



**UNIVERSIDADE FEDERAL DE OURO PRETO**  
**ESCOLA DE MINAS**  
**DEPARTAMENTO DE ENGENHARIA DE**  
**CONTROLE E AUTOMAÇÃO E TÉCNICAS FUNDAMENTAIS**



**HELDER JOSÉ SANTOS LUZ**

**SISTEMA DE AUTOMAÇÃO RESIDENCIAL/ PREDIAL UTILIZANDO ARDUINO E**  
**SISTEMA OPERACIONAL ANDROID**

**OURO PRETO – MG**

**2018**

**HELDER JOSÉ SANTOS LUZ**

**SISTEMA DE AUTOMAÇÃO RESIDENCIAL/ PREDIAL UTILIZANDO ARDUINO E  
SISTEMA OPERACIONAL ANDROID**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção de Grau em Engenheiro de Controle e Automação.

Orientador: Professor Dr. Paulo Marcos de Barros Monteiro

**OURO PRETO – MG**

**2018**

L979s

Luz, Helder José Santos.

Sistema de Automação Residencial/Predial utilizando Arduino e sistema operacional Android [manuscrito] / Helder José Santos Luz. - 2018.

48f.: il.: color.

Orientador: Prof. Dr. Paulo Marcos de Barros Monteiro.

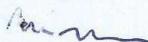
Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

1. Automação residencial. 2. Redes locais sem fio - Bluetooth. 3. Software. 4. Microcontrolador. I. Monteiro, Paulo Marcos de Barros. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 681.5

Catálogo: [ficha@sisbin.ufop.br](mailto:ficha@sisbin.ufop.br)

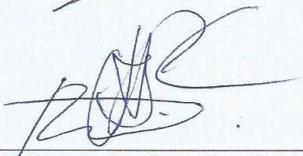
Monografia defendida e aprovada, em 07 de fevereiro de 2018, pela comissão avaliadora constituída pelos professores:



\_\_\_\_\_  
Prof. Dr. Paulo Marcos de Barros Monteiro - Orientador



\_\_\_\_\_  
Prof. Dr. Luiz Fernando Rispoli Alves – Professor Convidado



\_\_\_\_\_  
Prof. Dr. Agnaldo José da Rocha Reis – Professor Convidado

## AGRADECIMENTOS

Aos meus pais, José Bernardo e Maria Venância, por sempre acreditarem no meu potencial, por todo apoio motivacional. Minha família que torceu por mim em todos os momentos.

Agradeço a Escola de Minas e seus mestres, em especial ao professor Paulo Monteiro, que me apresentou o curso e sua estrutura, fator determinante para minha decisão de seguir a área de automação.

Aos amigos de graduação, por todos os conhecimentos compartilhados ao longo desses anos de estudos

Ao amigo João Paulo, responsável pela Ocça e Minas Energia, por toda ajuda e aprendizado na área de projetos e automação residencial.

*“A mudança é a única coisa permanente no mundo.”*

Steve Jobs

## RESUMO

Nos dias atuais, tem se buscado opções que ofereçam uma maior comodidade e conforto para o ser humano no dia a dia. Por isso os sistemas de automação residencial surgiram para melhorar a comodidade do usuário, atendendo a todas as suas necessidades. A ideia inicial ao se implantar um sistema de automação, é fazer a utilização de vários equipamentos eletrônicos comandados por um único *software*. Um ponto fundamental que deve ser considerado é a relação custo/benefício. Por este motivo propõe-se a ideia de um sistema acessível e de baixo custo, envolvendo o sistema operacional Android, juntamente com a plataforma de testes Arduino e *Bluetooth*.

**Palavras chave:** Automação residencial, *Bluetooth*, *software*, Arduino.

## **ABSTRACT**

Nowadays, we have been looking for options that offer a greater comfort for the human being in the day to day. That is why residential automation systems have come up to improve user comfort, catering to all your needs. The initial idea when implementing an automation system is to use of several electronic devices controlled by a single *software*. A key point to consider is the cost / benefit ratio. For this reason the idea of an affordable and inexpensive system, involving the Android operating system, together with the Arduino and *Bluetooth*.

**Keywords:** Home automation, *Bluetooth*, *software*, Arduino.

## LISTA DE FIGURAS

Figura 1: Ilustração de uma placa Arduino tipo Mega 2560.....	11
Figura 2: Ilustração redes Piconet formando uma Scatternet.....	13
Figura 3: Layout de desenvolvimento do App Inventor.....	15
Figura 4: Codificação de um aplicativo desenvolvido no App Inventor.....	15
Figura 5: Ambiente de desenvolvimento Arduino.....	16
Figura 6: Arquitetura básica do sistema.....	17
Figura 7: Tela inicial do aplicativo.....	20
Figura 8: Tela Menu do Aplicativo.....	21
Figura 9: Tela controle de iluminação (sala).....	21
Figura 10: Tela controle de iluminação (quarto).....	22
Figura 11: Tela controle de iluminação (cozinha).....	22
Figura 12: Tela de controle geral do sistema.....	23
Figura 13: Detalhe da montagem da central de comando.....	24
Figura 14: Vista frontal da central de comando e aplicativo instalado no smartphone. .....	24
Figura 15: Mesa de testes do sistema.....	25
Figura 16: Módulo Bluetooth HC-06.....	26
Figura 17: Módulo relé de 16 canais.....	27

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>11</b>
1.1. Considerações Iniciais.....	11
1.2. Objetivos .....	9
1.2.1. Gerais .....	9
1.2.2. Específicos .....	9
1.3. Justificativa.....	9
1.4. Tecnologias Utilizadas .....	10
<b>2. REVISÃO BIBLIOGRÁFICA</b> .....	<b>11</b>
2.1. Arduino .....	11
2.2. <i>Bluetooth</i> .....	11
2.3. Android .....	13
2.4. Módulo relé .....	14
2.5. <i>Softwares</i> .....	14
2.5.1. <i>App Inventor</i> .....	14
2.5.2. Arduino – IDE .....	15
<b>3. ARQUITETURA E FUNCIONAMENTO DO SISTEMA</b> .....	<b>17</b>
3.1. Componentes do sistema.....	18
3.1.1. Aplicativo.....	18
3.1.2. Central de comando .....	18
3.2. Funcionamento do sistema.....	18
3.2.1. Modo automático .....	18
3.2.2. Modo automático temporizado .....	19
3.2.3. Modo manual .....	20
3.3. Montagem do circuito .....	23
<b>4. TESTES</b> .....	<b>25</b>
4.1. Ambiente de teste.....	25
4.2. Testes realizados .....	27
4.2.1. Controle de cenário individual.....	27
4.2.2. Controle coletivo de cenários .....	28
4.2.3. Habilitando/desabilitando o sistema .....	28
<b>5. CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>29</b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>30</b>
<b>7. APÊNDICES</b> .....	<b>31</b>

7.1. APÊNCICE A – Código do programa do Arduino .....	31
---	----

# **1. INTRODUÇÃO**

## **1.1. Considerações Iniciais**

O conceito de automação residencial refere-se ao uso da tecnologia que tem a finalidade de tornar cômodas as tarefas habituais em uma residência. A automação residencial é capaz de proporcionar o conforto a um indivíduo, podendo ser inserido em qualquer tipo de utilidade doméstica. Com isso, a automação residencial tem influência direta na ergonomia.

Nos dias atuais está cada vez maior a procura por métodos que facilitem a forma como as pessoas executam suas atividades, assim como a sensação de segurança, conforto e bem-estar para as mesmas. Este conjunto de tecnologias caracteriza a domótica, pois este sistema tem como objetivo, comodidade e segurança, além de um menor e mais racional consumo de energia [1].

Cidades Inteligentes são definidas de várias formas na literatura. Vários aspectos podem ser considerados ao se incluir uma cidade nessa categoria. Entretanto, de acordo com o escopo deste trabalho que engloba comunicação de dados e desenvolvimento de sistemas para um grande número de usuários, será considerada a definição de cidades inteligentes como um ambiente em que as tecnologias da informação e da comunicação de dados são utilizadas para melhorar a qualidade de vida da população. Tudo isso criado de forma que os recursos tecnológicos sejam embutidos nos objetos do cotidiano para que sejam transparentes ao usuário [2].

As residências ou edifícios atuais possuem diversas redes que se destinam às inúmeras aplicações, onde uma rede separada e independente é utilizada para cada funcionalidade.

Deste modo é possível então observar a existência de redes específicas à utilização de controle de acesso, à detecção e controle de incêndios, à climatização entre outras. Estes fatores tornam a rede doméstica uma boa solução para diversos problemas encontrados comumente, podendo ser utilizado para qualquer que seja a finalidade, bastando um planejamento adequado.

Esta rede, sendo vista de maneira ampla permite, o desenvolvimento de sistemas complexos, ao que se refere às tarefas que poderão ser executadas, mas aos olhos

do habitante ou usuário são tarefas chatas e rotineiras que muitas pela utilização dos recursos das redes domésticas nem é notável pela maioria.

## **1.2. Objetivos**

### **1.2.1. Gerais**

Realizar um estudo sobre sistemas de automação residencial, considerando a particularidade de cada sistema existente, bem como as suas semelhanças e diferenças, vantagens e desvantagens de cada um. A partir desse estudo será desenvolvido um sistema que possa agregar conforto ao usuário e que ao mesmo tempo não tenha um alto grau de complexidade para ser operado.

### **1.2.2. Específicos**

Com o objetivo de adquirir conhecimentos diante de novas tecnologias, este trabalho propõe a criação de um sistema de automação residencial de baixo custo, que possa ser controlado por meio do uso de um *Smartphone*. O sistema utiliza como central o Arduino e a comunicação via *Bluetooth* para transmitir as informações com o smartphone. De posse de seu *Smartphone*, o usuário acessa o sistema, estabelece uma comunicação com o Arduino e realiza o controle dos dispositivos inclusos na malha de controle do sistema. A ideia inicial é realizar a divisão em vários sistemas independentes, devido ao alcance proporcionado pelo módulo *Bluetooth*. Tal sistema foi desenvolvido para o sistema operacional Android. Dentre as tecnologias utilizadas, recebem destaque a transmissão de dados via *Bluetooth* e o sistema operacional para dispositivos móveis Android. Este projeto se enquadra no conceito de automação residencial, domótica e cidades inteligentes.

## **1.3. Justificativa**

O uso de sistemas de automação residencial teve um crescimento de forma considerável, pois o usuário deseja ter em sua residência um sistema que ofereça a maior comodidade ao mesmo. Porém, alguns desses sistemas devido ao seu grau de desenvolvimento, são considerados caros para serem implantados em alguns tipos

de residências e prédios. Devido esse princípio, foi proposta a ideia de desenvolver um sistema que seja eficaz e que tenha ao mesmo tempo um custo acessível ao público além de ser de fácil manutenção.

#### **1.4. Tecnologias Utilizadas**

O critério para a escolha das tecnologias utilizadas para o desenvolvimento do sistema foi de acordo com a relevância em relação ao mercado, ou seja, buscou-se tecnologias bastante utilizadas e em ascensão. Verificou-se que o sistema operacional *Android* é o mais utilizado por *Smartphones* e *tablets*, de acordo com os dados divulgados de uma pesquisa feita pela *StatCounter* [5]. Além disso, tais dispositivos geralmente possuem comunicação via *Bluetooth* e a literatura apresenta diversas bibliotecas que facilitam a utilização desse tipo de comunicação de dados

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1. Arduino

O Arduino é uma plataforma de computação *open-source* baseado em uma simples placa com portas de entradas e saídas tanto digitais como analógicas. Possui um próprio ambiente de desenvolvimento baseado na Linguagem C++. O Arduino pode ser usado para desenvolver objetos interativos autônomos ou pode ser conectado a um software em seu computador (ex. *Flash*, *Processing*, *MaxMSP*). O Ambiente de desenvolvimento (IDE) *open-source* pode ser obtido gratuitamente (atualmente disponível para Mac OS X, Windows, e Linux) [3].

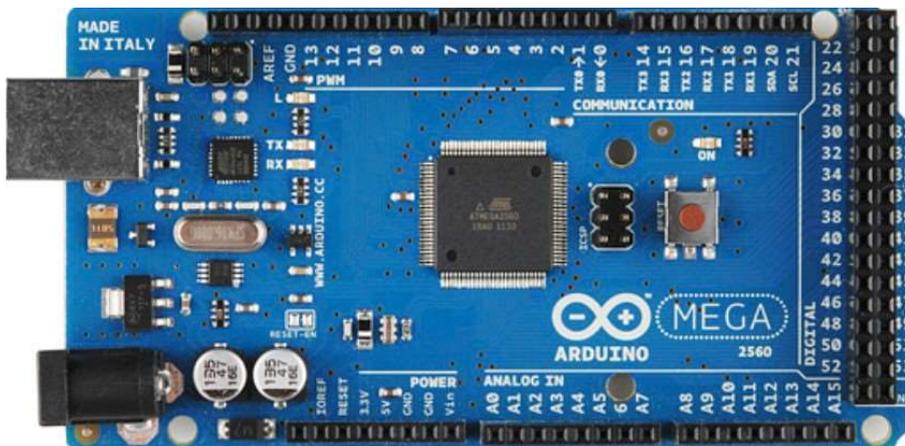


Figura 1: Ilustração de uma placa Arduino tipo Mega 2560.

Fonte: ELECTROSCHEMATICS.

### 2.2. Bluetooth

A tecnologia *Bluetooth* tem a função de realizar comunicação sem fio entre dois ou mais dispositivos em pequenas distâncias. Em uma rede *Bluetooth*, vários dispositivos podem ser conectados ao mesmo tempo para enviar e receber dados. Antes de iniciar uma transmissão de dados, o dispositivo *Bluetooth* deve realizar uma busca por novos dispositivos compatíveis que estejam dentro de seu raio de comunicação e só então criar uma conexão para troca de dados. Essa operação é chamada de pareamento de dispositivos. Ao realizar o pareamento, dependendo do fabricante do dispositivo e do

*software* utilizado, um dos dispositivos poderá solicitar uma senha para autorizar a criação da conexão [4].

Quando dois ou mais dispositivos utilizam o *Bluetooth* para realizar uma comunicação, eles formam uma rede denominada *Piconet*. Em uma rede *Piconet*, o dispositivo que inicia a comunicação é chamado de mestre e os dispositivos restantes são denominados escravos. O mestre tem a função de controlar como será feita a transmissão de dados e o sincronismo com os dispositivos escravos.

Cada rede *Piconet* é capaz de suportar até oito dispositivos conectados ao mesmo tempo (sendo um mestre e sete escravos). Usando um sistema de sobreposição de *Piconets*, é possível fazer com que o número de dispositivos conectados na mesma rede *Bluetooth* seja maior. Esse sistema de sobreposição é chamado de *Scatternet* e sua função é fazer com que uma *Piconet* se comunique com outras, dentro de um limite de alcance preestabelecido. A *Scatternet* permite também que um dispositivo escravo faça parte de mais de uma *Piconet*, coisa que não é permitida aos dispositivos mestres. Para que um dispositivo saiba quais e quantos dispositivos fazem parte da *Piconet*, um esquema de identificação é usado, sendo geralmente uma senha, que será pedida ao dispositivo quando o mesmo desejar se conectar à rede. A figura 1 mostra um esquema de funcionamento de redes *Piconets* formando uma rede *Scatternet*:

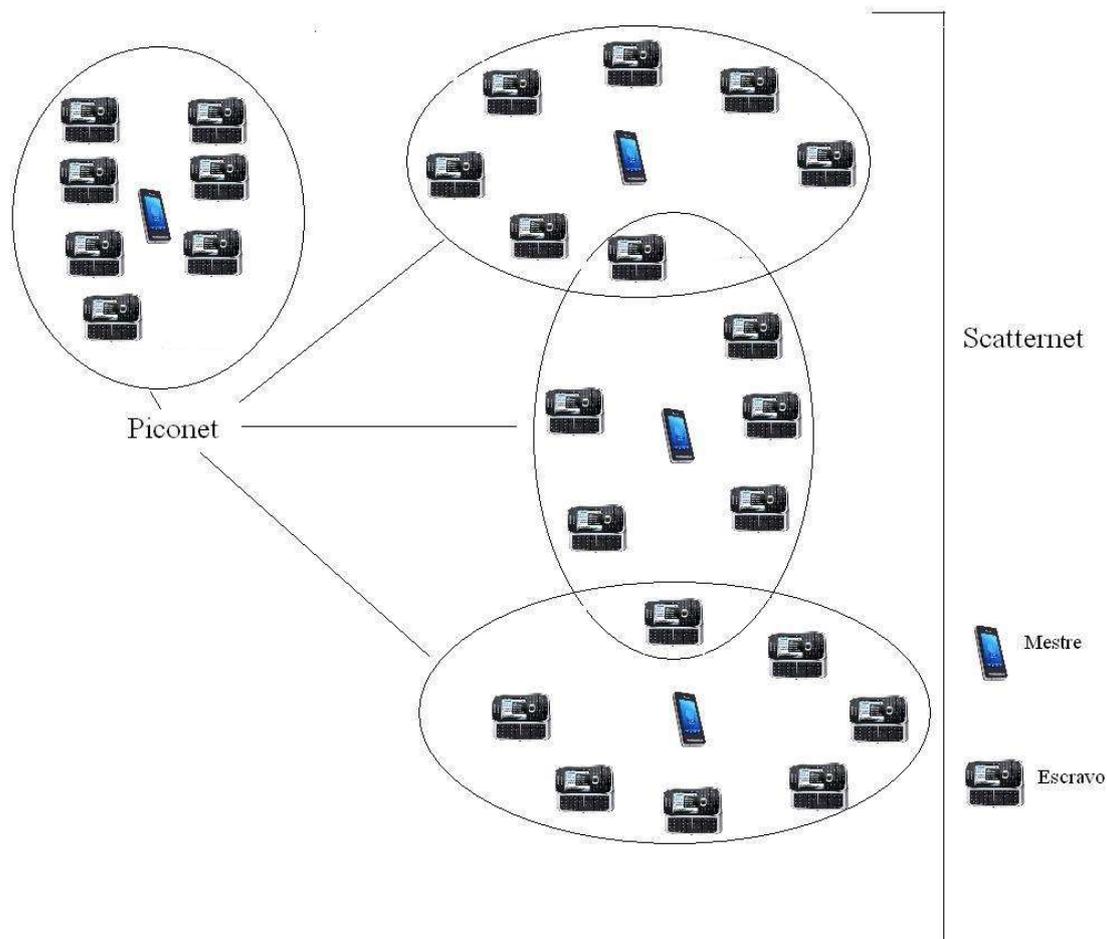


Figura 2: Ilustração redes Piconet formando uma Scatternet.

### 2.3. Android

O sistema operacional *Android* foi desenvolvido pela *Google* e é um *software* de uso livre, atualmente mantido pela *Open Handset Alliance (OHA)* [14]. As aplicações executadas pelo *Android* são desenvolvidas na linguagem de programação *Java*. Foi desenvolvido um pacote de instalação do *Android* chamado *Android SDK*, com o objetivo de facilitar a programação. Tal pacote adiciona ao *Eclipse* [14] (uma das mais conhecidas *IDEs* para *Java*) funcionalidades para desenvolvimento para *Android*. O *Android SDK* fornece um conjunto de bibliotecas, como o *SQLite*, que facilitam a utilização dos recursos encontrados nos dispositivos.

O *Android* funciona somente em *smartphones*, *tablets* ou emuladores. Ele possui várias versões, sendo mais utilizadas atualmente as versões 5.0 e 5.1. Uma

característica importante dos dispositivos que executam *Android* é o uso de telas sensíveis ao toque. Tal característica tem sido um dos fatores diferenciais na utilização do *Android*, pois facilita e agiliza a navegação do usuário e torna mais rápido o acesso aos aplicativos existentes no aparelho.

## **2.4. Módulo relé**

O relé é um interruptor eletromecânico aplicado em comandos elétricos, tendo a função de ligar ou desligar dispositivos. Quando uma corrente percorre o circuito, ela passa pela bobina, responsável pelo acionamento, onde um campo magnético é gerado. O relé pode ser disposto em módulos únicos ou em conjuntos montados em placas, como o da figura 17.

## **2.5. Softwares**

### **2.5.1. App Inventor**

O *App Inventor* é um aplicativo de código aberto criado pela *Google*, sendo que atualmente é gerido pelo *Massachusetts Institute of Technology (MIT)*. Ele usa uma interface gráfica que permite aos usuários arrastar e soltar objetos visuais para criar um aplicativo que pode ser executado em dispositivos *Android*. O *App Inventor* é uma alternativa aos programas *Eclipse* e *Android SDK* para criar aplicativos para o sistema *Android*. O sistema possui os modos de design e blocos, permitindo que o usuário personalize o *design* do seu aplicativo. Para ser instalado, o mesmo pode ser baixado por meio de um arquivo .apk ou através de um QR code.

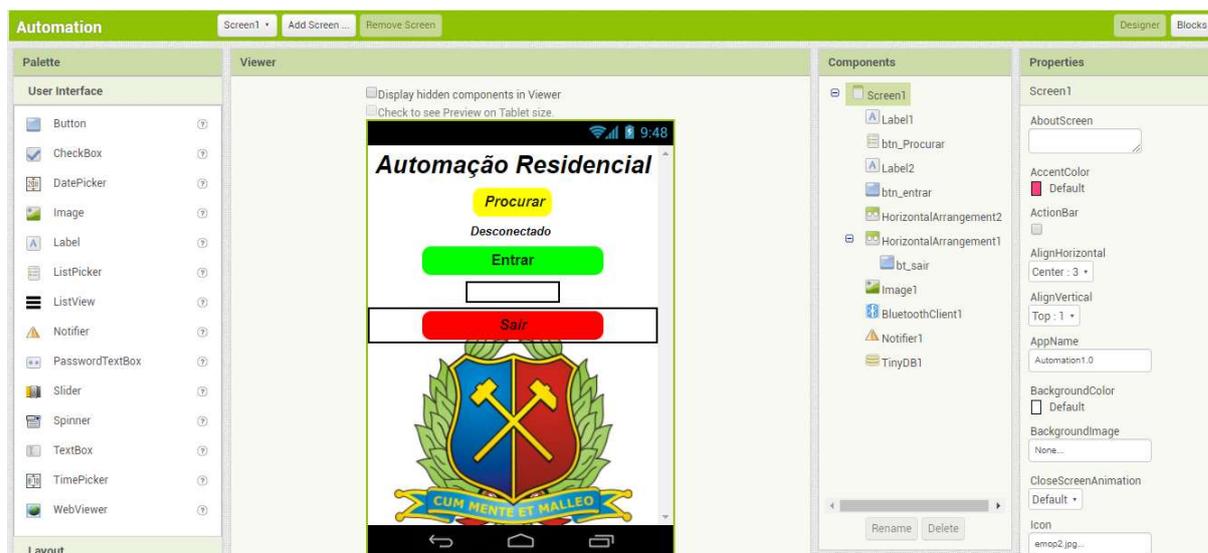


Figura 3: Layout de desenvolvimento do App Inventor.

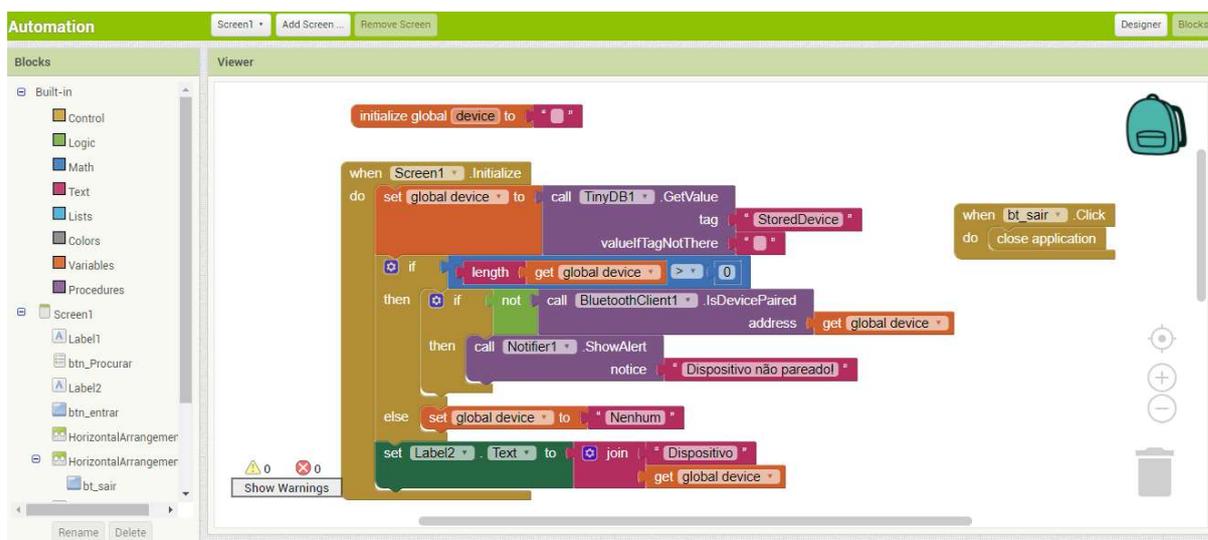


Figura 4: Codificação de um aplicativo desenvolvido no App Inventor.

## 2.5.2. Arduino – IDE

É um ambiente de desenvolvimento que torna possível a programação da placa Arduino e comunicação entre a placa e o programa, geralmente desenvolvido na linguagem de programação C++, mas também pode ser desenvolvido em Java e outros softwares de código livre. Sua versatilidade permite sua utilização em diferentes modelos de placas Arduino



Figura 5: Ambiente de desenvolvimento Arduino.

### 3. ARQUITETURA E FUNCIONAMENTO DO SISTEMA

Essa seção descreve a arquitetura e o funcionamento do sistema desenvolvido neste trabalho. O sistema funciona em dois modos, manual e automático. No modo manual, a central é desabilitada e o controle da iluminação é feita da forma convencional, por meio de interruptores. No modo automático, a central é ativada e o controle da iluminação é feito por meio dos comandos enviados pelo aplicativo instalado no *smartphone*. O módulo *Bluetooth* recebe o comando codificado e o envia para o Arduino que será responsável pela execução do comando.

A figura 6 ilustra a arquitetura do sistema e a comunicação entre os módulos. O envio do comando ocorre por meio de uma conexão entre o aplicativo do cliente e o módulo *Bluetooth* presente na central, com o intuito de enviar ao Arduino a ordem a ser executada pelo controlador.

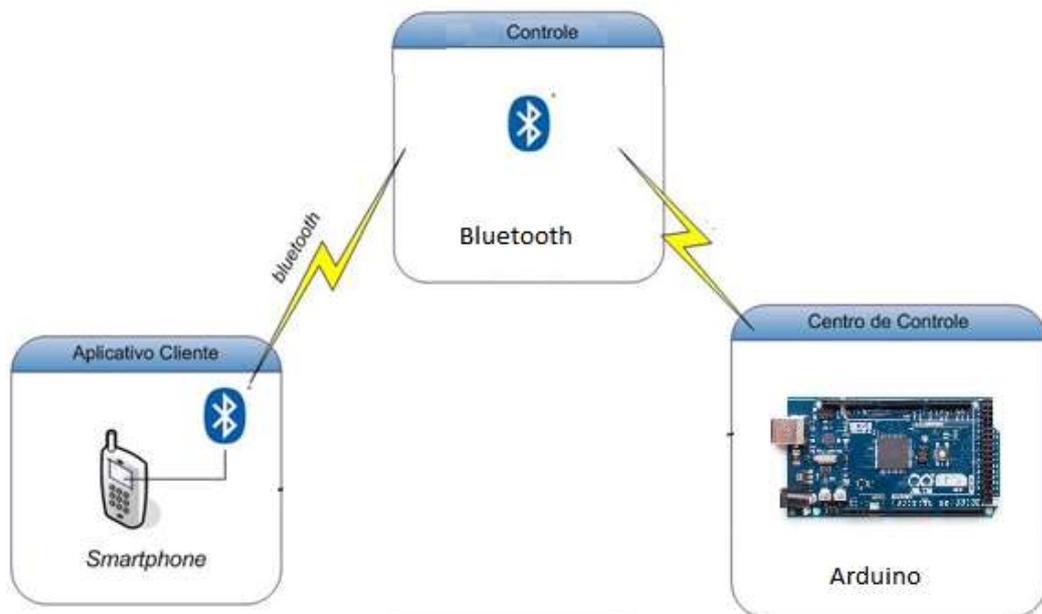


Figura 6: Arquitetura básica do sistema.

### **3.1. Componentes do sistema**

#### **3.1.1. Aplicativo**

O aplicativo foi desenvolvido na plataforma de desenvolvimento *App Inventor*, de modo que o mesmo fosse de entendimento mais simples possível. A aplicativo desenvolvido possui 6 telas diferentes, englobando tela inicial, tela de menus e telas de cenários, onde é possível realizar o controle do sistema de modo individual ou coletivo. A versão desenvolvida do aplicativo é voltada para *smartphones* dotados de sistema operacional Android.

#### **3.1.2. Central de comando**

A central de comando tem a função de armazenar todos os periféricos utilizados para controlar o sistema. A sua composição é formada pelos seguintes itens:

- Arduino Mega R3
- Módulo relé 16 canais
- Fonte DC 12V
- Fonte DC 5V
- Display LCD 16x2 5V
- Chave liga desliga 3 estágios

### **3.2. Funcionamento do sistema**

#### **3.3.**

##### **3.3.1. Modo automático**

O sistema funciona da seguinte forma: o usuário terá em seu *smartphone* (ou em outro dispositivo móvel) um aplicativo que se conectará via *Bluetooth* com a central de comando. Tal aplicativo faz uma busca para realizar a conexão com a central. A central, ao receber a solicitação por meio do *Bluetooth* enviará ao Arduino a mensagem codificada indicando qual operação que deverá ser realizada. Como foi dito anteriormente, o sistema poderá ser habilitado ou desabilitado por meio de uma chave localizada na central de comando. O aplicativo não possui bloqueio por senha.

Ao acessar o aplicativo, o cliente terá na tela inicial a opção de procura do dispositivo *Bluetooth* para realizar o pareamento. Caso não esteja pareado, o sistema irá mostrar uma mensagem indicando o não pareamento entre o *smartphone* e a central.

Para garantir que a conexão funcione de uma forma estável, o aplicativo se conecta ao módulo *Bluetooth* somente quando o usuário acessa uma tela de uma área específica que ele deseja controlar. Caso resolva trocar de tela, a conexão é desfeita automaticamente quando o usuário retornar à tela menu.

Em cada tela o usuário terá acesso para controlar uma área específica. No caso deste aplicativo, o usuário poderá controlar a iluminação de três cômodos de forma individual (sala, cozinha e quarto) e uma tela para acionamento coletivo (área geral). Vale ressaltar que a opção exercida em cada tela se mantém fixada mesmo quando o usuário não esteja navegando pela mesma.

O acionamento da iluminação é feito por relés individuais que recebem um sinal proveniente do Arduino. A placa é alimentada por uma tensão VDC de 5V independente do Arduino, sendo que cada relé pode suportar uma corrente máxima de 30 A na tensão alternada de 127/220V. Cada relé recebe um sinal emitido pelas portas de saída do Arduino (nível alto e baixo).

O Arduino é alimentado por uma tensão VDC que pode variar entre 7 e 12V com limite máximo de 20V e é responsável por fornecer a alimentação para o módulo *Bluetooth* HC-06 (3,3V) e o display LCD 16x2 (5V).

A alimentação do Arduino e da placa de relés é feita por duas fontes instaladas dentro da central de comando, sendo que funcionarão somente quando o modo automático estiver ativado.

### **3.3.2. Modo automático temporizado**

O funcionamento do sistema no modo automático temporizado é quase análogo ao modo automático comum, sendo que o usuário terá a opção de acionar a iluminação por um tempo pré-estabelecido. Assim que o tempo se esgotar, a iluminação selecionada se apagará automaticamente.

### 3.3.3. Modo manual

No modo manual a central é desabilitada por meio de acionamento de uma chave presente na parte externa da central, fazendo com que o controle de iluminação seja feito pelos interruptores já existentes na instalação elétrica residencial.

A inversão do modo manual para o modo automático é feita por uma chave de três estágios. No modo manual a fase alimenta o circuito da forma convencional. Quando a chave aciona o modo automático, a fase alimentará a saída dos relés responsáveis por acender cada lâmpada pré-estabelecida. O usuário tem a opção de desligar o sistema totalmente, impedindo seu funcionamento tanto no modo manual quanto no modo automático. Vale lembrar que a interrupção do cabo é feita somente no condutor fase, pois o condutor neutro jamais deverá ser interrompido.



Figura 7: Tela inicial do aplicativo.



Figura 8: Tela Menu do Aplicativo.



Figura 9: Tela controle de iluminação (sala).



Figura 10: Tela controle de iluminação (quarto).



Figura 11: Tela controle de iluminação (cozinha).



Figura 12: Tela de controle geral do sistema.

### 3.4. Montagem do circuito

A figuras 13 e 14 mostram a montagem da central de comando responsável pela operação do sistema.



Figura 13: Detalhe da montagem da central de comando.

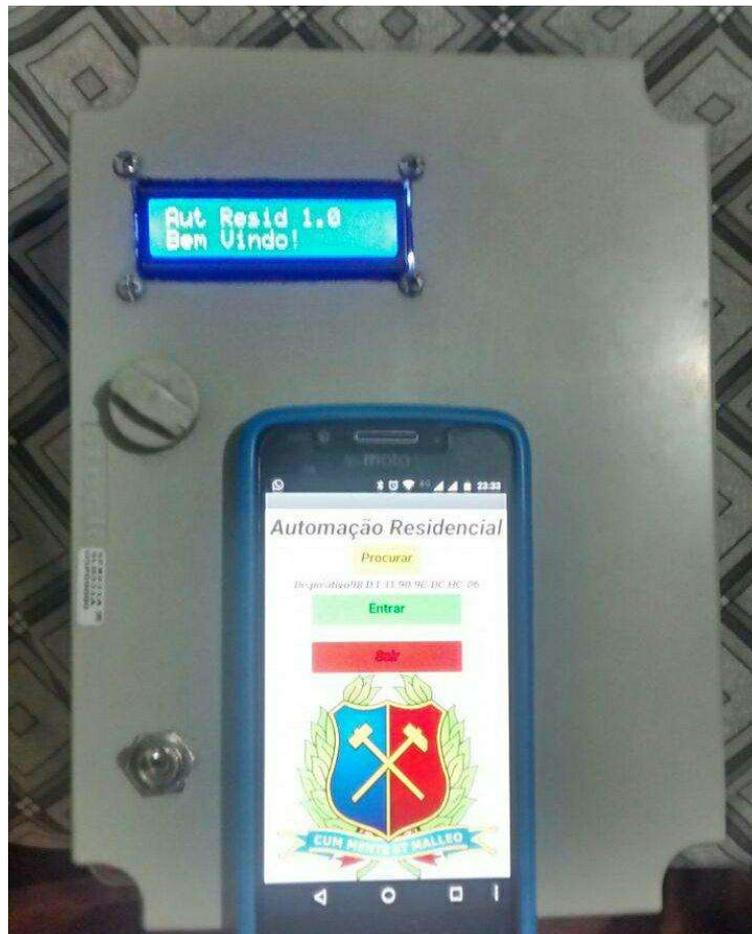


Figura 14: Vista frontal da central de comando e aplicativo instalado no *smartphone*.

## 4. TESTES

Foi desenvolvido um protótipo do sistema com todas as funcionalidades descritas anteriormente. Essa seção descreve o ambiente onde esse protótipo foi testado, buscando descrever e expor os pontos positivos e suas limitações. Isso se faz importante pois as análises aqui apresentadas fornecem noções claras dos pontos críticos a serem aperfeiçoados em trabalhos futuros, além de comprovar o funcionamento correto das partes implementadas (figuras 15 e 16).



Figura 15: Mesa de testes do sistema.

### 4.1. Ambiente de teste

O ambiente de teste foi composto por uma central e um *smartphone*. A central utilizada utilizava um Arduino Mega R3, um módulo *Bluetooth* HC-06, 03 módulos de relé com 4 canais cada. Em um primeiro momento foi utilizado um *smartphone Motorola Moto G1 XT 1033*, com sistema operacional *Android 5.0* que teve desempenho satisfatório, mas como o aparelho apresentou problemas de memória, o mesmo foi substituído por um *smartphone Motorola Moto G5 XT 1672*, com sistema operacional *Android 7.0*.

Em um primeiro momento chegou-se a se utilizar um módulo *Bluetooth* HC-05 que possuía maior velocidade de comunicação e transmissão de dados. Porém o mesmo demonstrou ter várias falhas que comprometeram a estabilidade da conexão com o *smartphone*, fazendo com que seu uso fosse descartado e o módulo HC-06 fosse adotado em seu lugar.

Como foi ampliado o número de canais de relés, foi adotada um módulo com 16 canais com alimentação independente do Arduino, que se mostrou incapaz de alimentar o novo módulo devido a corrente ser superior a corrente máxima suportada pelo Arduino.

Como os pinos digitais RX e TX do Arduino trabalham com uma tensão de 5V, foi necessário que se montasse um circuito divisor de tensão, pois o módulo HC-06 suporta no máximo 3,3V. O alcance do módulo HC-06 é de 10 metros (figura 16).

O valor dos resistores a serem utilizados foi calculado pela seguinte fórmula:

$$V_{saída} = \frac{R_2}{R_2+R_1} * V_{entrada}$$

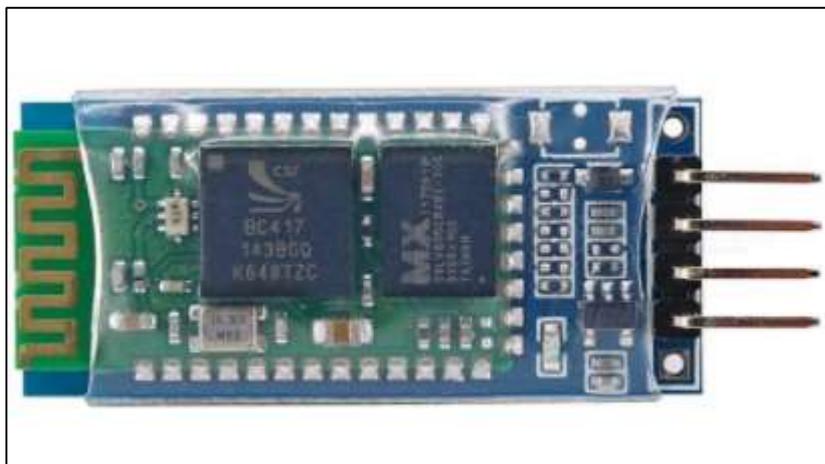


Figura 16: Módulo *Bluetooth* HC-06.



Figura 17: Módulo relé de 16 canais.

## 4.2. Testes realizados

Testes foram realizados no intuito de localizar pontos críticos na comunicação entre os módulos. Durante os testes, percebeu-se a conexão entre *smartphone* e *Bluetooth* era realizada somente uma única vez, impossibilitando que o aplicativo do dispositivo móvel mudasse de tela. O problema foi resolvido realizando uma mudança na programação do aplicativo, fazendo a desconexão automática no momento da mudança de tela, sendo que quando a nova opção fosse selecionada, a conexão era feita novamente, também de forma automática.

### 4.2.1. Controle de cenário individual

Nessa configuração, o aplicativo dá o comando para um único cenário (tela), onde se escolhe o objeto que deverá ser controlado. Uma vez que o comando é enviado, a central executará a ação somente nos componentes correspondentes ao cenário escolhido. Nessa seção, pode-se também ligar a iluminação por meio de um temporizador.

#### **4.2.2. Controle coletivo de cenários**

Nessa configuração, o aplicativo dá o comando para todos os cenários disponíveis, onde todos os objetos são selecionados, podendo ser habilitados ou desabilitados de forma simultânea.

#### **4.2.3. Habilitando/desabilitando o sistema**

O sistema possui uma chave seletora onde pode ser habilitado (modo automação) ou desabilitado (modo manual). A ideia para a introdução deste mecanismo é possibilitar ao usuário que ele use (ou não) o sistema automatizado.

## 5. CONCLUSÃO E TRABALHOS FUTUROS

O projeto buscou aproveitar as tecnologias já integradas aos *smartphones* e adaptadores *Bluetooth*, tecnologias bastante consolidadas e de baixo custo. Com isso, o baixo investimento em equipamentos e a quantidade cada vez maior de *smartphones* com sistema operacional Android são os pontos mais favoráveis para que esse tipo de sistema seja utilizado comercialmente.

Uma ideia para trabalhos futuros seria a substituição do módulo *Bluetooth* por um módulo *Wi-Fi* com o objetivo de melhorar a comunicação entre a central e o *smartphone*, permitindo também uma troca de dados mais segura. Em um segundo momento, será estudada a viabilidade de módulos sem fio, com o objetivo de reduzir o volume de cabeamento utilizado em um sistema de automação. É uma ideia que já é utilizada nos dias atuais, mas vários sistemas ainda fazem uso do sistema de ligação convencional.

Nos dias de hoje, quando se fala em automação residencial, encontramos sistemas diferentes de todos os tipos, mas como o tema é extremamente amplo, sempre há possibilidades para evolução dos sistemas existentes e surgimento de novas ideias a fim de melhorar o cotidiano do ser humano.

O sistema desenvolvido neste trabalho serviu como prova de conceitos. Para que o mesmo seja usado comercialmente, precisaria passar por algumas melhorias para garantir um funcionamento mais estável e seguro.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] “O que é automação residencial”, disponível em <http://www.forumdaconstrucao.com.br/conteudo.php?a=11&Cod=980>; Acesso em 10/03/2016.

[2] Droege, P. (ed.), (1997) Intelligent Environments -- Spatial Aspect of the Information Revolution, Oxford, Elsevier.

[3] “Arduino”, disponível em: [https://www.robocore.net/modules.php?name=GR\\_LojaVirtual&prod=120](https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=120); Acesso em 11/03/2016.

[4] “Infowester”, disponível em: <http://www.infowester.com/bluetooth.php>; Acesso em 11/03/2016.

[5] “Android passa Windows e se torna o sistema operacional mais usado do mundo”, disponível em: <https://g1.globo.com/tecnologia/noticia/android-passa-windows-e-se-torna-o-sistema-operacional-mais-usado-do-mundo.ghtml>; Acesso em 01/05/2017.

[6] “Arduino”, disponível em: <https://www.arduino.cc/en/Main/Software>; Acesso em 20/01/2018.

## 7. APÊNDICES

### 7.1. APÊNCICE A – Código do programa do Arduino

```
/*Código TCC
```

```
* Helder José Santos Luz
```

```
* Universidade Federal de Ouro Preto - UFOP
```

```
* Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais  
- DECAT
```

```
*/
```

```
#include "LiquidCrystal.h"
```

```
LiquidCrystal lcd(30, 31, 35, 34, 33, 32);
```

```
  // Serial para o modulo bluetooth
```

```
#include "SoftwareSerial.h"
```

```
SoftwareSerial bluetooth(0,1); // RX, TX
```

```
// Pinos onde está conectado o módulo de relês
```

```
const int rele_1 = 4;
```

```
const int rele_2 = 5;
```

```
const int rele_3 = 6;
```

```
const int rele_4 = 43;
```

```
const int rele_5 = 8;
```

```
const int rele_6 = 9;
```

```
//const int rele_7 = 41;
```

```
//const int rele_8 = 42;
```

```
//const int rele_9 = 43;
```

```
//const int rele_10 = 44;
```

```
//const int rele_11 = 45;
```

```
//const int rele_12 = 46;
```

```
int selecionado = rele_1;
int sel3 = rele_3;
int sel5 = rele_5;

// Constantes para ara facilitar a leitura do código
const int LIGADO = LOW;
const int DESLIGADO = HIGH;

void setup() {

    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("Aut Resid 1.0");
    lcd.setCursor (0,1);
    lcd.print ("Bem Vindo!");

    // inicalização das saídas
    digitalWrite(rele_1, DESLIGADO);
    pinMode(rele_1, OUTPUT);

    digitalWrite(rele_2, DESLIGADO);
    pinMode(rele_2, OUTPUT);

    digitalWrite(rele_3, DESLIGADO);
    pinMode(rele_3, OUTPUT);

    digitalWrite(rele_4, DESLIGADO);
    pinMode(rele_4, OUTPUT);

    digitalWrite(rele_5, DESLIGADO);
```

```
pinMode(rele_5, OUTPUT);

digitalWrite(rele_6, DESLIGADO);
pinMode(rele_6, OUTPUT);

/*digitalWrite(rele_7, DESLIGADO);
pinMode(rele_7, OUTPUT);

digitalWrite(rele_8, DESLIGADO);
pinMode(rele_8, OUTPUT);

digitalWrite(rele_9, DESLIGADO);
pinMode(rele_9, OUTPUT);

digitalWrite(rele_10, DESLIGADO);
pinMode(rele_10, OUTPUT);

digitalWrite(rele_11, DESLIGADO);
pinMode(rele_11, OUTPUT);

digitalWrite(rele_12, DESLIGADO);
pinMode(rele_12, OUTPUT);*/

// Porta serial - conversa com o PC
Serial.begin(9600);
Serial.println("Iniciado");

// porta serial - conversa com o modulo BlueTooth
bluetooth.begin(9600);
}
```

```
void loop() {  
  // processa bluetooth  
  if (bluetooth.available()) {  
    parser (bluetooth.read());  
  }  
  
  // processa Serial  
  if (Serial.available()) {  
    parser (Serial.read());  
  }  
}  
  
// Função que analisa os caracteres enviados  
// e executa os comandos apropriados  
void parser (char incoming) {  
  Serial.print(incoming);  
  
  switch (incoming) {  
    case 'A':  
    case 'a':  
      selecionado = rele_1;  
      lcd.begin (16,2);  
      lcd.setCursor (0,0);  
      lcd.print ("Quarto");  
      lcd.setCursor (0,1);  
      lcd.print ("Lamp 1 selec");  
      break;
```

```
case 'B':
```

```
case 'b':
```

```
    selecionado = rele_2;  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("Quarto");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 2 selec");  
    break;
```

```
case 'C':
```

```
case 'c':
```

```
    sel3 = rele_3;  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("Cozinha");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 3 selec");  
    break;
```

```
case 'D':
```

```
case 'd':
```

```
    sel3 = rele_4;  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("Cozinha");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 4 selec");  
    break;
```

```
case 'E':  
case 'e':  
    sel5 = rele_5;  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("Sala");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 5 selec");  
    break;  
  
case 'F':  
case 'f':  
    sel5 = rele_6;  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("Sala");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 6 selec");  
    break;  
  
case '+':  
    print_rele();  
    Serial.println("LIGA");  
    digitalWrite(seleccionado, LIGADO);  
    if (seleccionado == rele_1){  
        lcd.setCursor (0,0);  
        lcd.print ("QUARTO");
```

```
lcd.setCursor (0,1);
lcd.print ("Lamp 1 ON  ");
}
if (seleccionado == rele_2){
lcd.setCursor (0,0);
lcd.print ("QUARTO");
lcd.setCursor (0,1);
lcd.print ("Lamp 2 ON  ");
}
break;
case '-':
print_rele();
Serial.println("desliga");
digitalWrite(seleccionado, DESLIGADO);
if (seleccionado == rele_1){
lcd.setCursor (0,0);
lcd.print ("QUARTO");
lcd.setCursor (0,1);
lcd.print ("Lamp 1 OFF  ");
}
if (seleccionado == rele_2){
lcd.setCursor (0,0);
lcd.print ("QUARTO");
lcd.setCursor (0,1);
lcd.print ("Lamp 2 OFF  ");
}
break;

case '#': //Modo automático
if (seleccionado == rele_1){
```

```
digitalWrite (seleccionado, LIGADO);  
lcd.begin (16,2);  
lcd.setCursor (0,0);  
lcd.print ("QUARTO L1 AUT");  
lcd.setCursor (0,1);  
lcd.print ("Timer ON.");  
delay (10000);  
digitalWrite (seleccionado, DESLIGADO);  
lcd.begin (16,2);  
lcd.setCursor (0,0);  
lcd.print ("QUARTO L1 AUT");  
lcd.setCursor (0,1);  
lcd.print ("Timer OFF.");  
}  
if (seleccionado == rele_2){  
digitalWrite (seleccionado, LIGADO);  
lcd.begin (16,2);  
lcd.setCursor (0,0);  
lcd.print ("QUARTO L2 AUT");  
lcd.setCursor (0,1);  
lcd.print ("Timer ON.");  
delay (10000);  
digitalWrite (seleccionado, DESLIGADO);  
lcd.begin (16,2);  
lcd.setCursor (0,0);  
lcd.print ("QUARTO L2 AUT");  
lcd.setCursor (0,1);  
lcd.print ("Timer OFF.");  
delay (10000);
```

```
}  
break;  
  
/* case ':': //Modo manual  
if (seleccionado == rele_1){  
    digitalWrite (rele_7, LIGADO);  
    digitalWrite (rele_1, DESLIGADO);  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("QUARTO");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 1 Manual ");  
}  
if (seleccionado == rele_2){  
    digitalWrite (rele_8, LIGADO);  
    digitalWrite (rele_2, DESLIGADO);  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("QUARTO");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 2 Manual");  
}  
break;*/  
  
case '/':  
    print_rele();  
    Serial.println("LIGA");  
    //seleccionado = rele_3;  
    /*lcd.begin (16,2);  
    lcd.setCursor (0,0);
```

```
lcd.print ("COZINHA");
lcd.setCursor (0,1);
lcd.print ("LIGADO");*/
digitalWrite(sel3, LIGADO);
if (sel3 == rele_3){
    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("COZINHA");
    lcd.setCursor (0,1);
    lcd.print ("Lamp 3 ON ");
}
if (sel3 == rele_4){
    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("COZINHA");
    lcd.setCursor (0,1);
    lcd.print ("Lamp 4 ON ");
}
break;

case '*':
    print_rele();
    Serial.println("desliga");
/* lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("COZINHA");
    lcd.setCursor (0,1);
    lcd.print ("DESLIGADO");*/
    digitalWrite(sel3, DESLIGADO);
    if (sel3 == rele_3){
```

```
    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("COZINHA");
    lcd.setCursor (0,1);
    lcd.print ("Lamp 3 OFF ");
}
if (sel3 == rele_4){
    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("COZINHA");
    lcd.setCursor (0,1);
    lcd.print ("Lamp 4 OFF ");
}
break;

case '?': //Modo automático
if (sel3 == rele_3){
    digitalWrite (sel3, LIGADO);
    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("COZINHA L3 AUT");
    lcd.setCursor (0,1);
    lcd.print ("TIMER ON");
    delay (10000);
    digitalWrite (sel3, DESLIGADO);
    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("COZINHA L3 AUT");
    lcd.setCursor (0,1);
    lcd.print ("TIMER OFF");
```

```
}  
if (sel3 == rele_4){  
    digitalWrite (sel3, LIGADO);  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("COZINHA L4 AUT");  
    lcd.setCursor (0,1);  
    lcd.print ("TIMER ON");  
    delay (10000);  
    digitalWrite (sel3, DESLIGADO);  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("COZINHA L4 AUT");  
    lcd.setCursor (0,1);  
    lcd.print ("TIMER OFF");  
}  
break;  
  
/* case ':': //Modo manual  
if (sel3 == rele_3){  
    digitalWrite (rele_9, LIGADO);  
    digitalWrite (rele_3, DESLIGADO);  
    lcd.begin (16,2);  
    lcd.setCursor (0,0);  
    lcd.print ("COZINHA");  
    lcd.setCursor (0,1);  
    lcd.print ("Lamp 3 Manual ");  
}  
if (sel3 == rele_4){
```

```
digitalWrite (rele_10, LIGADO);
digitalWrite (rele_4, DESLIGADO);
lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("COZINHA");
lcd.setCursor (0,1);
lcd.print ("Lamp 4 Manual");
}
break;*/
```

```
case '$':
print_rele();
Serial.println("LIGA");
/*seleccionado = rele_5;
lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("SALA");
lcd.setCursor (0,1);
lcd.print ("LIGADO");*/
digitalWrite(sel5, LIGADO);
if (sel5 == rele_5){
    lcd.begin (16,2);
    lcd.setCursor (0,0);
    lcd.print ("SALA");
    lcd.setCursor (0,1);
    lcd.print ("Lamp 5 ON ");
}
if (sel5 == rele_6){
    lcd.begin (16,2);
    lcd.setCursor (0,0);
```

```
lcd.print ("SALA");
lcd.setCursor (0,1);
lcd.print ("Lamp 6 ON ");
}

break;

case '%':
print_rele();
Serial.println("desliga");
/*lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("SALA");
lcd.setCursor (0,1);
lcd.print ("DESLIGADO");*/
digitalWrite(sel5, DESLIGADO);
if (sel5 == rele_5){
lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("SALA");
lcd.setCursor (0,1);
lcd.print ("Lamp 5 OFF ");
}
if (sel5 == rele_6){
lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("SALA");
lcd.setCursor (0,1);
lcd.print ("Lamp 6 OFF ");
}
```

```
break;

case '>': //Modo automático
if (sel5 == rele_5){
  digitalWrite (sel5, LIGADO);
  lcd.begin (16,2);
  lcd.setCursor (0,0);
  lcd.print ("SALA L5 AUT");
  lcd.setCursor (0,1);
  lcd.print ("TIMER ON");
  delay (10000);
  digitalWrite (sel5, DESLIGADO);
  lcd.begin (16,2);
  lcd.setCursor (0,0);
  lcd.print ("SALA L5 AUT");
  lcd.setCursor (0,1);
  lcd.print ("TIMER OFF");

}

if (sel5 == rele_6){
  digitalWrite (sel5, LIGADO);
  lcd.begin (16,2);
  lcd.setCursor (0,0);
  lcd.print ("SALA L6 AUT");
  lcd.setCursor (0,1);
  lcd.print ("TIMER ON");
  delay (10000);
  digitalWrite (sel5, DESLIGADO);
  lcd.begin (16,2);
  lcd.setCursor (0,0);
```

```
lcd.print ("SALA L6 AUT");
lcd.setCursor (0,1);
lcd.print ("TIMER OFF");
}
break;

/*case '<': //Modo manual
if (sel5 == rele_5){
digitalWrite (rele_11, LIGADO);
digitalWrite (rele_5, DESLIGADO);
lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("SALA");
lcd.setCursor (0,1);
lcd.print ("Lamp 5 Manual ");
}
if (sel5 == rele_6){
digitalWrite (rele_12, LIGADO);
digitalWrite (rele_6, DESLIGADO);
lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("SALA");
lcd.setCursor (0,1);
lcd.print ("Lamp 6 Manual");
}
break;*/

case '@': //Ligar todo sistema
lcd.begin (16,2);
lcd.setCursor (0,0);
```

```
lcd.print ("Sistema");
lcd.setCursor (0,1);
lcd.print("Ligado");
digitalWrite(rele_1, LIGADO);
digitalWrite(rele_2, LIGADO);
digitalWrite(rele_3, LIGADO);
digitalWrite(rele_4, LIGADO);
digitalWrite(rele_5, LIGADO);
digitalWrite(rele_6, LIGADO);
//digitalWrite(rele_7, LIGADO);
//digitalWrite(rele_8, LIGADO);
//digitalWrite(rele_9, LIGADO);
//digitalWrite(rele_10, LIGADO);
//digitalWrite(rele_11, LIGADO);
//digitalWrite(rele_12, LIGADO);
break;

case '!': //Desligar todo sistema
lcd.begin (16,2);
lcd.setCursor (0,0);
lcd.print ("Sistema");
lcd.setCursor (0,1);
lcd.print("Desligado");
digitalWrite(rele_1, DESLIGADO);
digitalWrite(rele_2, DESLIGADO);
digitalWrite(rele_3, DESLIGADO);
digitalWrite(rele_4, DESLIGADO);
digitalWrite(rele_5, DESLIGADO);
digitalWrite(rele_6, DESLIGADO);
/*digitalWrite(rele_7, DESLIGADO);
```

```
digitalWrite(rele_8, DESLIGADO);  
digitalWrite(rele_9, DESLIGADO);  
digitalWrite(rele_10, DESLIGADO);  
digitalWrite(rele_11, DESLIGADO);  
digitalWrite(rele_12, DESLIGADO);*/  
break;
```

```
}//fim switch
```

```
}
```

```
// função auxiliar para informar a saída selecionada pela serial do computador
```

```
inline void print_rele() {  
  if (selecionado == rele_1) {  
    Serial.print(" Rele 1: ");  
  }  
  else if (selecionado == rele_2) {  
    Serial.print(" Rele 2: ");  
  }  
  else {  
    Serial.print(" Indefinido: ");  
  }  
}
```