dispositivo eletrônico, facilitando as interações.
- e) O cenário de interação será instrumentado com atuadores, possibilitando que dispositivos eletrônicos que não são projetados com controles remotos, possam usufruir da plataforma Remote Home.

- f) O projeto deverá contemplar: as diferentes tecnologias de rede sem fio, tais como, *Wi-Fi*, para conectar o controle remoto, o *middleware* e os atuadores; e a tecnologia de infravermelho, para integrar os atuadores e dispositivos eletrônicos.

No próximo capítulo será apresentada a solução proposta nesta dissertação, contemplando a especificação do modelo e a idealização de uma plataforma para a validação da abordagem proposta.

4 REMOTE HOME: CONTROLE UNIVERSAL PARA UM AMBIENTE RESIDENCIAL

Este capítulo aborda a estruturação da plataforma proposta. Inicialmente, serão apresentados os fatores que levaram a construção da plataforma em camadas, além disso, será apresentado o modelo em diagramas UML. Também é abordado o descritivo da plataforma, contemplando os componentes de hardware e software utilizados para o funcionamento de um controle remoto universal em um ambiente residencial.

4.1 INTRODUÇÃO

Conforme constatado no levantamento bibliográfico, o conceito de Automação Residencial com o uso dos controles remotos universais está se popularizando cada vez mais (RAJABZADEH *et al.*, 2010). Isso gera novas oportunidades para desenvolver soluções na área de Automação Residencial.

Desta maneira, após a investigação de diversos trabalhos científicos, foi desenvolvido o modelo de um controle remoto universal e validado com a construção de uma plataforma em três camadas, agregando as contribuições de trabalhos anteriores em uma única solução, que é caracterizada por ser uma plataforma dividida em camadas independentes. Deste modo, é possível controlar dispositivos eletrônicos que não foram projetados para serem comandados por controle remoto sem dependência de fabricante, marca ou modelo.

A plataforma Remote Home está dividida nas seguintes camadas:

- a) Camada de Interação – Representada por um aplicativo para dispositivos móveis que funciona como um controle remoto universal. Ele ajusta a sua interface de acordo com as características dos dispositivos eletrônicos existentes no ambiente, possibilitando a interação dos usuários.
- b) Camada de Gerência – Definida por um software correspondente a um *middleware*, responsável pela identificação e comunicação com os dispositivos eletrônicos disponíveis no cenário de interação.

- c) Camada de Atuadores – São representados por microprocessadores interligados aos dispositivos eletrônicos que são responsáveis por executar ações no ambiente. Por exemplo, ligar ou desligar a lâmpada, a televisão ou o ar-condicionado.

O restante deste capítulo está subdividido em duas partes principais: (i) a especificação do modelo como solução para a nova abordagem proposta; e (ii) a idealização de uma plataforma para comprovar a viabilidade do modelo e a eficiência da proposta.

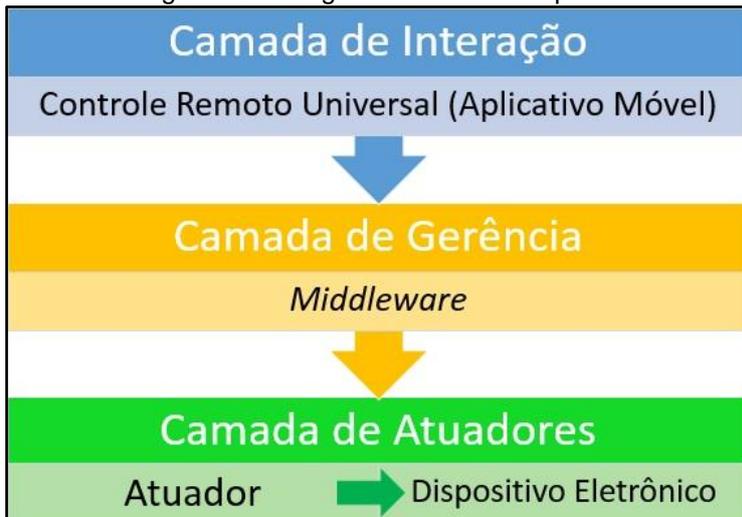
4.2 ESPECIFICAÇÃO DO MODELO

Levando em consideração os controles remotos universais disponíveis no mercado e a necessidade de evoluí-los, os requisitos funcionais do modelo proposto neste trabalho são:

- a) Permitir a interação do usuário com os dispositivos eletrônicos em um ambiente residencial de interação por intermédio de um aplicativo com as características de um controle remoto universal.
- b) Integrar o controle remoto universal, atuadores e dispositivos eletrônicos através da Camada de Gerência (*middleware*).
- c) Identificar e adicionar automaticamente novos atuadores na plataforma.
- d) Gerar automaticamente e em tempo real a interface gráfica do controle remoto universal de acordo com os dispositivos eletrônicos existentes no cenário de interação.

O modelo proposto é dividido em três camadas (Figura 46) para facilitar o desenvolvimento, a separação de interesses e o isolamento de problemas inerentes aos cenários de interação. A divisão dos componentes em camadas possibilita que a tecnologia empregada possa ser testada com diferentes recursos de hardware e software.

Figura 46 - Imagem do Modelo Proposto

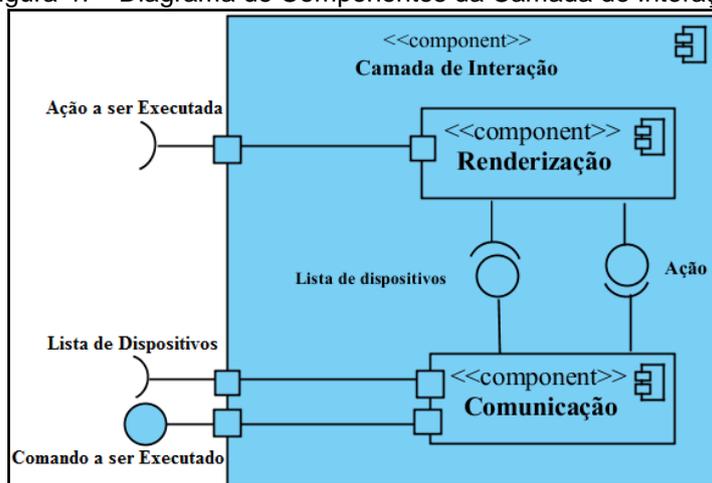


Fonte: Autoria própria do autor desta dissertação (2016).

4.2.1 Camada de Interação

A Camada de Interação é responsável em disponibilizar um aplicativo que proverá a interface para manipular os dispositivos eletrônicos existentes no cenário. Essa camada, está dividida em dois componentes: (i) *Renderização* da interface e (ii) *Comunicação*, como pode ser observado na Figura 47.

Figura 47 - Diagrama de Componentes da Camada de Interação



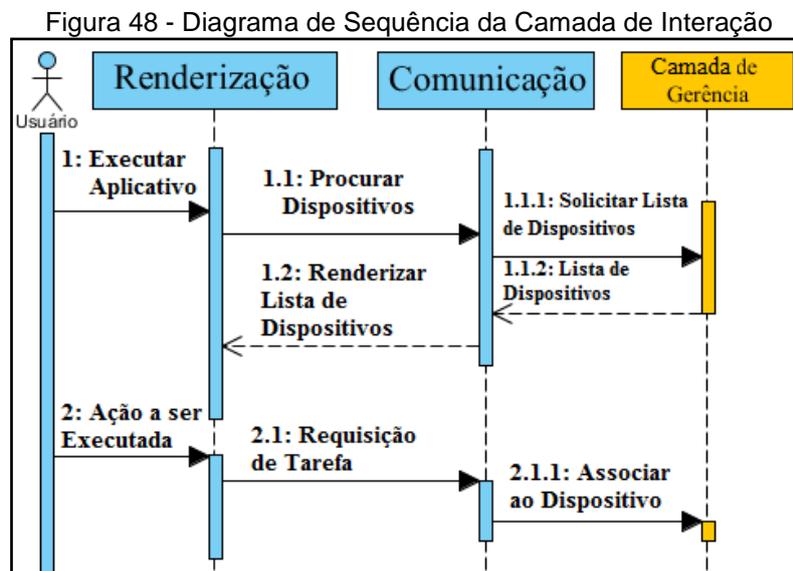
Fonte: Autoria própria do autor desta dissertação (2016).

O Componente de *Renderização* permite a construção automática de telas em função das características dos dispositivos eletrônicos. O princípio do seu funcionamento é

armazenar um conjunto de telas prontas e disponibilizar para cada dispositivo a que melhor se adapte as suas funcionalidades.

O Componente de Comunicação é responsável por solicitar, para a Camada de Gerência, a lista de dispositivos eletrônicos com suas características e encaminhar para o Componente de *Renderização*. Além disso, o Componente de Comunicação envia as requisições de tarefas solicitadas pelos usuários à Camada de Gerência para que os dispositivos eletrônicos possam ser acionados via controle remoto.

O diagrama de seqüência da Figura 48 apresenta o fluxo de funcionamento da Camada de Interação. Quando o usuário executa o aplicativo, será solicitada à Camada de Gerência as informações referentes a cada dispositivo eletrônico para que a lista possa ser disponibilizada para o aplicativo. Após essa fase de inicialização, o aplicativo estará apto a disponibilizar tarefas relacionadas a cada dispositivo eletrônico no cenário de interação.



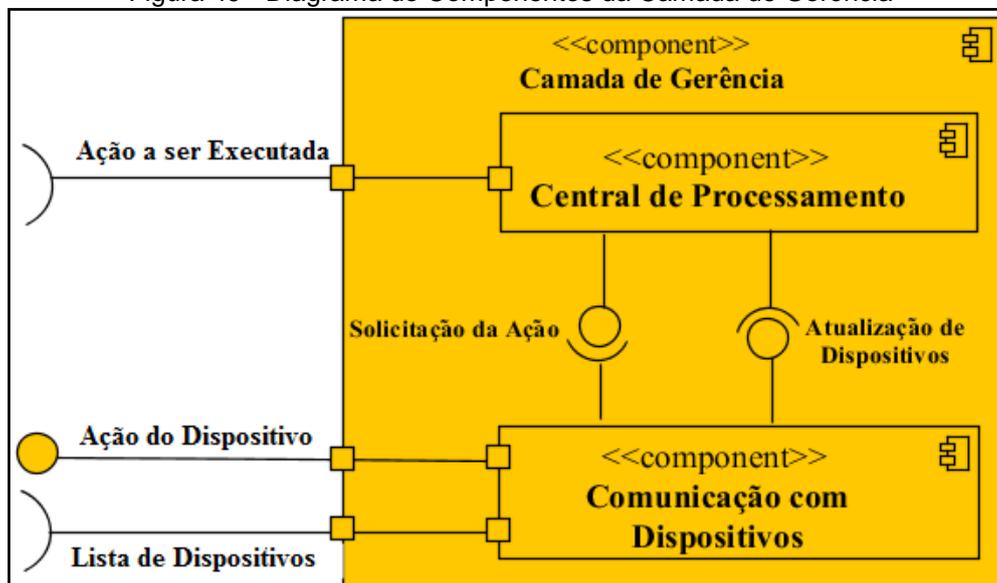
Fonte: Autoria própria do autor desta dissertação (2016).

4.2.2 Camada de Gerência

A Camada de Gerência é responsável pela detecção dos dispositivos eletrônicos no cenário de interação, por realizar a comunicação entre as camadas e por processar

as tarefas requeridas pelos usuários. Essa camada atua como um *middleware* e está dividida em dois componentes conforme apresentado na Figura 49.

Figura 49 - Diagrama de Componentes da Camada de Gerência



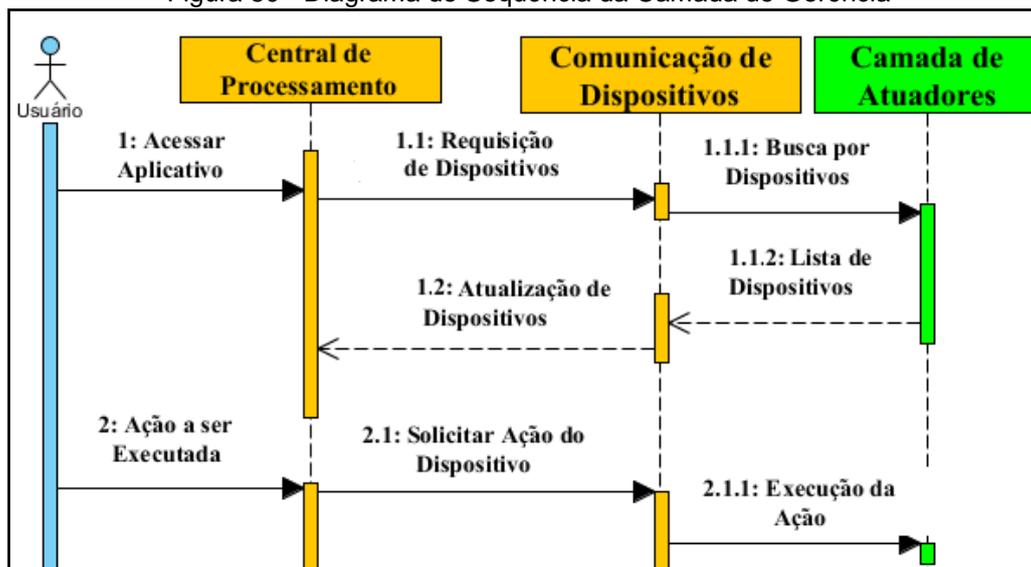
Fonte: Autoria própria do autor desta dissertação (2016).

A Central de Processamento é responsável pelo recebimento das requisições da Camada de Interação. Basicamente, esse componente identifica qual dispositivo está sendo acionado e qual função deve ser executada. Após a identificação da tarefa, a requisição é encaminhada para o componente de Comunicação com Dispositivos que transmite a tarefa requerida pelo usuário ao dispositivo eletrônico responsável em executá-la.

Outra função do componente de Comunicação com Dispositivos é identificar os dispositivos eletrônicos presentes no cenário de interação, que, necessariamente, devem estar vinculados a um atuador (microcontrolador). Basicamente, o componente localiza, na rede local, os atuadores, recebe suas informações (ID do Dispositivo Eletrônico e Funcionalidades) e envia uma lista contendo estes dados para a Camada de Interação.

Na Figura 50, é apresentado o diagrama de sequência dos componentes pertencentes a Camada de Gerência.

Figura 50 - Diagrama de Sequência da Camada de Gerência

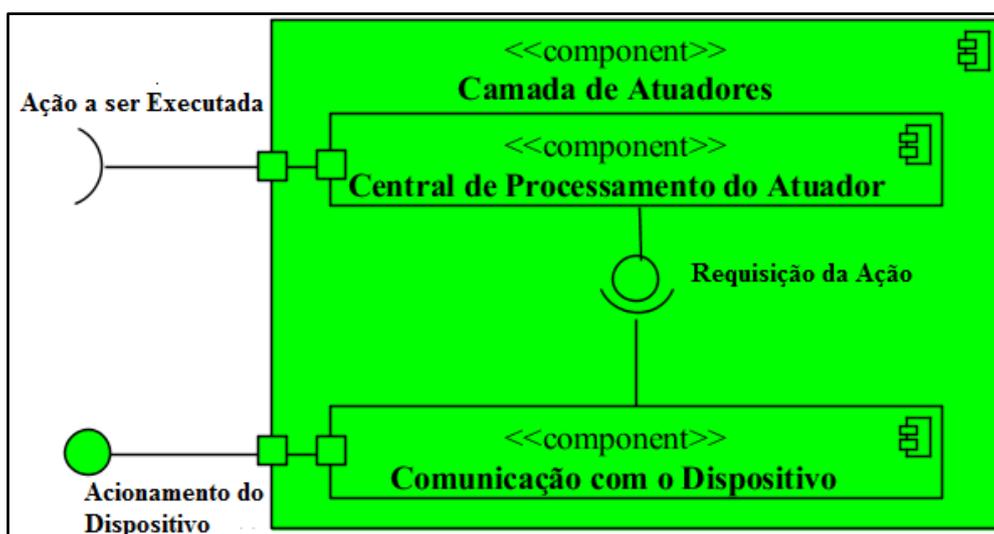


Fonte: Autoria própria do autor desta dissertação (2016).

4.2.3 Camada de Atuadores

A Camada de Atuadores associa os microcontroladores aos dispositivos eletrônicos no ambiente residencial. Isso possibilita que os comandos enviados pelos usuários sejam transformados em ações nos dispositivos eletrônicos. Essa camada está dividida em dois componentes: (i) Central de Processamento do Atuator e (ii) Comunicação com o Dispositivo Eletrônico (Figura 51).

Figura 51 - Diagrama de Componentes da Camada de Atuadores



Fonte: Autoria própria do autor desta dissertação (2016).

O componente Central de Processamento do Atuador contém a lógica de acionamento do dispositivo eletrônico, enquanto o Componente de Comunicação com o Dispositivo Eletrônico é responsável pela comunicação do microcontrolador com esse dispositivo.

Na próxima seção, será apresentada a materialização do modelo em uma plataforma, possibilitando validar a proposta e viabilizar um controle remoto universal, que permite a comunicação com diferentes eletrodomésticos e sistemas eletrônicos, independente da marca, tipo ou modelo.

4.3 A PLATAFORMA REMOTE HOME

Uma possibilidade para os dispositivos domésticos se integrarem uns aos outros e ficarem visíveis para uma comunicação remota, é conectá-los a uma rede de dados que os identifique por meio de endereços IP. Desta forma, o usuário poderá controlar todos eles através de diferentes modalidades, uma delas é o uso de um aplicativo instalado em dispositivo móvel via toques (SATRIA *et al.*, 2015).

Tomando como base a ideia proposta por Rajabzadeh *et al.* (2010) e seguindo as diretrizes definidas no modelo apresentado na seção anterior, neste projeto, foi desenvolvida a plataforma denominada de Remote Home como prova de conceito.

Conforme a proposta do modelo, essa plataforma foi dividida em três camadas (Figura 52): (i) a Camada de Interação, que é representada por um controle remoto universal através de um aplicativo móvel; (ii) a Camada de Gerência, concebida em um software (*middleware*) embarcado em um dispositivo computacional de tamanho reduzido; e (iii) a Camada de Atuadores, que foi idealizada com softwares embarcados em atuadores com a função de acionar os dispositivos eletrônicos.

Figura 52 - Estrutura da plataforma dividida em camadas



Fonte: Autoria própria do autor desta dissertação (2016).

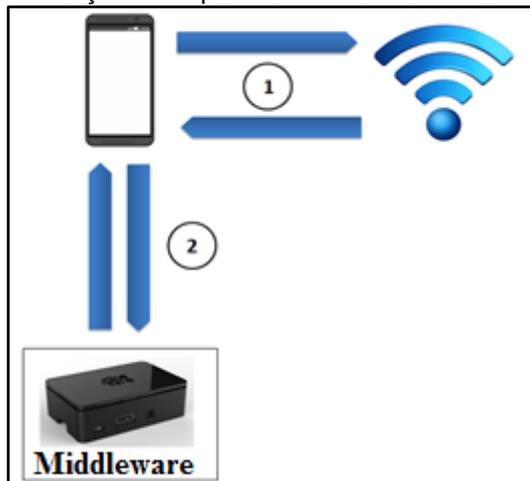
4.3.1 O Controle Remoto Universal

O aplicativo que desempenha as funcionalidades de um controle remoto universal foi desenvolvido com o uso da ferramenta Eclipse Luna⁵ que permite a criação de programas baseados na linguagem de programação Java. Além disso, foi adicionado o *kit* de desenvolvimento Android SDK, possibilitando o desenvolvimento do controle universal para dispositivos móveis, com o sistema operacional Android.

O controle remoto universal, na plataforma Remote Home, funciona da seguinte forma: quando o usuário inicializa o aplicativo, é pesquisado na rede local qual é o endereço IP do *middleware* (Passo 1 – Figura 53). Após encontrá-lo, o aplicativo faz uma requisição via HTTP para obter a lista de dispositivos eletrônicos e suas características (Passo 2 – Figura 53).

⁵Disponível em: <https://eclipse.org/luna/>

Figura 53 - Ações do aplicativo no módulo de inicialização



Fonte: Autoria própria do autor desta dissertação (2016).

Após receber a lista encapsulada em um formato JavaScript Object Notation (JSON), que coleta as informações vitais sobre os dispositivos eletrônicos, são elas: o identificador, IP do atuador e nome do dispositivo eletrônico que está acoplado a ele. Desse modo, é possível construir a interface do controle remoto universal, apresentando ao usuário os dispositivos existentes para interação conforme exibido na Figura 54.

Figura 54 - Lista de dispositivos para interação



Fonte: Autoria própria do autor desta dissertação (2016).

Quando o usuário escolher um dispositivo, o aplicativo constrói uma tela de acordo com as funcionalidades disponibilizadas pelo mesmo. Dessa maneira, o usuário pode escolher a ação que deseja executar. Para exemplificar, pode-se ver, à esquerda da Figura 55, a interface de uma televisão. A partir dela, o usuário pode executar algumas

ações, tais como: mudar de canal, aumentar ou diminuir o volume, além de desligar o dispositivo. Já no lado direito da Figura 55, é apresentada a interface de um ar-condicionado, onde podem ser contempladas ações como: ligar, desligar e aumentar ou diminuir a temperatura.

Figura 55 – Interfaces dinâmicas do controle remoto universal



Fonte: A autoria própria do autor desta dissertação (2016).

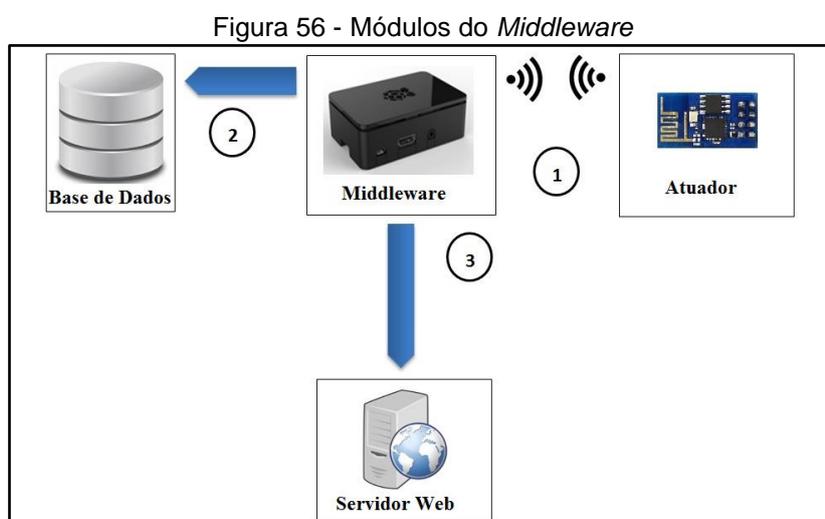
Após a execução de um comando por parte do usuário, o aplicativo inicia uma comunicação através do protocolo HTTP com o *middleware* por meio de seu endereço IP. Então o *middleware* identificará qual foi o comando enviado e acionará o atuador correspondente. Na próxima subseção, serão abordados em detalhes os processos executados pelo *middleware*.

4.3.2 O *Middleware*

O principal objetivo do *middleware* é realizar a comunicação e disponibilizar serviços entre o Remote Home e os dispositivos eletrônicos. Para o funcionamento do *middleware* na plataforma Remote Home, foi necessário encontrar um dispositivo central que pudesse ser pequeno o bastante para ser embarcado no ambiente, de forma que fosse completamente ubíquo para o usuário. Nessa prova de conceito, foi utilizado o Raspberry Pi⁶, um computador de tamanho extremamente reduzido, com alta capacidade de processamento, que utiliza o sistema operacional Linux Raspbian.

⁶Disponível em: <https://www.raspberrypi.org>

O *middleware* que funciona no Raspberry Pi foi desenvolvido em Java, ferramenta que oferece grande compatibilidade com diferentes plataformas e sistemas operacionais. Ele executa três ações que podem ser vistas através da Figura 56, são elas: (i) uma varredura na rede local para identificar os dispositivos eletrônicos presentes no ambiente; (ii) armazenamento das informações referentes aos dispositivos; e (iii) a inicialização um servidor *Web*, disponibilizando a lista com as informações no formato JSON.



Fonte: Autoria própria do autor desta dissertação (2016).

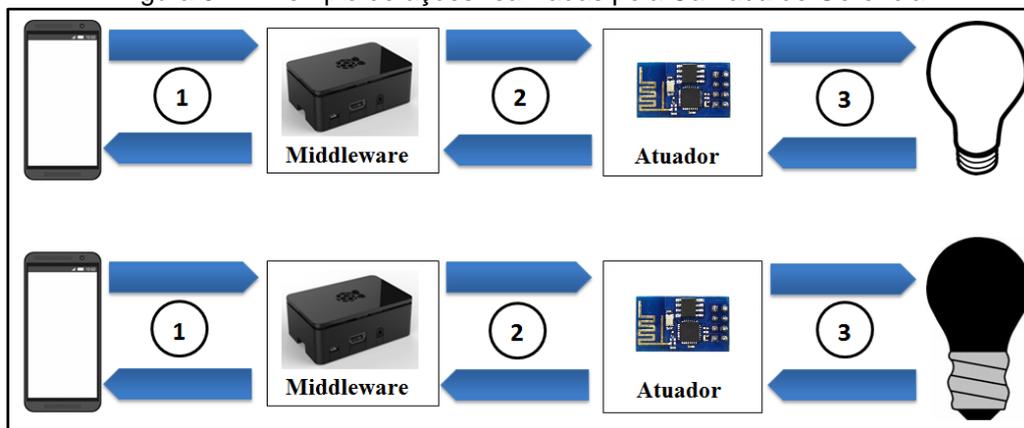
Quando o *middleware* é ligado, ele cria uma conexão com a rede *Wi-Fi* e funciona como um Ponto de Acesso para os atuadores se conectarem. Após essa inicialização, ele realiza uma varredura na rede para encontrar os atuadores e, conseqüentemente, os dispositivos eletrônicos presentes no ambiente (Passo 1 – Figura 56).

Assim que os atuadores forem encontrados, suas informações serão adicionadas na Base de Dados do *middleware* (Passo 2 – Figura 56). Desta forma, é criada a lista de atuadores no formato JSON, que será disponibilizada por meio de um Servidor *Web* (Passo 3 – Figura 56).

Quando o usuário solicitar uma ação no ambiente por meio do aplicativo Remote Home, esse enviará uma requisição ao *middleware* (Passo 1 – Figura 57), que encaminhará a solicitação para o Atuador correto (Passo 2 – Figura 57), que é um

microcontrolador integrado a um dispositivo eletrônico. Tal Atuador realizará a ação e atualizará o status do dispositivo eletrônico (Passo 3 – Figura 57).

Figura 57 - Exemplo de ações realizadas pela Camada de Gerência



Fonte: Autoria própria do autor desta dissertação (2016).

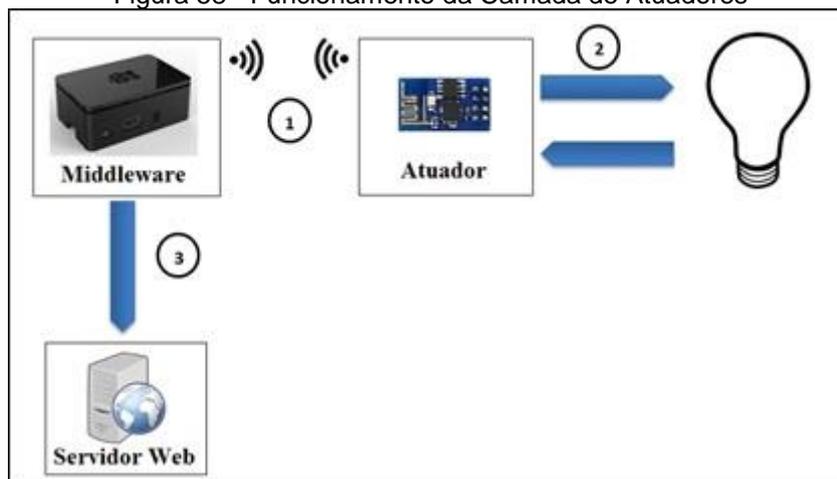
4.3.3 Atuadores acoplados aos Dispositivos Eletrônicos

Os atuadores possuem a finalidade de controlar os dispositivos eletrônicos, sendo estruturados em componentes de hardware (microcontroladores) e *firmware* embarcados, que são softwares responsáveis por gerenciar o hardware da plataforma.

As ações desempenhadas pelos Atuadores podem variar entre: (i) funções simples, como ligar/desligar ou abrir/fechar, como no caso de lâmpadas, portas e janelas; e (ii) desempenhar múltiplas funções, como controlar os canais e volume de uma televisão.

Quando um Atuador é ligado pela primeira vez, ele se conectará, via *Wi-Fi*, com o *middleware*, obtendo um endereço IP por DHCP (Passo 1 – Figura 58). Logo após, ele enviará para o *middleware* as informações referentes ao tipo de dispositivo que irá controlar (Passo 2 – Figura 58). Por fim, o *middleware* atualizará as informações referentes ao dispositivo eletrônico no Servidor *Web* (Passo 3 – Figura 58).

Figura 58 - Funcionamento da Camada de Atuadores



Fonte: Autoria própria do autor desta dissertação (2016).

Cada atuador terá o seu respectivo endereço IP, ID, *status* e nome. Dessa forma, se ele gerenciar uma televisão, por exemplo, irá se identificar para o *middleware* como tal, utilizando essas propriedades. Ao receber uma solicitação do *middleware*, o atuador identificará qual deverá ser a ação e irá executá-la, retornando uma confirmação. O *middleware*, por sua vez, irá repassar a informação para o controle remoto universal que identificará se a operação foi realizada com sucesso.

Como um dos objetivos da plataforma Remote Home é ser de baixo custo, foi necessário encontrar um atuador extremamente barato e eficiente para se comunicar com o *middleware* e os dispositivos eletrônicos no ambiente. Neste caso, foi utilizado o atuador ESP8266⁷, que é extremamente barato, possui um módulo *Wi-Fi* embutido e um microcontrolador ARM acoplado.

O ESP8266 possui portas digitais que funcionam para a realização das ações do atuador. Através da utilização dessas portas com a combinação binária é possível realizar uma série de ações nos dispositivos eletrônicos, tais como: ligar lâmpada, mudar o canal da televisão, dentre outras. A velocidade do processamento das combinações binárias permite o mapeamento das ações programadas no controle infravermelho. Dessa forma, torna-se possível configurá-lo para se comunicar com dispositivo eletrônico.

⁷Disponível em: <https://espressif.com/products/hardware/esp8266ex/overview/>

4.4 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentados o modelo, funcionamento da plataforma Remote Home e os aspectos mais importantes da solução proposta. Verificou-se que a plataforma é dividida em três camadas independentes entre si, de forma que cada uma delas possa ser representada por um módulo que contém um conjunto de componentes.

Essas camadas possuem funcionalidades específicas, que realizam toda a interação entre o usuário e o dispositivo eletrônico selecionado. Sendo que a camada de interação solicita informações dos dispositivos disponíveis no ambiente, construindo a sua interface para a solicitação do usuário e, ao mesmo tempo, a camada de gerência faz a intermediação entre as outras duas camadas, buscando os atuadores no ambiente e coletando informações dos dispositivos para enviar para a camada de interação. Finalmente, os atuadores irão receber as ações e acionar o dispositivo eletrônico desejado pelo usuário.

Para esta pesquisa, foram escolhidas as tecnologias Eclipse, Android, Java, Raspberry Pi e ESP8266 por serem open source e estarem a disposição no Laboratório do Pavilhão de Aulas 6 da universidade. Apesar disso, outras ferramentas ou dispositivos computacionais poderiam ser utilizados nesta plataforma, visto que ela foi projetada para funcionar com tipos de atuadores, dispositivos centrais ou dispositivos móveis diferentes.

O desenvolvimento do controle remoto universal, bem como, o *middleware* e as configurações dos atuadores podem ser visualizadas no Apêndice D desta dissertação. O próximo capítulo irá apresentar o experimento realizado para validar a plataforma construída e a análise dos resultados alcançados, com o objetivo de avaliar as potencialidades da plataforma Remote Home.

5 REALIZAÇÃO DO EXPERIMENTO, RESULTADOS E ANÁLISE

Neste capítulo, serão apresentados os detalhes da condução do experimento e a análise dos resultados obtidos, possibilitando que possam ser avaliados os pontos fortes e fracos da plataforma Remote Home. O experimento é a oportunidade para a plataforma ser testada em um ambiente real. Toda a sua execução foi realizada tomando como base o *framework* DECIDE (SHARP *et al.*, 2011), que viabiliza uma sequência lógica de passos a serem seguidos pelos avaliadores.

5.1 INTRODUÇÃO

O experimento é um importante elemento para validar a ideia ou o produto criado pelo pesquisador. Desta maneira, para a validação da plataforma Remote Home, foi realizado um experimento com trinta pessoas (estudantes e professores) da Universidade Salvador, que responderam a um questionário avaliando o controle remoto universal e a plataforma Remote Home. Com estes dados, foi possível reunir os indicadores necessários para apresentar os resultados finais desta dissertação.

5.2 PLANEJAMENTO DO EXPERIMENTO

O experimento relatado neste trabalho foi dividido em seis fases distintas, tomando como base as diretrizes propostas no *framework* DECIDE (SHARP *et al.*, 2011) que norteou os passos realizados durante todas as fases do experimento:

- **Determinar o objetivo da análise** – O foco do experimento foi obter informações referentes à usabilidade da plataforma proposta. As métricas utilizadas para realizar as avaliações foram propostas por Kronbauer e Santos (2013) e abrange a análise das seguintes métricas: eficiência, eficácia, satisfação, aprendizagem, operabilidade, acessibilidade, utilidade, flexibilidade e facilidade de uso dos usuários, que estão descritas no Quadro 8.

Quadro 8 - Descrição das Métricas

MÉTRICA	DESCRIÇÃO
Eficiência	Analisa a velocidade com que os usuários conseguem executar uma determinada função.
Eficácia	Mede a capacidade do usuário utilizar o conjunto de funções de forma correta.
Utilidade	Capacidade do usuário atingir certos objetivos propostos no experimento.
Satisfação	Define uma nota atribuída ao usuário pelo sentimento que o mesmo teve em relação ao projeto.
Aprendizagem	Facilidade no entendimento do uso do sistema por parte do usuário.
Operabilidade	Capacidade de utilização do sistema.
Acessibilidade	Diferentes formas de acesso à plataforma.
Flexibilidade	Mede o comportamento do sistema
Facilidade de Uso	Facilidade do usuário em relação ao uso do aplicativo, bem como uma menor quantidade de funções, porém mais objetivas.

Fonte: Kronbauer e Santos (2013).

- Explorar perguntas a serem respondidas – Tomando como base o objetivo a ser alcançado, foi elaborado um conjunto de perguntas (Quadro 9 e Quadro 10) que direcionam as análises dos dados, além de comprovar as potencialidades da abordagem proposta neste estudo.

Quadro 9 - Questões Utilizadas para o Controle Remoto Universal

PERGUNTAS		ATRIBUTO
1	Qual o seu nível de satisfação com a rapidez (eficiência) com que consegue realizar as tarefas através do controle remoto universal?	Eficiência
2	Qual o seu nível de satisfação com a precisão (eficácia) com que consegue executar as tarefas desejadas no controle remoto universal?	Eficácia
3	Qual é seu nível de satisfação com a utilização do controle remoto universal?	Satisfação
4	Qual o seu nível de satisfação com a aprendizagem das funcionalidades do controle remoto universal?	Aprendizagem
5	Quando houve uma interação indesejada, você conseguiu facilmente retornar para um estado anterior?	Operabilidade
6	Qual o seu nível de satisfação com a acessibilidade disponibilizada pelo controle móvel universal?	Acessibilidade
7	Qual o seu nível de satisfação com relação à flexibilidade (caminhos alternativos para executar uma tarefa) disponíveis no controle remoto universal?	Flexibilidade
8	Qual o seu nível de satisfação com a utilidade do controle móvel universal?	Utilidade
9	Qual é sua percepção com relação a facilidade de uso do controle móvel universal?	Facilidade de uso

Fonte: Autoria própria do autor desta dissertação (2016).

Quadro 10 - Questões Utilizadas para a Plataforma proposta

PERGUNTAS	
1	Você prefere utilizar a Plataforma Remote Home em vez de um controle remoto tradicional?
2	Você voltaria a utilizar a Plataforma Remote Home em outra ocasião?
3	Qual o seu nível de satisfação com a interação da Plataforma Remote Home?
4	Qual a sua satisfação em relação a velocidade de execução da Plataforma Remote Home?
5	Você tem alguma consideração que gostaria de relatar para aprimorar o desenvolvimento da plataforma proposta?

Fonte: Autoria própria do autor desta dissertação (2016).

- Escolher o método de avaliação – A abordagem escolhida para a obtenção dos dados foi a aplicação de um questionário com a escala de Likert (1932), na qual os valores variam de 1, para muito insatisfeito, até 5, para muito satisfeito.
- Identificar e Administrar as questões práticas – Nessa fase, durante a realização do experimento, foram especificados dois documentos utilizados: (i) um texto explicativo referente à proposta do trabalho e dicas simples para a utilização da plataforma Remote Home; e (ii) um roteiro a ser seguido pelo usuário, contemplando as tarefas que devem ser executadas, conforme apresentado no Quadro 11.

Quadro 11 - Roteiro de Ações

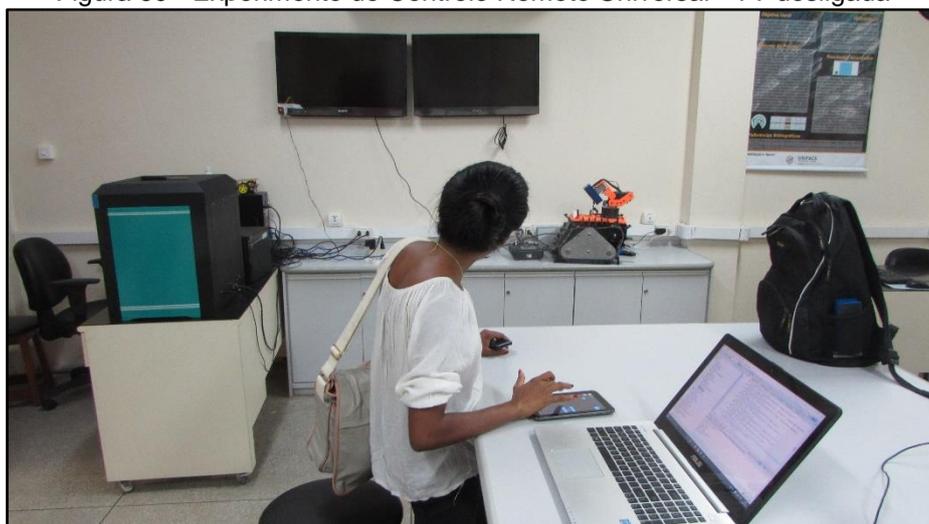
ETAPAS	AÇÕES
1	Ligar o ar-condicionado, ajustar a temperatura para um valor confortável e desligá-lo.
2	Ligar a televisão, sintonizar um canal de sua preferência, ajustar o volume e desligar o aparelho.

Fonte: Autoria própria do autor desta dissertação (2016).

O experimento foi realizado no Laboratório de Dispositivos Móveis e Sistemas Embarcados do Pavilhão de Aulas 6 (PA6) da Universidade Salvador (Figura 59 e Figura 60). Para construir o cenário de interação, foi necessário utilizar os seguintes recursos: dois dispositivos eletrônicos (televisão e ar-condicionado); dois atuadores ESP8266; um microprocessador Raspberry Pi e um dispositivo móvel com o sistema operacional Android.

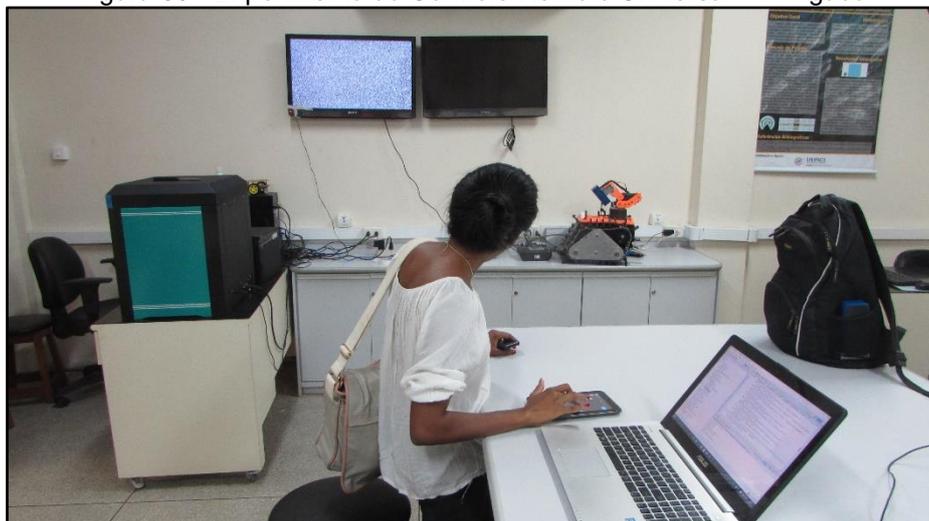
Os softwares desenvolvidos para a preparação da plataforma foram embarcados nos seus respectivos dispositivos computacionais e a comunicação entre eles foi disponibilizada via uma rede *Wi-Fi*. Entre os atuadores e os dispositivos eletrônicos, a rede sem fio utilizada foi o infravermelho.

Figura 59 - Experimento do Controle Remoto Universal - TV desligada



Fonte: Autoria própria do autor desta dissertação (2016).

Figura 60 - Experimento do Controle Remoto Universal - TV ligada



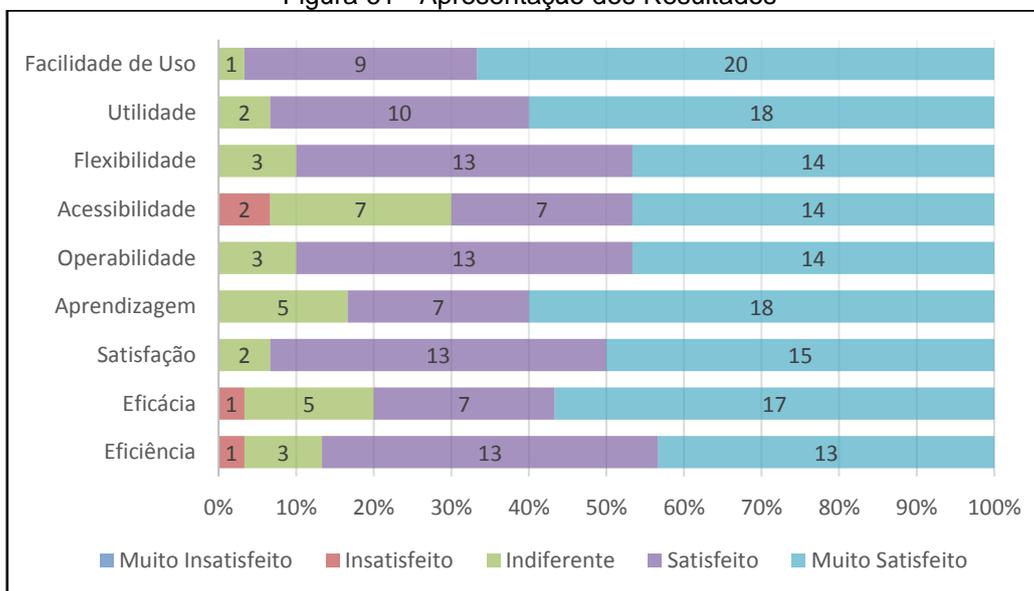
Fonte: Autoria própria do autor desta dissertação (2016).

Para a realização do experimento, foi criado um documento para a leitura do usuário, que é composto de um texto explicando toda a funcionalidade do trabalho e a proposta (Apêndice B). Após esse entendimento, foi entregue um

roteiro com as tarefas a serem realizadas no cenário de interação (Apêndice C).

- **Decidir como lidar com as questões éticas** – O projeto desta pesquisa foi submetido ao Comitê de Ética, identificado na Plataforma Brasil por meio do identificador CAAE: 51137215.4.0000.5033, sendo devidamente aprovado. O experimento foi conduzido preservando o anonimato dos participantes, pois, no Termo de Consentimento Livre e Esclarecido, está especificado que as informações pessoais dos participantes não seriam divulgadas. Além disso, todos os voluntários possuíam mais de dezoito anos de idade e gozavam de plena capacidade física e mental.
- **Estabelecer forma de avaliar, interpretar e apresentar os resultados** – A coleta de dados ocorreu no período de 16 a 18 de agosto de 2016, com a participação de trinta usuários. O gráfico apresentado na Figura 61 exibe os resultados obtidos após a realização do experimento.

Figura 61 - Apresentação dos Resultados



Fonte: Autoria própria do autor desta dissertação (2016).

O *framework* DECIDE mostrou-se eficiente para a condução do experimento que avaliou a utilização da plataforma Remote Home. Com a finalidade de analisar as métricas utilizadas durante a experiência, foram gerados diversos gráficos, objetivando facilitar a análise da solução.

5.3 ANÁLISE QUALITATIVA DO CONTROLE REMOTO UNIVERSAL

A análise qualitativa tem como principal meta identificar e medir o nível de concordância dos usuários em relação à usabilidade do controle remoto universal. A análise foi aplicada através de um questionário contendo nove questões formatadas em função da escala de Likert (1932) e uma questão discursiva, na qual os usuários opinaram, com suas próprias palavras, sobre a plataforma proposta.

As respostas apresentadas pelos usuários geraram um gráfico (Figura 61), onde estão distribuídas as respostas dos usuários através de números inteiros inseridos nas barras. Cada cor utilizada no gráfico representa uma das possíveis respostas referente à escala de Likert (1932) com cinco pontos, representando respostas que variam de muito insatisfeito a muito satisfeito. Estes dados demonstram que o controle remoto universal da plataforma Remote Home agradou a maioria dos usuários que participaram do experimento. Isso comprova que a ferramenta é eficiente e atende as solicitações dos usuários sem apresentar nenhum tipo de falha.

5.3.1 Avaliação da Eficiência

A Eficiência objetiva avaliar a rapidez com que os usuários conseguem executar uma tarefa desejada. Nesse quesito, podem ser destacados dois fatores importantes para o sucesso da plataforma: (i) a rapidez de interação do usuário com o aplicativo, o que demonstra a coerência do *layout* para a execução das tarefas; e (ii) a velocidade de processamento do *middleware* e dos atuadores, possibilitando que as ações ocorram em tempo real para contemplar as expectativas dos usuários.

Desta maneira, analisando a Figura 61, para responder a seguinte pergunta:

- Qual o seu nível de satisfação com a rapidez (eficiência) com que consegue realizar as tarefas através do controle remoto universal?

Assim, conclui-se que 87% dos usuários acharam o controle remoto universal eficiente, 10% foram indiferentes e 3% ficaram insatisfeitos.

5.3.2 Avaliação da Eficácia

A Eficácia mensura a assertividade com que um usuário consegue interagir com um sistema computacional. Segundo Coursaris e Kim (2011), esse atributo é um dos mais importantes para medir a usabilidade de um sistema, já que abrange a verificação da taxa de erros cometida pelos usuários.

Para avaliar este quesito foi proposta a seguinte pergunta:

- Qual o seu nível de satisfação com a precisão (eficácia) com que consegue executar as tarefas desejadas no controle remoto universal?

Os dados obtidos no experimento informam que 80% dos participantes ficaram satisfeitos ou muito satisfeitos com a sua eficácia ao interagir com o controle remoto universal. Deste modo, os resultados comprovam que a comunicação existente entre o controle remoto universal e o usuário é precisa, visto que ele consegue utilizar todas as funcionalidades do aplicativo sem nenhuma dificuldade e de forma rápida.

5.3.3 Avaliação da Satisfação

A satisfação identifica, de modo geral, o sentimento que o usuário tem a respeito de um produto (KRONBAUER ; SANTOS, 2013). Assim sendo, a seguinte pergunta foi proposta aos participantes do experimento para avaliar essa métrica:

- Qual é seu nível de satisfação com a utilização do controle remoto universal?

Pode-se concluir que o controle remoto universal possui potencialidades que agrada os participantes do experimento, já que 93% dos participantes ficaram satisfeitos ou muito satisfeitos com o aplicativo móvel.

5.3.4 Avaliação da Aprendizagem

Segundo a ISO/IES 9126-1 (2001), a métrica de Aprendizagem identifica a facilidade de aprendizado do sistema pelos seus potenciais usuários. A pergunta proposta aos usuários para avaliar a aprendizagem foi:

- Qual o seu nível de satisfação com a aprendizagem das funcionalidades do controle remoto universal?

Os resultados demonstraram que 80% dos participantes consideraram fácil ou muito fácil aprender as funcionalidades do controle remoto universal. Entretanto, 17% dos participantes ficaram indiferentes com relação ao aprendizado do aplicativo e 3% ficaram muito insatisfeitos, sendo possível supor que essa quantidade de participantes teve algum tipo de dificuldade para repetir alguma tarefa.

5.3.5 Avaliação da Operabilidade

Um dos principais desafios em projetos na área de Automação Residencial é conseguir vencer a baixa latência da rede de dados e obter o máximo de velocidade de processamento dos componentes embarcados para que a execução seja realizada em tempo real (SHARP *et al.*, 2011). Desta forma, a Operabilidade é outro fator relevante para o êxito de um projeto de interação.

No trabalho de Betiol e Cybis (2005), foi observado que, a princípio, os participantes dos testes de usabilidade acreditavam que uma conexão instável e lenta é mais prejudicial para a interatividade com um aplicativo móvel do que o tamanho reduzido das interfaces.

Nesse sentido, foi proposta a seguinte pergunta para os participantes do experimento:

- Quando houve uma interação indesejada, você conseguiu facilmente retornar para um estado anterior?

Os resultados obtidos mostram que o controle remoto universal contempla as expectativas da maioria dos usuários, com exceção de 10% deles, que ficaram indiferentes.

5.3.6 Avaliação da Acessibilidade

A Acessibilidade apresenta-se como a métrica que torna possível um acesso facilitado ao controle remoto universal. No experimento proposto, os usuários responderam a seguinte pergunta:

- Qual o seu nível de satisfação com a acessibilidade disponibilizada pelo controle remoto universal?

Os resultados obtidos informam que 70% dos participantes do experimento fizeram uma boa avaliação do sistema em relação à métrica de acessibilidade.

5.3.7 Avaliação da Flexibilidade

A Flexibilidade objetiva mede o comportamento do controle remoto universal para a execução de uma determinada tarefa no programa. Assim, a seguinte pergunta foi proposta para avaliar esse quesito:

- Qual o seu nível de satisfação com relação à flexibilidade (caminhos alternativos para executar uma tarefa) disponíveis no controle remoto universal?

Os resultados indicam que 90% dos participantes se dizem satisfeitos ou muito satisfeitos com a flexibilidade do sistema. Neste sentido, pode-se verificar que o controle remoto universal se comporta de forma adequada, já que o usuário consegue utilizar caminhos alternativos no sistema para realizar uma determinada ação sem apresentar falhas.

5.3.8 Avaliação da Utilidade

A Utilidade tem como principal meta medir a capacidade do usuário de atingir certos objetivos propostos no experimento, além de mensurar os benefícios trazidos pelo sistema. Para avaliar esta métrica, foi proposta a seguinte pergunta:

- Qual o seu nível de satisfação com a utilidade do controle remoto universal?

De acordo com os dados apresentados no gráfico da Figura 61, verificou-se que 93% dos participantes do experimento acreditam na utilidade do controle remoto universal para a sua utilização nas residências, visto que eles consideram o sistema útil para o propósito para o qual foi desenvolvido. Apenas 7% dos usuários responderam que eram indiferentes a proposta do projeto.

5.3.9 Avaliação da Facilidade de Uso

De acordo com os estudos de Coursaris e Kim (2011), a Facilidade de Uso é o atributo mais investigado nas avaliações de usabilidade. Segundo Chen *et al.* (2014), os aplicativos mais simples e fáceis de serem utilizados são aqueles que conseguem conquistar mais usuários e aumentam o poder de vendas no mercado.

Nesse sentido, foi proposta a seguinte pergunta aos participantes do experimento:

- Qual é sua percepção com relação à facilidade de uso do controle remoto universal?

Percebe-se que o controle remoto universal apresenta boa usabilidade, já que 97% dos participantes ficaram satisfeitos ou muito satisfeitos com o controle remoto universal.

5.4 ANÁLISE QUALITATIVA DO USO DA PLATAFORMA REMOTE HOME

Após a análise do controle remoto universal, foi necessário realizar a avaliação do uso da plataforma Remote Home. Neste caso, foram propostas cinco perguntas aos usuários, conforme foi apresentado no Fonte: Autoria própria do autor desta dissertação (2016).

Quadro 10 desta dissertação. Os resultados da avaliação podem ser observados nas próximas subseções.

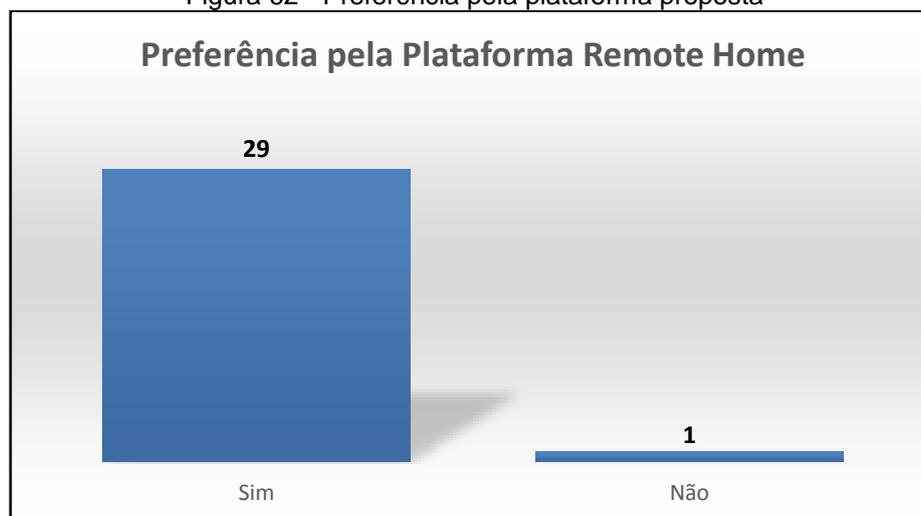
5.4.1 Avaliação da Preferência

A preferência define se o usuário substituiria seus controles remotos utilizados na residência pela solução proposta. A pergunta apresentada aos participantes foi:

- Você prefere utilizar a Plataforma Remote Home em vez de um controle remoto tradicional?

Conforme os resultados apresentados na Figura 62, é possível verificar que 97% dos usuários têm preferência por usar a plataforma Remote Home. Apenas 3% informaram que ainda preferem os controles remotos tradicionais.

Figura 62 - Preferência pela plataforma proposta



Fonte: Autoria própria do autor desta dissertação (2016).

5.4.2 Avaliação da Utilização

A utilização da plataforma Remote Home foi definida como a vontade do usuário de voltar a usar a ferramenta proposta em algum momento. Desta forma, a seguinte pergunta foi proposta aos participantes do experimento:

- Você voltaria a utilizar a plataforma Remote Home em outra ocasião?

Com a resposta dos participantes, foi possível identificar que todos os usuários gostariam de utilizar a plataforma novamente, comprovando as potencialidades e o sucesso do projeto.

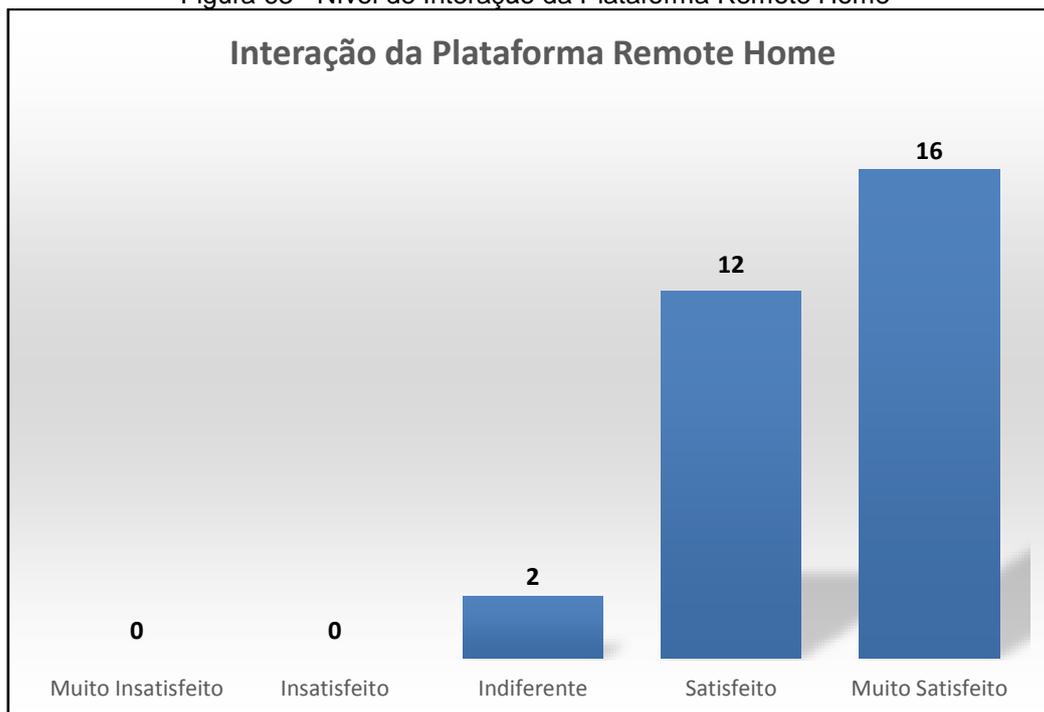
5.4.3 Avaliação da Interação

A interação da plataforma Remote Home foi analisada através da seguinte pergunta:

- Qual o seu nível de satisfação com a interação da Plataforma Remote Home?

De acordo com os dados da Figura 63, é possível constatar que 93% dos usuários ficaram satisfeitos ou muito satisfeitos com a interação entre os componentes que fazem parte da plataforma, enquanto que 7% informaram estar indiferentes.

Figura 63 - Nível de Interação da Plataforma Remote Home



Fonte: Autoria própria do autor desta dissertação (2016).

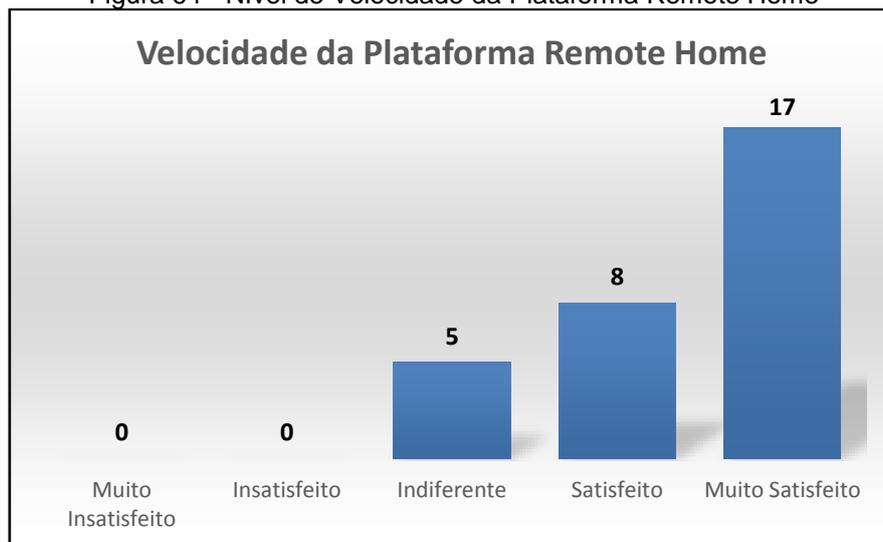
5.4.4 Avaliação da Velocidade de Execução

A velocidade de execução também é um ponto a ser avaliado, visto que é necessário identificar se os usuários ficaram satisfeitos com a velocidade de processamento e a comunicação entre os componentes da plataforma Remote Home. Desta maneira, a seguinte questão foi proposta aos participantes do experimento:

- Qual a sua satisfação em relação a velocidade de execução da Plataforma Remote Home?

Na Figura 64, é possível verificar o resultado obtido, identificando-se que 83% dos usuários ficaram satisfeitos ou muito satisfeitos com a velocidade da plataforma.

Figura 64 - Nível de Velocidade da Plataforma Remote Home



Fonte: Autoria própria do autor desta dissertação (2016).

5.5 CONSIDERAÇÕES FINAIS

Neste capítulo, foi apresentado o experimento realizado, além da análise qualitativa sobre o controle remoto universal e a plataforma Remote Home. Esse controle remoto foi avaliado através das métricas de eficiência, eficácia, satisfação, aprendizagem, operabilidade, acessibilidade, flexibilidade, utilidade e facilidade de uso (KRONBAUER; SANTOS, 2013). Para realizar as medições dos resultados, foi usada a escala de Likert (1932).

A partir dos resultados, percebeu-se que a plataforma e o aplicativo propostos neste projeto atenderam de maneira satisfatória aos anseios dos participantes do experimento, já que as notas foram positivas em todas as métricas avaliadas.

Um dado que reforça esta afirmação é o questionamento sobre a reutilização da plataforma Remote Home. Percebeu-se que 100% dos usuários informaram que o utilizariam novamente. Além disso, 97% dos usuários informaram que utilizariam a plataforma Remote Home permanentemente em relação a um controle remoto tradicional. Desta forma, percebe-se que a ferramenta foi validada com sucesso entre os usuários.

No próximo capítulo desta dissertação, serão demonstradas as conclusões, bem como os trabalhos futuros para a solução proposta.

6 CONCLUSÕES E TRABALHOS FUTUROS

A intenção desse trabalho foi desenvolver uma solução que forneça mais qualidade de vida para as pessoas, principalmente aquelas com dificuldade de movimento, idosos e crianças. Com a utilização da plataforma Remote Home, é possível oferecer uma solução que contempla conforto e comodidade para o usuário de forma que ele tenha o poder de gerenciar os dispositivos eletrônicos remotamente com comodidade e simplicidade.

Para chegar a este objetivo, foi realizado um estudo bibliográfico, descrito no Capítulo 3, no qual foram demonstrados diversos trabalhos na área de controle remoto universal para um ambiente residencial, tanto no âmbito acadêmico como para produtos comerciais.

No estudo, foi identificado que algumas soluções são simples sem a utilização de um software central (*middleware*), tais como: Chueh e Fanjiang (2012), Lamine e Abid (2014), Byun *et al.* (2015) e Satria *et al.* (2015). A falta de utilização do *middleware* pode trazer algumas limitações, dentre elas podemos destacar:

- a) Utilização de apenas uma tecnologia de rede sem fio para comunicação do controle remoto universal com os atuadores.
- b) Necessidade de um alto poder de processamento por parte do controle remoto universal para conseguir capturar os dados dos atuadores, montar um mapeamento dos mesmos com todas as informações e construir a tela para o usuário.
- c) Falta de segurança, tornando possível que um usuário mal-intencionado consiga enviar comandos aos dispositivos sem estar autorizado.

Outras abordagens, como o caso de Logitech (2016), Mobimax (2016) e Das *et al.* (2013), apresentam soluções que utilizaram a conexão direta entre o controle remoto universal e os dispositivos eletrônicos, sem a existência de atuadores, limitando as propostas em alguns aspectos, tais como:

- a) Limitação do recolhimento de algumas informações sobre um dispositivo eletrônico por parte do controle remoto universal;

- b) Impossibilidade de comunicação do controle remoto universal com os dispositivos eletrônicos através de tecnologias de redes diferentes, visto que a presença dos atuadores é o fator preponderante para a existência de mais de uma tecnologia de rede.

Como forma de criar uma solução completa, nesta dissertação, foi proposta uma abordagem que contempla um modelo e sua materialização em uma plataforma para controlar todos os dispositivos em uma residência. Ela está alicerçada no modelo de múltiplas camadas, proposto por Edwards e Foreword (1998). Inspirado nele, foram criadas as três camadas (interação, gerência e atuadores). Para validar a proposta, foi realizado um experimento com 30 usuários, objetivando dar validade a plataforma e avaliar a usabilidade do controle remoto universal. Os resultados apresentados durante o experimento comprovaram o sucesso da plataforma Remote Home.

Os usuários utilizaram o sistema com facilidade, comprovando a boa usabilidade do controle remoto universal. Além disso, afirmaram ter uma experiência agradável com as interações com a plataforma Remote Home, comprovando que a plataforma responde em tempo real aos comandos executados por seus usuários.

Nas próximas subseções, serão descritas as contribuições e limitações da plataforma Remote Home, além da discussão dos futuros trabalhos relacionados a esta pesquisa.

6.1 CONTRIBUIÇÕES

O principal legado deste trabalho refere-se à construção de um modelo e sua consolidação em uma plataforma para a área de Automação Residencial, com as seguintes contribuições:

- a) Criar uma plataforma que prevê a integração de diferentes dispositivos eletrônicos em um ambiente residencial, independente do seu modelo e/ou marca.
- b) Desenvolver uma plataforma que utiliza camadas independentes para gerenciar dispositivos eletrônicos.
- c) Controlar dispositivos que não possuem um controle remoto como dispositivo padrão.

- d) Disponibilizar um aplicativo com a finalidade de ser um controle remoto universal, permitindo a interação dos usuários de forma fácil e dinâmica.
- e) Construir uma interface para o aplicativo que se molda automaticamente em função dos dispositivos eletrônicos existentes no cenário de interação.
- f) Desenvolver um *middleware* que identifica automaticamente os dispositivos no ambiente.

No decorrer do desenvolvimento desta pesquisa, uma das estratégias utilizadas para verificar a aceitação do projeto por parte da comunidade científica, foi submeter artigos referentes aos resultados observados durante a sua realização. Essa ação possibilitou a obtenção de inúmeras contribuições que permitiram progressos na pesquisa. A seguir, estão listados em ordem cronológica, os artigos científicos publicados ao longo do desenvolvimento desta dissertação:

- Gomes, F. F. B., Kronbauer, A. H.; Araújo, B.B. **Interações via Dispositivos Móveis em um Ambiente Inteligente**. VII Fórum de Tecnologia da UNEB, Alagoinhas - BA, Brasil, outubro de 2014.
- Kronbauer, A. H.; Gomes, F. F. B.; Araújo, B. B. **Remote Home: Um Controle Remoto Universal Para Ambientes Residenciais**. XXI Simpósio Brasileiro de Sistemas Multimídia e *Web* - WebMedia, Manaus - AM, Brasil, outubro de 2015.

6.2 LIMITAÇÕES

Quando se realiza uma pesquisa científica, sempre irão aparecer limitações para a solução proposta. Os limites servem como motivação para que novos pesquisadores continuem a investigar e melhorar a solução proposta inicialmente. Dessa maneira, ao utilizar a plataforma Remote Home, percebe-se que ela precisa de pontos de melhoria em alguns aspectos, conforme pode ser visto a seguir:

- a) O controle remoto universal limita-se apenas ao sistema operacional Android, sendo necessário novas implementações para Windows OS, Blackberry OS e iOS.
- b) As interfaces criadas carecem de algumas melhorias gráficas, no estágio atual são rudimentares, servindo apenas para validar o modelo proposto. Para uma

versão comercial, seria interessante ter a participação de um especialista na área de *designer*, acrescentando melhorias gráficas ao aplicativo.

- c) Os participantes do experimento foram alunos e professores que não possuíam restrições físicas, seria importante realizar um teste da plataforma com pessoas que tivessem restrições de locomoção.
- d) O experimento foi realizado com apenas dois atuadores e dispositivos eletrônicos, sendo necessária a realização de novos experimentos com uma quantidade maior de atuadores e uma variedade maior de dispositivos eletrônicos.

6.3 TRABALHOS FUTUROS

A popularização da área de Automação Residencial está intrinsecamente ligada a evolução da eletrônica e da computação, que vem tornando os dispositivos computacionais cada vez menores e mais rápidos. Além disso, são criados cada vez mais aplicativos móveis e a utilização de *smartphones* tem crescido exponencialmente. Como resultado, as soluções que utilizam plataformas móveis podem se tornar mais baratas para a criação de soluções na área de domótica.

Para dar continuidade a este trabalho, faz-se necessário realizar as melhorias citadas na seção 6.2, sendo importante ressaltar o desenvolvimento dessa solução para os sistemas operacionais: Blackberry OS, iOS ou Windows OS. Além disso, o teste da plataforma Remote Home com outros modelos de atuadores aumentará a sua abrangência de utilização.

Outro ponto importante a ser destacado é a utilização de outras modalidades de interação, já que esse é um dos principais recursos disponibilizados pelo *middleware*. Isso torna possível que pessoas que tenha problemas físicos ou cognitivos possam utilizar o cenário de interação, mesmo com restrições de visão, audição, fala e coordenação motora. Neste sentido, basta integrar ao ambiente outros sensores que possibilitem capturar comandos via gestos, voz ou ondas cerebrais.

Por fim, pretende-se que os resultados encontrados durante a realização desta pesquisa sejam apresentados em novos trabalhos acadêmicos, tais como: artigos

científicos e periódicos. Desta forma, espera-se que o trabalho forneça contribuições que possam ser utilizadas para a melhoria contínua da solução proposta nesta dissertação, contemplando novas expectativas e anseios dos especialistas nas áreas de Automação Residencial e Interação Humano-Computador.

REFERÊNCIAS

AHSAN, K. et al. UniCon Remote Control Model – A Mobile System for Assistive Technology. **Research Journal of Recent Sciences**, Karachi, p. 95-102, 2014.

ANDROID. **The Android Source Code**, 2016. Disponível em: <<https://source.android.com/source/index.html>>. Acesso em: 4 nov. 2016.

BETIOL, A. H.; CYBIS, W. A. Usability Testing of mobile devices: a comparison of three approaches. In: HUMAN-COMPUTER INTERACTION - INTERACT, Roma, 2005. **Anais...** 2005. 470-481.

BITTINS, B.; SIECK, J.; HERZOG, M. Supervision and Regulation of Home Automation Systems with Smartphones. In: UKSIM FOURTH EUROPEAN MODELLING SYMPOSIUM ON COMPUTER MODELLING AND SIMULATION, 2010. **Anais...** 2010. p. 444-448.

BOLZANI, C. A. M. **Residências inteligentes**. São Paulo: Livraria da Física, 2007.

BYUN, S. B.; KIM, Y. H.; LEE, Y. H. Development of Remote Control Dongle Connecting to Smartphone. **International Journal of Information and Electronics Engineering**, p.295-299, 2015.

CAMPOS, A. L. P. S.; CABRAL, M. M. A. Sistemas de Automação Residencial de Baixo Custo: Uma Realidade Possível. **HOLOS**, Instituto Federal do Rio Grande do Norte, v. 3, Natal, p.26-32, 2009.

CHEN, N.; LIN, J.; HOI, S. C. H.; XAO, X.; ZHANG, B. AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 36., Hyderabad, 2014. **Proceedings...** 2014. p.767-778.

CHUEH, T. F.; FANJIANG, Y. Y. Universal Remote Control on Smartphone. In: INTERNATIONAL SYMPOSIUM ON COMPUTER, CONSUMER AND CONTROL, 2012, Taichung. **Proceedings...** 2012. p. 658-661.

CNET. **Gear 4 Unity Remote**. 2016. Disponível em: <<http://www.cnet.com/uk/products/gear4-unityremote/>>. Acesso em: 15 jun. 2016.

COOK, D. J.; AUGUSTO, J. C.; JAKKULA, V. R. Ambient Intelligence: technologies, applications and opportunities. **Pervasive and Mobile Computing**, Amsterdã, 2009, 277-298.

COUSARIS, C. K.; KIM, D. J. A Meta-Analytical Review of Empirical Mobile Usability Studies. **Journal Usability Studies**, v.6, p. 117-171, 2011.

CRACKBERRY. **The Evolution of Blackberry in Pictures**. Disponível em: <<http://crackberry.com/evolution-blackberry-pictures>>. Acesso em: 6 ago. 2016.

DA SILVA, E. L.; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. Florianópolis: Universidade Federal de Santa Catarina, 2005.

DAS, S.R. et al Home Automation and Security for Mobile Devices. In: IEEE PerCom Workshop on Pervasive Communities and Service Clouds, 1., 2013. **Proceedings...** 2013. p.141-146.

DESAI, N. Automated Selection of Remote Control User Interfaces in Pervasive Smart Spaces. In: OF THE HCIC WINTER WORKSHOP. 2002. **Proceedings...** 2002. p. 1-12.

DO, T. M. T.; BLOM, J.; GATICA-PEREZ, D. Smartphone usage in the wild: a large scale analysis of applications and context. In: INTERNATIONAL CONFERENCE ON MULTIMODAL INTERFACES, 13., Nova York, 2011. **Proceedings...** 2011. p.353-360.

EDWARDS, J.; FOREWORD, R. EDWARDS, Jeri; FOREWORD BY-ORFALI, Robert. **3 Tier Client/Server at Work**. [S.l.]: John Wiley & Sons, Inc., 1998.

HANSEN, J. **An investigation of smartphone applications**: exploring usability aspects related to wireless personal area networks, context-awareness, and remote information access. 2012. Tese (Doutorado)- Brunel University, School of Information Systems, Computing and Mathematics. Brunel, 2012.

IEEE. **IEEE803.11 - Wireless Local Area Networks**. Disponível em: <<http://www.ieee802.org/11/>>. Acesso em: 7 ago. 2016.

IRDROID. **Irdroid**. Disponível em: <www.irdroid.com>. Acesso em:03 jun. 2016.

IRULE. **iRule**. Disponível em: <<http://getirule.com/how-it-works/>>. Acesso em: 3 jun. 2016.

ILOUNGE. **New Potato Technologies FLPR Universal Remote for iPod touch + iPhone**. Disponível em: <<http://www.ilounge.com/index.php/reviews/entry/new-potato-technologies-flpr-universal-remote-for-ipod-touch-iphone/>>. Acesso em: 20 jun. 2016.

ISO/IES 9126-1. Software Engineering - Software Product Quality - Part 1: Quality Model. [S.l.]: International Organization for Standardization, 2001.

JACKSON, W. **Android Apps for Absolute Beginners**. Nova York: Apress, 2011.

JADHAV, A. et al. **Universal Mobile Application Development (UMAD) On Home Automation**. Pune: Marathwada Mitra Mandal's Institute of Technology, University of Pune, India Network and Complex Systems, 2012.

JOHNSON, K. et al. **The Innovative Success that is Apple, Inc.** Huntington: Marshall University, Huntington, 2012.

LEVINSON, J. **Building Client/Server Applications with VB.NET**: an example-driven approach. Nova York: Apress, 2003.

KOO, B. et al. R-URC: RF4CE-based Universal Remote Control Framework using Smartphone. In: INTERNATIONAL CONFERENCE OF COMPUTATIONAL SCIENCE AND ITS APPLICATIONS. 2010, Seoul. **Proceedings...** 2010.

KRONBAUER, A. H.; SANTOS, C. A. S. Uma análise das abordagens para avaliar a usabilidade de smartphones: estado da arte e novas tendências. In: SIMPÓSIO BRASILEIRO SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 2013, Manaus. **Anais...** 2013. p. 452-461.

KRONBAUER, A. H.; SANTOS, C. A. S. Um modelo de avaliação da usabilidade baseado na captura automática de dados de interação do usuário em ambientes reais. In: BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, 10., 2011, Porto de Galinhas. **Anais...** 2011. p.114-123.

LAMINE, Hayet; ABID, Hafedh. Remote Control of a Domestic Equipment From an Android Application Based on Raspberry Pi Card. In: INTERNATIONAL CONFERENCE ON SCIENCES AND TECHNIQUES OF AUTOMATIC CONTROL & COMPUTER ENGINEERING, 15., 2014, Hammanet. **Proceedings...** 2014. p. 903-908.

LEITTE, JAMIESON DA P. et al. Automadroid-Automação residencial com dispositivos móveis. **Revista de Controle e Automação**, Belém, v. 1, n. 1, 2013.

LIKERT, R. A Technique for the Measurement of Attitudes. **Archives of Psychology**, 1932.

LOGITECH. **Logitech**. Disponível em: <www.logitech.com>. Acesso em: 2 jun. 2016.

MARELI, A. et al. Um framework de desenvolvimento de aplicações ubíquas em ambientes inteligentes. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 31., Brasília, 2013. **Anais...** 2013. p. 643-656.

MARTIN, T. **The Evolution of Smartphone**. 2014. Disponível em: <<http://pocketnow.com/2014/07/28/the-evolution-of-the-smartphone#toc-0>>. Acesso em: 0 ago. 2016.

MENDONÇA, V. R. L.; BITTAR, T. B.; DIAS, M. de S. Um estudo dos Sistemas Operacionais Android e iOS Para o Desenvolvimento de Aplicativos. In: ENCONTRO ANUAL DE COMPUTAÇÃO - ENACOMP, Catalão, 2011. **Anais...** 2011.

MELLOWARE. **X10 Commander - Control Your X10 devices anywhere in the world**. Disponível em: <<http://melloware.com/x10commander/>>. Acesso em: 2 jun. 2016.

MOBIMAX. **L5 Universal Remote**. Disponível em: <www.mobimax.com.br>. Acesso em: 2 jun. 2016.

MURATORI, J. R. ; BÓ, P. H. D. Automação residencial: conceitos e aplicações. **Educere**, 2013.

NETO, M. C. M. Desenvolvendo Aplicações Ubíquas com Arduino e Raspberry Pi. In: BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB, 20., João Pessoa, 2014. **Proceedings...** 204.

NICHOLS, J.; MAYERS, B. A. Controlling home and office appliances with smart phones. **IEEE Pervasive Computing**, v. 5, n. 3, p. 60-67, 2006.

PANTH, S.; JIVANI, M. Home Automation System (HAS) using Android for Mobile Phone. **International Journal of Electronics and Computer Science Engineering**, 2011.

PHONESDATA, **All Brands of Smartphones**. Disponível em: <<http://phonesdata.com/en/smartphones>>. Acesso em: 5 ago. 2016.

POOJA, D.; LATHKAR, Y. V.; PURUSHOTTAM, D. Smart Home Using Android Application. **International Journal of Research in Engineering and Technology**, Pune, 2014.

RAJABZADEH, A.; MANASHTY, A. R.; JAHROMI, Z. F. **A Mobile Application for a Smart House Remote Control System**. World Academy of Science, Engineering and Technology, Kermanshah, 2010.

RASPBERRY PI. **Raspberry Pi**. Disponível em: <www.raspberrypi.org>. Acesso em: 5 ago. 2016.

REED, B. **A Brief History of Smartphones - How the smartphone went from a high-end enterprise device to an everyday consumer staple**. Disponível em: <http://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html>. Acesso em: 6 ago. 2016.

RIEKKI, J.; SANCHÉZ, I.; PYYKKONEN, M. **Touch & Control: Interacting with Services by Touching RFID Tags**. Ubiquitous Intelligence and Computing. Lecture Notes in Computer Science. Berlim: [s.n.], 2008. p.63-577.

ROY, J.; ROY, K. N. Design of Smart Universal Remote using Mobile for Home Automation. **Journal of Computare Engineering**, 2014, 73-80.

SALARIAN, H.; CHIN, K. W.; NAGHDY, H. Coordination in wireless sensor–actuator networks: A survey. **Journal of Parallel and Distributed Computing**, v. 72, n. 7, p. 856-867, 2012.

SAMSUNG. **Samsung AllShare Control**. Disponível em: <<http://www.samsung.com/br/apps/mobile/allshare>>. Acesso em: 3 ago. 2016.

SATRIA, A. et al. The framework of Home Remote Automation System based on Smartphone. **International journal of smart home**, v. 9, n. 1, p. 53-60, 2015.

SEITER, E. et al. **Remote control: television, audiences, and cultural power**. Routledge, 2013.

SGARBI, J. A.; TONIDANDEL, F. Domótica Inteligente: automação residencial baseada em Comportamento. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 15., Gramado, 2004. **Anais...** 2004.

SHARP, H.; ROGERS, Y.; PREECE, J. **Interaction Design: Beyond Human-Computer Interaction**. 3rd. New York: John Wiley & Sons. 2011.

SHASTRI, D. Home Automation System with Universally Used Mobile Application Platform. **Journal of Electronics and Communication Engineering**, Amroha, 2014, 1-6.

SOMANI, N. A.; PATEL, Y. Zigbee: A Low Power Wireless Technology For Industrial Applications. **International Journal of Control Theory and Computer Modelling**, 2012.

THE STATISTICS PORTAL. **Number of apps available in leading app stores as of June 2016**, Disponível em: <<http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>>. Acesso em: 1 jul. 2016.

WAVESPARK. **IR Universal Remote**. Disponível em: <<http://iruniversalremote.com>>. Acesso em: 10 ago. 2016.

WEISER, M. The Computer For The 21st Century. **Scientific American**, p. 78-89, 1991.

WAKKARY, R. et al. An Ambient Intelligence Platform for Physical Play. In: ANNUAL ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA, 13., Cingapura, 2005. **Proceedings...** 2005. p.764-773.

WAYHOUSE, **Wayhouse Automação Residencial**. Disponível em: <<http://wayhouse.com.br>>. Acesso em: 3 jul. 2016.

YAN, M.; SHI, H. Smart Living Using Bluetooth-Based Android Smartphone, **International Journal of Wireless & Mobile Networks**, Melbourne, 2013.

ZHONG, Y. et al. Smart Home on Smart Phone. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING, 13., Pequim, 2011. **Proceedings...** 2011. p. 467-468.

ANEXO A – DADOS DO PROJETO DE PESQUISA

UNIVERSIDADE SALVADOR -
UNIFACS/BA



PARECER CONSUBSTANCIADO DO CEP

DADOS DO PROJETO DE PESQUISA

Título da Pesquisa: UMA PLATAFORMA PARA INTERAÇÕES NATURAIS EM AMBIENTES

Pesquisador: Artur Henrique Kronbauer

Área Temática:

Versão: 1

CAAE: 51137215.4.0000.5033

Instituição Proponente: Universidade Salvador - UNIFACS/BA

Patrocinador Principal: Financiamento Próprio

DADOS DO PARECER

Número do Parecer:1.366.396

Apresentação do Projeto:

O projeto intitulado "Uma plataforma para interações naturais em ambientes inteligentes" contextualiza a computação ubíqua enquanto um mecanismo que facilite a interação entre usuários e dispositivos computacionais de modo que os comandos disparados pelos usuários sejam os mais naturais possíveis. E que para proporcionar esta funcionalidade, os sistemas ubíquos deverão capturar informações sobre o ambiente para dinamicamente se adaptar ao contexto e automaticamente executar ações apropriadas a cada mudança no cenário de interação.

Objetivo da Pesquisa:

O objetivo geral deste projeto é pesquisar e desenvolver alternativas tecnológicas para a criação de uma plataforma multimodal, possibilitando a interação de pessoas com sistemas computacionais de forma natural, utilizando gestos, voz e toques em dispositivos móveis. Já os objetivos específicos são: - Criar um middleware que identifique automaticamente os sensores (equipamentos para captura de imagens, sons e dispositivos móveis), os atuadores (microcontroladores que possam acionar funcionalidades em dispositivos eletrônicos) e possa fazer a comunicação entre eles.

- Construir uma infraestrutura para o tratamento de imagens, sons e acionamentos remotos via dispositivos móveis, permitindo a interpretação de comandos executados pelos usuários no cenário de interação.

Continuação do Parecer: 1.366.396

Avaliação dos Riscos e Benefícios:

O questionário refere-se apenas a critérios de usabilidade e desempenho da plataforma, não se utilizando de métodos invasivos ou que exponha as participantes, dessa forma não se observa riscos aparentes para o conjunto amostral da pesquisa. No tocante aos benefícios, após o desenvolvimento do projeto será possível identificar se a plataforma proposta contempla índices de usabilidade aceitáveis, tem a potencialidade de ajudar na inclusão social e pode auxiliar a vida das pessoas em tarefas do cotidiano.

Comentários e Considerações sobre a Pesquisa:

Com o progresso dos sistemas embarcados e das tecnologias que utilizam redes sem fio, percebe-se o avanço na área de Ambientes Inteligentes e a evolução dos estudos de interações entre os seres humanos e dispositivos eletrônicos por meio de ações naturais. Este cenário favorece o desenvolvimento da computação ubíqua que está pautada na idéia da computação estar presente em todos os locais e ser transparente

aos seres humanos. Assim como a evolução dos motores, que hoje estão presentes na maioria dos artefatos que cercam as pessoas, mas são imperceptíveis, os sistemas computacionais devem seguir o mesmo caminho e tornarem-se onipresentes.

Considerações sobre os Termos de apresentação obrigatória:

O projeto apresenta todos os termos de apresentação obrigatória.

Recomendações:

Sem recomendações.

Conclusões ou Pendências e Lista de Inadequações:

Projeto eticamente apto para desenvolvimento.

Considerações Finais a critério do CEP:

Mantido parecer do relator.

Este parecer foi elaborado baseado nos documentos abaixo relacionados:

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_602604.pdf	19/11/2015 16:49:48		Aceito
Outros	Lattes_Fabio_Gomes.pdf	19/11/2015 16:48:49	Artur Henrique Kronbauer	Aceito
Outros	Lattes_Italo_Ribeiro.pdf	19/11/2015 16:47:57	Artur Henrique Kronbauer	Aceito

UNIVERSIDADE SALVADOR -
UNIFACS/BA



Continuação do Parecer: 1.366.396

Outros	Lattes_Igor_Pimenta.pdf	19/11/2015 16:47:28	Artur Henrique Kronbauer	Aceito
Outros	Lattes_Artur_Kronbauer.pdf	19/11/2015 16:46:46	Artur Henrique Kronbauer	Aceito
Declaração de Instituição e Infraestrutura	Declaracao_de_Anuencia.pdf	19/11/2015 11:24:14	Artur Henrique Kronbauer	Aceito
Folha de Rosto	Folha_de_Rosto.pdf	19/11/2015 11:21:36	Artur Henrique Kronbauer	Aceito
Orçamento	Orcamento.pdf	16/11/2015 13:08:21	Artur Henrique Kronbauer	Aceito
Cronograma	Cronograma.pdf	16/11/2015 12:45:26	Artur Henrique Kronbauer	Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	Termo_de_Consentimento_Livre_e_Escl arecido.pdf	16/11/2015 10:59:08	Artur Henrique Kronbauer	Aceito
Projeto Detalhado / Brochura Investigador	Projeto_Detalhado.pdf	16/11/2015 10:50:50	Artur Henrique Kronbauer	Aceito

Situação do Parecer:

Aprovado

Necessita Apreciação da CONEP:

Não

SALVADOR, 14 de Dezembro de 2015

Assinado por:

TATIANA SENNA GALVÃO NONATO ALVES
(Coordenador)

APÊNDICE A – QUESTIONÁRIO PROPOSTO AOS PARTICIPANTES DO EXPERIMENTO



Avaliação da Plataforma

Programa de Pós-Graduação em Sistemas e Computação (PPGCOMP)
Laureate International Universities
Avenida Luís Viana Filho, Nº 3146, Imbuí - CEP: 41720-200 - Salvador-Bahia.
Telefone (71) 3021-2800.

Questionário Proposto aos Participantes do Experimento

Local: _____ Data: ___/___/___

Parte 1 - Dados Pessoais

- 1) Nome: _____
- 2) Idade: _____
- 3) Curso de Graduação: _____
- 4) Ano de Formação: _____
- 5) Atividade Profissional: _____
- 6) Empresa onde trabalha: _____
- 7) Sexo: () Masculino () Feminino

Parte 2 – Avaliação de Usabilidade

8) Qual o seu nível de satisfação com a rapidez (eficiência) com que consegue realizar as tarefas através do controle remoto universal?

	Muito Satisfeito
	Satisfeito
	Indiferente
	Insatisfeito
	Muito Insatisfeito

9) Qual o seu nível de satisfação com a precisão (eficácia) com que consegue executar as tarefas desejadas no controle remoto universal?

	Muito Satisfeito
	Satisfeito
	Indiferente

	Insatisfeito
	Muito Insatisfeito

10) Qual é seu nível de satisfação com a utilização do controle remoto universal?

	Muito Satisfeito
	Satisfeito
	Indiferente
	Insatisfeito
	Muito Insatisfeito

11) Qual o seu nível de satisfação com a aprendizagem das funcionalidades do controle remoto universal?

	Muito Satisfeito
	Satisfeito
	Indiferente
	Insatisfeito
	Muito Insatisfeito

12) Quando houve uma interação indesejada, você conseguiu facilmente retornar para um estado anterior com o objetivo de iniciar a tarefa novamente?

	Muito Fácil
	Fácil
	Médio
	Difícil
	Muito Difícil

13) Qual o seu nível de satisfação com a acessibilidade disponibilizada pelo controle remoto universal?

	Muito Satisfeito
	Satisfeito
	Indiferente
	Insatisfeito
	Muito Insatisfeito

14) Qual o seu nível de satisfação com relação à flexibilidade (caminhos alternativos para executar uma tarefa) disponíveis no controle remoto universal?

	Muito Satisfeito
--	------------------

	Satisfeito
	Indiferente
	Insatisfeito
	Muito Insatisfeito

15) Qual o seu nível de satisfação com a utilidade do controle remoto universal?

	Muito Satisfeito
	Satisfeito
	Indiferente
	Insatisfeito
	Muito Insatisfeito

16) Qual a sua percepção com relação a facilidade de uso do controle remoto universal?

	Muito Fácil
	Fácil
	Médio
	Difícil
	Muito Difícil

Parte 3 – Questões Referentes a Plataforma Remote Home

17) Você prefere utilizar a Plataforma Remote Home ao invés de um controle remoto tradicional?

	Não
	Sim

18) Você voltaria a utilizar a Plataforma Remote Home em outra ocasião?

	Não
	Sim

19) Qual o seu nível de satisfação com a interação da Plataforma Remote Home?

	Muito Satisfeito
	Satisfeito

	Indiferente
	Insatisfeito
	Muito Insatisfeito

20) Qual a sua satisfação em relação a velocidade de execução da Plataforma Remote Home?

	Muito Satisfeito
	Satisfeito
	Indiferente
	Insatisfeito
	Muito Insatisfeito

21) Você tem alguma consideração que gostaria de relatar para aprimorar o desenvolvimento da plataforma proposta?

APÊNDICE B - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO



Termo de Consentimento Livre e Esclarecido

Comitê de Ética em Pesquisa da UNIFACS
 Programa de Pós-Graduação em Sistemas e Computação (PPGCOMP)
 Laureate International Universities
 Avenida Luís Viana Filho, Nº 3146, Imbuí - CEP: 41720-200 - Salvador-Bahia.
 Telefone (71) 3021-2800.

Remote Home: Um Controle Remoto Residencial Universal

Eu _____; estou sendo convidado (a) a participar de um experimento para avaliar a usabilidade e aspectos práticos relacionados a utilização de uma plataforma de interação baseada em interações em um dispositivo móvel via toques em ambiente inteligente, desenvolvida para facilitar a interação das pessoas com dispositivos eletrônicos em ambientes domésticos ou empresariais.

A plataforma foi concebida como projeto de pesquisa do professor Dr. Artur Henrique Kronbauer, vinculado ao Programa de Pós-Graduação em Sistemas Computacionais (PPGCOMP) da Universidade Salvador. O referido pesquisador pode ser contatado pelo telefone (71) 99111-8409 ou pelo endereço de e-mail arturhk@gmail.com.

Recebi esclarecimentos sobre a pesquisa e estou ciente de que minha privacidade será respeitada, ou seja, meu nome e imagem serão mantidos em sigilo. Eu autorizo a utilização do questionário contendo as minhas respostas preenchidas durante a realização do experimento, entendendo que as informações serão utilizadas somente para os fins desta pesquisa, bem como serão divulgadas apenas em artigos e na redação dos trabalhos científicos orientados pelo professor Dr. Artur Henrique Kronbauer.

Estou ciente que poderei solicitar esclarecimentos quanto a quaisquer dúvidas durante a realização do experimento e terei acesso aos resultados obtidos. Tenho ciência de que poderei me recusar a responder qualquer pergunta e que posso me negar a participar do estudo ou retirar meu consentimento a qualquer momento, sem prévia justificativa.

Manifesto meu livre consentimento em participar.

Salvador, _____ de _____ de _____.

Nome e assinatura do participante

Nome e assinatura do pesquisador

APÊNDICE C - ROTEIRO PARA UTILIZAÇÃO DA PLATAFORMA



UNIFACS
UNIVERSIDADE SALVADOR
LAUREATE INTERNATIONAL UNIVERSITIES

Roteiro para utilização da plataforma

Programa de Pós-Graduação em Sistemas e Computação (PPGCOMP)
Laureate International Universities
Avenida Luís Viana Filho, Nº 3146, Imbuí - CEP: 41720-200 - Salvador-Bahia.
Telefone (71) 3021-2800.

Remote Home: Um Controle Remoto Residencial Universal

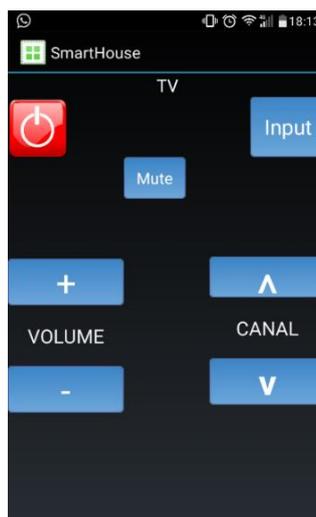
O experimento será realizado nas dependências da UNIFACS, no período de 16/08 até 19/08. Os usuários serão orientados pelo aluno do Mestrado de Sistemas e Computação Fabio Fonseca Barbosa Gomes sobre os procedimentos de utilização das interfaces para realizar o experimento.

O experimento será composto de:

- 1 *smartphone*;
- 1 dispositivo central (*middleware*);
- 2 atuadores;
- 2 eletrodomésticos (televisão e ar condicionado);

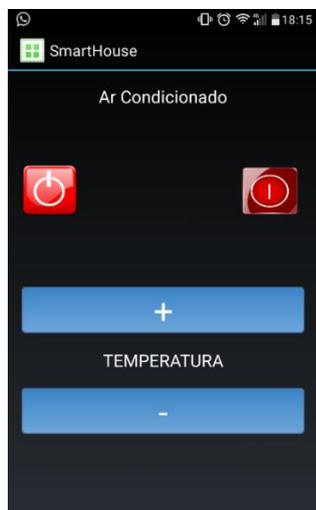
O roteiro do experimento está composto pelas seguintes etapas:

- 1) Explicação do projeto ao participante.
- 2) Assinatura do Termo de Consentimento Livre e Esclarecido;
- 3) Explicação das tarefas com as interfaces do controle universal, que seguirá os seguintes passos:
 - a. Iniciar aplicativo.
 - b. Selecionar a interação com a Televisão
 - c. (i) Utilização do botão  para ligar o eletrodoméstico.
 - d. (ii) Aumentar ou diminuir volume com os botões + ou - .
 - e. (iii) Utilizar novamente o botão  para desligar a televisão.
 - f. A imagem da interface de televisão é apresentada abaixo:



g. O usuário voltará para a tela anterior utilizando o botão de retornar em seu smartphone e irá selecionar o botão de ar condicionado.

h. Ligar o Ar Condicionado utilizando o botão  e aumentar ou diminuir a temperatura, a interface do ar condicionado é apresentada na figura abaixo:



i. Finalmente o usuário irá encerrar a aplicação.

- 4) Após a realização das tarefas, os usuários irão responder a um questionário.
- 5) Os usuários serão dispensados do experimento quando terminarem de responder ao questionário.

APÊNDICE D - CÓDIGO-FONTE DO CONTROLE REMOTO UNIVERSAL

Classe Acionamento

```
package com.aml.smarthouse;
import org.json.JSONException;
import org.json.JSONObject;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class Acionamento extends Activity
{ //Início da Principal
    Button bt01, bt02;
    TextView tv08;
        String ipRasp;
        String ipAtuador;
        String idAtuador;
        String nomeAtuador;
    AsyncHttpClient netClient = null;

    int delay;
    Handler h;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    { //Botões e ferramentas
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tomada);
        getViews(); //Inicia botões, textviews...

        Intent iDados = getIntent();
```

```

if (iDados != null)
{
    Bundle parametros = iDados.getExtras();
    if (parametros != null)
    {
        ipRasp= parametros.getString("ipRasp");
        ipAtuador= parametros.getString("ipAtuador");
        idAtuador= parametros.getString("idAtuador");
        nomeAtuador= parametros.getString("nomeAtuador");
    }
}

netClient = new AsyncHttpClient();
bt01 .setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v)
    {
        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        //http://localhost/get?atuador=12&acao=123
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=01",new
AsyncHttpResponseHandler()
        {
        }
        );
    }
});

bt02 .setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v)
    {
        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=0",new
AsyncHttpResponseHandler()
        {
        }
    }
});

```

```

        );
    }
});
} //fim da OnCreate

public void getViews()
{
    tv08 = (TextView)findViewById(R.id.temperaturaText);
    bt01 = (Button)findViewById(R.id.bt01);
    bt02 = (Button)findViewById(R.id.bt02);
}
} // fim da principal

```

Classe Ar

```

package com.aml.smarthouse;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;

public class ar extends Activity{

    //Início da Principal

    Button bt01, bt02, bt03, bt04;
    TextView tv08;
    String ipRasp;
    String ipAtuador;
    String idAtuador;
    String nomeAtuador;
    AsyncHttpClient netClient = null;

    int delay;

```

```

Handler h;

@Override
protected void onCreate(Bundle savedInstanceState)
{//Botões e ferramentas
    super.onCreate(savedInstanceState);
    setContentView(R.layout.ar);
    getViews(); //Inicia botões, textviews...
    Intent iDados = getIntent();
    if (iDados != null)
    {
        Bundle parametros = iDados.getExtras();
        if (parametros != null)
        {
            ipRasp= parametros.getString("ipRasp");
            ipAtuador= parametros.getString("ipAtuador");
            idAtuador= parametros.getString("idAtuador");
            nomeAtuador= parametros.getString("nomeAtuador");
        }
    }

    netClient = new AsyncHttpClient();
    bt01 .setOnClickListener(new OnClickListener() { // Ligar Ar
        @Override
        public void onClick(View v)
        {
            if (netClient==null)
                return;

            netClient.setTimeout(2000);
            http://localhost/get?atuador=12&acao=123
            netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=04",new
AsyncHttpResponseHandler()
            {
            }
            );
        }
    });

    bt02 .setOnClickListener(new OnClickListener() { // Desliga Ar
        @Override

```

```

public void onClick(View v)
{
    if (netClient==null)
        return;
    netClient.setTimeout(2000);
    netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=05",new
AsyncHttpResponseHandler()
    {
        }
    );
}
});

```

```

bt03 .setOnClickListener(new OnClickListener() { // aumenta temperatura do Ar
@Override
public void onClick(View v)
{
    if (netClient==null)
        return;
    netClient.setTimeout(2000);
    netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=6",new
AsyncHttpResponseHandler()
    {
        }
    );
}
});

```

```

bt04 .setOnClickListener(new OnClickListener() { // diminui temperatura do Ar
@Override
public void onClick(View v)
{
    if (netClient==null)
        return;
    netClient.setTimeout(2000);
    netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=7",new
AsyncHttpResponseHandler()
    {
        }
    );
}
});

```

```

    });
} //fim da OnCreate

public void getViews()
{
    tv08 = (TextView)findViewById(R.id.temperaturaText);
    bt01 = (Button)findViewById(R.id.bt01);
    bt02 = (Button)findViewById(R.id.bt02);
    bt03 = (Button)findViewById(R.id.bt03);
    bt04 = (Button)findViewById(R.id.bt04);
}
} // fim da principal

```

Classe Atuadores

```

package com.aml.smarthouse;

public class Atuadores
{
    public String id;
    public String ip;
    public String nome;

    public Atuadores()
    {
        setId("");
        setIp("");
        setNome("");
    }

    public Atuadores(String Id, String Ip, String nome)
    {
        setId(id);
        setIp(ip);
        setNome(nome);
    }

    public String getId()
    {
        return id;
    }

    public void setId(String id)
    {
        this.id = id;
    }

    public String getIp() {
        return ip;
    }
}

```

```

        public void setIp(String ip) {
            this.ip = ip;
        }

        public String getNome() {
            return nome;
        }

        public void setNome(String nome) {
            this.nome = nome;
        }
    }

```

Classe Controle

```

package com.aml.smarthouse;

import java.util.ArrayList;

public class Controle
{

    public String id;
    public String ip;
    public String nome;

    private ArrayList<Atuadores> atuadores;

    public Controle()
    {
        id="";
        ip="";
        nome="";
        atuadores = new ArrayList<Atuadores>();
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getIp() {
        return ip;
    }

    public void setIp(String ip) {
        this.ip = ip;
    }

    public String getNome() {
        return nome;
    }
}

```

```

public void setNome(String nome) {
    this.nome = nome;
}

public ArrayList<Atuadores> getAtuadores() {
    return atuadores;
}

public void setAtuadores(ArrayList<Atuadores> atuadores) {
    this.atuadores = atuadores;
}

public boolean add(Atuadores object) {
    return atuadores.add(object);
}

public void clear() {
    atuadores.clear();
}

public Atuadores get(int index) {
    return atuadores.get(index);
}

public Atuadores getAtuadores(int pos) {
    return atuadores.get(pos);
}

public int size() {
    return atuadores.size();
}

public Atuadores set(int index, Atuadores object) {
    return atuadores.set(index, object);
}
}

```

Classe CustomListAdapter

```

package com.aml.smarthouse;

import java.util.ArrayList;
import android.app.Activity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class CustomListAdapter extends ArrayAdapter<String> {

```

```

private final Activity context;
private final ArrayList<String> itemname;
private final ArrayList<Integer> imgid;

public CustomListAdapter(Activity context, ArrayList<String> itemname, ArrayList<Integer>
imgid) {
    super(context, R.layout.mylist, itemname);
    // TODO Auto-generated constructor stub
    this.context=context;
    this.itemname=itemname;
    this.imgid=imgid;
}

public View getView(int position,View view,ViewGroup parent) {
    LayoutInflater inflater=context.getLayoutInflater();
    View rowView=inflater.inflate(R.layout.mylist, null,true);
    TextView txtTitle = (TextView) rowView.findViewById(R.id.item);
    ImageView imageView = (ImageView) rowView.findViewById(R.id.icon);
    txtTitle.setText(itemname.get(position));
    imageView.setImageResource(imgid.get(position));
    return rowView;
};
}

```

Classe MainActivity

```

package com.aml.smarthouse;

import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.ArrayList;
import java.util.Enumeration;
import org.apache.http.conn.util.InetAddressUtils;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import android.app.Activity;
import android.content.Context;

```

```
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.NetworkInfo.DetailedState;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity implements OnItemClickListener
{
    byte conexao;
    boolean bOnline = false;
    boolean bAutoscan = false;
    boolean bNetworking = false;
    boolean bConnected = false;
    boolean bFromPref = false;
    boolean bContinuous = false;
    boolean resposta=false;

    private ListView list;
    String exactUrl = null;      // exemplo http://10.0.0.14/hometech
    String subnetUrl = null;    // exemplo http://10.0.0.*/hometech
    String pingUrl = null;
    int iScanTimeout,iContInterval;
    int pingFrom,pingTo;
    int px;

    TextView tv01;

    AsyncHttpClient netClient = null;
    Controle controle;
```

```

CustomListAdapter adapter;
ArrayList<String> listAtuador;
ArrayList<Integer> listImg;

@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    inicializa();
    VerificaConexao(); //Função que verifica qual o tipo de conexão 3g ou wifi
    netClient = new AsyncHttpClient();
    controle = new Controle();
    tv01.setText("Aguarde"); //O nome da textView é tv01
    if(conexao==2)
    {
        Toast.makeText(this, "Em breve opção para usar pelo 3g" ,
Toast.LENGTH_LONG).show();
        finish();
    }

    if(conexao==1)
    {
        //O dispositivo está conectado via Wi-Fi
        //Toast.makeText(this, "WiFi" , Toast.LENGTH_LONG).show();
        //ip = "http://" + ip.substring(0, idx1);
        bOnline = isNetworkPresent();
        setupVars();
        conex();
        if (resposta==false)
            conex();
    }
} //Fim da onCreate
//Imprimir Atuador
public void imprimirAtuador(){
    listAtuador = new ArrayList<String>();
    listImg = new ArrayList<Integer>();

    for (Atuadores a : controle.getAtuadores()){
        listAtuador.add(a.getNome());
    }
}

```

```

        if (a.getNome().equals("Tomada")){
            listImg.add(R.drawable.tomada1); // botão de tomada
        }else if (a.getNome().equals("TV")){
            listImg.add(R.drawable.tv1); // botão de TV
        }else if (a.getNome().equals("Ar")){
            listImg.add(R.drawable.ar1); // botão de ar condicionado
        }else{
            listImg.add(R.drawable.botao1); // botão de iluminação
        }
    }
    adapter=new CustomListAdapter(this, listAtuador, listImg);
    list = (ListView) findViewById(R.id.list);
    list.setAdapter(adapter);
    list.setOnItemClickListener(this);
}

@Override
public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
    // TODO Auto-generated method stub
    //item = item.split(" ")[3];

    //Verificar qual a classe que vai ser chamada
    Intent Itela2;
    if (listImg.get(position) == R.drawable.tomada1)
    {
        Itela2 = new Intent (this, Acionamento.class); //Instancia Objeto Itela2 da classe
        Intent passando o nome das classes como parâmetro
    }
    else if (listImg.get(position) == R.drawable.tv1){
        Itela2 = new Intent (this, tv.class); //Instancia Objeto Itela2 da classe Intent
        Intent passando o nome das classes como parâmetro
    }else if (listImg.get(position) == R.drawable.ar1){
        Itela2 = new Intent (this, ar.class); //Instancia Objeto Itela2 da classe Intent
        Intent passando o nome da classe de ar condicionado como parâmetro
    }else{
        Itela2 = new Intent (this, Acionamento.class); //Instancia Objeto Itela2 da classe
        Intent passando o nome das classes como parâmetro
    }
}

```

```

        Bundle bundleParametros = new Bundle(); //Instancia o objeto "bundleParametros" da
        classe Bundle
        bundleParametros.putString("ipRasp", exactUrl.replace("/info","")); // Chave ou key e
        exactURL o valor a ser passado
        bundleParametros.putString("idAtuador", controle.getAtuadores(position).getId());
        bundleParametros.putString("ipAtuador", controle.getAtuadores(position).getIp());
        bundleParametros.putString("nomeAtuador",
        controle.getAtuadores(position).getNome());
        Itela2.putExtras(bundleParametros);
        startActivity(Itela2);

    }

    public void inicializa()
    {
        tv01 = (TextView)findViewById(R.id.tv01);
    }

    public void VerificaConexao()
    {
        final ConnectivityManager connMgr =
        (ConnectivityManager)this.getSystemService(Context.CONNECTIVITY_SERVICE);
        final android.net.NetworkInfo wifi
        =connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
        final android.net.NetworkInfo mobile
        =connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
        if( wifi.isAvailable() && wifi.getDetailedState() ==
        DetailedState.CONNECTED)
        {
            //Toast.makeText(this, "Conectado via Wi-Fi",
            Toast.LENGTH_LONG).show();
            conexao =1;
        }
        else if( mobile.isAvailable() && mobile.getDetailedState() ==
        DetailedState.CONNECTED )
        {
            //Toast.makeText(this, "Conectado via rede móvel ",
            Toast.LENGTH_LONG).show();
            conexao=2;
        }
        else
        {

```

```

        Toast.makeText(this, "Você precisa se conectar a uma rede para usar " ,
Toast.LENGTH_LONG).show();
        finish();
    }
}

private void setupVars()
{
    exactUrl = null;
    iScanTimeout = 50;
    iContInterval =1000;
    bAutoscan = true;
    if (bOnline)
        subnetUrl = getSubnet() + ":8080/info";
} // Fim da setupVars
private void connectUrl()
{
    if (netClient==null)
        return;
    netClient.setTimeout(1000);
    netClient.get(exactUrl,new AsyncHttpResponseHandler() {

        @Override
        public void onSuccess(String response) {
            // result
            bConnected = true;
            tv01.setText("CONECTADO ao dispositivo em\n"+exactUrl);
            resposta=true;
            leitura();
            setProgressBarIndeterminateVisibility(bNetworking);
        }

        @Override
        public void onFailure(Throwable e,String response) {
            tv01.setText("Não encontrado\n "+exactUrl);
            setProgressBarIndeterminateVisibility(bNetworking);
            bAutoscan=true;
            conex();
        }
    });
}

```

```

} // Fim da connectUrl
private void scanNet()
{

    pingUrl = subnetUrl.replace("**",Integer.toString(pingFrom));
    if (netClient==null)
        return;

    netClient.setTimeout(iScanTimeout);
    netClient.get(pingUrl,new AsyncHttpResponseHandler() {

        @Override
        public void onSuccess(String response) {

            exactUrl = pingUrl;
            tv01.setText("Dispositivo encontrado em \n"+exactUrl);
            bAutoscan = false;
            bNetworking = false;

            // btnConnect.setText(R.string.btn_connect);
            setProgressBarIndeterminateVisibility(bNetworking);
            connectUrl();

        }
        @Override
        public void onFailure(Throwable e,String response)
        {
            tv01.setText("Escaneando endereços \n "+pingUrl);
            if (pingFrom<pingTo)
            {
                pingFrom++;
                if (bNetworking)
                    scanNet();
                else {
                    tv01.setText("Pressione para scan \n"+subnetUrl);
                    if (bAutoscan)
                    {
                        //btnConnect.setText(R.string.btn_scan);
                    }
                }
            }
        }
    }
}

```

```

    }
    else //btnConnect.setText(R.string.btn_connect);
    setProgressBarIndeterminateVisibility(bNetworking);
    }
}
});
} //Fim da scanNet

private boolean isNetworkPresent()
{
    ConnectivityManager cm = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    //NetworkInfo activeWifi = cm.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
    NetworkInfo activeWifi = cm.getActiveNetworkInfo();
    return ((activeWifi!=null)&&(activeWifi.isConnected()));
}

public String getSubnet() {
    String subnet = null;
    try {
        for (Enumeration<NetworkInterface> en = NetworkInterface.getNetworkInterfaces();
en.hasMoreElements();) {
            NetworkInterface intf = (NetworkInterface) en.nextElement();
            for (Enumeration<InetAddress> enumIpAddr = intf.getInetAddresses();
enumIpAddr.hasMoreElements();) {
                InetAddress inetAddress = (InetAddress) enumIpAddr.nextElement();
                if (!inetAddress.isLoopbackAddress()) {
                    subnet = inetAddress.getHostAddress().toString();
                    if (InetAddressUtils.isIPv4Address(subnet)) {
                        int idx1 = subnet.lastIndexOf(".");
                        subnet = "http://" + subnet.substring(0,idx1) + ".**";
                        return subnet;
                    }
                }
            }
        }
    }
    catch (SocketException e) {
    }
    return null;
}

```

```

    }
private void conex ()
{
    if ( (!bNetworking) &(bAutoscan) )
    {
        bNetworking = true;
        pingFrom = 1;
        pingTo = 254;

        setProgressBarIndeterminateVisibility(bNetworking);
        // btnConnect.setText(R.string.btn_stop);
        scanNet();
    }
}
private void leitura()
{
    //String[] nome = null;          ;
    if (netClient==null)
        return;

    netClient.setTimeout(1000);
    netClient.get(exactUrl,new AsyncHttpResponderHandler() {

        @Override
        public void onSuccess(String response)
        {
            Atuadores atuador;
            // exemplo : {"res":"OK","temp":"25.0","temp2":"45.4"}

            try {
                String id =null;
                String ip = null;
                String nome = null;

                JSONArray arr = new JSONArray(response);
                int tamanho = arr.length();
                for (int i=0; i<arr.length(); i++) //loop por objeto
                {JSONObject jsonProductObject = arr.getJSONObject(i);
                    id = jsonProductObject.getString("id");
                    ip = jsonProductObject.getString("ip");

```

```

        nome = jsonProductObject.getString("nome");
        atuador = new Atuadores();

        atuador.setId(id);
        atuador.setIp(ip);
        atuador.setNome(nome);
        controle.add(atuador);
        imprimirAtuador();

        tv01.setText("Qtd de Atua:" + tamanho + "ip:" + ip + "id:" + id);
    }
}
catch (Throwable e)
{
    tv01.setText("Erro no Json");
}
finally
{
}
}

@Override
public void onFailure(Throwable e, String response) {
}
});
}

public void ConsultaAtuadores()
{
    for (int i = 0; i < controle.getAtuadores().size(); i++) {
        controle.getAtuadores(i);
        String id = controle.getAtuadores(i).getId();
        String ip = controle.getAtuadores(i).getIp();
        tv01.setText("Qtd de Atua:" + controle.getAtuadores().size() + "ip:" + ip + "id:" + id);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    Atuaadores:"

```

```

        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
} //Fim da classe main

```

Classe TV

```

package com.aml.smarthouse;

import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class tv extends Activity
{
    TextView tv08;
    Button btn_power, btn_down_canal, btn_up_canal, btn_up_volume, btn_down_volume,
    btn_source, btn_mute;
    String ipRasp;
    String ipAtuador;
    String idAtuador;

```

```

    String nomeAtuador;
    AsyncHttpClient netClient = null;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tv);
        getViews(); //Inicia botões, textviews...
        Intent iDados = getIntent();
        if (iDados != null)
        {
            Bundle parametros = iDados.getExtras();
            if (parametros != null)
            {
                ipRasp= parametros.getString("ipRasp");
                ipAtuador= parametros.getString("ipAtuador");
                idAtuador= parametros.getString("idAtuador");
                nomeAtuador= parametros.getString("nomeAtuador");
            }
        }

        netClient = new AsyncHttpClient();
        // tv07.setText(dados);
        /*
        h = new Handler(); //timer
        delay = 1000; //milliseconds
        h.postDelayed(new Runnable(){
            public void run(){
                leitura();
                h.postDelayed(this, delay);
            }
        }, delay); //timer

        btn_mute .setOnClickListener(new OnClickListener() { //mudo
            @Override
            public void onClick(View v)
            {

```

```

        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        //http://localhost/get?atuador=12&acao=123
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=14",new
AsyncHttpResponderHandler()
        {
            }
        );
    }
});

//ligar tv
btn_power .setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v)
    {
        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        //http://localhost/get?atuador=12&acao=123
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=02",new
AsyncHttpResponderHandler()
        {
            }
        );
    }
});

//diminuir canal
btn_down_canal.setOnClickListener(new OnClickListener() {

        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        http://localhost/get?atuador=12&acao=123

```

```

        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=10",new
AsyncHttpResponseHandler()
        {
        }
        );
    }
});

//aumentar canal
btn_up_canal.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v)
    {
        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        http://localhost/get?atuador=12&acao=123
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=9",new
AsyncHttpResponseHandler()
        {
        }
        );
    }
});
//input
btn_source.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v)
    {
        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        http://localhost/get?atuador=12&acao=123
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=13",new
AsyncHttpResponseHandler()
        {
        }
        );
    }
});

```

```

btn_up_volume.setOnClickListener(new OnClickListener() { // aumentar volume da tv
    @Override
    public void onClick(View v)
    {
        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        http://localhost/get?atuador=12&acao=123 esta ação está em ServidorHTTP.JAVA
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=11",new
AsynchHttpResponseHandler()
        {
        }
        );
    }
});

```

```

btn_down_volume.setOnClickListener(new OnClickListener() { // diminuir volume da tv
    @Override
    public void onClick(View v)
    {
        if (netClient==null)
            return;
        netClient.setTimeout(2000);
        //http://localhost/get?atuador=12&acao=123 esta ação está em ServidorHTTP.JAVA
        netClient.post(ipRasp+"/get?atuador="+idAtuador+"&acao=12",new
AsynchHttpResponseHandler()
        {
        }
        );
    }
});
} //fim da onCreate

```

```

public void getViews()
{
    tv08 = (TextView)findViewById(R.id.temperaturaText);
    btn_source = (Button)findViewById(R.id.btn_source);
    btn_up_canal = (Button)findViewById(R.id.btn_up_canal);
    btn_power = (Button)findViewById(R.id.btn_power);
    btn_down_canal = (Button)findViewById(R.id.btn_down_canal);
}

```

```

        btn_up_volume = (Button)findViewById(R.id.btn_up_volume);
        btn_down_volume = (Button)findViewById(R.id.btn_down_volume);
        btn_mute = (Button)findViewById(R.id.btn_mute);
    }
}

```

CÓDIGO-FONTE DO *MIDDLEWARE*

Classe Atuador

```

package com.bean;

public class Atuador {
    private int id;
    private String mac;
    private String nome;
    private String ip;
    private int status;

    public Atuador(){
        setId(0);
        setMac("");
        setNome("");
        setStatus(0);
    }

    public Atuador(int id, String mac, String nome, int status)
    {
        setId(id);
        setMac(mac);
        setNome(nome);
        setStatus(status);
    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getMac() {
        return mac;
    }
    public void setMac(String mac) {
        this.mac = mac;
    }
    public int getStatus() {
        return status;
    }
    public void setStatus(int status) {
        this.status = status;
    }

    public String getNome() {

```

```

        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getIp() {
        return ip;
    }

    public void setIp(String ip) {
        this.ip = ip;
    }
}

```

Classe Controller

```

package com.bean;

import java.util.ArrayList;
import java.util.Date;

public class Controller {
    private int id;
    private String nome;
    private Date dataAcesso;
    private String tipo;
    private ArrayList<Atuador> atuadores;

    public Controller(){
        id = 0;
        nome = "";
        dataAcesso = new Date(System.currentTimeMillis());
        tipo = "";
        atuadores = new ArrayList<Atuador>();
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}

```

```
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public Date getDataAcesso() {
    return dataAcesso;
}
public void setDataAcesso(Date data) {
    this.dataAcesso = data;
}
public String getTipo() {
    return tipo;
}
public void setTipo(String tipo) {
    this.tipo = tipo;
}
public ArrayList<Atuador> getAtuadores() {
    return atuadores;
}

public Atuador getAtuador(int pos) {
    return atuadores.get(pos);
}

public void setAtuadores(ArrayList<Atuador> atuadores) {
    this.atuadores = atuadores;
}

public void addAtuadores(Atuador atuador) {
    this.atuadores.add(atuador);
}

public Atuador buscaAtuadores(Atuador a){

    for (int i = 0; i < this.getAtuadores().size(); i++) {
        if (a.getId() == this.getAtuador(i).getId()){
            a.setIp(this.getAtuador(i).getIp());
            this.getAtuador(i).setStatus(a.getStatus());
        }
    }
}
```

```

        return a;
    }
}
a.setStatus(123);
return a;
}
}

```

Classe JSONParser

```
package com.raspberry;
```

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

```

```
@SuppressWarnings("deprecation")
```

```
public class JSONParser
```

```
{
```

```
    static InputStream is = null;
```

```
    static JSONObject jsonObj = null;
```

```
    static String json = "";
```

```
    public JSONObject getJSONFromUrl(String url)
```

```
{
```

```
    // Requisição HTTP
```

```
    try {
```

```
        // defaultHttpClient
```

```
        DefaultHttpClient httpClient = new DefaultHttpClient(); // instancia o objeto httpClient para
receber a URL do atuador ou raspbe
```

```
        HttpPost httpPost = new HttpPost(url);
```

```

        HttpResponse httpResponse = httpClient.execute(httpPost);
        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

try {
    BufferedReader reader = new BufferedReader(new InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
        System.out.println(line); //imprime todo o Objeto Json da url
    }
    is.close();
    json = sb.toString();
} catch (Exception e) {
}

// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
}
catch (JSONException e) {

    System.out.println("error on parse data in jsonparser.java");
}

// retorno da String do JSON
return jsonObj;
}
}

```

Classe ListaWebServerHttpAtuador

```

package com.raspberry;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import org.json.JSONException;
import org.json.JSONObject;
import com.bean.Atuador;

public class ListaWebServerHttpAtuador
{
    static int contador =0;

    static public Atuador ping (final String address, int tempo) throws IOException
    {

        Atuador atuador = new Atuador();
        try {

            final URL url = new URL("http://" + address+"/T");
            final HttpURLConnection urlConn = (HttpURLConnection)
            url.openConnection();
            urlConn.setConnectTimeout(tempo); // Tempo de resposta em ms
            //enviado por parâmetro
            urlConn.connect();

            if (urlConn.getResponseCode() == HttpURLConnection.HTTP_OK)
            {
                System.out.println("Equipamento encontrado em: " + address);
                //System.out.println("Ping em "+address +" Sucesso!");
                contador++;
                atuador = leituraAtuador("http://"+address+"/T"); //leitura
                //é a função dessa classe que chama a classe do
                //jsonparse
                atuador.setIp(address);

                return atuador;

                //fazer função para testar por json se é o atuador e guardar o
                //nome, ip, cômodo, qtd de equipamentos de cada host
                encontrado(atuadores);
            }
        }
    }
}

```

```

    }

    else
    {
        System.out.println("Falhou: " + address);
        return null;
    }
}

catch (Exception e)
{
    //System.err.println("Ping FALHOU: " + address + " - " +
    e);
    return null;
}

} // fim da função PING

public static Atuator leituraAtuator (String url) throws JSONException
{
    String conteudo;
    JSONObject object;
    JSONParser jsonParser;
    Atuator atuator = new Atuator();

    try
    {
        jsonParser = new JSONParser();
        System.out.println(url);
        System.out.println("passo 01");
        object = jsonParser.getJSONFromUrl (url);
        System.out.println("passo 02");
        conteudo = object.getString("id");
        System.out.println("passo 03");
        atuator.setId(object.getInt("id"));
        atuator.setNome(object.getString("nome"));

        atuator.setStatus(Integer.parseInt(object.getString("lamp")));
    }
}

```

```

        System.out.println("Json encontrado, valor de parametro chave: "+
        conteudo);
        return atuator;
    }
    catch(Exception e)
    {
        System.out.println("Json não encontrado");
        return null;
    }
} // fim da função leitura.

```

```
} // fim da classe
```

Classe PostHttp

```

package com.raspberry;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;

@SuppressWarnings("deprecation")
public class PostHttp
{

    private DefaultHttpClient httpClient;

    public void postt(String url2)
    {

```

```

        try {
            /*
                final URL url = new URL(url2);
                final HttpURLConnection urlConn = (HttpURLConnection)
url.openConnection());
                urlConn.setConnectTimeout(1000); // Tempo de resposta em ms enviado por
parâmetro
                urlConn.connect();

                if (urlConn.getResponseCode() == HttpURLConnection.HTTP_OK)
                {
                    System.out.println(" post ok : ");
                }

                else
                {
                    System.out.println("Falhou post : ");
                }
            }
        */

        URL url = new URL(url2);
        URLConnection yc = url.openConnection();
        BufferedReader in = new BufferedReader(new InputStreamReader(yc.getInputStream()));
        String inputLine;

        while ((inputLine = in.readLine()) != null)
            System.out.println(inputLine);

        in.close();
    }

    catch (Exception e)
    {
        //System.err.println("Ping FALHOU: " + address + " - " + e);
    }
}

```

```

        /*
        try {

                httpClient = new DefaultHttpClient();
                HttpPost httpPost = new HttpPost(url);

                HttpResponse httpResponse = httpClient.execute(httpPost);
                HttpEntity httpEntity = httpResponse.getEntity();
        } catch (Exception e)
        {
                System.out.println("Falha no Post");
                //System.err.println("Ping FALHOU: " + address + " - " + e);

        }

        /*try {
        // defaultHttpClient

        httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(url);

        HttpResponse httpResponse = httpClient.execute(httpPost);
        HttpEntity httpEntity = httpResponse.getEntity();

        } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
        } catch (ClientProtocolException e) {
                e.printStackTrace();
        } catch (IOException e) {
                e.printStackTrace();
        }
        */
    }
}

```

Classe ConnectionMySQL

```
package com.DAO.Conecction;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConnectionMySQL {

    private static final String URL = "jdbc:mysql://localhost/ami_hand";
    private static final String USER = "root";
    private static final String SENHA = "123";

    public static Connection obterConexao() throws SQLException{

        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return (Connection) DriverManager.getConnection(URL, USER, SENHA);
    }
}
```

Classe ServidorHttp

```
package com.raspberry;

import java.io.IOException;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.URI;
import java.net.UnknownHostException;
import java.util.HashMap;
import java.util.Map;
import com.bean.Atuador;
import com.bean.Controller;
```

```

import com.context.ConfigurationHandHandler;
import com.sun.net.httpserver.HttpExchange;
import com.sun.net.httpserver.HttpHandler;
import com.sun.net.httpserver.HttpServer;

public class ServidorHttp
{
    static Controller sensor;
    static int contador=0;
    static int temp = 20;

    public static void main(String[] args) throws Exception
    {
        sensor = new Controller();
        encontra();
        ConfigurationHandHandler configuration = new ConfigurationHandHandler(sensor);
        HttpServer server = HttpServer.create(new InetSocketAddress(8080), 0);
        server.createContext("/info", new InfoHandler());
        server.createContext("/get", new GetHandler());
        server.createContext("/configurationHand", configuration);
        server.setExecutor(null); // creates a default executor
        server.start();
        System.out.println("O servidor foi iniciado");
    } // Fim do main

    // http://localhost:8000/info
    static class InfoHandler implements HttpHandler
    {
        public void handle(HttpExchange httpExchange) throws IOException
        {
            //Buscar os atuadores
            //encontra();

            //imprimir json com todos os atuadores dinamicamente com o ip, tipo, estado e o id a partir da
            classe Sensor
            String response = consultaAtuadores();

            ServidorHttp.writeResponse(httpExchange, response.toString());
            System.out.println("Válido info");
            return;
        }
    }

```

```

}

static class GetHandler implements HttpHandler
{
    public void handle(HttpExchange httpExchange) throws IOException
    {
        //String response = "Olá, o Servidor está online em http://localhost/get?atuador=12&acao=123";
        String response = "[{status:ok}]";
        String acaoAtuador = "";
        //Obter os Parametros
        URI requestedUri = httpExchange.getRequestURI();
        String query = requestedUri.getRawQuery();
        Map<String, String> parameters = ServidorHttp.queryToMap(query);

        System.out.println("Obter atuador:" + parameters.get("atuador") + " Ação: "
+parameters.get("acao"));

        Atuador atuadorResq = new Atuador(Integer.parseInt(parameters.get("atuador")), "", "",
Integer.parseInt(parameters.get("acao")));
        PostHttp obj = new PostHttp();

        atuadorResq = sensor.buscaAtuadores(atuadorResq);

        //Ligar LED
        if (atuadorResq.getStatus() == 0)
            acaoAtuador = "http://" + atuadorResq.getIp() + "/L"; // Ligar LED
        else if(atuadorResq.getStatus() == 1)
            acaoAtuador = "http://" + atuadorResq.getIp() + "/H"; // Desligar LED

        //Ligar Ar
        else if(atuadorResq.getStatus() == 4)
            acaoAtuador = "http://" + atuadorResq.getIp() + "/H"; // Ligar ar
        else if(atuadorResq.getStatus() == 5)
            acaoAtuador = "http://" + atuadorResq.getIp() + "/L"; // Desligar ar
        //Mudar Temperatura do Ar
        else if(atuadorResq.getStatus() == 6){
            temp++;
            if (temp > 21)
                temp = 21;
        }
    }
}

```

```

        acaoAtuador = "http://" + atuadorResq.getIp() + "/" + String.valueOf(temp); //Aumenta
Temperatura
    }else if(atuadorResq.getStatus() == 7){
        temp--;
        if (temp < 18)
            temp = 18;
        acaoAtuador = "http://" + atuadorResq.getIp() + "/" + String.valueOf(temp); //Diminui
Temperatura
    }

    else if(atuadorResq.getStatus() == 15)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/m"; // Louver

//Ligar TV
    else if(atuadorResq.getStatus() == 2)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/P"; // Ligar ou Desligar TV
    else if(atuadorResq.getStatus() == 3)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/R"; // Desligar TV - Verificar se é
necessário
    else if(atuadorResq.getStatus() == 13)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/i"; // Source - Verificar se é necessário
//Mudar Canal da TV
    else if(atuadorResq.getStatus() == 9)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/c"; //Mudar Canal TV
    else if(atuadorResq.getStatus() == 10)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/cm"; // Diminuir Canal TV
//Aumentar Volume da TV
    else if(atuadorResq.getStatus() == 11)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/v"; // Aumenta Volume da TV
    else if(atuadorResq.getStatus() == 12)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/vm"; // Diminui Volume da TV
    else if(atuadorResq.getStatus() == 14)
        acaoAtuador = "http://" + atuadorResq.getIp() + "/mu"; // Mute da TV
    obj.postt(acaoAtuador);

    ServidorHttp.writeResponse(httpExchange, response);
    System.out.println("Válido get");
}
}

```

```
public static void writeResponse(HttpExchange httpExchange, String response) throws
IOException
```

```
{
    httpExchange.sendResponseHeaders(200, response.length());
    OutputStream os = httpExchange.getResponseBody();
    os.write(response.getBytes());
    os.close();
}
```

```
public static Map<String, String> queryToMap(String query){
    Map<String, String> result = new HashMap<String, String>();
    if (query != null){
        for (String param : query.split("&")) {
            String pair[] = param.split("=");
            if (pair.length>1) {
                result.put(pair[0], pair[1]);
            }else{
                result.put(pair[0], "");
            }
        }
    }
    return result;
}
```

```
public static void encontra() throws UnknownHostException
```

```
{
    System.out.println("IP/Localhost: " + InetAddress.getLocalHost().getHostAddress());
    //String ip = InetAddress.getLocalHost().getHostAddress();
    String ip = "192.168.0.1";

    String oct[] = ip.split("\\.");
    String hostAtuadores;
    int tempo;
    tempo = 50;

    for (int i = 0; i < 255; i++)
    {
        hostAtuadores = oct[0] + "." + oct[1] + "." + oct[2] + ".";
        hostAtuadores = hostAtuadores + String.valueOf(i);
    }
}
```

```

        Atuador atuador;
        try {
            if(Integer.parseInt(oct[3]) != i)
            {
                atuador = ListaWebServerHttpAtuador.ping(hostAtuadores,
tempo);

                if (atuador != null){

                    //Inserir no Arraylist de atuadores
                    sensor.addAtuadores(atuador);

                    System.out.println("Equipamento encontrado em: " +
hostAtuadores);

                    // obj.leitura(hostAtuadores);
                    contador++;

//fazer função para testar por json se é o atuador e guardar o nome, ip, cômodo, qtd de
equipamentos de cada host encontrado(atuadores);
                }else
                {
                    System.out.println("Falhou: " + hostAtuadores);
                }
            }
        }catch (Exception e)
        {
            System.err.println("Ping FALHOU: " + hostAtuadores + " - " +
e);
        }
    }

    System.out.println("O Número de web servers é: " + contador);
    // if(contador==0)
    //encontra();
}

public static String consultaAtuadores(){
    String retorno = "";

    retorno = "[";
    for (int i = 0; i < sensor.getAtuadores().size(); i++) {

```

```

        sensor.getAtuador(i);

        if(i > 0)
        {
            retorno = retorno + ",";
        }

        retorno = retorno + "{";
        retorno = retorno + "\"id\": \""+sensor.getAtuador(i).getId()+"\", ";
        retorno = retorno + "\"ip\": \""+sensor.getAtuador(i).getIp()+"\", ";
        retorno = retorno + "\"mac\": \""+sensor.getAtuador(i).getMac()+"\", ";
        retorno = retorno + "\"nome\": \""+ sensor.getAtuador(i).getNome() + "\", ";
        retorno = retorno + "\"status\": \""+ sensor.getAtuador(i).getStatus() + "\" ";
        retorno = retorno + "}\n ";
    }
    retorno = retorno + "]";
    return retorno;
}
}

```

CÓDIGO-FONTE DOS ATUADORES

Ar condicionado

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <IRremoteESP8266.h>

IRsend irsend(2);

const char* ssid    = "AML";
const char* password = "raspberry";
ESP8266WebServer server(80);
String webString="";

void handle_root()
{
    server.send(200, "text/plain", "Hello from the esp8266, read from /T or /H and /L ");
}

```

```

    delay(100);
}

void setup(void)
{
  irsend.begin();
  Serial.begin(115200);
  pinMode(2, OUTPUT);

  // Conexão com a Wi-Fi
  WiFi.begin(ssid, password);
  Serial.print("\n\r\n\rWorking to connect");

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  server.on("/", handle_root);

  server.on("/T", []()
  { // Informações do Atuador

    webString="{\"lamp\":\\"00\", \"id\":\\"03\", \"nome\": \"Ar\"}";
    server.send(200, "text/plain", webString);
  });

  server.on("/H", []()
  {
    unsigned int irSignal1[] = {8850,4500, 500,600, 600,500, 550,1650, 550,600, 600,1600, 650,500,
550,550, 550,600, 600,500, 550,1700, 550,550, 600,500, 600,550, 550,550, 550,1700, 600,1600,
550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 600,500, 650,500, 550,550, 550,600,
550,550, 550,600, 550,550, 550,600, 550,600, 550,550, 550,550, 550,600, 550,550, 550,600, 550,550,
550,1700, 550,600, 550,1650, 550,1650, 550,600, 550,550, 550};

    webString="GPIO2 esta enviado";

```

```

int khz = 38; // 38kHz frecuencia usada no protocolo NEC
irsend.sendRaw(irSignal1, sizeof(irSignal1) / sizeof(irSignal1[0]), khz);
delay(40);
server.send(200, "text/plain", webString);
});

server.on("/L", []()
{
    unsigned int irSignal[] = {8850,4450, 550,550, 600,550, 550,1650, 550,600, 550,550, 550,600,
550,550, 550,600, 550,550, 550,1700, 550,550, 550,600, 550,550, 550,600, 550,1650, 550,1650,
550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,600, 550,550, 550,550, 600,550,
550,550, 550,600, 550,550, 550,600, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550,
600,1650, 550,600, 550,550, 550,1650, 550,600, 550,550, 550};
    webString="GPIO2 esta enviado";
    int khz = 38; // 38kHz frecuencia usada no protocolo NEC
    irsend.sendRaw(irSignal, sizeof(irSignal) / sizeof(irSignal[0]), khz);
    delay(40);
    server.send(200, "text/plain", webString);
});

server.on("/m", []()
{
    unsigned int irSignal2[] = {8900,4450, 550,550, 650,500, 550,1650, 550,550, 550,1700, 550,550,
650,500, 550,1650, 550,600, 550,1650, 550,550, 550,600, 550,1650, 550,600, 550,1650, 550,600,
550,550, 550,600, 600,500, 550,600, 550,550, 550,600, 550,550, 600,550, 600,500, 550,600, 550,550,
550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 600,550, 550,550, 550,600, 550,550, 550,600,
550,1650, 550,600, 550,600, 550,1650, 550,550, 550,550, 600}; webString="GPIO2 esta enviado";
    int khz = 38; // 38kHz frecuencia usada no protocolo NEC
    irsend.sendRaw(irSignal2, sizeof(irSignal2) / sizeof(irSignal2[0]), khz);
    delay(40);
    server.send(200, "text/plain", webString);
});

server.on("/18", []()
{
    unsigned int irSignal10[] = {8900,4450, 600,500, 600,500, 550,1700, 550,550, 550,1650, 550,600,
550,550, 650,500, 550,600, 550,1650, 550,550, 550,600, 550,550, 550,600, 600,1600, 550,1700,
550,550, 550,600, 550,550, 550,600, 550,550, 550,550, 550,600, 550,600, 550,550, 550,600, 550,550,
550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600,
550,1650, 550,600, 550,1650, 550,1650, 600,550, 550,550, 550};
    webString="GPIO2 esta enviado";
    int khz = 38; // 38kHz frecuencia usada no protocolo NEC
    irsend.sendRaw(irSignal10, sizeof(irSignal10) / sizeof(irSignal10[0]), khz);
    delay(40);

```

```

server.send(200, "text/plain", webString);
});

server.on("/19", []()
{
    unsigned int irSignal4[] = {8900,4450, 500,600, 550,550, 550,1650, 600,550, 550,1650, 550,600,
550,550, 550,600, 550,550, 550,1700, 550,550, 550,600, 550,550, 550,1650, 550,600, 550,600,
550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550,
550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600,
550,1650, 550,600, 550,1650, 550,600, 550,1650, 550,1650, 550};
    webString="GPIO2 esta enviado";
    int khz = 38; // 38kHz frecuencia usada no protocolo NEC
    irsend.sendRaw(irSignal4, sizeof(irSignal4) / sizeof(irSignal4[0]), khz);
    delay(40);
    server.send(200, "text/plain", webString);
});

server.on("/20", []()
{
    unsigned int irSignal5[] = {8900,4450, 500,600, 550,550, 550,1700, 550,550, 550,1650, 550,600,
550,550, 550,600, 550,600, 550,1650, 550,550, 550,600, 550,550, 550,1700, 550,550, 550,1700,
550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550,
550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600, 550,550, 550,600,
600,1600, 550,600, 550,1650, 550,600, 550,1650, 550,550, 550};
    webString="GPIO2 esta enviado";
    int khz = 38; // 38kHz frecuencia usada no protocolo NEC
    irsend.sendRaw(irSignal5, sizeof(irSignal5) / sizeof(irSignal5[0]), khz);
    delay(40);
    server.send(200, "text/plain", webString);
});

server.on("/21", []()
{
    webString="GPIO2 esta enviado";
    irsend.sendNEC(0x28460000, 32);
    delay(40);
    server.send(200, "text/plain", webString);
});

server.begin();
Serial.println("HTTP server started");
}

```

```
void loop(void)
{
  server.handleClient();
}
```

Televisão

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <IRremoteESP8266.h>

IRsend irsend(2);

const char* ssid = "AML";
const char* password = "raspberry";

ESP8266WebServer server(80);

String webString="";
void handle_root()
{
  server.send(200, "text/plain", "Hello from the esp8266, read from /T or /H and /L ");
  delay(100);
}

void setup(void)
{
  irsend.begin();
  Serial.begin(115200);
  pinMode(2, OUTPUT);

  // Conexão com Wi-Fi
  WiFi.begin(ssid, password);
  Serial.print("\n\r\n\rWorking to connect");

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```

    Serial.print(".");
}
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

server.on("/", handle_root);

server.on("/T", []()
{
    Informações do Atuador

    webString="{\"lamp\": \"00\", \"id\": \"02\", \"nome\": \"TV\"}"; // Microcontroler has a hard time with float
    to string
    server.send(200, "text/plain", webString); // send to someones browser when asked
});

server.on("/H", []()
{
    webString="GPIO2 esta ligada";
    digitalWrite(2, LOW);
    server.send(200, "text/plain", webString); // send to someones browser when asked
});

server.on("/L", []()
{
    webString="GPIO2 esta desligada";
    digitalWrite(2, HIGH);
    server.send(200, "text/plain", webString); // send to someones browser when asked
});

server.on("/P", []()
{
    webString="GPIO2 esta enviado";

    for(int i=0; i<3;i++)
    {
        irsend.sendSony(0xa90, 12);
        delay(40);
    }
}

```

```

}
server.send(200, "text/plain", webString);      // send to someones browser when asked
});

server.on("/i", []()
{
  webString="GPIO2 esta enviado";
  Serial.println("Sony");
  for(int i=0; i<3;i++)
  {
    irsend.sendSony(0xa50, 12);
    delay(40);
  }
  server.send(200, "text/plain", webString);    // send to someones browser when asked
});

server.on("/v", []()
{
  webString="GPIO2 esta enviado";
  Serial.println("Sony");
  for(int i=0; i<3;i++)
  {
    irsend.sendSony(0x490, 12);
    delay(40);
  }
  server.send(200, "text/plain", webString);    // send to someones browser when asked
});

server.on("/vm", []()
{
  webString="GPIO2 esta enviado";
  Serial.println("Sony");
  for(int i=0; i<3;i++)
  {
    irsend.sendSony(0xC90, 12);
    delay(40);
  }
  server.send(200, "text/plain", webString);    // send to someones browser when asked
});

```

```
server.on("/vm", []()  
{  
  webString="GPIO2 esta enviado";  
  Serial.println("Sony");  
  for(int i=0; i<3;i++)  
  {  
    irsend.sendSony(0xC90, 12);  
    delay(40);  
  }  
  server.send(200, "text/plain", webString);      // send to someones browser when asked  
});
```

```
server.on("/c", []()  
{  
  webString="GPIO2 esta enviado";  
  Serial.println("Sony");  
  for(int i=0; i<3;i++)  
  {  
    irsend.sendSony(0x90, 12);  
    delay(40);  
  }  
  server.send(200, "text/plain", webString);      // send to someones browser when asked  
});
```

```
server.on("/cm", []()  
{  
  webString="GPIO2 esta enviado";  
  Serial.println("Sony");  
  for(int i=0; i<3;i++)  
  {  
    irsend.sendSony(0x890, 12);  
    delay(40);  
  }  
  server.send(200, "text/plain", webString);      // send to someones browser when asked  
});
```

```
server.on("/mu", []()  
{  
  webString="GPIO2 esta enviado";
```

```
Serial.println("Sony");
for(int i=0; i<3;i++)
{
  irsend.sendSony(0x290, 12);
  delay(40);
}
server.send(200, "text/plain", webString);      // send to someones browser when asked
});
server.begin();
Serial.println("HTTP server started");
}

void loop(void)
{
  server.handleClient();
}
```