|  |
| --- |
| FACULDADES METROPOLITANAS UNIDAS  DEPARTAMENTO DE ENGENHARIA ELÉTRICA  PROJETO DE GRADUAÇÃO  CAIO BARBOZA DIAS  FELIPE BARBOSA LOPES  FELIPPE OKA  AUTOMAÇÃO RESIDENCIAL:  controle remoto universal via aplicativo móvel |
| são paulo – Sp  NOVEMBRO/2016 |

|  |
| --- |
| CAIO BARBOZA DIAS  FELIPE BARBOSA LOPES  FELIPPE OKA  AUTOMAÇÃO RESIDENCIAL:  CONTROLE REMOTO UNIVERSAL VIA APLICATIVO MÓVEL  Monografia apresentada à Banca Examinadora do Centro Universitário das Faculdades Metropolitanas Unidas, como exigência parcial para a obtenção de título de Graduação em Engenharia Elétrica sob a orientação do Professor Alexandre Manente Pinto. |
| SÃO PAULO – SP  NOVEMBRO/2016 |

CAIO BARBOZA DIAS

FELIPE BARBOSA LOPES

FELIPPE OKA

AUTOMAÇÃO RESIDENCIAL:

CONTROLE REMOTO UNIVERSAL VIA APLICATIVO MÓVEL

Monografia apresentada à Banca Examinadora do Centro Universitário das Faculdades Metropolitanas Unidas, como exigência parcial para a obtenção de título de Bacharel em Engenharia Elétrica sob a orientação do Professor Alexandre Manente Pinto.

Data da aprovação:

\_\_/12/2016

Banca examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Alexandre Manente Pinto

FMU - Orientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr.

FMU

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr.

FMU

Dedicamos este trabalho a Sergio Oka, a todos os professores que nos ajudaram durante a trajetória do curso, especialmente ao professor Alexandre Manente por nos orientar e tornar possível a realização desse trabalho.

Agradecemos a Ricardo Alexandre Carmona por disponibilizar um conjunto de materiais referenciais que foram de grande ajuda. Agradecemos também aos nossos colegas de classe que nos ajudaram com sugestões e críticas sempre construtivas.

**RESUMO**

Este trabalho apresenta a aplicação da tecnologia Bluetooth presente em smartphones como ferramenta de interface do usuário com dispositivos eletrônicos, (e.g., televisores, aparelhos de som, tocadores de DVD), via Arduino. A proposta deste trabalho é apresentar um sistema que, utilizando-se de tecnologias atuais, possa trazer comodidade ao usuário final, uma vez que a grande maioria dos aparelhos eletrônicos modernos poderão ser controlados por esse protótipo, e também trazer à tona os benefícios da automação residencial e do conceito internet das coisas. Para isto será realizado a elaboração e implementação de um dispositivo de controle universal para aparelhos eletrônicos, comandado por smartphones. Como ferramenta de auxílio será utilizada a IDE Microsoft Visual Studio para desenvolvimento das aplicações do Arduino (C++) em conjunto com a IDE Apple Xcode para desenvolvimento do aplicativo iOS (Swift).

**Palavras-chave:** Controle remoto universal, internet das coisas, automação residencial.

**Abstract**

This paper shows an application for smartphone's embedded Bluetooth technology as an interfaced tool to modern electronic devices, (e.g., televisions, audio systems, DVD players), through Arduino. The article's main goal is to introduce an ecosystem which, using the current available technology, brings a lot of convenience to the end user, since that mostly of modern eletronic devices can be controlled through this prototype, as well as bring to light the benefits that home automation may provide. In order to do so, it'll be designed and built an universal control device for eletronic gadgets, managed by smartphones. As a supplementary tool, it’ll be used the Microsoft's Visual Studio IDE to develop the Arduino (C++) applications, simultaneously with Apple's Xcode IDE to develop the iOS application (Swift).

**Keywords:** Universal remote control, internet of things, home automation.

**LISTA DE FIGURAS**

[Figura 1.1: IDE Microsoft Visual Studio 21](#_Toc467003147)

[Figura 1.2: IDE Apple Xcode 22](#_Toc467003148)

[Figura 1.3: Tipos de comunicações 23](#_Toc467003149)

[Figura 1.4: Símbolo do Bluetooth 24](#_Toc467003150)

[Figura 1.5: Ondas eletromagnéticas 26](#_Toc467003151)

[Figura 1.6: Microcontrolador ATMEGA328P-PU 27](#_Toc467003152)

[Figura 1.7: Arduino Uno 30](#_Toc467003153)

[Figura 1.8: Shields do Arduino 31](#_Toc467003154)

[Figura 2.1: Lista de componentes do circuito 33](#_Toc467003155)

[Figura 2.2: Componentes do Arduino Uno 35](#_Toc467003156)

[Figura 2.3: Esquemático do circuito 36](#_Toc467003157)

[Figura 2.4: Ilustração da montagem do circuito 37](#_Toc467003158)

[Figura 2.5: Relação Core Bluetooth e protocolo BLE 38](#_Toc467003159)

[Figura 2.6: Representação da estrutura de objeto do periférico BLE 39](#_Toc467003160)

[Figura 2.7: Função de notificação e conexão com o periférico 39](#_Toc467003161)

[Figura 2.8: Fluxograma de navegação do aplicativo 41](#_Toc467003162)

[Figura 2.9: Tela inicial após conexão com módulo Bluetooth 42](#_Toc467003163)

[Figura 2.10: Tela de categorias 42](#_Toc467003164)

[Figura 2.11: Tela de fabricantes 43](#_Toc467003165)

[Figura 2.12: Tela de dispositivos 43](#_Toc467003166)

[Figura 2.13: Tela de envio de comandos 44](#_Toc467003167)

[Figura 2.14: Diagrama de objetos UML 44](#_Toc467003168)

[Figura 2.15: Exemplo de cadastro de nova categoria 45](#_Toc467003169)

[Figura 2.16: Seleção de botão sem comando para novo dispositivo 46](#_Toc467003170)

[Figura 2.17: Modo de recebimento de comando 47](#_Toc467003171)

[Figura 2.18: Envio das características do comando 48](#_Toc467003172)

[Figura 2.19: Recebimento das características do comando 48](#_Toc467003173)

[Figura 2.20: Diagrama de comunicação Bluetooth 49](#_Toc467003174)

[Figura 2.21: Código de leitura e interpretação de dados 50](#_Toc467003175)

[Figura 2.22: Diagrama de comunicação IR 51](#_Toc467003176)

[Figura 2.23: Código de captura do sinal IR e envio via serial 52](#_Toc467003177)

[Figura 3.1: Hardware final 53](#_Toc467003178)

**LISTA DE TABELAS**

[Tabela 1.1: Most Popular Coding Languages, 2015 18](#_Toc467002469)

[Tabela 1.2: Tabela comparativa 24](#_Toc467002470)

[Tabela 1.3: Subdivisões do infravermelho 26](#_Toc467002471)

[Tabela 1.4: Principais Fabricantes de Microcontroladores 28](#_Toc467002472)

[Tabela 2.1: Tabela de especificação técnica do Arduino Uno 33](#_Toc467002473)

**LISTA DE ABREVIATURAS E SIGLAS**

ABNT Associação Brasileira de Normas Técnicas

BEN Balanço Energético Nacional

EPE Empresa de Pesquisa Energética

IBGE Fundação Instituto Brasileiro de Geografia e Estatística

Mtep Milhões de toneladas equivalentes de petróleo

UFES Universidade Federal do Espírito Santo

**LISTA DE SÍMBOLOS**

vr(t) Tensão instantânea sobre o resistor (V)

t Tempo (s)

R Valor da resistência do resistor (Ω)

i(t) Corrente instantânea no elemento (A)

vl(t) Tensão instantânea sobre o indutor (V)

L Valor da indutância do indutor (H)

vc(t) Tensão instantânea sobre o capacitor (V)

C Valor da capacitância do capacitor (F)

**SUMÁRIO**

[INTRODUçÃO 14](#_Toc467001704)

[Objetivo Geral 15](#_Toc467001705)

[Objetivos Específicos 15](#_Toc467001706)

[1 FUNDAMENTOS TEÓRICOS RELACIONADOS À PESQUISA 17](#_Toc467001707)

[1.1 Tecnologias 17](#_Toc467001708)

[1.1.1 Domótica 17](#_Toc467001709)

[1.1.2 Linguagens de Programação 17](#_Toc467001710)

[1.1.2.1 C/C++ 18](#_Toc467001711)

[1.1.2.2 Swift 19](#_Toc467001712)

[1.1.3 Integrated Development Environment (IDE) 19](#_Toc467001713)

[1.1.3.1 Microsoft Visual Studio 19](#_Toc467001714)

[1.1.3.2 Apple XCode 20](#_Toc467001715)

[1.2 Dispositivos 21](#_Toc467001716)

[1.2.1 Transmissão de dados 21](#_Toc467001717)

[1.2.1.1 Comunicação Bluetooth 23](#_Toc467001718)

[1.2.1.2 Comunicação Infravermelho 24](#_Toc467001719)

[1.2.2 Microcontroladores 26](#_Toc467001720)

[1.2.3 Plataforma Arduino 29](#_Toc467001721)

[2 METODOLOGIA 31](#_Toc467001722)

[2.1 Hardware 31](#_Toc467001723)

[2.1.1 Componentes 31](#_Toc467001724)

[2.1.2 Módulo Bluetooth HM-10 32](#_Toc467001725)

[2.1.3 Arduino Uno 33](#_Toc467001726)

[2.1.4 Circuito 34](#_Toc467001727)

[2.2 Software 36](#_Toc467001728)

[2.2.1 iOS 36](#_Toc467001729)

[2.2.1.1 Conexão Bluetooth 37](#_Toc467001730)

[2.2.1.2 Navegação 39](#_Toc467001731)

[2.2.1.3 Extensão 44](#_Toc467001732)

[2.2.1.4 Comunicação dos comandos 46](#_Toc467001733)

[2.2.2 Arduino 48](#_Toc467001734)

[2.2.2.1 Comunicação Bluetooth 48](#_Toc467001735)

[2.2.2.2 Comunicação Infravermelho 50](#_Toc467001736)

[3 RESULTADOS 52](#_Toc467001737)

[3.1 Hardware 52](#_Toc467001738)

[3.2 Software 52](#_Toc467001739)

[4 COMENTÁRIOS E CONCLUSÕES 53](#_Toc467001740)

[5 PROJETOS FUTUROS 53](#_Toc467001741)

[6 REFERÊNCIAS BIBLIOGRÁFICAS 54](#_Toc467001742)

# INTRODUçÃO

Com a expansão e popularização dos conceitos de Automação Residencial, Domótica e Casas Inteligentes, a demanda por automação dos instrumentos domésticos e a centralização do controle dos processos de operacionalização destes instrumentos vem aumentando gradativamente. Um dos principais responsáveis por essa proliferação são os smartphones. Hoje, estima-se que o número de smartphones em uso no Brasil já ultrapassa à de PCs, com os valores de 154 e 152 milhões de unidades, respectivamente (GVcia, 2015). Com suas altas capacidades de processamento e suas tecnologias embarcadas,os smartphones estão convergindo para se tornarem cada vez mais a principal interface do usuário no gerenciamento destes processos, e dentre essas tecnologias embarcadas, a que mais vem se destacando para esta finalidade é o Bluetooth.

Com isto, este projeto consiste em criar um dispositivo central utilizando Arduino, uma plataforma física de computação de código aberto baseado numa simples placa microcontroladora e um ambiente de desenvolvimento para escrever o código para a placa. O Arduino pode ser usado para desenvolver objetos interativos, admitindo entradas de uma série de sensores ou chaves, e controlando uma variedade de luzes, motores ou outras saídas físicas. Projetos do Arduino podem ser independentes ou podem se comunicar com o software rodando em seu computador (como Flash, Processing, MaxMSP.). Os circuitos podem ser montados à mão ou comprados pré-montados; o software de programação de código-livre pode ser baixado da internet gratuitamente. Este será o intermédio do smartphone com o aparelho eletrônico que será controlado. Estes aparelhos poderão ser quaisquer dispositivos que possam ser remotamente controlados via IR (infravermelho), tais como televisores, aparelhos de som, DVD e Blueray players, ar condicionado, entre outros, sendo possível a expansão modular para diversos outros aparelhos e marcas.

## Objetivo Geral

O objetivo deste trabalho é apresentar um sistema que, utilizando-se de tecnologias atuais, possa trazer comodidade ao usuário final, tendo em vista que, como é apresentado em um estudo publicado pela The NextWeb em Janeiro de 2016, 68% dos americanos acreditam que as casas inteligentes serão tão comuns quanto os smartphones dentro de 10 anos. O custo de possuir uma casa é a maior despesa na vida de um proprietário. A habitação também consome a maior parte do orçamento de uma pessoa comum, respondendo por 33% das suas despesas anuais. Produtos para o lar inteligentes prometem economizar tempo, energia e dinheiro para os proprietários, com 45% dos usuários de produtos inteligentes dizendo que o uso destes produtos economizou US $ 1.100 por ano, e 87% dizendo que eles fizeram suas vidas mais fáceis (AURESIDE, 2016). Uma vez que a grande maioria dos aparelhos eletrônicos modernos poderá ser controlada por esse protótipo, apresentaremos um dos benefícios que a automação residencial pode trazer.

“A Automação Residencial tem mostrado que a integração de dispositivos eletroeletrônicos e eletromecânicos aumenta consideravelmente os benefícios se comparados com os sistemas isolados, de eficiência limitada. É também uma aliada na redução do consumo de recursos como água e energia elétrica, além de trazer maior conforto e segurança aos usuários.”

(BOLZANI, Caio. Integrador certificado pela Aureside, Associação Brasileira de Automação Residencial e responsável pelo LAR, o Laboratório de Automação Residencial da Poli -USP, em texto publicado no blog da Aureside no ano de 2015)

## Objetivos Específicos

Será desenvolvido tanto o hardware quanto o software do projeto. Na parte de hardware, será implementado um circuito composto em sua essência por: um Arduino, um LED IR emissor, um módulo Bluetooth. Já na parte de software, será desenvolvido um aplicativo mobile para a plataforma iOS, escrito na linguagem Swift, uma linguagem de programação consistente e intuitiva, desenvolvida pela Apple para a criação de apps para iOS, Mac, Apple TV e Apple Watch. Neste aplicativo mobile, o usuário seleciona o aparelho no qual deseja realizar o controle e sua respectiva marca/modelo, é então apresentado um layout tradicional de controles remotos no qual o usuário já está habituado, contendo botões virtuais como volume, canal, energia, entre outros. Ao executar um comando no aplicativo, o mesmo é enviado para o Arduino via Bluetooth, onde transmitirá a operação ao dispositivo selecionado através do LED IR, traduzindo-a para o protocolo esperado pelo dispositivo.

# FUNDAMENTOS TEÓRICOS RELACIONADOS À PESQUISA

## Tecnologias

### Domótica

Para (BOLZANI, 2004) a palavra domótica originou-se do latim domus que significa casa. É a técnica moderna de engenharia dos sistemas prediais e equipamentos a eles ligados. Com a utilização de novas tecnologias pretende-se criar sistemas de controle automático de uma residência ou edifício, automatizando os processos repetitivos, tornando as residências inteligentes.

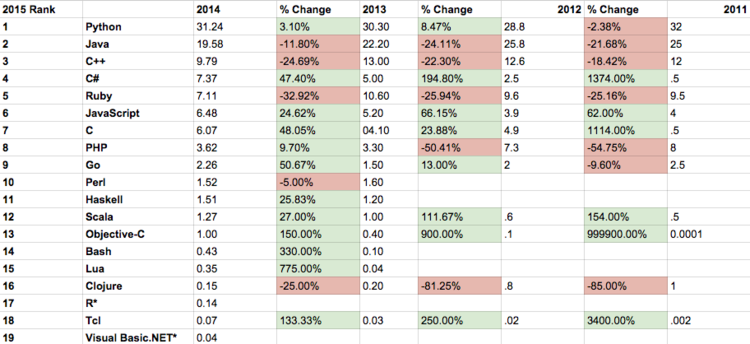
Ainda de acordo com (BOLZANI, 2004) os dispositivos inteligentes são a parte fundamental de uma residência inteligente. São equipamentos eletroeletrônicos que além de exercerem as funções para as quais foram concebidos, agregam hardware e software adicionais que lhes provém recursos extras, permitindo o controle e gerenciamento remoto e sua interconexão em rede.

### Linguagens de Programação

A linguagem de programação é uma forma ou meio de comunicação entre os humanos e os computadores. Conforme (GUDWIN, 1997), foram desenvolvidas, ao longo da história da computação, diversas “linguagens de programação”, cada qual, a seu tempo, introduzindo facilidades e recursos que foram tornando a tarefa de programar mais fácil e menos susceptível a erros. Atualmente, com as linguagens visuais (também chamadas por alguns de linguagens de quarta geração), programar deixou de ser uma arte restrita a um grupo de indivíduos, para tornar-se uma ferramenta a mais dentro do escopo do usuário comum.

Codeeval é uma plataforma online onde programadores demonstram suas habilidades em programação participando de competições e desafios. Esta plataforma publica anualmente uma pesquisa resultante de uma análise de mais de 600.000 programas utilizados nos testes, competições e desafios realizados por mais de 2.000 programadores chamada “**Most Popular Coding Languages”. A tabela a seguir demonstra em porcentagem os dados referentes ao ano de 2015.**

Tabela 1.1: Most Popular Coding Languages, 2015



Fonte: <http://blog.codeeval.com/2015#.U_aU9vldV9n>

#### C/C++

uma linguagem estr uturalmente simples e de grande por tabi-

lidade. Poucas s ˜

ao as arquiteturas de computadores para que um compi-

lador C n ˜

ao exista. Al ´

em disso, o compilador da linguagem gera c ´

odigos

mais enxutos e velozes do que muitas outras linguagens.

uma linguagem estr uturalmente simples e de grande por tabi-

lidade. Poucas s ˜

ao as arquiteturas de computadores para que um compi-

lador C n ˜

ao exista. Al ´

em disso, o compilador da linguagem gera c ´

odigos

mais enxutos e velozes do que muitas outras linguagens.

uma linguagem estr uturalmente simples e de grande por tabi-

lidade. Poucas s ˜

ao as arquiteturas de computadores para que um compi-

lador C n ˜

ao exista. Al ´

em disso, o compilador da linguagem gera c ´

odigos

mais enxutos e velozes do que muitas outras linguagens.

Segundo (BACKES, 2013) a linguagem C é uma linguagem estruturalmente simples e de grande portabilidade. Poucas são as arquiteturas de computadores para que um compilador C não exista. Além disso, o compilador da linguagem gera códigos mais enxutos e velozes do que muitas outras linguagens.

Mesmo softwares sendo escritos em outras linguagens de programação, eles dependem C ++ pois os compiladores mais populares são em C ++, como Visual Studio, e o compilador Intel. Este aspecto também está presente em softwares em execução no Windows que também são implementados em C ++ (como o pacote Office). A linguagem é onipresente; até mesmo o seu telefone celular e seu carro certamente contêm componentes programados em C ++, afirma (GOTTSCHLING, 2015).

Para (BACKES, 2013) a linguagem C é uma linguagem procedural, ou seja, ela permite que um problema complexo seja facilmente decomposto em módulos, onde cada módulo representa um problema mais simples. Além disso, ela fornece acesso de baixo nível à memória, o que permite o acesso e a programação direta do microprocessador.

A grande diferença entre as duas está nos paradigmas usados, enquanto que C é uma linguagem procedural, C++ é orientada a objetos (além de procedural). E devido à semelhança de sintaxes, se não forem usados objetos em um programa, um código C++ pode se tornar igual a um código C.

#### Swift

Swift é uma linguagem de programação gratuita e de em código aberto, consistente e intuitiva, desenvolvida pela Apple para a criação de apps para iOS, Mac, Apple TV e Apple Watch. Ela foi criada para dar ainda mais liberdade para os desenvolvedores. Swift é fácil de usar e em código aberto, para que qualquer pessoa com uma boa ideia consiga fazer coisas surpreendentes.

A Apple afirma em seu website que Apps criados com Swift são rápidos e dinâmicos. Um algoritmo comum de busca, por exemplo, obtém-se o resultado até 8,4x mais rápido que Python, a linguagem mais popular em 2015 segundo pesquisa realizada pela plataforma Codeeval apresentada no tópico 1.1.2 Linguagens de Programação.

### Integrated Development Environment (IDE)

De acordo com (MARCONI; HARADA, 2014) o ambiente de desenvolvimento integrado é um editor de texto customizado para escrita de código fonte e possui ferramentas de depuração de erros, ferramentas de ajuda sobre a linguagem, um compilador e outras ferramentas úteis para a escrita de um programa (e.g., editor de texto adaptado para a linguagem de programação, ferramenta de depuração de programas que auxilia na identificação e remoção de erros de programação, um conjunto de bibliotecas com código que pode ser reaproveitado e a documentação da linguagem).

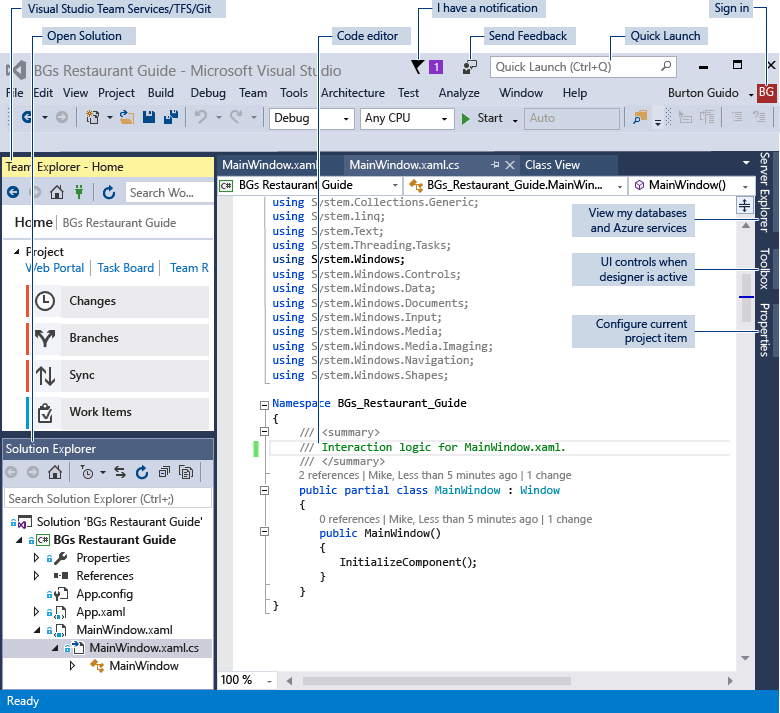
#### Microsoft Visual Studio

A IDE Visual Studio foi utilizada para o desenvolvimento do programa do Arduino Uno. Segundo sua desenvolvedora Microsoft, Microsoft Visual Studio é um conjunto de ferramentas para a criação de software, desde a fase de planejamento até o design de interface com o usuário, codificação, testes, depuração, analise de qualidade do código e desempenho, a implantação para os clientes, e coleta de telemetria no uso. Todas estas ferramentas estão presentes no Ambiente Desenvolvimento Integrado (IDE) Visual Studio.

Você pode usar o Visual Studio para criar vários tipos de aplicações, desde simples aplicativos e jogos para smartphone, até grandes e complexos sistemas que sustentam empresas e centros de dados. Por padrão, Visual Studio, fornece suporte para as linguagens C #, C e C ++, JavaScript, F # e Visual Basic.

A figura a seguir exemplifica a interface do programa:

Figura .: IDE Microsoft Visual Studio



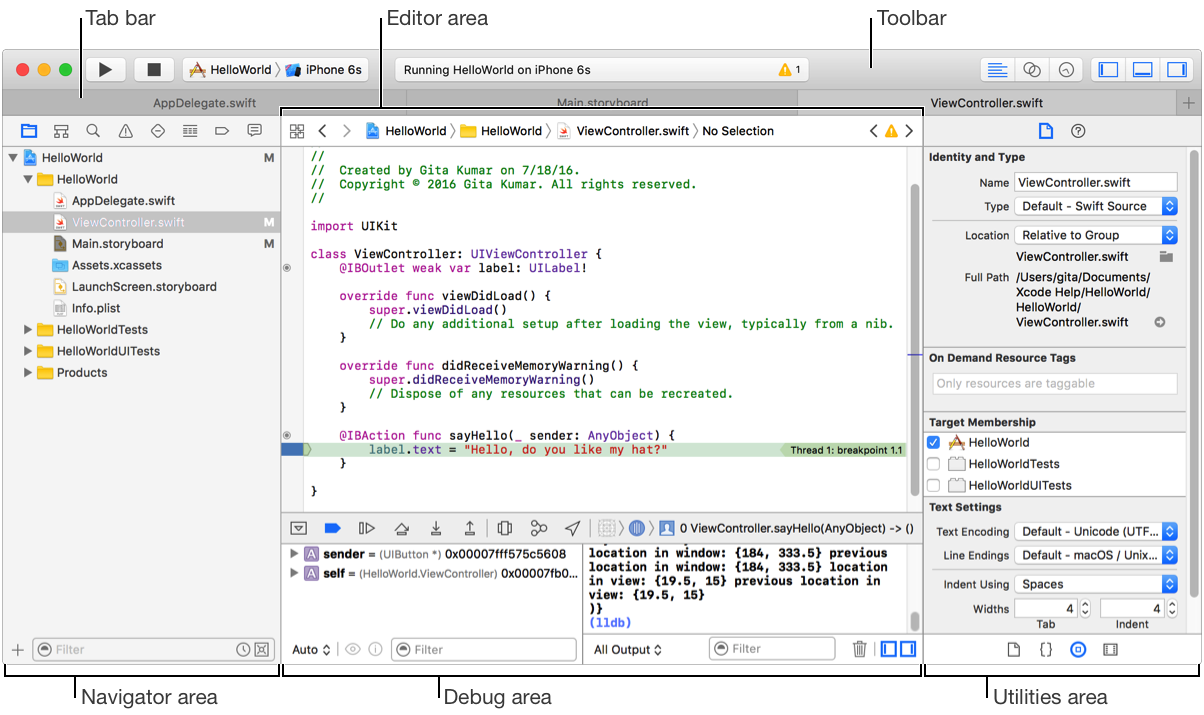
Fonte: <https://msdn.microsoft.com/en-us/library/dn762121.aspx>

#### Apple XCode

A IDE Apple XCode fui utilizada para o desenvolvimento da aplicação para smartphone do protótipo. De acordo com a desenvolvedora Apple, Xcode é o ambiente de desenvolvimento integrado (IDE) utilizado para construir aplicativos para os produtos da Apple, incluindo iPad, iPhone, Apple Watch, Apple TV e Mac. Xcode fornece ferramentas para gerenciar todo o seu fluxo de trabalho de desenvolvimento a criação de seu aplicativo para testes, otimização e enviá-lo para a App Store.

A figura a seguir exemplifica a interface do programa:

Figura .: IDE Apple Xcode



Fonte: <http://help.apple.com/xcode/mac/8.0/#/devc8c2a6be1>

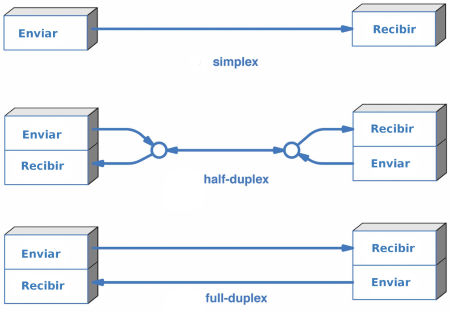
## Dispositivos

### Transmissão de dados

Para levar a informação sob formas de sinais de ondas eletromagnéticas de um dispositivo a outro precisamos de um meio de transmissão, seja ele guiado (cabo coaxial, fibra óptica, par de cobre trançado) ou não guiado (vácuo, ar). Para (STARLIN, 2007) pode haver conectividade ponto-a-ponto ligação entre dois dispositivos ou multipontos (meio partilhado por mais de dois dispositivos). E há também os tipos de comunicações:

* Simplex – comunicação unidirecional (TV)
* Half-duplex – comunicação bidirecional alternada (walkie-talkie)
* Full-duplex – comunicação bidirecional simultânea (telefone)

Figura .: Tipos de comunicações



Fonte: <http://www.twalves.com/e-fundamental/9-ano/modos-de-transmissao-de-dados>

Segundo (FOROUZAN, 2006), as ondas de rádio podem ser omnidirecionais, ou seja, quando o transmissor (antena) envia um sinal, ele se propaga por todas as direções. Então, o transmissor e o receptor não precisam estar alinhados para haver a troca de informações e isso é uma grande vantagem, em contrapartida o sinal fica suscetível às interferências, caso outra antena propague ondas na mesma frequência ou banda.

As micro-ondas são sempre unidirecionais. Para haver a troca de informação as antenas precisam estar focadas, alinhadas para que a comunicação aconteça. A vantagem é que apenas o par de antenas que estiverem alinhados irão poder se comunicar e também não há risco de interferências eletromagnéticas em outras antenas. A desvantagem dela é que se as torres estiverem em uma distância longa, deverá ter o cuidado de não ter obstáculos entre elas. Esse problema normalmente é resolvido colocando as torres em lugares altos. Outro problema comum é por conta da curvatura do planeta, que para ser contornado será necessário colocar repetidores de sinal ou regenerar a intensidade do sinal.

#### Comunicação Bluetooth

Utilizando a tecnologia “transmissão por salto de frequência”, que permite uma conexão segura e a prova de interferências. Usa-se o mesmo padrão de comunicação para acesso de informações em redes Lan e acesso à internet. Podendo ter alcance variado, dependendo da classe de rádio empregada:

* Classe 1: 100 metros (utilizado em industrias);
* Classe 2: 10 metros (utilizado em dispositivos móveis);
* Classe 3: 1 metro (utilizados em dispositivos móveis também.

Figura .: Símbolo do Bluetooth



Fonte: <https://donjajo.com/bluetooth-fix-a2dp-source-profile-connect-failed-xx-protocol-not-available-linux/#.WBWH5PkrLIU>

Conforme (FIGUEIREDO; NAKAMURA, 2015) opera na faixa de frequência de 2,4GHz, na banda ISM (*Industrial Scientific and Medical*), no modo “*full duplex*” com taxa de transferência de até 1Mb/s com baixo consumo (10 a 100mW) disponível mundialmente.

Essa conexão forma redes denominadas Piconet, que suportam até oito dispositivos, sendo eles: um “mestre” e 7 “escravos” podendo se comunicar entre si. Para um processo mais seguro, pode-se fazer o uso de senhas ou novas camadas de segurança por software.

Segundo (KOBAYASHI, 2004) existem diversas aplicações, dentre elas:

* Aplicações Pessoais (transferência de arquivos, domótica);
* Aplicações Corporativas (automação industrial);
* Aplicações Médicas (monitoramento de pacientes, banco de dados);
* Entretenimento (jogos).

Como entrou em diversos campos, o Bluetooth está em constante evolução. E com isso vem sendo aprimorado a cada versão. Todas as versões são compatíveis entre si, porém há uma limitação de velocidade e funções da versão mais antiga.

Segue uma tabela comparativa abaixo:

Tabela .: Tabela comparativa

|  |  |
| --- | --- |
| **Versão** | **Descritivo** |
| Bluetooth 1.0 | A versão 1.0 se trata das primeiras especificações do protocolo |
| Bluetooth 1.1 | Lançada em fevereiro de 2001, estabelecendo o Bluetooth como um padrão de conexão mundial, responsável também, por corrigir problemas da anterior |
| Bluetooth 1.2 | Lançada em novembro de 2003 e possibilitou o aumento das conexões e da segurança, além do processamento de voz avançado |
| Bluetooth 2.0 | Lançada em novembro de 2004. Essa versão diminuiu o consumo de energia, aumentou a velocidade de transmissão de dados para até 3 MB |
| Bluetooth 2.1 | Lançada em agosto de 2007, apresentou melhorias na segurança (inclusive nos recursos de criptografia) e na conexão com outros dispositivos |
| Bluetooth 3.0 | Lançada em abril de 2009, tem como grande atrativo as altas taxas de transferência de dados, atingindo até 24 Mbps |
| Bluetooth 4.0 | Lançada em 2010, exige muito menos energia, tornando-se ideal para tablets e smartphone |

Fonte: <http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2011/03/um-pequeno-guia-sobre-o-bluetooth-seu-uso-e-funcionamento.html>

A última versão é a 4.0 e segundo a Bluetooth Special Interest Group (SIG), a nova versão adiciona uma especificação de baixo consumo de energia, para transmitir pequenos volume de dados a pequenas distâncias, tudo alta velocidade, que já estavam introduzidas no Bluetooth 3.0, superando até redes Wi-Fi 802.11 para a transmissão de dados a uma velocidade de até 25 MB/s.

#### Comunicação Infravermelho

Segundo (VANZETTI, 1972) a energia infravermelha é uma luz invisível, porque o seu comprimento de onda é muito longo para ser visto. É a parte do espectro eletromagnético percebido pelo calor. A radiação infravermelha causa vibração de átomos ou grupos de átomos em um composto e estas vibrações podem ter velocidades e amplitudes diferentes. Sendo compreendidas entre 300GHz até 400THz, com comprimentos de onda de 770nm a 1mm.

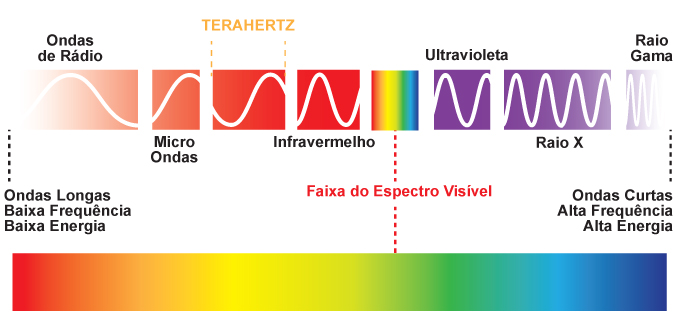
De acordo com VANZETTI (1972, p. 30):

“Quando a energia infravermelha colide com algum material físico, a porção que é absorvida pelo corpo produz um efeito primário: Aumenta a energia de partículas atômicas e subatômicas da qual a matéria física é formada, o que podemos detectar como o fenômeno de aumento de temperatura. Essa mudança pode ser mensurada por um dos efeitos secundários, como a variação de características físicas (volume, pressão, etc.), ou características químicas (como em fotografia infravermelha), ou em propriedades elétricas (condutividade, coeficiente dielétrico, emissão secundária, etc.). A velocidade com que esses efeitos secundários se manifestam pode ser de algumas horas, necessárias para uma reação química ou de frações de nano segundos para uma segunda emissão”.

Para a transmissão de dados, é utilizada frequências muito altas. Como a luz, o sinal infravermelho não pode penetrar em objetos, logo as transmissões por infravermelho são diretas ou difusas.

Na figura 1 temos as ondas eletromagnéticas:

Figura .: Ondas eletromagnéticas



Fonte: <http://www.dreaminc.com.br/sala_de_aula/infravermelho/>

Na tabela 1-3, temos as subdivisões do infravermelho com seus respectivos comprimentos de onda:

Tabela .: Subdivisões do infravermelho

|  |  |  |
| --- | --- | --- |
| **Designação** | **Abreviação** | **Comprimento de onda (µm)** |
| Infravermelho próximo | NIR | 0,75 - 3 |
| Infravermelho médio | MIR | 3 - 6 |
| Infravermelho distante | FIR | 6 - 15 |
| Infravermelho extremo | XIR | 15 - 1000 |

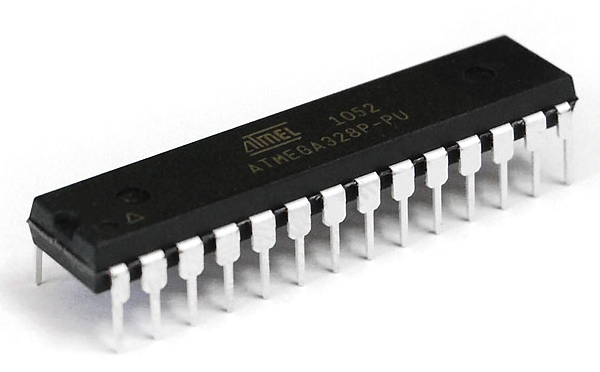
Fonte: <http://www.las.inpe.br/~cesar/Infrared/espectro.htm>

### Microcontroladores

Segundo (GIMENEZ, 2010), um microcontrolador é um componente eletrônico com a capacidade de ser programado, sendo que tem um sistema completo inserido num único circuito integrado. Que contém CPU (*Central Processing Unit*), memória de dados RAM (*Random Access Memory*) e memória ROM (*Read Only Memory*) para o controle de processos lógicos. Que leva o controle de periféricos (I/O), como led, botões, display, relés, conversores USB, conversores ETHERNET, sensores (pressão, temperatura, etc.), resistências, etc.

O consumo é muito baixo, por volta de *miliwatts*, por conta da baixa frequência de *clock*, alguns MHz, se comparado com microprocessadores atuais. São componentes de baixo custo e alta versatilidade, com isso, vem sendo utilizado em larga escala há décadas.

Figura .: Microcontrolador ATMEGA328P-PU



Fonte: <http://ktechnics.com/shop/atmega328p-pu/>

Segundo os professores Nobuo Oki e Suely Cunha Amaro Mantovani, da Faculdade de Engenharia de Ilha Solteira, os primeiros Microcontroladores foram baseados em uma tecnologia de hardware com grande limitação de memória e processamento. Na segunda fase da história, os componentes já apresentam uma complexidade bem alta, em relação a anterior. Com displays a interatividade se torna bem amigável e a introdução de interfaces seriais de comunicação abrem um leque maior de possibilidades. E com o avanço tecnológico, o custo diminui.

Na terceira fase e última, os Microcontroladores tomam características de hardware e software de alta capacidade, com 32 e 64 bits, com uma gama de recursos para processamento digital de sinais e outras atividades.

A seguir, na tabela 1.4 apresentamos algumas das várias famílias dos principais fabricantes:

Tabela .: Principais Fabricantes de Microcontroladores

Fonte: Revista Elektor (02/2016)

A arquitetura de um microcontrolador, normalmente consiste em periféricos, barramentos e um núcleo de processamento:

* Periféricos: caracterizam o conjunto de funcionalidades disponíveis pelo microcontrolador e são controlados pelo processador. Por exemplo, memória, porta serial, porta paralela e conversor A/D;
* Barramento: é dividido em dados e endereços, consiste nas linhas de comunicação entre o processador e os periféricos;
* Núcleo de processamento: consiste no processador de dados e na administração dos periféricos.

Na seção 2.1.3 está detalhado o microcontrolador utilizado no Arduino Uno, o ATmega328P-PU.

### Plataforma Arduino

O autor (MCROBERTS, 2011) afirma que:

“Um Arduino e um microcontrolador de placa única e um conjunto de software para programa-lo. O hardware consiste em um projeto simples de hardware livre para o controlador, com um processador Atmel AVR e suporte embutido de entrada/saída. O software consiste de uma linguagem de programação padrão e do bootloader que roda na placa. ”

De acordo com (MCROBERTS, 2011), o Arduino é um pequeno computador, no qual, com entradas e saídas (I/O) se comunica com componentes externos, podendo fazer a programação para processar.

O Arduino tem sido muito utilizado para pequenos projetos, na grande maioria, para desenvolver objetos interativos independentes ou até envio de informações se conectado a algum meio de comunicação, como infravermelho.

Figura .: Arduino Uno

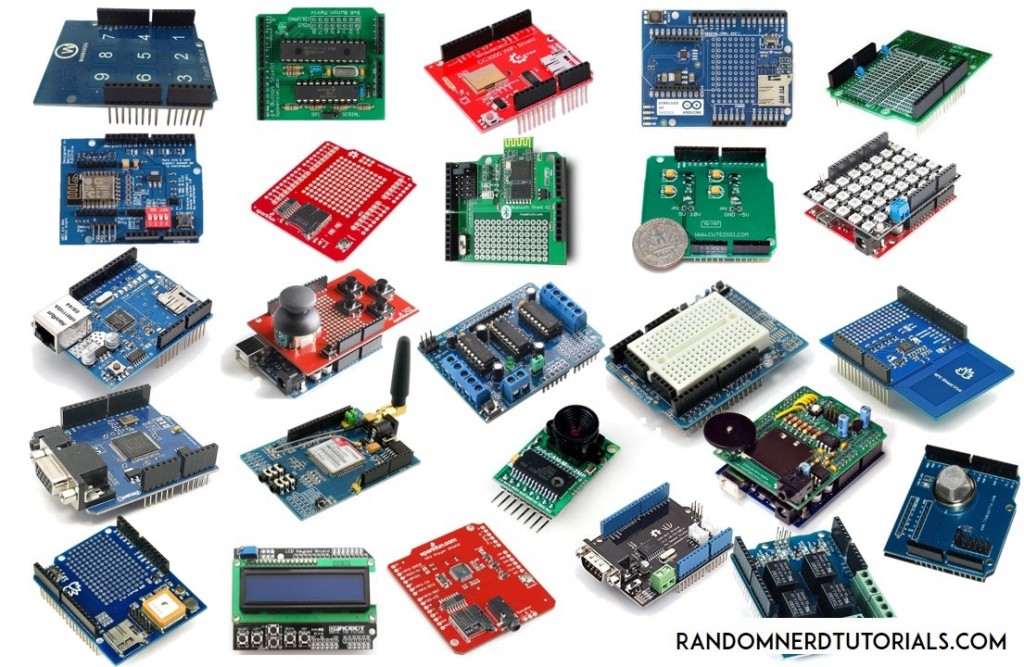


Fonte: <http://www.arduino.org/products/boards/arduino-uno>

Ele pode ser conectado à inúmeros periféricos, como LEDs, displays, botões, motores, interruptores, sensores (distância, pressão, temperatura), receptor GPS, módulo Ethernet, Shields (placas de circuito contendo outros dispositivos) e outros dispositivos que possa ser controlado ou receba e envie informação.

Para fazer a programação é necessário do IDE do Arduino, um software livre que compreende a linguagem baseada em C. Permitindo que escreva as instruções e através de um cabo, permitindo o *upload* do programa para o Arduino.

Figura .: Shields do Arduino



Fonte: http://www.dexterindustries.com/shop/arduberry-raspberry-pi-arduino-shields/

# METODOLOGIA

## Hardware

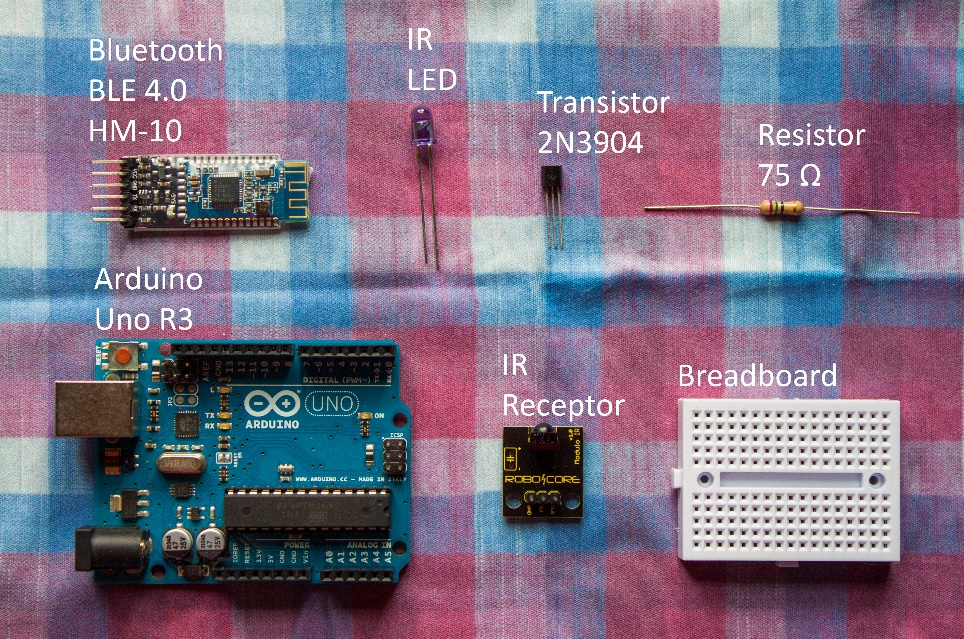
Silva Filho e Siqueira Filho (2006) definem há hardware como sendo a parte física e mecânica do computador, tais como suas peças e componentes eletrônicos. Esta seção trata da elaboração do protótipo e montagem do circuito eletrônico, bem como seus respectivos componentes.

### Componentes

Para o desenvolvimento do protótipo, foi implementado um circuito composto pelos seguintes componentes:

* LED (Light Emmiting Diode) IR (infra-red): responsável por propagar o comando ao dispositivo através de luz infravermelha;
* Transistor NPN 2N3904: responsável por amplificar a corrente a ser fornecida ao LED IR, a fim de ampliar o alcance do sinal;
* Módulo Bluetooth BLE 4.0 HM-10: responsável por realizar a comunicação entre o smartphone e o Arduino;
* Receptor IR 38kHz: responsável por coletar sinal infravermelho de controles remotos;
* Resistor: responsável por controlar a corrente do circuito;
* Arduino Uno: responsável por interpretar os comandos recebidos pelo módulo Bluetooth e reproduzi-los através de sinal infravermelho;

Figura .: Lista de componentes do circuito



### Módulo Bluetooth HM-10

De acordo com o Dasheet do produto disponibilizado pelo fabricante JNHuamao Technology Company, O **Módulo Bluetooth 4.0 HM-10 Master/Slave** é um módulo bluetooth que tem como peça principal o chipset CC2540/1 que opera nos modos master e slave, o modo é configurado através de comando AT. Para utilização do **Módulo Bluetooth 4.0 HM-10 Master/Slave, diz o fabricante em suas especificações (**JNHuamao Technology Company **Datasheet, 2014)** basta conectar os pinos TX e RX do módulo bluetooth ao microcontrolador para fazer comunicação serial. É importante ressaltar que o módulo possui configuração padrão de fábrica, tais como velocidade de comunicação, nome do dispositivo, senha de pareamento etc. No entanto, estes parâmetros poderão ser alterados por comando AT (comunicação serial com o computador).

Características importantes:

* Bluetooth V4.0 BLE
* Banda: 2.4GHz ISM
* Tensão e corrente de trabalho: 3.3/5V 50mA
* Longo alcance: Até 100 metros em campo aberto
* Temperatura de trabalho: -5 ~ +65 Graus Centígrados

De acordo com o fabricante do chipset CC2540/1 (TEXAS INSTRUMENTS, 2.4-GHz Bluetooth® low energy System-on-Chip, 2012) o baixo consumo de energia e o alto custo benefício em relação ao seu desempenho permite que sistemas Bluetooth de baixa potência mestre/escravo sejam construídos com um custo extremamente baixo.

### Arduino Uno

A versão do Arduino utilizada no projeto é o Arduino UNO, e segundo (MCROBERTS, 2011) difere das versões prévias por não utilizar o chip FTDI, que conduz a USB para a serial. Em vez disso, ela utiliza um Atmega8U2, programado como um conversor USB para serial. Isso confere à placa muitas vantagens quando comparada à sua predecessora, a Duemilanove. Primeiro, o chip Atmega é muito mais barato que o chip FTDI, diminuindo o preço das placas. Segundo, e mais importante, ele permite que o chip USB tenha seu firmware atualizado, para que o Arduino seja exibido em seu PC como outro dispositivo, tal como um mouse ou joystick de jogos. Isso abre uma série de novas possibilidades para o Arduino.

A tabela a seguir, retirada do site oficial do Arduino, demonstra as especificações técnicas do Arduino Uno. Nela há parâmetros que serão utilizados como base de cálculo para definição de parâmetros como, por exemplo, a resistência R1 representada na figura 3.2.

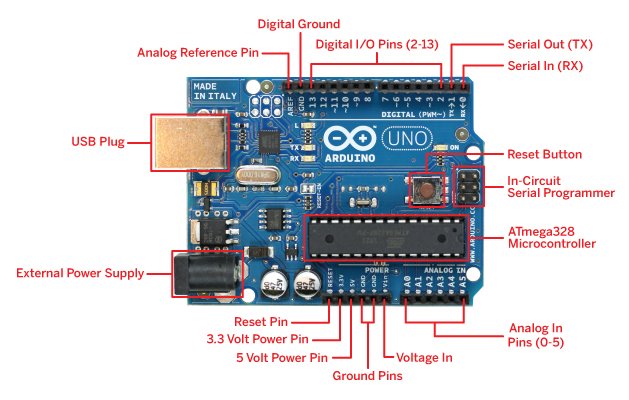
Tabela .: Tabela de especificação técnica do Arduino Uno

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](http://www.atmel.com/Images/doc8161.pdf) |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| LED\_BUILTIN | 13 |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

Fonte: [www.arduino.cc/en/Main/ArduinoBoardUno](http://www.arduino.cc/en/Main/ArduinoBoardUno)

A figura a seguir demonstra a posição física de cada componente da placa:

Figura .: Componentes do Arduino Uno

Fonte: <https://www.robomart.com/arduino-uno-online-india>

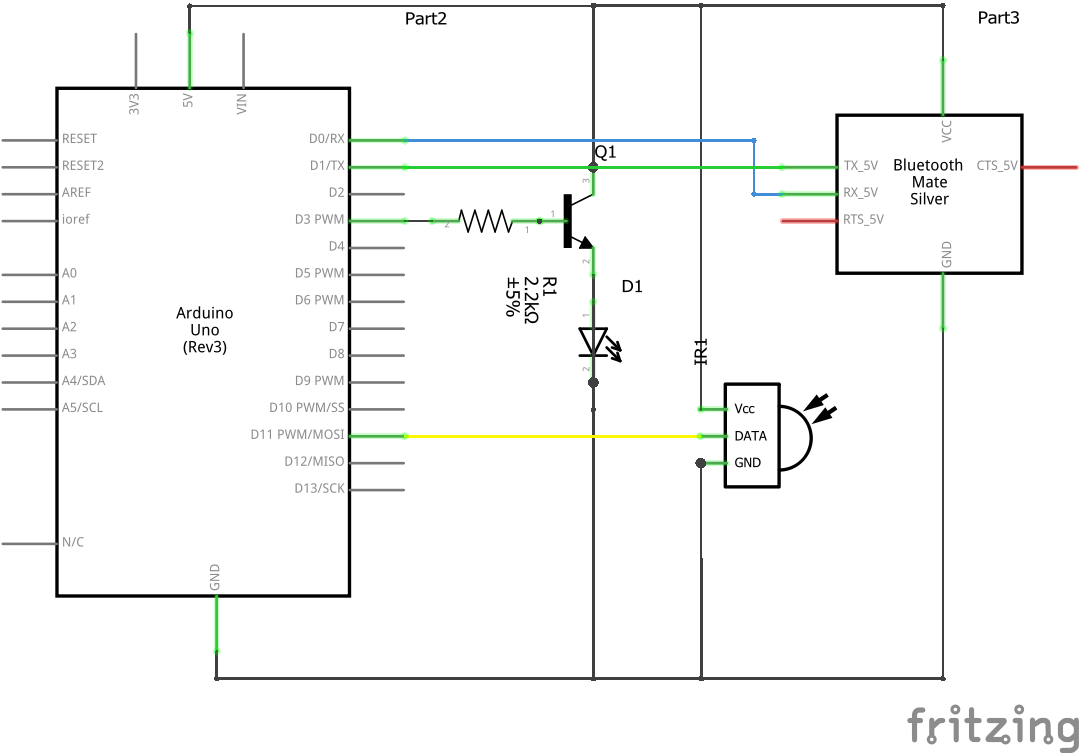
### Circuito

O circuito foi idealizado visando a simplicidade de sua reprodução, permitindo que qualquer entusiasta do assunto possa implementa-lo sem grandes custos.

A alimentação do circuito é fornecida pela porta de 5V do Arduino, que por sua vez é alimentado por uma bateria de 9V.

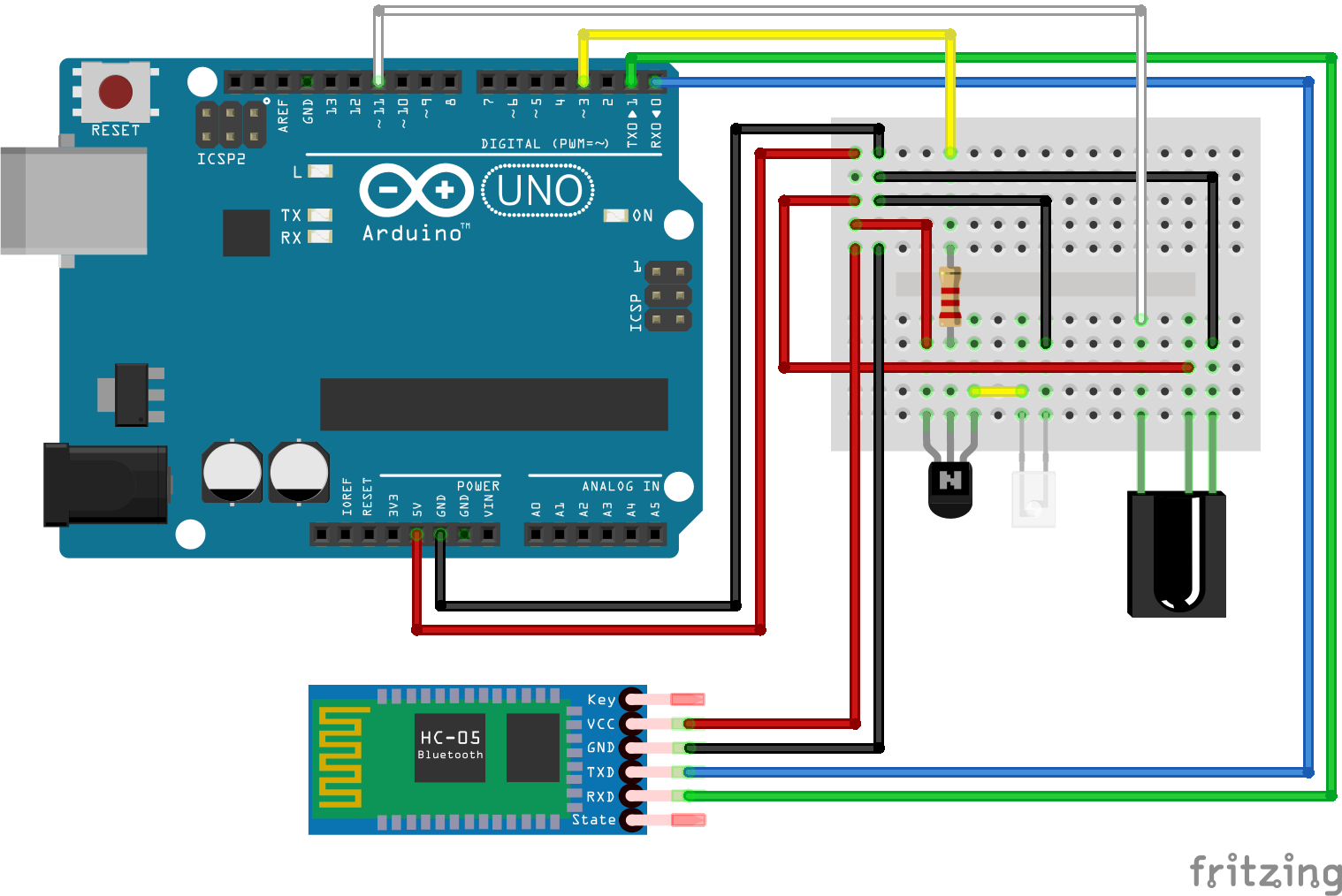
O esquemático do circuito é ilustrado na Figura 2.3.

Figura .: Esquemático do circuito



Para melhor exemplificação da montagem do circuito, a Figura 2.4 ilustra sua implementação na breadboard.

Figura .: Ilustração da montagem do circuito



A corrente máxima de saída de uma porta digital do Arduino é de

## Software

De acordo com Silva Filho e Siqueira Filho (2006), software consiste de toda a parte lógica do computador, sendo essa um conjunto de instruções não-ambíguas e relacionadas cujo especificam a realização de tarefas determinadas.

### iOS

Para o desenvolvimento da aplicação iOS, foi utilizado o ambiente de desenvolvimento integrado (IDE – Integrated Development Environment) Xcode 8. O uso desta ferramenta é obrigatório para desenvolvimento de aplicativos para o sistema operacional iOS.

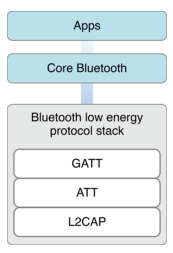
O software proposto no projeto foi implementado sobre a versão 10 do iOS e escrito na versão 3 da linguagem Swift, sendo as últimas versões de ambas as tecnologias no momento da elaboração do projeto.

O aplicativo tem por objetivo servir de repositório dos dispositivos e códigos cadastrados, bem como orquestrador da comunicação Bluetooth e transmissor do comando IR.

#### Conexão Bluetooth

A conexão com o módulo Bluetooth é feita através da implementação do protocolo BLE (Bluetooth Low Energy) 4.0, disponível através do Framework (conjunto de código agrupado e segregado à fim de cumprir uma funcionalidade específica) Apple Core Bluetooth. Este Framework tem por objetivo abstrair a camada de baixo nível de comunicação com o hardware, disponibilizando-a através de uma API (Interface de programação de aplicativos) de alto nível.

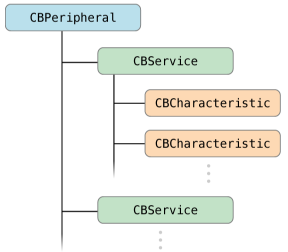
Figura .: Relação Core Bluetooth e protocolo BLE



Fonte: <http://apple.co/2g85rNP>

A conexão se inicia com o aplicativo buscando por periféricos BLE disponíveis ao seu alcance. Todos os dispositivos aderentes ao protocolo respondem a requisição. Cada periférico possuirá um ou mais serviços (funcionalidades específicas) e cada serviço uma ou mais características (forma de interação com o serviço).

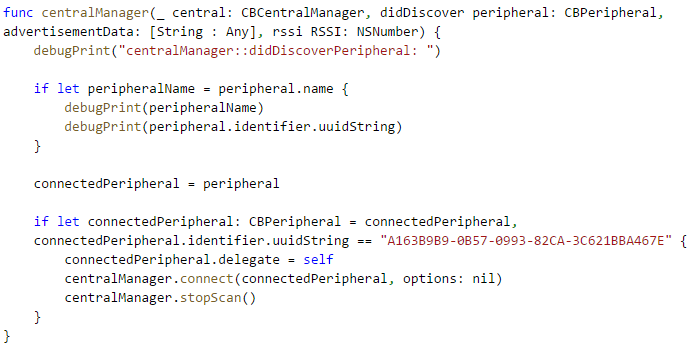
Figura .: Representação da estrutura de objeto do periférico BLE



Fonte: <http://apple.co/2fNyQbZ>

Para cada periférico retornado, é disparado uma função de notificação contendo seus respectivos dados. Caso o ID do periférico seja o mesmo do módulo Bluetooth do circuito, é solicitada a conexão com o periférico e a parada de busca de novos. No módulo utilizado neste projeto o ID esperado é “A163B9B9-0B57-0993-82CA-3C621BBA467E”.

Figura .: Função de notificação e conexão com o periférico



#### Navegação

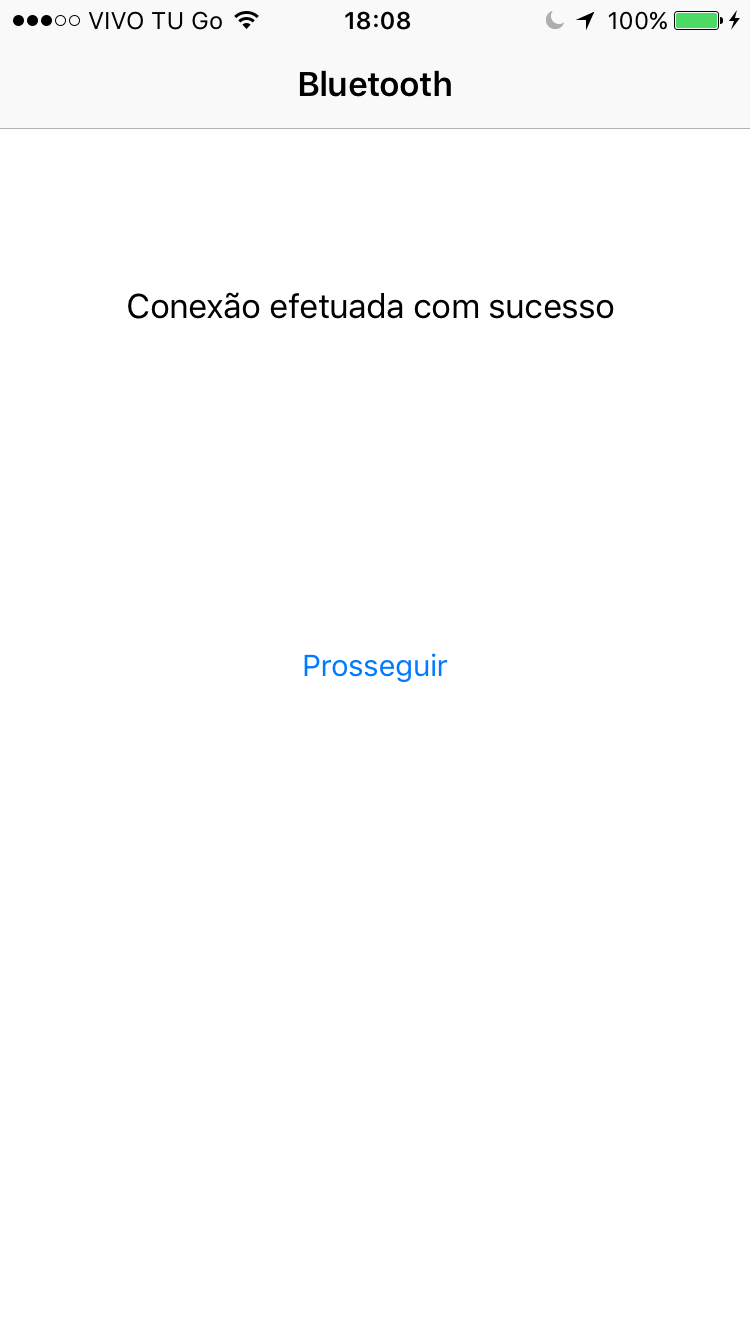
A aplicação consiste de um fluxo de navegação marjoritariamente direto, contendo apenas alguns possíveis desvíos para inclusão de novos dados. Abaixo o fluxograma de ilustração do fluxo de navegação:

Figura .: Fluxograma de navegação do aplicativo



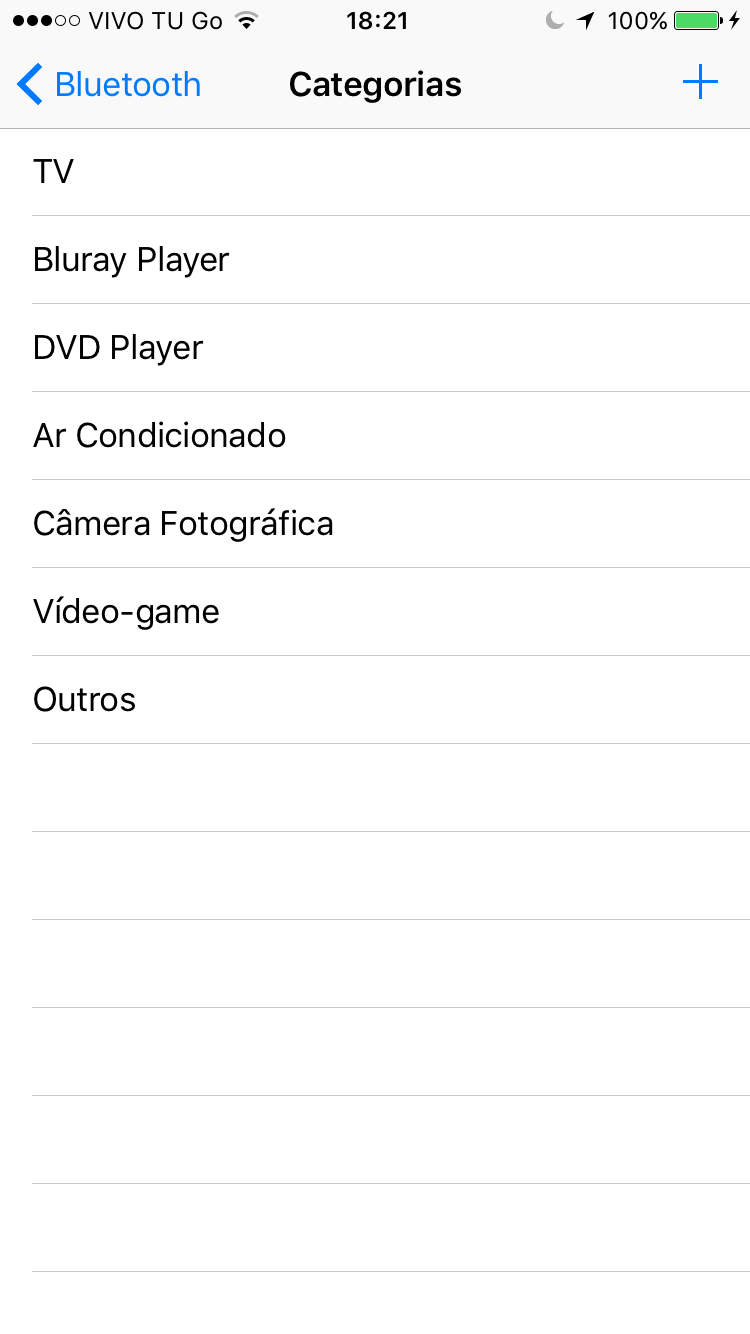
Após a conexão com o módulo Bluetooth, a navegação no aplicativo é liberada:

Figura .: Tela inicial após conexão com módulo Bluetooth



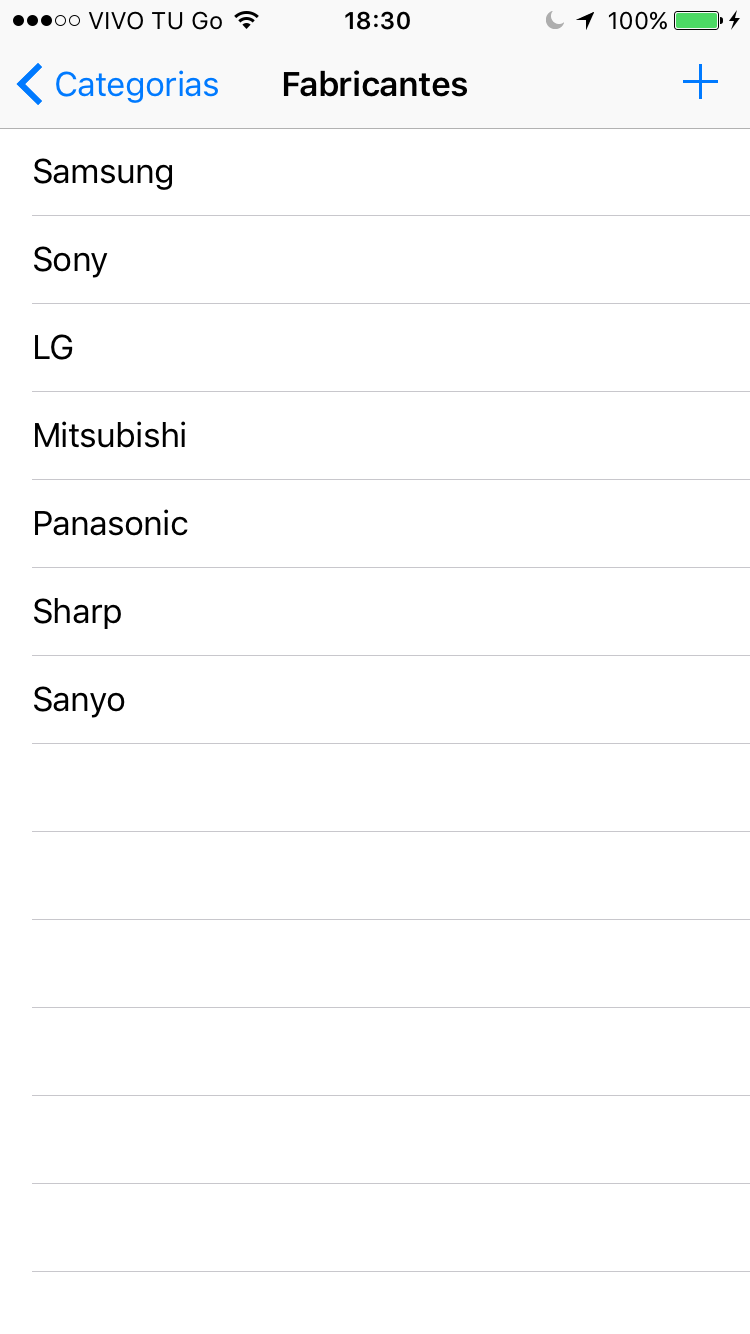
Em seguida é exibida a tela de categorias de dispositivos. Nesta tela é selecionado o tipo de dispositivo a ser controlado.

Figura .: Tela de categorias



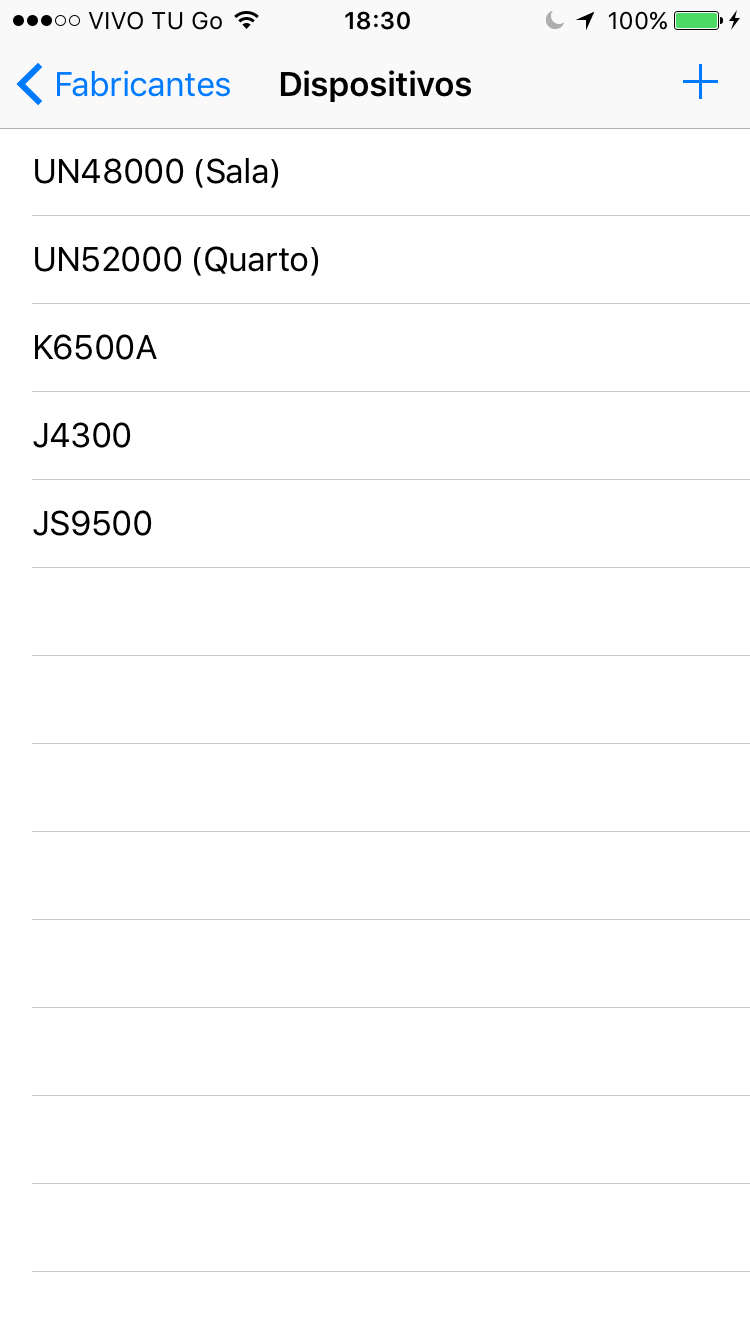
Ao selecionar uma categoria, é apresentada a lista de fabricantes cadastrados para esta categoria.

Figura .: Tela de fabricantes



Ao selecionar um fabricante, são apresentados todos os dispositivos cadastrados deste fabricante para a categoria em questão.

Figura .: Tela de dispositivos



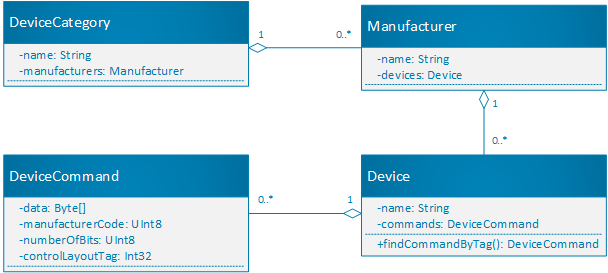
Após a seleção do dispositivo, é apresentada a tela de envio de comandos. Esta tela possui uma disposição de botões semelhante à de um controle remoto convencional.

Figura .: Tela de envio de comandos



Cada tela possui um repositório de dados próprio proveniente da coleção de objetos do item atual. Este relacionamento pode ser observado no diagrama abaixo:

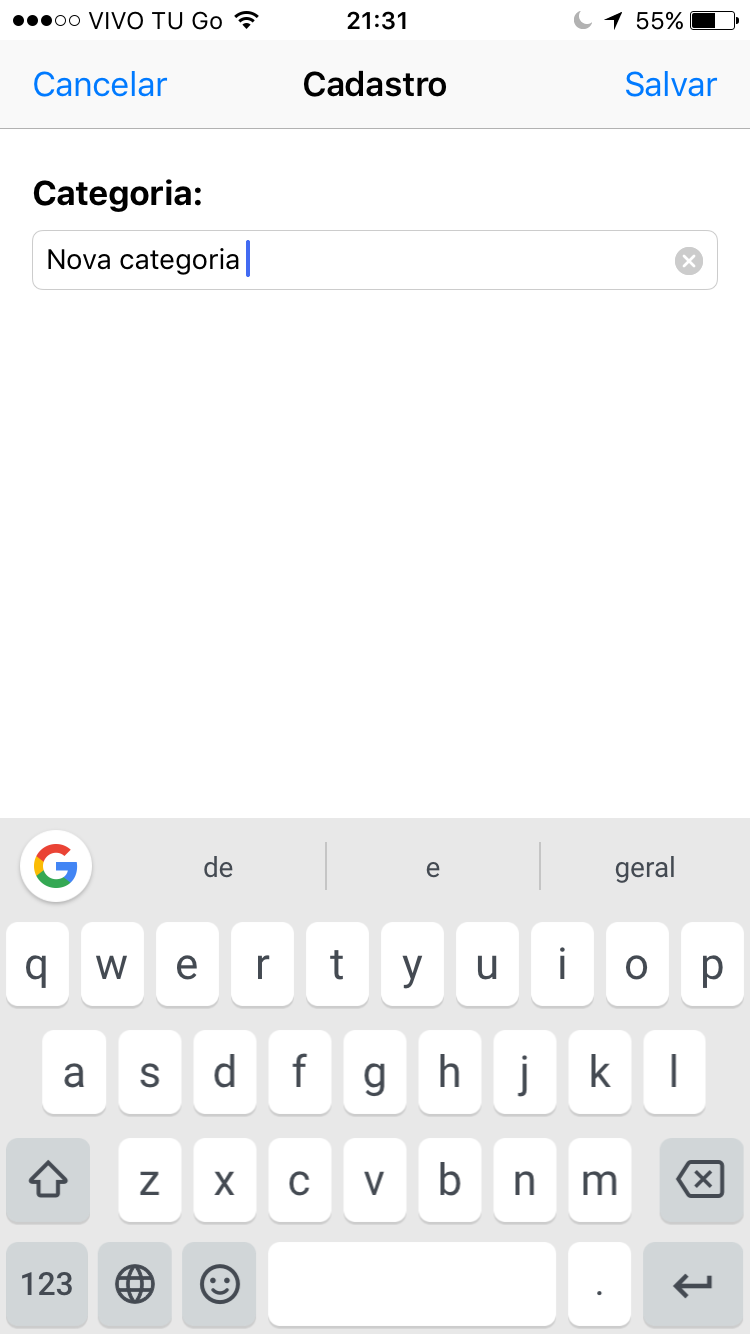
Figura .: Diagrama de objetos UML



#### Extensão

O aplicativo permite a extensão dos itens cadastrados, permitindo a inclusão de novas categorias, fabricantes e dispositivos. Para isto, basta apenas selecionar o botão “+” na lista desejada e preencher os dados do novo item.

Figura .: Exemplo de cadastro de nova categoria



Ao cadastrar um novo dispositivo, não haverá nenhum comando configurado, permitindo assim que sejam cadastrados e mapeados os comandos para seus respectivos botões.

Figura .: Seleção de botão sem comando para novo dispositivo



Selecionando a opção “Cadastrar”, o aplicativo entra em modo de recebimento de dados e se comunica com o módulo Bluetooth para notificar a mudança de estado. A partir deste momento, o aplicativo passa a monitorar a comunicação Bluetooth à fim de verificar o recebimento dos dados e então armazena-los em forma de comando.

Figura .: Modo de recebimento de comando



#### Comunicação dos comandos

Ao selecionar um botão com comando atrelado ou cadastrar um comando novo, os dados deste comando são divididos e comunicados via Bluetooth em três passos, cada um representando uma característica do comando. Estas características são:

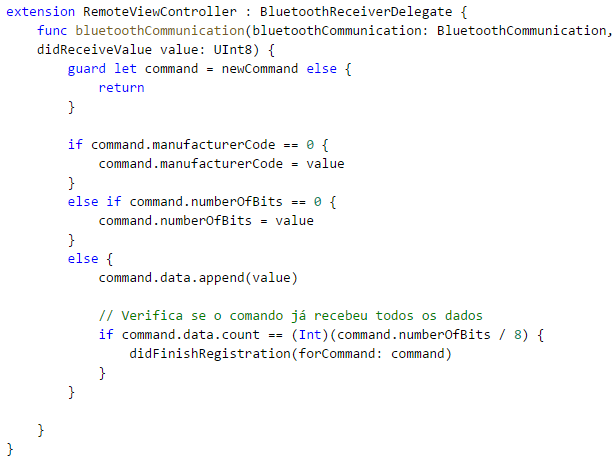
1. Código do fabricante: ID numérico sequencial para cada fabricante. Tem por objetivo definir o protocolo de comunicação IR a ser utilizado;
2. Número de bits: Descreve a quantidade total de bits do comando;
3. Dados: Array (conjunto) de Bytes cujo efetivamente contém os dados representativos do comando;

Cada uma destas características é enviada/recebido do módulo Bluetooth de forma sequencial e serializada. Esta comunicação também é feita através do protocolo BLE, utilizando-se do Framework Apple Core Bluetooth.

Figura .: Envio das características do comando



Figura .: Recebimento das características do comando



### Arduino

Conforme mencionado anteriormente, o microcontrolador utilizado no Arduino Uno é o ATMEGA16U2 da ATMEL. Para o desenvolvimento do software operacional deste microcontrolador foi utilizado a IDE Microsoft Visual Studio 2015. As principais razões para utilização deste ambiente são devido a produtividade que ele proporciona, através de recursos como IntelliSense (auto-complemento de instruções) e marcação dinâmica e instantânea de instruções incorretas, tais como erros de tipografia.

A linguagem utilizada para este desenvolvimento foi C++, devido seu suporte nativo pelo microcontrolador e sua extensibilidade e fácil manutenção por ser uma linguagem orientada a objeto.

#### Comunicação Bluetooth

A comunicação Bluetooth do Arduino com o aplicativo móvel ocorre em duas etapas sequenciais e interdependentes:

1. Entre o aplicativo e o módulo HM-10 via Bluetooth 4.0 através do protocolo BLE;
2. Entre o módulo HM-10 e o Arduino via comunicação serial;

Figura .: Diagrama de comunicação Bluetooth

C:\Users\Beatriz\Downloads\Untitled Diagram (1).png

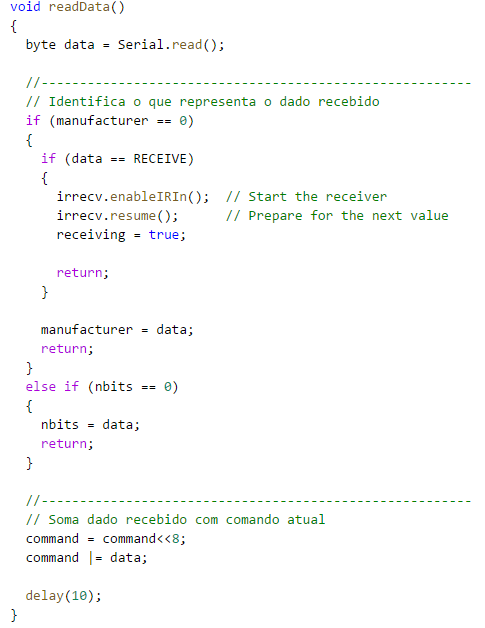
Desta forma, toda a parte de comunicação Bluetooth é abstraída do código do Arduino, necessitando apenas ler (e transmitir) os comandos através da porta serial.

Assim como no aplicativo móvel, a transmissão dos comandos é feita em três partes, cada uma representando uma das características do comando. Sendo elas:

1. Código do fabricante: ID numérico sequencial para cada fabricante. Tem por objetivo definir o protocolo de comunicação IR a ser utilizado;
2. Número de bits: Descreve a quantidade total de bits do comando;
3. Dados: Array (conjunto) de Bytes cujo efetivamente contém os dados representativos do comando;

Para leitura destes dados, o loop (função de repetição contínua no Arduino) valida se possui algum dado disponível na porta serial. Caso tenha, efetua a leitura deste dado e verifica à qual das três características se refere, com base na ordem de recebimento. Após o recebimento de todos os dados, é feita a comunicação IR para envio do comando.

Figura .: Código de leitura e interpretação de dados



#### Comunicação Infravermelho

A comunicação Infravermelho entre o Arduino e o dispositivo a ser controlado ocorre logo após o recebimento dos dados necessários para montagem do comando. A interpretação destes dados e o controle do sinal de saída para formação do comando desejado, ocorre através da biblioteca open-source IRRemote.

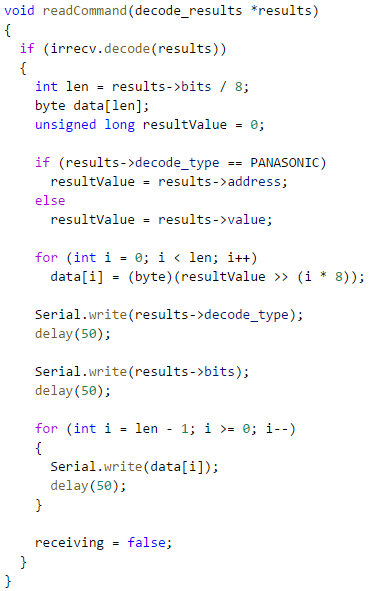
Esta biblioteca disponibiliza um método de comunicação para cada um dos principais protocolos existentes. Caso esteja se comunicando com um dispositivo de protocolo proprietário/exclusivo, é possível manipular a biblioteca de forma customizada para o envio de dados sequenciais independentes de protocolo, replicando os sinais e frequência da fonte original do dado.

Figura .: Diagrama de comunicação IR

C:\Users\Beatriz\Downloads\Untitled Diagram (2).png

O mesmo processo ocorre na leitura de novos comandos, onde após receber a notificação de mudança de estado do aplicativo móvel, o Arduino libera a biblioteca para recebimento de dados. A partir deste momento, o Arduino passa a monitorar os dados da biblioteca, para identificar o recebimento de sinais provenientes do leitor IR. Caso algum sinal seja identificado, estes dados são capturados e transmitidos de volta ao aplicativo móvel via Bluetooth (através do módulo HM-10 na porta serial).

Figura .: Código de captura do sinal IR e envio via serial

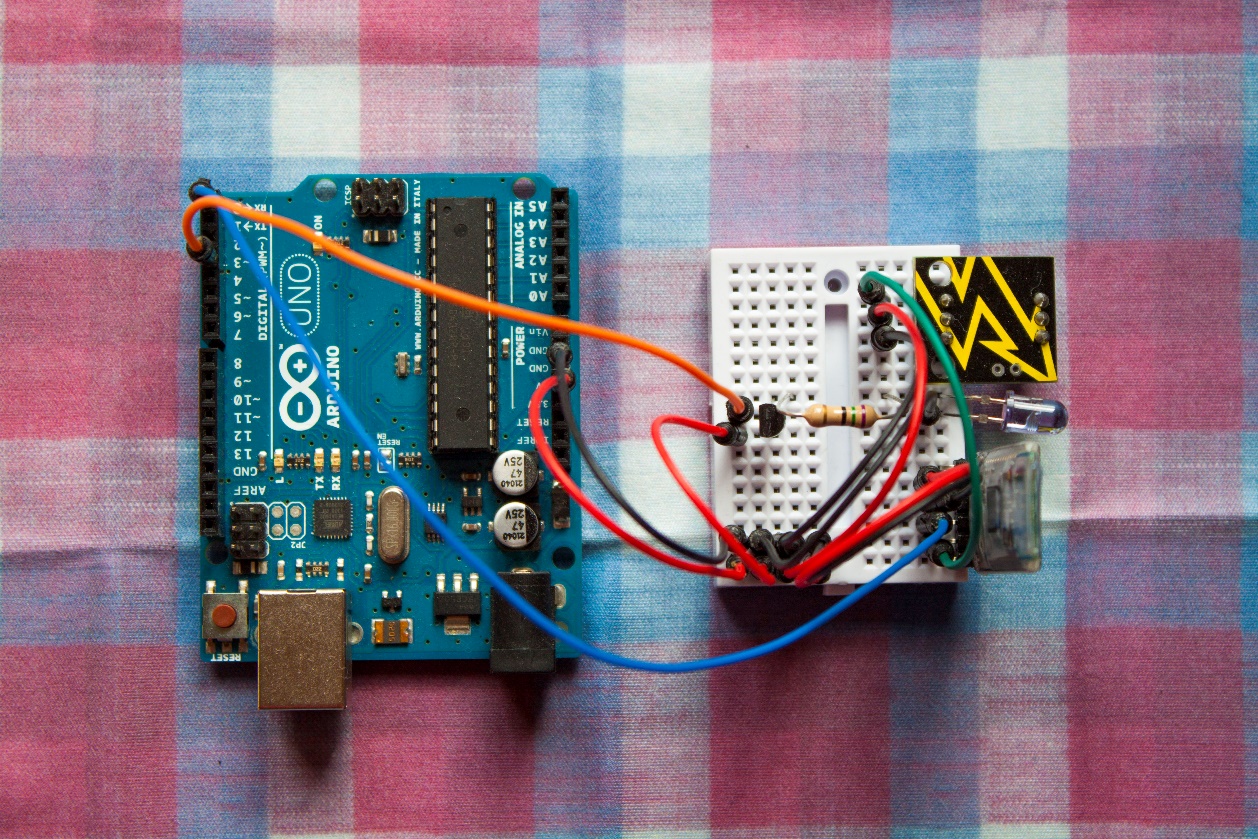


# RESULTADOS

## Hardware

Após a finalização da montagem do circuito, o hardware final ficará da seguinte forma:

Figura .: Hardware final



## Software

O aplicativo móvel após finalizado permite rodar em qualquer dispositivo móvel que possua a versão 10 ou superior do iOS (sistema operacional de dispositivos móveis Apple). No projeto foi utilizado um iPhone 6 rodando a versão 10.1.1 do iOS.

Figura 3.2: Dispositivo móvel rodando aplicação

# COMENTÁRIOS E CONCLUSÕES

Esse projeto teve como objetivo o desenvolvimento de um protótipo que tornasse possível o controle remoto de qualquer aparelho receptor de comando via infravermelho. O objetivo foi alcançado com sucesso, porém durante o desenvolvimento o emissor de dados infravermelhos não atingiu o alcance desejado. Sendo assim foi instalado um transistor para o aumento da corrente do emissor infravermelho e consequentemente o seu alcance.

De início o aplicativo mobile está apenas disponível para smartphones com sistema iOS, restringindo o número de usuários.

# PROJETOS FUTUROS

Para o projeto se tornar viável comercialmente, uma estrutura deverá ser projetada, como um case, um invólucro que torne o produto mais apresentável aos olhos do comprador.

Como projeto futuro seria indispensável o desenvolvimento da aplicação para smartphones de sistema Android.

Também deverá ser considerada a troca da placa do circuito que atualmente é Arduino Uno para a placa Arduino Pro Mini, que resultará em um produto com um menor tamanho físico, melhorando o manuseio e apresentação ao consumidor.

A implementação de um módulo WIFI tornará o uso do protótipo mais cômodo para o usuário uma vez que a configuração e a comunicação do smartphone com o produto poderá ser realizada via rede doméstica do usuário.

# REFERÊNCIAS BIBLIOGRÁFICAS

APPLE, **Swift. Uma linguagem aberta e poderosa para todo mundo criar apps incríveis**. <[www.apple.com/br/swift/](http://www.apple.com/br/swift/)> Acesso em 29 de Maio de 2016.

APPLE. **Xcode overview**. Disponível em: <<http://help.apple.com/xcode/mac/8.0/#/devc8c2a6be1>> Acesso em 5 de novembro de 2016.

ARDUINO**, What is Arduino?**. Disponível em: <[www.arduino.cc/en/Guide/Introduction](http://www.arduino.cc/en/Guide/Introduction)> Acesso em 25 de Maio de 2016.

AURESIDE ASSOCIAÇÃO BRASILEIRA DE AUTOMAÇÃO RESIDENCIAL E PREDIAL - AURESIDE**, 2016: Previsões para casas inteligentes e internet das coisas**. Disponível em: <[www.aureside.org.br/noticias/2016-previsoes-para-casas-inteligentes-e-internet-das-coisas](http://www.aureside.org.br/noticias/2016-previsoes-para-casas-inteligentes-e-internet-das-coisas)> Acesso em 25 de Maio de 2016.

BACKES, André R. **Linguagem C: Completa e descomplicada**. 1. ed. Rio de Janeiro: Elsevier Ltda., 2013.

BLUETOOTH, **What is Bluetooth technology**, Disponível em: <<https://www.bluetooth.com/>> Acesso em 11 de novembro de 2016.

BOLZANI, Caio A. de Moraes. **Desenvolvimento de um Simulador de Controle de Dispositivos Residenciais Inteligentes: Uma Introdução aos Sistemas Domóticos**. 2004. Tese (Mestre em Engenharia) - Escola Politécnica da Universidade de São Paulo, Universidade de São Paulo, São Paulo.

BOZANI, Caio A. de Moraes. **Desmistificando a Domótica**. Disponível em: <[www.aureside.blogspot.com.br/2015/08/desmistificando-domotica.html](http://www.aureside.blogspot.com.br/2015/08/desmistificando-domotica.html)> Acesso em 25 de Maio de 2016.

BRAY, Jennifer; STURMAN, Charles. **Bluetooth Connect Cables**. 1. ed. PrenticeHall, 2000.

FIGUEIREDO, Carlos M. S.; NAKAMURA, Eduardo. **Computação Móvel: Novas Oportunidades e Novos Desafios**, 2003, Universidade Federal do Amazonas.

FOROUZAN, Behrouz A. **Comunicação de Dados e Redes de Computadores**. 3. ed. Porto Alegre: Bookman, 2006.

GIMENEZ, Salvador P. **Microcontroladores 8051**. 1. ed. São Paulo: Editora Érica, 2010.

GOTTSCHLING, Peter. **Discovering Modern C++: An Intensive Course for Scientists, Engineers, and Programmers**. 1. ed. Indianapolis: Addison-Wesley, 2015.

GUDWIN, Ricardo R. **Linguagens de programação: Mini e Microcomputadores: Software**. Departamento de Engenharia de Computação e Automação Industrial, Universidade Estadual de Campinas Faculdade de Engenharia Elétrica e de Computação, 1997.

HUDSON JR., Richard D., **O espectro infravermelho**. Disponível em: <<http://www.las.inpe.br/~cesar/Infrared/espectro.htm>> Acesso em 09 de novembro de 2016.

KOBAYASHI, Carlos Yassunori. **A Tecnologia Bluetooth e aplicações**. Disponível em: <<http://grenoble.ime.usp.br/movel/monografia_bluetooth.pdf>> Acesso em 22 de outubro de 2016.

LOMBARDI, Rafael R. Rezende. **Controle Remoto Infravermelho Para Automação**. 2006. Dissertação (Graduação) – Centro Universitário de Brasília – UNICEUB, Faculdade de Ciências Exatas e Tecnologia – FAET, Brasília. Disponível em: < <http://www.repositorio.uniceub.br/bitstream/123456789/3120/2/20126521.pdf>> Acesso em 29 de outubro de 2016.

MARCONI, Guilherme; HARADA, Ricardo Akira. **Introdução A Programação Em C**. 1. ed. Ebook: Clube dos Autores, 2014.

MCROBERTS, M. **Arduino básico**. São Paulo: Novatec, 2011.

MICROSOFT. **Visual Studio IDE**, Disponível em: <<https://msdn.microsoft.com/en-us/library/dn762121.aspx>> Acesso em 5 de novembro de 2016.

OLIVEIRA, Hélio M. **Engenharia de Telecomunicações**. 1. ed. Recife: HM, 2012.

OLIVEIRA, Ricardo A. Rabelo. **Bluetooth e Multimídia**. In: WORKSHOP EM TRATAMENTO DE IMAGENS, 4., 2003, Minas Gerais. *Anais...* NPDI/DCC/ICEx/UFMG, 2003. p. 14-25.

PACIEVITCH, Yuri. **C++**. Infoescola. Disponível em: <[www.infoescola.com/informatica/cpp/](http://www.infoescola.com/informatica/cpp/)> Acesso em 29 de Maio de 2016.

PIYARE, Rajeev; TAZIL, M., **Bluetooth based home automation system using cell phone**. In:INTERNATIONAL SYMPOSIUM ON CONSUMER ELECTRONICS, 15., 2011, Fiji. Department of Electrical & Electronics Engineering, Fiji National University, 2011. Disponível em: <<https://www.researchgate.net/publication/231182479_Bluetooth_based_home_automation_system_using_cell_phone>> Acesso em 12 de Outubro de 2016.

SILVA, Elieser Costa; BENTO, Janderson Santos. **Domótica: Uma Visualização com Microcontrolador Arduino**. 2014. Dissertação (Graduação) – Instituto Federal do Pará. Campus Itaituba, Coordenação de Informática – Curso de Tecnologia em analises e desenvolvimento de sistemas, Itaituba.

SILVA, Mauricio C.; GAMBARATO, Vivian T. Santos, **Domótica e tecnologias utilizadas na automação residencial**, 2016, Faculdade de Tecnologia de Botucatu, FATEC-BT, Botucatu. Disponível em: <<http://www.fatecbt.edu.br/seer/index.php/tl/article/download/389/270>> Acesso em 12 de outubro de 2016

SIQUEIRA FILHO, J.B.; SILVA FILHO, J.B. **Tecnologia da informação para administradores**. 2. ed. Fortaleza: Universidade de Fortaleza, 2006.

STARLIN, Gorki. **Redes de Computadores/Comunicação de  
Dados TCP/IP - Conceitos, Protocolos e Uso**. Rio de Janeiro: Alta Books, 2007.

TRINETO, Rafael Descio, **Ambiente para controle de eletroeletrônicos via dispositivos móveis**, 2008, Universidade Positivo – Núcleo de ciências exatas e tecnológicas, UP/NCET, Curitiba. Disponível em: <<http://www.up.edu.br/blogs/engenharia-da-computacao/wp-content/uploads/sites/6/2015/06/2008.22.pdf>> Acesso em 20 de Outubro de 2016.

VANZETTI, Riccardo. **Practical applications of infrared techniques: A new tool in a new di mengion for problem solving**. Nova Iorque: J Wiley, 1972.