

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
ESPECIALIZAÇÃO EM TECNOLOGIA JAVA E DESENVOLVIMENTO PARA
DISPOSITIVOS MÓVEIS

SÉRGIO MORIBE

**AUTOMAÇÃO DE SALA DE *HOME THEATER* UTILIZANDO
DISPOSITIVOS MÓVEIS BASEADOS EM *ANDROID***

MONOGRAFIA DE ESPECIALIZAÇÃO

**CURITIBA
2013**

SÉRGIO MORIBE

**AUTOMAÇÃO DE SALA DE *HOME THEATER* UTILIZANDO
DISPOSITIVOS MÓVEIS BASEADOS EM *ANDROID***

Monografia de Especialização apresentada ao Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Especialista em Tecnologia Java e Desenvolvimento para Dispositivos Móveis”.

Orientador: Paulo Maurício de Lucchi
Bordin

**CURITIBA
2013**

À Ivone, Daniele e Erick.

AGRADECIMENTOS

A todos os professores do curso pelos conhecimentos transmitidos.

Ao professor orientador Paulo Maurício de Lucchi Bordin, por sua atenção e apoio neste trabalho.

O primeiro e indispensável passo para obter as coisas que você deseja da vida é decidir o que você quer.

Ben Stein

RESUMO

MORIBE, Sérgio. Automação de Sala de *Home Theater* utilizando dispositivos móveis baseados em *Android*. 2013. 53 f. Monografia de Especialização – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

A utilização de uma simples sala de *Home Theater* pode ser muito complicada devido a enorme quantidade de equipamentos, controles remotos, interruptores, siglas e botões. Com o desenvolvimento da tecnologia, mais e mais busca-se soluções que nos tragam conforto, principalmente quando o assunto é automação residencial. A tendência é concentrar todos os controles num único controle remoto universal, porém muitas vezes, estes apresentam grande quantidade de botões e funções pouco intuitivas e que poucos são capazes de utilizar eficientemente. O protótipo desenvolvido neste trabalho visa utilizar os dispositivos móveis de uso pessoal como os *smartphones* e *tablets* que já estão atualmente sempre à mão das pessoas, como um controle remoto universal, porém, com uma interface simples e intuitiva, além de acrescentar outras funcionalidades como o controle da iluminação ambiente. Para isto, foi desenvolvido um aplicativo para plataforma Android e um protótipo eletrônico utilizando um kit de desenvolvimento com ARM Cortex M3 Stellaris EKS-LM3S6965 da *Texas Instruments*. O aplicativo faz a interface do controle e comunica via Wi-Fi com uma placa de controle *Ethernet* que envia sinais infra-vermelhos (IR) para os equipamentos a serem controlados. Os objetivos foram alcançados considerando que o protótipo foi concluído com todos os testes funcionais realizados e seu funcionamento comprovado.

Palavras-chave: Automação residencial. Home Theater. Dispositivos móveis. Controle remoto.

ABSTRACT

MORIBE, Sérgio. Automation of Home Theater's Room Using Mobile Devices Based on Android. 53 f. Monografia de Especialização – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

The use of a single room home theater can be very complicated due to the huge amount of equipment, remote controls, switches, buttons and acronyms. With the development of technology, more and more we seek solutions bring comfort, especially when the issue is home automation. The tendency is to concentrate all the controls in a single universal remote control, however often they have lots of buttons and functions unintuitive and few are able to efficiently use. The prototype developed in this study aims to use mobile devices for personal use such as smartphones and tablets that are already currently handy people, as a universal remote control, however, with a simple and intuitive interface, as well as adding other features such as control of ambient lighting. For this, we developed an application for the Android platform and a prototype electronic using a development kit with ARM Cortex M3 Stellaris EKS-LM3S6965 Texas Instruments. The application acts as an interface and communicates control via Wi-Fi with a board Ethernet control sends infrared signals (IR) for the equipment to be controlled. Objectives were achieved considering that the prototype was completed with all functional tests performed and demonstrated its operation.

Keywords: Home automation. Home Theater. Mobile devices. Remote control.

LISTA DE FIGURAS

FIGURA 1	– Controles Remotos	14
FIGURA 2	– Transmissor e Receptor IR	19
FIGURA 3	– Modulação Sony SIRC	20
FIGURA 4	– Protocolo Sony SIRC	20
FIGURA 5	– Modulação RCA	21
FIGURA 6	– Protocolo RCA	21
FIGURA 7	– Modulação RC5	22
FIGURA 8	– Protocolo RC5	22
FIGURA 9	– Kit de Desenvolvimento LM3S6965	24
FIGURA 10	– Dispositivo Beacon	26
FIGURA 11	– Dispositivo RedEye	27
FIGURA 12	– Dispositivo BlueIR	27
FIGURA 13	– Tela do aplicativo Peel Smart Remote	28
FIGURA 14	– Diagrama de Caso de Uso	32
FIGURA 15	– Diagrama de Sequencia Controlar TV	34
FIGURA 16	– Diagrama de Sequencia Controlar <i>Blu-Ray</i>	35
FIGURA 17	– Diagrama de Sequencia Controlar <i>Home Theater</i>	35
FIGURA 18	– Diagrama de Sequencia Controlar Iluminação	35
FIGURA 19	– Diagrama de Comunicação Controlar TV	36
FIGURA 20	– Diagrama de Comunicação Controlar <i>Blu-Ray</i>	36
FIGURA 21	– Diagrama de Comunicação Controlar <i>Home Theater</i>	37
FIGURA 22	– Diagrama de Comunicação Controlar Iluminação	37
FIGURA 23	– Diagrama de Classes Controlar TV	38
FIGURA 24	– Diagrama de Classes Controlar <i>Home Theater</i>	39
FIGURA 25	– Diagrama de Classes Controlar <i>Blu-Ray</i>	39
FIGURA 26	– Diagrama de Classes Controlar Cenários de Iluminação	40
FIGURA 27	– Ícone do Aplicativo Desenvolvido	41
FIGURA 28	– Tela do Emulador Android SDK para o controle da TV	41
FIGURA 29	– Telas de Controle Remoto TV e <i>Home Theater</i>	42
FIGURA 30	– Tela do Controle Remoto TV orientação paisagem	43
FIGURA 31	– Telas de Controle Remoto <i>Blu-Ray</i> e Iluminação	43
FIGURA 32	– Tela do Controle Remoto <i>Home Theater</i> no <i>tablet</i>	44
FIGURA 33	– Diagrama do Sistema de Controle Remoto IR	44
FIGURA 34	– Circuito <i>driver</i> de saída IR	45
FIGURA 35	– Protótipo Montado	46
FIGURA 36	– Sinal IR do Controle Remoto Samsung	47
FIGURA 37	– Sinal IR Modulado do Controle Remoto Samsung	47

LISTA DE QUADROS

QUADRO 1 – Códigos do Controle Remoto Samsung	48
---	----

LISTA DE SIGLAS

AURESIDE	Associação Brasileira de Automação Residencial
ARM	Advanced RISC Machine
BPF	Band Pass Filter - Filtro passa faixa
CCS	Code Composer Studio
GPS	Global Positioning System - Sistema de Posicionamento Global
HTTP	Hyper Text Transfer Protocol
IR	Infra-Red - Infra-Vermelho
IDE	Integrated Development Environment - Ambiente Integrado de Desenvolvimento
JVM	Java Virtual Machine - Máquina Virtual Java
JTAG	Joint Test Action Group
LED	Light Emitting Diode - Diodo Emissor de Luz
LSB	Least Significant Bit - Bit menos Significativo
MSB	Most Significant Bit - Bit mais Significativo
MVC	Modelo Visão Controle
OHA	Open Handset Alliance
OMG	Object Management Group
POO	Projeto Orientado a Objeto
PWM	Pulse Width Modulation - Modulação por Largura de Pulso
RCA	Radio Corporation of America
RISC	Reduced Instruction Set Computer
SDK	Software Development Kit
SIRC	Serial Infra-Red Control
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
Wi-Fi	Wireless-Fidelity

SUMÁRIO

1 INTRODUÇÃO	12
1.1 MOTIVAÇÃO	13
1.2 OBJETIVOS	14
1.2.1 Objetivo Geral	15
1.2.2 Objetivos Específicos	15
1.3 ESTRUTURA DO TRABALHO	15
2 REVISÃO BIBLIOGRÁFICA	16
2.1 A PLATAFORMA ANDROID	16
2.1.1 Sistema Operacional Linux	17
2.1.2 Máquina Virtual Dalvik	17
2.1.3 Ambiente de Desenvolvimento	17
2.2 TEORIA DO CONTROLE REMOTO IR (INFRA-VERMELHO)	18
2.2.1 O LED IR	18
2.2.2 O Transmissor IR e o Receptor IR	18
2.2.3 Exemplos de protocolos IR	19
2.2.3.1 Protocolo Sony SIRC	20
2.2.3.2 Protocolo RCA	21
2.2.3.3 Protocolo Philips RC5	22
2.3 O KIT DE DESENVOLVIMENTO STELLARIS EKS-LM3S6965	23
2.3.1 Ambiente Integrado de Desenvolvimento	23
2.4 DESENVOLVIMENTO DE SOFTWARE	24
2.4.1 UML	25
2.5 SOLUÇÕES EQUIVALENTES	26
2.5.1 Aplicativos de Controle Remoto específicos para <i>Smart TVs</i>	26
2.5.2 Equipamentos Similares	26
2.5.3 Aplicativos de Controle Remoto específicos para <i>smartphones e tablets</i> com IR embutido no aparelho	28
3 DESENVOLVIMENTO	29
3.1 LEVANTAMENTO DE REQUISITOS	29
3.1.1 Requisitos Funcionais	29
3.1.2 Requisitos Não Funcionais	29
3.1.3 Restrições	30
3.2 MODELAGEM	30
3.2.1 Modelo de Caso de Uso	30
3.2.1.1 Definição dos Atores	30
3.2.1.2 Definição dos Casos de Uso	31
3.2.1.3 Diagrama de Caso de Uso	31
3.2.2 Levantamento de Classes	32
3.2.2.1 Identificando Classes Fronteira	33
3.2.2.2 Identificando Classes Entidade	33
3.2.2.3 Identificando Classes Controle	33

3.2.3 Estudo das Interações entre Objetos	34
3.2.3.1 Diagrama de Sequência	34
3.2.3.2 Diagrama de Comunicação	36
3.2.4 Relacionamento entre Classes e Diagrama de Classes	37
3.3 <i>DESIGN</i> DA INTERFACE	40
3.4 DESENVOLVIMENTO DO APLICATIVO ANDROID	40
3.4.1 Telas do Aplicativo	42
3.5 DESENVOLVIMENTO DO <i>HARDWARE IR</i> E <i>FIRMWARE</i>	43
3.5.1 Circuito de Saída IR	45
3.5.2 Desenvolvimento do <i>Firmware</i>	45
4 RESULTADOS	49
5 CONCLUSÕES	50
5.1 TRABALHOS FUTUROS	51
REFERÊNCIAS	52

1 INTRODUÇÃO

Nos últimos anos, estão ocorrendo mudanças significativas em todos os aspectos da vida humana. O desenvolvimento tecnológico e o advento da internet alteraram para sempre o modo de vida e as relações pessoais (BOLZANI, 2004). A grande evolução da microeletrônica propiciou o desenvolvimento de equipamentos mais ágeis e com custo e tamanho mais reduzidos. O rápido avanço do simples celular para os atuais *smartphone* é um bom exemplo disto.

Com o *smartphone* pode-se dizer, com certeza, que as pessoas estão com um computador completo na palma das mãos, o que vem de encontro com o conceito de dispositivos móveis, ou seja, dispositivos que apresentam as características de portabilidade, usabilidade, funcionalidade e conectividade (LEE et al., 2005).

Nesta época repleta de mudanças, está se presenciando mais uma transformação na vida de todos, o advento dos dispositivos e residências inteligentes (BOLZANI, 2004). As residências inteligentes tem atraído o interesse de muitas pessoas, pois oferece instrumentos para tornar os lares mais confortáveis, seguros, práticos e eficientes.

Um sistema completo para residências inteligentes ou automação residencial é muito amplo, pois engloba todos os sistemas de uma residência tais como, rede de telefonia, rede de informática, rede elétrica, sistema de segurança, sistema de iluminação, sistemas hidráulico e gás, controle predial, climatização e entretenimento (LEE et al., 2005).

Neste trabalho é abordado o desenvolvimento de um protótipo, somente para a Automação de uma Sala de *Home Theater* utilizando dispositivos móveis baseados em *Android* como controle remoto.

Como o foco desta monografia é o desenvolvimento para dispositivos móveis, foi utilizado como placa eletrônica de controle um kit de desenvolvimento baseado em processador ARM Cortex M3 *Stellaris EKS-LM3S6965* da *Texas Instruments*, sendo

desenvolvidos apenas os circuitos de interface conectados a mesma.

Foi desenvolvido o aplicativo para dispositivo móvel baseado em Android com interface intuitiva e fácil de ser utilizada. Este aplicativo acessa através da rede *wireless* do tipo Wi-Fi com roteador conectado a rede doméstica TCP/IP onde está conectado a placa eletrônica de controle que através de sinais IR (Infra-Vermelho) controla os equipamentos da sala de *Home Theater*. A escolha do Sistema Android para o desenvolvimento foi baseado na sua grande aceitação no mercado e, por ser um sistema de código livre, torna-se muito mais acessível e fácil na obtenção das ferramentas e equipamentos necessários para o desenvolvimento.

Os resultados obtidos são satisfatórios visto que todas as funcionalidades do sistema foram testadas. O alcance do IR ficou em torno de 4 metros e o tempo de resposta menor que 1 segundo o que garante sua aplicação como controle remoto de *Home Theaters*.

1.1 MOTIVAÇÃO

O mercado de automação residencial vem crescendo a uma média de 35% ao ano em número de projetos. Segundo José Roberto Murati (Conselheiro da AURESIDE) “O público que hoje está comprando a sua primeira casa ou apartamento, de forma geral, é de pessoas jovens antenadas com as tecnologias e que gostariam de utilizar seus *smartphones* ou *tablets* também na automação residencial” (TECPAR, 2011).

Estudos mostram que hoje em dia mais de 3 bilhões de pessoas possuem um aparelho celular, e isso corresponde a mais ou menos metade da população mundial (LECHETA, 2010).

O público alvo está nas classes sociais AA, A e B composto por pessoas que gostam dos benefícios da tecnologia formando um mercado emergente com grandes oportunidades surgindo.

Atualmente existe uma grande disponibilidade de serviços como banda larga, TV digital, telefonia celular, comércio eletrônico e introdução de novas tecnologias com custos decrescentes.

Uma simples sala de *home theater* pode tornar as coisas muito complicadas devido a enorme quantidade de equipamentos, controles remotos, interruptores, siglas

e botões. A tendência é concentrar todos os controles num único controle remoto universal, porém muitas vezes, estes apresentam grande quantidade de botões e funções pouco intuitivas e que poucos são capazes de utilizar eficientemente. A Figura 1 ilustra a quantidade de controles remotos de uma sala de *Home Theater*.

A grande perspectiva de crescimento do mercado e os benefícios que o trabalho pode trazer, como a possibilidade de se utilizar os *smartphones* e/ou *tablets* como controle remoto universal aliando uma interface intuitiva e fácil de ser utilizada, motivaram o desenvolvimento deste trabalho.



Figura 1: Quantidade de controles remotos numa sala de *Home Theater*

Fonte: Autoria própria

1.2 OBJETIVOS

Aqui serão descritos os objetivos geral e específicos do trabalho.

1.2.1 OBJETIVO GERAL

Desenvolver um protótipo para controlar uma sala de *Home Theater* utilizando *smartphone* ou *tablet* baseados em Android.

1.2.2 OBJETIVOS ESPECÍFICOS

- Modelar o aplicativo a ser desenvolvido.
- Desenvolver aplicativo Android para *smartphone* e *tablet* para controle da sala de *Home Theater*.
- Desenvolver o *firmware* da ferramenta de desenvolvimento baseada em ARM Cortex M3 Stellaris EKS-LM3S6965 para receber os comandos dos dispositivos móveis e enviá-los via infra-vermelho para os equipamentos da sala de *Home Theater*.
- Realizar os testes de funcionamento dos aplicativos e *firmware*.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido em 5 capítulos organizados da seguinte maneira.

O capítulo 1 apresenta uma introdução ao tema do trabalho, a motivação, o objetivo geral e os objetivos específicos, além de sua estrutura.

O capítulo 2 faz um revisão bibliográfica dos conhecimentos prévios necessários ao desenvolvimento do trabalho, tais como, a plataforma Android, a teoria do controle remoto IR, o kit de desenvolvimento *Stellaris* EKS-LM3S6965, o desenvolvimento de *softwares* e as soluções equivalentes existentes no mercado.

O capítulo 3 aborda o desenvolvimento realizado, expondo as fases de levantamento de requisitos, modelagem UML, *design* da interface, desenvolvimento do aplicativo e desenvolvimento do *hardware IR* e *firmware*.

O capítulo 4 explana sobre os resultados obtidos com desenvolvimento do protótipo.

O capítulo 5 trata das conclusões e desenvolvimentos futuros.

2 REVISÃO BIBLIOGRÁFICA

Para o desenvolvimento do trabalho, foi necessária a realização de uma pesquisa para ampliação do conhecimento e definição do escopo do mesmo.

2.1 A PLATAFORMA ANDROID

O Android consiste numa plataforma de desenvolvimento para aplicativos móveis, baseada numa versão modificada do sistema operacional Linux, com diversas aplicações já instaladas e um ambiente de desenvolvimento poderoso. Pode ser considerado como um sistema operacional para dispositivos móveis de caráter *open source* que foi originalmente desenvolvido por uma empresa fundada em 2003 chamada Android Inc. Em 2005 o Google como parte de sua estratégia para entrar no mercado de dispositivos móveis, comprou a Android Inc, assumindo o desenvolvimento da plataforma (LEE, 2011). Atualmente não é apenas o Google que mantem o Android e sim um grupo formado por empresas líderes do mercado de telefonia como a Motorola, LG, Samsung, Sony Ericsson e muitas outras. Esse grupo, chamado de OHA (*Open Handset Alliance*) foi criado com a intenção de padronizar uma plataforma de código aberto e livre para celulares, visando atender as expectativas e tendências do mercado atual (LECHETA, 2010).

O Android foi concebido para ser uma plataforma verdadeiramente livre. Por exemplo, uma aplicação pode invocar qualquer uma das principais funcionalidades do telefone, como fazer chamadas, enviar mensagens de texto, usar o GPS (*Global Positioning System*) ou usar a câmera, permitindo aos desenvolvedores criar experiências mais ricas para os usuários (OHA, 2007). Android sendo de código aberto, permite que desenvolvedores do mundo inteiro possam contribuir, adicionando novas funcionalidades ou corrigindo falhas (LECHETA, 2010).

2.1.1 SISTEMA OPERACIONAL LINUX

“O sistema operacional Android foi baseado no kernel 2.6 do Linux, e é responsável por gerenciar a memória, os processos, *threads* e a segurança dos arquivos e pastas, além de redes e drivers” (LECHETA, 2010).

O sistema operacional permite que vários aplicativos possam ser executados simultaneamente, gerenciando e otimizando a utilização da memória. A vantagem de se basear no Linux, além de ser de código livre, é a segurança e robustez que o sistema já apresenta ao longo dos anos de desenvolvimento.

2.1.2 MÁQUINA VIRTUAL DALVIK

O desenvolvimento de aplicações Android é feito na linguagem Java. O Java é uma linguagem de programação desenvolvida para ser portátil, isto é, que programas escritos em Java poderiam ser executados através de uma máquina virtual JVM em computadores diferentes não importando o sistema operacional (DEITEL; DEITEL, 2005). Porém devido aos recursos de *hardware* reduzidos dos dispositivos móveis, houve a necessidade de adaptar a JVM.

O Android utiliza uma máquina virtual específica e otimizada para dispositivos móveis, chamada Dalvik. Depois que o código fonte Java é compilado, ele é convertido para o formato *.dex* (*Dalvik Executable*). Todos os arquivos fonte do sistema são compilados (*.dex*) e compactados junto com outros arquivos de recursos como imagens, formando um único arquivo com extensão *.apk* (*Android Package File*) que representa a aplicação final, pronta para ser distribuída e instalada (LECHETA, 2010).

2.1.3 AMBIENTE DE DESENVOLVIMENTO

Os principais ambientes de desenvolvimentos são o Eclipse, Netbeans e IntelliJ IDEA. O Google disponibiliza gratuitamente o Android SDK (*Software Development Kit*) que contém toda biblioteca de API (*Application Programming Interface*) e as ferramentas necessária para compilar, testar e emular aplicativos para Android. Atualmente no site para desenvolvedores em Android, existe para *download* um pacote de programas chamado *ADT Bundle* que contém todos os programas necessários para o desenvolvimento, ou seja, além do Android SDK a IDE (*Integrated Development Environment* Eclipse com o *plugin* ADT (*Android Development Tools*

além de outros programas auxiliares (GOOGLE, 2013), sendo este, o ambiente de desenvolvimento adotado neste trabalho.

2.2 TEORIA DO CONTROLE REMOTO IR (INFRA-VERMELHO)

Praticamente quase todos os equipamentos de áudio e vídeo são controlados por sinais IR, tornando-se a maneira mais barata de controlar remotamente equipamentos dentro do campo de alcance de um LED IR.

Aqui será explicado a teoria de operação do controle remoto IR e alguns dos protocolos utilizados nos principais equipamentos eletrônicos comercializados.

2.2.1 O LED IR

O LED IR (*Light Emitting Diode Infra-red*) emite uma onda em 950nm que está abaixo do espectro visível, ou seja, o ser humano não consegue ver o sinal de IR, sendo esta uma das razões de se utilizá-lo para propósitos de controle.

Porém existem no ambiente muitas outras fontes de IR, tais como o sol, lâmpadas, velas e tudo mais que emita calor. Existe então, a necessidade de tomar algumas precauções para evitar que ruídos indesejados atrapalhem o sinal de controle. A modulação é o recurso utilizado para isto, onde faz-se o LED IR piscar em uma frequência particular que pode ser sintonizada e detetada pelo receptor de IR, ignorando qualquer outro sinal. Frequências entre 30kHz e 60 kHz são utilizados e a modulação é composta de “marcas” (pulsando IR) e “espaços” (sem IR). A relação entre marcas e espaços depende do protocolo utilizado que será descrito posteriormente. (BERGMANS, 2011)

2.2.2 O TRANSMISSOR IR E O RECEPTOR IR

O transmissor pode ser implementado simplesmente por um LED IR e um transistor em corte e saturação conforme mostrado na Figura 2. A corrente que passa pelo LED IR pode variar de 100mA a 1A. Uma boa relação entre os parâmetros do LED IR, a capacidade da bateria e a máxima distância de controle deve ser obtida. Estas correntes altas podem ser aplicadas, pois os pulsos no LED são muito curtos, mas a média da potência dissipada não deve ultrapassar o máximo valor especificado.

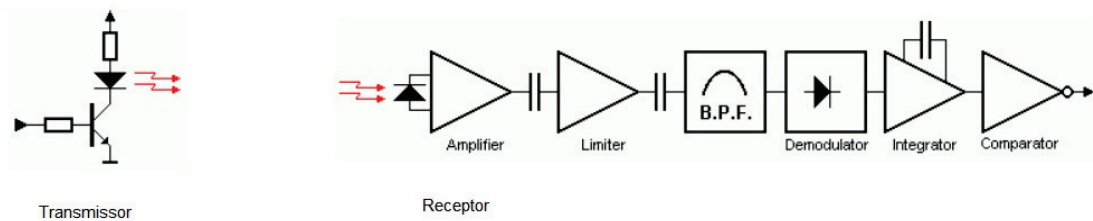


Figura 2: Transmissor e Receptor IR.

Fonte: (BERGMANS, 2011)

O receptor IR também é mostrado na Figura 2 e é composto por um diodo detetor de IR cujo sinal é amplificado e limitado para manter o nível do pulso recebido constante, não importando a distância do controle. O filtro passa faixa BPF (*Band Pass Filter*) é sintonizado na frequência de modulação (30kHz a 60kHz). Os próximos blocos demodulador, integrador e comparador são para detetar a frequência modulada. Apesar da aparente complexidade do receptor, hoje em dia, todo o bloco receptor é encontrado encapsulado em um único componente.

2.2.3 EXEMPLOS DE PROTOCOLOS IR

Diversos fabricantes durante o desenvolvimento dos controles remotos via IR, criaram seus próprios protocolos para transmissão do sinal IR, surgindo então, vários protocolos diferentes. Para o perfeito funcionamento com os diversos equipamentos, o controle remoto universal deve transmitir seguindo o protocolo específico para cada tipo de equipamento. As próximas seções apresentarão alguns exemplos de protocolos utilizados.

2.2.3.1 PROTOCOLO SONY SIRC

Características:

- 12, 15 e 20 bits
- 5 bits de endereço e 7 bits de comando (para protocolo de 12 bits)
- frequência de modulação 40kHz

Modulação para lógica 1 com *burst* (trem de pulsos) de 1,2ms e pausa de 0,6ms e para lógica 0 com *burst* de 0,6ms e pausa de 0,6ms, conforme mostrado na Figura 3.

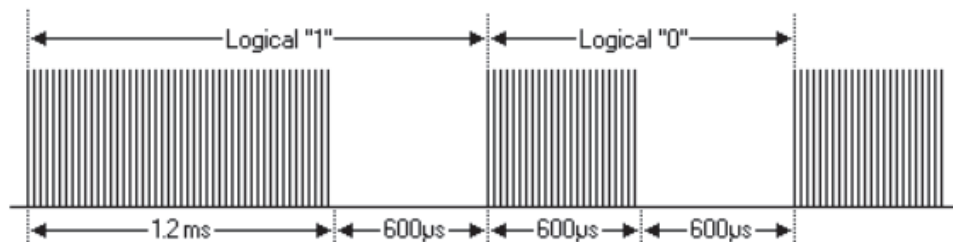


Figura 3: Modulação Sony SIRC

Fonte: (BERGMANS, 2011)

O protocolo SIRC (*Serial Infra-Red Control*) é composto por um *burst* de início de 2,4ms seguido de uma pausa de 0,6ms. Então são transmitidos os 7 bits de comando e em seguida os 5 bits de endereço, sempre começando pelo bit menos significativo. A Figura 4 mostra a transmissão do comando 19 e endereço 1.

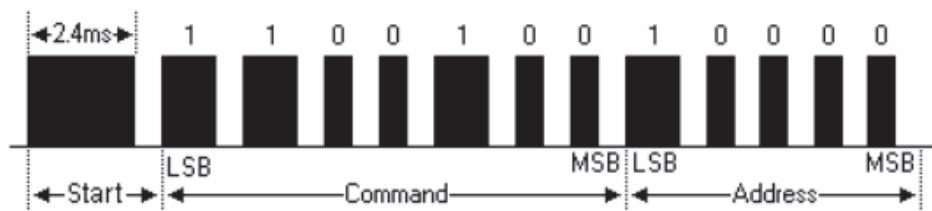


Figura 4: Protocolo Sony SIRC - Transmissão do comando 19 e endereço 1

Fonte: (BERGMANS, 2011)

2.2.3.2 PROTOCOLO RCA

Características:

- 12 bits
- 4 bits de endereço e 8 bits de comando
- frequência de modulação 56kHz

Modulação para lógica 1 com *burst* de 0,5ms e pausa de 2,0ms e para lógica 0 com *burst* de 0,5ms e pausa de 1,0ms, conforme mostrado na Figura 5.

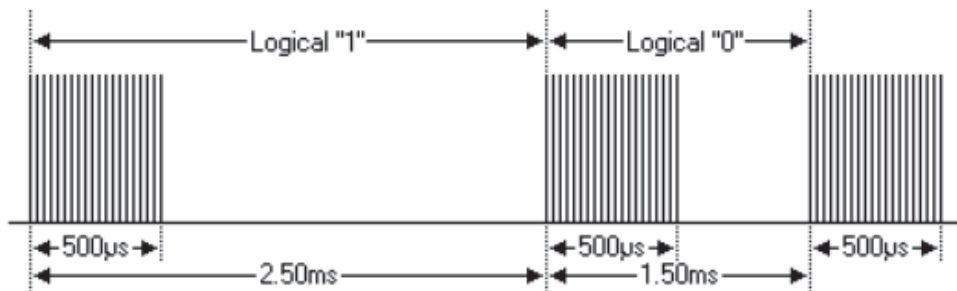


Figura 5: Modulação RCA

Fonte: (BERGMANS, 2011)

O protocolo RCA é composto por um *burst* de início de 4ms seguido de uma pausa de 4ms. Então são transmitidos os 4 bits de endereço e em seguida os 8 bits de comando. Uma nova sequência com os bits invertidos é transmitida, sempre começando pelo bit mais significativo. A Figura 6 mostra a transmissão do comando 68h e endereço Ah.

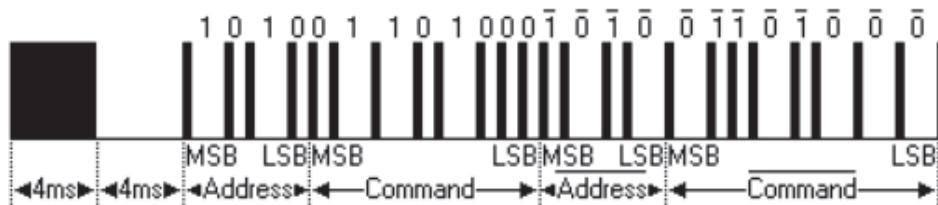


Figura 6: Protocolo RCA - Transmissão do endereço Ah e comando 68h

Fonte: (BERGMANS, 2011)

2.2.3.3 PROTOCOLO PHILIPS RC5

Características:

- 11 bits
- 5 bits de endereço e 6 bits de comando
- frequência de modulação 36kHz

Um dos protocolos mais utilizados pela simplicidade. Utiliza modulação bifásica em portadora IR de 36kHz. Todos os bits tem o mesmo comprimento de 1,778ms. Para o bit 1 com pausa de 889 μ s e *burst* de 889 μ s. Para o bit 0 com *burst* de 889 μ s e pausa de 889 μ s, conforme mostrado na Figura 7.

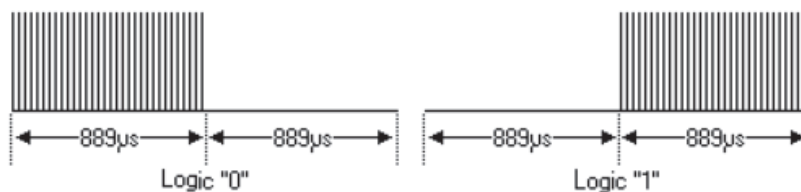


Figura 7: Modulação Philips RC5

Fonte: (BERGMANS, 2011)

O protocolo RC5 é composto por 2 pulsos de início de lógica 1 seguido de um bit *toggle*. Este bit é invertido toda vez que a tecla é apertada novamente, isto serve para o receptor poder distinguir se a tecla foi mantida pressionada ou se está sendo repetidamente pressionada. Os próximos 5 bits representam o endereço seguidos de mais 6 bits de comando, sempre começando pelo bit mais significativo. A Figura 8 mostra a transmissão do endereço 5h e comando 35h.



Figura 8: Protocolo RC5 - Transmissão do endereço 5h e comando 35h

Fonte: (BERGMANS, 2011)

2.3 O KIT DE DESENVOLVIMENTO STELLARIS EKS-LM3S6965

O Kit de desenvolvimento Stellaris EKS-LM3S6965 é uma placa eletrônica de avaliação que fornece uma maneira fácil e barata de iniciar desenvolvimentos de aplicações com recursos de *Ethernet* e microcontrolador *Stellaris* LM3S6965 ARM Cortex M3 da *Texas Instruments*.

Características do kit:

- Microcontrolador *Stellaris* LM3S6965 ARM Cortex M3
- Controlador *Ethernet* 10/100
- Conexão USB para comunicação serial, emulação e alimentação
- Display gráfico OLED
- Led e chaves disponível para uso geral
- Alto-falante magnético
- Pinos de I/O disponíveis
- Interface JTAG padrão de 20 pinos
- *Slot* para cartão MicroSD

A Figura 9 mostra a foto do kit *Stellaris* LM3S6965.

O *Stellaris* LM3S6965 ARM Cortex M3 é um microcontrolador de 32 bits RISC de 50MHz, 256KB de memória *Flash*, 64KB de memória SRAM e com *Ethernet* integrada. Recentemente a *Texas Instruments* não recomenda este microcontrolador para novos desenvolvimentos, sendo que deve-se utilizar a nova família de microcontroladores denominada Tiva. O trabalho ainda foi realizado com o LM3S6965 devido a disponibilidade do kit de desenvolvimento e ciente que segundo a própria *Texas Instruments* a migração é simples e fácil.

2.3.1 AMBIENTE INTEGRADO DE DESENVOLVIMENTO

Para o ambiente de desenvolvimento do kit *Stellaris* EKS-LM3S6965 a *Texas Instruments* disponibiliza o CCS (*Code Composer Studio*) e outros fabricantes de

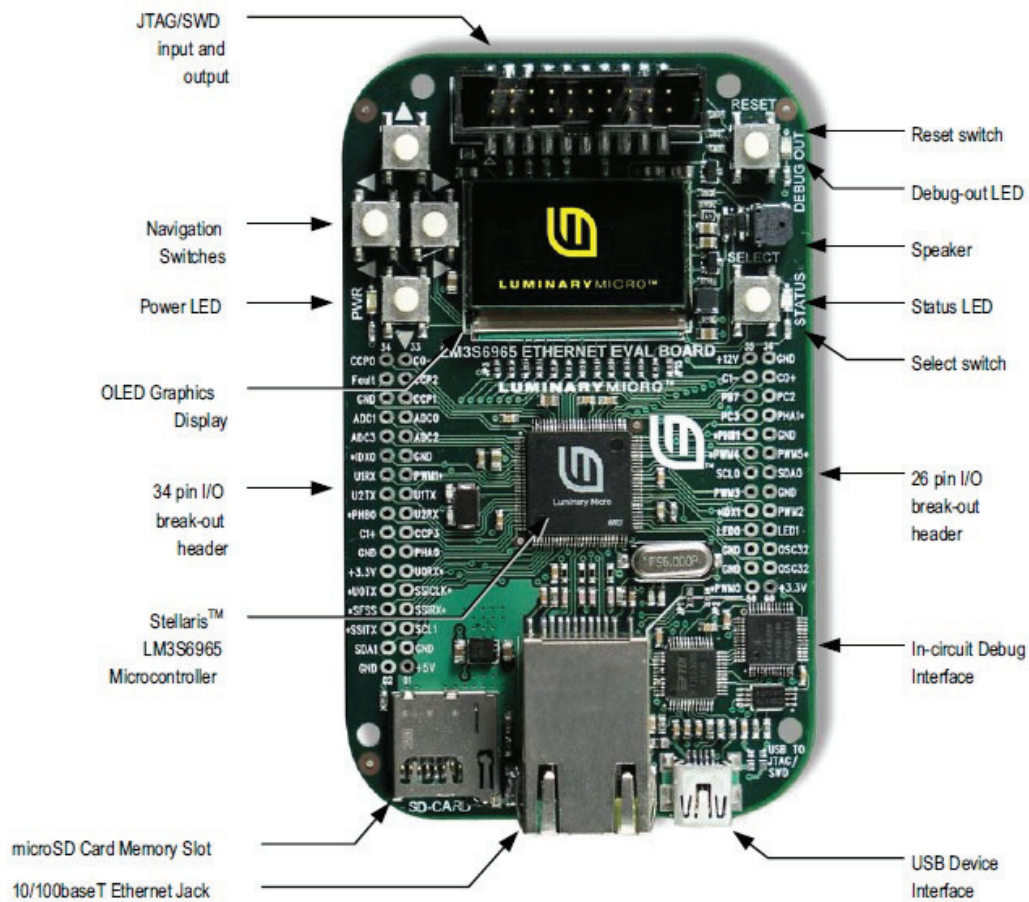


Figura 9: Kit de Desenvolvimento LM3S6965

Fonte: (TEXAS, 2010)

ambientes de desenvolvimento como a IAR SYSTEMS e a KEIL disponibilizam uma versão para avaliação, ambos com limitação de código de 32KB.

No trabalho foi utilizado a IDE da IAR SYSTEMS devido ao conhecimento prévio do autor neste ambiente para outros microcontroladores e sabendo que a limitação de 32KB não seria problema para o mesmo.

2.4 DESENVOLVIMENTO DE SOFTWARE

Com o desenvolvimento das tecnologias de informação, a área de desenvolvimento de software entre 1989 e 1994 teve sua evolução marcada pelo fenômeno que foi chamado a guerra dos métodos, quando surgiram quase 50 métodos de desenvolvimento orientados a objetos. Entre os mais importantes estavam o método de G. Booch, a *Object Modeling Technique* de J. Rumbaugh e o método

Objectory de I. Jacobson que introduziu a modelagem de casos de uso em 1987 (PIMENTEL, 2007).

Como cada método possuía uma notação própria, surgiram problemas de comunicação, treinamento e portabilidade. Em 1994 J. Rumbaugh, I. Jacobson e G. Booch juntaram-se num esforço para unificação dos métodos através da empresa *Rational Software Corporation*. O primeiro grande resultado desse esforço foi a criação da *Unified Modeling Language* (UML), apresentada na versão 1.0 em 1997 (PIMENTEL, 2007).

2.4.1 UML

A UML foi adotada desde 1997 como padrão internacional pelo OMG *Object Management Group*, sendo uma linguagem para visualizar, especificar, construir e documentar os artefatos de um sistema de software. A UML especifica um conjunto de diagramas e seus componentes, todos com notação e comportamento bem definidos.(PIMENTEL, 2007).

Existe três modos de aplicar UML (LARMAN, 2007):

- UML como rascunho - diagramas incompletos e informais criados para explorar partes específicas do problema.
- UML como planta de software - diagramas de projeto relativamente detalhados usados para engenharia reversa e/ou geração de código.
- UML como linguagem de programação - especificação executável completa de um sistemas de software em UML. Gera automaticamente a partir dos diagramas UML o código executável.

Seguindo uma abordagem ágil, onde a finalidade da modelagem é principalmente a compreensão do problema. Enfatiza-se, a UML como rascunho e o ato de modelar deve fornecer um modo melhor de entender o problema e a solução para rapidamente explorar alternativas e o caminho para um bom projeto orientado a objeto (POO). Esta visão consistente com métodos ágeis tem sido chamada de modelagem ágil (LARMAN, 2007).

2.5 SOLUÇÕES EQUIVALENTES

Descreve-se aqui algumas soluções equivalentes encontradas atualmente no mercado.

2.5.1 APLICATIVOS DE CONTROLE REMOTO ESPECÍFICOS PARA *SMART TVS*

Já existe atualmente diversos aplicativos disponíveis na *Play Store* da Google que fazem a função de controle remoto. Porém estes aplicativos somente funcionam para *Smart TVs* e equipamentos *Home Theater* que possuem entrada para rede *Ethernet* via cabo ou Wi-Fi. Ficam então, limitados nos equipamentos mais caros de última geração e não controlam outros equipamentos da sala de *Home Theater* que não tem este recurso.

2.5.2 EQUIPAMENTOS SIMILARES

Dispositivo *Beacon* da *Griffin Technology*, utiliza tecnologia sem fio *Bluetooth* para comunicação com o *Smartphone* e IR com o sistema de áudio e vídeo. Funciona junto com o aplicativo *Dijit*, disponibilizado gratuitamente na *Play Store* da Google ou na *iTunes* da Apple, porém tem que adquirir o dispositivo *Beacon*. É comercializado em versões separadas para o *iOS* e o *Android*. A Figura 10 ilustra uma TV sendo controlada com o *Beacon* e um *iPhone*.



Figura 10: Controle de TV com Beacon e iPhone

Fonte: (GRIFFIN, 2013)

Dispositivo *RedEye* da *Think Flood*, utiliza tecnologia sem fio *Wi-Fi* para comunicação com o *Smartphone* e IR com o sistema de áudio e vídeo. Funciona junto com o aplicativo *RedEye*, disponibilizado gratuitamente na *Play Store* da Google

ou na iTunes da Apple, porém tem que adquirir o dispositivo RedEye. A Figura 11 mostra o dispositivo RedEye.



Figura 11: Dispositivo RedEye

Fonte: (THINKFLOOD, 2013)

Dispositivo *BlueIR* da *PDAwin*, utiliza tecnologia sem fio *Bluetooth* para comunicação com o *Smartphone* e *IR* com o sistema de áudio e vídeo. Funciona junto com o aplicativo *BlueIR*, disponibilizado gratuitamente na *Play Store* da Google, porém tem que adquirir o dispositivo *BlueIR*. O diferencial deste dispositivo é que pode-se tirar a foto do controle remoto original e configurar as teclas do mesmo na tela do *smartphone*. A Figura 12 ilustra o esquema de controle com o *BlueIR*.

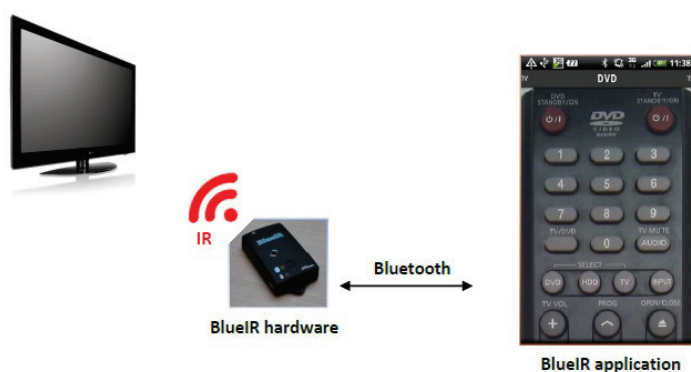


Figura 12: Dispositivo BlueIR

Fonte: (PDAWIN, 2012)

2.5.3 APLICATIVOS DE CONTROLE REMOTO ESPECÍFICOS PARA SMARTPHONES E TABLETS COM IR EMBUTIDO NO APARELHO

Alguns *smartphones* como o Samsung Galaxy S4 o HTC one e o *tablet* Samsung Galaxy Note já vêm com o IR embutido no aparelho. Então aplicativos como o *Peel Smart Remote* que vem instalado no Galaxy S4/Galaxy Note II e o mbhRemote comercializado na *Play Store* da Google, controlam diretamente os dispositivos de áudio e vídeo através do IR.

A Figura 13 mostra a tela do aplicativo *Peel Smart Remote* sendo executado em um *Tablet Samsung Galaxy Note*.

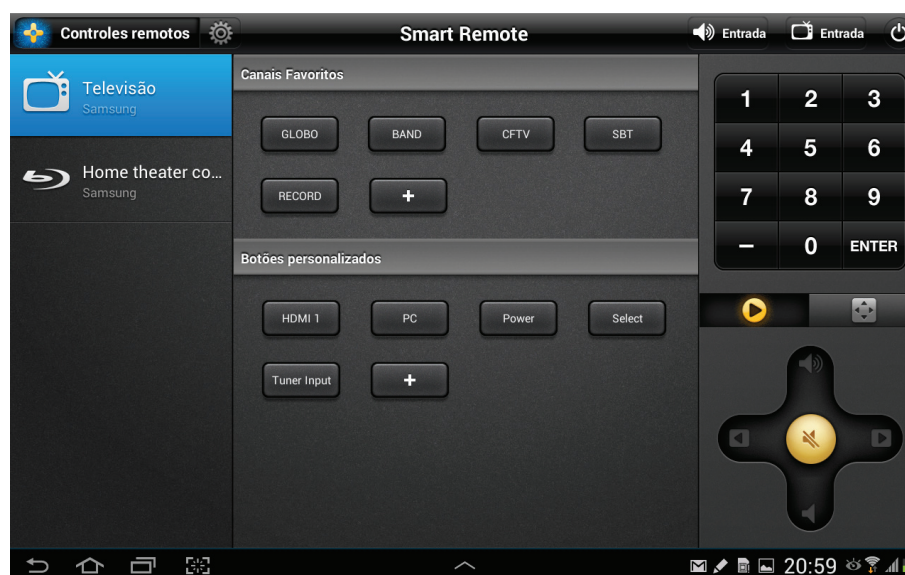


Figura 13: Tela do aplicativo Peel Smart Remote

Fonte: Autoria própria

3 DESENVOLVIMENTO

Para o desenvolvimento foram executadas algumas etapas descritas a seguir.

3.1 LEVANTAMENTO DE REQUISITOS

Requisitos são as capacidades e condições às quais o sistema ou projeto deve atender (LARMAN, 2007).

Os requisitos funcionais abordam o que o sistema deve fazer e os requisitos não funcionais declaram características de qualidade que o sistema deve possuir e que estão relacionados às suas funcionalidades.

Segue a descrição dos requisitos funcionais, não funcionais e as restrições do trabalho proposto.

3.1.1 REQUISITOS FUNCIONAIS

- Controlar as principais funcionalidades da televisão;
- Controlar as principais funcionalidades do *Home Theater*;
- Controlar as principais funcionalidades do aparelho de DVD/*Blu-Ray*;
- Controlar as funcionalidades do controlador de iluminação;

3.1.2 REQUISITOS NÃO FUNCIONAIS

- A interface deve ser simples e intuitiva com uso de toques de tela (*touch screen*);
- O aplicativo deve funcionar em dispositivos móveis *smartphones e tablets* com sistemas operacional Android a partir da versão 2.3 *Gingerbread*;
- O tempo de resposta deve ser rápido o suficiente para agradar o usuário;

3.1.3 RESTRIÇÕES

- O dispositivo móvel deve ter acesso a rede Wi-Fi;
- A sala de *Home Theater* deve ter redes *Ethernet* e Wi-Fi;
- O dispositivo IR deve ter campo de visão para todos os equipamentos da sala;

3.2 MODELAGEM

Na modelagem do aplicativo, foi utilizada a linguagem UML aplicando a visão de modelagem ágil, ou seja, implementando somente os principais diagramas necessários para entender o sistema.

Foi utilizado o programa para modelagem UML chamado *astah community* que é a versão livre do programa e atende os requisitos para modelagem ágil do sistema.

3.2.1 MODELO DE CASO DE USO

Modelo de caso de uso ilustra as funções pretendidas do sistema (casos de uso), suas vizinhanças (atores) e relacionamentos entre os casos de uso e atores (diagrama de casos de uso) (PIMENTEL, 2007).

A fase de concepção do modelo de caso de uso é a identificação dos atores e casos de uso principais do sistemas.

3.2.1.1 DEFINIÇÃO DOS ATORES

Atores são entidades externas ao *software*. Eles representam algo ou alguém que deve interagir com o sistema. No sistema proposto, foram definidos os seguintes atores:

- Usuários - Todos os usuários da sala de *Home Theater* que se beneficiarão com o uso do sistema;
- Dispositivo IR - Placa eletrônica que faz a interface entre os dispositivos móveis e os equipamentos da sala de *Home Theater*

3.2.1.2 DEFINIÇÃO DOS CASOS DE USO

Casos de uso representam as funcionalidades fornecidas pelo sistema, isto é, que capacidades serão providas para o ator pelo sistema (PIMENTEL, 2007).

Pode-se dizer que os casos de uso são requisitos funcionais do sistema, embora nem todos requisitos funcionais sejam casos de uso. É uma boa maneira de manter as coisas simples, ajudando a identificar os objetos e a compreender o que o sistema deve realizar (LARMAN, 2007).

No sistema proposto, foram definidos os seguintes casos de uso:

- Controlar TV - Usuário irá interagir através de uma tela com as teclas específicas para o controle da televisão, os controles serão enviados via rede Wi-Fi e *Ethernet* para o Dispositivo IR que efetivamente controla a televisão via IR;
- Controlar *Blu-Ray* - Usuário irá interagir através de uma tela com as teclas específicas para o controle do *Blu-Ray*, os controles serão enviados via rede Wi-Fi e *Ethernet* para o Dispositivo IR que efetivamente controla o *Blu-Ray* via IR;
- Controlar *Home Theater* - Usuário irá interagir através de uma tela com as teclas específicas para o controle do *Home Theater*, os controles serão enviados via rede Wi-Fi e *Ethernet* para o Dispositivo IR que efetivamente controla o *Home Theater* via IR;
- Controlar Iluminação - Usuário irá interagir através de uma tela com as teclas específicas para o controle da Iluminação, os controles serão enviados via rede Wi-Fi e *Ethernet* para o Dispositivo IR que efetivamente controla a Iluminação via IR;

3.2.1.3 DIAGRAMA DE CASO DE USO

Diagrama de Caso de Uso é uma visão gráfica de alguns ou todos os atores, casos de usos e seus relacionamentos identificados para um sistema.

A Figura 14 apresenta o diagrama de casos de uso do aplicativo desenvolvido.

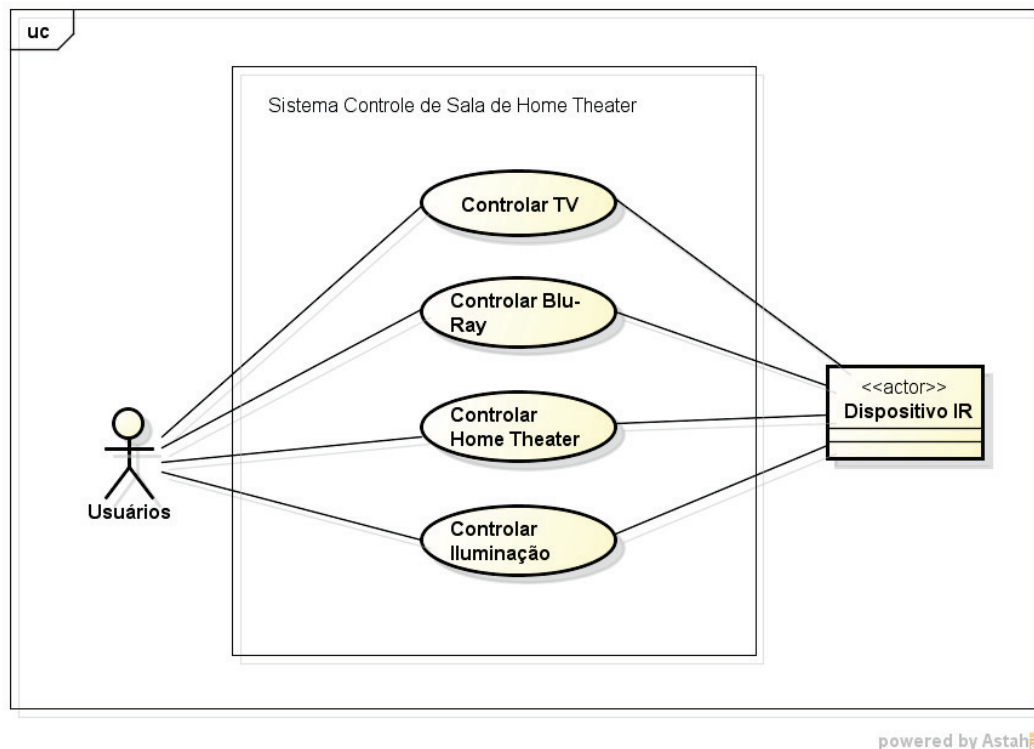


Figura 14: Diagrama de caso de uso do sistema controle de sala de *Home Theater*

Fonte: Autoria própria

3.2.2 LEVANTAMENTO DE CLASSES

Uma classe representa um conjunto de objetos com características afins, ou seja, uma classe define o comportamento dos objetos através de seus métodos e quais estados ele é capaz de manter através de seus atributos.

Identificação das classes do sistema usando o padrão MVC (Modelo Visão Controle) que se ajustam aos estereótipos de classe entidade, controle e fronteira.

- Entidade - Classe que representa coisas. Contém dados que representam uma identidade.
- Controle - Classe que controla a execução de processos. Contém o algoritmo do fluxo de execução de todo ou de parte dos casos de uso e comandam outras classes na execução de procedimentos.
- Fronteira - Classe que faz a interface com o exterior, ou seja, com atores. Contém o protocolo de comunicação com os atores.

3.2.2.1 IDENTIFICANDO CLASSES FRONTEIRA

Identifica-se e lista-se aqui as classes que fazem a interface dos atores com os casos de uso:

- CIntUsuCtlTV - Classe Interface Usuários com Controlar TV
- CIntUsuCtlBlu - Classe Interface Usuários com Controlar *Blu-Ray*
- CIntUsuCtlHT - Classe Interface Usuários com Controlar *Home Theater*
- CIntUsuCtlllu - Classe Interface Usuários com Controlar Iluminação
- CIntDispIR - Classe Interface Dispositivo IR com todos casos de uso Controlar

3.2.2.2 IDENTIFICANDO CLASSES ENTIDADE

Identifica-se e lista-se aqui as classes tipo entidade para cada grupo de dados que representam uma mesma coisa:

- CEntidadeTV - Classe Entidade TV
- CEntidadeBlu - Classe Entidade *Blu-Ray*
- CEntidadeHT - Classe Entidade *Home Theater*
- CEntidadellu - Classe Entidade Iluminação

3.2.2.3 IDENTIFICANDO CLASSES CONTROLE

Identifica-se e lista-se aqui as classes tipo controle para cada caso de uso:

- CControleCtlTV - Classe Controle do caso de uso Controlar TV
- CControleCtlBlu - Classe Controle do caso de uso Controlar *Blu-Ray*
- CControleCtlHT - Classe Controle do caso de uso Controlar *Home Theater*
- CControleCtlllu - Classe Controle do caso de uso Controlar Iluminação

3.2.3 ESTUDO DAS INTERAÇÕES ENTRE OBJETOS

Os objetos de um sistema computacional interagem entre si através de comunicações. O diagrama de sequência é uma notação para a descrição destas interações (STADZISZ, 2002).

3.2.3.1 DIAGRAMA DE SEQUÊNCIA

Descreve ao longo de uma linha de tempo a sequência das comunicações necessárias entre os objetos para a realização dos processos em um sistema computacional (STADZISZ, 2002).

Segue os diagramas de sequência criados para o sistema:

Figura 15 mostra o diagrama de sequência do caso de uso Controlar TV.

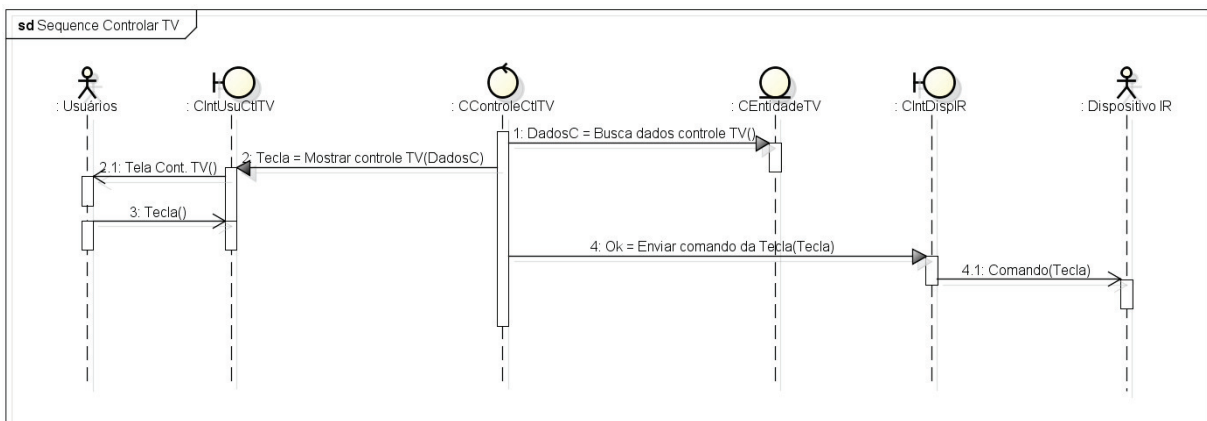


Figura 15: Diagrama de Sequência do caso de uso Controlar TV

Fonte: Autoria própria

Figura 16 mostra o diagrama de sequência do caso de uso Controlar *Blu-Ray*.

Figura 17 mostra o diagrama de sequência do caso de uso Controlar *Home Theater*.

Figura 18 mostra o diagrama de sequência do caso de uso Controlar Iluminação.

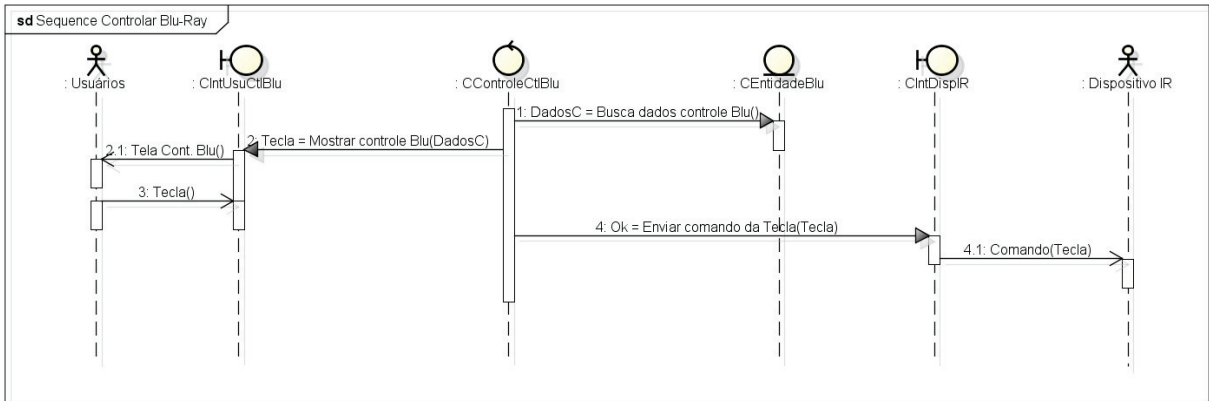


Figura 16: Diagrama de Sequência do caso de uso Controlar *Blu-Ray*
Fonte: Autoria própria

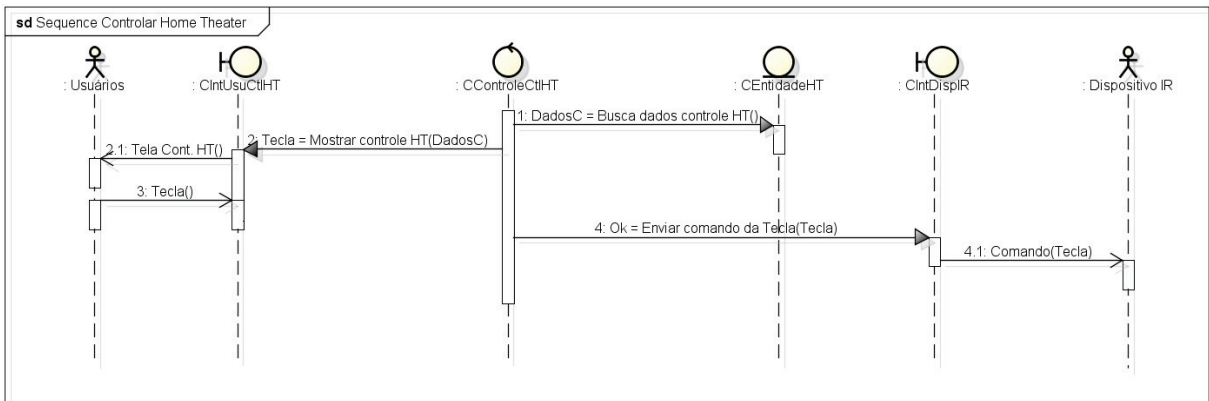


Figura 17: Diagrama de Sequência do caso de uso Controlar *Home Theater*
Fonte: Autoria própria

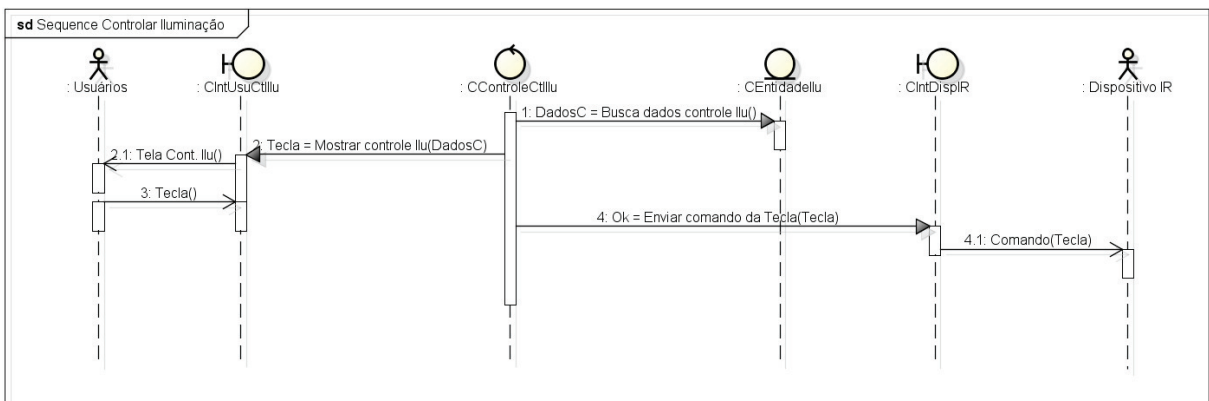


Figura 18: Diagrama de Sequência do caso de uso Controlar *Iluminação*
Fonte: Autoria própria

3.2.3.2 DIAGRAMA DE COMUNICAÇÃO

Diagramas de comunicação derivam dos diagramas de sequência. Descrevem as interações de comunicação entre os objetos, sem a noção de tempo, para a realização de um caso de uso.

Segue os diagramas de comunicação criados para o sistema:

A Figura 19 mostra o diagrama de comunicação do caso de uso Controlar TV.

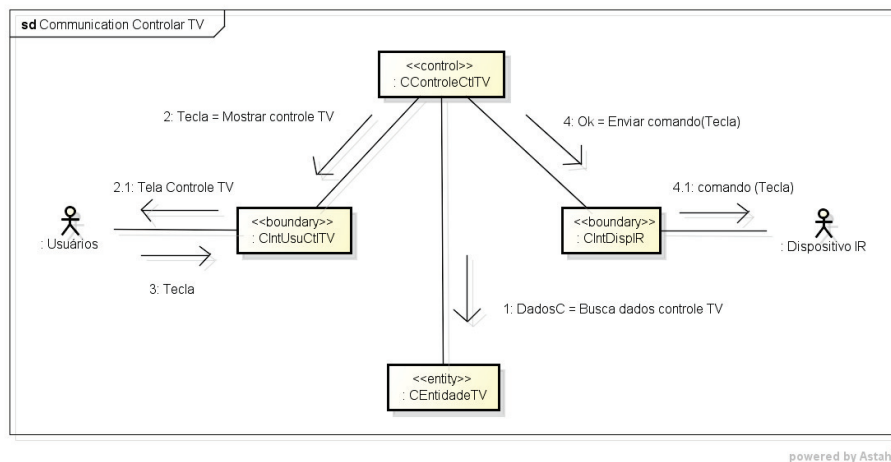


Figura 19: Diagrama de Comunicação do caso de uso Controlar TV

Fonte: Autoria própria

A Figura 20 mostra o diagrama de comunicação do caso de uso Controlar *Blu-Ray*.

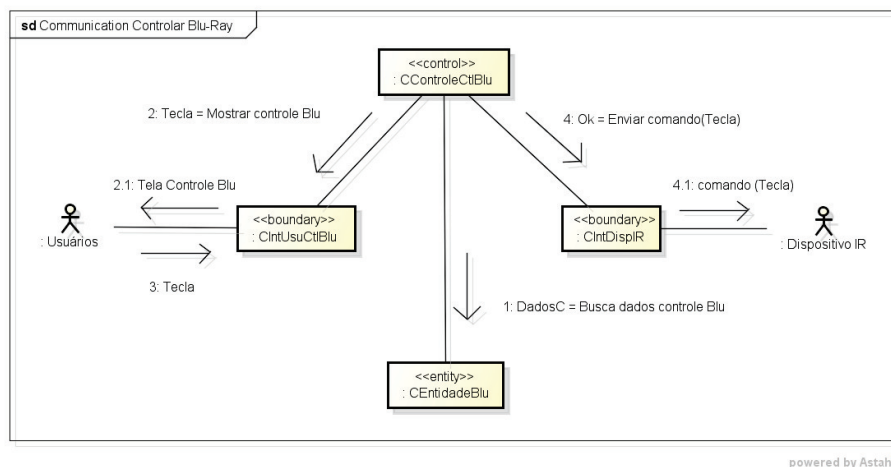


Figura 20: Diagrama de Comunicação do caso de uso Controlar *Blu-Ray*

Fonte: Autoria própria

A Figura 21 mostra o diagrama de comunicação do caso de uso Controlar *Home Theater*.

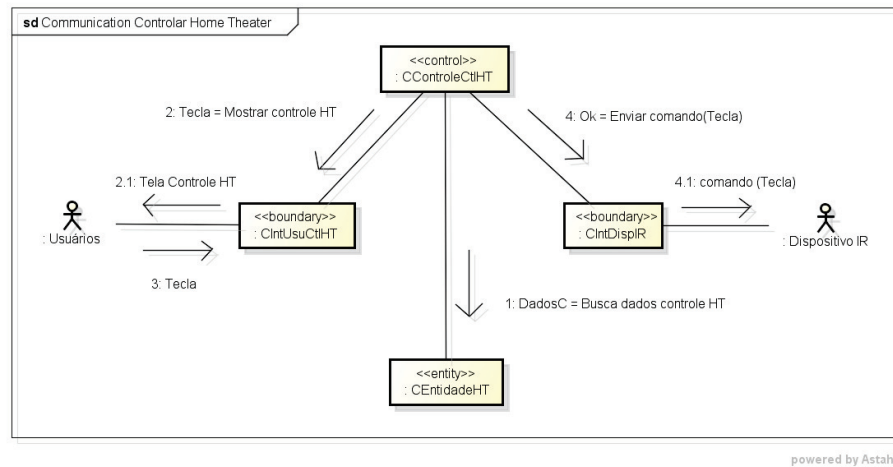


Figura 21: Diagrama de Comunicação do caso de uso Controlar *Home Theater*

Fonte: Autoria própria

A Figura 22 mostra o diagrama de comunicação do caso de uso Controlar Iluminação.

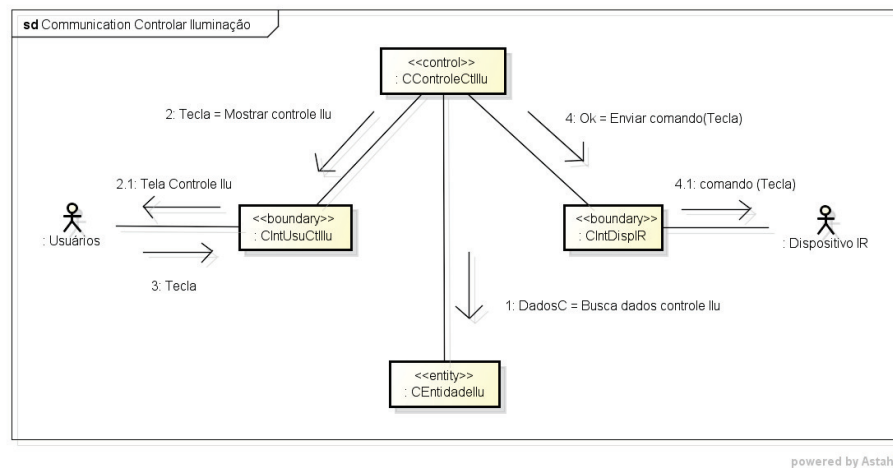


Figura 22: Diagrama de Comunicação do caso de uso Controlar Iluminação

Fonte: Autoria própria

3.2.4 RELACIONAMENTO ENTRE CLASSES E DIAGRAMA DE CLASSES

Os relacionamentos estruturais entre classes que existem em todo sistema orientado a objetos, permite a interação entre os objetos para a realização dos casos de uso.

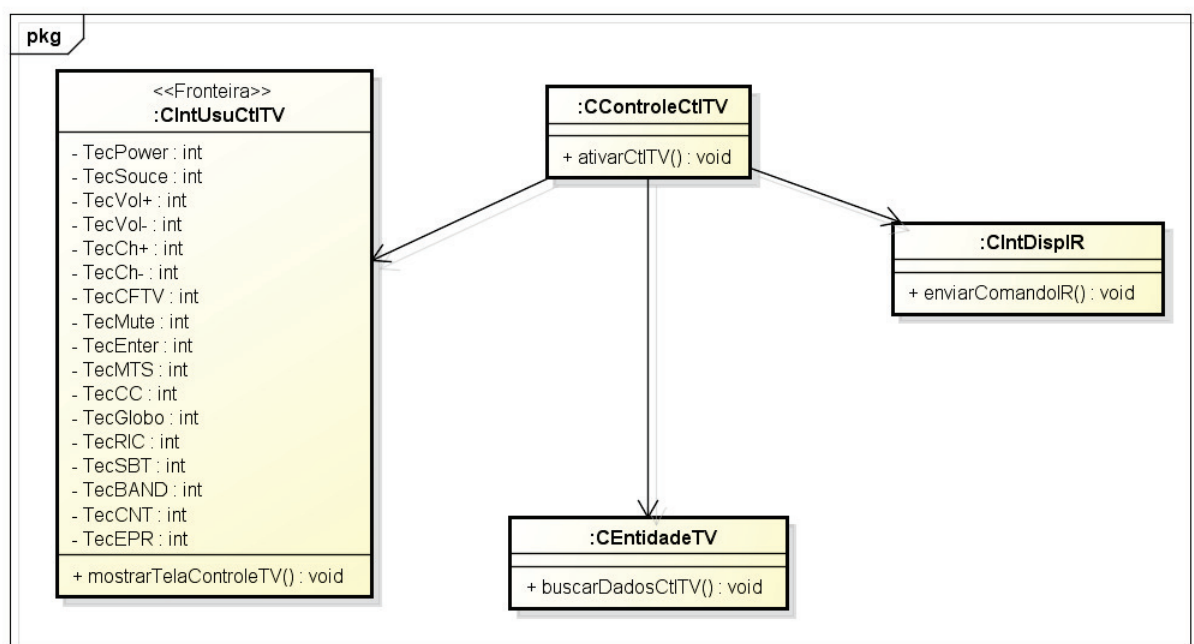
Estes relacionamentos podem ser:

- Associações - São comunicações entres as classes.
- Agregação - Subdivisão de classes maiores em classes menores.
- Generalização - Quando uma classe herda atributos e métodos de uma outra classe mais genérica.

Os diagramas de classe são compostos pela definição das classes e destes relacionamentos, sendo um dos diagramas mais importantes para o desenvolvimento de programas orientados a objetos, pois eles descrevem o esqueleto do sistema sendo projetado e a partir dele já é possível a geração parcial do código fonte (STADZISZ, 2002).

Segue os diagramas de classes criados para o sistema:

Figura 23 mostra o diagrama de classes do caso de uso Controlar TV.

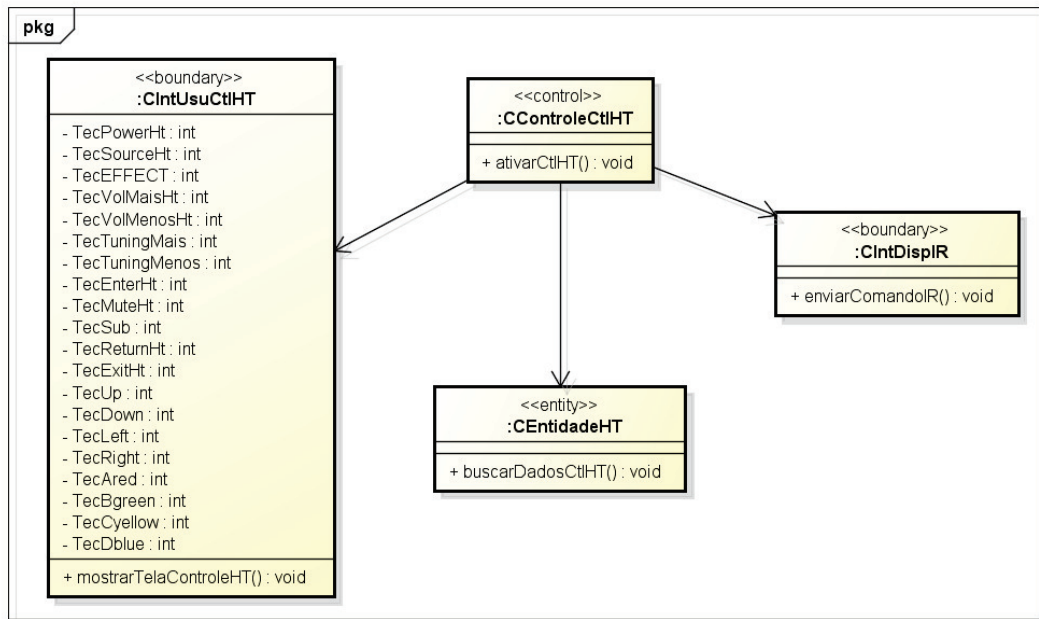


powered by Astah

Figura 23: Diagrama de Classes do caso de uso Controlar TV

Fonte: Autoria própria

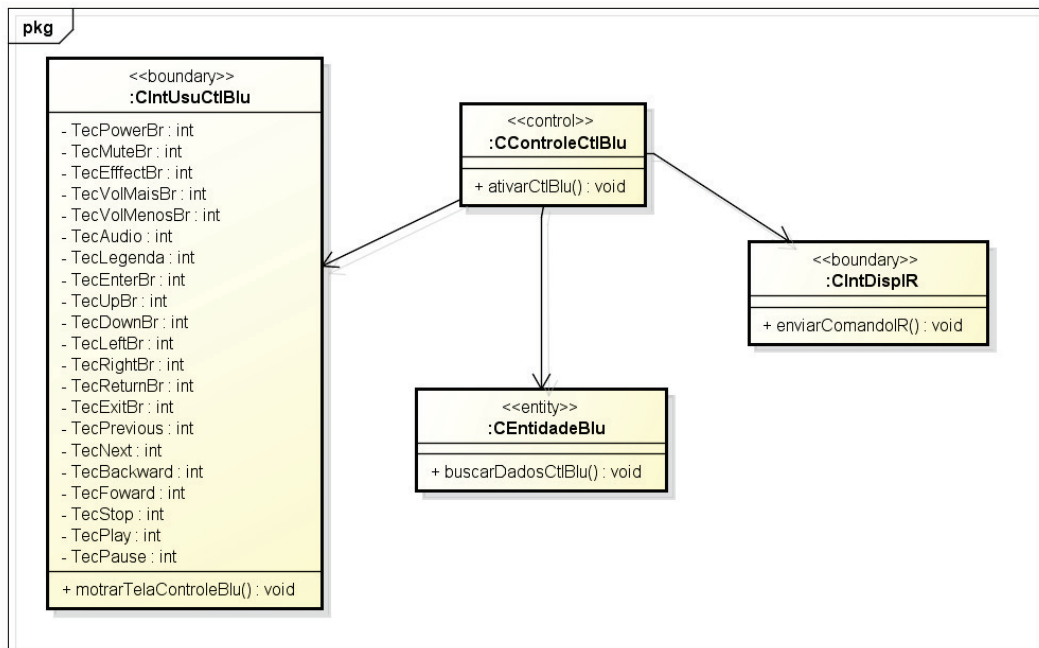
Figura 24 mostra o diagrama de classes do caso de uso Controlar *Home Theater*.



powered by Astah

Figura 24: Diagrama de Classes do caso de uso Controlar *Home Theater*

Fonte: Autoria própria



powered by Astah

Figura 25: Diagrama de Classes do caso de uso Controlar *Blu-Ray*

Fonte: Autoria própria

Figura 25 mostra o diagrama de classes do caso de uso Controlar *Blu-Ray*.

Figura 26 mostra o diagrama de classes do caso de uso Controlar cenários de iluminação.

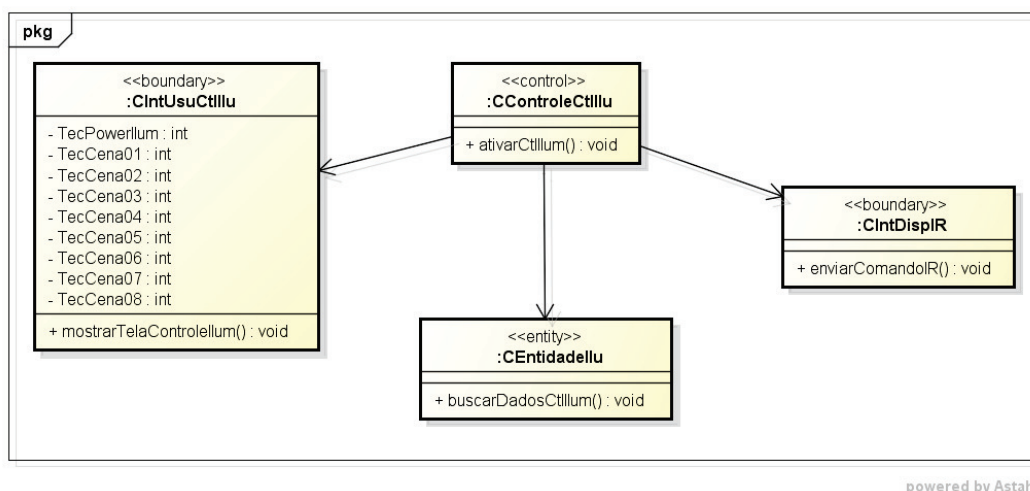


Figura 26: Diagrama de Classes do caso de uso Controlar Cenários de Iluminação

Fonte: Autoria própria

3.3 DESIGN DA INTERFACE

Para que o aplicativo fosse simples e intuitivo, no projeto da interface, foram utilizados ícones com desenhos que traduzissem bem as ideias de controle remoto, TV, *Home Theater* e iluminação. A mudança de telas foi simplificada com o uso de abas e optou-se em utilizar apenas as teclas principais e mais utilizadas de cada controle.

A aba selecionada apresenta o ícone diferenciado por cor e todas as teclas possuem *feedback* por som (*beep*) e por mudança momentânea de cor.

A Figura 27 mostra o ícone do aplicativo e a Figura 28 mostra a tela do emulador Android SDK para o controle da TV.

3.4 DESENVOLVIMENTO DO APLICATIVO ANDROID

O protótipo do aplicativo é composto por quatro telas, cada uma representa uma atividade (*Activity*). Cada tela tem uma função específica de controle remoto de Televisão, *Home Theater*, *Blu-Ray* e Cenários de Iluminação, sendo as mesmas selecionadas por abas na parte superior das telas.



Figura 27: Ícone do Aplicativo Desenvolvido

Fonte: (ICONESBR, 2013)

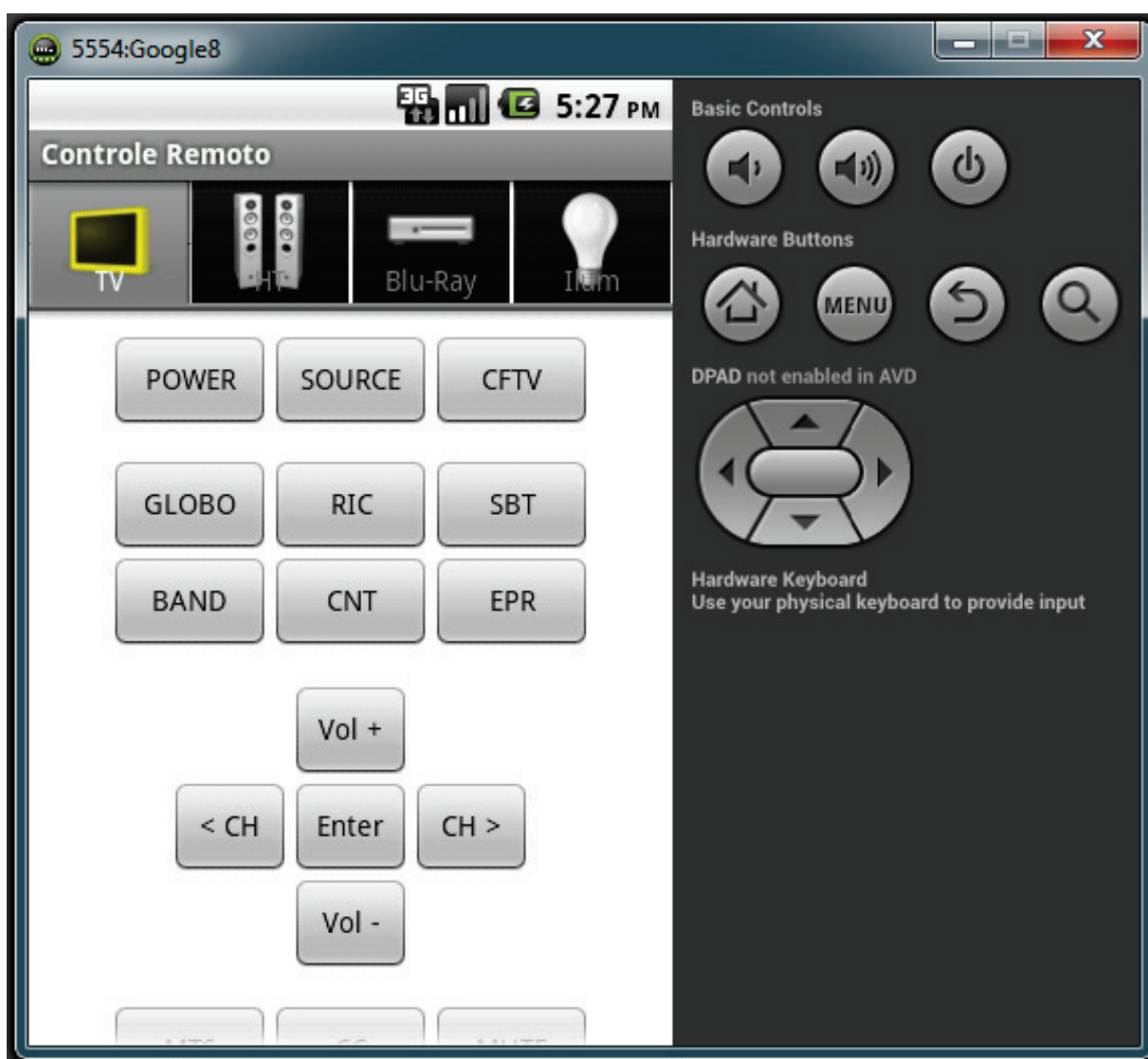


Figura 28: Tela do Emulador Android SDK para o controle da TV

Fonte: Autoria própria

O princípio de funcionamento é igual para todos os controles, sendo que ao pressionar uma tecla do controle é feita uma requisição HTTP com envio de

parâmetros por GET, onde os parâmetros são enviados diretamente na URL. Estes parâmetros estão explicados na seção 3.5.2 de Desenvolvimento do *Firmware*.

3.4.1 TELAS DO APLICATIVO

A Figura 29 mostra como ficaram as telas de controle da TV e do *Home Theater* diretamente no celular Samsung Galaxy S3.

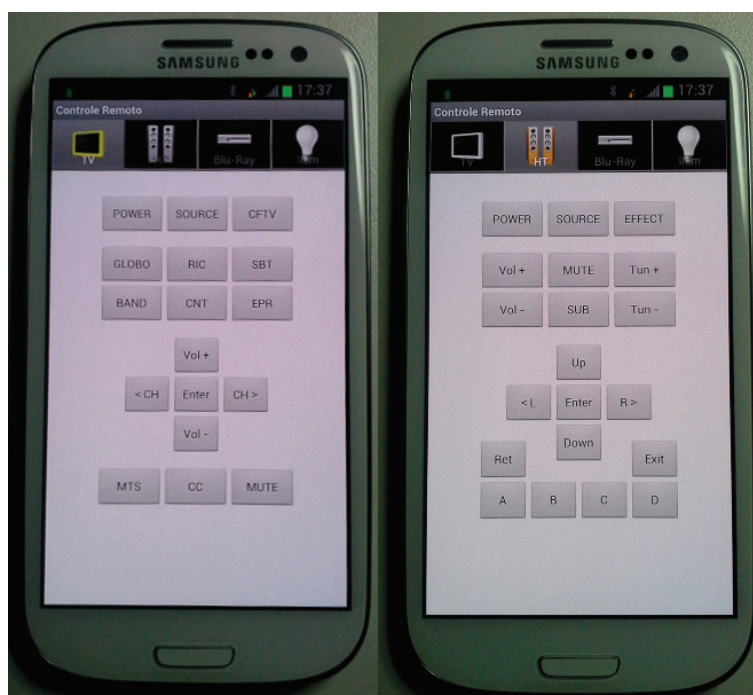


Figura 29: Telas de Controle Remoto TV e *Home Theater* no Samsung Galaxy S3

Fonte: Autoria própria

O aplicativo também permite a utilização na orientação horizontal (paisagem) com o uso de *scroll* de tela. A Figura 30 mostra a tela de controle da TV nesta orientação no celular Samsung Galaxy S3.

A Figura 31 mostra como ficaram as telas de controle do *Blu-Ray* e dos cenários de iluminação diretamente no celular Samsung Galaxy S3.

A Figura 32 mostra a tela de controle do *Home Theater* executando no *tablet* Samsung Galaxy Note 10.1.



Figura 30: Tela do Controle Remoto TV orientação paisagem no Samsung Galaxy S3

Fonte: Autoria própria



Figura 31: Telas de Controle Remoto *Blu-Ray* e Cenários de Iluminação no Samsung Galaxy S3

Fonte: Autoria própria

3.5 DESENVOLVIMENTO DO *HARDWARE IR* E *FIRMWARE*

Para o desenvolvimento do protótipo eletrônico do controle remoto IR foi utilizado o Kit Stellaris EKS-LM3S6965 da *Texas Instruments* que apresenta uma interface *Ethernet* para conexão com a rede doméstica. O kit também disponibiliza acesso a saídas PWM onde foi ligado o circuito de saída IR. A placa do kit é interligada, então, via *Ethernet* em um roteador *Wi-Fi* que forma uma rede *wireless* na qual



Figura 32: Tela do Controle Remoto *Home Theater* no *tablet* Samsung Galaxy Note 10.1

Fonte: Autoria própria

os *smartphones* ou *tablets* podem se conectar e através do aplicativo desenvolvido controlar os equipamentos do *Home Theater* e iluminação. A Figura 33 mostra os componentes envolvidos no sistema de controle da sala de *Home Theater*.

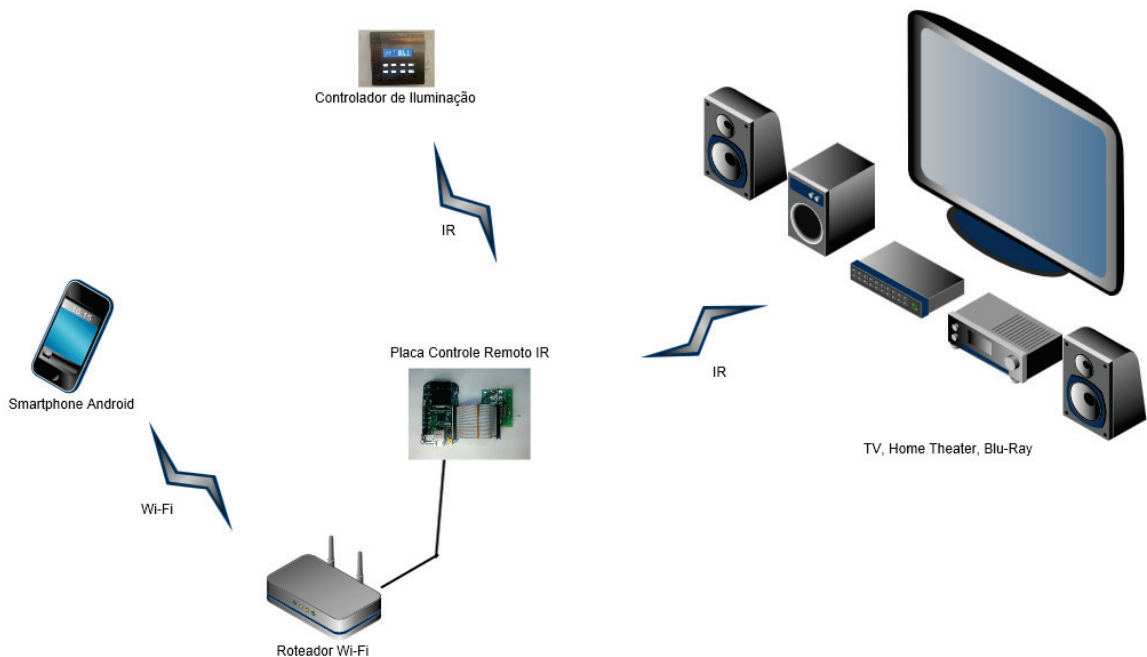


Figura 33: Diagrama do Sistema de Controle Remoto IR utilizando Smartphones Android

Fonte: Autoria própria utilizando *Gliffy on line*

O controlador de iluminação utilizado foi o da empresa Artiere de 8 cenas e 8 zonas adaptável numa caixa de interruptores 4x4. A grande vantagem deste

controlador é que as cenas programáveis podem ser acionadas por controle remoto infravermelho, sendo que bastou adaptar o controlador para enviar os sinais IR deste controlador de iluminação para ter o controle dos cenários de iluminação pelo sistema desenvolvido.

3.5.1 CIRCUITO DE SAÍDA IR

O circuito de saída IR consiste simplesmente de um transistor como *driver* para suportar uma corrente de pelo menos 100mA para o LED IR. A necessidade de uma corrente alta é importante para conseguir um maior alcance do IR. Esta corrente pode ser alta porque os pulsos do código são muitos rápidos, devendo-se cuidar apenas que não exceda o pico máximo de corrente suportado pelo LED IR (BERGMANS, 2011).

A Figura 34 mostra o esquema elétrico do circuito de saída implementado.

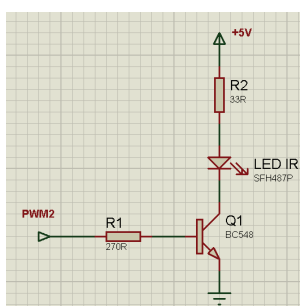


Figura 34: Circuito *driver* de saída IR

Fonte: Autoria própria

A Figura 35 mostra a foto do protótipo montado com o circuito de saída IR.

3.5.2 DESENVOLVIMENTO DO *FIRMWARE*

O *firmware* foi desenvolvido com o ambiente integrado de desenvolvimento (IDE) da IAR System em linguagem C e utilizando as bibliotecas da *Texas Instruments Stellaris Ware* para o ARM Cortex M3. Foi implementado um servidor web do tipo *GET* que obtém os parâmetros na própria *url*. Os parâmetros passados são dois *bytes* para endereçamento e um *byte* para o dado, conforme descrito a seguir:

- Eq_H ⇒ Equipamento parte Alta

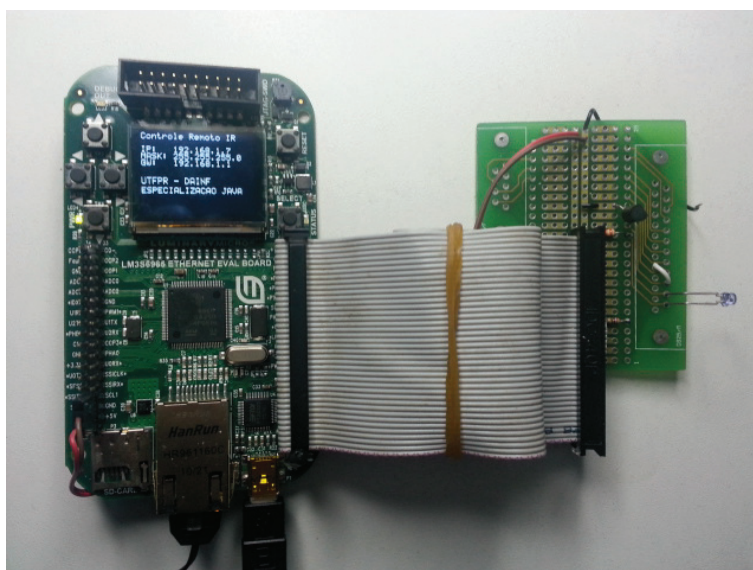


Figura 35: Protótipo Montado

Fonte: Autoria própria

- Eq_L \Rightarrow Equipamento parte Baixa
- DadoIR \Rightarrow Dado do IR (Infla-vermelho)

Apesar dos diversos protocolos estudados na pesquisa realizada conforme o item 2.2.3 que apresenta alguns exemplos de protocolos IR, foi identificado que somente o controlador de iluminação utiliza o protocolo conhecido Philips RC5. Os demais equipamentos da sala de *Home Theater* utilizados nos testes eram da marca *Samsung* que tem um protocolo próprio e que não foi encontrado nenhuma documentação sobre o mesmo. Foi feito, então, o levantamento do protocolo utilizando um osciloscópio eletrônico *Tektronix TDS2002C* e um sensor IR PHSC38 da *Photonic*. A Figura 36 mostra uma tela do osciloscópio com um sinal do controle remoto da *Samsung*.

O sinal IR analisado apresentou um sinal de *start* com *burst* de 4,5ms e pausa de 4,5ms seguidos de 32 sinais de bits, formando 4 bytes. Foram considerados o bit 0 com *burst* de 0,57ms e pausa de 0,57ms e o bit 1 com *burst* de 0,57ms e pausa de 1,71ms. Foi considerado também começando do bit menos significativo (LSB) para o mais significativo (MSB), ficando os dois primeiros bytes para o endereço do equipamentos e penúltimo para o dado do sinal IR e o último verificou-se que é o complemento do penúltimo byte. A Figura 37 mostra os sinais modulados de *Start*, bit 0 e bit 1.

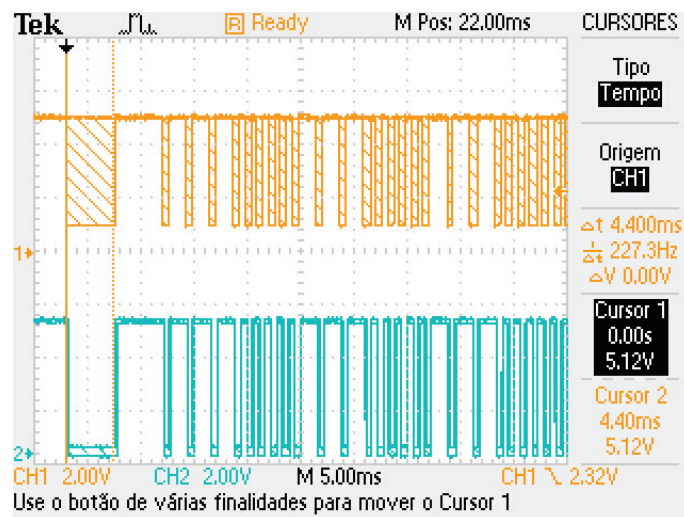


Figura 36: Sinal IR do Controle Remoto Samsung

Fonte: Autoria própria

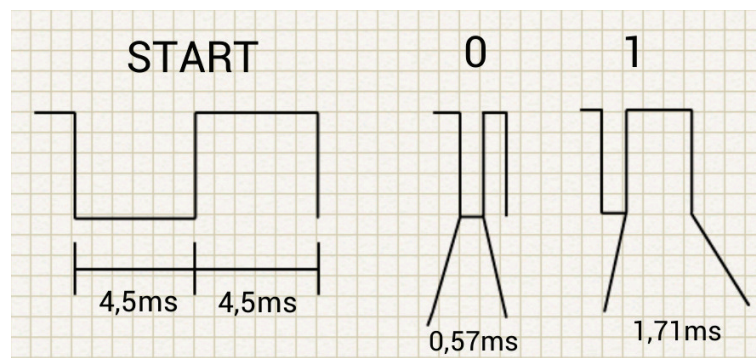


Figura 37: Sinal IR Modulado do Controle Remoto Samsung

Fonte: Autoria própria

O Quadro 1 mostra os códigos levantados para as principais teclas do controle para TV, *Home Theater*.

CONTROLE TV																
	EQ_H				EQ_L				DADO				COMPLEMENTA DADO			
	LSB			MSB	LSB			MSB	LSB			MSB	LSB		MSB	
POWER	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
SOURCE	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
VOL+	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
VOL-	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
CH+	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0
CH-	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0
ENTER	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	1
MUTE	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0
MTS	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
CC	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0
BAND (2)	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
CFTV (3)	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
SBT (4)	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
CNT (6)	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
RIC (7)	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
EPR (9)	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
GLOBO (1)	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
CONTROLE HOME THEATER																
POWER	1	1	0	0	0	0	1	0	1	1	0	0	1	0	1	0
SOURCE	1	1	0	0	0	0	1	0	1	1	0	0	1	0	1	0
VOL+	1	1	0	0	0	0	1	0	1	1	0	0	1	1	0	0
VOL-	1	1	0	0	0	0	1	0	1	1	0	0	1	1	0	0
TUN+	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
TUN-	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
EFFECT	1	1	0	0	0	0	1	0	1	1	0	0	1	1	1	0
MUTE	1	1	0	0	0	0	1	0	1	1	0	0	1	1	0	0
SUB	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
UP	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
DOWN	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
RIGHT	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
LEFT	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
ENTER	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
RETURN	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
EXIT	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
A	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
B	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
C	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
D	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
CONTROLE BLU-RAY																
AUDIO	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
LEGENDA	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
PREVIOUS	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
NEXT	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
BACKWARD	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
FORWARD	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
STOP	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
PLAY	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0
PAUSE	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	0

Quadro 1: Códigos do Controle Remoto Samsung

Fonte: Autoria própria

4 RESULTADOS

Para validação do protótipo, foi utilizado primeiramente o emulador Android SDK com a placa de controle com o LED IR apontada para uma TV Samsung. Após todos os comandos estarem funcionando perfeitamente pelo emulador. Foram instalados e testados seu funcionamento em um celular Samsung Galaxy S3 e num Tablet Samsung Galaxy Note 10.1. Todos os comandos funcionaram satisfatoriamente.

O equipamento foi instalado numa sala de *Home Theater* e todas as funcionalidades foram testadas, sendo comprovado o funcionamento do sistema.

O tempo de resposta deste tipo de solução depende da rede *Wi-Fi* e *Ethernet* em uso. Nos testes realizados, foi utilizado uma rede doméstica comum, com um roteador *wireless* da TP-Link TL-642G e um roteador Encore ENH908-NWY+. O tempo de resposta obtido não passou de 1 segundo, sendo bem satisfatório para operação como controle remoto.

O alcance do IR ficou em torno de 4 metros o que é bem razoável para as salas de *Home Theaters*, considerando que o equipamento normalmente seria instalado no centro da mesma.

Um problema que já era esperado, devido as características do IR, é que o equipamento é bem direcional, devendo o LED IR ficar apontado em direção aos sensores IR dos equipamentos.

5 CONCLUSÕES

Não há dúvidas de que com desenvolvimento dos dispositivos móveis, em especial dos *smartphones* e *tablets*, estão ocorrendo mudanças significativas em vários setores da vida humana. A comodidade da portabilidade e de estar o tempo todo “conectado” incentiva a criação de muitos aplicativos para todo os setores. Isto faz com que os *smartphones* e *tablets* sejam o sonho de consumo de muitas pessoas. A tendência de crescimento do mercado é promissora.

O desenvolvimento e implementação deste protótipo vem de encontro com esta tendência. Aproveitando que o *smartphone* já está a mão do usuário, controlar os equipamentos de uma sala de *Home Theater* com uma interface simples e intuitiva diretamente do *smartphone* proporciona satisfação e conforto ao usuário.

A escolha da plataforma Android foi motivada pelo seu grande crescimento e aceitação, por ser um sistema de código livre, com disponibilidade de ferramentas de desenvolvimento gratuitas, grande quantidade de documentação e exemplos de códigos. Porém, para atingir outros usuários de sistemas iOS e Windows Phone, é interessante e está previsto o desenvolvimento também para estas plataformas.

O protótipo desenvolvido atingiu os objetivos propostos, conforme os testes realizados, executou satisfatoriamente o controle dos equipamentos de uma sala de *Home Theater*. Uma desvantagem deste tipo de solução é o posicionamento da placa de controle que deve estar com o transmissor IR no campo de visão e apontado para os equipamentos. Isto será facilitado com o desenvolvimento futuro do equipamento com tecnologia sem fio *Bluetooth* e utilização de pilhas para poder posicionar o mesmo na melhor posição dentro da sala.

Através das pesquisas de produtos similares, identificou-se um bom nicho de mercado, onde o aplicativo é disponibilizado gratuitamente na *Google Play* como demonstração e para funcionalidade total há a necessidade de aquisição do dispositivo eletrônico.

Este trabalho foi útil para por em prática os diversos conhecimentos adquiridos no curso e demonstrou a viabilidade deste tipo de produto. Muito ainda tem que ser desenvolvido para que este trabalho possa virar um produto comercializável, porém espera-se que sirva também, como um incentivo para que outros desenvolvedores ponham em prática seus projetos.

5.1 TRABALHOS FUTUROS

- Desenvolver o *hardware* com tecnologia *Bluetooth*, IR e alimentado por pilhas para fazer a interface entre os dispositivos móveis com os equipamentos da sala de *Home Theater*.
- Desenvolver caixa para acomodar a placa de controle.
- Implementar o modo de aprendizagem dos códigos de controle remoto.
- Implementar telas de controle remoto configuráveis.
- Melhorar *design* das teclas e telas.
- Ofertar o aplicativo na *Play Store*
- Desenvolver o aplicativo para os sistemas iOS e Windows Phone.

REFERÊNCIAS

- BERGMANS, S. **IR Remote Control Theory**. 2011. Disponível em: <www.sbprojects.com/knowledge/ir/index.php>. Acesso em: 21 de maio de 2013.
- BOLZANI, C. A. M. **Residências Inteligentes**. [S.l.]: Editora Livraria da Física, 2004.
- DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**. [S.l.]: Pearson Prentice Hall, 2005.
- GOOGLE. **Get the Android SDK**. 2013. Disponível em: <<http://developer.android.com/sdk/index.html>>. Acesso em: 25 de maio de 2013.
- GRIFFIN. **Beacon - Dispositivo Bluetooth para IR**. 2013. Disponível em: <<http://store.griffintechology.com/smartphone/beacon-android>>. Acesso em: 27 de maio de 2013.
- ICONESBR. **A Melhor Coletânea de Ícones do Brasil**. 2013. Disponível em: <<http://www.iconesbr.net>>. Acesso em: 27 de junho de 2013.
- LARMAN, C. **Utilizando UML e Padrões: uma introdução a análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. [S.l.]: Artmed Editora S.A., 2007.
- LECHETA, R. R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. [S.l.]: Novatec Editora, 2010.
- LEE, V.; SCHNEIDER, H.; SCHELL, R. **Aplicações Móveis: Arquitetura, projeto e desenvolvimento**. [S.l.]: Pearson / Makron Books, 2005.
- LEE, W.-M. **Beginning Android Application Development**. [S.l.]: Wiley Publishing, 2011.
- OHA. **Open Handset Alliance**. 2007. Disponível em: <www.openhandsetalliance.com>. Acesso em: 22 de maio de 2013.
- PDWIN. **Blue IR User Guide**. [S.l.], 2012. Disponível em: <<http://www.pdwin.com/blir/BlueIR.pdf>>. Acesso em: 28 de maio de 2013.
- PIMENTEL, A. R. **Projeto de Software Usando a UML. UFPR**. Julho 2007.
- STADZISZ, P. C. **Projeto de Software usando a UML**. 2002. CEFET-PR.
- TECPAR, P. **Mercado de automação residencial dá sinais de crescimento**. 2011. Disponível em: <<http://portal.tecpar.br/index.php/pt/noticias/1781-mercado-de-automacao-residencial-da-sinais-de-crescimento>>. Acesso em: 25 de maio de 2013.

TEXAS. **User's Manual Stellaris LM3S6965 Evaluation Board**. [S.l.], 2010. Disponível em: <<http://www.ti.com/lit/ml/spmt127b/spmt127b.pdf>>. Acesso em: 26 de maio de 2013.

THINKFLOOD. **RedEye - Dispositivo Wi-Fi para IR**. 2013. Disponível em: <<https://thinkflood.com/products/redeye>>. Acesso em: 27 de maio de 2013.