

CARLOS RANIENY PEREIRA ROCHA



# **LIBRAS e ACESSIBILIDADE:** **EDUCAÇÃO, RECURSOS TECNOLÓGICOS** **e QUALIDADE DE VIDA**

## **Direção Editorial**

Prof.º Dr. Adriano Mesquita Soares

## **Autor**

Carlos Ranieny Pereira Rocha

## **Capa**

AYA Editora

## **Revisão**

O Autor

## **Executiva de Negócios**

Ana Lucia Ribeiro Soares

## **Produção Editorial**

AYA Editora

## **Imagens de Capa**

br.freepik.com

## **Área do Conhecimento**

Linguística, Letras e Artes

# **Conselho Editorial**

Prof.º Dr. Aknaton Toczec Souza  
Centro Universitário Santa Amélia  
Prof.ª Dr.ª Andreia Antunes da Luz  
Faculdade Sagrada Família  
Prof.º Dr. Carlos López Noriega  
Universidade São Judas Tadeu e Lab.  
Biomecatrônica - Poli - USP  
Prof.º Me. Clécio Danilo Dias da Silva  
Centro Universitário FACEX  
Prof.ª Dr.ª Daiane Maria De Genaro Chirolí  
Universidade Tecnológica Federal do Paraná  
Prof.ª Dr.ª Déborah Aparecida Souza dos Reis  
Universidade do Estado de Minas Gerais  
Prof.ª Dr.ª Eliana Leal Ferreira Hellvig  
Universidade Federal do Paraná  
Prof.º Dr. Gilberto Zammar  
Universidade Tecnológica Federal do Paraná  
Prof.ª Dr.ª Ingridi Vargas Bortolaso  
Universidade de Santa Cruz do Sul  
Prof.ª Ma. Jaqueline Fonseca Rodrigues  
Faculdade Sagrada Família  
Prof.º Dr. João Luiz Kovaleski  
Universidade Tecnológica Federal do Paraná  
Prof.º Me. Jorge Soistak  
Faculdade Sagrada Família  
Prof.º Me. José Henrique de Goes  
Centro Universitário Santa Amélia  
Prof.ª Dr.ª Leozenir Mendes Betim  
Faculdade Sagrada Família e Centro de  
Ensino Superior dos Campos Gerais  
Prof.ª Ma. Lucimara Glap  
Faculdade Santana

Prof.º Dr. Luiz Flávio Arreguy Maia-Filho  
Universidade Federal Rural de Pernambuco  
Prof.º Me. Luiz Henrique Domingues  
Universidade Norte do Paraná  
Prof.º Dr. Marcos Pereira dos Santos  
Faculdade Rachel de Queiroz  
Prof.º Me. Myller Augusto Santos Gomes  
Universidade Estadual do Centro-Oeste  
Prof.ª Dr.ª Pauline Balabuch  
Faculdade Sagrada Família  
Prof.º Me. Pedro Fauth Manhães Miranda  
Centro Universitário Santa Amélia  
Prof.ª Dr.ª Regina Negri Pagani  
Universidade Tecnológica Federal do Paraná  
Prof.º Dr. Ricardo dos Santos Pereira  
Instituto Federal do Acre  
Prof.ª Ma. Rosângela de França Bail  
Centro de Ensino Superior dos Campos  
Gerais  
Prof.º Dr. Rudy de Barros Ahrens  
Faculdade Sagrada Família  
Prof.º Dr. Saulo Cerqueira de Aguiar Soares  
Universidade Federal do Piauí  
Prof.ª Ma. Sílvia Apª Medeiros Rodrigues  
Faculdade Sagrada Família  
Prof.ª Dr.ª Sílvia Gaia  
Universidade Tecnológica Federal do Paraná  
Prof.ª Dr.ª Sueli de Fátima de Oliveira Miranda  
Santos  
Universidade Tecnológica Federal do Paraná  
Prof.ª Dr.ª Thaisa Rodrigues  
Instituto Federal de Santa Catarina

© 2021 - **AYA Editora** - O conteúdo deste Livro foi enviado pelos autores para publicação de acesso aberto, sob os termos e condições da Licença de Atribuição Creative Commons 4.0 Internacional (**CC BY 4.0**). As ilustrações e demais informações contidas desta obra são integralmente de responsabilidade de seus autores.

R672 Rocha, Carlos Ranieny Pereira

Libras e acessibilidade: educação, recursos tecnológicos e qualidade de vida [recurso eletrônico] / Carlos Ranieny Pereira Rocha. -- Ponta Grossa: Aya, 2021. 53 p. --ISBN: 978-65-88580-56-1

Inclui biografia

Inclui índice

Formato: PDF

Requisitos de sistema: Adobe Acrobat Reader.

Modo de acesso: World Wide Web.

DOI 10.47573/aya.88580.1.2914

Título 1. Língua brasileira de sinais. 2. Software. 3. Qualidade de vida. I.

CDD: 419

Ficha catalográfica elaborada pela bibliotecária Bruna Cristina Bonini - CRB 9/1347

**International Scientific Journals Publicações  
de Periódicos e Editora EIRELI  
AYA Editora©**

**CNPJ:** 36.140.631/0001-53

**Fone:** +55 42 3086-3131

**E-mail:** contato@ayaeditora.com.br

**Site:** <https://ayaeditora.com.br>

**Endereço:** Rua João Rabello Coutinho, 557  
Ponta Grossa - Paraná - Brasil  
84.071-150

## LISTA DE ABREVIATURAS E SIGLAS

APP	Application
FTDI	Future Technology Devices International
GND	Graduated neutral density filter
IDE	Integrated Development Environment
ISO/IEC	Organization of Standardization / International Electrotechnical Commission
LASER	Light Amplification by Stimulated Emission of Radiation
LDR	Light Dependent Resistor
LED	Light Emitting Diode
LP	Linguagem de programação
MIT	Massachusetts Institute of Technology
PC	Personal Computer
PWM	Pulse-Width Modulation
USB	Universal Serial Bus
VCC	Continuous Current Voltage

# Resumo

---

Com o avanço das tecnologias da informação e comunicação, a automação residencial tem se mostrado uma necessidade cada vez maior a comunidade surda. O surgimento de novas plataformas e dispositivos tem ampliado cada vez mais o leque de possibilidades deste cenário. Nesta perspectiva, este trabalho está motivado em explorar as possibilidades do desenvolvimento de um sistema de automação residencial de baixo custo baseado na integração entre a Eletrônica com os Sistemas de Informação, com base no uso da plataforma Arduino e do Android, contemplando três segmentos básicos: segurança patrimonial, monitoramento de abastecimento de água e auxílio no consumo de energia elétrica. Assim, este aplicativo tem por objetivo, além de explorar as possibilidades técnicas de utilizar projetos de automação residencial desenvolvidos com o uso da plataforma Arduino e seus sensores, proporcionar um maior nível de segurança, comodidade e economia de energia e água residenciais aos surdos. O método utilizado, portanto, foi o da revisão de literatura aliada a pesquisa experimental. Neste trabalho utilizou-se as linguagens de programação C++ - para a programação das placas Arduino – e a ferramenta App Inventor – como auxílio ao desenvolvimento do aplicativo, denominado Infragualuz. O resultado obtido mostrou que é perfeitamente possível monitorar e automatizar tarefas residenciais de maneira integrada com esta tecnologia.

**Palavras-Chave:** arduino. android. automação residencial.

# SUMÁRIO

<b>APRESENTAÇÃO</b>	<b>7</b>
<b>COMUNIDADE SURDA, LIBRAS E ACESSIBILIDADE</b>	<b>9</b>
<b>Arduino</b>	<b>9</b>
<b>Mobile</b>	<b>13</b>
<b>Linguagens de programação</b>	<b>15</b>
<b>Ferramenta utilizada para auxílio no desenvolvimento</b>	<b>17</b>
<b>Usabilidade</b>	<b>18</b>
<b>Automação residencial</b>	<b>19</b>
<b>Aplicações similares e funcionalidade do Infragualuz</b>	<b>20</b>
<b>DESENVOLVIMENTO DO APLICATIVO, PROGRAMAÇÃO DA PLACA DE ARDUINO E MONTAGEM DOS CIRCUITOS</b>	<b>22</b>
<b>O projeto</b>	<b>22</b>
<b>Parte lógica</b>	<b>24</b>
<b>Parte física</b>	<b>39</b>
<b>RESULTADOS OBTIDOS</b>	<b>45</b>
<b>CONSIDERAÇÕES FINAIS</b>	<b>47</b>
<b>REFERÊNCIAS</b>	<b>49</b>
<b>SOBRE O AUTOR</b>	<b>50</b>
<b>ÍNDICE REMISSIVO</b>	<b>51</b>

# APRESENTAÇÃO

Prezados leitores,

O reconhecimento da automação no meio residencial e industrial tem se expandido rapidamente, devido ao avanço tecnológico, que por sua vez, possibilitou a facilidade de acesso aos dispositivos capazes de desenvolverem sistemas automáticos. Rapidamente essas tecnologias cresceram e tiveram o seu lugar no mercado cada vez mais competitivo. Conseqüentemente, a oferta pela procura da automação residencial aumentou, ocasionando uma queda no seu preço, além de atribuir uma concorrência à disposição dos consumidores (BOLETIM INDUSTRIAL, 2013).

Com relação à qualidade de vida de pessoas com algum tipo de deficiência, uma questão que ganha relevo é a acessibilidade, que pode ser percebida como a possibilidade de realização de atividades, em variados ambientes, visando o conforto, segurança e autonomia do indivíduo e ainda pode relacionar-se as possibilidades de comunicação do indivíduo por meio de recursos tecnológicos. Deste modo o que se percebe é que algumas dos entraves encontrados pelos surdos é a grande dificuldade de se comunicar com outras pessoas, ou mesmo para utilizarem muitos dos recursos tecnológicos voltados à automação residencial.

Pode-se considerar que a necessidade pela busca de conforto, praticidade e demais fatores de facilidade em acesso à um ambiente doméstico, amplia a imensa busca das tecnologias que acobertam a automação residencial (WORTMEYER; FREITAS; CARDOSO, 2005).

Visando atender a este anseio por automações residenciais para o público surdo, surge a ideia da utilização da plataforma Arduino em projetos de automação, possibilitando uma solução viável e flexível, através de um software e hardware livre.

O Arduino é uma plataforma que se destaca por facilitar o desenvolvimento de projetos de automação e pela sua flexibilidade a diversos sistemas, além de ser uma plataforma open-source de hardware e software, isso torna possível que qualquer pessoa o modifique, adaptando-o e implementando-o da maneira que lhe for conveniente. Com a utilização do Arduino é possível apresentar a sociedade soluções de projetos a baixo custo comparados com os existentes hoje no mercado.

É importante destacar também o aumento vertiginoso da utilização de smartphones e dispositivos móveis nos dias atuais em todas as classes e níveis sociais, fazendo com que surgisse também uma ampla variedade de aplicativos, para os mais diversos fins. Destacamos aqui a capacidade de integração entre aplicativos móveis e a plataforma Arduino. Essa junção Arduino + mobile veio despertar nas pessoas a necessidade de obter informações residenciais em seus celulares tablets<sup>1</sup> e demais aparelhos móveis.

Porém, existe um universo de demandas e possibilidades, tanto de sensores, aplicativos e funcionalidades quanto de automação residencial em diversas tarefas cotidianas: segurança patrimonial, monitoramento de insumos (água, luz), controles de entrada e saída, identificação, sensores de incêndio e muitas outras.

Assim, a pergunta a ser feita é: quais as reais aplicabilidades e possibilidades de automa-

---

<sup>1</sup> Tipo de computador portátil, com tela sensível ao toque (touchscreen).

tizar várias tarefas residenciais ao mesmo tempo utilizando Arduino, Android e seus sensores?

Diante desta realidade, para mostrar a viabilidade técnica do cenário acima descrito, este trabalho tem como título Infragualuz: desenvolvimento de um aplicativo interligado a um sistema de monitoramento através da integração da eletrônica com sistemas de informação usando Arduino.

Os tópicos abordados no trabalho foram considerados pelo fato de que na atualidade a segurança é primordial para o bem estar social. O monitoramento de água faz-se necessário em tempos de crise hídrica. A energia elétrica também possui um fator ambiental importante e, por essa razão, merece iniciativas que auxiliem na economia deste recurso.

Assim, o objetivo geral deste projeto é desenvolver um aplicativo capaz de integrar a automação de mais de uma tarefa residencial em uma única solução, em plataforma Arduino.

Como objetivos específicos, tem-se: A exploração das possibilidades de automação residencial através de aplicativos e Arduino; desenvolvimento, teste e aplicação de programação da plataforma Arduino; teste e aplicação de sensores de água e controles para automação de luzes residenciais; implantação de módulo bluetooth para comunicação entre smartphone e Arduino bem como, testar eficiência de sensores infravermelho e luz na automação residencial.

Para a elaboração desse trabalho, primeiramente foi realizada uma revisão de literatura, a fim de fundamentar nossa investigação. E num segundo momento, utilizou-se a pesquisa experimental, que de acordo com Antônio Carlos Gil (2002), “Essencialmente, a pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto.” (GIL, 2002, p. 47). Nesta fase é que será implementado o proposto.

Este trabalho fora estruturado do seguinte modo: Introdução onde expôs-se de maneira geral os elementos que compõem essa monografia. No capítulo 1 há a exploração dos conceitos que fundamentam teoricamente essa pesquisa. Esse capítulo apresenta a discussão dos conceitos da plataforma Arduino, Mobile, Sistema Operacional Android, Linguagens de programação, App Inventor, Usabilidade, Automação residencial, bem como apontamos para algumas aplicações similares.

O capítulo 2 apresenta as etapas do processo de desenvolvimento do aplicativo aqui desenvolvido, o Infragualuz, bem como a programação da placa Arduino e a montagem dos circuitos, itens que se complementam na composição do todo desse trabalho. Esse capítulo foi dividido em duas partes: uma parte lógica que contém a parte abstrata dessa monografia e, é composta pela programação da placa Arduino Uno e pela programação do aplicativo mobile. E uma parte física composta pelos circuitos montados para o monitoramento residencial via infravermelho; o monitoramento do abastecimento residencial de água e para o monitoramento do status das lâmpadas em uma residência.

O capítulo 3 apresenta os resultados finais da pesquisa detalhando o aplicativo desenvolvido.

Na conclusão retomou-se os aspectos mais importantes discutido ao longo desse trabalho como intuito avaliar se os objetivos propostos foram alcançados.



# COMUNIDADE SURDA, LIBRAS E ACESSIBILIDADE

Antes de se embasar teoricamente este trabalho cabe pontuar que objetiva-se aqui o desenvolvimento de um aplicativo para sistemas mobiles Android + Arduino que contemple as funções de segurança, monitoramento de abastecimento de água e auxílio no consumo de energia elétrica, utilizando a placa Arduino.

Para o embasamento teórico deste trabalho, portanto, foram abordados os conceitos da plataforma Arduino, IDE do Arduino, Sistema Operacional Android, Mobile, Linguagens de programação (C++ e java), App Inventor, Aplicações similares, Usabilidade e Automação residencial conceitos e tecnologias estas que serviram como fio condutor a implementação do aplicativo Infragualuz.

## Arduino

A plataforma de computação física, Arduino compõe-se de vários modelos de placa de valor acessível e por um software gratuito, tais atributos aliados a fatores de ordem funcional atribuíram larga utilização a plataforma desde sua elaboração, em 2005. De acordo com Massimo Banzi e com o engenheiro eletricista, Michael Shiloh, co-fundadores do projeto Arduino:

O hardware e o software do Arduino são de fonte aberta. A filosofia da fonte aberta promove uma comunidade que compartilha seu conhecimento generosamente. Isso é ótimo para iniciantes porque com frequência há disponível ajuda geograficamente perto, e sempre há ajuda online, em níveis de habilidade diferentes e para um inacreditável número de tópicos. (BANZI; SHILOH, 2015, p. 17)

Os autores ratificam ainda que o Arduino, plataforma desenvolvida em 2005 na Itália, configura-se como uma ferramenta de desenvolvimento “open source”, ou seja, baseia-se em um sistema de códigos abertos, que fazem referencia ao código-fonte de um software, que pode ser ajustado para diversos fins. A princípio sua concepção baseou-se em um hardware de fácil manipulação, aprendizado e que tivesse um baixo custo, visto estar voltado para o estudo e projetos escolares. Tendo tal plataforma encontrado grande aceitação e sucesso foi premiada em 2006 com uma menção honrosa na categoria Comunidades Digitais, pela Prix Ars Electronica. Visto tamanha aceitação observa-se a partir de então um crescente número de projetos que se baseiam nessa tecnologia.

Essa ferramenta é frequentemente associada à filosofia de “Physical Computing” (Termo em inglês que significa Computação Física, tal termo utiliza elementos da eletrônica na criação de novos materiais e produtos possibilitando desenvolver equipamentos digitais de computação respondam à, realidade física em que nós vivemos, aproveitando software e hardware para este fim). Desse modo Banzi e Shiloh (2015) esclarecem que:

O Arduino é uma plataforma de computação física de fonte aberta para a criação de objetos interativos independentes ou em colaboração com softwares do computador. Ele foi projetado para artistas, designers e outros profissionais que queiram incorporar a computação física a seus projetos sem que para isso precisem ter se formado em Engenharia Elétrica. (BANZI, SHILOH, 2015, p. 17)

Assim os projetos desenvolvidos com o uso do Arduino podem ser independentes ou podem estabelecer comunicação com um computador para a realização da tarefa, com uso de software específico. Segundo McRoberts (2011) o Arduino:

Em termos práticos, um Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de Hardware e Software (MCROBERTS, 2011, p. 22).

No entanto, Justen (2011) chama a atenção para o fato de que apesar de o Arduino ser um computador independente, ele precisa de um computador doméstico, para que suas funcionalidades possam ser exploradas. O autor evidencia que a integração com computadores pode ser feita através de porta USB, por meio da qual pode-se desenvolver um software e executar no próprio Arduino.

Com relação a sua programação o Arduino usa uma IDE própria, localizada no site do projeto e a codificação feita para a programação denomina-se Sketch.

Um dos motivos pelos quais a plataforma Arduino obteve tamanha aceitação foi a sua alta coesão com Shields<sup>2</sup> o que significa dizer que placas de diferentes tipos podem ser conectadas ao Arduino. Tais placas vão desde sensores a placas auxiliares e promovem uma verdadeira flexibilidade de uso do Arduino.

As múltiplas possibilidades de uso da plataforma não se restringem a variedade de Shields podem ser interligadas a ele, mas também a uma grande diversidade de tipos de placas Arduino, sendo elas: UNO, Due, Leonardo, Mega2560, MegaADK, Micro, Mini, Nano, Ethernet, Esplora, ArduinoBT, Fio, Pro(168), Pro(328), ProMini, LillyPad e seus derivados. Para trabalhos robustos é mais indicado o Arduino Mega2560. Desse modo seguem abaixo as Figuras 1 e 2 que ilustram as placas Uno e Mega:

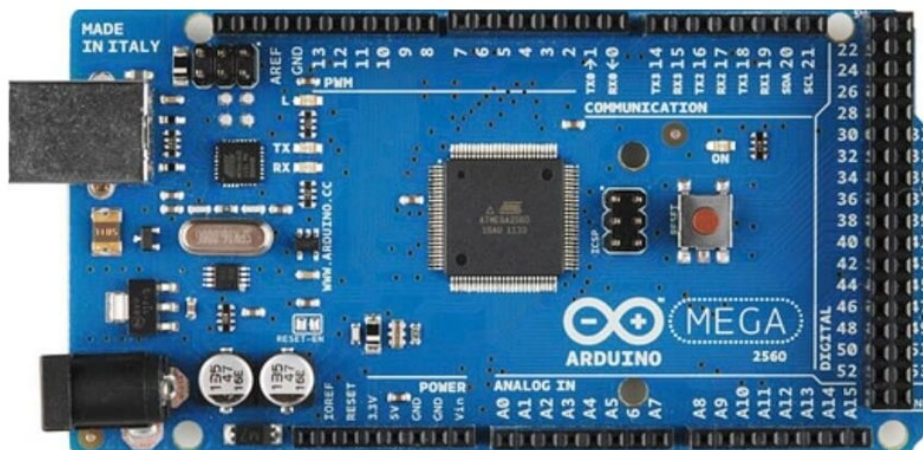
**Figura 1 - Arduino Uno**



**Fonte: MCROBERTS, 2011, p. 23.**

<sup>2</sup> Placas de expansão de hardware que aumentam as funcionalidades da placa Arduino ao se conectarem

Figura 2 - Arduino Mega



Fonte: MILETO, 2015, s.p.

## Placa Arduino Uno

Nesse trabalho será utilizada a placa Arduino Uno que é uma placa baseada no microcontrolador ATmega328P. A placa possui 14 pinos de entrada / saída digitais, desses pinos 06 podem ser usados como saídas PWM, possui ainda 06 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, uma tomada de força, um cabeçalho ICSP e um botão de reinicialização.

O termo "Uno" advém do italiano e significa "Um". Tal nome foi escolhido para marcar o lançamento do software IDE 1.0. A placa Uno foi, portanto, a primeira de uma série de placas USB Arduino.

A placa de Arduino Uno, configura-se como uma evolução de precursora, a Duemilanove que fazia uso de um chip FTDI, que possui um valor mais alto o que conferia um preço elevado a placa. No lugar dele a placa possui um microcontrolador baseado no ATmega328P, que apresenta um custo-benefício melhor pois além de ter um menor preço, permite que o chip USB tenha seu firmware atualizado, de modo que o Arduino possa ser exibido tanto em um PC quanto em outros dispositivo, o que aponta para uma série de novas possibilidades para o Arduino. Para McRoberts (2011):

Talvez o Arduino mais versátil, e daí o mais popular, seja o Arduino Uno (ou seu predecessor, o Duemilanove). Isso ocorre porque ele utiliza um chip padrão de 28 pinos, ligado a um soquete de circuito integrado (CI). A beleza desse sistema é que, se você criar alguma coisa com um Arduino e depois quiser transformá-la em algo permanente, em vez de utilizar uma placa Arduino relativamente cara, poderá simplesmente retirar o chip da placa e colocá-lo em sua própria placa de circuito, em seu dispositivo personalizado. Dessa forma, você tem um dispositivo embarcado personalizado, o que é muito bacana (MCROBERTS, 2011, p. 25).

A placa pode, ainda, ser alimentada através da conexão USB ou com uma fonte de alimentação externa.

O Arduino Uno pode ser programado por meio do software IDE, uma vez que o ATmega328 no Arduino já vem pré-programado com um bootloader<sup>3</sup> que lhe possibilita o carregamento

<sup>3</sup> Programa que é inicializado sempre que ligamos um dispositivo para ativação do Sistema Operacional.

de um novo código sem que seja necessário o uso de um programador externo.

Na seção que segue serão explicitadas as especificidades da interface de programação do Arduino: a IDE do Arduino.

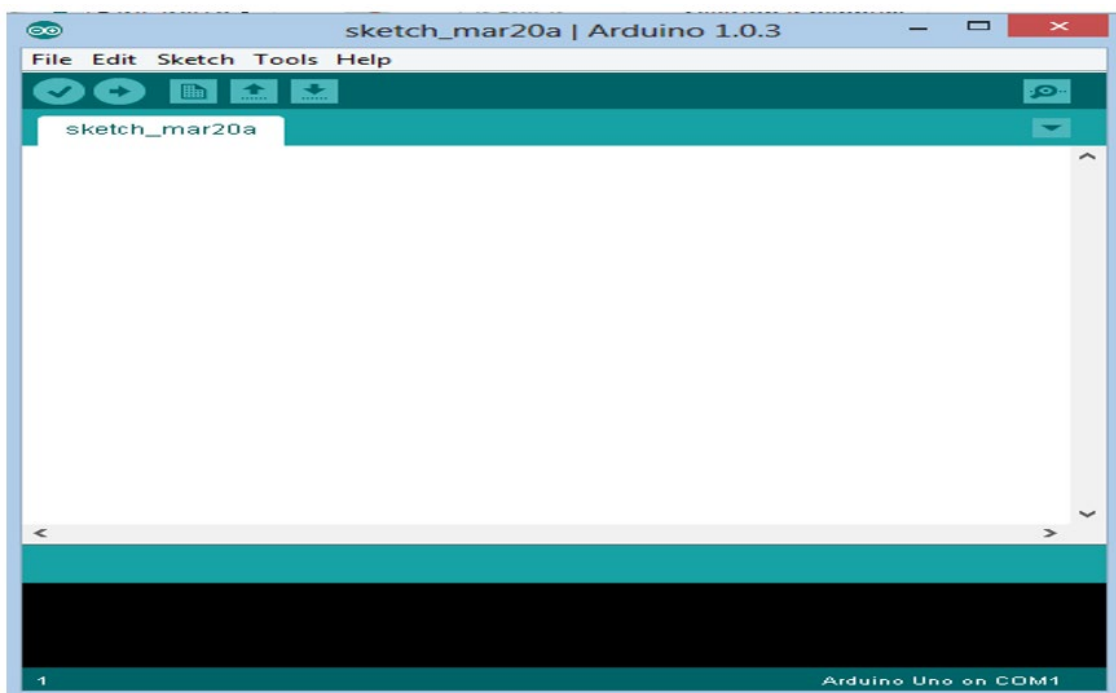
## IDE do Arduino

McRoberts (2011) aponta que para fins de programação da plataforma Arduino utiliza-se o IDE do Arduino, que se apresenta como um software livre onde se escreve o código baseado na linguagem C++.

Essa IDE permitirá a escrita de um programa, que deverá ser enviado para o Arduino. Assim o Arduino, executará tais instruções e interagirá com os dispositivos conectados a ele. Dessa forma chamam-se Sketch, os programas desenvolvidos no âmbito do Arduino. Assim o IDE do Arduino é exatamente o software que será utilizado para escrever os programas e, depois, fazer o upload deles para a placa. Para download do IDE mais recente deve-se acessar a página do Arduino.

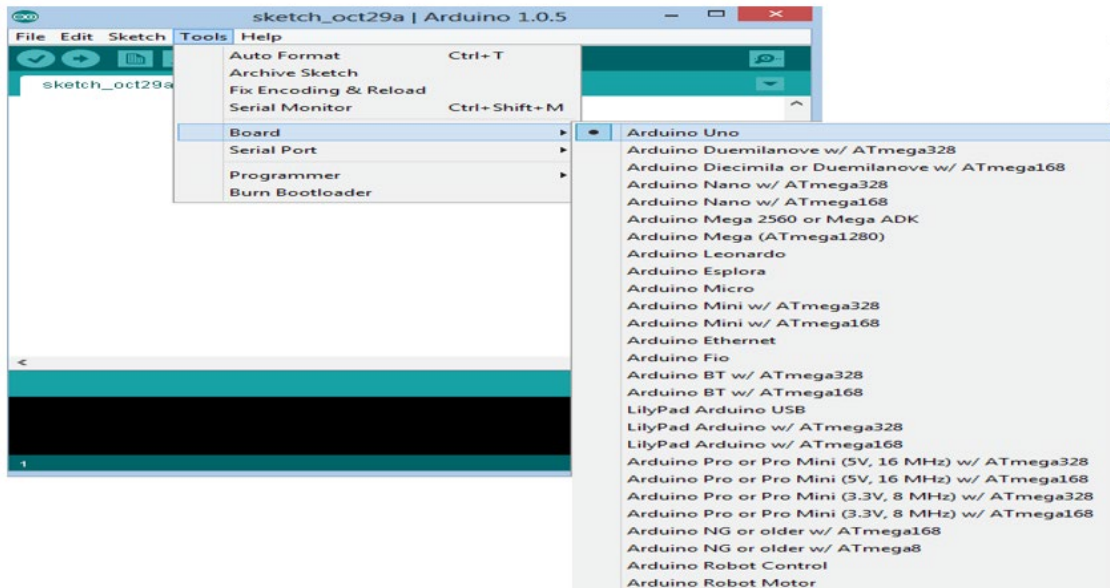
O IDE divide-se em três partes, a saber: a Toolbar, localizada no topo, a Window, localizada no centro, e a janela de mensagens, localizada na base. As Figuras 3 e 4 mostram a IDE:

**Figura 3 - IDE do Arduino**



Fonte: MCROBERTS, 2011, p. 29.

Figura 4 - Menu de Boards do Arduino



Fonte: MCROBERTS, 2011, p. 30.

## Mobile

O desenvolvimento de aplicativos móveis tornou-se uma realidade e estão cada vez mais presentes na sociedade. Dentre as plataformas mobile podemos citar os Sistemas Operacionais iOS da Apple e o Android do Google. Conforme Rabello (2009):

O desenvolvimento de aplicações para dispositivos móveis tem evoluído exponencialmente com o tempo e já se tornou um padrão de desenvolvimento quase onipresente no sentido de que muitas empresas estão escalonando ou recrutando novos desenvolvedores para formarem seus grupos de desenvolvimento para sistemas móveis a fim de criar novas ou adaptar soluções de serviços existentes para suprir a demanda do mercado (empresas bancárias disponibilizando serviços de operações financeiras pelo celular, controle de estoque de materiais e outros). (Rabello, 2009, p. 12)

O surgimento desta tecnologia tem possibilitado o desenvolvimento de uma série de soluções que visam atender as demandas sociais. Impulsionados pelo apelo deste mercado, é que elaborou-se esse trabalho, com o intuito da criação de aplicativo para a plataforma mobile Android que atenda algumas demandas residenciais.

No próximo tópico serão abordadas características e funcionalidades do sistema operacional Android, software fundamental para o desenvolvimento desta pesquisa.

## Android

A implementação de aplicativos em Android possui objetivos variados, tais como entretenimento, comunicação entre pessoas, propagação da informação e dentre outros a segurança e automação residencial, finalidade deste projeto. Assim torna-se fundamental aqui a exposição de algumas peculiaridades deste software livre utilizado como base operacional para a implementação do aplicativo ora descrito.

Configurando-se atualmente como o sistema operacional móvel mais utilizado em todo o mundo<sup>4</sup>, o Android possui uma história bastante recente que data de 2003, ano em que na Ca-  
<sup>4</sup> NIELSEN IBOPE; *Mobile marketing association (MMA)*.

lifórnica, os empresários Andy Rubin, Chris White, Nick Sears e Rich Miner, fundaram a Android Inc. que posteriormente foi adquirida pela empresa Google, uma vez observada a exigência dos usuários e o grande crescimento do mercado dos dispositivos móveis no mundo. De acordo com Ricardo R. Lecheta (2013):

O Android é a resposta do Google para ocupar esse espaço. Consiste em uma nova plataforma de desenvolvimento para aplicativos móveis, baseada em um sistema operacional Linux, com diversas aplicações já instaladas e, ainda, um ambiente de desenvolvimento bastante poderoso, ousado e flexível. (LECHETA, 2013, p. 20).

## A plataforma Android

De acordo com Ableson (2012), o Android é um ambiente de software escrito para dispositivos móveis, sendo assim possui um Sistema Operacional (OS) baseado em um Kernel Linux<sup>5</sup>, uma Interface de Usuário (IU), aplicativos de usuário, bibliotecas de código, frameworks de aplicativo, suporte a multimídia e outros. O autor esclarece ainda, que enquanto os componentes do OS estão escritos em C ou C++, os aplicativos de usuários são escritos em Java para Android. Mesmo os aplicativos embutidos são escritos em Java.

Um dos pontos de destaque do Android, e também um dos motivos pelos quais o Android foi escolhido como sistema operacional-base deste projeto, é sua arquitetura extremamente flexível, que possibilita a integração de aplicações. Acerca dessa funcionalidade, Lecheta afirma que:

É possível integrar aplicações de uma forma simples, sejam elas desenvolvidas por você, sejam aplicações nativas. Por exemplo, imagine que sua aplicação precise consultar a agenda de contatos para selecionar determinado amigo, e logo depois visualizar o endereço dele em um mapa. Bom, que existe a agenda de contatos e o Google Maps no Android todos já sabem, mas será que é possível utilizá-los e integrá-los em nossas aplicações? A resposta é sim. Integração é uma das palavras chaves em aplicações corporativas, e a arquitetura do Android foi criada justamente pensando nisso (LECHETA, 2013, p. 22).

Tal arquitetura constitui-se por camadas, sendo cada camada responsável pela gerência de seus processos. Compõem essa arquitetura: A Camada de Aplicações, local onde se acomodam todos os aplicativos relacionados ao sistema operacional; a Camada de Bibliotecas, que agrega, por exemplo, as bibliotecas C/C++, e as bibliotecas de multimídia; Camada de Runtime, localiza-se aí a máquina virtual Dalvik que possui a função de executar vários processos paralelamente e a Camada de Kernel Linux, núcleo do sistema operacional Android.

Outro fator que faz com que o Android seja ideal para a execução deste projeto é o seu sistema operacional, baseado no Linux, que conforme Lecheta (2013), proporciona o gerenciamento da memória e dos processos pelo próprio Android, o que permite a execução de diversas aplicações ao mesmo tempo. Tal funcionalidade permite, por exemplo, que o usuário possa executar as ações do aplicativo a que se refere este projeto em segundo plano enquanto acessa a internet ou realiza uma ligação por exemplo.

E, por fim há que se destacar a popularidade do Android, fator que também motiva sua utilização neste projeto, já que o sucesso dele entre as pessoas que possuem ou pretendem possuir um smartphone tem crescido de forma vertiginosa, o que faz com que essa platafor-

<sup>5</sup> Um kernel Linux é o que fornece uma camada fundamental de abstração de hardware, além de serviços base como gerenciamento de processos, memória e sistema de arquivos. O kernel é onde os drivers específicos de hardware são implementados — recursos como Wi-Fi e Bluetooth são encontrados aqui. A camada Android é projetada para ser flexível, com muitos componentes opcionais que dependem muito da disponibilidade do hardware específico de um dado dispositivo. Esses componentes incluem recursos como teclas sensíveis ao toque, câmeras, receptores GPS e acelerômetros. (ABLESON, 2012, p. 10).

ma tecnológica seja capaz de movimentar o mercado, isso porque é a primeira plataforma de aplicativos de código-fonte aberto para celulares e por isso é amplamente utilizada pelo grande mercado de telefonia celular global.

Essa versatilidade mercadológica que a plataforma apresenta - uma vez que o Android suporta uma multiplicidade de dispositivos de hardware e não somente dispositivos de ponta associados tipicamente com caros smartphones, o que o torna enormemente popular; sua arquitetura extremamente flexível, que possibilita a integração de aplicações e o seu sistema operacional, fundamentado no Linux configuram, portanto, os motivos pelos quais escolhemos implementar um aplicativo mobile para Android.

## Linguagens de programação

A metodologia de desenvolvimento desse projeto envolveu o uso da linguagem de programação C++, no que tange a programação da placa Arduino Uno e ainda da ferramenta App Inventor, escrita em Java.

Varejão (2004) define linguagem de Programação como: “um instrumento utilizado pelo profissional de computação para escrever programas, isto é, conjuntos de instruções a serem seguidas pelo computador para realizar um determinado processo.” (VAREJÃO, 2004, p. 01).

Sendo assim conclui-se que, por meio das linguagens de programação os seres humanos tem a possibilidade de criar algoritmos que ao serem traduzidos para a linguagem de máquina originam programas com as mais variadas funções. Varejão (2004) ainda pontua que as:

Linguagens de programação foram criadas para tornar mais produtivo o trabalho dos programadores. Logo, em última instância, o objetivo das linguagens de programação é tornar mais efetivo o processo de desenvolvimento de software. É importante enfatizar que esse processo existe tanto para tornar mais produtiva a geração e manutenção de software quanto para garantir que ele seja produzido atendendo a padrões de qualidade (VAREJÃO, 2004, p. 03).

Para que um programa escrito em uma Linguagem de programação seja executado é preciso que ele seja traduzido para a linguagem de máquina. Para tanto, faz-se necessário aplicar um programa ou método de implementação que gere o código e defina como o programa irá se comportar. Varejão (2004) especifica, assim três métodos gerais de implementação de Linguagens de Programação. São eles: a Compilação, a Interpretação e o método Híbrido.

A Compilação consiste em compilar e conseqüentemente traduzir integralmente o programa fonte para o código de máquina. Após ser traduzido, o programa pode ser executado diretamente. Varejão afirma que: “A grande vantagem desse método de tradução é a otimização da eficiência na execução dos programas. [...] Uma outra vantagem do processo de compilação é requerer apenas o código executável para que o programa possa ser executado.” (VAREJÃO, 2004, p. 16 e 17). Dentre as linguagens que adotam o método da compilação, destacamos a C++.

Outro método geral de implementação de Linguagens de Programação, descrito por Varejão (2004), é a Interpretação pura em que se verifica a presença de um programa interpretador que age como um simulador de um computador virtual e que entende as instruções da Linguagem de programação. Sobre esse método o autor pontua que o mesmo apresenta como vantagens:

a facilidade para prototipação, visto que se pode executar comandos e partes do programa assim que são construídos, verificando se atuam corretamente; a facilidade de depuração, visto que as mensagens de erro podem se referir diretamente ao código fonte; e a facilidade de escrever programas mais flexíveis, visto que o interpretador da LP está presente durante a execução (VAREJÃO, 2004, p. 17).

Por fim o autor apresenta, o terceiro método geral de implementação de Linguagens: o Híbrido que consiste na combinação da execução eficiente e da portabilidade de programas, com o uso da aplicação combinada dos dois métodos anteriores: a Compilação e a Interpretação. O método Híbrido possui em sua base um código intermediário, que não é específico de uma plataforma computacional, além de ser mais fácil de ser interpretado, pois compreende duas fases a compilação para um código intermediário e a interpretação deste código. Varejão (2004) nos informa que:

Embora ainda de execução mais lenta que o código compilado, a interpretação do código intermediário é muito mais rápida que a interpretação pura do código fonte porque as instruções do código intermediário são muito mais simples que as do código fonte e porque a maior parte das verificações de erro é realizada já na etapa de compilação. Por sua vez, como o código intermediário não é específico para uma plataforma, os programas já compilados para este código podem ser portados para as mais diferentes plataformas sem necessidade de adaptação ou mesmo recompilação, bastando que exista um interpretador do código intermediário instalado na plataforma onde se deseja executar o programa (VAREJÃO, 2004, p. 18).

Sendo assim não há, nesse método, uma tradução do código para código de máquina e sim uma simples interpretação do código intermediário. Cabe aqui ressaltar que o Java adota o método híbrido.

Para a programação lógica desse projeto foram utilizadas duas vertentes de abordagem: A programação da placa Arduino Uno e a programação do aplicativo Infragualuz instalado em um smartphone com Sistema Operacional Android.

No que tange a programação da placa Arduino Uno utilizou-se a linguagem de programação C++ (com pequenas modificações) no ambiente de programação IDE do Arduino.

Já para a programação do aplicativo Infragualuz, fez-se o uso de uma ferramenta escrita na linguagem híbrida, Java, o App Inventor que roda uma interface gráfica baseada em blocos que representam os elementos lógicos da programação.

## C++

Segundo Varejão (2004) a linguagem de programação C++ foi criada por Bjarne Stroustrup no Bell Labs. A LP (Linguagem de Programação) foi lançada com o intuito de ser uma extensão de C com orientação a objetos, embora tenha se tornado muito mais complexa.

A linguagem C++ apresenta uma nova forma de comentários. Além dos símbolos /\* e \*/, oferecidos em C, pode-se também utilizar o símbolo //, a partir do qual todo o texto até o final da linha é ignorado.

## Java

Embora, a metodologia desse trabalho não contemple a programação do aplicativo Infragualuz por meio da linguagem Java, considera-se importante destacar tal LP, uma vez que a mesma é a linguagem de programação em que é escrita a ferramenta App Inventor, utilizada aqui



para a programação do referido aplicativo.

Conforme Varejão (2004), a tecnologia Java foi criada pela Sun Microsystems e, é fortemente baseada em C++, embora seja mais simplificada. Foi originalmente elaborada para ser usada na construção de softwares para sistemas de controle embutido, mas acabou não sendo usada para este fim. É largamente utilizada em função de suas características de portabilidade e confiabilidade.

## **Ferramenta utilizada para auxílio no desenvolvimento**

Levando-se em consideração as necessidades didáticas da programação de dados, inovações práticas e metodológicas tem surgido no âmbito da Computação, em especial no que tange as Linguagens de programação. Foram apropriadas por este trabalho, as possibilidades didáticas da Linguagem de programação visual, especificamente do App Inventor, com o intuito de desenvolver-se um aplicativo Android+Arduino que contemplará três funções básicas, sendo elas: segurança, monitoramento de abastecimento de água e auxílio no consumo de energia elétrica, utilizando a placa Arduino. Desse modo, faz-se necessário apresentar-se aqui essa ferramenta específica.

Esta ferramenta foi utilizada tendo em vista sua praticidade no que tange a programação do aplicativo Android e ainda porque é possível utilizar todos os recursos instalados pelo fabricante no celular, haja vista que se trata de uma linguagem bastante rica que dá acesso aos sensores do aparelho. É importante observar a linguagem gráfica (App Inventor) adequa-se a escrita de programas não muito extensos, como o que programou-se aqui.

### **App inventor**

O App Inventor é um produto lançado pelo Google labs, que permite o desenvolvimento de aplicações para Android por meio do sistema de blocos que especificam a lógica das aplicações. Nesse sentido as aplicações são criadas visualmente por meio do estilo drag and drop.

O App Inventor possui uma licença open source, mas não é uma ferramenta multi-plataforma, adequando-se apenas ao universo dos celulares Androids.

De acordo com a pesquisadora da Universidade Federal Rural de Pernambuco (UFRPE), Tancicleide Gomes (2013):

O desenvolvimento de uma aplicação nesta ferramenta é realizado através de duas janelas: App Inventor Designer e Blocks Editor. A janela App Inventor Designer é executada a partir do navegador e permite criar visualmente a interface do usuário, ao clicar e arrastar os componentes da Palette, tais como botões, caixas de texto, figuras, animações, sons, dentre outros, para o Viewer (GOMES, 2013, p. 156).

Desse modo na janela Blocks Editor da ferramenta App Inventor maneja-se o comportamento dos componentes anteriormente selecionados na janela App Inventor Designer. Nesta janela, o desenvolvedor tem acesso a blocos conectáveis, manipulados por meio do esquema arrastar e soltar.

Ressalta-se que a área de blocos é a responsável por especificar o comportamento do aplicativo. Ainda conforme Gomes (2013), dentre os blocos que constituem o App Inventor, estão os:

- Control blocks – que possuem a função de controlar os eventos.
- Logic blocks - que possuem a função de auxiliar as operações lógicas, por meio de comparação entre números.
- Match blocks – neles estão contidas as operações matemáticas.
- Text blocks - que possuem a função de realizar manipulações ou operação com texto.
- List blocks - que possuem a função de realizar as operações com listas.
- Colors blocks - que possuem a função de criar cores diferentes das exibidas.
- Variables blocks - que possuem a função de manipulação de uma variável.

Importante ressaltar que embora tenha sido criado pelo Google Labs atualmente a ferramenta pertence ao MIT Labs, do Massachusetts Institute of Technology e ainda que trata-se de uma versão beta, que, portanto deverá evoluir muito ainda.

## Usabilidade

O conceito de usabilidade, embora não se aplique apenas ao universo computacional, é largamente utilizado nesse campo, existindo, inclusive, uma norma para tal, a ISO/IEC 9126, que é uma norma de qualidade para softwares. A ISO/IEC 9126 foi a primeira norma a definir o termo usabilidade, fornecendo um padrão de qualidade de softwares e segundo ela a Usabilidade é: “um conjunto de atributos de software relacionado ao esforço necessário para seu uso e para o julgamento individual de tal uso por determinado conjunto de usuários”. (ISO/IEC 9126)

Neste sentido a Usabilidade configura-se como o potencial que o produto/software possui em ser compreendido, em ser operado e em ser atraente ao público, ao usuário final.

Machado Neto (2013), com base em Shneiderman e Plaisant (2009) especifica um conjunto de métricas que constituem o caráter da Usabilidade:

**Tempo de aprendizagem.** Tempo necessário para que um usuário típico do sistema aprenda as ações relevantes que devem ser executadas, a fim de que sejam realizadas as principais atividades disponibilizadas pelo sistema.

**Desempenho (performance).** Tempo necessário para que o usuário execute as tarefas principais do sistema.

**Taxa de erros cometidos pelo usuário.** Contagem do número de erros que os usuários cometem para executar cada tarefa fundamental do sistema e respectiva listagem desses erros. Apesar de o tempo gasto pelo usuário para identificar um erro e em seguida se recuperar dele também ser contabilizado na métrica de desempenho, a taxa de erros é uma preocupação capital que precisa ser tratada de forma separada.

**Sedimentação do conhecimento por experiência.** Visa analisar a facilidade para que um usuário típico execute as funcionalidades principais do sistema uma hora, um dia ou uma semana depois de tê-las realizado pela primeira vez. Esta métrica tem um relacionamento estreito com a medida “tempo de aprendizagem” e é diretamente afetada pela frequência com a qual o usuário interage com as interfaces do sistema.

**Satisfação subjetiva.** Métrica para analisar o quanto os usuários gostaram de utilizar as interfaces do sistema. Essa pergunta pode ser respondida por meio de entrevistas com os usuários ou com questionários contendo questões na forma de escalas de satisfação ou mesmo na forma aberta, em que os usuários se sentem completamente livres para responderem com as próprias palavras (MACHADO NETO, 2013, p. 02 e 03).

Assim sendo, constata-se que nos dias atuais a crescente utilização dos sistemas computacionais leva a necessidade de adaptação desses sistemas a fim de que se tornem mais interativos e adequados aos variados tipos de usuários. Machado Neto (2013), nessa perspectiva, aponta para as diretrizes para criação de interfaces de dispositivos móveis objetivando um sistema que entre em consonância com as referidas métricas de Usabilidade e que consiga efetivamente auxiliar ao usuário na execução das tarefas. Conforme o autor para o desenvolvimento de uma interface agradável é preciso:

- 01- Destaque a principal atividade da aplicação e garanta os subsídios necessários para que o usuário complete qualquer tarefa. [...]
- 02- Invista os maiores esforços nos fatores da aplicação mais importantes do ponto de vista do usuário. [...]
- 03- Pense no design da interface como uma atividade a ser preenchida de cima para baixo. [...]
- 04- Disponibilize um caminho lógico para o usuário. [...]
- 05- Torne a interação fácil e óbvia. [...]
- 06- Facilite a entrada de dados. [...]
- 07- Estimule a conectividade e o comportamento colaborativo. [...]
- 08- Torne a interface mais realista possível. [...]
- 09- Dê suporte à mudança de orientação. [...]
- 10- Mantenha o usuário ciente de qualquer ação. [...]
- 11- Dê controle ao usuário. [...]
- 12- Crie uma página de ajuda. [...]
- 13- Aposte em um design minimalista. [...]
- 14- Use imagens e gráficos em alta definição e editados profissionalmente. [...]
- 15- Use componentes na medida adequada (MACHADO NETO, 2013, p. 14 a 17).

Foi, portanto observando tais diretrizes que foi proposto a elaboração do aplicativo Infraqualuz, objeto desse trabalho, haja vista que as interfaces de usuário não são apenas um meio de comunicação entre homem e o smartphone, mas constituem um forte instrumento de interação entre eles.

## **Automação residencial**

De acordo com Bolzani (2004), a automação residencial define-se como um conjunto de tecnologias que auxiliam na gerência e execução de tarefas que compõem o dia-a-dia doméstico.

A automação residencial possui a finalidade de tornar mais prática à vida moderna, que impõe uma rotina acelerada as pessoas, que por vezes acabam não tendo condições de operar

tarefas básicas em seus domicílios. Desse modo, a automação residencial entra em consonância com essa imposição da vida urbana moderna, já que visa exatamente à facilitação da rotina doméstica dos moradores. É, pois nesse sentido, que foram pensados os objetivos desse trabalho, que versam sobre o desenvolvimento um aplicativo que integre o Sistema Operacional Android e a plataforma Arduino com o intuito de gerir três funções domésticas básicas: a segurança, o monitoramento de abastecimento de água e o controle do consumo de energia elétrica, proporcionando dessa maneira um maior nível de segurança, comodidade e de consumo racional de energia elétrica.

Há algum tempo atrás a automação residencial era considerada item de luxo, direcionado apenas as camadas detentoras de maior poder aquisitivo, no entanto nos dias atuais o que se observa é que esta tecnologia vem ganhando cada vez mais espaço, tendo em vista a redução de custos proporcionada pela inserção de novas tecnologias, open source e de softwares livres, como o Arduino, por exemplo, base para a elaboração desse projeto.

Segundo Adonis Accardi e Eugeni Dodonov (2012), baseados em Almeida (2009) vários são os elementos envolvidos no processo de automação residencial, os autores afirmam que estão compreendidos nesse processo Controladores, Sensores, Atuadores, Barramentos e Interfaces. Desse modo pontuam que:

Os Controladores controlam os dispositivos automatizados (sensores e atuadores). Monitora as informações dos sensores, podendo enviar comandos para que um atuador ative ou desative algum equipamento. De maneira geral podem possuir interfaces independentes, na forma de um controle remoto, ou serem sofisticadas centrais de automação (ALMEIDA, 2009). Os Sensores são os dispositivos que detectam estímulos, medem e monitoram grandezas físicas e eventos (temperatura, umidades etc.), convertendo-as em um valor passível de manipulação por sistemas computacionais. São eles que encaminham as informações aos controladores sobre algum evento, para que os controladores possam enviar os comandos adequados para os atuadores (ALMEIDA, 2009). Os Atuadores são dispositivos eletro- mecânicos, que recebem os comandos do sistema de automação e ativam os equipamentos automatizados. São os módulos de acionamento ligados entre a rede elétrica e os equipamentos (ALMEIDA, 2009). Existem atuadores para portas, janelas, persianas, fechadura magnética, sirene, indicadores luminosos, etc. O Barramento é o meio físico responsável pelo transporte das informações (rede elétrica, telefônica etc.). As Interfaces são os dispositivos ou mecanismos (navegador de internet, celular, painéis, controles remotos, interruptores etc.) que permitem ao usuário visualizar as informações e interagir com o sistema de automação (ACCARDI; DODONOV, 2012, p. 157).

O que se observa, portanto é que um sistema de automação residencial é elaborado com base na utilização tanto de dispositivos eletrônicos quanto de redes de dados, o que garante o controle a distância de equipamentos bem com a automatização de tarefas em uma residência. Nesta perspectiva este trabalho está motivado em explorar as possibilidades do desenvolvimento de um sistema de automação residencial de baixo custo baseado na integração entre a Eletrônica com os Sistemas de Informação, com base no uso da plataforma Arduino e do Sistema Operacional Android.

## **Aplicações similares e funcionalidade do Infragualuz**

O atual mercado de sistemas mobiles voltados para a automação residencial, conta hoje com alguns aplicativos que possuem finalidades parecidas com as do aplicativo Infragualuz.

O que nota-se, porém é a existência de projetos que contemplam a funcionalidade de acender e apagar a luz em uma residência através de aplicativo Android usando Arduino. No en-

tanto o que se pretende aqui é projetar um sistema de monitoramento residencial que terá como objetivo proporcionar ao usuário maior comodidade em seu dia-a-dia haja visto, que o mesmo poderá saber se as lâmpadas de sua residência estão acessas ou apagadas por meio do aplicativo Infragualuz, que lhe proporcionará visualizar os cômodos de sua casa em uma interface gráfica onde cores indicarão o status das lâmpadas, podendo o usuário acendê-las ou apagá-las a seu critério.

Outra função do sistema a ser desenvolvido é o monitoramento da segurança residencial via cerca de infravermelho. Tal monitoramento se dará por meio de alertas enviados (em caso de invasão) ao aplicativo do usuário que poderá, ainda, por meio de seu smartphone ou tablet ligar ou desligar a referida cerca.

Esse sistema pretende ainda possibilitar ao usuário que monitore o fluxo de abastecimento de água em sua residência uma vez que o mesmo será informado em seu aplicativo acerca do status de abastecimento em sua casa. Para as três funcionalidades do aplicativo Infragualuz será realizado o uso da associação entre a plataforma Arduino, o sistema operacional Android e a IDE do Arduino.

Nota-se assim que o que se pretende aqui é o desenvolvimento complexo de monitoramento residencial que abarcará uma implementação tanto física quanto programada que se baseia em pesquisas anteriores, mas que avança nos aspectos acima mencionados.

# DESENVOLVIMENTO DO APLICATIVO, PROGRAMAÇÃO DA PLACA DE ARDUINO E MONTAGEM DOS CIRCUITOS

Este capítulo pretende apresentar as etapas do processo de desenvolvimento do aplicativo proposto, o Infragualuz, bem como a programação da placa de Arduino e a montagem dos circuitos, itens que se complementam na composição do todo desse trabalho.

Para tanto, considerou-se importante a divisão do conteúdo desse capítulo em: parte lógica que abarca os softwares aqui implementados, ou seja a parte abstrata dessa monografia, composta pela programação da placa Arduino Uno e pela programação do aplicativo mobile. E parte física que se compõe dos circuitos montados para o monitoramento residencial via infravermelho; o monitoramento do abastecimento residencial de água e para o monitoramento do status das lâmpadas em uma residência. Antes, porém detalhar-se-á o projeto desenvolvido:

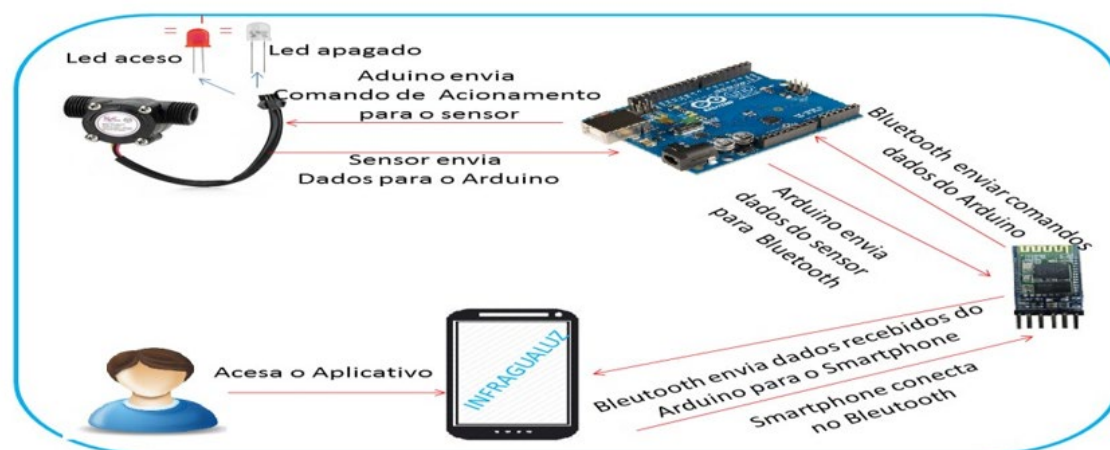
## O projeto

Para a realização deste trabalho, foram utilizados os seguintes componentes eletrônicos: Placa Arduino Uno, Protoboard de 170 furos, LEDs de 5 mm, jumpers, Resistores 330 ohms, Módulo bluetooth, Push buttons, além de um Sensor de fluxo de água para os sistema de monitoramento do abastecimento e um Módulo LDR, um Diodo Laser e um Buzzer para o protótipo da cerca de segurança.

A seguir tem-se as Figuras 05, 06 e 07 referentes aos diagramas de cada uma das funcionalidades do aplicativo Infragualuz.

Na Figura 05, tem-se o diagrama de funcionamento referente ao sistema de controle de abastecimento de água, uma vez que, através de um aparelho com o Sistema Operacional Android e com o aplicativo Infragualuz já instalado, o aplicativo será capaz de sinalizar através da mudança do status de verificação quando da interrupção no abastecimento de água ou quando da sua normalização onde essa verificação se dá por meio de um Sensor de Fluxo de Água, assim a placa Arduino (dotada de um módulo Bluetooth) é capaz de estabelecer essa comunicação com o aplicativo Infragualuz.

Figura 5 - Diagrama geral - Água

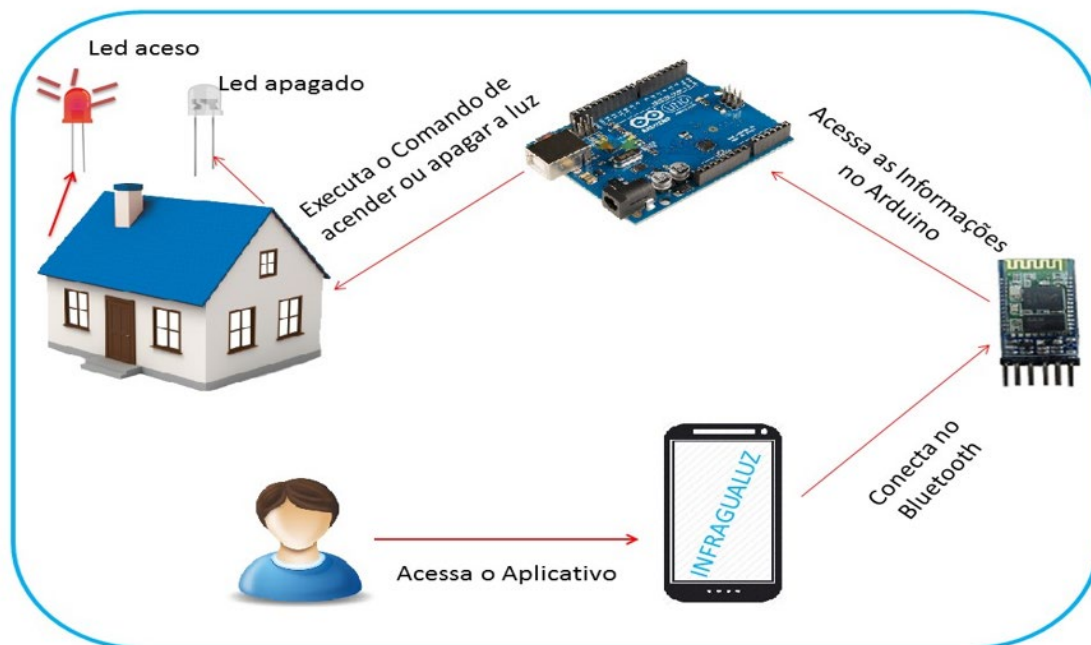


Fonte: PRÓPRIA, 2017.

Na Figura 6, tem-se o diagrama de funcionamento referente ao sistema de economia

de energia composto por um aplicativo – O Infragualuz – que, por sua vez estará instalado em um aparelho dotado do Sistema Operacional Android. Esse aplicativo será capaz de sinalizar da ocorrência de luzes acesas em uma residência e por meio de uma interface amigável o usuário será capaz de realizar a comutação do estado das mesmas, ou seja, acender ou apagar as luzes. Uma vez que o aplicativo Infragualuz realiza a comunicação com a placa Arduino se conectando ao módulo Bluetooth instalado na placa.

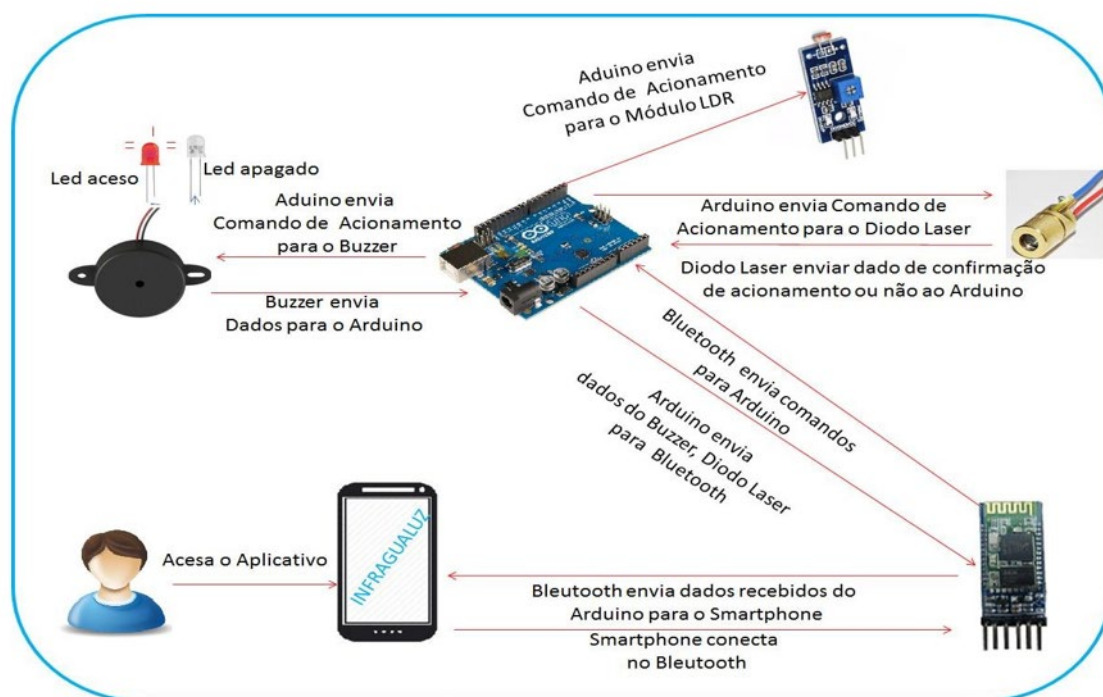
**Figura 6- Diagrama geral – Lâmpadas**



**Fonte: PRÓPRIA, 2017.**

Na Figura 07, nota-se o diagrama de funcionamento referente ao Sistema de Segurança, uma vez que, através de um aparelho com o Sistema Operacional Android e com o aplicativo Infragualuz já instalado, o aplicativo será capaz de inicializar ou interromper o funcionamento da cerca. Esse sistema é dotado de um diodo laser que está conectado com um módulo LDR (que funciona basicamente através da incidência de luz em seu material) que ao ser interceptado fará soar um Buzzer (aparelho capaz de emitir sons de alertas) avisando da ocorrência de interceptação, e no aplicativo Infragualuz também ocorrerá à sinalização de interceptação.

Figura 7 - Diagrama geral – Cerca de segurança



Fonte: PRÓPRIA, 2017.

## Parte lógica

Levando-se em consideração um dos objetivos referentes ao âmbito lógico do projeto, qual seja o de implementar um aplicativo que faça uso do Android e da IDE do Arduino, como softwares principais com a finalidade de contemplar as funções de segurança, monitoramento de abastecimento de água e auxílio no consumo de energia elétrica, a metodologia aqui utilizada abarca duas vertentes:

1 - A programação da placa Arduino Uno com uso da LP C++, no ambiente de programação IDE;

2 - A programação de aplicativo mobile, desenvolvido para Android com o uso da ferramenta App Inventor.

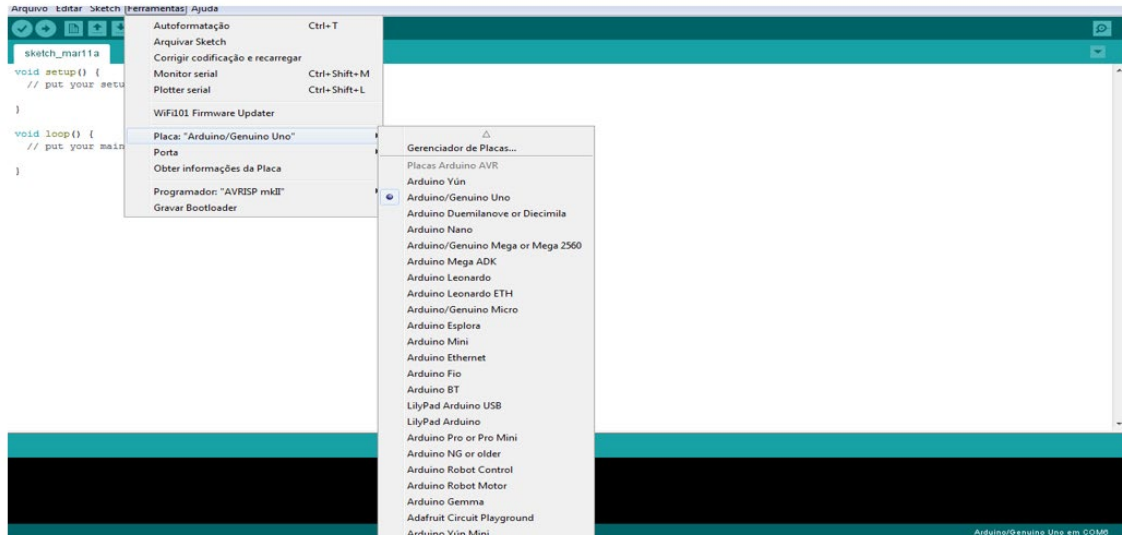
### A programação da placa Arduino Uno

Para iniciar a programação da placa Arduino Uno, foi necessário efetuar a instalação dos drivers, responsáveis pela habilitação das portas para conexão com o computador, no Windows 7. Em seguida procedemos ao download da IDE do Arduino, ambiente de programação que permite a escrita do programa na linguagem C++.

Na sequência procedeu-se a escolha da porta e do tipo de placa a serem utilizados, conforme a Figura 8 e a Figura 9:

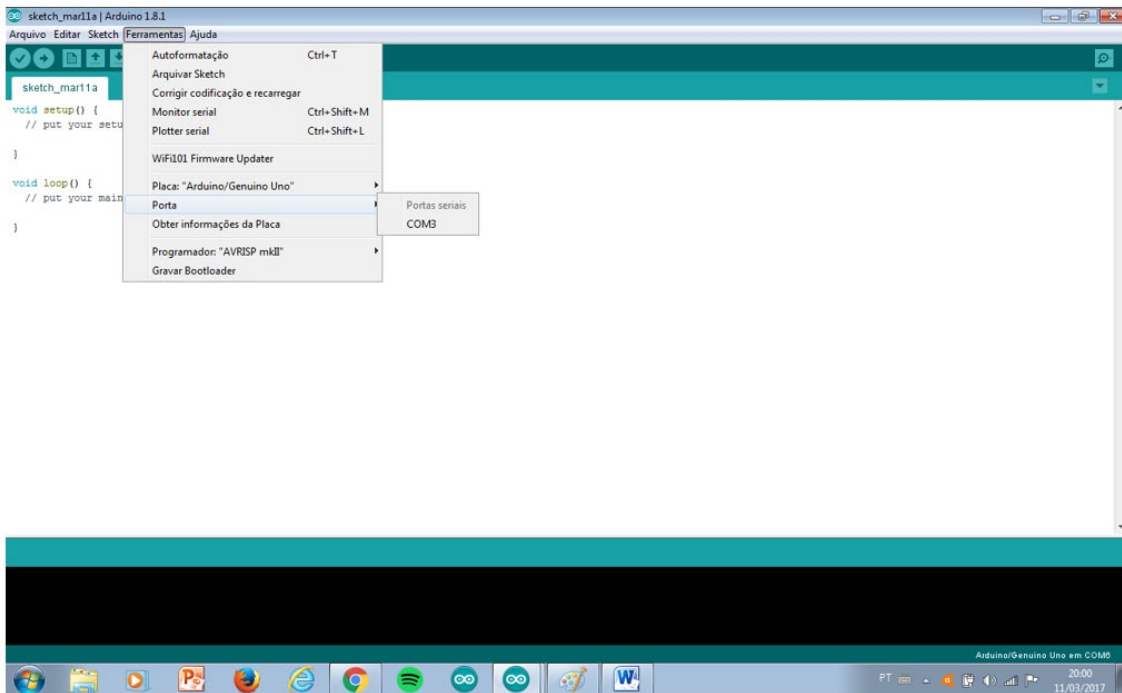


Figura 8 - IDE do Arduino (Seleção de placa)



Fonte: PRÓPRIA, 2017.

Figura 9 - IDE do Arduino (Seleção de porta)



Fonte: PRÓPRIA, 2017.

Após ser conectada ao computador por meio do cabo USB, a placa já está pronta para receber o código (Sketch).

O código/Sketch foi então inserido no IDE que apresenta duas funções principais, são elas: o `setup()`, que é executado apenas uma vez com a finalidade de inicializar as bibliotecas, as variáveis<sup>6</sup> e as pinagens<sup>7</sup> e o `loop()` que é responsável pela execução da maioria dos programas e, é continuamente exe-

<sup>6</sup> São posições na memória de programa do ARDUINO marcadas com um nome e o tipo de dado que irão ser armazenados nessa posição. Essa posição de memória pode receber valores durante a execução do programa e podem ser alterados a qualquer momento pelo programa e deve ser respeitado o tipo de dado da variável. Disponível em: <https://fbseletronica.wordpress.com/2013/05/16/curso-arduino-variaveis>.

<sup>7</sup> São pinos de entrada e saída dos componentes eletrônicos e do Arduino.

cutado enquanto o Arduino estiver sendo alimentado. Com isso passa-se então ao teste a fim de se verificar se houve algum erro no programa.

## Os Códigos de programação das placas Arduino

A seguir são descritos os códigos programados em C++ para cada uma das placas de Arduino Uno que compõem esse projeto, haja vista que foram utilizadas três placas, sendo uma para cada protótipo a ser implementado.

### Código: Protótipo de monitoramento do status das lâmpadas

A programação da placa de Arduino para o protótipo de monitoramento do status das lâmpadas de uma residência que se beneficiará pelo uso de sensores de luminosidade que indicarão a presença de lâmpadas acessas para fins de economia de energia elétrica, seguiu os seguintes parâmetros, conforme a Figura 10:

Figura 10 - Código: Protótipo de monitoramento do status das lâmpadas

```
sketch_mar12a$  
  
int led1 = 10;  
int led2 = 9;  
boolean status1 = false;  
boolean status2 = false;  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(led1, OUTPUT);  
  pinMode(led2, OUTPUT);  
  digitalWrite(led1, LOW);  
  digitalWrite(led2, LOW);  
}  
  
void loop() {  
  char comando = Serial.read();  
  int chavel = digitalRead(4);  
  int chave2 = digitalRead(3);  
  
  if (chavel == HIGH) {  
    if (status1) {  
      digitalWrite(led1, LOW);  
      status1 = false;  
      Serial.println("1desligado");  
    } else {  
      digitalWrite(led1, HIGH);  
      status1 = true;  
      Serial.println("1ligado");  
    }  
  }  
  
  if (chave2 == HIGH) {  
    if (status2) {  
      digitalWrite(led2, LOW);  
      status2 = false;  
      Serial.println("2desligado");  
    } else {  
      digitalWrite(led2, HIGH);  
      status2 = true;  
      Serial.println("2ligado");  
    }  
  }  
  
  delay(250);  
  
  if(comando == 'L') {  
    if (status1) {  
      digitalWrite(led1, LOW);  
      status1 = false;  
      Serial.println("1desligado");  
    } else {  
      digitalWrite(led1, HIGH);  
      status1 = true;  
      Serial.println("1ligado");  
    }  
  }  
  
  if(comando == 'F') {  
    if (status2) {  
      digitalWrite(led2, LOW);  
      status2 = false;  
      Serial.println("2desligado");  
    } else {  
      digitalWrite(led2, HIGH);  
      status2 = true;  
      Serial.println("2ligado");  
    }  
  }  
  
  if(comando == 'S') {  
    if (status1) {  
      Serial.println("1ligado");  
    } else {  
      Serial.println("1desligado");  
    }  
    delay(250);  
    if (status2) {  
      Serial.println("2ligado");  
    } else {  
      Serial.println("2desligado");  
    }  
    delay(250);  
  }  
}
```

Fonte: PRÓPRIA, 2017.

Segue descrição das configurações utilizadas no código:

- O Bloco 01 contém a definição dos pinos para conexão dos LEDs com Arduino;
- Em seguida a definição da função boolean para retorno do estado (acesso ou apagado) de cada LED;
- Serial.begin (9600) função responsável pela inicialização da comunicação serial do Arduino na velocidade 9600 bps(bits por segundo);
- A definição dos pinos dos LEDs como pinos de saída;

- A função responsável por determinar que os LEDs inicializaram no estado LOW (desligado);
- Função `Serial.read` usada para ler todos os dados que chegam na serial do Arduino;
- Bloco composto com a função `digitalRead` responsável por realizar a leitura e controle de cada LED nele designado;
- Bloco contendo a função de igualar o comando `chave1` (`chave2`, `chave3`) com a função `HIGH` para determinar o estado em que o LED se encontrará quando o comando `chave` for ativado. A função `if` é dotada do comando `status1` (`status2`, `status3`) para realizar a verificação do estado do LED levando em consideração a variável booleana, caso o LED esteja acessado a função `digitalWrite(LED1, LOW)` realiza a tarefa de desligar o LED e o comando `status1` (`status2`, `status3`) = `false` mudará para `false` para que a próxima vez que ele for executado possa se encontrar no estado desligado, já a função `Serial.println` envia a string para o serial informando que o LED foi desligado.
- Caso contrário as afirmações anteriores o `digitalWrite(led1, HIGH)` realiza a tarefa de ligar o LED e o comando `status1` (`status2`, `status3`) = `true` mudará para `true` para que a próxima vez que ele for executado possa se encontrar no estado ligado, já a função `Serial.println` envia a string para o serial informando que o LED foi ligado. E a função `delay` responsável pela intermitência do LED ao ser ativado.
- Este bloco tem conexão com o desenvolvimento do aplicativo realizado na ferramenta App Inventor uma vez que a execução dessa função é realizada no aplicativo para ativação e desativação dos LEDs.

### **Código: Protótipo de monitoramento do abastecimento residencial de água**

A programação da placa de Arduino para o protótipo de monitoramento do abastecimento residencial de água que se beneficiará pelo uso de sensores de umidade que indicarão a passagem do fluxo de água pelo referido protótipo, seguiu os seguintes comandos como mostra a Figura 11:

**Figura 11 - Código: Protótipo de monitoramento do abastecimento de água**

```
int con_
double litros;
double quant;
float vazao;
int quant_Pulso;
int i=0;
int led1_verde = 5;
int led2_vermelho = 6;

void setup(){
  Serial.begin(9600);
  attachInterrupt(0, incpulso, RISING); // interrupção RISING
                                     // = borda de subida
  pinMode(8, INPUT);
  Serial.println("Inicio"); //Imprime Inicio na serial
  pinMode(led1_verde, OUTPUT);
  pinMode(led2_vermelho, OUTPUT); }

void loop (){
  quant_Pulso = 0;
  sei();
  delay (2000);
  cli();
  quant= cont_pulso; //litros = (quant)/330;
  vazao = quant_Pulso / 7.5;
  i++;

  Serial.print(" L/min - ");
  Serial.print(i);
  Serial.println("s");
  Serial.println("-----");

  if (vazao > 5) {
    digitalWrite(led1_verde, HIGH);
    digitalWrite(led2_vermelho, LOW);
  }

  if (vazao < 4) {
    digitalWrite(led1_verde, LOW);
    digitalWrite(led2_vermelho, HIGH);
  }

  if (Serial.available())
  {
    if (Serial.read()=='A'){
      cont_pulso=0;
    }
  }
}

void incpulso (){
  cont_pulso++;
}
```

**Fonte: PRÓPRIA, 2017.**

Segue descrição das configurações utilizadas no código:

- O Bloco 01 contém a definição dos pinos para conexão dos LEDs com Arduino;
- Bloco contendo a definição da função boolean para retorno do estado (acesso ou apagado) de cada LED;
- Serial.begin (9600) função responsável pela inicialização da comunicação serial do Arduino na velocidade 9600;
- Bloco contendo a definição dos pinos dos LEDs como pinos de saída;
- Bloco contendo a função responsável por determinar que os LEDs inicializaram no estado LOW (desligado);
- Função Serial.read usada para ler todos os dados que chegam na serial do Arduino;
- Bloco composto com a função digitalWrite responsável por realizar a leitura e controle de cada LED nele designado;
- Bloco contendo a função de igualar o comando chave1 (chave2, chave3) com a função HIGH para determinar o estado em que o LED se encontrará quando o comando chave for ativado. A função if é dotada do comando status1 (status2, status3) para realizar a verificação do estado do LED levando em consideração a variável booleana, caso o LED esteja acesso a função digitalWrite(LED1, LOW) realiza a tarefa de desligar o LED e o comando status1 (status2, status3) = false mudará para false para que a próxima vez que ele for executado possa se encontrar no estado desligado, já a função Serial.println envia a string para o serial informando que o LED foi desligado.
- Caso contrário as afirmações anteriores o digitalWrite(led1, HIGH) realiza a tarefa de

ligar o LED e o comando status1 (status2, status3) = true mudará para true para que a próxima vez que ele for executado possa se encontrar no estado ligado, já a função Serial.println envia a string para o serial informando que o LED foi desligado. E a função delay responsável pela intermitência do LED ao ser ativado

- Este bloco tem conexão com o desenvolvimento do aplicativo realizado na ferramenta App Inventor uma vez que a execução dessa função é realizada no aplicativo para ativação e desativação dos LEDs.

## Código: Protótipo de cerca de segurança

A programação da placa de Arduino para o protótipo de monitoramento de cerca de segurança baseada em um circuito eletrônico de monitoramento residencial via infravermelho, seguiu os seguintes parâmetros como mostra a Figura 12:

**Figura 12 - Código: Protótipo de monitoramento da cerca de segurança**

```
int laser = 10;
int led1 = 11;
int led2 = 12;
int portaLDR = A5;
const int buzzer = 13;
boolean status1 = false;
void setup() {
  Serial.begin(9600);
  pinMode(laser, OUTPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  digitalWrite(laser, LOW);
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  pinMode(buzzer, OUTPUT);
}
void loop() {
  char comando = Serial.read();
  int estado = analogRead(portaLDR);
  int chave2 = digitalRead(3);

  //LIGAR E DESLIGAR LASER
  if (chave2 == HIGH) {
    if (status1) {
      // executa essa parte se ele estiver ligado
      digitalWrite(laser, LOW);
      digitalWrite(led1, LOW);
      status1 = false;
      Serial.println("Idesligado");
    }
    else {
      // executa essa parte se ele estiver desligado
      digitalWrite(laser, HIGH);
      digitalWrite(led1, HIGH);
      status1 = true;
      Serial.println("Iligado");
    }
  }
  delay(250);
  if (estado > 600) { //estado = LDR E BUZZER
    tone(buzzer, 200);
    delay(500);
    digitalWrite(led2, HIGH);
  }
  else {
    if (chave2 == HIGH) {
      // executa essa parte se ele estiver ligado
      noTone(buzzer);
      digitalWrite(led2, LOW);
      Serial.println("Icom");
    }
    else {
      delay(250);
    }
  }
  if (comando == 'L') {
    if (status1) {
      digitalWrite(laser, LOW);
      digitalWrite(led1, LOW);
      noTone(buzzer);
      digitalWrite(led2, LOW);
      status1 = false;
      Serial.println("Idesligado");
    }
    else {
      digitalWrite(laser, HIGH);
      digitalWrite(led1, HIGH);
      status1 = true;
      Serial.println("Iligado");
    }
  }
  if (comando == 'a') {
    if (status1) {
      Serial.println("Iligado");
    }
    else {
      Serial.println("Idesligado"); // envia a string para informar
      //que o led está desligado
    }
  }
  delay(250);
}
```

**Fonte: PRÓPRIA, 2017.**

Segue descrição das configurações utilizadas no código:

- O bloco 01 contém a definição dos pinos para conexão do diodo laser, dos LEDs, LDR e Buzzer com Arduino, definição da função boolean para retorno do estado (acesso ou apagado) de cada diodo laser e LED;
- Serial.begin (9600) função responsável pela inicialização da comunicação serial do Arduino na velocidade 9600;
- Bloco contendo a definição dos pinos do diodo laser, LEDs e Buzzer como pinos de saída no Arduino;
- Bloco contendo a função responsável por determinar que os LEDs inicializaram no estado LOW (desligado);
- Função Serial.read usada para ler todos os dados que chegam na serial do Arduino;

- Função analogRead responsável por identificar e ler os valores fornecidos pelo LDR;
- Bloco composto com a função digitalRead responsável por realizar a leitura e controle do diodo laser e do LED nele designado;
- Bloco contendo a função de igualar o comando chave2 com a função HIGH para determinar o estado em que o LED se encontrará quando o comando chave2 for ativado.
- A função if é dotada do comando status1 para realizar a verificação do estado (acesso ou apagado) do laser e do LED levando em consideração a variável booleana (boolean). Se o laser e o LED estiverem acessos a função digitalWrite(laser, LOW), digitalWrite(led1, LOW) realiza a tarefa de desligar o laser e o LED e o comando status1 = false mudará para o modo false assim a próxima vez que ele for executado ele pode ser encontrado no estado desligado, já a função Serial.println envia a string para o serial informando que o laser e o LED foram desligados.
- Caso contrário as afirmações anteriores, a função digitalWrite(laser, HIGH), digitalWrite(led1, HIGH) realiza a tarefa de ligar o diodo laser e o LED, e o comando status1 = true mudará para o modo true assim a próxima vez que ele for executado ele pode ser encontrado no estado ligado, já a função Serial.println envia a string para o serial informando que o laser e o LED foram ligados. E a função delay é responsável pela intermitência do LED ao ser ativado
- Neste bloco a função if é responsável por comparar o comando estado responsável por obter os valores fornecidos pelo LDR na função analogRead com o valor esperado para que o LDR possa entrar em modo de ativação, ou seja, quando o valor passado não for mais encontrado na função analogRead o Buzzer entrará em modo ativo (emitindo um som) e o LED2 se acenderá.
- Neste bloco o if é responsável comparar o estado presente da chave com a função HIGH que aponta o estado atual do Buzzer e do LED2, assim ao executar essa tarefa tanto o Buzzer quanto o LED2 serão desligados.
- Este bloco tem conexão com o desenvolvimento do aplicativo realizado na ferramenta App Inventor uma vez que a execução dessa função é realizada no aplicativo para ativação e desativação do laser, Buzzer e dos LEDs.

## A programação do aplicativo mobile – Infragualuz

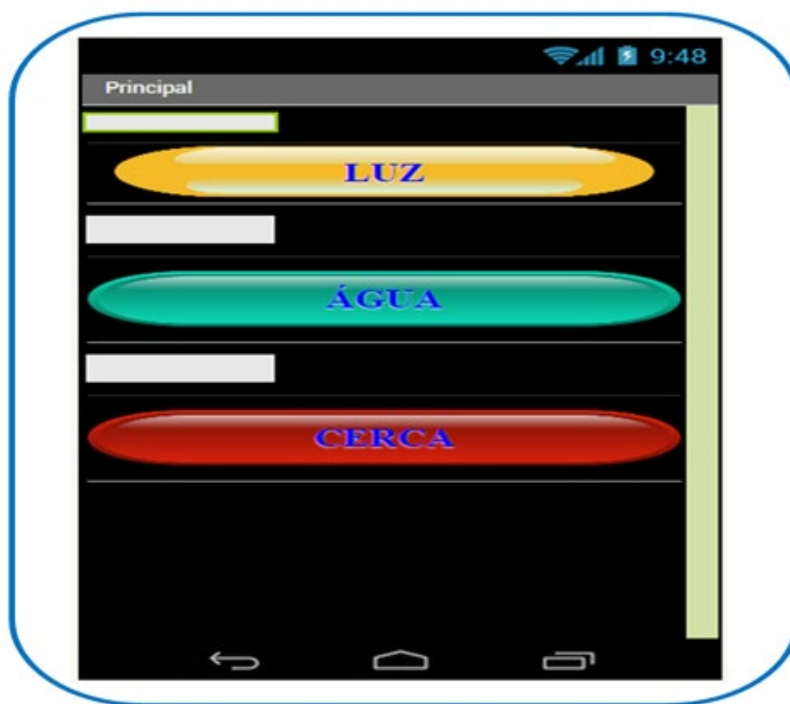
Para a programação do aplicativo, ora denominado Infragualuz, fez-se o uso da ferramenta App Inventor mencionada anteriormente. Ressalta-se que o aplicativo será instalado em um aparelho smartphone da marca LG – modelo K10. Dessa maneira passa-se a descrição da programação do aplicativo.

O aplicativo será composto por uma tela principal que contem os botões de acesso as telas de controle das funções do aplicativo. As telas de controle são três: controle do abastecimento de água, controle da cerca de segurança e controle do status das lâmpadas (essa tela dá acesso às telas de controle referentes aos cômodos da residência: Sala, Quarto, Cozinha, Banheiro, Varanda e Quintal).

Para iniciar a programação foi necessário o login na página do MIT App Inventor 2<sup>8</sup>, passa-se então a página de designer, onde se elabora o layout do aplicativo. Nessa página encontram-se disponíveis uma lista de funcionalidades que auxiliam na criação do app. A seguir constam as Figuras 13, 14, 15 e 16 relativas às telas da interface gráfica que procurou atender aos princípios e métricas de Usabilidade:

Na Figura 13, tem-se a tela principal do aplicativo Infragualuz com as três funcionalidades do mesmo.

**Figura 13 - Tela principal do Infragualuz**

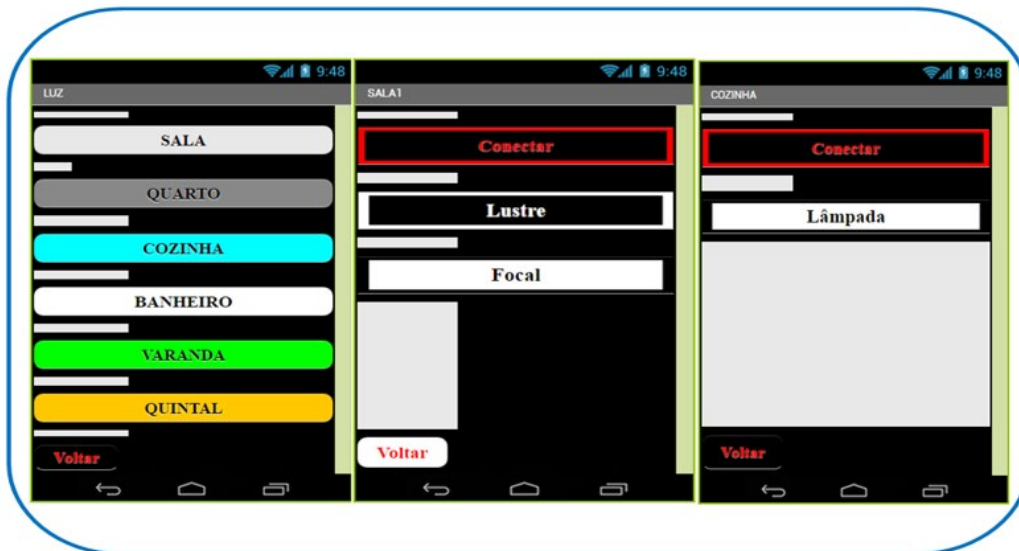


**Fonte: PRÓPRIA, 2017.**

A Figura 14 apresenta a tela referente ao sistema de economia de energia do aplicativo Infragualuz, com a disposição dos cômodos referente ao sistema projetado.

<sup>8</sup> Disponível em: <http://ai2.appinventor.mit.edu/>. Acesso em: 03/01/17.

Figura 14 - Tela monitoramento do status das lâmpadas



Fonte: PRÓPRIA, 2017.

Na Figura 15, tem-se a tela referente ao sistema de abastecimento de água do aplicativo Infragualuz.

Figura 15 - Tela monitoramento do abastecimento de água



Fonte: PRÓPRIA, 2017.

A Figura 16 apresenta a tela referente ao sistema de Segurança do aplicativo Infragualuz, onde se nota o botão central capaz de acionar e desativar a cerca e também a sinalização da mesma, caso ocorra à interceptação.



Figura 16 - Tela comandos da Cerca de segurança



Fonte: PRÓPRIA, 2017.

Após definida a interface gráfica, passa-se a página de blocks, onde, por meio do sistema drag and drop configura-se a linguagem visual através do agrupamento de blocos que contém funções e condições para o desenvolvimento e funcionamento do aplicativo.

### Os Códigos de programação do aplicativo

Abaixo seguem os códigos programados com o auxílio da ferramenta App inventor para o desenvolvimento do aplicativo.

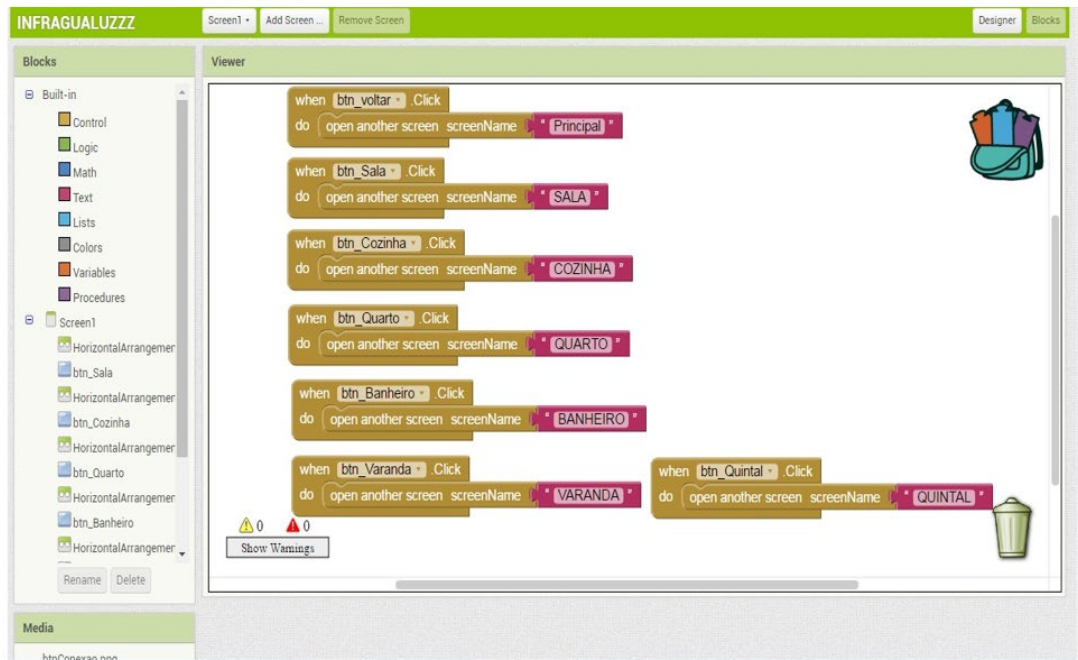
#### Código: Monitoramento do status das lâmpadas

Dentro da construção da lógica presente no protótipo do monitoramento no consumo de energia elétrica, tem-se no primeiro instante as funções para navegação nas janelas do primeiro protótipo, onde a função:

When btn\_voltar. Click, em conjunto a, do open another screen screenName (“ ”) “Principal” fazem com que ao serem chamadas elas retornem para a página a qual foi solicitada, neste caso a tela denominada “Principal”, e essa mesma tarefa é aplicada aos demais blocos.

Conforme nota-se na Figura 17 em que há a tela de programação referente aos comandos do aplicativo para acesso e retrocesso dos menus.

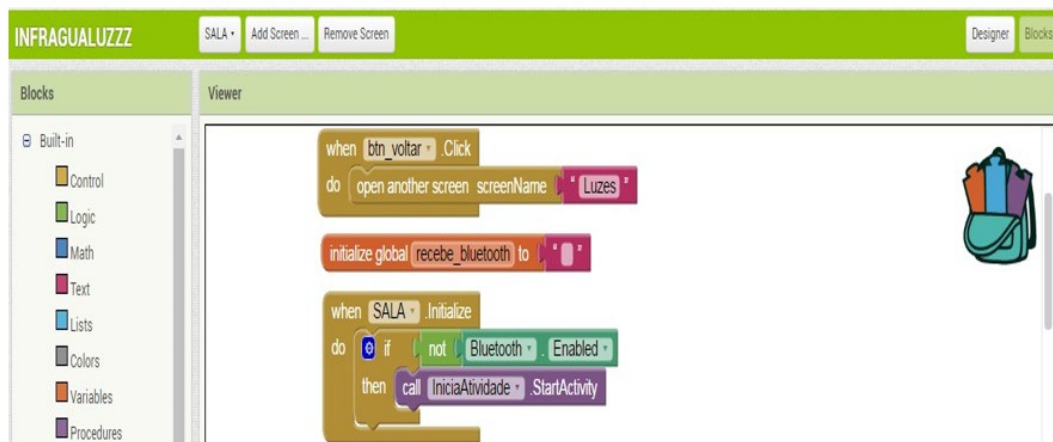
**Figura 17 - Código: Tela inicial do monitoramento do status das lâmpadas**



Fonte: PRÓPRIA, 2017.

Na Figura 18, tem-se a tela de programação referente aos comandos do aplicativo de monitoramento de economia de energia.

**Figura 18 - Código: Tela do monitoramento da sala**



Fonte: PRÓPRIA, 2017.

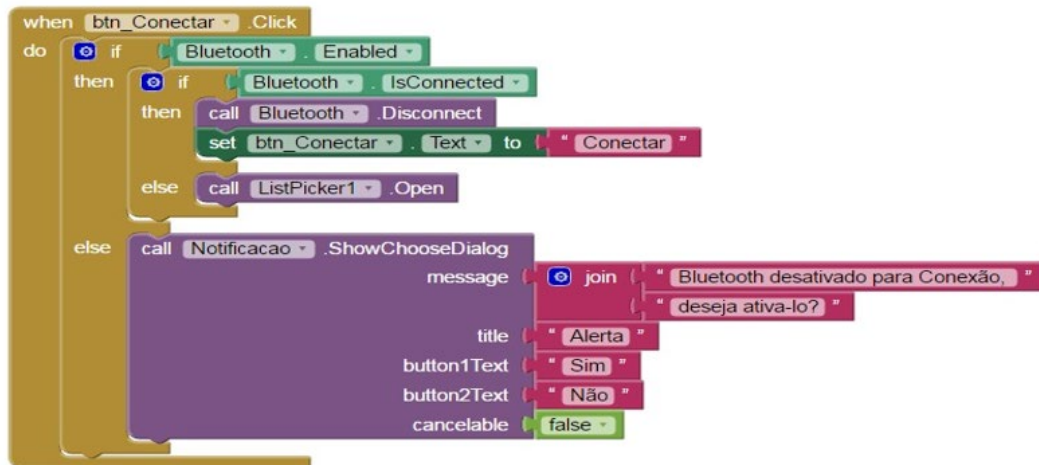
No primeiro bloco da Figura 18, nota-se que a função When btn\_voltar. Click, em conjunto a, do open another screen screenName “Screen1” fazem com que ao serem chamadas elas retornem para a página a qual foi solicitada, neste caso a tela denominada “Screen1”;

Segundo bloco: Initialize global recebe\_bluetooth to variável responsável por receber as informações que chegam até o bluetooth;

Terceiro bloco: responsável por tentar habilitar a função bluetooth do aparelho usado;

Na Figura 19, tem-se a tela de programação referente à configuração para conexão do módulo Bluetooth.

Figura 19 - Código: Tela de configuração do Bluetooth



Fonte: PRÓPRIA, 2017.

Na Figura 19 nota-se no quarto bloco: a função Bluetooth . Enable é responsável pela habilitação do bluetooth do aparelho Android do usuário. O comando call Notificacao.ShowChooseDialog é responsável pela seguinte tarefa: se o usuário antes de realizar a ativação do bluetooth no seu aparelho Android, tentar ativar o botão de “Conectar” na interface do seu aplicativo um alerta será exibido informando: Bluetooth não está ativado para realizar uma conexão, deseja ativar o mesmo agora? e logo em seguida as opções “Sim” e “Não” para a escolha do usuário. Agora se o bluetooth do aparelho Android do usuário já estiver habilitado e for pressionado na interface do aplicativo a função call ListPicker1 . Open será habilitado. Já a função call Bluetooth .Disconnect será ativado para o usuário optar por desativar o bluetooth assim o botão na interface do aplicativo mudará sua informação que estava indicando “Conectar” para “Desconectar”.

Na Figura 20, segue a tela de programação referente à ativação para conexão do módulo Bluetooth.

Figura 20 - Código: Tela de ativação do Bluetooth



Fonte: PRÓPRIA, 2017.

Este quinto bloco é responsável por receber a resposta informada pelo usuário no bloco anterior se a resposta tenha sido “Sim”, nisso a função call IniciaAtividade .StartActivity realizará a ativação do bluetooth no aparelho Android do usuário.

A Figura 21 refere-se à tela de programação para a listagem dos Bluetooth já pareados com o aparelho com o Sistema Operacional Android.

Figura 21 - Código: Tela de listagem dos Bluetooth



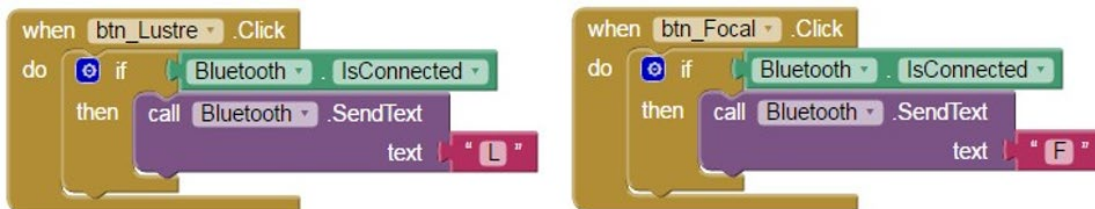
```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to Bluetooth .AddressesAndNames
```

Fonte: PRÓPRIA, 2017.

Sexto bloco: este bloco de funções é habilitado assim que o usuário pressiona o botão “Conectar” na interface do aplicativo, ele é responsável por listar todos os bluetooth já pareados com o aparelho Android em uso.

A Figura 22 refere-se à tela de programação que realizará a ativação dos LEDs presentes no projeto.

Figura 22 - Código: Tela de ativação dos LEDs



```
when btn_Lustre .Click
do if Bluetooth .IsConnected
then call Bluetooth .SendText text "L"

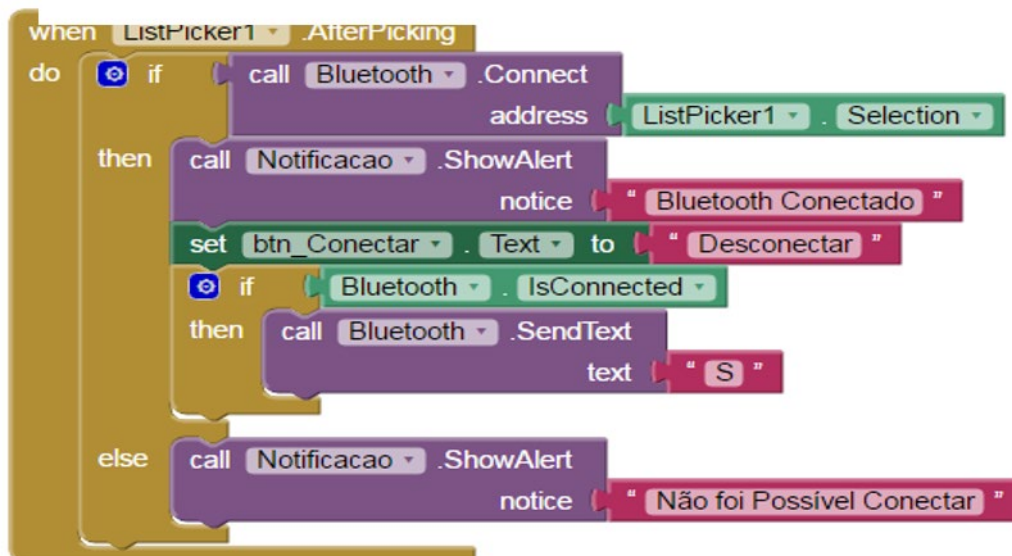
when btn_Focal .Click
do if Bluetooth .IsConnected
then call Bluetooth .SendText text "F"
```

Fonte: PRÓPRIA, 2017.

Na Figura acima encontra-se o sétimo bloco: quando a função When btn\_Lustre .Click for ativado pelo usuário e o bluetooth também estiver ativo, a função call Bluetooth .SendText text (“L”) terá a tarefa de enviar um caractere para o Arduino que por sua vez já terá as informações programas. Essas mesmas informações se aplicam para o bloco ao lado.

A Figura 23 refere-se à tela de programação para seleção do Bluetooth que realizará a conexão do aplicativo com a placa Arduino.

Figura 23 - Código: Tela de seleção do Bluetooth



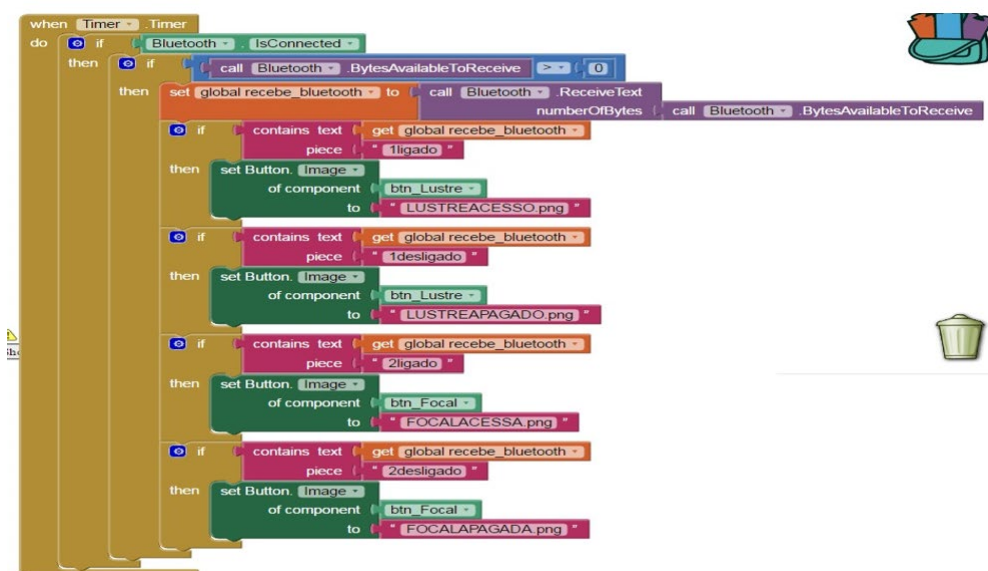
```
when ListPicker1 .AfterPicking
do if Bluetooth .Connect address ListPicker1 .Selection
then call Notificacao .ShowAlert notice "Bluetooth Conectado"
set btn_Conectar .Text to "Desconectar"
if Bluetooth .IsConnected
then call Bluetooth .SendText text "S"
else call Notificacao .ShowAlert notice "Não foi Possivel Conectar"
```

Fonte: PRÓPRIA, 2017.

A Figura 23 apresenta o oitavo bloco: depois de listado todos os bluetooths já pareados com o aparelho Android em uso, o usuário irá escolher com qual deles deseja se conectar e uma mensagem de “Bluetooth Conectado” será exibida, depois de realizada essa conexão a palavra “Conectar” que, antes era exibida no botão na interface do aplicativo mudará para “Desconectar”, mas se ao tentar se conectar a um determinado bluetooth e essa conexão não for bem sucedida uma mensagem será exibida “Não foi Possível Conectar”.

Na Figura 24, que segue tem-se o nono bloco: responsável pelo recebimento de todos os dados pelo bluetooth se esse já estiver habilitado no aplicativo. O bluetooth vai receber todos os dados vindo do Arduino desde que estes seja maiores que “0”, mas sem nenhum tipo de tratamento dos dados, para esse tratamento é usado a função contains text get global recebe\_bluetooth , onde o dado que chegar entra na função piece que está recebendo os dados “1ligado” que por sua vez irá efetuar a acionamento do LED nele configurado, caso essa função contains text get global recebe\_bluetooth, receba no piece “2desligado” ele irá desativar o LED nele configurado.

Figura 24 - Código: Tela de status

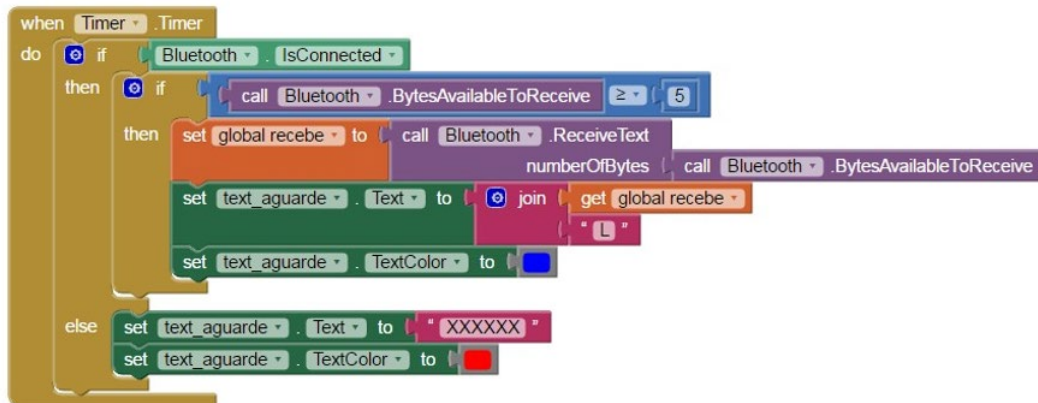


Fonte: PRÓPRIA, 2017.

### Código: Monitoramento do abastecimento residencial de água

Os 08 primeiros blocos referentes à programação do monitoramento do abastecimento de água contém a mesma programação que o monitoramento do status das lâmpadas. Sendo assim seguem as Figuras 25 e 26 e os códigos que diferem dos já apresentados, no que tange ao monitoramento do abastecimento residencial de água.

Figura 25 - Código: Tela de acompanhamento de status



Fonte: PRÓPRIA, 2017.

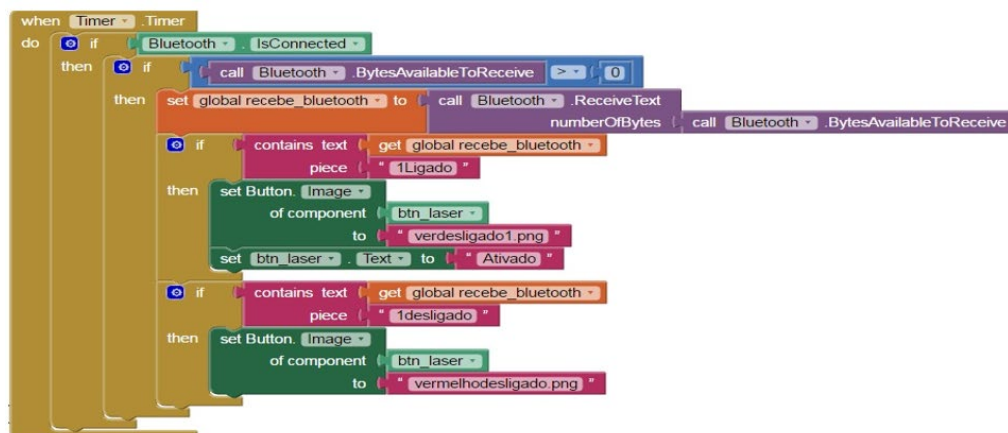
O nono bloco da Figura 25 é responsável pelo recebimento de todos os dados pelo bluetooth se esse já estiver habilitado no aplicativo. O bluetooth vai receber todos os dados vindo do Arduino desde que estes seja maiores que “5” e realizará a impressão dos dados obtidos do Sensor de Fluxo de Água na tela do aplicativo. Quando o bluetooth receber o valor definido para ativação da contagem de água presente no Sensor o texto presente na tela do aplicativo ficará azul, caso contrário ficará o texto ficará vermelho até a ativação.

### Código: Monitoramento da Cerca de segurança

Os 08 primeiros blocos referentes à programação do monitoramento da Cerca de segurança contém a mesma programação que o monitoramento do status das lâmpadas. Sendo assim seguem os códigos que diferem dos já apresentados, no que tange ao no que tange ao monitoramento da Cerca de segurança.

Assim a Figura 26 é responsável pelo recebimento de todos os dados pelo bluetooth se esse já estiver habilitado no aplicativo. O bluetooth vai receber todos os dados vindo do Arduino desde que estes seja maiores que “0”, mas sem nenhum tipo de tratamento dos dados, para esse tratamento é usado a função constains text get global recebe\_bluetooth, onde o dado que chegar entra na função piece que por sua vez está recebendo os dados “1ligado” que por sua vez irá efetuar a acionamento do laser e do LED nele configurado, caso essa função constains text get global recebe\_bluetooth, receba no piece “2desligado” ele irá desativar o diodo laser e o LED nele configurado.

Figura 26 - Código: Tela de acompanhamento de status



Fonte: PRÓPRIA, 2017.

## Parte física

Nesta altura do texto monográfico devemos levar em consideração três dos objetivos referentes ao âmbito físico do nosso projeto, sendo eles:

O desenvolvimento de um protótipo para o monitoramento do status das lâmpadas em uma residência que se beneficiará pelo uso de sensores de luminosidade, que indicarão a presença de lâmpadas acessas para fins de economia de energia elétrica;

O desenvolvimento de um protótipo para o monitoramento do abastecimento residencial de água que se beneficiará pelo uso do sensor de fluxo de água;

O desenvolvimento de um protótipo de cerca de segurança fundamentada em um circuito eletrônico de monitoramento residencial via infravermelho.

## O Circuito: Protótipo de monitoramento do status das lâmpadas

Para a montagem do circuito eletrônico que irá compor em conjunto com o Sketch programado na IDE do Arduino, o protótipo de monitoramento do status das lâmpadas do nosso projeto, fez-se necessário o uso dos seguintes componentes eletrônicos:

- 01 Placa Arduíno Uno
- 01 Protoboard de 170 furos
- 08 LEDs de 5 mm
- Jumpers
- 10 Resistores 330 Ω
- 01 Módulo bluetooth RS232 HC-05
- 08 Push buttons

Figura 27 - Código: Componentes eletrônicos



Fonte: PRÓPRIA, 2017.

A Protoboard é uma matriz de contatos, é utilizada para fazer diversas aplicações. Constitui-se de uma base plástica, com vários orifícios que se destinam à inserção de terminais de componentes eletrônicos.

O LED é um dispositivo semiconductor que possui uma junção diodo, que ao ser polarizado corretamente emite energia luminosa.

Os fios jumper são peças plásticas que envolvem um filamento metálico que conduz eletricidade.

O Resistor é um dispositivo que possui as funções de geração de calor, limitação da corrente elétrica e produção de queda de tensão. A unidade de medida utilizada pelo resistor é o Ohm ( $\Omega$ ).

O Módulo Bluetooth é utilizado para efetuar a comunicação entre o Arduino e outros dispositivos com bluetooth, via wireless.

Os Push buttons são microchaves de toque, botões de pressão que possuem uma mola em seu mecanismo.

O Sensor de fluxo de água é utilizado para medir a quantidade de água em litros que passa pelo mesmo.

O Módulo LDR é um resistor que sofre variação da resistência com a luz.

O Diodo Laser é um dispositivo semiconductor, formado por uma junção pn semelhante ao encontrado em um diodo de emissão de luz (LED).

O Buzzer é um dispositivo de áudio que é utilizado principalmente como beeps de teclado. Quanto ao processo de conexão dos componentes/montagem do circuito procedeu-se da seguinte forma:

LED - No processo de montagem do circuito foi acoplado a cada LED um resistor para limitação de corrente, impedindo a queima do mesmo. Cada polo do LED é ligado a um GND e a um VCC.



BLUETOOTH - Foi conectado um divisor de tensão no contato RX do bluetooth para controle de tensão, impedindo a queima do mesmo e ainda um jumper a entrada TX do Arduino.

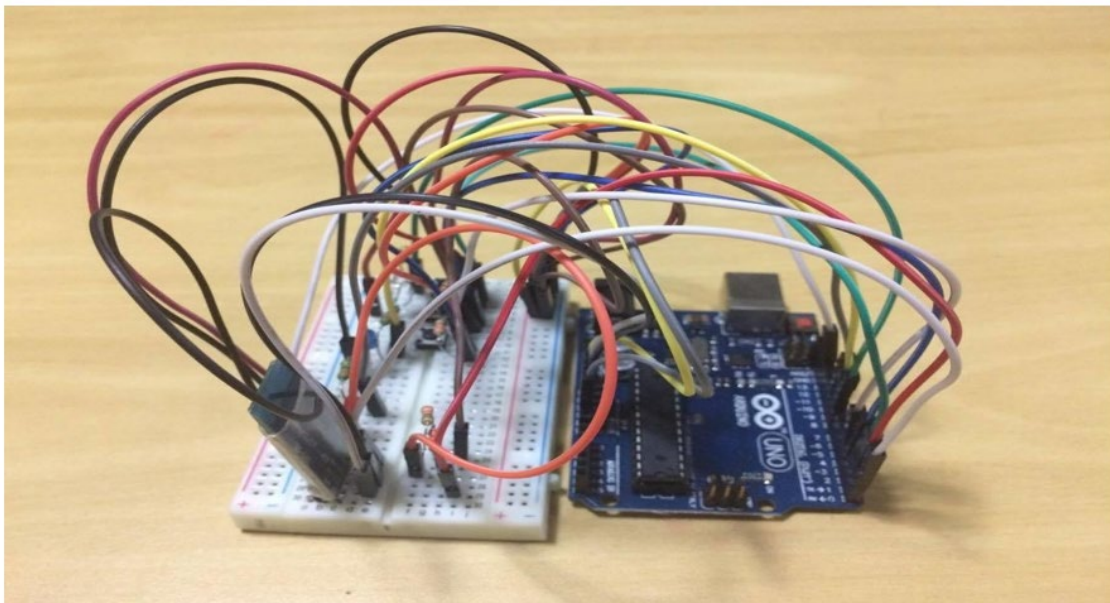
No contato VCC foi conectado um jumper que também foi conectado a entrada 5v do Arduino.

No contato TX foi conectado um jumper que também foi conectado a entrada RX do Arduino.

PUSH BUTTONS - Foi conectado um resistor para evitar a queima dos Push buttons que foram também interligados a um jumper ligado na saída GND. Há ainda outro jumper ligado a entrada VCC de 3.3v.

A Figura 28 contém o circuito de monitoramento do status das lâmpadas:

**Figura 28 - Circuito: Monitoramento do status das lâmpadas**



Fonte: PRÓPRIA, 2017

### **O Circuito: Protótipo de monitoramento do abastecimento residencial de água**

Para a montagem do circuito eletrônico que irá compor em conjunto com o Sketch programado na IDE do Arduino, o protótipo de monitoramento do abastecimento residencial de água do nosso projeto, fez-se necessário o uso dos seguintes componentes eletrônicos:

- 01 placa de Arduino Uno
- 01 Protoboard de 170 furos
- 08 LEDs de 5 mm
- Jumpers
- 04 Resistores de 330  $\Omega$
- 01 Módulo bluetooth RS232 HC-05
- 01 Sensor de fluxo de água 1/2" YF-S201 (conforme Figura 29)

**Figura 29 - Sensor de fluxo de água**



**Fonte: PRÓPRIA, 2017**

Quanto ao processo de conexão dos componentes/montagem do circuito procedeu-se da seguinte forma:

**LED** - No processo de montagem do circuito foi acoplado a cada LED um resistor para controle de tensão, impedindo a queima do mesmo. Cada polo do LED é ligado a um GND e a um VCC 5v. Um dos LEDs está conectado através de um jumper na entrada 6 do Arduino e o outro na entrada 5.

**BLUETOOTH** - Foi conectado um divisor de tensão no contato RX do bluetooth para controle de tensão, impedindo a queima do mesmo e também um jumper a entrada TX do Arduino. Além desses foi conectado, ainda, um jumper na entrada GND do Arduino.

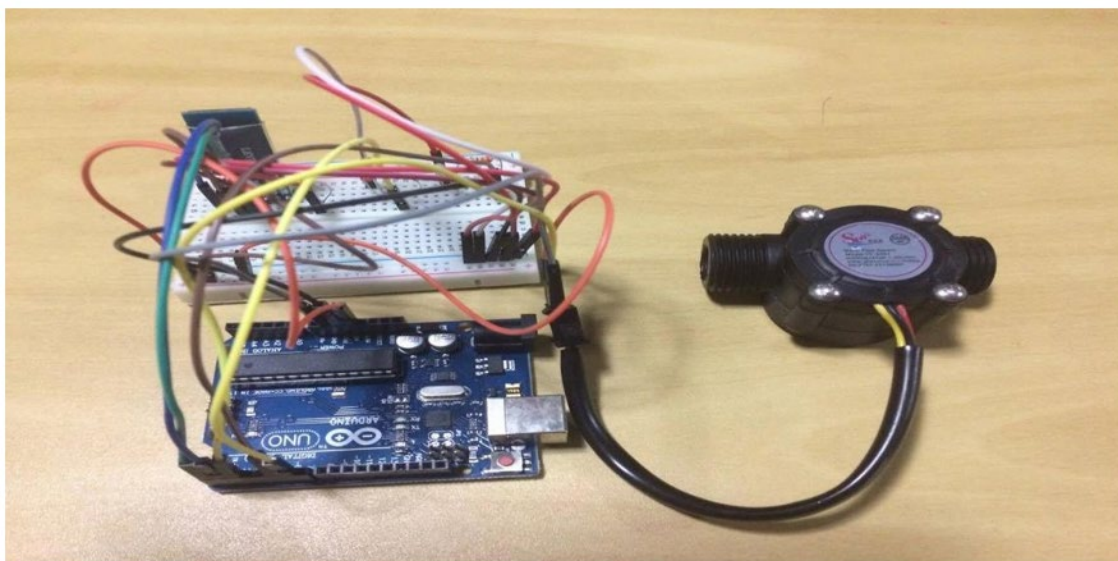
No contato VCC foi conectado um jumper que também foi conectado a entrada 5v do Arduino.

No contato TX foi conectado um jumper que também foi conectado a entrada RX do Arduino.

**SENSOR DE FLUXO DE ÁGUA** - O sensor de fluxo de água é dotado de três contatos, sendo o GND conectado a entrada GND do Arduino; o VCC conectado ao VCC de 5v do Arduino e saída PWN ligada ao pino 2 do Arduino.

A Figura 30 contém o circuito de monitoramento do abastecimento de água:

**Figura 30 - Circuito: Monitoramento do abastecimento de água**



Fonte: PRÓPRIA, 2017

### **O Circuito: Protótipo de cerca de segurança**

Para a montagem do circuito eletrônico que irá compor em conjunto com o Sketch programado na IDE do Arduino, o protótipo de monitoramento residencial via infravermelho do nosso projeto, fez-se necessário o uso dos seguintes componentes eletrônicos:

- 01 placa de Arduino Uno
- 01 Protoboard de 170 furos
- 08 LEDs de 5 mm
- Jumpers
- 08 Resistores de 330  $\Omega$
- 01 Módulo bluetooth RS232 HC-05
- 01 Push buttons
- 01 Módulo LDR (Conforme figura abaixo)
- 01 Diodo Laser 5v (Conforme figura abaixo)
- 01 Buzzer (Conforme figura abaixo)

Quanto ao processo de conexão dos componentes/montagem do circuito procedeu-se da seguinte forma:

**LED** - No processo de montagem do circuito foi acoplado a cada LED um resistor para limitação de corrente, impedindo a queima do mesmo. Cada polo do LED é ligado a um GND e a um VCC 5v. Um dos LEDs está conectado através de um jumper na entrada 12 do Arduino e o outro na entrada 11.

**BLUETOOTH** - Foi conectado um divisor de tensão no contato RX do bluetooth para con-

trole de tensão, impedindo a queima do mesmo e também um jumper a entrada TX do Arduino. Além desses foi conectado, ainda, um jumper na entrada GND do Arduino.

No contato VCC foi conectado um jumper que também foi conectado a entrada 5v do Arduino.

No contato TX foi conectado um jumper que também foi conectado a entrada RX do Arduino.

MÓDULO LDR - Foi conectado um resistor no contato VCC ao Módulo LDR e também jumper conectado a alimentação 5v. Foi conectado um jumper no contato GND do módulo LDR ligado ao GND do Arduino. Na entrada analógica do módulo LDR foi ligado um jumper conectado a entrada analógica A5 do Arduino.

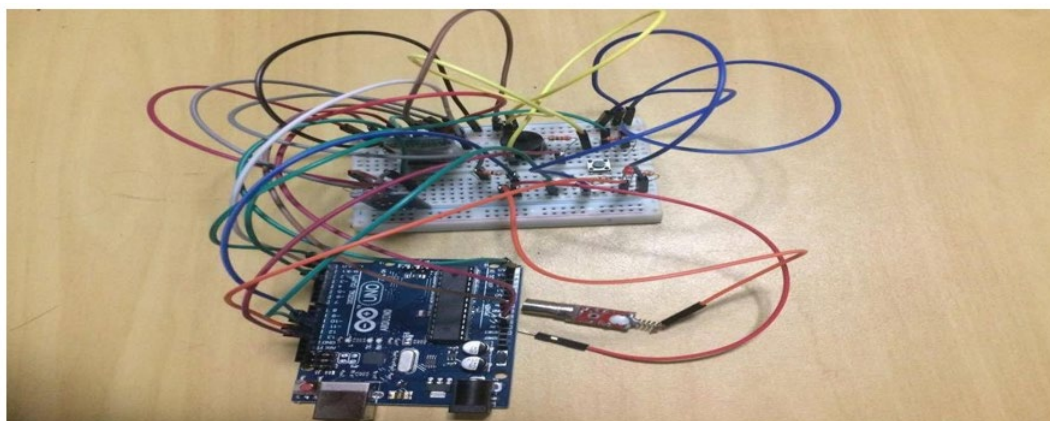
DIODO LASER - Foi ligado ao GND do Diodo Laser um resistor para controle de tensão, impedindo a queima do mesmo. No VCC do Diodo Laser foi ligado um jumper conectado a entrada 10 do Arduino.

BUZZER - No contato GND do Buzzer foi conectado um jumper ligado ao GND do Arduino. No contato positivo do Buzzer foi conectado um resistor para controle de tensão, impedindo a queima do mesmo. Foi conectado um jumper a entrada 13 do Arduino.

PUSH BUTTONS - Foi conectado um resistor para evitar a queima dos Push buttons que foram também interligados a um jumper ligado na saída GND. Outro jumper ligado a entrada 3 do Arduino, há ainda outro jumper conectado ao VCC de 5v.

A Figura 31 contém o circuito do protótipo da cerca de segurança:

**Figura 31 - Circuito: Protótipo da cerca de segurança**



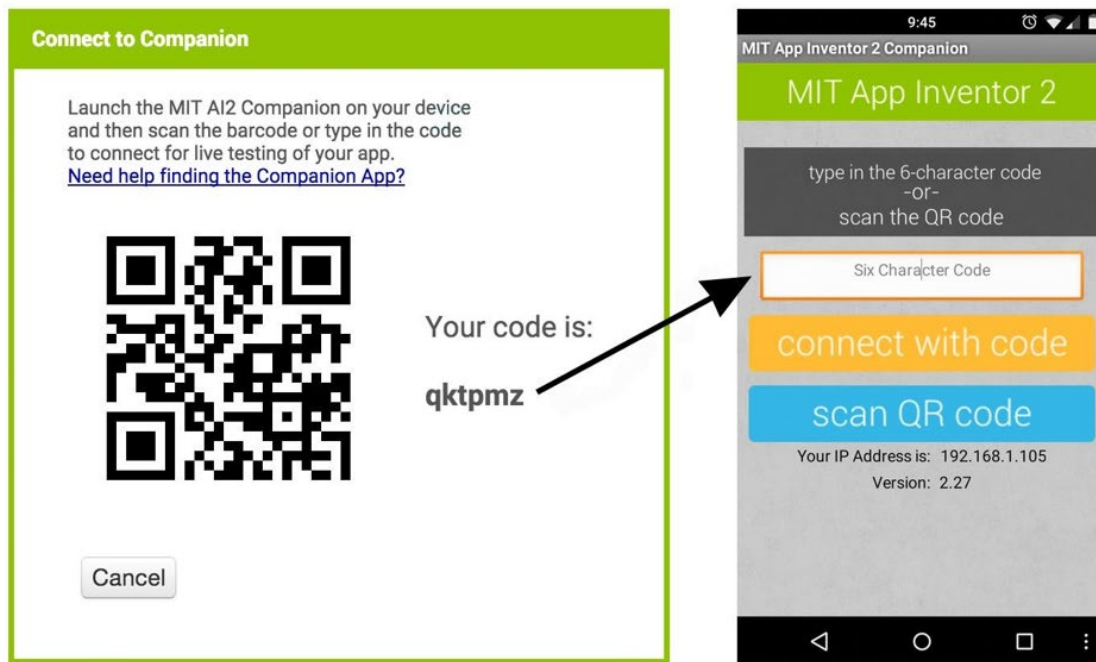
Fonte: PRÓPRIA, 2017

## RESULTADOS OBTIDOS

O capítulo de desenvolvimento mostrou como a solução de automação residencial Infraqualuz foi projetado e construído.

Ao término do desenvolvimento do aplicativo na plataforma APP INVENTOR, o arquivo do mesmo pode ser enviado pelo celular a ser instalado via download, ou através de um QR-CODE que viabiliza o acesso direto ao pacote do aplicativo para Android, conforme Figura 32:

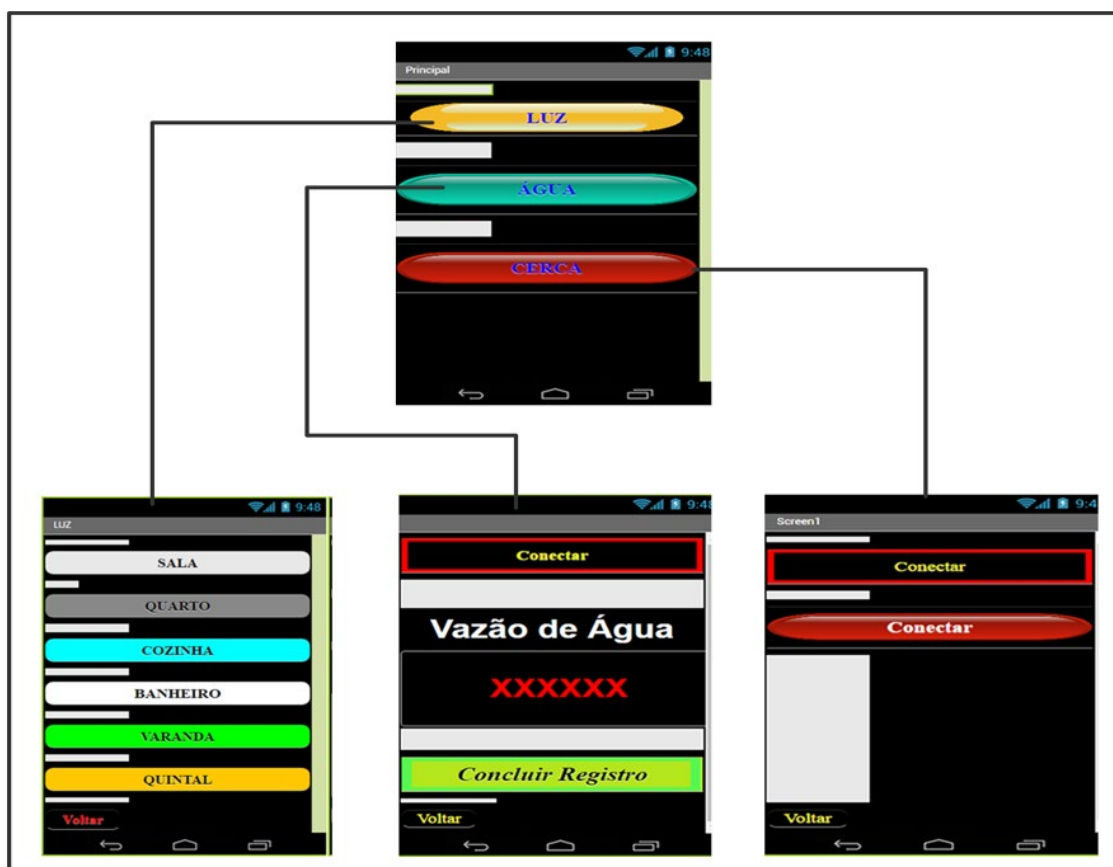
**Figura 32 - Tela de download do pacote do aplicativo Android na plataforma App Inventor.**



Fonte: PRÓPRIA, 2017

Assim, conforme proposto nos objetivos deste trabalho, esboço e projeto das funções, e por fim o desenvolvimento do aplicativo e da interligação dos componentes eletrônicos na placa Arduino, obteve-se a solução integrada, conforme a Figura 33:

Figura 33 - Aplicativo obtido após o processo de desenvolvimento



Fonte: PRÓPRIA, 2017

A tela LUZ permite o controle de liga/desliga das lâmpadas dos cômodos especificados: sala, quarto, cozinha, banheiro, varanda e quintal.

A tela ÁGUA mostra a vazão de água atual, permitindo concluir se há ou não água disponível, ou até mesmo se não houver consumo residencial no momento, pode indicar possível vazamento.

A tela CERCA mostra a ativação da cerca de segurança patrimonial que poderia ser colocado no topo dos muros, onde o feixe de infravermelho, ao ser interrompido por um possível invasor, dispara uma ação, que poderia ser um alarme ou uma ligação para um telefone pré-programado na placa do Arduino. Também pode-se utilizar para um melhor controle e acesso ao sistema utilizar um módulo Ethernet Shield W5100 onde se pode controlar todo o sistema via internet com o auxílio do aplicativo Infragualuz.

O aplicativo é o ponto central de envio e recebimento de informações, que unifica o acesso às placas Arduino e seus sensores para que a automação residencial aconteça de um único lugar.

## CONSIDERAÇÕES FINAIS

Buscou-se investigar, neste trabalho, por meio de pesquisa bibliográfica e experimental quais são as dificuldades e os benefícios do desenvolvimento de um aplicativo interligado a um sistema de monitoramento por meio da integração da eletrônica com sistemas de informação usando a plataforma Arduino.

Procedeu-se, dessa forma, ao apanhado do referencial teórico para embasar teoricamente os escritos e também orientar quanto ao desenvolvimento. Trouxe-se à tona os conceitos de Arduino, Mobile, Android, Linguagens de programação, App Inventor, Usabilidade e Automação residencial. Optou-se, então por dedicar um capítulo a explanação de tais conceitos.

No decorrer dessas discussões alguns pontos foram destacados: Inicialmente, que a plataforma Arduino possui um grande potencial para a criação de diversos projetos e que essa plataforma física de fonte aberta pode estabelecer comunicação com um computador para a realização de tarefas.

Constatou-se ainda, nas discussões teóricas que a plataforma Arduino é o mais versátil e mais popular, portanto, o mais adequado ao nosso projeto, é o Arduino Uno. Ainda com relação à programação da placa evidenciou-se que para fins de programação, o Arduino utiliza-se do IDE do Arduino, que se apresenta como um software livre onde se escreve o código baseado na linguagem C++.

Posteriormente verificou-se que o surgimento da tecnologia mobile possibilita o desenvolvimento de uma série de soluções que visam atender as demandas sociais. Foi, portanto, motivados pelo apelo deste mercado, que elaborou-se esse trabalho, com o intuito da criação de aplicativo para a plataforma mobile Android que atenda algumas demandas residenciais.

A escolha do Android como sistema operacional para o aplicativo Infragualuz, deu-se em função de sua versatilidade mercadológica; de sua arquitetura extremamente flexível, que possibilita a integração de aplicações bem como de seu sistema operacional, baseado no Linux.

Importante ressaltar que fez-se aqui o uso do Módulo Bluetooth, para a comunicação do aplicativo com o Arduino, em detrimento do uso da Ethernet Shield face aos custos elevados apresentados pelo último. E ainda porque o custo-benefício do uso do Bluetooth adequa-se de maneira satisfatória a este projeto.

Foi feito ainda, um adendo na discussão a fim de se pontuar algumas questões de ordem práticas acerca das Linguagens de programação. Pontuou-se que as Linguagens de programação compreendem os métodos de implementação: Compilação (que abarca a LP C++), a Interpretação e o método Híbrido (no interior do qual encontra-se a LP Java). Fez-se tais apontamentos com a finalidade de abordarmos a linguagem C++, por meio da qual programou-se as placas de Arduino Uno e ainda de situarmos a linguagem (Java) que é empregada pela ferramenta (App Inventor), que usamos como auxílio ao desenvolvimento do aplicativo, uma vez que o App Inventor é escrito na LP Java.

Assim, para comprovar a afirmação inicial (de que projetos desenvolvidos com o uso da plataforma Arduino podem estabelecer comunicação com um computador) projetou-se o desenvolvimento de um aplicativo mobile para Android (o Infragualuz) que tem o objetivo de pro-

porcionar maior segurança, economia energética e monitoramento do abastecimento de água aos usuários do mesmo, haja vista que a automação residencial é uma evidente aliada da vida moderna, que impõe uma rotina acelerada as pessoas, por possuir a finalidade de tornar mais prática a rotina cotidiana.

Acredita-se ter alcançado os objetivos específicos dessa pesquisa: O desenvolvimento de um protótipo para o monitoramento do status das lâmpadas em uma residência que se beneficiará pelo uso de sensores de luminosidade que indicarão a presença de lâmpadas acessas para fins de economia de energia elétrica; O desenvolvimento de um protótipo para o monitoramento do abastecimento residencial de água que se beneficiará pelo uso de sensores de umidade que indicarão a passagem do fluxo de água pelo referido protótipo e o desenvolvimento de um protótipo de cerca de segurança baseada em um circuito eletrônico de monitoramento residencial via infravermelho, uma vez que atingindo o objetivo geral: O desenvolvimento de um aplicativo Android+Arduino que contemple três funções básicas, são elas: segurança, monitoramento de abastecimento de água e auxílio no consumo de energia elétrica, utilizando a placa Arduino. Saliencia-se ainda que tais recursos propiciaram um universo de possibilidades as pessoas surdas beneficiando assim sua qualidade de vida.

Conclui-se este trabalho, portanto, na perspectiva de aperfeiçoar-se posteriormente o trabalho aqui iniciado, aprimorando-o e suscitando novas discussões no âmbito acadêmico.



## REFERÊNCIAS

- ABLESON, W. Frank. Android em ação. Tradução Eduardo Kraszczuk. Rio de Janeiro: Elsevier, 2012.
- ACCARDI, Adonis; DODONOV, Eugeni. Tecnologias, Infraestrutura e Software Automação Residencial: Elementos Básicos, Arquiteturas, Setores, Aplicações e Protocolos. Revistatis, São Carlos, v. 1, n. 2, p. 156-166, nov. 2012. Disponível em: <<http://revistatis.dc.ufscar.br/index.php/revista/article/view/27/30>>. Acesso em: 18 set. 2016.
- BANZI, Massimo; SHILOH, Michael. Primeiros passos com o Arduino. 2. ed. Trad: Aldir José Coelho Corrêa da Silva. São Paulo: Novatec, 2015.
- BOLZANI, C. A. M. Residências Inteligentes. São Paulo: Livraria da Física, 2004.
- BOLETIM INDUSTRIAL. Brasil: Mercado de automação residencial movimentou R\$ 4 bilhões e projeta aumento de 30%. Disponível em: <<http://boletimindustrial.com.br/brasil-mercado-de-automacao-residencialmovimentou-r-4-bilhoes-e-projeta-aumento-de-30/>>. Acesso em: 18 mai. 2016.
- COMPONENTES ELETRÔNICOS. Disponível em: <http://www.eletronicadidatica.com.br/protoboard.html>. Acesso em: 14 fev. 2017.
- GIL, Antônio Carlos. Como elaborar projetos de pesquisa. 4. ed. São Paulo: Atlas, 2002.
- GOMES, Tancicleide C. S.; GOMES, Jeane C. B. de Melo. Lógica de Programação: Iniciação Lúdica com App Inventor. Revista da Escola Regional de Informática, Recife, v. 2, nº 2, 2013.
- JUSTEN, Álvaro. Curso de Arduino. São Paulo, 2011. (Apostila).
- LECHETA, Ricardo R. Google Android - Aprenda a criar aplicações para dispositivos móveis com o Android Sdk. 2. Ed. São Paulo: Novatec, 2013.
- MACEDO, Renato Soares de; PRIETCH, Soraia Silva. Utilizando a Placa Arduino como Ferramenta de Apoio ao Ensino de Programação. IV Escola regional de informática (ERI- MT), Alto Araguaia: UNEMAT, 2013. p. 21 - 26. Disponível em: <[http://200.129.241.80/eri2013/arquivos/ERI\\_2013\\_Completos.pdf](http://200.129.241.80/eri2013/arquivos/ERI_2013_Completos.pdf)>. Acesso em: 16 nov. 2015.
- MCRBERTS, Michael. Arduino Básico. São Paulo: Novatec, 2011.
- MILETO, Thales da. Engenharia Aplicada & Descomplicada: Arduino - O Que é e Quais as Suas Possibilidades? Disponível em: <<http://www.thalesdamileto.files.wordpress.com/2015/08/arduino-mega.jpg>>. Acesso em 05 out. 2015.
- NIELSEN IBOPE; MOBILE MARKETING ASSOCIATION (MMA). Disponível em: <[http://www.nielsen.com/br/pt/search.html?q=ANDROID&sp\\_cs=UTF-8](http://www.nielsen.com/br/pt/search.html?q=ANDROID&sp_cs=UTF-8)>. Acesso em: 10 set. 2015.
- OLIBÁRIO, José Machado Neto. Usabilidade da interface de dispositivos móveis: heurísticas e diretrizes para o design. 2013.118 f. Dissertação (Mestrado) – Universidade de São Paulo.
- RABELLO, Ramon Ribeiro. Android: Um Novo paradigma de desenvolvimento móvel. Revista WebMobile Magazine, Edição 18, 2009.
- VAREJÃO, Flávio. Linguagens de Programação: Conceitos e Técnicas. Rio de Janeiro: Elsevier, 2004.
- WORTMEYER, Chales; FREITAS, Fernando; CARDOSO, Líuam. Automação residencial: Busca de tecnologias visando o conforto, a economia, a praticidade e a segurança do usuário. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA SEGeT 2005, 2, 2005. Disponível em: <[http://www.aedb.br/seget/arquivos/artigos05/256\\_SEGET%20-%20Automacao%20Residencial.pdf](http://www.aedb.br/seget/arquivos/artigos05/256_SEGET%20-%20Automacao%20Residencial.pdf)>. Acesso em 19 de jan. 2017.

## **SOBRE O AUTOR**

### **CARLOS RANIENY PEREIRA ROCHA**

Graduado em Sistemas de Informação pela Universidade Estadual de Montes Claros - UNIMONTES e pós graduado em LIBRAS pelas Faculdades Favenorte. Possui experiência em desenvolvimento de sistemas para acessibilidade a pessoa surda. Atua como professor no Departamento de Comunicação e Letras (Unimontes).

# ÍNDICE REMISSIVO

## A

abastecimento 5, 8, 9, 17, 20, 21, 22, 24, 27, 28, 30, 32, 37, 39, 41, 42, 43, 48  
acessibilidade 7, 50  
android 5  
Android 5, 8, 9, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 35, 36, 37, 45, 47, 48, 49  
aplicativos 7, 8, 13, 14, 15, 20  
arduino 5, 25  
Arduino 5, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 36,  
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49  
automação 5, 7, 8, 13, 19, 20, 45, 46, 48, 49

## B

baixo custo 5, 7, 9, 20

## C

capacidade 7  
comunicação 5, 7, 8, 9, 13, 19, 22, 23, 26, 28, 29, 40, 47  
consumo 5, 9, 17, 20, 24, 33, 46, 48  
contato 3, 41, 42, 43, 44

## D

dispositivos 5, 7, 12, 13, 14, 15, 19, 20, 40, 49

## E

economia 5, 8, 22, 26, 31, 34, 39, 48, 49  
eletrônica 8, 9, 47  
energia elétrica 5, 8, 9, 17, 20, 24, 26, 33, 39, 48

## H

hardware 7, 9, 10, 14, 15

## I

informação 5, 8, 13, 35, 47

## M

mercado 7, 13, 14, 15, 20, 47, 49  
mobile 7, 8, 13, 15, 22, 24, 30, 47  
monitoramento 5, 7, 8, 9, 17, 20, 21, 22, 24, 26, 27, 28, 29, 32, 33, 34, 37, 38, 39, 41,  
42, 43, 47, 48  
móveis 7, 13, 14, 19, 49

## N

necessidade 5, 7, 16, 19

## P

plataforma 5, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 20, 21, 45, 47  
processo 8, 15, 20, 22, 40, 42, 43, 46  
projetos 5, 7, 9, 20, 47, 49

## R

recursos tecnológicos 7

residenciais 5, 7, 8, 13, 47

residencial 5, 7, 8, 9, 13, 19, 20, 21, 22, 27, 29, 37, 39, 41, 43, 45, 46, 47, 48, 49

## S

segurança 5, 7, 8, 9, 13, 17, 20, 21, 22, 24, 29, 30, 33, 38, 39, 43, 44, 46, 48, 49

segurança patrimonial 5, 7, 46

sensores 5, 7, 8, 10, 17, 20, 26, 27, 39, 46, 48

sistemas automáticos 7

sociais 7, 13, 47

software 7, 9, 10, 11, 12, 13, 14, 15, 18, 47

surda 5, 50

surdo 7

## T

tecnologias 5, 7, 9, 19, 20, 49

## U

USB 4, 10, 11, 25

