

# 8

## CAPÍTULO

# INTRODUÇÃO AO ESTUDO DA PLATAFORMA ARDUINO PARA DESENVOLVIMENTO DE APLICAÇÕES EM INTEGRIDADE ESTRUTURAL

*Bruno Pereira Barella<sup>1</sup>*

*José dos Reis V. Moura Junior<sup>1</sup>*

*Romes A. Borges<sup>1</sup>*

**Agradecimentos:** Os autores agradecem à Unidade Acadêmica Especial de Matemática e Tecnologia pelo suporte no desenvolvimento do trabalho apresentado.

---

<sup>1</sup> Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão, Catalão, Brasil.

E-mail de contato: [brunobarella@hotmail.com](mailto:brunobarella@hotmail.com), [zereis@ufg.com.br](mailto:zereis@ufg.com.br)

**Resumo:** A área de manutenção tem se desenvolvido nas últimas décadas no que é denominado monitoramento de integridade estrutural. Nessas abordagens, estruturas complexas e/ou de elevado custo são monitoradas periodicamente utilizando técnicas sofisticadas de sensoriamento e modelagem. Para início do trabalho de pesquisa nessa linha na UFG-RC, foram definidas algumas das primeiras atividades para o desenvolvimento em plataforma Arduino. O propósito específico deste trabalho foi desenvolver e apresentar os primeiros passos no desenvolvimento na plataforma Arduino utilizando comunicação via Bluetooth e conexão com um aplicativo de *smartphone* Android. O sistema foi desenvolvido e testado, apresentando confiabilidade na sua funcionalidade única, que é o acendimento via celular e conexão Bluetooth de um *led* acoplado à plataforma Arduino.

**Palavras-chave:** Monitoramento de integridade estrutural. Plataforma Arduino. App Inventor.

**Abstract:** In the past years, structural health monitoring, one of the maintenance themes, has been developed in a more mature field. In these studies, featured by complex structures or high cost systems, components are monitored periodically by the use of sophisticated techniques of sensing and modeling. Starting up this subject in UFG-RC, it was defined some initial tasks of code development in Arduino framework. The specific purpose of this contribution is to build and illustrate a first step in the Arduino framework, showing Bluetooth connectivity and an example applied to Android app. The system was developed and tested, showing high reliability in its single functionality, i.e., the system remotely turned on/off a led connected to the Arduino framework using a mobile phone.

**Keywords:** Structural health monitoring. Arduino environment. App Inventor.

## 1 INTRODUÇÃO

Devido a algumas necessidades especiais de projeto em engenharia, aliadas ao desenvolvimento de novos materiais, foram criados dispositivos e materiais que vêm sendo utilizados com sucesso como sensores e como atuadores que são facilmente incorporados a uma dada estrutura de interesse. Baseados nas características físicas desses componentes, alguns sistemas de controle significativos puderam ser implementados, substituindo, em muitos casos, servomecanismos tradicionalmente utilizados. Essas novas estruturas, denominadas “estruturas inteligentes”, utilizando-se desses novos materiais, denominados “materiais inteligentes”, vêm ganhando reconhecidas aplicações na indústria, assim como nos centros de pesquisa em todo o mundo, revolucionando o projeto de sistemas de controle e prometendo um profundo efeito nas técnicas de projeto (BANKS; SMITH; WANG, 1996; MOURA JR.; STEFFEN JR.; INMAN, 2008).

Os métodos de avaliação de integridade estrutural não destrutivos, que são também denominados técnicas de ensaios não destrutivos e, principalmente, por ensaios não destrutivos (END), são caracterizados por manter, após o ensaio de avaliação, as mesmas características da estrutura em análises anteriores ao ensaio (BRAY; McBRIDE, 1992).

Pelo seu caráter, tais ensaios são utilizados principalmente na manutenção e controle da integridade estrutural de componentes em uso. Em alguns casos, uma parada do equipamento se faz necessária para a aplicação de uma dessas técnicas, mas sobretudo uma parada temporária, sem a necessidade obrigatória da substituição de componentes após a avaliação (BRAY; McBRIDE, 1992).

Algumas outras técnicas destrutivas aplicadas em manutenção de máquinas ou equipamentos são tão precisas ou melhores que as técnicas não destrutivas para avaliação de integridade estrutural, mas em virtude da necessidade de substituição de componentes a cada avaliação, as técnicas não destrutivas têm ganhado grande atenção. Com isso, ocorreu um aumento na confiabilidade e segurança do processo, uma vez que os próprios componentes em uso passam a ser avaliados. Com a expansão da tecnologia, tornou-se importante a criação de sistemas de engenharia cujos projetos têm que considerar a futura necessidade de utilização de técnicas não destrutivas para monitorar em funcionamento (BRAY; McBRIDE, 1992).

Nesse contexto, foi proposto o tema de pesquisa em monitoramento de integridade estrutural, cujo desenvolvimento envolve implementações de circuitos eletrônicos (GIBILISCO, 2002) para monitoramento de sinais e acoplamentos às pastilhas piezoelétricas utilizadas em estruturas inteligentes (MOURA JR.; STEFFEN JR., 2006). O trabalho apresentado aqui foi o início do desenvolvimento de três conceitos chave para a implementação dos dispositivos que serão trabalhados: plataforma Arduino, aplicações wireless por Bluetooth e desenvolvimento de aplicativos em *smartphones*.

A plataforma Arduino surgiu na cidade de Ivrea, na Itália, em 2005, com o intuito de ensinar programação e prototipagem a um baixo custo (McROBERTS, 2011). A placa de código aberto *open source* baseia-se em um circuito simples de entradas e saídas e uma fonte reguladora de tensão. A linguagem de programação é simplificada pela união da linguagem C e C++, e seu *hardware* é feito através de um microcontrolador Atmel AVR de 8 bits. O componente da fonte reguladora de tensão, na grande maioria das placas, contém um cristal oscilador de 16 MHz. Os microcontroladores vem com um *bootloader* pré-programado para facilitar a programação, ou seja, sem a necessidade de ter um *chip* programador externo. No mercado, encontram-se muitas placas, as quais se diferenciam pela velocidade de processamento, quantidade de memória *flash* e tamanho da placa (McROBERTS, 2011).

O Arduino IDE é uma aplicação escrita em Java, desenvolvida com o propósito de introduzir a programação a leigos (McROBERTS, 2011). Ela contém recursos capazes de compilar e enviar o código à placa com um único clique do mouse e uma biblioteca chamada *Wiring*, que possui a capacidade de programação em C/C++, permitindo assim a criação de diversas operações de entrada e saída, sendo necessário definir apenas duas funções para produzir um programa funcional:

- *setup* (), inserida logo após a declaração das variáveis, com a finalidade de inicializar as configurações das pinagens ou portas;
- *loop* (), chamada de bloco de repetição.

A principal função do Arduino em um sistema é facilitar a prototipagem e implementação do controle do sistema interativo da mesma forma que um CLP (controlador lógico programável) controla sistemas de funcionamento industriais (NUNES, 2002).

Atualmente, encontram-se no mercado diversas placas, e as mais utilizadas são a UNO, a MEGA 2560 (utilizada no projeto) e a DUE.

A placa UNO contém um microcontrolador ATmega 328 que tem 32 KB de memória *flash* e 14 pinos de I/O, dos quais 6 são para saídas analógicas. O ponto fraco dessa placa é a quantidade de entradas e saídas, que são relativamente poucas, e a quantidade de memória; o ponto forte é que a placa é a mais barata encontrada no mercado.

A placa MEGA 2560 contém um microcontrolador ATmega 2560, que tem 256 KB de memória *flash* e 54 pinos de I/O, dos quais 16 são para saídas analógicas. O ponto fraco da placa é que não pode se trocar o microcontrolador da placa como na placa UNO, e os pontos fortes são o grande número de entradas e saídas e a quantidade de memória disponível.

A placa DUE contém um microcontrolador AT915AM3X8E que tem 512 KB de memória *flash* e 54 pinos de I/O, dos quais 12 são para saídas analógicas. O ponto fraco da placa é que ela trabalha com uma tensão de operação diferente, que gira em torno dos 3.3 V, e atualmente é uma das placas mais caras; o ponto forte e que a difere das outras placas é a velocidade do *clock*, que é de 84 MHz, enquanto nas outras placas é de apenas 16 MHz. A placa também suporta uma corrente de 130 mA, maior do que a convencional encontrada nas placas, de 40 mA.

Observando a aplicação Bluetooth, a qual foi desenvolvida em 1994 pela Ericsson, buscando uma forma barata e eficiente de comunicação sem fio entre celulares e seus acessórios, em 1998 as grandes indústrias do ramo realizaram um consórcio para conduzir e aprofundar o estudo dessa forma de conexão, formando o chamado Bluetooth Special Interest Group (MONK, 2014). O nome

Bluetooth veio em homenagem a um rei da Dinamarca e Noruega, pois o mesmo unificou as tribos de seu país, o mesmo objetivo do projeto Bluetooth.

No atual projeto, o componente utilizado com a função de transmissão de dados foi o HC-06, que tem um alcance relativamente alto, pois o mesmo é da classe 2, com um alcance de até 10 metros e expansível se conectado a um dispositivo da classe 1, tensão de operação compatível com a fornecida na placa e uma comunicação padrão de 9.600 bps.

A plataforma de desenvolvimento do aplicativo, chamada de App Inventor, foi inventada pelo professor Harold Abelson, do MIT, em 2009, e permite criar aplicativos para *smartphone* sem conhecimento de programação. Foi testada em diversas instituições voltadas à educação e, em 2011, virou uma plataforma de código aberto.

Além de disponibilizar recursos de conectividades das mais diversas formas, o App Inventor possui um *software* que simula um aparelho celular com o intuito de verificar e visualizar a aplicação desenvolvida. Depois de verificar as funcionalidades das aplicações, a plataforma disponibiliza os aplicativos nas diversas lojas para realização de *downloads*.

## 2 OBJETIVOS

O presente trabalho buscou o desenvolvimento introdutório e didático de:

- introdução à modelagem de circuitos usando o *software* Fritzing;
- desenvolvimento do circuito usando tecnologia Bluetooth na plataforma Arduino;
- implementação em *software* App Inventor de sistema de comunicação com Arduino via Bluetooth para *smartphone*.

## 3 MÉTODOS

### 3.1 Plataforma Arduino

A primeira etapa do desenvolvimento consistiu na simulação do circuito e sua implementação física. O circuito foi montado e projetado no *software* Fritzing, o qual possibilitou realizar simulações com os componentes disponíveis do Arduino, além de permitir a simulação para análise de seu comportamento eletrônico.

O *software* Fritzing simula uma prototipagem e consegue rodar a programação em diversos componentes, simulando, assim, o projeto por completo. Para utilizá-lo, deve-se saber quais componentes serão utilizados para procurar em sua biblioteca de componentes. A Figura 1 apresenta a área de busca de componentes do *software*.

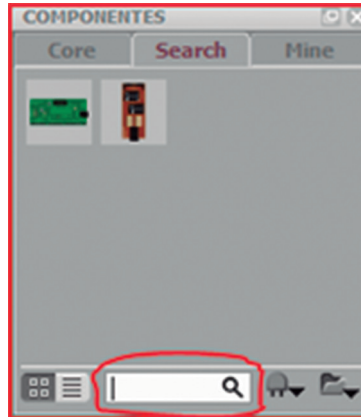


Figura 1. Busca de componentes.

Em seguida, deve-se configurar as propriedades de cada componente, por exemplo, quantos ohms um resistor deverá ter, qual a capacitância que um capacitor deverá possuir, e outras. Na Figura 2, pode-se observar o local onde se configuram as propriedades dos componentes selecionados.

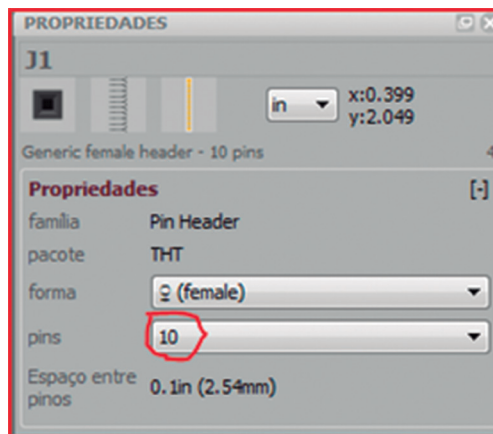
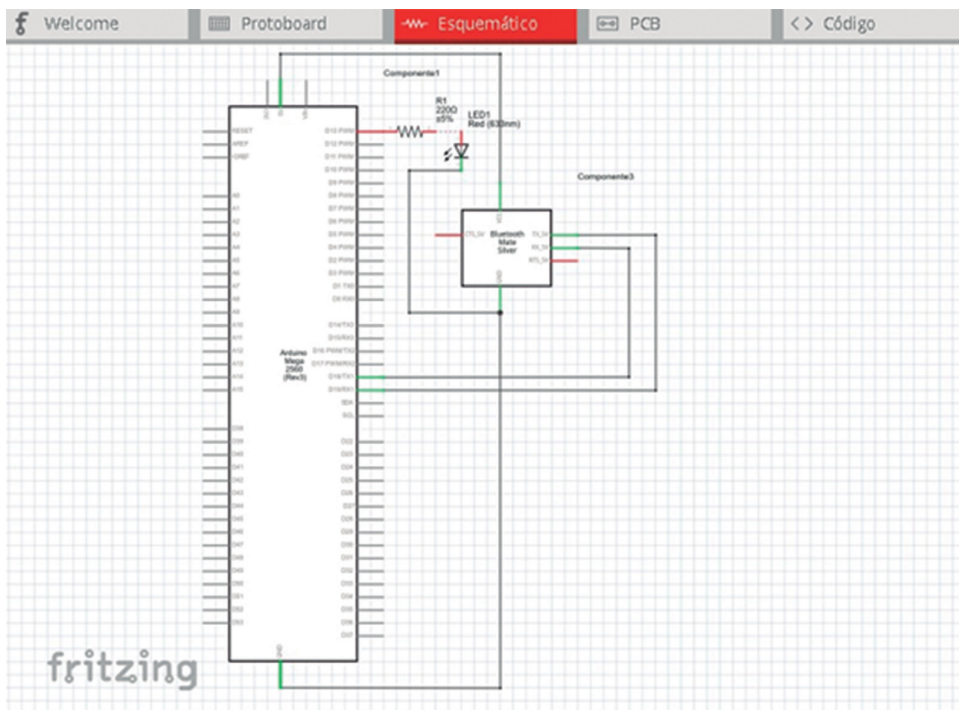


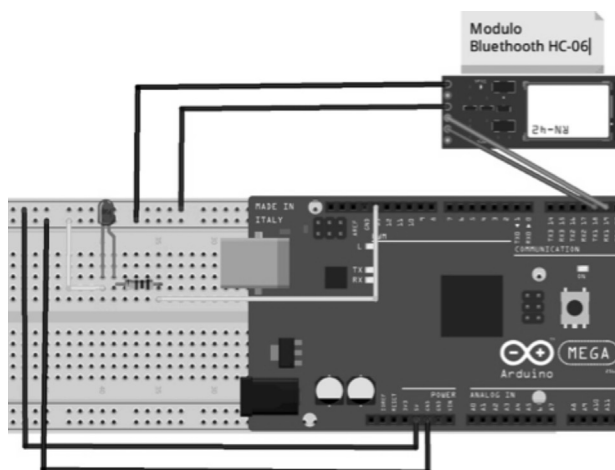
Figura 2. Propriedades dos componentes.

Feitas as configurações, o próximo passo é realizar as ligações dos componentes na *proto*board do sistema. O *software* tem 4 abas: “Protoboard”, “Esquemático”, “PCB” e “Código”. A primeira aba é a prototipagem do sistema; a segunda é o esquemático do projeto, a referência de todo o projeto; a terceira é o desenvolvimento da placa PCB, que poderá ser desenvolvida no próprio *software*, caso necessário; e a quarta aba é a do código, onde se pode programar microcontroladores, placas e componentes programáveis. A Figura 3 apresenta as abas do *software* e o esquemático do projeto.



**Figura 3.** Abas do Fritzing e esquemático do projeto.

A Figura 4 apresenta o circuito desenvolvido para o trabalho encontrado na aba “Protoboard” contendo o módulo Bluetooth, placa Arduino, *protoboard*, *led* (diodo emissor de luz) e um resistor de 330 ohms. A alimentação do módulo e da placa está representada pelas duas primeiras fileiras da *protoboard*.



**Figura 4.** Circuito de ligações da placa e do módulo Bluetooth.

Para o trabalho em questão, a montagem foi realizada utilizando um Arduino Mega 2560, uma placa *protoboard*, um resistor de 330 ohms, um *led* de 5 mm e um módulo Bluetooth do modelo HC-06. Como a funcionalidade desejada era acender um *led*, a porta 13 foi conectada ao resistor de 330 ohms e ao *led* em série. A função do resistor é proporcionar uma queda de tensão elétrica para que o *led* não queime, pois a saída digital da porta 13 é de 5 volts e o *led* vermelho opera com uma tensão na faixa de 1,9-2,1 volts e uma corrente de 20 mA contínua, 50 mA para pico de 10%.

O acionamento do *led* é realizado por meio do módulo Bluetooth, que tem como funcionalidade enviar e receber dados por meio das portas RX (recebe dados) e TX (transmite dados). A montagem na placa dos pinos ocorre de forma invertida: o RX do módulo se conecta no TX0 da placa, e o TX do módulo, no TX0 da placa. A alimentação do módulo é de 5 volts, conectado à tensão que o pino 5V da placa fornece, e o terra do módulo foi conectado ao GND da placa.

A placa Arduino Mega 2560 opera com uma tensão de 5 a 12 volts, e a tensão de funcionamento de todos os pinos é de 5 volts. Portanto, foi utilizada a porta USB do *notebook* como a fonte de alimentação da placa. O desenvolvimento do código para a plataforma Arduino foi realizado em linguagem C/C++ própria do Arduino. A Figura 5 apresenta o código em linguagem C desenvolvido para o propósito específico da aplicação.

```
int ledpin = 13;
String readString;

void setup(){
  Serial.begin(9600);
  pinMode(ledpin, OUTPUT);
}

void loop(){
  while (Serial.available()){
    delay(3);
    char c = Serial.read();
    readString += c;
  }
  if (readString.length() >0){
    Serial.println(readString);
    if(readString == "ON")
    {
      digitalWrite(ledpin,HIGH);
    }
    if (readString == "OFF")
    {
      digitalWrite(ledpin, LOW);
    }
    readString="";
  }
}
```

Figura 5. Programação na IDE do Arduino.



O código se inicia com a declaração da variável *ledpin* do tipo *int* (inteiro), e indica o endereçamento da porta 13 da placa. Em seguida, a variável *readString*, do tipo *string* (texto), é declarada como a função que irá armazenar a palavra ou mensagem de comando recebida pela porta serial. Logo, a porta serial é inicializada com a velocidade de transmissão serial de 9.600 bits por segundo, e o pino 13 é definido como saída.

Na função *void loop* (), é declarado que, enquanto houver algum bit na porta serial do Arduino, ele irá realizar essa função, a qual utiliza um tempo de verificação de 3 em 3 segundos e faz a chamada da função *Serial.read* () para atribuir seu resultado de leitura para a variável *c*, do tipo *char* (caractere), e guarda o primeiro byte disponível na porta serial.

Em seguida, se o que a função tenha lido na porta serial for maior que 0, o programa imprime na porta serial a palavra guardada na variável *readString*. Se a variável ou palavra for igual a *ON*, aciona-se a porta 13 (nível lógico alto), mas se a variável for igual a *OFF*, desativa-se a porta 13 (nível lógico baixo).

## 3.2 Aplicativo no *smartphone*

Em seguida ao desenvolvimento do circuito eletrônico e seu teste/validação, foi realizada a implementação do código para *smartphone* em plataforma App Inventor. A ferramenta App Inventor possibilita a criação de aplicativos para *smartphones* e possui um editor de interface gráfica através de um simples “arrasto” de componentes, como é o caso de botões, *labels*, recursos, listas etc.

A página de desenho da interface da aplicação é constituída por 4 zonas fundamentais: “Palette”, “Viewer”, “Components” e “Properties”. Na zona “Palette”, pode-se encontrar todos os tipos de componentes de uma aplicação, como é o caso de botões, listas, caixas de seleção, etiquetas etc. Na zona “Viewer”, a qual imita um ecrã de um dispositivo Android, pode-se ver o aspecto da futura aplicação, e é possível arrastar da “Palette” os componentes para o ecrã do “Viewer”.

Na zona “Components”, pode-se encontrar a lista de componentes que já foram utilizados anteriormente na aplicação, numa forma de hierarquia, vendo que componente depende de qual. As propriedades de um componente podem ser definidas na zona “Properties” – por exemplo, se for um botão, vai-se até à lista de componentes ou ao ecrã e seleciona-se o botão que se quer alterar; serão exibidos na zona de propriedades todas as propriedades deste, permitindo alterar a cor, o texto, a posição, o tamanho etc. A interface de desenvolvimento do aplicativo pode ser observada na Figura 6. No canto superior direito estão dois botões que permitem mudar da página de desenho de interface para a página de edição de blocos.

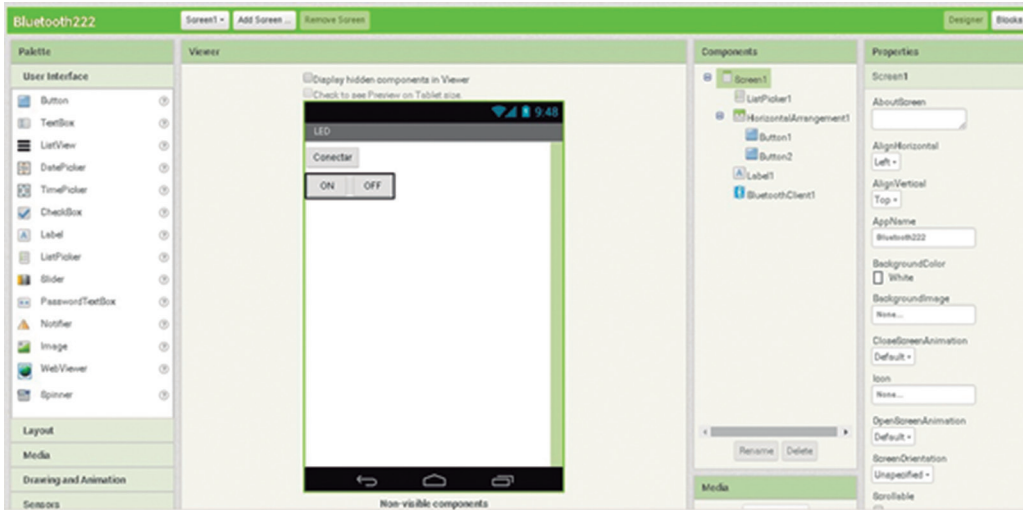


Figura 6. Interface de desenvolvimento.

A Figura 7 apresenta a interface de edição de blocos de programação, uma visão geral da tela de edição.

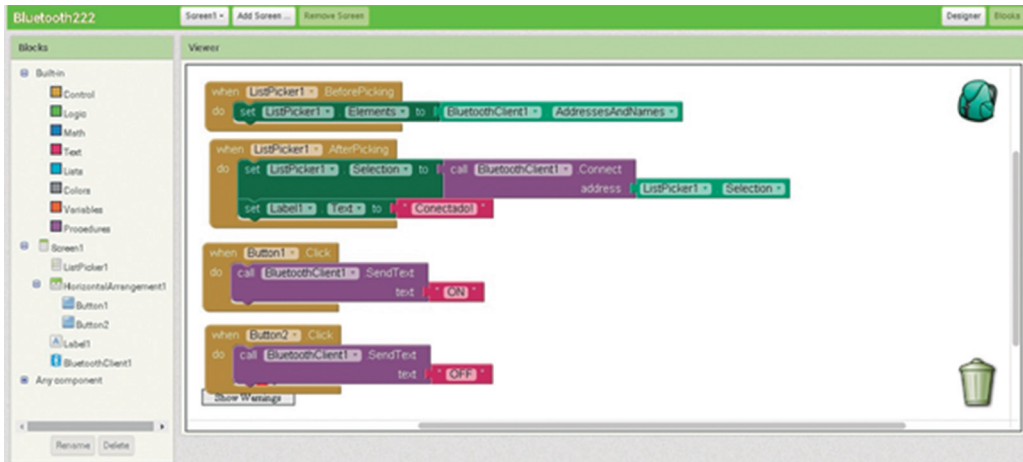
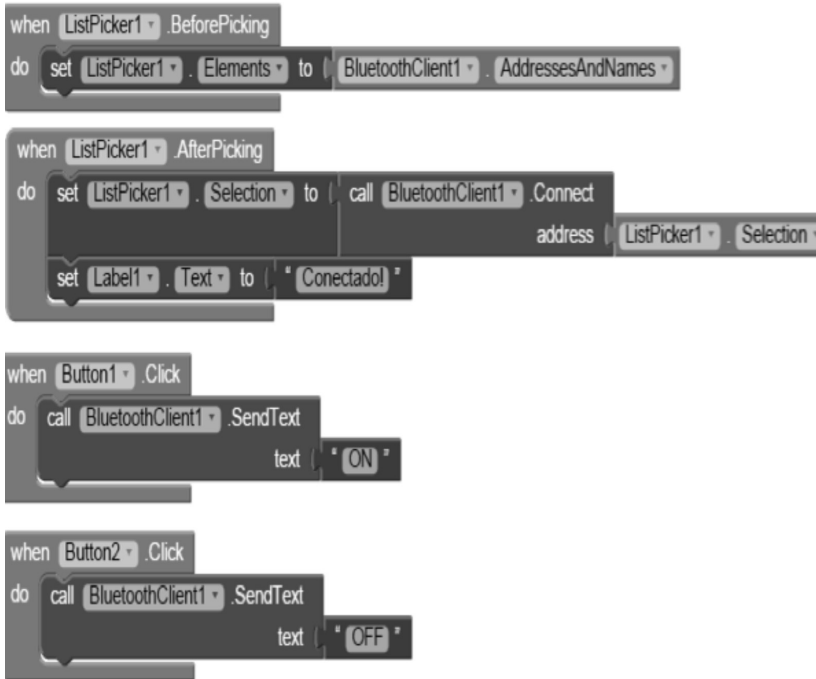


Figura 7. Edição dos blocos de programação.

O código desenvolvido para o aplicativo em estudo no trabalho está ilustrado na Figura 8.

A plataforma para desenvolvimento desse aplicativo foi escolhida por ser de fácil implementação, gratuidade e modularidades com funcionalidades avançadas, como o uso de módulos Bluetooth, como o utilizado no estudo de caso.



**Figura 8.** Programação para o aplicativo.

O código é composto por quatro blocos, que foram designados desta forma: o primeiro bloco define o uso de um botão que, quando clicado, gera uma função retornando uma lista com elementos disponíveis para pareamento. No segundo bloco, a função define que, depois que o dispositivo for selecionado na lista gerada pela primeira função e o pareamento for realizado com sucesso, uma mensagem aparecera na *label1* (“conectado!”). No terceiro bloco, o programa envia a *string ON* para o módulo Bluetooth se o elemento *Button1* (botão no aplicativo) for pressionado. No quarto e último bloco, o aplicativo também envia uma *string OFF* se o botão *Button2* for pressionado.

## 4 RESULTADOS

O objetivo específico de desenvolver um aplicativo inicial para *smartphone* que controlasse via Bluetooth um dispositivo na plataforma Arduino foi alcançado com sucesso. Realizou-se testes de conectividade do módulo com o *smartphone*, verificando por um determinado tempo se haveria algum atraso na transmissão/recebimento de dados, porém o sistema não apresentou nenhuma anomalia. Verificou-se a distância máxima da conectividade, a qual apresentou perfeito funcionamento em aproximadamente 9 metros de distância do módulo Bluetooth.

O único problema constatado foi que o módulo Bluetooth teve um aquecimento, pois foi submetido a um funcionamento de aproximadamente 5 horas. Para solucionar o problema, pode-se utilizar um dissipador de calor juntamente com uma pasta térmica.

A Figura 9 ilustra a foto do sistema físico desenvolvido com o *smartphone*.

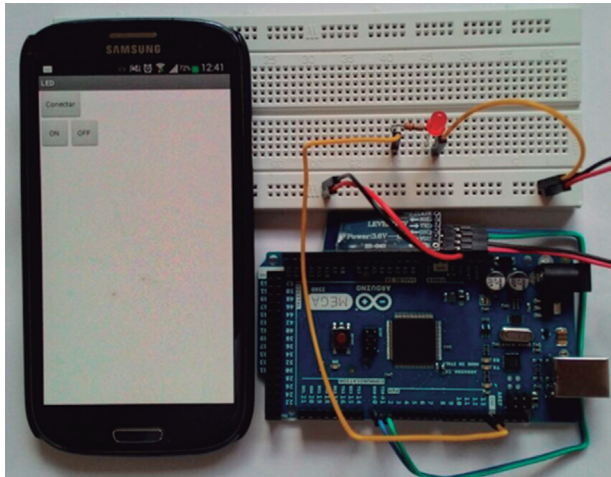


Figura 9. Resultado do sistema.

## 5 DISCUSSÃO

O desenvolvimento do sistema como um todo encarou certos desafios, como, por exemplo, a compatibilidade da tecnologia Bluetooth com o desenvolvimento do aplicativo e também a necessidade de se pesquisar funções na linguagem de programação que se adaptariam melhor com a comunicabilidade do sistema.

Observando o aplicativo, que se encontra bastante simples, porém muito eficiente para o objetivo desejado, destaca-se a simplicidade da programação, a qual é feita em blocos e, em seguida, convertida para a linguagem Java, assim rodando em qualquer aparelho Android que tenha uma *virtual machine*.

Analisando o *hardware*, composto pela placa Arduino e o módulo Bluetooth, este também atendeu perfeitamente as necessidades, mas, devido ao custo, poderia ter sido implementado junto a um microcontrolador 16F628A, reduzindo em cerca de 50% o preço do projeto.

O trabalho foi utilizado como um método introdutório do assunto, e posteriormente será integrado a projetos mais avançados utilizando a tecnologia sem fio, a fim de promover o conforto e a confiabilidade dos sistemas. Pode-se aplicar o atual projeto a sistemas digitais que necessitem de um controle sem fio, substi-

tuindo o *led* por um relé e acionando uma carga com tensão maior, por exemplo uma lâmpada ou máquina industrial, ou também controlar um sinal analógico de um determinado sensor/atuador piezoelétrico com o intuito de identificar falhas em estruturas. Para o meio científico, o projeto pode ser utilizado como base para o desenvolvimento de aplicações e pesquisas nas áreas de controle e na área da ciência da computação, por se tratar de um controle de sinal por meio de um *smartphone*.

## 5 CONCLUSÃO

O desenvolvimento desse projeto proporcionou a introdução do estudante de primeiro período de Matemática Industrial aos assuntos correlacionados, adquirindo prática no desenvolvimento de aplicativos e montagem de circuitos.

Os próximos passos na continuidade deste projeto são a geração de sinais e o aumento na complexidade de controle, assim como a conversão de dados analógicos-digitais-analógicos e o uso de pastilhas piezoelétricas em vez do elemento *led*.

## REFERÊNCIAS

- BANKS, H. T.; SMITH, R. C.; WANG, Y. *Smart materials structures – modeling, estimation and control*. Paris: John Wiley & Sons, 1996.
- BRAY, D. E.; McBRIDE, D. *Nondestructive testing techniques*. Nova Jersey: John Wiley and Sons, 1992.
- GIBILISCO, S. *Manual de eletrônica e de telecomunicações*. Rio de Janeiro: Reichman Affonso Editores, 2002.
- McROBERTS, M. *Arduino básico*. São Paulo: Novatec, 2011.
- MONK, S. *Projetos com Arduino e Android*. Porto Alegre: Bookman, 2014.
- MOURA JR., J. R. V.; STEFFEN JR., V. Impedance based health monitoring for aeronautic structures using statistical meta-modeling. *Journal of Intelligent Material Systems and Structures*, v. 17, n. 11, p. 1023-1036, 2006
- MOURA JR., J. R. V.; STEFFEN JR., V.; INMAN, D. J. Optimization of monitoring parameters of a space tubular structure by using genetic algorithms. In: *Proceedings of SPIE, the International Society for Optical Engineering*, 2008.
- NUNES, R. J. C. *Análise comparativa de tecnologias para domótica*. São Paulo, 2011.

