

UNIVERSIDADE NOVE DE JULHO – UNINOVE

RENATO ALEXANDRE PAVIN  
GILMAR CORRÊA DE OLIVEIRA  
WELLINGTON PIMENTEL DA SILVA

**SISTEMA PARA AUXILIAR PESSOAS COM DEFICIÊNCIA VISUAL  
NO DESCARTE CORRETO DE ITENS EM LIXEIRAS DE COLETA  
SELETIVA**

SÃO PAULO  
2013

RENATO ALEXANDRE PAVIN  
GILMAR CORRÊA DE OLIVEIRA  
WELLINGTON PIMENTEL DA SILVA

**SISTEMA PARA AUXILIAR PESSOAS COM DEFICIÊNCIA VISUAL  
NO DESCARTE CORRETO DE ITENS EM LIXEIRAS DE COLETA  
SELETIVA**

Monografia apresentada à Diretoria dos Cursos de Informática da Universidade Nove de Julho como requisito parcial para obter a Graduação de Bacharel em Ciência da Computação.

Linha de Pesquisa: Automação e Robótica

Orientadora: Prof<sup>a</sup>. Dra. Andréia Miranda Domingues

SÃO PAULO

2013

## RESUMO

De acordo com o Ministério da Saúde - MS, a pessoa com deficiência visual ou baixa visão tem sua funcionalidade comprometida, com prejuízo na sua capacidade de realização de tarefas. A questão do lixo tem sido, há muito anos, uma das maiores preocupações das Administrações Públicas e das populações dos grandes centros. Segundo o Ministério do Meio Ambiente, a reciclagem por meio de coleta seletiva, retorna os materiais descartados ou restantes de processos produtivos e de consumo para destiná-los à fabricação de novos bens, com o objetivo de economizar recursos e energia. Embora exista a possibilidade de automação de lixeiras recicláveis para facilitar o descarte correto de lixo por deficientes visuais, não existem muitos projetos voltados para esse público afetando a sua inclusão na sociedade. Assim, o presente trabalho apresenta o desenvolvimento de um sistema que auxilie pessoas com deficiência visual parcial a descartar corretamente itens em lixeiras de coleta seletiva através de comando de voz e de orientação por leds e avisos sonoros. O sistema é controlado por placa Arduino Uno, sensor de presença infra vermelho PIR, receptor de comando de voz, Buzzer e Leds. O software Arduino detecta a presença do usuário pelo sensor infravermelho e capta o comando de voz do usuário que informa o item a ser descartado (Papel, Plástico, Vidro ou Metal). A seguir, leds e avisos sonoros indicam ao usuário em qual recipiente da lixeira deve ser descartado corretamente o item informado. O sistema desenvolvido segue o padrão de cores estabelecido pelo Conselho Nacional do Meio Ambiente que é o órgão que institui normas para ações que visam qualidade de vida coletiva. Pessoas com deficiência visual podem realizar o descarte correto de itens em lixeiras de coleta seletiva. Visa também contribuir com Políticas Públicas ensinando crianças a respeitar o meio ambiente propondo práticas de descarte consciente e exercício da cidadania. O sistema possibilita adaptar lixeiras de coleta seletiva para auxiliar pessoas com a visão parcialmente comprometida, bem como adultos e crianças com menos instrução de leitura a descartarem itens corretamente.

**Palavras chave:** Lixeira de coleta seletiva, Deficiência visual, Sistema Arduino

## LISTA DE ILUSTRAÇÕES

Figura 1 - Coleta Seletiva.....	9
Figura 2 – <i>Software Arduino</i> .....	12
Figura 3 – Placa do Sistema <i>Arduino Uno</i> .....	13
Figura 4 – Sensor de Movimento PIR .....	14
Figura 5 – Dispositivo de Sinalização de Áudio ( <i>Buzzer</i> ) .....	15
Figura 6 – Esquema de instalação do <i>Buzzer</i> .....	15
Figura 7 - Modulo de Comando de Voz .....	16
Figura 8 – Interface de gravação de voz ISD1820 .....	17
Figura 9 - Esquema de Lixeira Coletora de material reciclável .....	18
Figura 10 - Tela Principal do AVEI.....	19
Figura 11 - Componentes Funcionais do <i>GuideCane</i> .....	21
Figura 12 - Protótipo de bengala de apoio a cegos - 2008.....	22
Figura 13 - Plataforma de testes desenvolvida.....	22
Figura 14 - Testes de campo com pessoas com deficiência .....	23
Figura 15 - Imagem com exemplo de marcas circulares .....	24
Figura 16 - Sistema sendo testado por um utilizador .....	25
Figura 17 - Esquema explicativo do sistema .....	26
Figura 18 - Visão geral do sistema .....	27
Figura 19 – Fluxograma de Funcionamento do Sensor <i>PIR</i> .....	28
Figura 20 – Fluxograma de Funcionamento do Sistema Geral .....	29
Figura 21 – Esquema de Funcionamento do Sistema Geral.....	30
Figura 22 – Cores Padrão das Lixeiras.....	31
Figura 23 – Esquema de conexão do Sensor de Presença <i>PIR</i> .....	33
Figura 24 – Tela do <i>Arduino</i> com o programa do Sensor de Presença .....	34
Figura 25 – Código do Programa do Sensor PIR .....	35
Figura 26 – Gravador de Voz ISD1820.....	36
Figura 27 – Gravador de Voz conectado a Placa <i>Arduino</i> .....	37
Figura 28 – Programa para habilitar o player quando for ativado o sensor.....	38
Figura 29 – Procedimento para habilitar o player .....	38

Figura 30 – Placa de Comando de Voz ( <i>Voice Recognition Module V2</i> ) .....	40
Figura 31 – Esquema de conexão da placa de Comando de Voz.....	41
Figura 32 – Conversor <i>USB TTL</i> .....	42
Figura 33 – Esquema de ligação do Módulo com o Conversor <i>USB TT</i> .....	42
Figura 34 – Tela do <i>Software</i> .....	43
Figura 35 – Gravação dos comandos para o Módulo .....	44
Figura 36 – Esquema de conexão do Módulo de Voz na placa <i>Arduino</i> .....	44
Figura 37 – Diagrama de Conexão Elétrico do Módulo de Voz na placa <i>Arduino</i> .....	45
Figura 38 – Tela 1 – Programa aguardando o comando de voz compatível.....	46
Figura 39 – Tela 2 - Ativação do sinal sonoro junto com a <i>LED Verde</i> . ....	47
Figura 40 – Tela 3 - Ativação do sinal sonoro junto com a <i>LED Amarela</i> . ....	48
Figura 41 – Tela 4 - Ativação do sinal sonoro junto com a <i>LED Vermelha</i> . ....	49
Figura 42 – Tela 5 - Ativação do sinal sonoro junto com a <i>LED Azul</i> .....	50
Figura 43 – Esquema de conexão do <i>LED</i> .....	51
Figura 44 – Código do Programa do <i>LED</i> .....	52
Figura 45 – <i>Buzzer</i> conectado a placa <i>Protoboard</i> .....	53
Figura 46 – Programação para emitir sinal sonoro do <i>buzzer</i> .....	54
Figura 47 – Diagrama Elétrico: Sinal sonoro com a <i>LED</i> . ....	55
Figura 48 – Sistema desenvolvido segue as cores padrão do CONAMA .....	56
Figura 49 – Potes sem acabamento. ....	57
Figura 50 – Criação e desenvolvimento da lixeira. ....	58
Figura 51 – Lixeira com <i>LEDs</i> e <i>buzzers</i> na parte frontal para demonstração.....	59
Figura 52 – Conexão e distribuição dos cabos através do <i>Protoboard</i> . ....	60
Figura 53 – Conexão e distribuição dos cabos. ....	60
Figura 54 – Sensor infravermelho detectando a presença de um usuário .....	61
Figura 55 – Usuário utilizando o microfone do sistema. ....	62
Figura 56 – <i>LED</i> e sinal sonoro indicando local para descarte .....	63
Figura 57 – Diagrama de Dados das Tabelas do Sistema .....	65
Figura 58 – Fluxograma de Funcionamento do Sistema Geral .....	69

## **LISTA DE TABELAS**

Tabela 1 – Custo do Sistema.....	28
----------------------------------	----

## LISTA DE ABREVIATURAS E SIGLAS

<i>AVEI</i>	Ambiente Virtual de Ensino Inteligente
<i>Braille</i>	Sistema de leitura e grafia com o tato para cegos
<i>Buzzer</i>	Sinal Sonoro
C/C++	Linguagens de programação orientada a objetos.
Dev-C++	Ambiente integrado de desenvolvimento de aplicações escritas em C/C++.
CONAMA	Conselho Nacional do Meio Ambiente
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	( <i>Integrated Development Environment</i> ), um ambiente integrado para desenvolvimento de <i>software</i>
JPEG	Joint Photographic Experts Group (Formato de codificação de imagens).
<i>LED</i>	<i>LED</i> - Light Emitting Diode (diodo emissor de luz)
MSQL	Server Microsoft SQL Server
OM	Orientação e Mobilidade
PC	Personal Computer – Computador Pessoal
PD	Pessoas com Deficiência
PIR	Passive <i>Infra Red</i>
PLC	Power Line Communication
PPDs	Pessoas Portadoras de Deficiência
RS-485	O padrão de comunicação capaz de prover uma forma bastante robusta de comunicação multiponto para controle de sistemas e na transferência de pequenas quantidades de dados, até 10Mbps.
SGBD	Sistema Gerenciador de Banco de Dados Relacional
SS	Substituição Sensorial
TWLD	<i>LED</i> com sinalizador sonoro de 80db
USB	Universal Serial Bus
WEB	World Wide Web
<i>Wiring</i>	Plataforma de prototipagem eletrônica de <i>hardware</i> livre composta por uma linguagem de programação, um ambiente de desenvolvimento integrado (IDE) e um microcontrolador de placa única.

# SUMÁRIO

## LISTA DE ILUSTRAÇÕES

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

RESUMO.....	3
<b>1. INTRODUÇÃO .....</b>	<b>7</b>
1.1 MOTIVAÇÃO / JUSTIFICATIVA.....	7
<b>2. OBJETIVOS .....</b>	<b>10</b>
2.1. OBJETIVOS GERAIS .....	10
2.2. OBJETIVOS ESPECÍFICOS .....	10
<b>3. REFERENCIAL TEÓRICO.....</b>	<b>11</b>
3.1. SISTEMA ARDUINO.....	11
3.2. LINGUAGEM WIRING.....	14
3.3. SENSOR DE MOVIMENTO PIR (PASSIVE INFRA RED.).....	14
3.4. BUZZER – DISPOSITIVO DE SINALIZAÇÃO DE ÁUDIO .....	15
3.5. MÓDULO DE COMANDO DE VOZ .....	16
3.6. GRAVADOR DE VOZ ISD1820.....	17
<b>4. REVISÃO DE LITERATURA.....</b>	<b>18</b>
4.1. PROJETO R3.....	18
4.2. AVEI – AMBIENTE VIRTUAL DE ENSINO INTELIGENTE.....	19
4.3. THE GUIDECANE – TECNOLOGIA DE ASSISTÊNCIA A DEFICIENTES VISUAIS .....	20
4.4. BENGALA DE APOIO A CEGOS COM DETECÇÃO DE BURACOS .....	21
4.5. ORIENTAÇÃO DE CEGOS USANDO VISÃO DE COMPUTADOR.....	24
4.6. AUXÍLIO À MOBILIDADE DEFICIENTES VISUAIS - ELETRO ESTIMULAÇÃO TÁTIL .....	26
<b>5. MATERIAIS E MÉTODOS .....</b>	<b>28</b>
5.1. SENSOR DE MOVIMENTO PIR .....	28
5.2. MODELO DO SISTEMA .....	30
5.3. PROCEDIMENTOS DE DESENVOLVIMENTO DO SISTEMA PROPOSTO.....	31
5.4. ESTIMATIVA DE CUSTO.....	31
5.5. DESENVOLVIMENTO DO PROJETO .....	33
<b>6 RESULTADOS E DISCUSSÕES.....</b>	<b>56</b>



6.1	FUNCIONAMENTO DA LIXEIRA .....	57
6.2	MODELAGEM DO SISTEMA – CÓDIGOS DE BARRA .....	64
6.3	BASE DE DADOS .....	64
6.3.1	DIAGRAMA DE DADOS .....	65
6.3.2	SCRIPTS DA CRIAÇÃO DA BASE DE DADOS .....	66
6.3.3	CARGA INICIAL NO SISTEMA .....	68
<b>7</b>	<b>LIMITAÇÕES E TRABALHOS FUTUROS .....</b>	<b>70</b>
<b>8</b>	<b>CONCLUSÕES .....</b>	<b>71</b>
	<b>REFERÊNCIAS .....</b>	<b>72</b>
	<b>APÊNDICE A .....</b>	<b>73</b>
	<b>APÊNDICE B .....</b>	<b>77</b>
	<b>FOLHA DE APROVAÇÃO DO PROJETO .....</b>	<b>81</b>

# 1. INTRODUÇÃO

## 1.1 Motivação / Justificativa

De acordo com dados do IBGE do Censo Demográfico de 2010, a deficiência visual atinge 35,8 milhões de pessoas sendo que 18,8% delas tem dificuldade para enxergar mesmo fazendo uso de óculos ou de lentes de contato. A deficiência visual severa caracteriza-se pela grande dificuldade de enxergar ou por não conseguir enxergar de modo algum. Ela atinge 6,6 milhões de pessoas (0,3% da população) sendo que dessas, mais de 506 mil são efetivamente cegos.

A deficiência visual é definida como a perda total ou parcial, congênita ou adquirida, da visão (Sassaki, 2003). O nível de acuidade visual pode variar o que determina dois grupos de deficiência:

- Cegueira - há perda total da visão ou pouquíssima capacidade de enxergar, o que leva a pessoa a necessitar do Sistema *Braille* como meio de leitura e escrita.
- Baixa visão ou visão subnormal - caracteriza-se pelo comprometimento do funcionamento visual dos olhos, mesmo após tratamento ou correção. As pessoas com baixa visão podem ler textos impressos ampliados ou com uso de recursos óticos especiais.

Segundo Sassaki (2003), a construção de uma verdadeira sociedade inclusiva passa também pelo cuidado com a linguagem. Na linguagem se expressa, voluntariamente ou involuntariamente, o respeito ou a discriminação em relação às pessoas com deficiências.

Ao longo dos anos, os termos que definem a deficiência foram adequando-se à evolução da ciência e da sociedade. Atualmente, o termo correto a ser utilizado é: Pessoa com Deficiência, que faz parte do texto aprovado pela Convenção Internacional para Proteção e Promoção dos Direitos e Dignidades das Pessoas com Deficiência, aprovado pela Assembleia Geral da ONU, em 2006 e ratificada no Brasil em julho de 2008.

Sabe-se que toda a população tem responsabilidade sobre o aumento da poluição, mas não se dá atenção para essa realidade. Muitas pessoas não consideram importante o ato de jogar o lixo no local correto, talvez por falta de informação e/ou incentivo. É provável que para muitas dessas pessoas, as lixeiras recicláveis são apenas latas de lixo, provavelmente por conta de não terem a oportunidade de conhecer os processos de reciclagem na escola e/ou no seu convívio social.

A questão do lixo nos grandes centros urbanos é apresentada como um dos maiores desafios a ser enfrentado pelas Administrações Públicas e pelas populações dos grandes centros urbanos. A poluição ambiental aumenta dia a dia além da falta de espaço para disposição dos resíduos, e também outras preocupações como: falta de espaço para aterros sanitários, catação em lixões, degradação dos recursos naturais, custos elevadíssimos de coleta e deposição dentre outros. Várias medidas vêm sendo adotadas ao longo dos anos, mas ainda há um grande número de pessoas que não se atentaram para a real necessidade de conservação do meio ambiente. Como é possível verificar, a realização de serviços ambientalmente adequados de limpeza urbana é imprescindível para a vida saudável da população.

Diminuir as quantidades de lixo a ser tratado e eliminado, assim como reduzir a extração de matérias-primas necessárias à produção de novos bens de consumo são as razões que devem levar a contribuir para a reciclagem através da coleta seletiva.

A Figura 1 apresenta uma pessoa com deficiência (PD) se dirigindo para a lixeira de coleta seletiva.



Figura 1 - Coleta Seletiva

A preocupação com a coleta seletiva através de lixeiras diferenciadas já faz parte do cotidiano das pessoas, porém nem sempre eficaz. As pessoas não direcionam sua atenção para o ato de jogar algo no lixo, e sem consciência e atenção acaba misturando o material seco com o material úmido, o que acaba gerando um baixo reaproveitamento desse lixo.

De uma forma geral as pessoas não têm o hábito de reciclar e os motivos podem ser vários, seja ele pela desinformação ou até mesmo pelo descaso. Projetos diferenciados e de caráter educativo podem atuar principalmente no objetivo de auxiliar as pessoas com deficiência (PD).

Embora exista a possibilidade de automação de lixeiras recicláveis para facilitar o descarte correto de lixo por deficientes visuais, não existem muitos projetos voltados para esse público afetando a sua inclusão na sociedade. Cabe ressaltar ainda, que principalmente crianças e pessoas com menos instrução de leitura também encontram problemas na identificação para descarte correto de itens em lixeiras recicláveis.

## 2. OBJETIVOS

### 2.1. Objetivos Gerais

Desenvolver um sistema que auxilie pessoas com deficiência visual no descarte correto de itens em lixeiras de coleta seletiva através de comando de voz e de orientação por *LEDs* e avisos sonoros.

### 2.2. Objetivos Específicos

- Utilizar o Sistema *Arduino* (*hardware* e *software*) para elaboração do trabalho;
- Identificar a presença de um usuário diante da lixeira por meio de um sensor de movimento;
- Executar um arquivo de som que pergunta ao usuário o item que será descartado (Papel, Plástico, Vidro ou Metal);
- Utilizar placa de reconhecimento de comando de voz, que detecta a resposta do usuário informando o item a ser descartado;
- Auxiliar pessoas com comprometimento parcial da visão a identificar o recipiente correto para descarte de itens por meio de *LEDs* indicativos e sinais sonoros;
- Auxiliar crianças não alfabetizadas a identificar corretamente o recipiente correto para descarte de itens em lixeiras de coleta seletiva;
- Modelar um sistema de leitura de códigos de barra de produtos industrializados para identificação e auxílio para descarte correto de itens (Papel, Plástico, Vidro ou Metal).

### 3. REFERENCIAL TEÓRICO

Conceitos teóricos e importantes relacionados ao tema do presente trabalho, tais como: Placa *Arduino Uno V.3*, Linguagem *Wiring*, Sensor de Movimento PIR, *Buzzer* (dispositivo de sinalização de Áudio), Placa de Comando de Voz e Placa ISD1820 Gravador de voz e player.

#### 3.1. Sistema *Arduino*

O projeto *Arduino* deu-se início na Cidade de Ivrea, na Itália, em 2005 através de seus idealizadores, os professores Massimo Banzi e David Cuartielles, esse último um engenheiro espanhol em visita à Itália. O projeto surgiu com o intuito de interagir em projetos escolares, já que existia uma dificuldade para o ensino de eletrônica e programação de computadores para pessoas que não eram da área. Além disso, havia a dificuldade em encontrar placas poderosas que fossem de baixo custo.

Muitos pesquisadores também têm utilizado o Sistema *Arduino* para fins educativos, auxiliando pessoas a adquirir conhecimentos de engenharia, eletrônica ou computação, para que elas possam desenvolver projetos utilizando essas placas sem grandes dificuldades. As principais diferenças do *Arduino* para outras placas de microcontroladores são: o baixo custo, integração com o sistema IDE, programável via USB, e grande suporte da comunidade on-line. O *hardware Arduino* tem como componente básico a entrada USB, a entrada de alimentação, o pino ICSP, pinos de entrada analógica, pinos digitais, pinos de alimentação, botão reset. O *software*

*Arduino*, ou seja, a IDE *Arduino*, é programada na linguagem *Wiring* que está disponível no Site do *Arduino*.

O sistema é composto por *hardware* e *software* programado na linguagem *Wiring* (Figura 2). O *Arduino* é uma plataforma de código aberto (*open-source*), de prototipagem eletrônica fundamentada na flexibilidade e na fácil usabilidade de *hardware* e de *software*. O *software* foi criado a partir da própria IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado), disponível no site do *Arduino*. Este ambiente facilita a escrita de códigos e o

envio de dados através da placa I/O (*Input-Output*). O *software* tem versões para diversos sistemas operacionais tais como: *Windows*, *Macintosh OSX* e *Linux*.

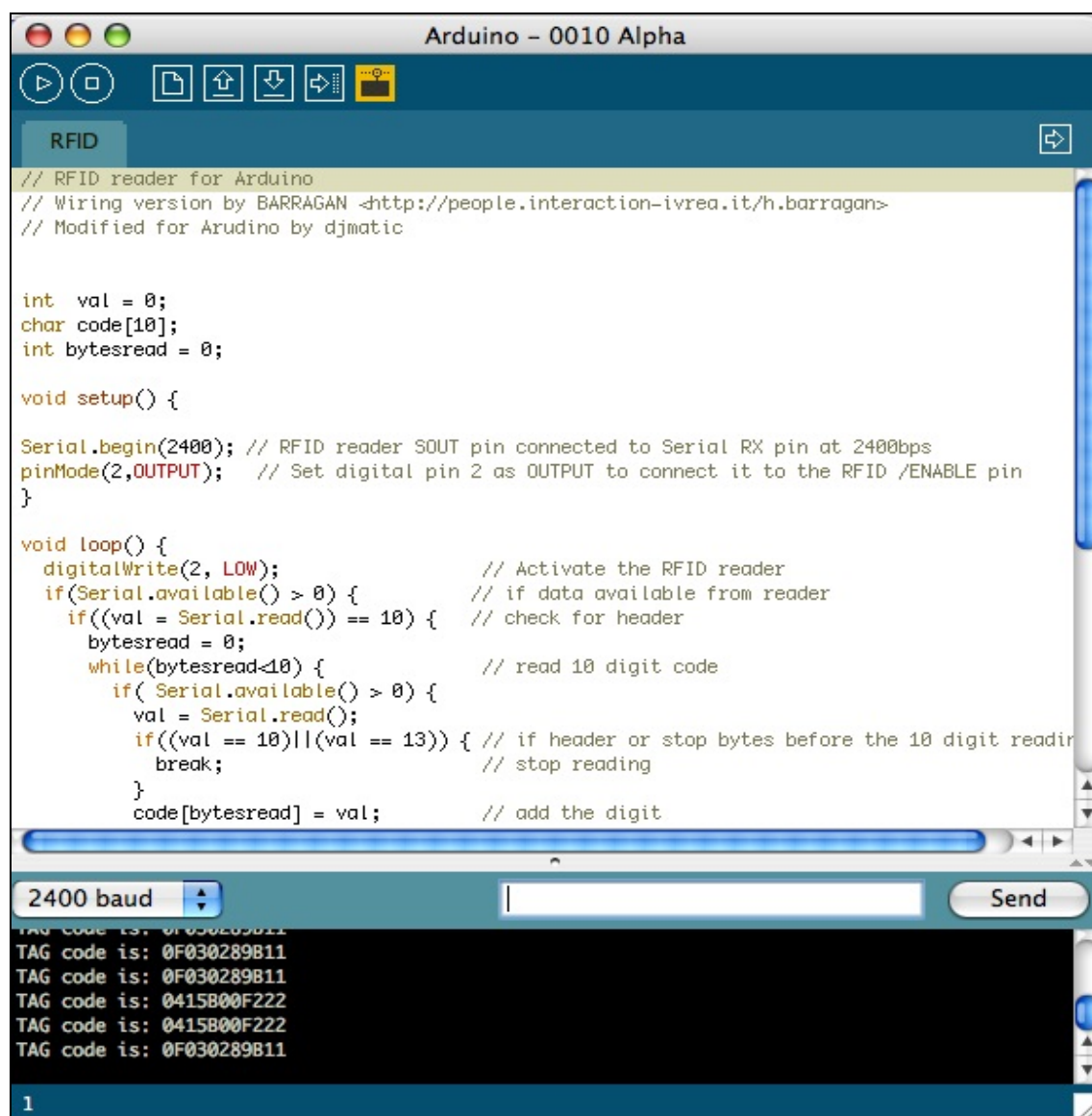


Figura 2 – Software Arduino

A Placa *Arduino UNO* possui as seguintes características: 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um de 16 MHz ressonador cerâmico, uma conexão USB, um conector de alimentação, um cabeçalho ICSP, e um botão de reset. Ele contém tudo o necessário para suportar o microcontrolador, bastando conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC para DC ou bateria para começar.

A Figura 3 apresenta a placa *Arduino Uno* que é responsável por controlar o sensor de movimento, *LEDs*, *Buzzer*, placa de mensagem de voz e a placa de comando de voz.

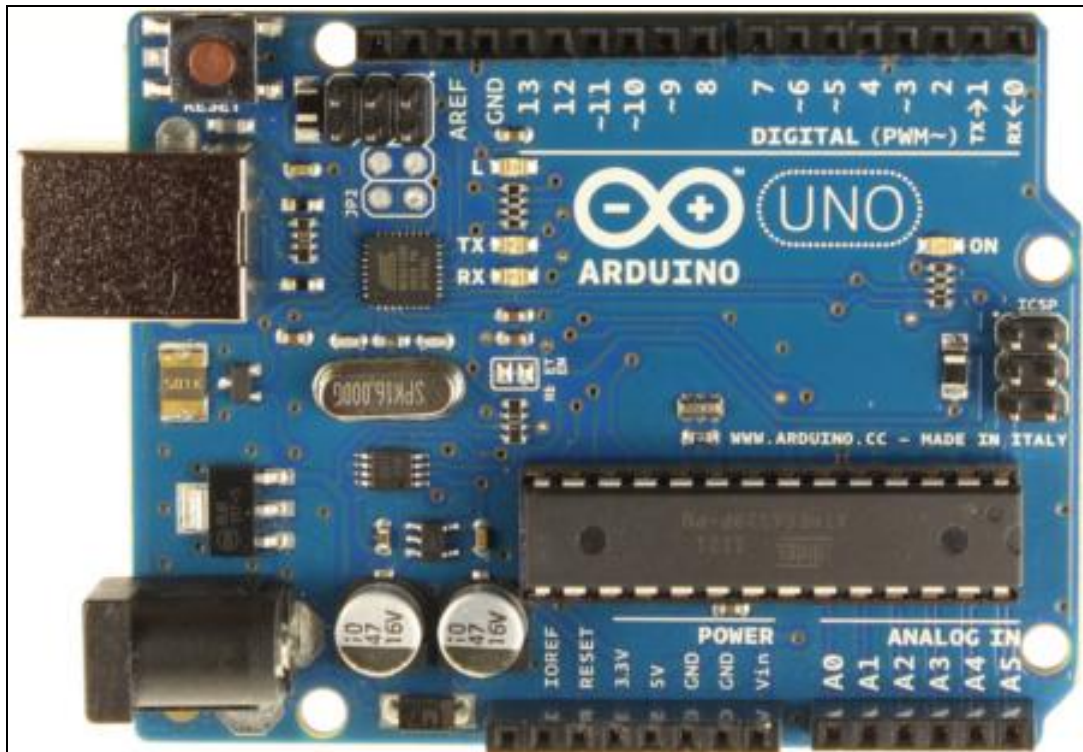


Figura 3 – Placa do Sistema *Arduino Uno*



### 3.2. Linguagem *Wiring*

A linguagem do *Arduino* é baseada em C e C++, possui código aberto (open-source) e biblioteca de nome *Wiring*. O software *Arduino* junto com o *Libc AVR*, (o pacote *Libc AVR* é um subconjunto da biblioteca C, padrão para microcontroladores *Atmel AVR 8-bit RISC*) dá liberdade para o usuário usar as suas funções. A placa *Arduino* é fundamentada nos microcontroladores *Atmel ATMEGA8* e *ATMEGA168*, essa permite ao usuário experiente criar seus próprios modelos de placa mãe *Arduino*.

### 3.3. Sensor de Movimento PIR (Passive Infra Red.)

Este dispositivo eletrônico mede infravermelho (IR) (Figura 4), que irradia de objetos em seu campo de visão, podendo detectar objetos em um ângulo menor de 100 graus numa proximidade de até 7 m do sensor. Sua tensão mínima é de 5VDC, podendo chegar a 12VDC, e mede aproximadamente 3,5 x 2,5 cm.

Seu funcionamento se dá no momento em que uma pessoa ou objeto entra na área monitorada, onde o sensor detecta uma mudança no infravermelho emitida a partir do corpo do invasor.



Figura 4 – Sensor de Movimento PIR

### 3.4. *Buzzer* – Dispositivo de Sinalização de Áudio

*Buzzer* (Figura 5) é um dispositivo de sinalização de áudio, onde sua função específica é emitir som. Seu acionamento e temporização serão determinados em nosso projeto conforme uma linha de programação específica.



Figura 5 – Dispositivo de Sinalização de Áudio (*Buzzer*)

A Figura 6 apresenta o esquema de instalação do dispositivo de sinalização de áudio em placa *Arduino Uno*.

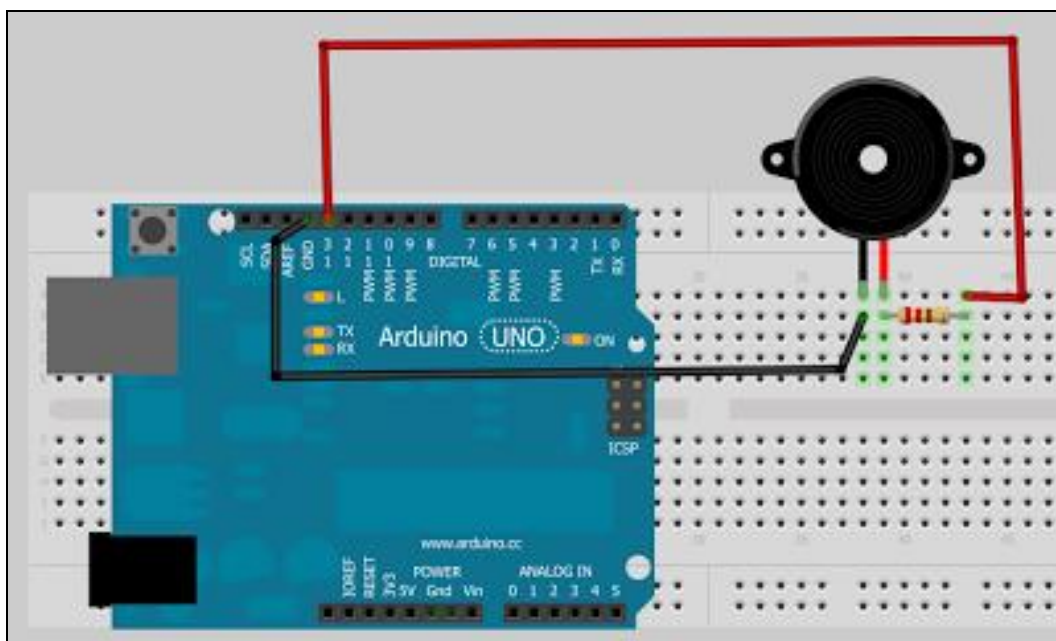


Figura 6 – Esquema de instalação do *Buzzer*

### 3.5. Módulo de Comando de Voz

A interface de Comando de Voz (Figura 7) permite comandar equipamentos, dispositivos, aparelhos eletrônicos através da voz. Permite gravar até 15 comandos de voz em 3 grupos, sendo 5 comandos em cada grupo. Sua configuração é feita via *UART RS232* nível *TTL*, podendo ser conectado diretamente ao canal de comunicação serial do computador. O módulo de comando de voz tem como características básicas: Tensão 4.5-5.5V, corrente de trabalho: <40mA, interface digital *UART* nível *TTL* 5V, Interface analógica: conector para microfone mono canal 3.5mm e dimensões de 30mm x 47.5mm.

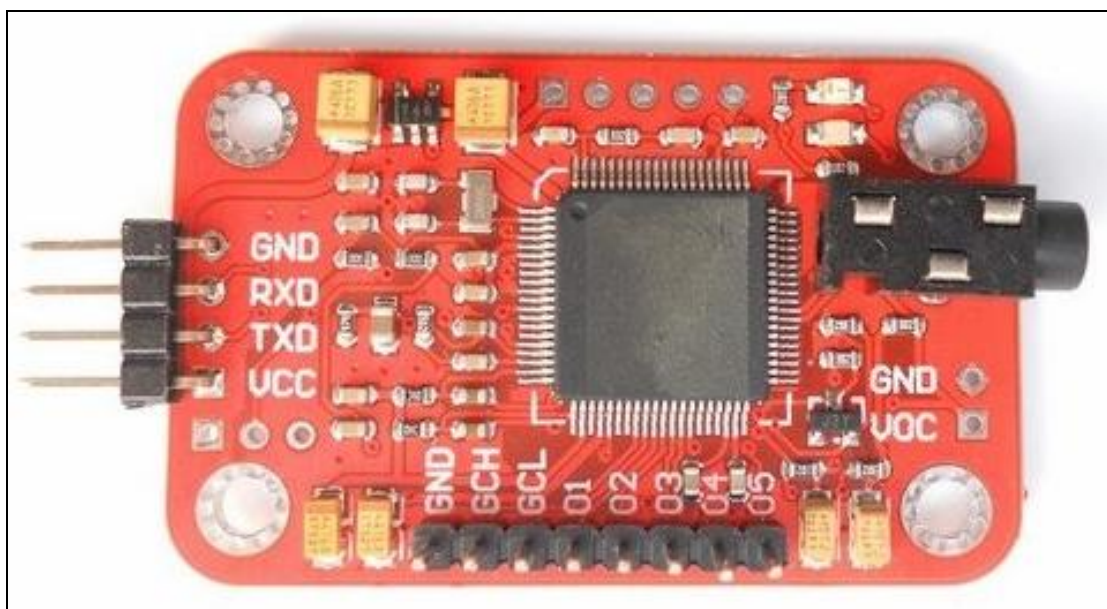


Figura 7 - Modulo de Comando de Voz

### 3.6. Gravador de Voz ISD1820

A interface de gravação de voz (Figura 8) tem como função principal fazer a interação Homem / Máquina. Esse módulo é baseado no chip ISD1820 que permite a gravação de voz e reprodução da mensagem armazenada em memória não volátil (mesmo sem energia a mensagem continuará salva). O módulo pode ser controlado por um microcontrolador *Arduino* ou diretamente pelos botões existentes na placa. Possui como características: chip principal: ISD1820, Tensão de operação: 3 ~5 VDC, Dimensões 38mm x 42,5mm. Pode ser ligado diretamente um auto falante de 8  $\Omega$  0.5W. Grava a voz diretamente pelo microfone embutido na placa com tempo máximo de gravação entre 8 a 20 segundos.

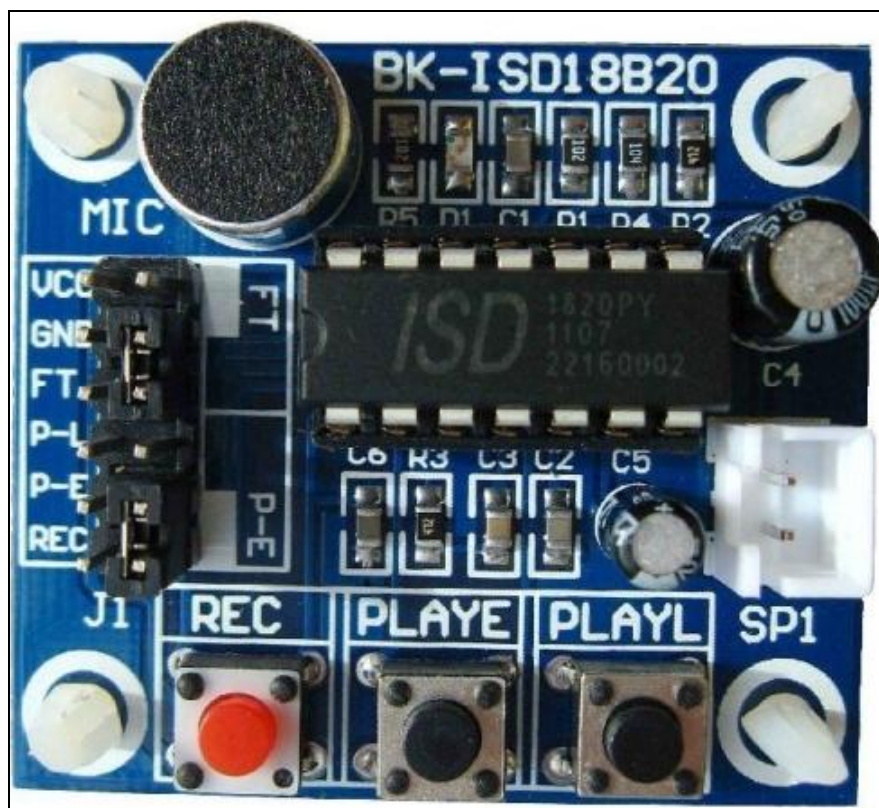


Figura 8 – Interface de gravação de voz ISD1820

## 4. REVISÃO DE LITERATURA

Este capítulo apresenta um levantamento da literatura sobre projetos que foram desenvolvidos como auxílio às pessoas com deficiência visual e que visam a sua inclusão na sociedade.

### 4.1. Projeto R3

Esta pesquisa foi realizada por SANTOS e col. (2011) na Faculdade de Zootécnica e Engenharia de Alimentos e teve como objetivo a partir da instalação de uma lixeira com sensores que produzirá efeitos sonoros ao depositar o material irá conscientizar os alunos, professores e funcionários do Campus USP de Pirassununga a separar e coletar adequadamente o lixo reciclável.

Os pesquisadores mostraram que a lixeira coletora possui grandes dimensões para causar um impacto visual logo no início de sua instalação (Figura 9). A portinhola localizada na parte superior frontal possui um sensor que ao sentir o movimento gerado pela deposição do material descartado dentro da lixeira, emitirá um sinal à CPU que se encontra na parte inferior, ao receber este sinal a CPU gerará o barulho de aplausos para o usuário da lixeira.

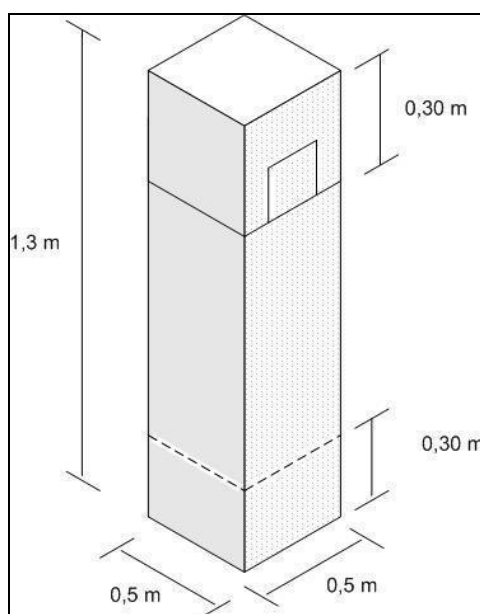


Figura 9 - Esquema de Lixeira Coletora de material reciclável



No entanto, algumas limitações surgiram, para o desenvolvimento do Projeto R3 é necessário um período inicial de 2 meses de planejamento estratégico para determinar as diretrizes a serem seguidas, como todos os integrantes do grupo são estudantes, há pouco tempo livre em comum entre os mesmos, dificultando a reunião do grupo, os funcionários responsáveis pela limpeza podem encontrar dificuldade para manipular e remover os sacos de lixo da nova lixeira e uma ameaça externa a dificuldade em se montar o sistema sonoro.

#### 4.2. AVEI – Ambiente Virtual de Ensino Inteligente

Esta pesquisa foi realizada por NUNES e col.(2001) na Universidade Regional Integrada - RS e teve como objetivo a implementação de um *software* educacional inteligente chamado AVEI, que utiliza tecnologias para o desenvolvimento de um *software* educacional adaptável ao aluno, dentre estas tecnologias é destacado Sistemas Multiagentes, Realidade Virtual e Educação à Distância (Figura 10).



Figura 10 - Tela Principal do AVEI

AVEI usa técnicas de Realidade Virtual para simular o mundo real das crianças, modela um *software* educacional destinado a preparar crianças na faixa etária de 8 a 10 anos para a conscientização com relação à coleta e seleção do lixo urbano. A ação do aluno acontece em dois ambientes diferenciados: ambiente externo e interno. Tanto no ambiente externo como no interno o *aluno* terá disponível, lixeiras especiais para a seleção adequada do lixo encontrado.

Os pesquisadores mostraram que AVEI contribui para a área de inteligência artificial na educação, os modelos de Agentes Reativos neste trabalho trazem inovações tecnológicas para implementação de *softwares* educacionais inteligentes, transformando em *softwares* mais adaptáveis ao aluno.

No entanto, os agentes reativos têm evidenciado suas limitações, além de não serem muito versáteis, eles têm problemas para obter conhecimento que dependa de raciocínio, pois o raciocínio envolve elementos que não estão ligados à percepção. Para trabalho futuro consiste em desenvolver uma ferramenta de autoria no contexto de *software* educacional para implementar micro mundos virtuais baseados nos modelos de agentes reativos utilizados no AVEI.

#### **4.3. The *GuideCane* – Tecnologia de Assistência a Deficientes Visuais**

Esta pesquisa foi realizada por ULRICH e col. (2001) na Universidade de Michigan e teve como objetivo desenvolver um dispositivo projetado para ajudar usuários cegos ou deficientes visuais a navegar com segurança e rapidez entre os obstáculos e outros perigos. O usuário empurra o *GuideCane* para frente, quando os sensores ultra sônicos da *GuideCane* detecta um obstáculo, o computador embutido determina uma direção adequada (Figura 11).

Os pesquisadores mostraram que os resultados dos testes é que todos os indivíduos que participaram do teste só precisou de alguns minutos de treinamento para atravessar ambientes desordenados na velocidade de caminhada de até 1 m/s. Além disso, cegos e alguns com olhos vendados observou que andar com o *GuideCane* era muito intuitiva e pouco esforço exigido consciente.

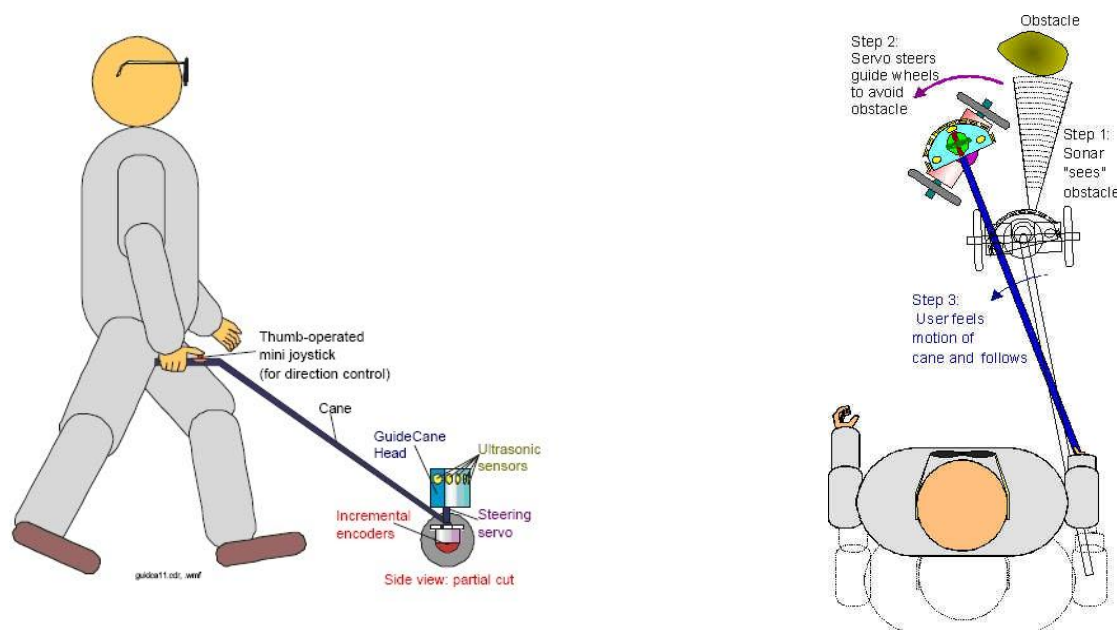


Figura 11 - Componentes Funcionais do *GuideCane*

No entanto, algumas limitações surgiram como falhas do sistema de desvio de obstáculos causadas por obstáculos que não foram detectados pelos sonares. O *GuideCane* não é capaz de detectar obstáculos como mesas saliente, ao ar livre não possui a capacidade de detectar bordas das calçadas.

Eles pretendem ainda dar continuidade no projeto através de melhorias nos sonares, freios e configuração das rodas. Outra melhoria seria na adição de saída de voz, não só para guiar o usuário para um local, mas também fornecer informações sobre o ambiente.

#### 4.4. Bengala de apoio a cegos com detecção de buracos

Esta pesquisa foi realizada por ROSA (2009) na Universidade de Aveiro-Portugal e teve como objetivo de desenvolver uma bengala de apoio a cegos que tivesse a capacidade de detectar buracos e desníveis no solo por meio de ultrassom (Figura 12). Foram estudados alguns sistemas de apoio a cegos (tanto comerciais como em desenvolvimento) e entrevistados alguns deficientes visuais de forma a compreender melhor quais as funcionalidades que podiam e deviam ser incluídas na bengala.



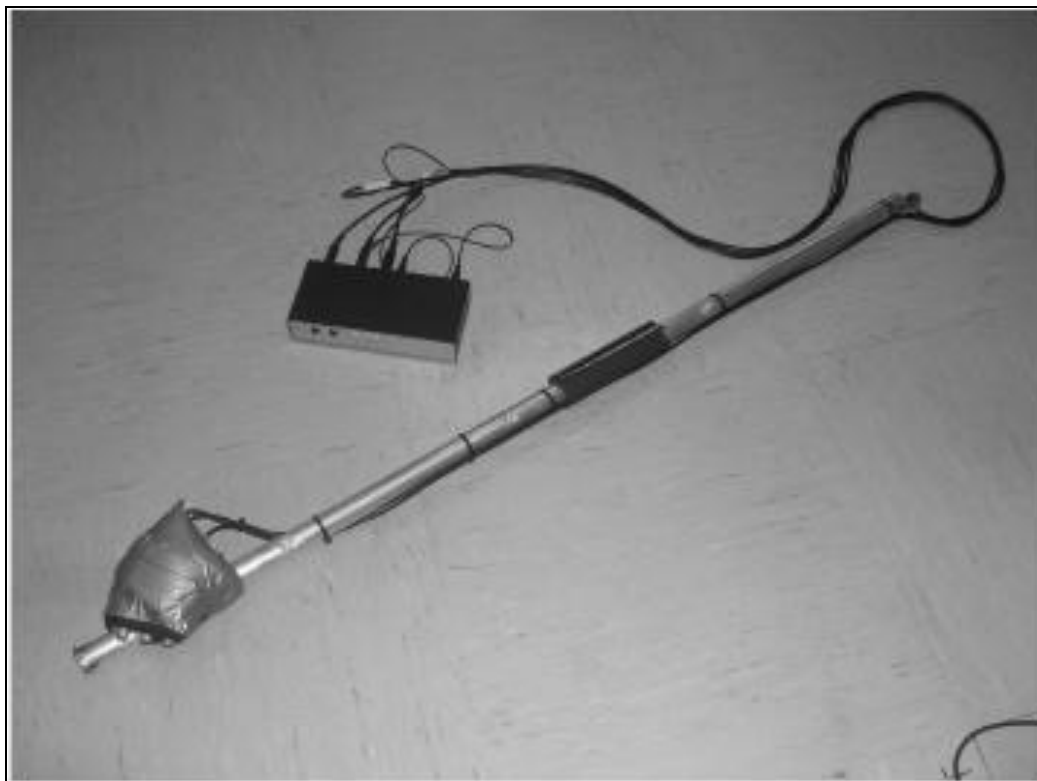


Figura 12 - Protótipo de bengala de apoio a cegos - 2008

Tendo em conta que o desenvolvimento desta tecnologia já tinha sido iniciado anteriormente no âmbito de outro projeto e deveria continuar no futuro, os pesquisadores desenvolveram uma plataforma de testes (Figura 13) para medir e registrar o atraso do sistema na detecção de desníveis.



Figura 13 - Plataforma de testes desenvolvida

O núcleo do trabalho consistiu em desenvolver um algoritmo capaz de a presença de desníveis de forma rápida e confiável. Os ensaios realizados com a plataforma de testes apresentaram tempos de atraso reduzidos, viabilizando a implementação do algoritmo na bengala.

Foi construído um protótipo da bengala para possibilitar o teste do algoritmo desenvolvido em ambientes reais. Foram feitos testes com alguns cegos (Figura 14) que experimentaram o protótipo e mostraram grande entusiasmo, pois todos os buracos e desníveis que apareceram no seu caminho foram detectados antecipadamente pela bengala. O equipamento mostrou-se muito eficaz em escadas, detectando todos os degraus individualmente, informando claramente o utilizador do começo e fim das mesmas. Durante o decorrer dos testes com cegos, apenas uma vez o equipamento demonstrou defeito, mas a impressão que a bengala causou aos cegos foi muito positiva.

A Figura 14 mostra os testes realizados com pessoas com deficiência utilizando a bengala em locais com diferentes obstáculos.



Figura 14 - Testes de campo com pessoas com deficiência

Apesar dos bons resultados obtidos, ainda ficaram alguns problemas por resolver. São então deixadas algumas sugestões para trabalho futuro, para que um dia esta tecnologia possa chegar às lojas e ajudar os deficientes visuais.

#### 4.5. Orientação de Cegos usando Visão de Computador

Esta pesquisa foi realizada por FERNANDES (2010) na Universidade De Trás-Os-Montes e Alto Douro de Portugal com o objetivo de desenvolver um sistema baseado em Visão por Computador que aumente a mobilidade de pessoas cegas, ou portadoras de deficiência visual, através do reconhecimento visual de características específicas. O sistema fornecerá constantemente informações que possibilitem a correção da trajetória do utilizador, orientando-o no meio envolvente de forma segura evitando zonas potencialmente perigosas como estradas, passagens de nível, lagos, etc.

Os pesquisadores mostraram, em primeiro lugar, resultados com testes de desempenho onde foram colocadas várias marcas circulares (Figura 15) no pavimento formando uma linha imaginária, e foi contabilizado o tempo que o sistema necessitou para processar cada frame. Foi também verificado se em cada frame a marca circular era, ou não detectada.



Figura 15 - Imagem com exemplo de marcas circulares

Um utilizador com visão natural fez o percurso de modo a que pelo menos uma marca estivesse sempre presente na imagem. Deste modo foi possível verificar se o sistema consegue reconhecer as marcas, que se garantem estarem sempre presentes em todos os frames capturados. Como resultado observou-se que foram “perdidos” muitos frames, ou seja, houve muitas falhas na detecção dos círculos. Outra observação é a demora a processar cada frame.

Um esquema explicativo dos componentes do sistema pode ser visto nas Figuras 16 e 17.



Figura 16 - Sistema sendo testado por um utilizador

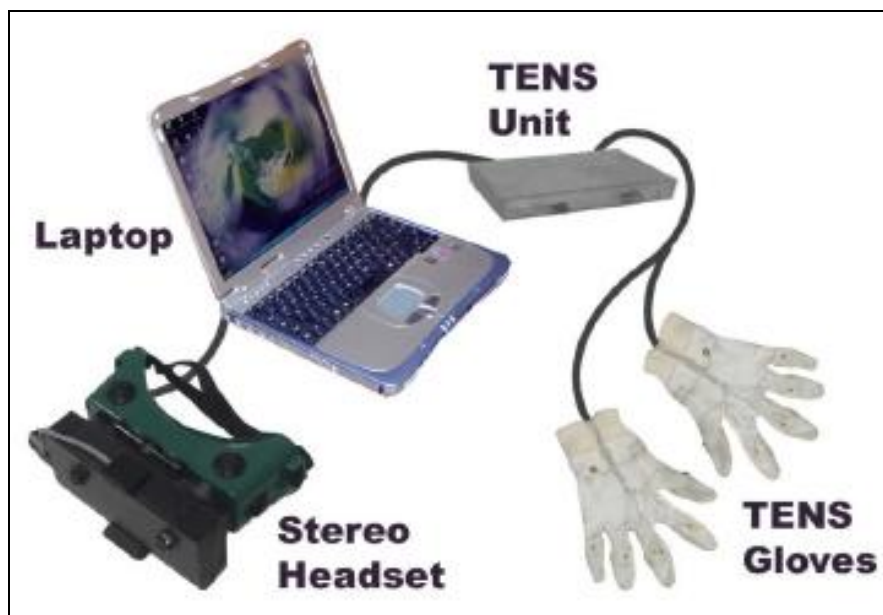


Figura 17 - Esquema explicativo do sistema

Foi utilizado como ferramenta para desenvolvimento desta aplicação a Biblioteca de *software ARToolkit*, que foi usada para criar aplicações de realidade aumentada em tempo real e pelo fato da mesma ser código open source, o que trouxe também grande importância neste tipo de implementação uma vez que é possível analisar todo o código-fonte e detectar a origem de erros, durante o desenvolvimento.

Como trabalho futuro poderão ser feitas novas implementações e funcionalidades, uma vez que o projeto ainda encontra-se em desenvolvimento.

#### **4.6. Auxílio à Mobilidade Deficientes Visuais - Eletro Estimulação Tátil**

Esta pesquisa foi realizada por PEREIRA e col. (2004) na Politécnica da USP, onde descreve um sistema desenvolvido para o auxílio de deficientes visuais em um problema prático de sua vida diária, de familiarização de um ambiente desconhecido, que pelas técnicas habituais de orientação e mobilidade é muito demora e nem sempre é possível de ser completado.

Foi proposta a Substituição da Visão Sensorial (SS) pelo tato (Figura 18), fazendo com que as imagens fossem capturadas, tratadas digitalmente para detectar seus contornos e reduzir resolução, e posteriormente passadas ao usuário



através de eletro estimulação cutâneo no abdômen. Foram desenvolvidos os equipamentos, o *software* para tratar a imagem capturada num PC e uma matriz de eletrodos flexível.

Entre os problemas decorrentes da deficiência visual, destaca-se a locomoção, incluindo a dificuldade de montar uma representação mental de ambientes desconhecidos. Foram tratados como primeiros problemas as pessoas portadoras de deficiência (PPDs) visual no ensino de técnicas de orientação e mobilidade (OM) que consistem basicamente em algoritmos de busca exaustiva do ambiente.

Foram desenvolvidos testes em um sistema que busca passa informações visuais através de eletro estimulação táteis na região do abdômen, dando seqüência a estudos preliminares iniciados em 1993, agregando técnicas adicionais para focar na solução de problemas específicos.

Como trabalho futuro o equipamento deverá contar com um programa para encontrar cartazes e sinais de imagem capturada e estimular diretamente na pele do usuário as informações neles contidas. A também a intenção de diminuir o tamanho do o sistema, tornando-o mais portátil e com maior autonomia de bateria.

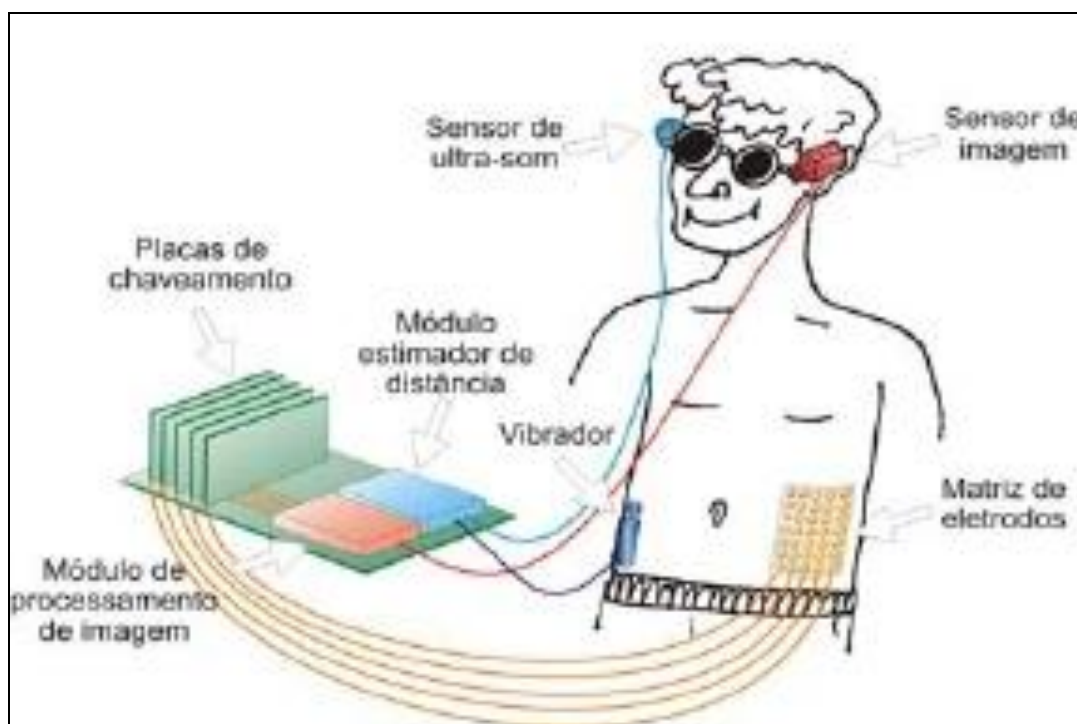


Figura 18 - Visão geral do sistema

## 5. MATERIAIS E MÉTODOS

Este capítulo apresenta uma descrição dos materiais e métodos utilizados para desenvolvimento do sistema proposto.

### 5.1. Sensor de Movimento PIR

A Figura 19 ilustra o funcionamento do sensor de movimento *PIR*, onde a sua participação é inicial em todo o sistema já que é através de seu acionamento que todo o processo se inicia. A partir da detecção da presença de uma pessoa o sensor envia um comando acionando a placa de mensagem de voz onde serão dadas as instruções para que se escolha do tipo de produto para ser descartado. Conforme a escolha será acionada o sinal sonoro e o *LED* correspondente à cor da lixeira para o descarte do produto.

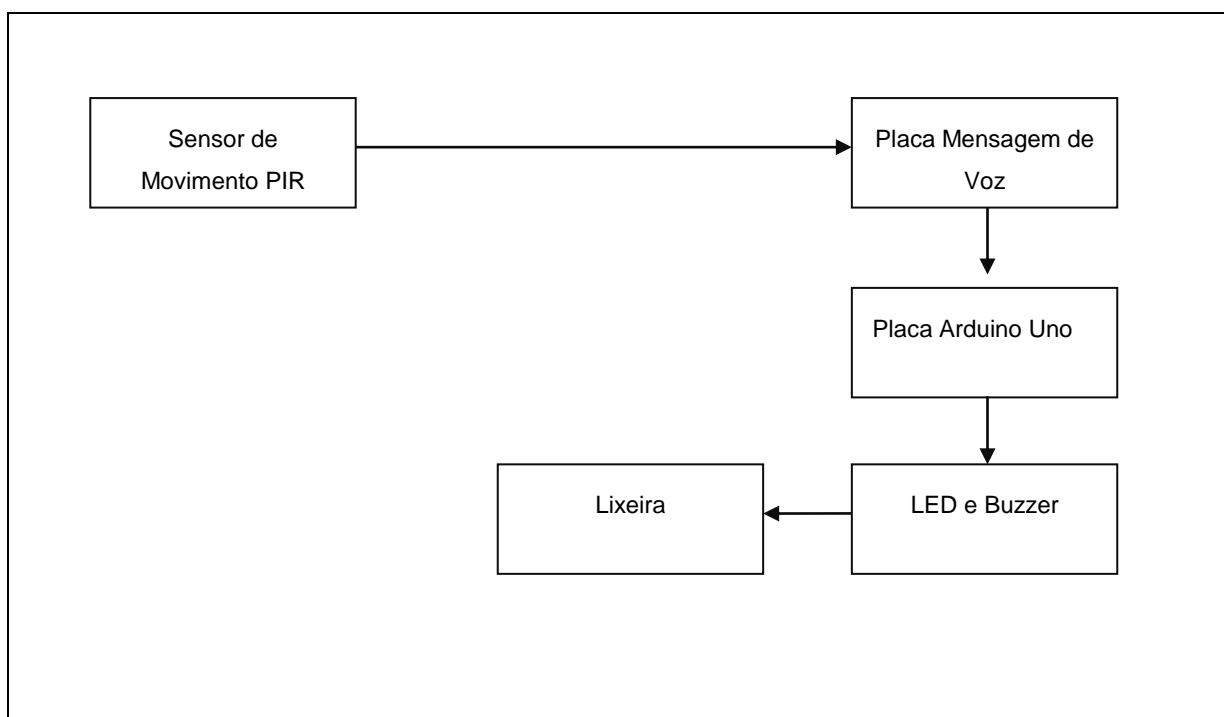


Figura 19 – Fluxograma de Funcionamento do Sensor *PIR*

A Figura 20 apresenta o fluxograma da seqüência lógica do sistema desde o seu início ao fim do processo. O sistema captura a presença de uma pessoa e aciona a placa de mensagem de voz, que por esta é dada a opção de acesso ao sistema onde deverá ser dito o tipo de produto para o descarte. Conforme a escolha de descarte será acionada a *LED* juntamente com um sinal sonoro correspondente à cor da lixeira para o descarte do produto.

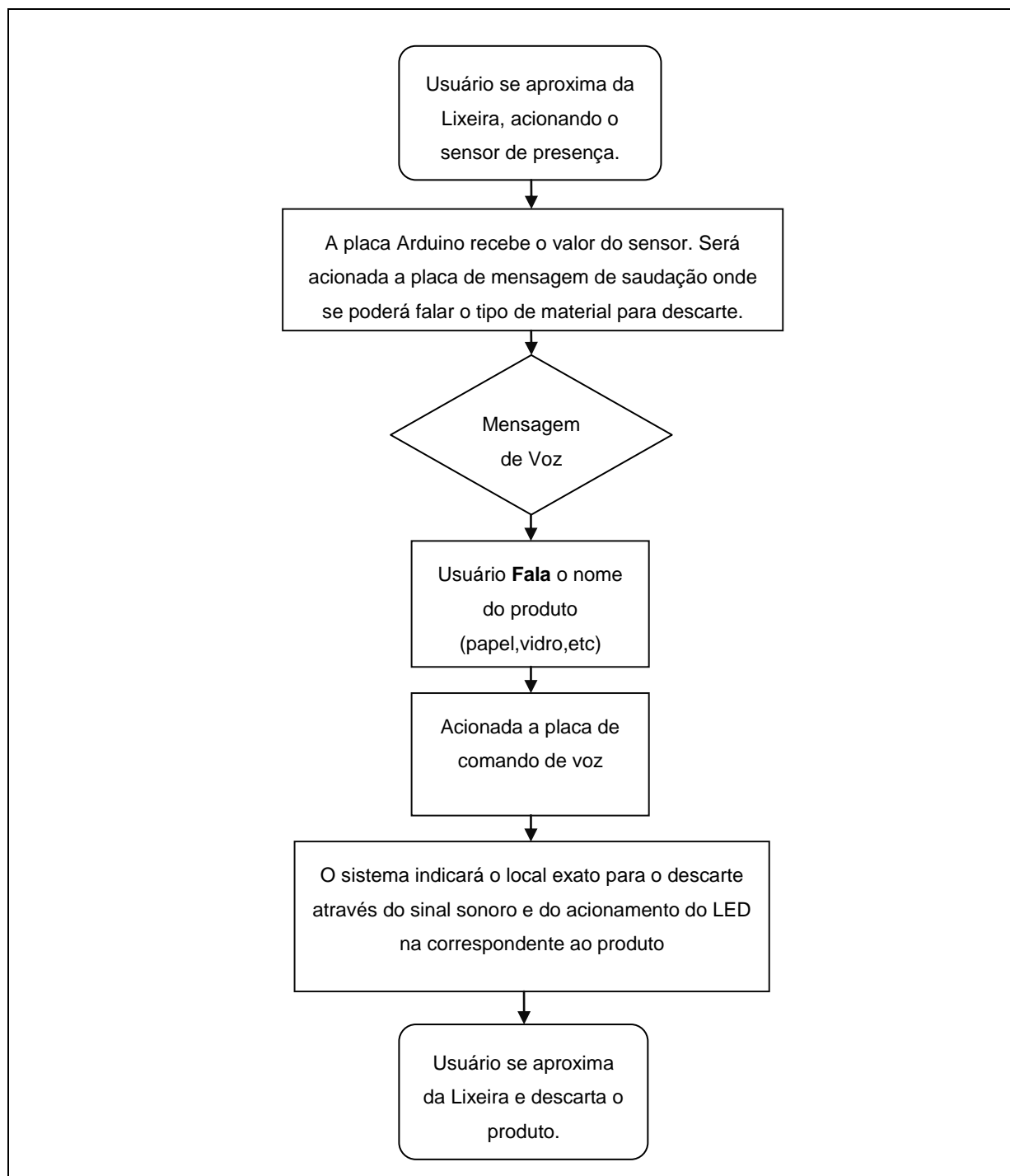


Figura 20 – Fluxograma de Funcionamento do Sistema Geral



## 5.2. Modelo do Sistema

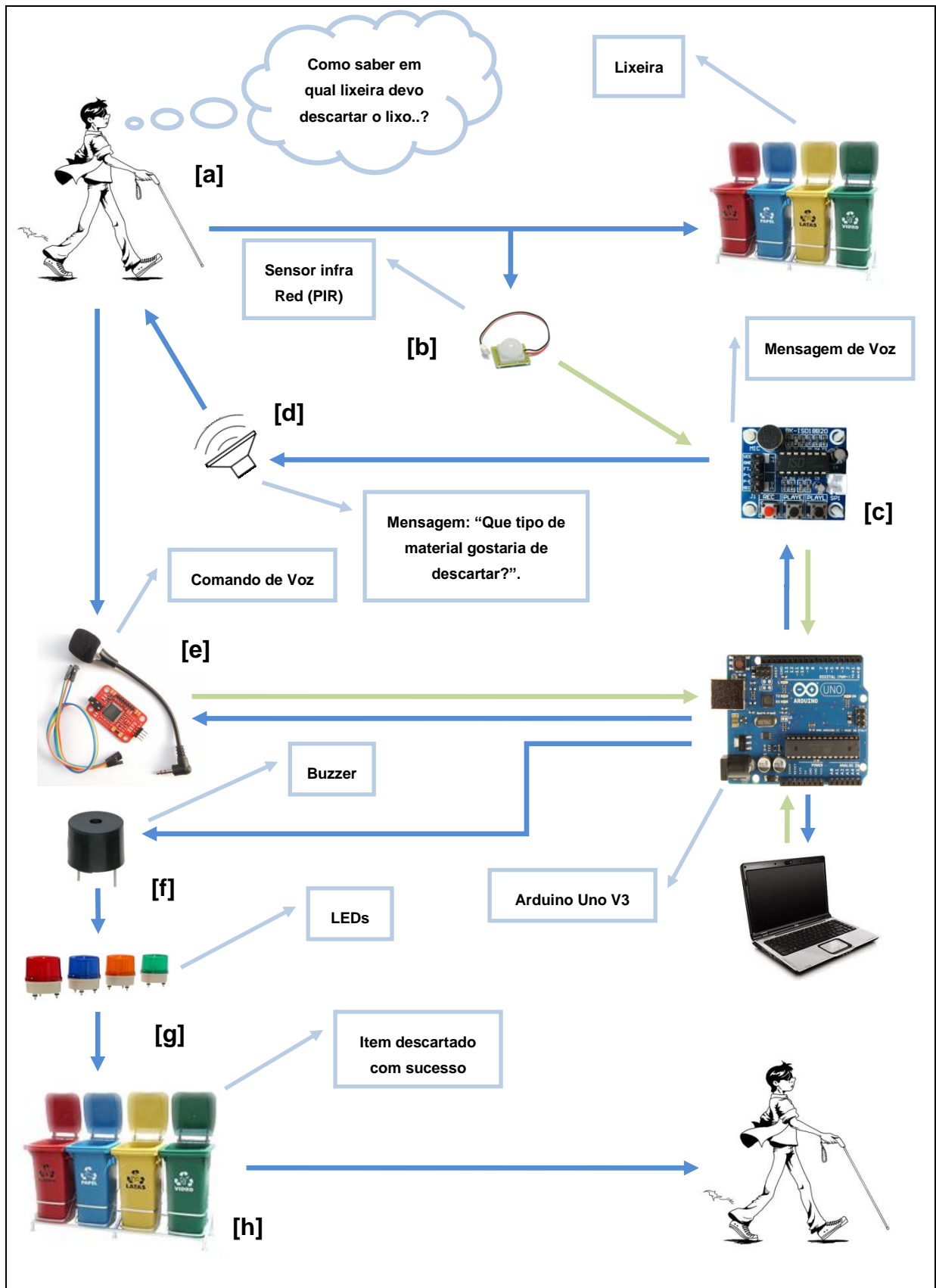


Figura 21 – Esquema de Funcionamento do Sistema Geral

### 5.3 Procedimentos de Desenvolvimento do Sistema Proposto

Ao aproximar-se da lixeira, o deficiente visual (Figura 21 [a]) será capitado por um sensor infravermelho (Figura 21 [b]) que ativará a placa de mensagem de voz (Figura 21 [c]) criando a interatividade homem/máquina.

Será acionada uma mensagem de saudação (Figura 21 [d]) onde serão dadas as instruções para início de todo o processo. Quando o DV começar a falar o tipo de objeto que tenha em mãos, ou seja, papel (lixeira cor azul), vidro (lixeira cor verde), lata (lixeira cor amarela) ou plástico (lixeira cor vermelho), será então acionada a placa de Reconhecimento de Voz (Figura 21 [e]).

Quando o usuário quiser jogar uma lata de refrigerante na lixeira, ele apenas dirá “Lata”, então a lixeira que corresponde o material do tipo lata (lixeira cor amarela) acenderá uma *LED* (Figura 21 [g]) na cor amarela e emitirá um sinal sonoro através do *Buzzer* (Figura 21 [f]) durante quatro segundos indicando o local exato para o depósito do resíduo reciclável (Figura 21 [h]).

As lixeiras seguiram as cores padrão (Figura 22) determinadas pelo Conselho Nacional do Meio Ambiente - CONAMA.



Figura 22 – Cores Padrão das Lixeiras

### 5.4 Estimativa de Custo

A seguir são apresentados alguns valores referentes os materiais que foram utilizados para o desenvolvimento do sistema (Tabela 1).

Tabela 1 - Custo do Sistema

ITENS	VALOR	VALOR
	R\$ 70,00*	US\$21,95**
	R\$ 60,00*	US\$ 19,98**
	R\$ 140,00*	US\$32,98**
	R\$ 19,98*	US\$ 6,95**
	R\$ 2,50*	- -
	R\$ 11,00*	- -
	R\$ 35,00*	- -
	R\$ 1,00*	US\$ 0,30**

\*Valores referentes a Lojas on-line no Brasil - Laboratório de Garagem / Mercado Livre

\*\* Valores referentes a Loja Amazon.com / Ebay.com

## 5.5 Desenvolvimento do Projeto

A seguir são apresentadas 4 Fases para desenvolvimento do sistema.

### 5.5.1 Fase I – Conectar o Sensor de *Infra Red*.

A primeira fase do projeto consiste em utilizar o sensor de presença *infra red* conectado à placa de comando *Arduino UNO*

O dispositivo poderá detectar objetos em um ângulo menor de 100 graus numa proximidade de 1,5 mts do sensor, e assim poderá captar a presença de pessoas que se aproximem das lixeiras. A partir disso, o sistema irá acionar a placa de mensagem de voz, onde em seguida haverá uma interação entre Homem / Máquina, pois a placa perguntará à pessoa qual objeto ela deseja descartar

A Figura 23 mostra o Sensor de Presença *Infra Red* (PIR) conectado à placa *Arduino UNO*. O sensor *PIR* tem três fios: *GND* (marrom), *VCC* (vermelho) e Saída (preto). Para que o sensor *PIR* funcione corretamente foi necessário um resistor de *10Kohm* como *Pull-Up*.

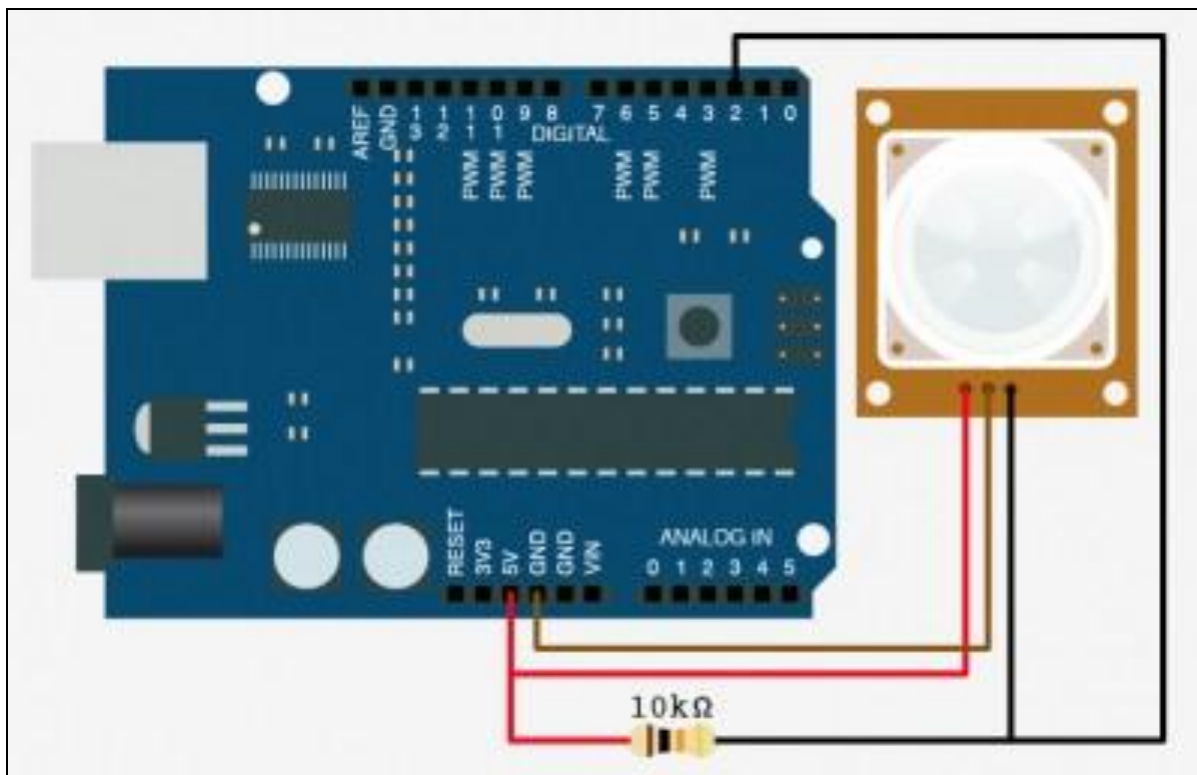
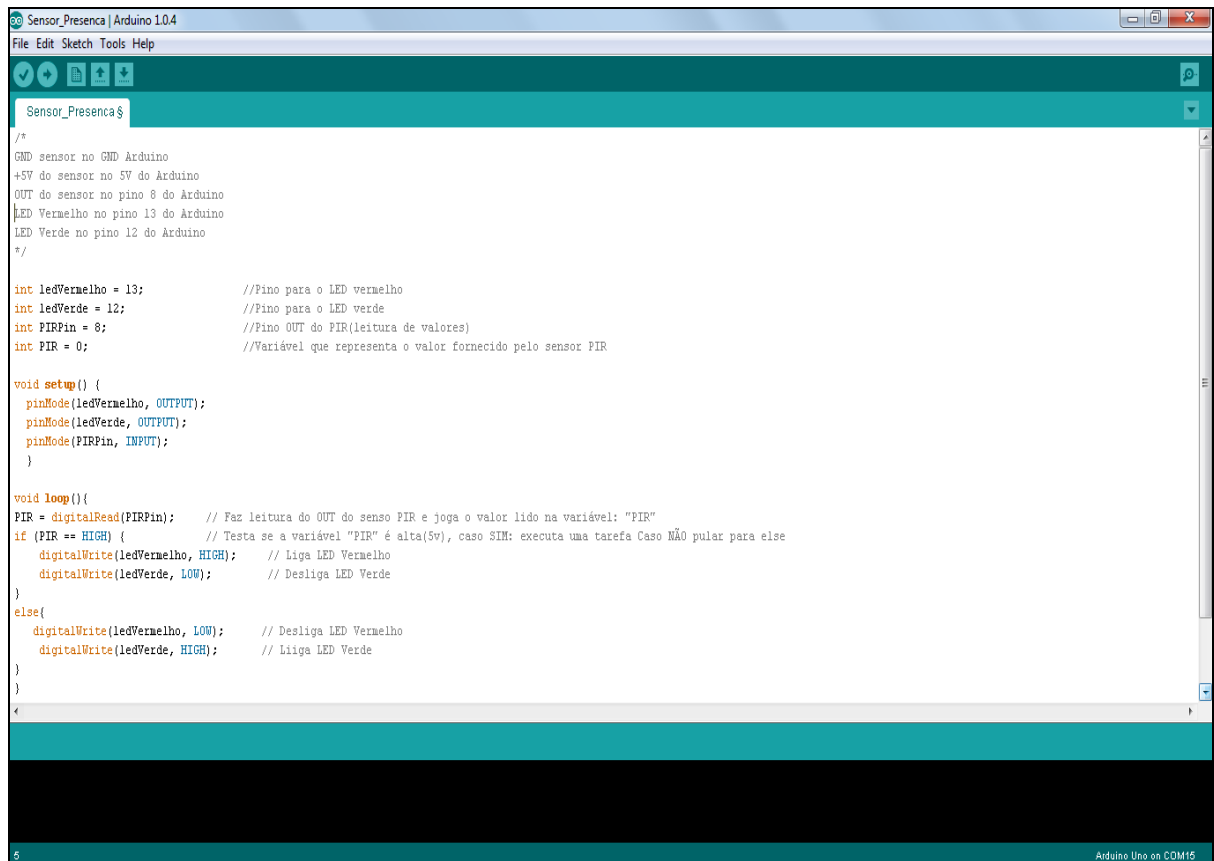


Figura 23 – Esquema de conexão do Sensor de Presença *PIR*

A Figura 24 mostra tela do programa *Arduino* 1.0.4 com os comandos para programação do Sensor de Presença *PIR*.



```
/*
  GND sensor no GND Arduino
  +5V do sensor no 5V do Arduino
  OUT do sensor no pino 8 do Arduino
  LED Vermelho no pino 13 do Arduino
  LED Verde no pino 12 do Arduino
*/

int ledVermelho = 13;      //Pino para o LED vermelho
int ledVerde = 12;         //Pino para o LED verde
int PIRPin = 8;            //Pino OUT do PIR(leitura de valores)
int PIR = 0;              //Variável que representa o valor fornecido pelo sensor PIR

void setup() {
  pinMode(ledVermelho, OUTPUT);
  pinMode(ledVerde, OUTPUT);
  pinMode(PIRPin, INPUT);
}

void loop(){
  PIR = digitalRead(PIRPin); // Faz leitura do OUT do senso PIR e joga o valor lido na variável: "PIR"
  if (PIR == HIGH) {        // Testa se a variável "PIR" é alta(5v), caso SIM: executa uma tarefa Caso NÃO pular para else
    digitalWrite(ledVermelho, HIGH); // Liga LED Vermelho
    digitalWrite(ledVerde, LOW);      // Desliga LED Verde
  }
  else{
    digitalWrite(ledVermelho, LOW);   // Desliga LED Vermelho
    digitalWrite(ledVerde, HIGH);     // Liiga LED Verde
  }
}
```

Figura 24 – Tela do *Arduino* com o programa do Sensor de Presença

A Figura 25 mostra o programa, juntamente com a sua explicação para ativação do Sensor de Presença *PIR*.

```

/*
GND sensor no GND Arduino
+5V do sensor no 5V do Arduino
OUT do sensor no pino 8 do Arduino
LED Vermelho no pino 13 do Arduino
LED Verde no pino 12 do Arduino
*/
int ledVermelho = 13;      //Pino para o LED vermelho
int ledVerde = 12;        //Pino para o LED verde
int PIRPin = 8;           //Pino OUT do PIR(leitura de valores)
int PIR = 0;              //Variável que representa o valor fornecido pelo sensor PIR
void setup() {
  pinMode(ledVermelho, OUTPUT);
  pinMode(ledVerde, OUTPUT);
  pinMode(PIRPin, INPUT);
}

void loop(){
  PIR = digitalRead(PIRPin); // Faz leitura do OUT do sensor PIR e joga o vlr lido na variável: "PIR"
  if (PIR == HIGH) {        // Testa se a variável "PIR" é alta(5v), SIM: executa- NÃO pular para else
    digitalWrite(ledVermelho, HIGH); // Liga LED Vermelho
    digitalWrite(ledVerde, LOW);      // Desliga LED Verde
  }
  else{
    digitalWrite(ledVermelho, LOW);   // Desliga LED Vermelho
    digitalWrite(ledVerde, HIGH);     // Liga LED Verde
  }
}
}

```

Figura 25 – Código do Programa do Sensor PIR

### 5.5.2 Fase II – Gravador de Voz ISD1820

A segunda fase do projeto consiste em conectar a placa de Gravador de Voz à placa *Arduino UNO*. A Figura 26 mostra a placa de gravação de voz conectada à placa *Arduino*. A função principal desse gravador de voz é estabelecer, juntamente com o sensor de presença, uma interação Homem / Máquina dando início assim a todo o processo de funcionamento do sistema.

Esse módulo é baseado no chip ISD1820 que permite a gravação de voz e reprodução da mensagem armazenada em memória não volátil (mesmo sem energia a mensagem continuará salva). O módulo será controlado por um microcontrolador *Arduino UNO* e sua gravação poderá ser feita diretamente pelos botões existentes na placa ou via microcomputador.

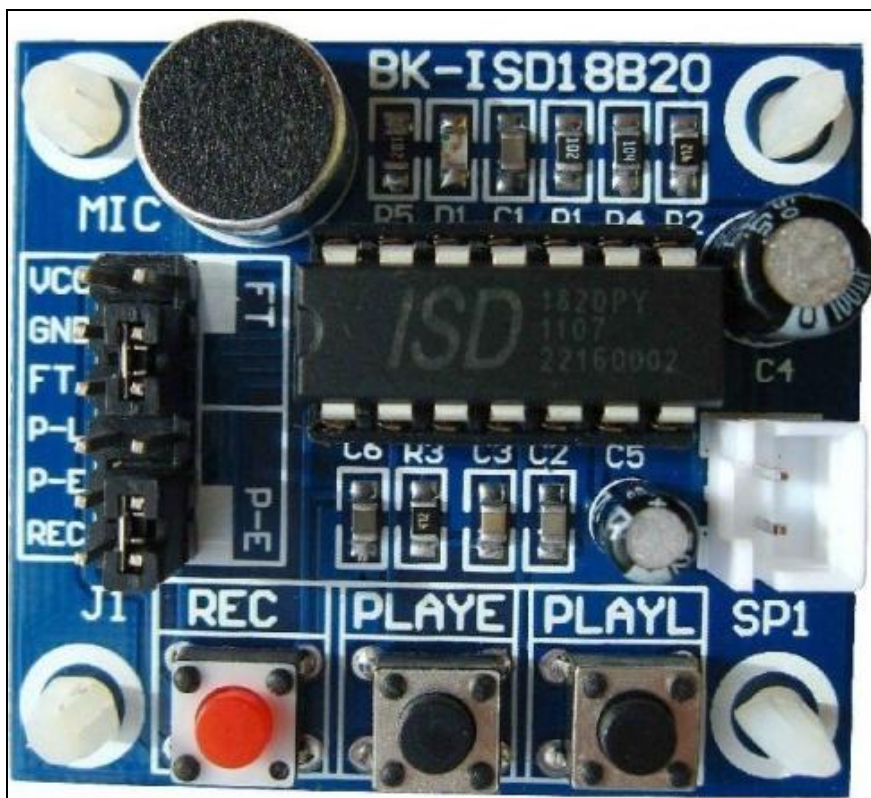


Figura 26 – Gravador de Voz ISD1820

Suas características principais são: chip principal ISD1820, Tensão de operação: 3 ~5 VDC, Dimensões 38mm x 42,5mm.



A placa de Gravação de Voz ISD1820 (Figura 27) será conectada na placa *Arduino Uno* pela seguinte sequência de pinos: o pino *VCC* ao 5V e o pino *GND* ao *GND*. A voz será gravada diretamente pelo microfone embutido na placa e o tempo máximo de mensagem gravada será de 20 segundos.

Para a reprodução da mensagem de voz gravada foi utilizado um auto falante de  $8\ \Omega$   $0.5W$  acoplado ao sistema.

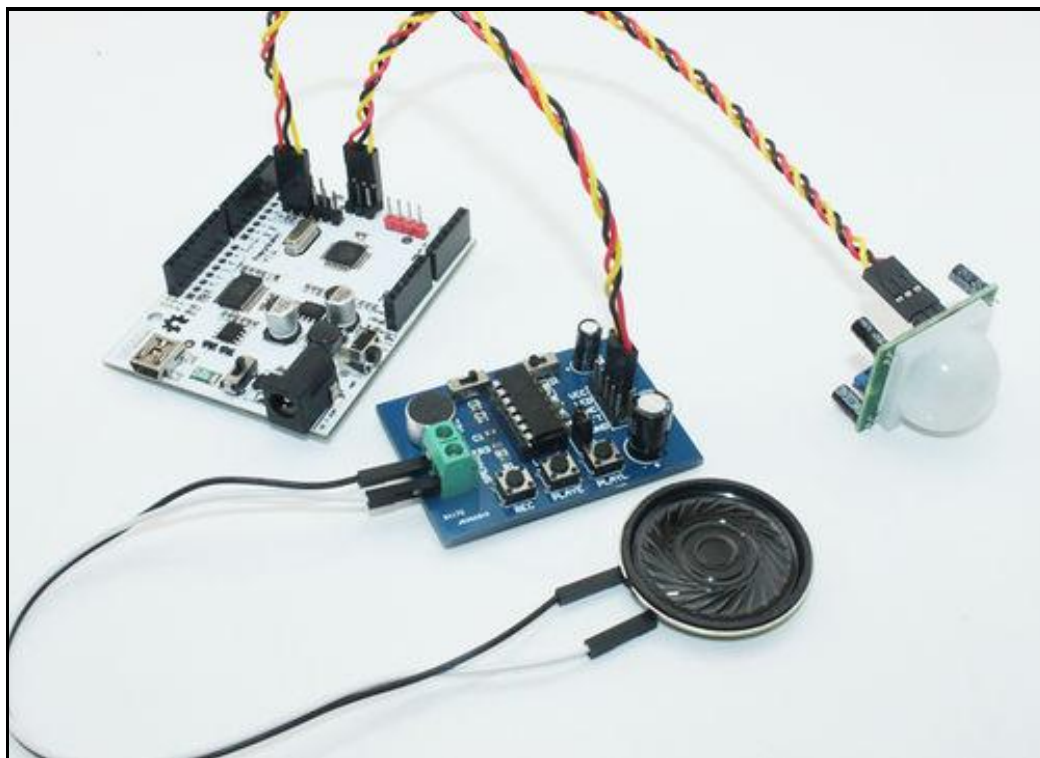


Figura 27 – Gravador de Voz conectado a Placa *Arduino*

Em seguida foi utilizado o microcomputador para fazer a gravação da mensagem de saudação utilizando a seguinte sequência de passos: Primeiro será colocado o pino REC em nível alto para executar a função. Depois será enviado um pulso de nível alto e logo após foi retornado para o nível baixo no pino *P-E (PLAYE)* para reproduzir toda a mensagem gravada. Colocou-se o pino *P-L (PLAYL)* em nível alto pelo tempo que se deseja reproduzir a mensagem e depois volta para o nível baixo para interromper a execução.

O módulo ISD1820 vai reproduzir uma mensagem informando quais os tipos de materiais recicláveis que podem ser descartados naquele local, então após o final da mensagem o DV deverá falar qual o tipo de material ele deseja descartar.



Os códigos nas Figuras 28 e 29 apresentam a seguinte lógica desse processo.

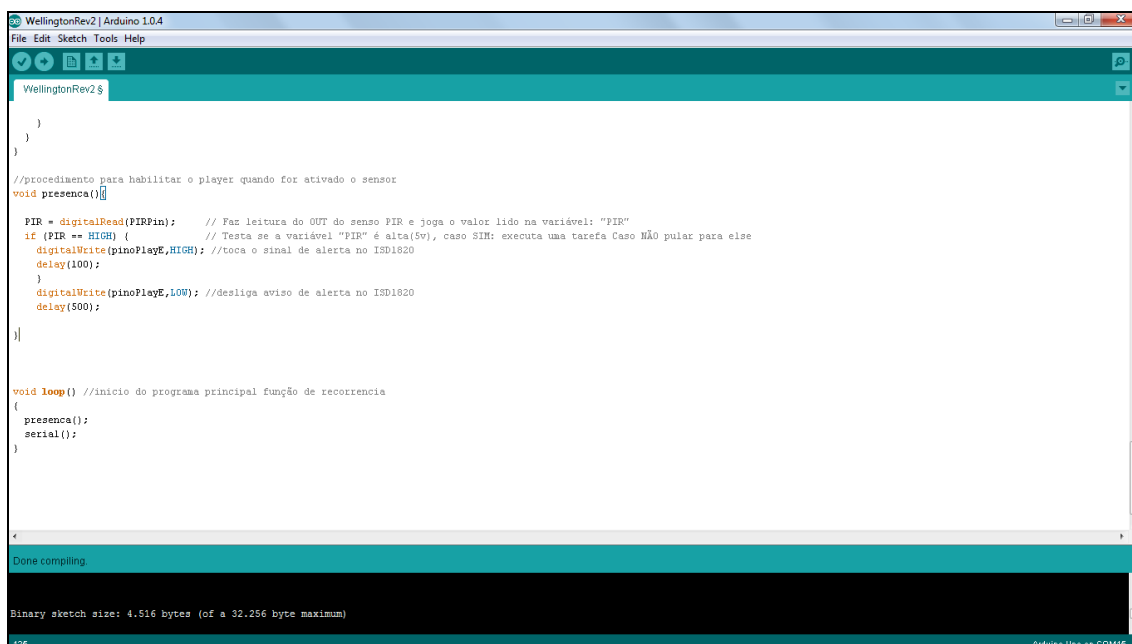


Figura 28 – Programa para habilitar o player quando for ativado o sensor

```

//procedimento para habilitar o player quando for ativado o sensor
void presenca(){

  PIR = digitalRead(PIRPin); // Faz leitura do OUT do senso PIR e joga o valor lido na variável:
  "PIR"
  if (PIR == HIGH) { // Testa se a variável "PIR" é alta(5v), caso SIM: executa uma tarefa
  Caso NÃO pular para else
    digitalWrite(pinoPlayE,HIGH); //toca o sinal de alerta no ISD1820
    delay(100);
  }
  digitalWrite(pinoPlayE,LOW); //desliga aviso de alerta no ISD1820
  delay(500);
}

```

Figura 29 – Procedimento para habilitar o player

### 5.5.3 Fase III – Comando de Voz

A terceira fase do projeto consiste em conectar a placa de Comando de Voz à placa *Arduino UNO* e estabelecer comunicação através de programa pré-definido.

O módulo de Comando de Voz (*Voice Recognition Module V2*) produzido pela ELECHOUSE ([www.elechouse.com](http://www.elechouse.com)) e utilizado neste projeto, está capacitado a realizar a gravação de até 15 comandos diferentes, onde estarão divididos em 3 grupos com 5 comandos cada. Cada comando de voz tem o comprimento máximo de 1300ms, que assegura que a maioria das palavras pode ser gravada e reconhecida. No processo de gravação do comando de voz, uma vez iniciado não se pode parar o processo de gravação até que se termine todos os 5 comandos de voz de um grupo. No projeto aqui apresentado utilizou-se apenas 4 comandos com a finalidade de controlar o acionamento do sinal sonoro juntamente com o *LED* na cor correspondente as lixeiras.

A placa tem como função reconhecer o comando de voz dado pelo deficiente visual. Para isso foi necessário a gravação de quatro mensagens correspondentes aos comandos de voz de acordo com o tipo de material a ser descartável na lixeira reciclável.

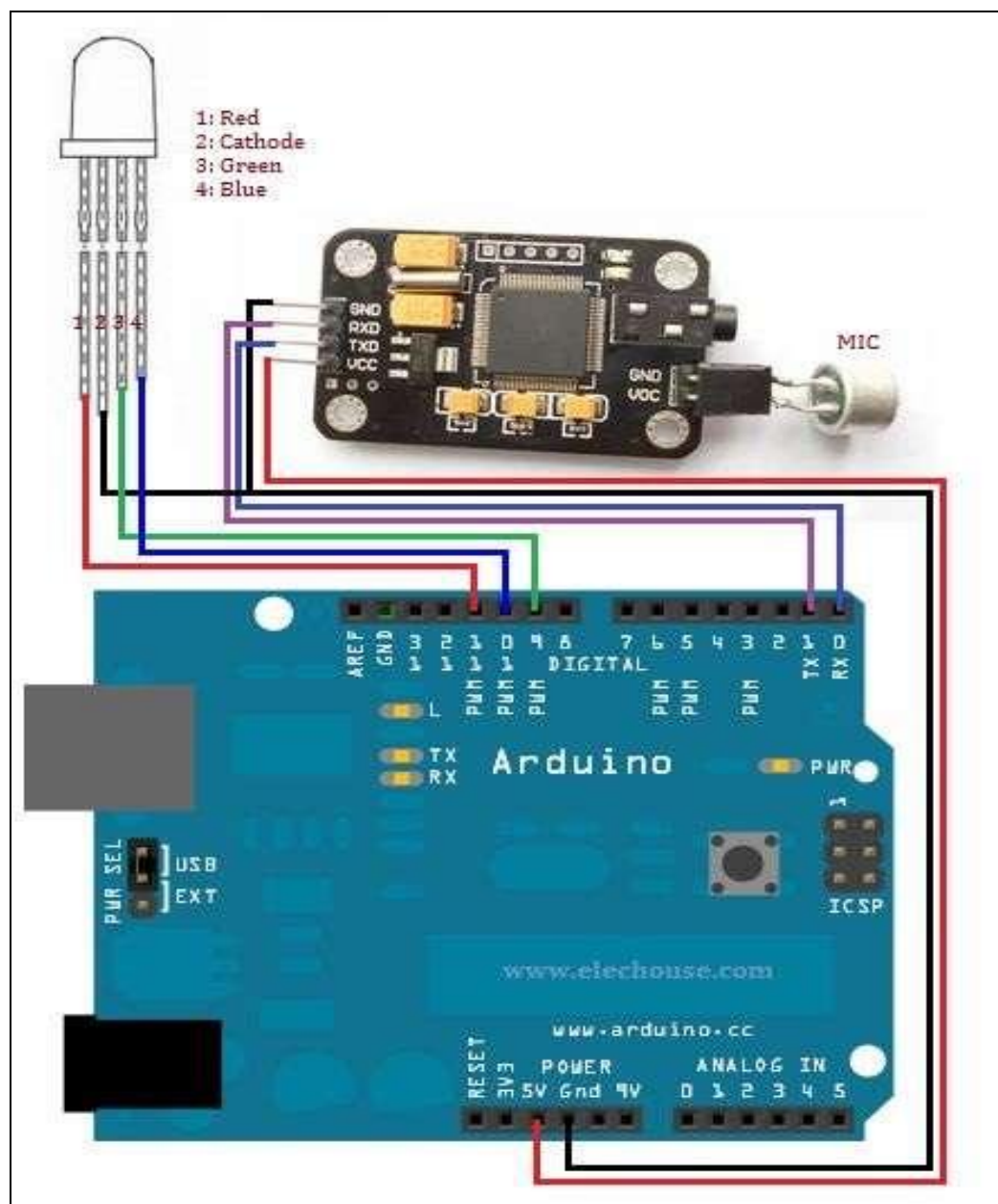
Dessa forma, quando o DV ou qualquer outro usuário disser o tipo de material que deseja descartar será acionado um *LED* e um sinal sonoro na lixeira própria correspondente, por exemplo: “quando o usuário optar em descartar uma lata de refrigerante na lixeira ele apenas falará “Lata”, então a lixeira que corresponde o material do tipo lata acenderá uma *LED* na cor amarela e emitirá um sinal sonoro durante quatro segundos, indicando dessa forma o local exato para o descarte.

A Figura 30 mostra a placa de Comando de Voz (*Voice Recognition Module V2*) e seus cabos para conexão.



Figura 30 – Placa de Comando de Voz (*Voice Recognition Module V2*)

A Figura 31 mostra a placa de Comando de Voz (*Voice Recognition Module V2*) conectada à placa *Arduino UNO*. Na placa há 4 cores de fios: GND (preto), RXT (Roxo), TXD (azul) e VCC (vermelho).



Porém, para gravar os comandos no microcontrolador do módulo, é necessário um conversor *USB TTL* (Figura 32) que irá interpretar os comandos enviados para o microcontrolador. Pode ser utilizado qualquer conversor *USB TTL* que possua interface de comunicação *USB* ou *Serial*.



Figura 32 – Conversor *USB TTL*

A Figura 33 mostra o esquema de ligação do *Voice Recognition Module V2* com o conversor *USB TTL* para que sejam enviados os comandos ao microcontrolador do módulo.

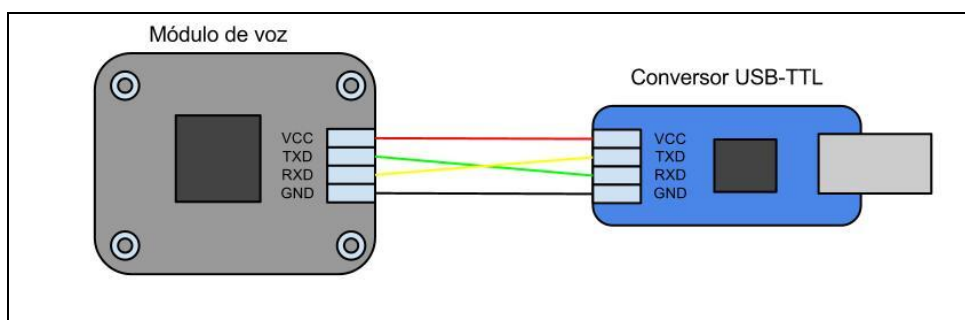


Figura 33 – Esquema de ligação do Módulo com o Conversor *USB TT*

Para a gravação desses comandos de voz utilizou-se o *Software* gratuito *AccessPort V1.37* (<http://www.sudt.com/en/ap/download.htm>), para comunicação serial (Figura 34).

Os comandos de configuração do módulo serão enviados a partir do *AccessPort V1.37*, sendo que estes comandos são pré definidos pelo fabricante do microcontrolador. Os comandos poderão ser encontrados no manual elaborado pela ELECHOUSE ([www.elechouse.com/elechouse/images/product/Voice Recognition Module/Manual.pdf](http://www.elechouse.com/elechouse/images/product/Voice%20Recognition%20Module/Manual.pdf)).

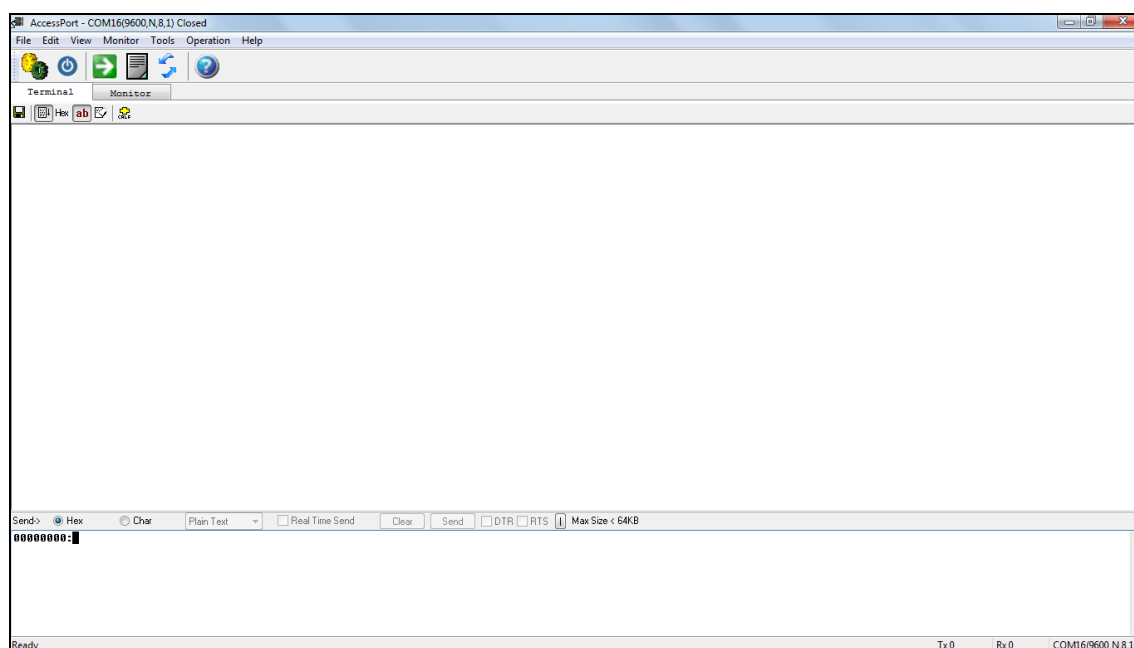


Figura 34 – Tela do *Software*

A Figura 35 mostra o processo de gravação no *AccessPort*. Para começar a gravação inicialmente se envia o comando “AA 11” que representa ao microcontrolador a ação de gravar o primeiro grupo de comandos, sendo que é possível gravar até 3 grupos de comandos com 5 tipos diferentes em cada. A completa gravação de um comando é representada como “*Finish one*”, sendo que ao finalizar a gravação dos 5 comandos do grupo a mensagem “*Group1 finished*” é exibida. Depois de finalizada a gravação do primeiro grupo de instruções foi enviado o comando “AA 21” que representa ao microcontrolador a ação de importar os comandos gravados no grupo 1 e iniciar o reconhecimento destes comandos.

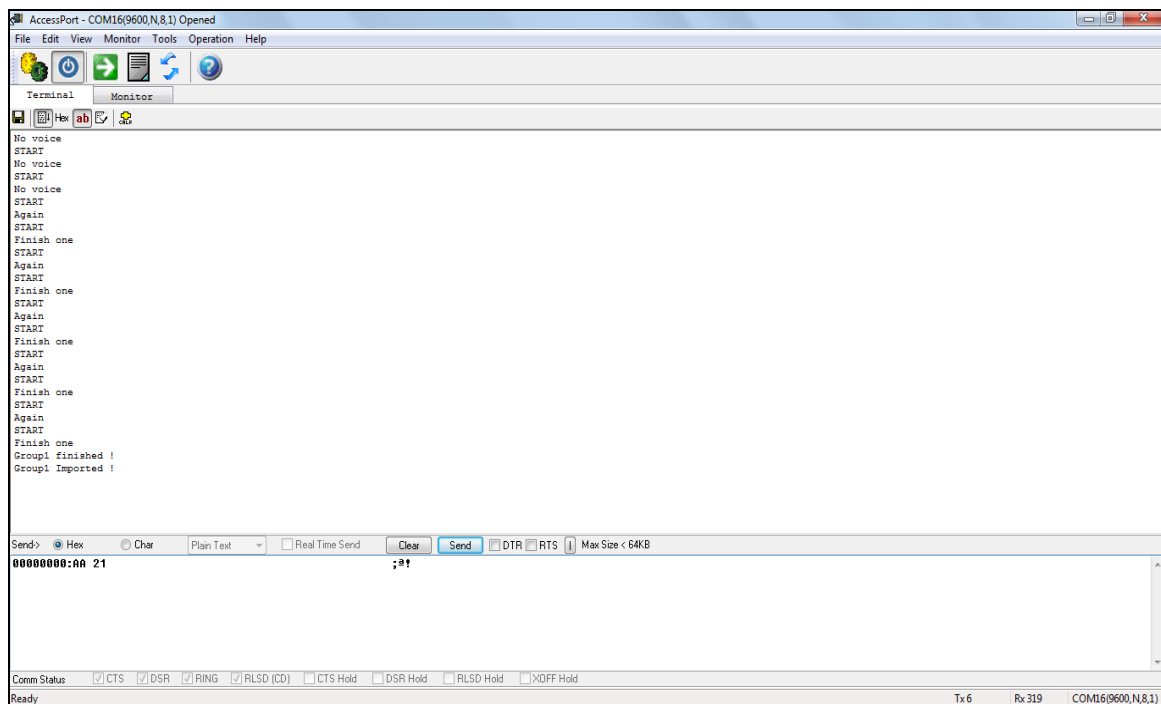


Figura 35 – Gravação dos comandos para o Módulo

Com a programação do módulo concluída pode-se iniciar a montagem do circuito junto ao *Arduino* conforme a Figura 36.

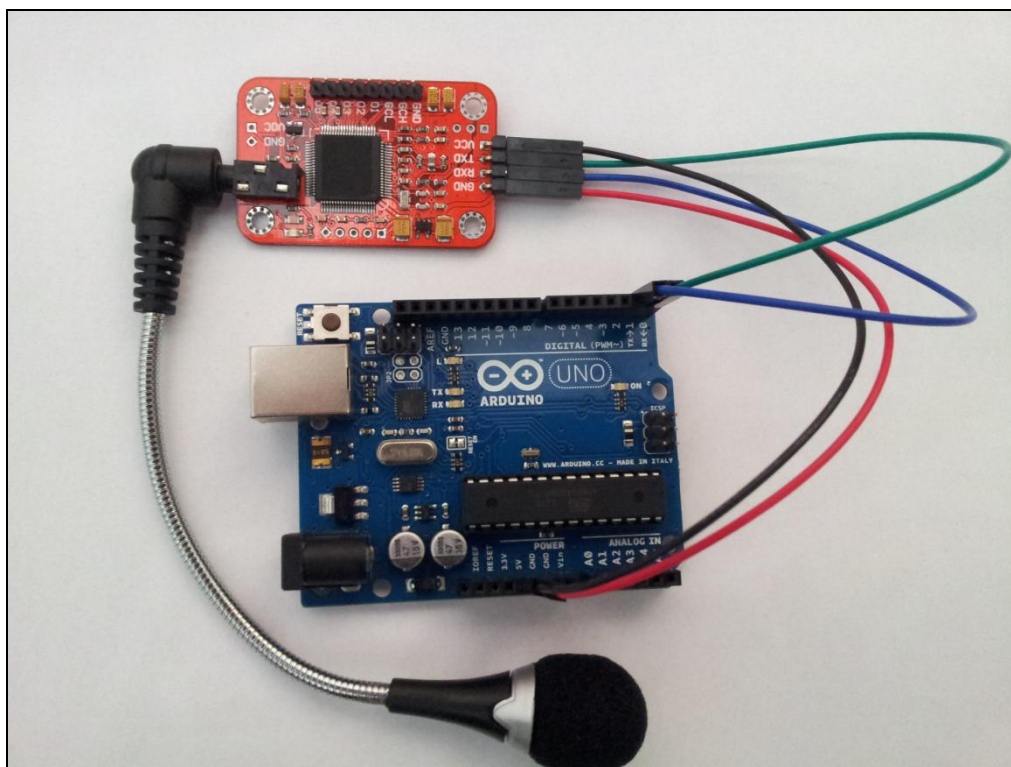


Figura 36 – Esquema de conexão do Módulo de Voz na placa *Arduino*



A Figura 37 mostra o Diagrama de Conexão Elétrico da placa de Comando de Voz (*Voice Recognition Module V2*) conectada à placa *Arduino UNO*. Na placa há 4 tipos de cores de fios conforme sequencia abaixo:

- GND (**preto**)
- RXT (**Roxo**)
- TXD (**azul**)
- VCC (**vermelho**)

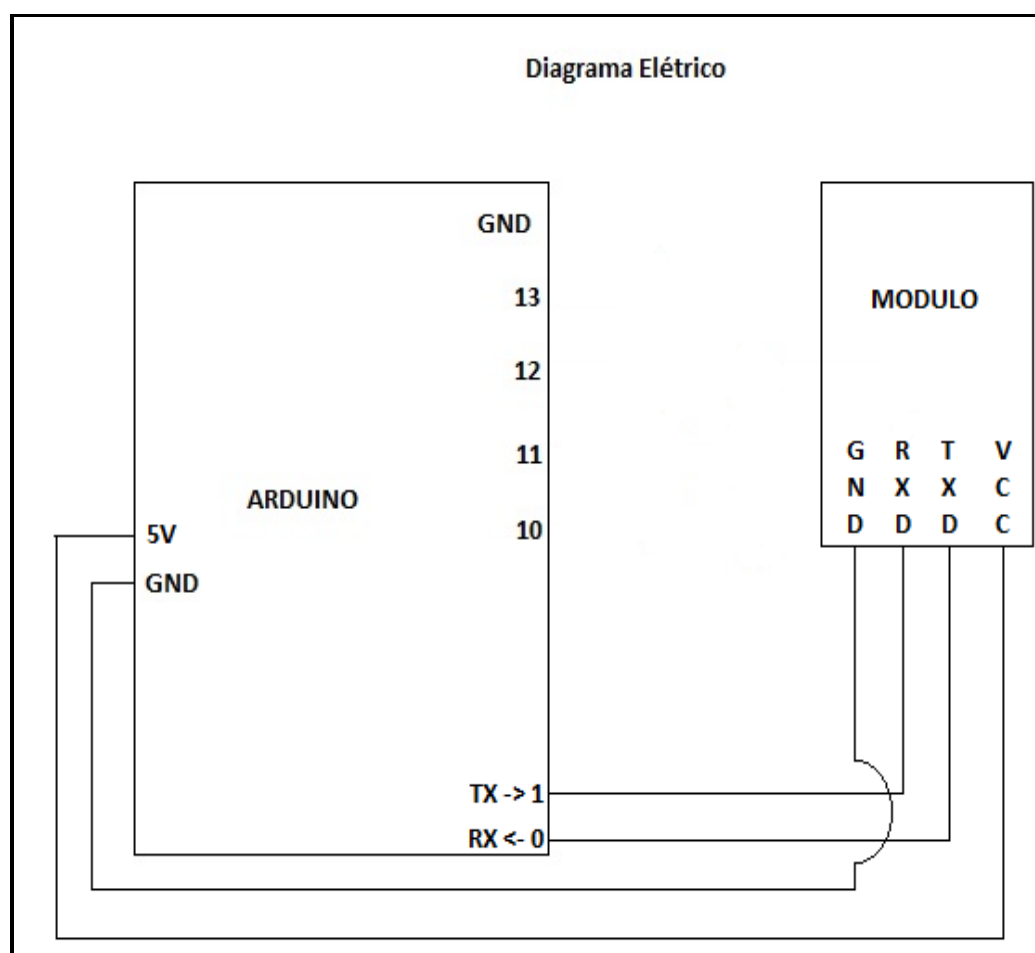
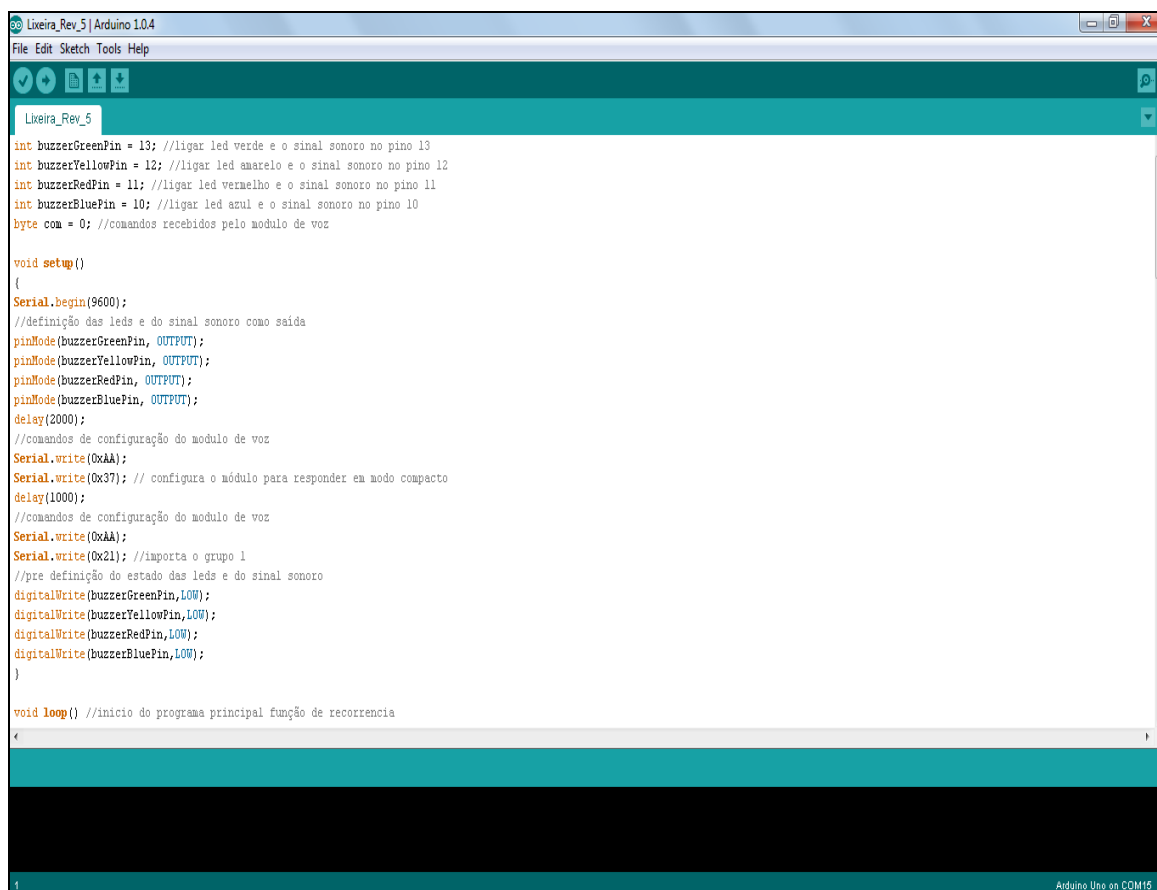


Figura 37 – Diagrama de Conexão Elétrico do Módulo de Voz na placa *Arduino*



Depois de feita a conexão do Módulo de Voz com o *Arduino*, foi necessário programar a placa *Arduino*.

Na primeira parte do programa (Figura 38), o *Arduino* inicializa o grupo 1 e fica aguardando o comando enviado pelo módulo que por sua vez aguarda comandos de voz compatíveis com os gravados.



```
Liveira_Rev_5 | Arduino 1.0.4
File Edit Sketch Tools Help

Liveira_Rev_5

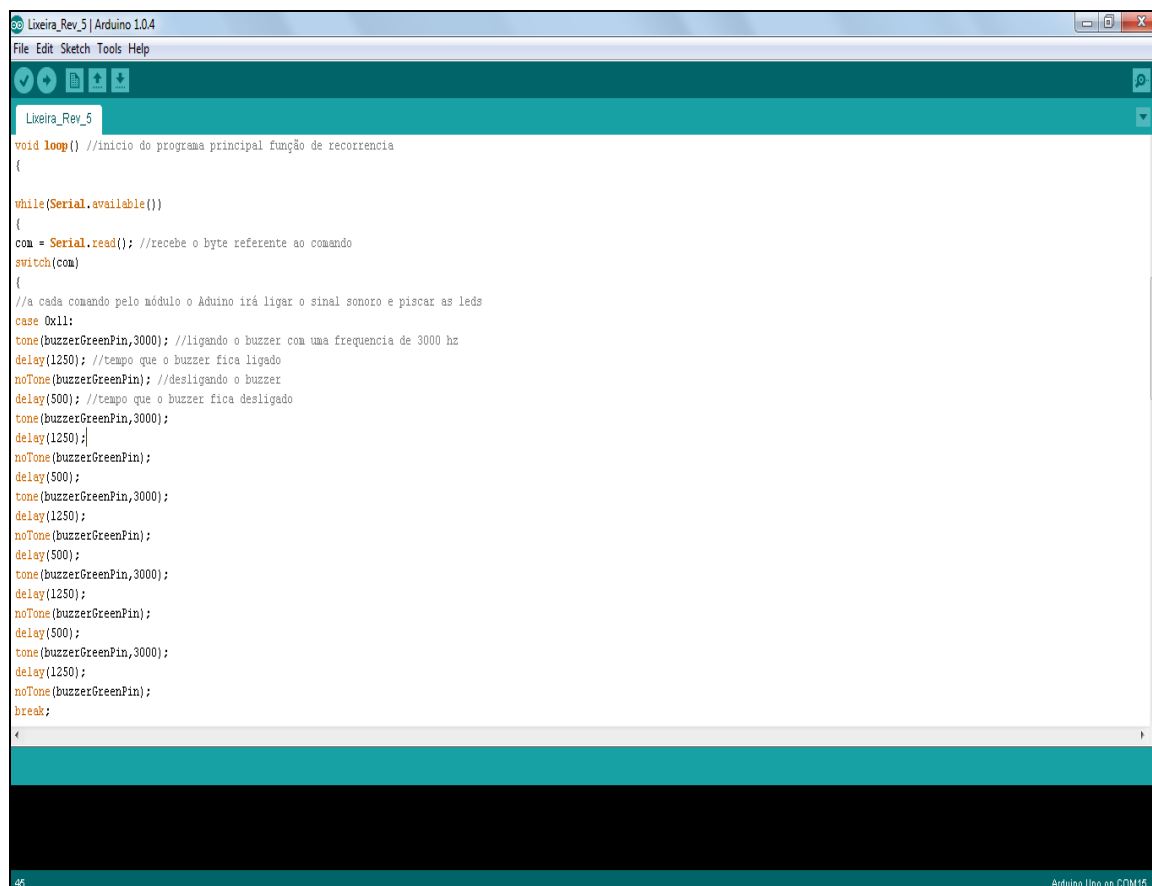
int buzzerGreenPin = 13; //ligar led verde e o sinal sonoro no pino 13
int buzzerYellowPin = 12; //ligar led amarelo e o sinal sonoro no pino 12
int buzzerRedPin = 11; //ligar led vermelho e o sinal sonoro no pino 11
int buzzerBluePin = 10; //ligar led azul e o sinal sonoro no pino 10
byte com = 0; //comandos recebidos pelo modulo de voz

void setup()
{
  Serial.begin(9600);
  //definição das leds e do sinal sonoro como saída
  pinMode(buzzerGreenPin, OUTPUT);
  pinMode(buzzerYellowPin, OUTPUT);
  pinMode(buzzerRedPin, OUTPUT);
  pinMode(buzzerBluePin, OUTPUT);
  delay(2000);
  //comandos de configuração do modulo de voz
  Serial.write(0xAA);
  Serial.write(0x37); // configura o módulo para responder em modo compacto
  delay(1000);
  //comandos de configuração do modulo de voz
  Serial.write(0xAA);
  Serial.write(0x21); //importa o grupo 1
  //pre definição do estado das leds e do sinal sonoro
  digitalWrite(buzzerGreenPin,LOW);
  digitalWrite(buzzerYellowPin,LOW);
  digitalWrite(buzzerRedPin,LOW);
  digitalWrite(buzzerBluePin,LOW);
}

void loop() //início do programa principal função de recorrência
{
}
```

Figura 38 – Tela 1 – Programa aguardando o comando de voz compatível

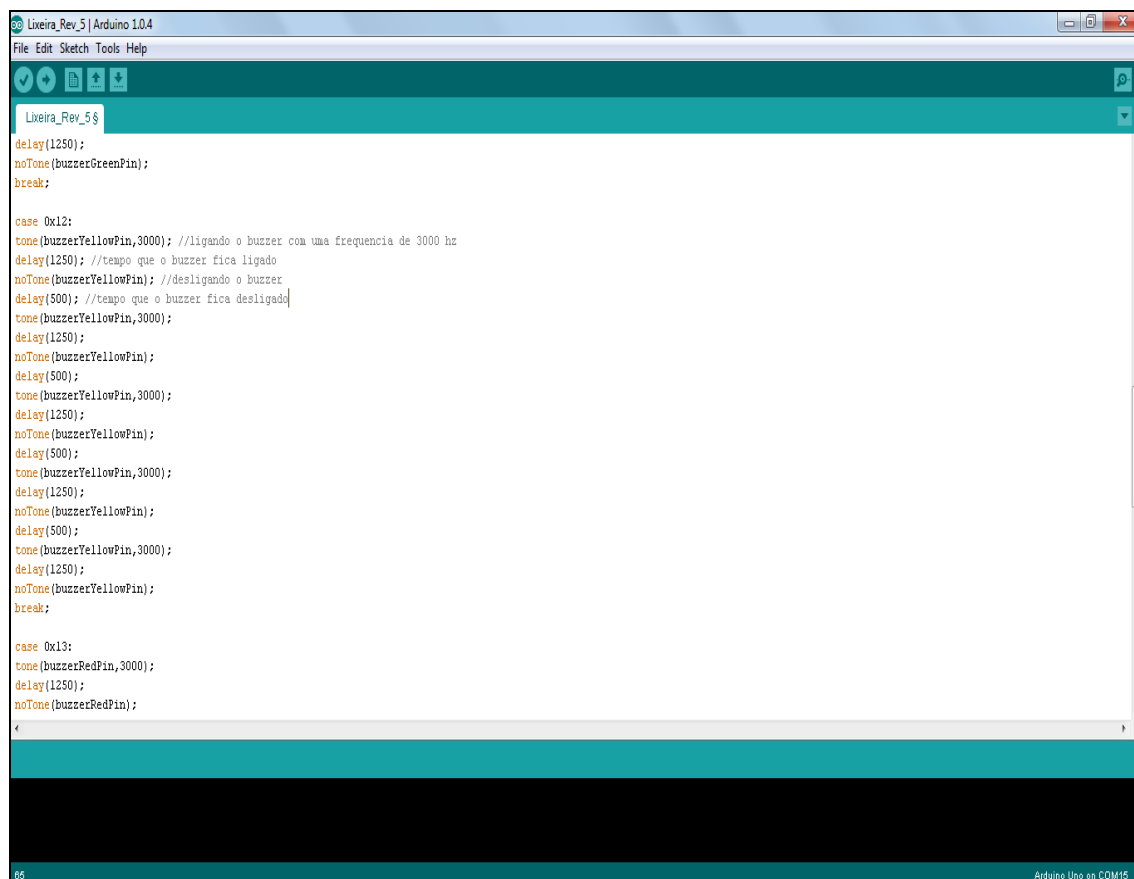
Na segunda parte do programa, a cada comando dado pelo módulo o *Arduino* irá ligar o sinal sonoro e piscar a *LED* Verde. No código fonte, a função `case 0x11` (Figura 39) à emissão do som se dá através da função `tone`, onde o segundo parâmetro informa a frequência do som que você quer emitir, que neste caso é *3000hz*. O código fonte abaixo faz o mesmo emitir um som de *3000hz* num intervalo de um em um segundo.



```
void loop() //inicio do programa principal função de recorrência
{
  while(Serial.available())
  {
    com = Serial.read(); //recebe o byte referente ao comando
    switch(com)
    {
      //a cada comando pelo módulo o Aduino irá ligar o sinal sonoro e piscar as leds
      case 0x11:
        tone(buzzerGreenPin,3000); //ligando o buzzer com uma frequencia de 3000 hz
        delay(1250); //tempo que o buzzer fica ligado
        noTone(buzzerGreenPin); //desligando o buzzer
        delay(500); //tempo que o buzzer fica desligado
        tone(buzzerGreenPin,3000);
        delay(1250);
        noTone(buzzerGreenPin);
        delay(500);
        tone(buzzerGreenPin,3000);
        delay(1250);
        noTone(buzzerGreenPin);
        delay(500);
        tone(buzzerGreenPin,3000);
        delay(1250);
        noTone(buzzerGreenPin);
        delay(500);
        tone(buzzerGreenPin,3000);
        delay(1250);
        noTone(buzzerGreenPin);
        break;
    }
  }
}
```

Figura 39 – Tela 2 - Ativação do sinal sonoro junto com a *LED* Verde.

Terceira parte do programa (Figura 40) a cada comando dado pelo módulo o *Arduino* irá ligar o sinal sonoro e piscar a *LED* Amarela. No código fonte, a função `case 0x12` à emissão do som se dá através da função `tone`, onde o segundo parâmetro informa a frequência do som que você quer emitir, que neste caso é 3000hz. O código fonte faz o mesmo emitir um som de 3000hz num intervalo de um em um segundo.



```
Liveira_Rev_5 | Arduino 1.0.4
File Edit Sketch Tools Help

Liveira_Rev_5$

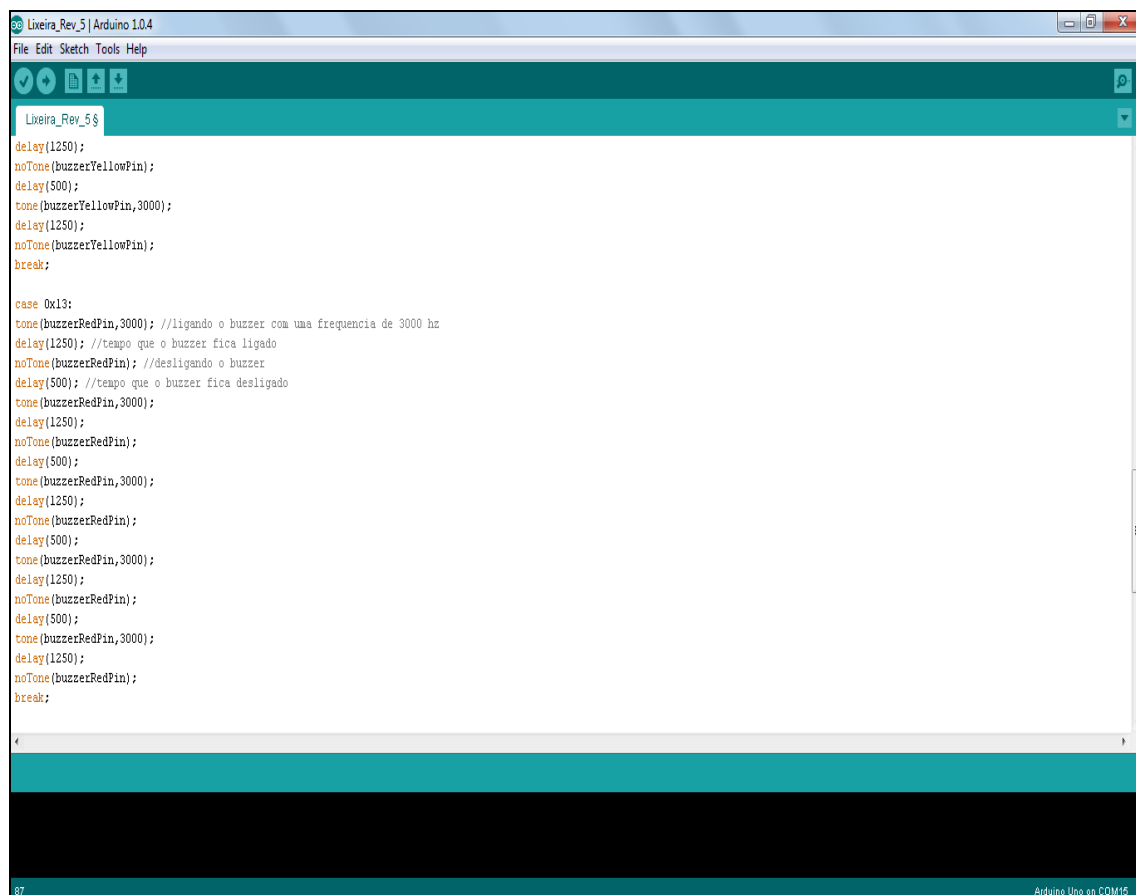
delay(1250);
noTone(buzzerGreenPin);
break;

case 0x12:
tone(buzzerYellowPin,3000); //ligando o buzzer com uma frequencia de 3000 hz
delay(1250); //tempo que o buzzer fica ligado
noTone(buzzerYellowPin); //desligando o buzzer
delay(500); //tempo que o buzzer fica desligado
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
break;

case 0x13:
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
```

Figura 40 – Tela 3 - Ativação do sinal sonoro junto com a *LED* Amarela.

Na quarta parte do programa, a cada comando dado pelo módulo o *Arduino* irá ligar o sinal sonoro e piscar a *LED* Vermelha. No código fonte, a função `case 0x13` (Figura 41) à emissão do som se dá através da função `tone`, onde o segundo parâmetro informa a frequência do som que você quer emitir, que neste caso é 3000hz. O código fonte faz o mesmo emitir um som de 3000hz num intervalo de um em um segundo.

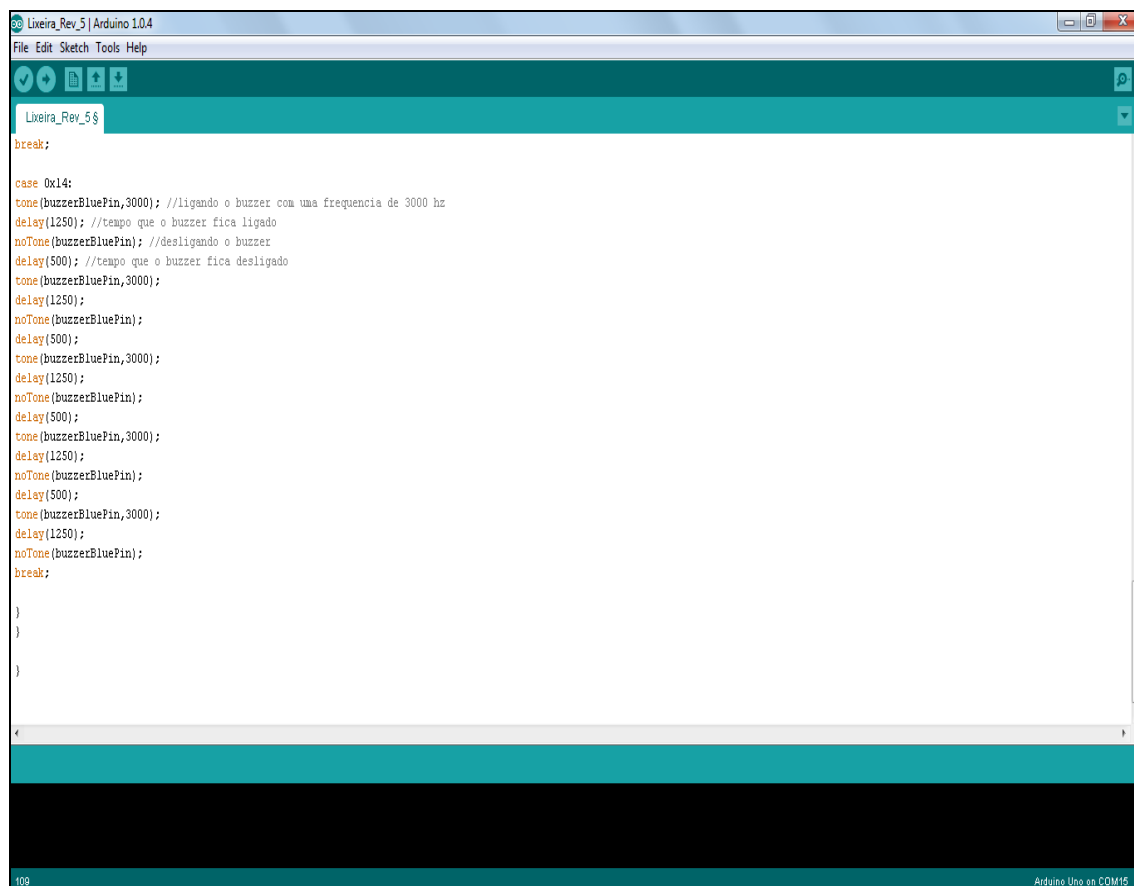


```
Lixeira_Rev_5$
delay(1250);
noTone(buzzerYellowPin);
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
break;

case 0x13:
tone(buzzerRedPin,3000); //ligando o buzzer com uma frequencia de 3000 hz
delay(1250); //tempo que o buzzer fica ligado
noTone(buzzerRedPin); //desligando o buzzer
delay(500); //tempo que o buzzer fica desligado
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
break;
```

Figura 41 – Tela 4 - Ativação do sinal sonoro junto com a *LED* Vermelha.

Última parte do programa a cada comando dado pelo módulo o *Arduino* irá ligar o sinal sonoro e piscar a *LED* Azul. No código fonte, a função `case 0x14` da (Figura 42) à emissão do som se dá através da função `tone`, onde o segundo parâmetro informa a frequência do som que você quer emitir, que neste caso é *3000hz*. O código fonte faz o mesmo emitir um som de *3000hz* num intervalo de um em um segundo.



```

break;

case 0x14:
tone(buzzerBluePin,3000); //ligando o buzzer com uma frequencia de 3000 hz
delay(1250); //tempo que o buzzer fica ligado
noTone(buzzerBluePin); //desligando o buzzer
delay(500); //tempo que o buzzer fica desligado
tone(buzzerBluePin,3000);
delay(1250);
noTone(buzzerBluePin);
delay(500);
tone(buzzerBluePin,3000);
delay(1250);
noTone(buzzerBluePin);
delay(500);
tone(buzzerBluePin,3000);
delay(1250);
noTone(buzzerBluePin);
delay(500);
tone(buzzerBluePin,3000);
delay(1250);
noTone(buzzerBluePin);
break;

}

}

}

```

Figura 42 – Tela 5 - Ativação do sinal sonoro junto com a *LED* Azul.

Os comandos gravadas foram de acordo com tipo de material que deve ser descartado nas lixeiras recicláveis. Para a lixeira da cor verde foi gravado o comando “Vidro”, na cor amarela o comando gravado foi “Lata”, na cor vermelha o comando gravado foi “Plástico” e na cor azul o comando gravado foi “Papel”. Então quando o Deficiente Visual ou qualquer outro usuário disser o tipo de material que deseja descartar será acionada uma *LED* e um sinal sonoro na lixeira correspondente. Por exemplo, quando o usuário quiser jogar uma lata de refrigerante na lixeira, ele apenas dirá “Lata”, então a lixeira que corresponde o material do tipo

lata acenderá uma *LED* na cor amarela e emitirá um sinal sonoro durante quatro segundos.

O *Voice Recognition Module V2* mostrou-se muito eficaz nos testes com diferentes tipos de vozes, onde foram também reconhecidas vozes do tipo masculino, feminino e infantil.

#### 5.5.4 Fase IV – *LEDs* e Sinal Sonoro

Esta fase constitui a montagem física da placa *Arduino Uno* e do *LED* juntamente em uma matriz de contato chamada *Protoboard*, onde serão feitas as conexões condutoras para a montagem de circuitos elétricos para inserção de todos os componentes sem a necessidade de solda.

A Figura 43 mostra a placa *Arduino Uno* ligada à um *LED* de cor vermelha através da matriz de contato *Protoboard*.

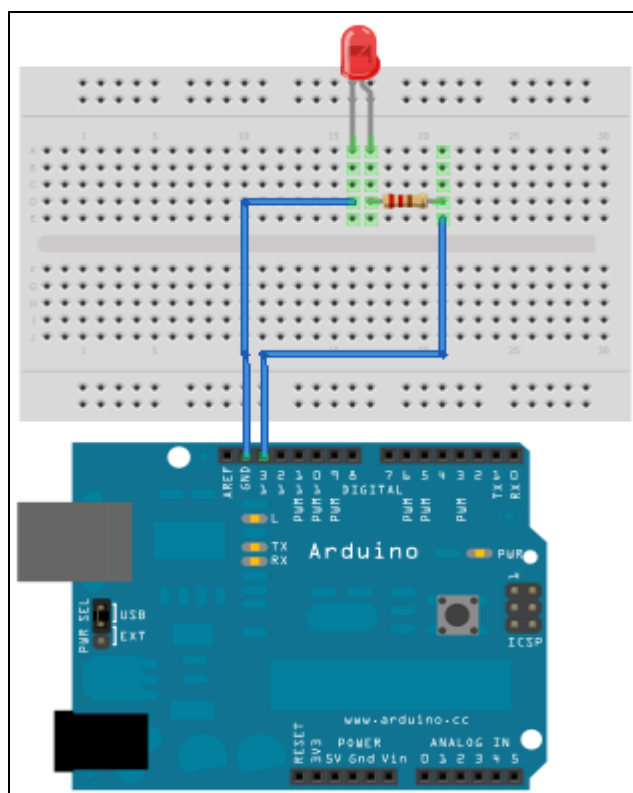


Figura 43 – Esquema de conexão do *LED*

A Figura 44 mostra a linha de códigos do programa do *LED*.

*pinMode(13, OUTPUT);* No loop principal, você acende o LED com o código:

*digitalWrite(13, HIGH);*

Isto supõe 5 volts ao pino 13. Que cria a diferença de potencial no LED, acendendo-o. Então você o desliga com o código:

*digitalWrite(13, LOW);*

Que retorna o pino 13 a 0 volts, e apaga o LED. Entre o acender e apagar do LED, você necessita de um tempo para a pessoa perceber a mudança de estado, então o comando *delay()* é usado.

*/\**

*Blink*

*Turns on an LED on for one second, then off for one second, repeatedly.*

*This example code is in the public domain.*

*\*/*

*void setup() {*

*// initialize the digital pin as an output.*

*// Pin 13 has an LED connected on most Arduino boards:*

*pinMode(13, OUTPUT);*

*}*

*void loop() {*

*digitalWrite(13, HIGH); // set the LED on*

*delay(1000); // wait for a second*

*digitalWrite(13, LOW); // set the LED off*

*delay(1000); // wait for a second*

*}*

Figura 44 – Código do Programa do *LED*.

Para reproduzir um sinal sonoro em cada lixeira foi usado um *Buzzer* para cada cor de *LED*. *Buzzer* (Figura 5) nada mais é do que um pequeno alto-falante capaz de emitir sons em diversas frequências, ou seja, é um dispositivo de sinalização de áudio, onde sua função específica é emitir som. Seu acionamento e temporização serão determinados em nosso projeto conforme uma linha de programação específica.

Dentro do *Buzzer* existem duas camadas de metal e uma camada interna de cristal piezo elétrico. Quando é enviada uma corrente elétrica através do *Buzzer*, a camada interna vibra na mesma frequência reproduzindo assim um som. Se enviarmos corrente em uma frequência de 440hz, como por exemplo, será emitido o som da nota Lá. Ou seja, quanto maior a corrente maior o “volume” do som. O *Buzzer* utilizado nesse projeto é muito pequeno, com isso a frequência utilizada é de 3000hz.

A Figura 45 apresenta o esquema de instalação do dispositivo de sinalização de áudio *BUZZER* em placa *Arduino Uno* juntamente com os *LEDs* correspondentes a cor de cada lixeira.

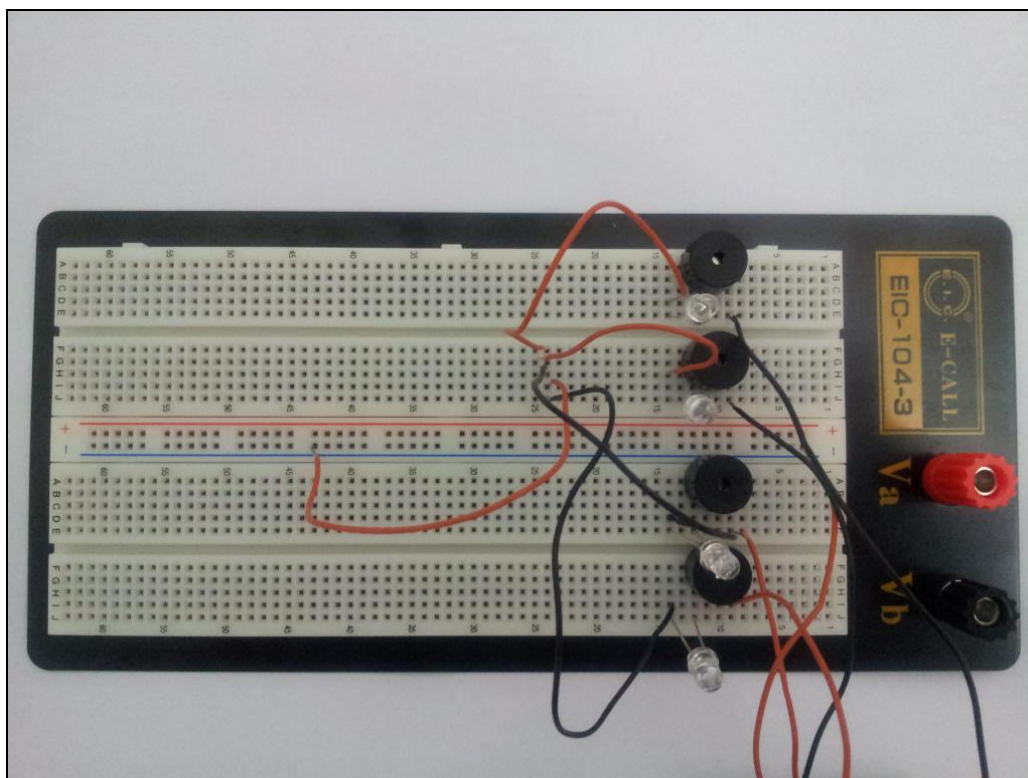


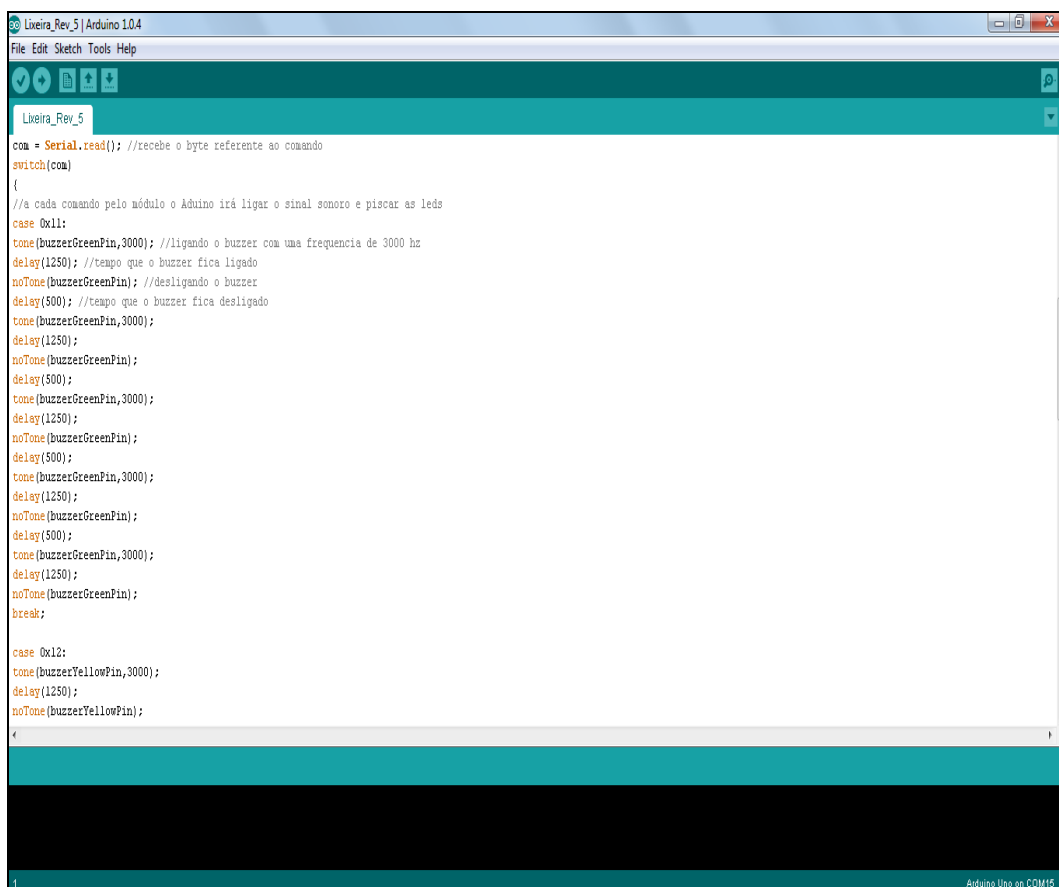
Figura 45 – *Buzzer* conectado a placa *Protoboard*.



No presente projeto, para o *Buzzer* reproduzir um som foi preciso fazer com que um pino emita corrente e pare de emitir em uma determinada frequência, isso é um pouco complicado de ser fazer na mão (mas não é impossível). Por isso a linguagem de programação do *Arduino* vem com uma função chamada `tone ( )`.

Conforme mostrado na Figura 46, esta função pede o número de um pino, a frequência e, opcionalmente, a duração.

Além da função `tone ( )` existe a função “`no Tone ( )`”, que recebe o número de um pino e finaliza o som neste pino.



```

Linha_Rev_5 | Arduino 1.0.4
File Edit Sketch Tools Help

Linha_Rev_5

com = Serial.read(); //recebe o byte referente ao comando
switch(com)
{
  //a cada comando pelo módulo o Arduino irá ligar o sinal sonoro e piscar as leds
  case 0x11:
    tone(buzzerGreenPin,3000); //ligando o buzzer com uma frequencia de 3000 hz
    delay(1250); //tempo que o buzzer fica ligado
    noTone(buzzerGreenPin); //desligando o buzzer
    delay(500); //tempo que o buzzer fica desligado
    tone(buzzerGreenPin,3000);
    delay(1250);
    noTone(buzzerGreenPin);
    delay(500);
    tone(buzzerGreenPin,3000);
    delay(1250);
    noTone(buzzerGreenPin);
    delay(500);
    tone(buzzerGreenPin,3000);
    delay(1250);
    noTone(buzzerGreenPin);
    delay(500);
    tone(buzzerGreenPin,3000);
    delay(1250);
    noTone(buzzerGreenPin);
    break;

  case 0x12:
    tone(buzzerYellowPin,3000);
    delay(1250);
    noTone(buzzerYellowPin);
  }
}
  
```

Figura 46 – Programação para emitir sinal sonoro do *buzzer*

Os *LEDs* foram colocados na mesma corrente elétrica que os *Buzzers*, assim quando é enviada uma corrente elétrica para o *Buzzer* automaticamente o *LED* é acionado. No presente projeto o sinal sonoro enviado pelo *Buzzer* é de um bipe alternado durante 4 segundos, ou seja, o *LED* ficará piscando também durante 4 segundos.

A Figura 47 apresenta o diagrama elétrico do funcionamento do *Buzzer* com o *LED* do *Arduino*.

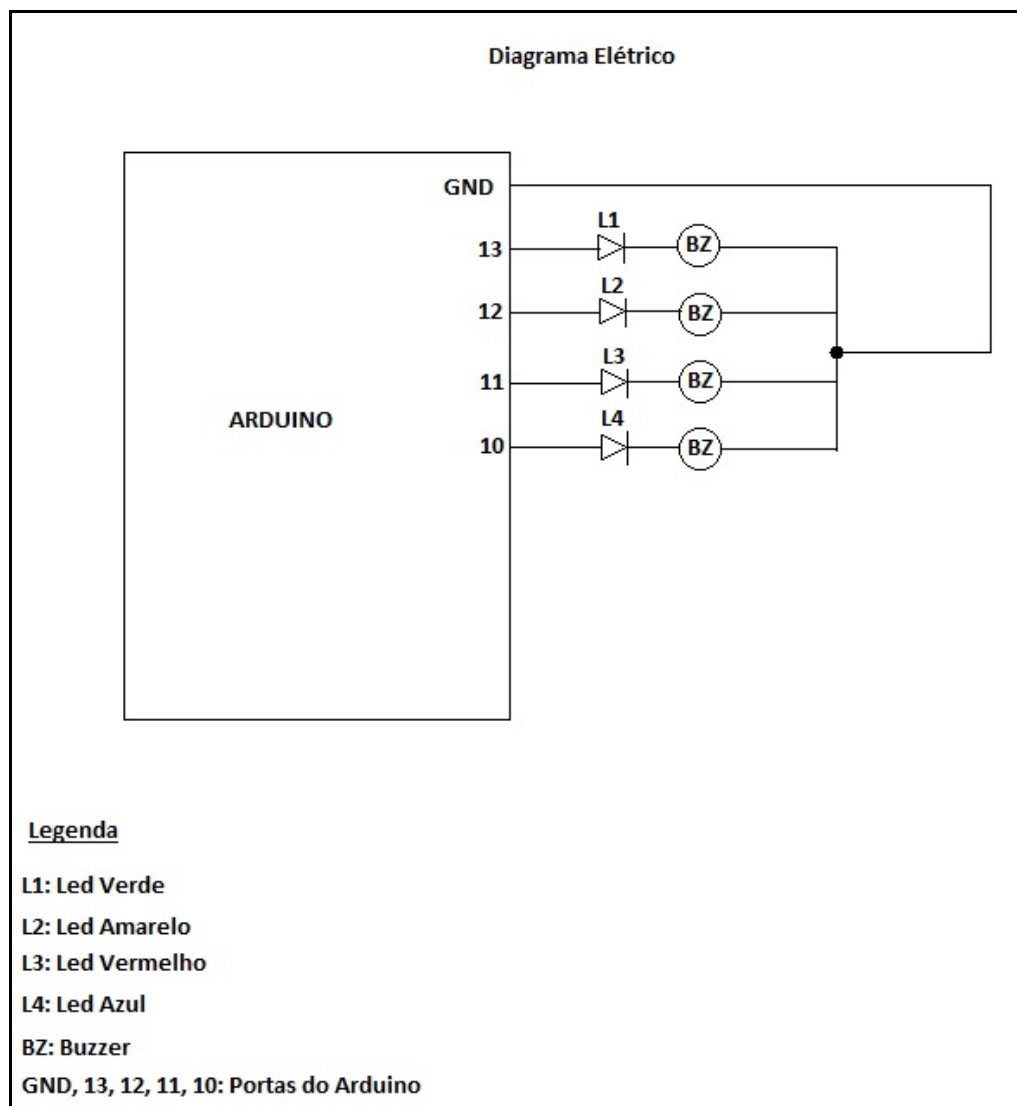


Figura 47 – Diagrama Elétrico: Sinal sonoro com a *LED*.

O Apêndice A contém as linhas de programação para que cada *LED* acenda juntamente com o *Buzzer* correspondente a cor da lixeira. Será habilitado para que cada *LED* pisque por até 4 segundos juntamente com o *Buzzer*, que emitirá o sinal sonoro assim que o *LED* acender durante toda a sequência de 4 segundos.

Por fim, o Apêndice B contém o código fonte de toda a programação final do sistema.

## 6 RESULTADOS E DISCUSSÕES

Este capítulo apresenta uma abordagem dos resultados alcançados a partir do desenvolvimento do trabalho.

O sistema desenvolvido segue o padrão de cores estabelecido pelo Conselho Nacional do Meio Ambiente – CONAMA [9] (Figura 48) que é o órgão que institui normas para proteção do meio ambiente e para ações que visam à qualidade de vida coletiva. Pessoas com deficiência visual podem realizar o descarte correto de itens em lixeiras de coleta seletiva.

O projeto visa contribuir com Políticas Públicas ensinando crianças a respeitar o meio ambiente. Por fim, propõe práticas de descarte consciente e exercício da cidadania.



Figura 48 – Sistema desenvolvido segue as cores padrão do CONAMA

## 6.1 Funcionamento da Lixeira

No presente projeto todos os componentes serão apresentados dentro de um mesmo ambiente, ou seja, placas, protaboard, sensor, *LEDs*, *buzzers* e lixeiras foram alocados todos na mesma plataforma de exposição, e para uma melhor compreensão e visualização de seu funcionamento, as lixeiras foram desenvolvidas em tamanho reduzido.

Foram utilizados pequenos potes plásticos transparentes (Figura 49) para simular um modelo de lixeira com dimensões pequenas e que fosse adequado para representar o projeto



Figura 49 – Potes sem acabamento.

Dentro da ideia de criação de um modelo personalizado de lixeira (Figura 50), específico para a apresentação e que pudesse atender as exigência do projeto, foi elaborada uma tampa, confeccionada toda em papel cartão, em um formato que facilitasse o descarte de resíduos sem que o usuário tenha contato direto com outros produtos já descartados. Para isso foi criado uma “rampa” na entrada principal da lixeira que dificulta a retirada de qualquer produto uma vez descartado pelo usuário, evitando assim que pessoas fiquem retirando materiais já depositados.



Figura 50 – Criação e desenvolvimento da lixeira.

A escolha do pote transparente colaborou para a demonstração do projeto e também na identificação da lixeira pelo usuário, já que uma vez escolhida a opção de descarte o *LED* irá acender dentro da própria lixeira e sua luz de auto brilho acenderá por completo iluminando todo o recipiente. O *LED* de auto brilho estará instalado, para demonstração, na parte interna e sua luz será refletida pelo papel laminado instalado no fundo da lixeira.



A Figura 51 apresenta a maquete da lixeira pronta para os testes. Os *LEDs* de auto brilho e os *buzzers* foram colocados na parte frontal da lixeira, facilitando a compreensão de todo funcionamento do sistema e também da instalação do cabeamento.

A lixeira encontra-se com os adesivos na cor padrão determinada pelo CONAMA.



Figura 51 – Lixeira com *LEDs* e *buzzers* na parte frontal para demonstração.

As Figuras 52 e 53 apresentam toda a parte de cabeamento distribuída para a conexão e o funcionamento correto do sistema.

Nota-se que as lixeiras utilizam a maior quantidade de cabos para seu funcionamento, ou seja, um par de fios para o *LED* e mais um par de fios para o *Buzzer*. Estes cabos passam primeiro pelo *Protoboard* antes de serem distribuídos entre as placas controladoras.

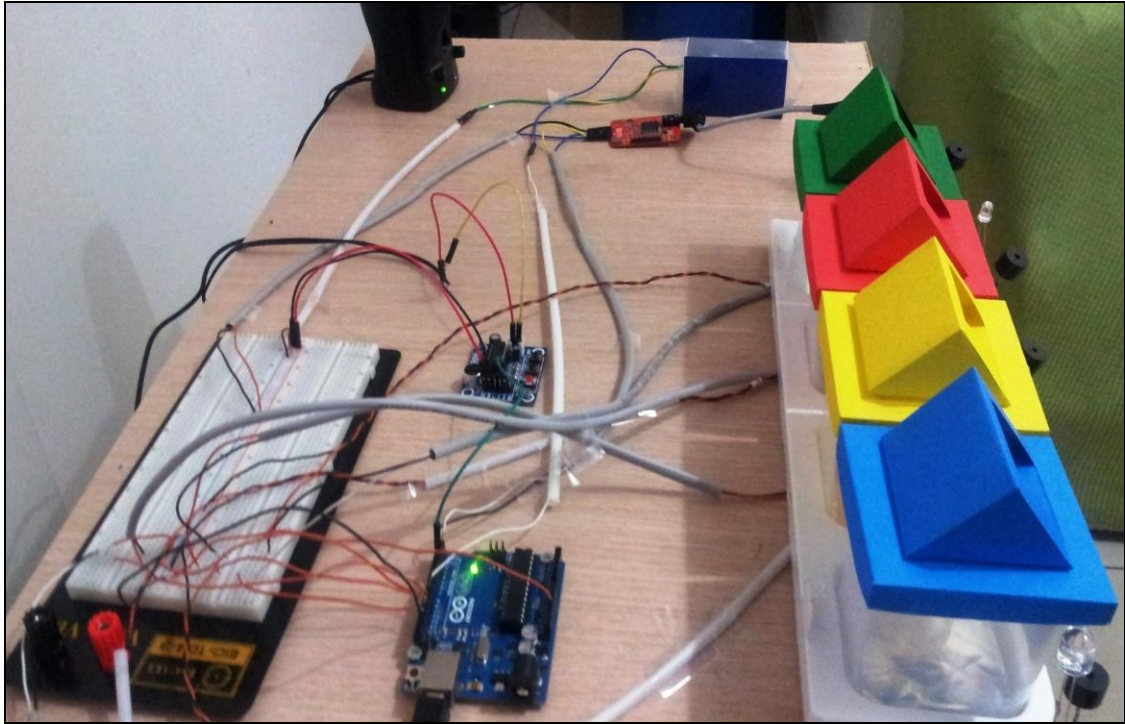


Figura 52 – Conexão e distribuição dos cabos através do *Protoboard*.

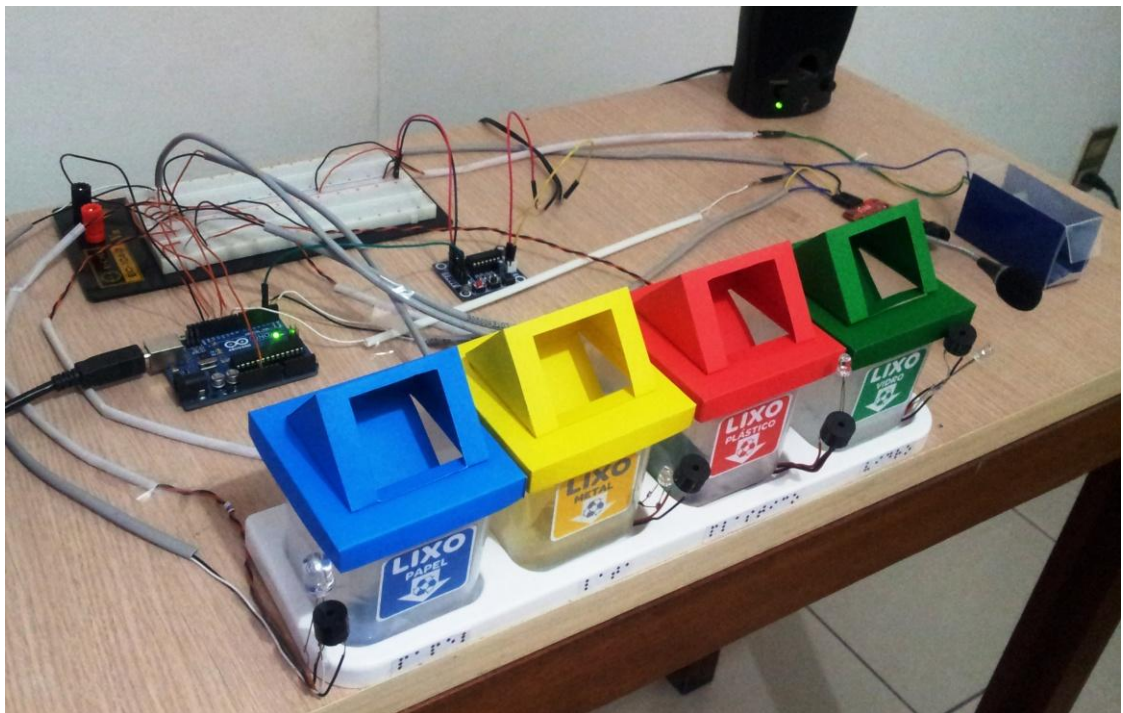


Figura 53 – Conexão e distribuição dos cabos.

Na primeira etapa de funcionamento da lixeira, quando o usuário se aproxima para fazer o descarte do material reciclável (Figura 54), ele será detectado pelo sensor infravermelho que ativará o módulo de gravação de voz.



Figura 54 – Sensor infravermelho detectando a presença de um usuário

No módulo de gravação de voz ISD1820 estará gravada uma mensagem de saudação do tipo: “Ola, por favor, falar qual o tipo de material que você deseja jogar fora: Vidro, Plástico, Lata ou Papel.”.

Essa mensagem notificará o usuário para que ele diga no microfone, colocado ao lado da lixeira, o tipo de material que tenha em mãos para o descarte: papel, lata, vidro ou plástico.



A Figura 55 mostra o usuário utilizando o microfone para falar a palavra “papel”, que foi o material escolhido para ser descartado nesta apresentação.



Figura 55 – Usuário utilizando o microfone do sistema.

Após o usuário dizer o tipo de objeto que tenha em mãos, ou seja, papel (lixeira cor azul), vidro (lixeira cor verde), lata (lixeira cor amarela) ou plástico (lixeira cor vermelho), será acionada a placa de Reconhecimento de Voz (*Voice Recognition Module V2*), que tem a função de reconhecer o comando de voz dado pelo usuário e indicar o local exato para o descarte através do acionamento do *LED* e do sinal sonoro.

Esta placa reconhece os comandos de voz através de palavras chaves pré-definidas, como por exemplo: papel, vidro, lata e plástico.

A Figura 56 mostra que após o usuário falar o comando de voz “Papel”, o sistema aciona o sinal sonoro através do *buzzer* juntamente com o *LED* na cor azul, que corresponde à lixeira para o descarte do papel.

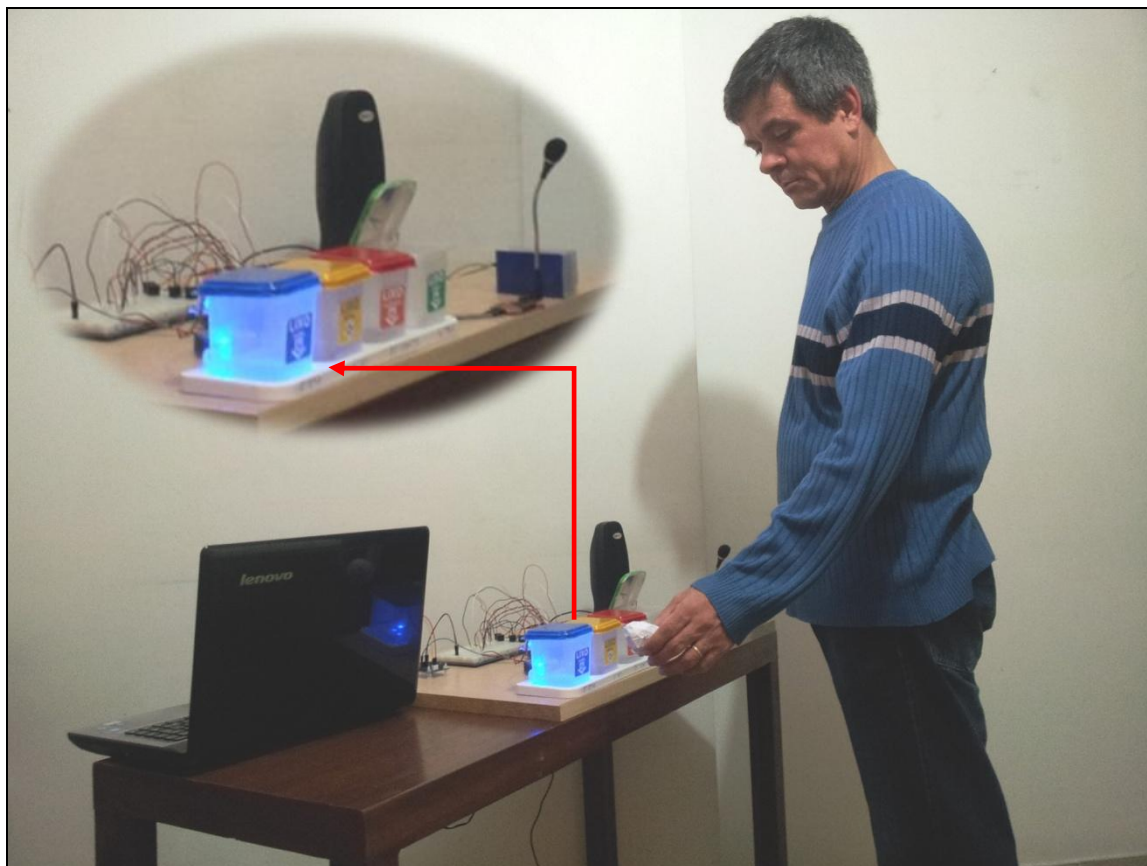


Figura 56 – *LED* e sinal sonoro indicando local para descarte

O aviso luminoso e o sinal sonoro chamam a atenção do usuário para o local exato do descarte do papel necessitando somente que ele se direcione ao local indicado.

Esse modelo pode ser facilmente instalado com parafusos em lixeiras convencionais por meio de caixas frontais de acrílico sobressalentes em que pessoas com visão parcialmente comprometida pudessem ser guiadas não apenas pelo som do sistema, mas também pela iluminação dos *LEDs* instalados.

## **6.2 Modelagem do Sistema – Códigos de Barra**

Abaixo segue o descritivo do desenvolvimento do sistema modelado para leitura de códigos de barra. O sistema é composto por dois módulos (administração e principal), porém os dois trabalham com a mesma base de dados.

## **6.3 Base de Dados**

A Base de Dados é composta por 4 tabelas: TB\_CATEGORIA (tabela que armazena todos os tipos possíveis de lixeiras), TB\_PRODUTOS (tabela que armazena todos os produtos que poderão ser lidos pelo código de barras), TB\_USUARIOS (tabela que armazena todos os usuários que terão acesso ao sistema) e LOG\_SISTEMA (tabela que armazenará todas as ações realizadas dentro do sistema).

Todas as tabelas trabalham com exclusão lógica, ou seja, nenhum dado é excluído da base. Isso é feito para que sejam mantidos a integridade e histórico das informações.

### 6.3.1 Diagrama de Dados

O Diagrama de Dados abaixo apresenta a modelagem de todas as relações e entidades das tabelas do sistema idealizado (Figura 57).

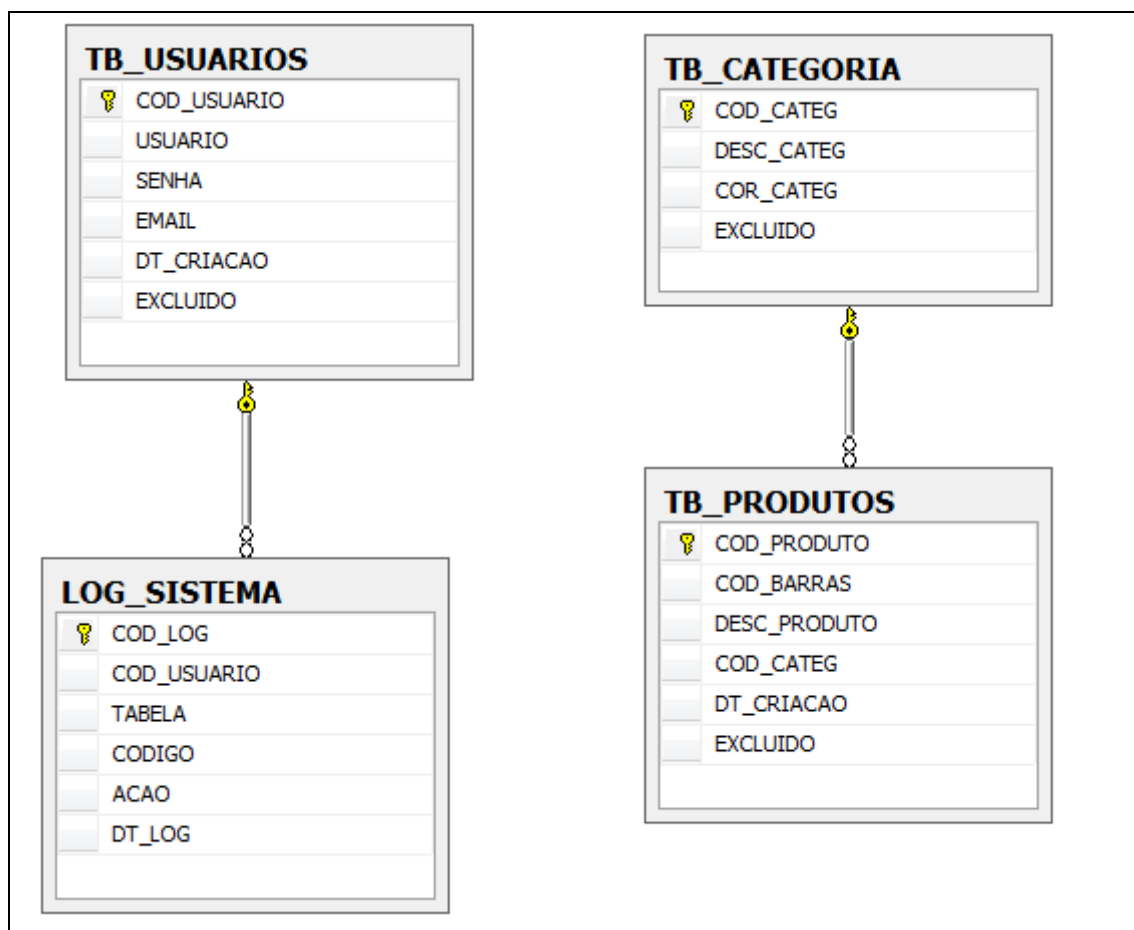


Figura 57 – Diagrama de Dados das Tabelas do Sistema

### 6.3.2 Scripts da criação da Base de Dados

#### Criando a base de dados

```
CREATE DATABASE DBSISTEMA;
```

#### Criando a tabela de usuarios:

```
CREATE TABLE TB_USUARIOS (  
    COD_USUARIO          INT IDENTITY(1,1),  
    USUARIO              VARCHAR(50),  
    SENHA                VARCHAR(12),  
    EMAIL                VARCHAR(100),  
    DT_CRIACAO           DATETIME,  
    EXCLUIDO             VARCHAR(1)  
);  
  
ALTER TABLE TB_USUARIOS ADD CONSTRAINT PK_COD_USUARIO PRIMARY KEY (COD_USUARIO);
```

#### Criando a tabela log de sistema:

```
CREATE TABLE LOG_SISTEMA (  
    COD_LOG              INT IDENTITY(1,1),  
    COD_USUARIO          INT,  
    TABELA               VARCHAR(100),  
    CODIGO               INT,  
    ACAO                 VARCHAR(MAX),  
    DT_LOG               DATETIME  
);  
  
ALTER TABLE LOG_SISTEMA ADD CONSTRAINT PK_COD_LOG PRIMARY KEY (COD_LOG);  
ALTER TABLE LOG_SISTEMA  
    ADD CONSTRAINT FK_LOGSISTEMA_TBUSUARIOS FOREIGN KEY (COD_USUARIO)  
REFERENCES TB_USUARIOS(COD_USUARIO);
```

### Criando a tabela de categorias

```
CREATE TABLE TB_CATEGORIA (  
    COD_CATEG          INT IDENTITY(1,1),  
    DESC_CATEG         VARCHAR(100),  
    COR_CATEG          VARCHAR(50),  
    EXCLUIDO           VARCHAR(1)  
);  
ALTER TABLE TB_CATEGORIA ADD CONSTRAINT PK_COD_CATEG PRIMARY KEY (COD_CATEG);
```

### Criando a tabela de produtos

```
CREATE TABLE TB_PRODUTOS (  
    COD_PRODUTO        INT IDENTITY(1,1),  
    COD_BARRAS         VARCHAR(20),  
    DESC_PRODUTO       VARCHAR(100),  
    COD_CATEG          INT,  
    DT_CRIACAO         DATETIME,  
    EXCLUIDO           VARCHAR(1)  
);  
ALTER TABLE TB_PRODUTOS ADD CONSTRAINT PK_COD_PRODUTO PRIMARY KEY (COD_PRODUTO);  
ALTER TABLE TB_PRODUTOS  
    ADD CONSTRAINT FK_TBPRODUTOS_TBCATEGORIA FOREIGN KEY (COD_CATEG)  
REFERENCES TB_CATEGORIA(COD_CATEG);
```

### 6.3.3 Carga Inicial no Sistema

#### Carga na tabela de categorias

```
INSERT INTO TB_CATEGORIA VALUES ('PAPÉIS','AZUL','N');
INSERT INTO TB_CATEGORIA VALUES ('PLÁSTICOS','VERMELHO','N');
INSERT INTO TB_CATEGORIA VALUES ('VIDROS','VERDE','N');
INSERT INTO TB_CATEGORIA VALUES ('METAIS','AMARELO','N');
INSERT INTO TB_CATEGORIA VALUES ('MADEIRAS','PRETO','N');
INSERT INTO TB_CATEGORIA VALUES ('RESÍDUOS PERIGOSOS','LARANJA','N');
INSERT INTO TB_CATEGORIA VALUES ('RESÍDUOS AMBULATORIOS E SERVIÇOS DE
SAÚDE','BRANCO','N');
INSERT INTO TB_CATEGORIA VALUES ('RESÍDUOS RADIOATIVOS','ROXO','N');
INSERT INTO TB_CATEGORIA VALUES ('RESÍDUOS ORGÂNICOS','MARROM','N');
INSERT INTO TB_CATEGORIA VALUES ('RESÍDUOS GERAIS NÃO RECICLÁVEL','CINZA','N');
```

#### Carga na tabela de produtos

```
INSERT INTO TB_PRODUTOS VALUES('7898386200083','ÁGUA MINERAL FRESCCA NATURAL SEM
GÁS',2,GETDATE(),'N');
INSERT INTO TB_PRODUTOS VALUES('7896052600731','ÁGUA MINERAL SCHIN NATURAL SEM
GÁS',2,GETDATE(),'N');
INSERT INTO TB_PRODUTOS VALUES('7891321042489','CADERNO 1 MATÉRIA WORLD CLASS 96
FOLHAS',1,GETDATE(),'N');
```

#### Carga na tabela de usuarios

```
INSERT INTO TB_USUARIOS VALUES
('WELLINGTON','acesso','WELPSILVA@HOTMAIL.COM',GETDATE(),'N');
INSERT INTO TB_USUARIOS VALUES ('RENATO','acesso','',GETDATE(),'N');
INSERT INTO TB_USUARIOS VALUES ('GILMAR','acesso','',GETDATE(),'N');
INSERT INTO TB_USUARIOS VALUES ('ANDREIA','acesso','',GETDATE(),'N');
```

A Figura 58 apresenta o fluxograma da seqüência lógica do sistema incluindo a parte de trabalhos futuros, onde é apresentada a visão geral de seu funcionamento com a opção do leitor de código de barras.

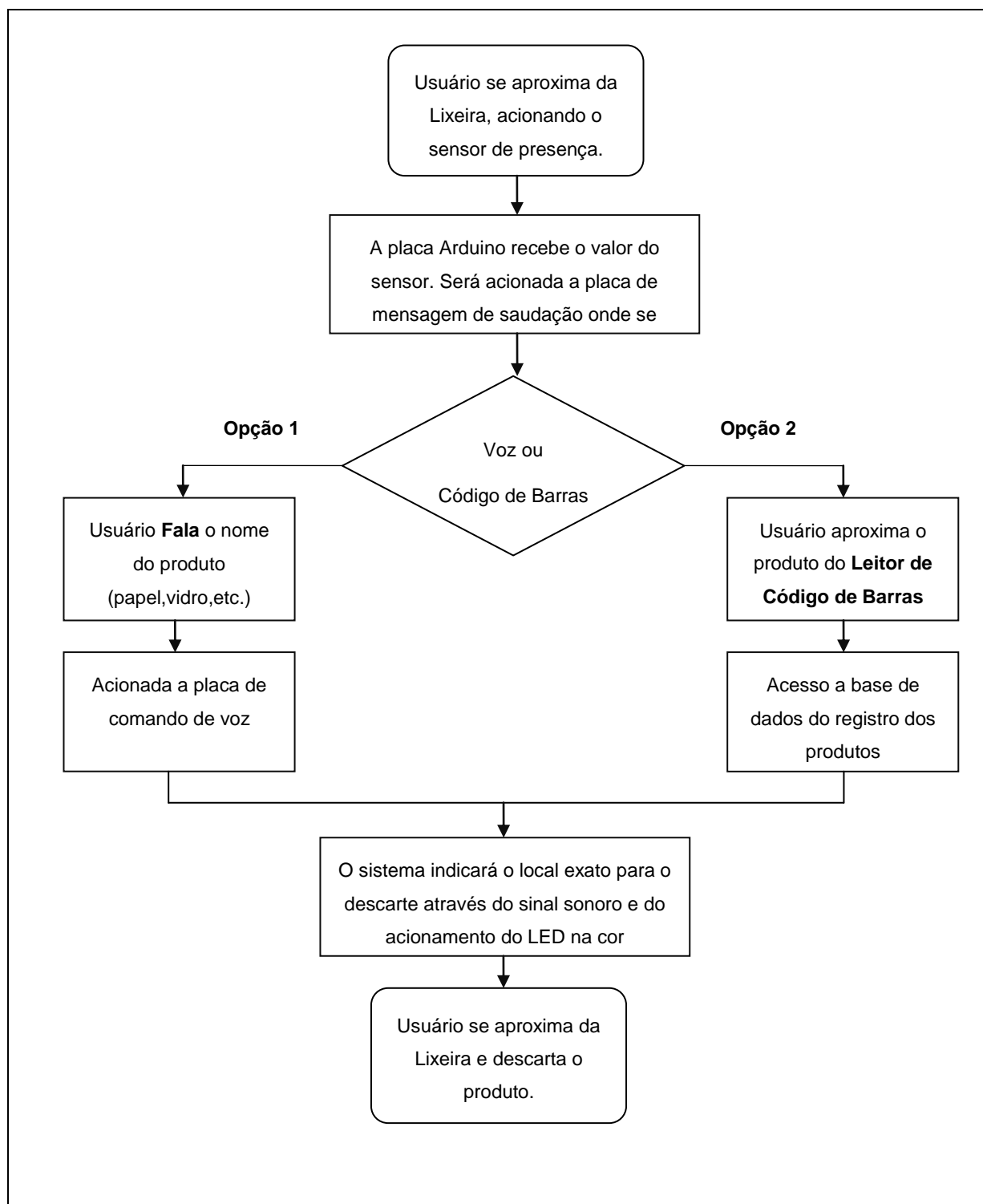


Figura 58 – Fluxograma de Funcionamento do Sistema Geral



## 7 LIMITAÇÕES E TRABALHOS FUTUROS

Uma dificuldade encontrada durante a realização do trabalho foi à localização de componentes no mercado interno que possibilitassem o desenvolvimento de um modelo de baixo custo e que atendessem às necessidades propostas. Além disso, quando se optou por importar tais componentes, os prazos de entrega não foram cumpridos conforme pré-estabelecidos pelos fornecedores.

Futuramente, pretende-se disponibilizar o projeto para ser testado com pessoas com deficiência visual. Elas contribuirão para que sejam realizados ajustes no projeto. Intenciona-se ainda que crianças não alfabetizados utilizem o sistema para que as adaptações necessárias sejam realizadas.

Também em continuidade ao presente trabalho, a placa *Arduino* pode ser alimentada por uma bateria 5 Volts, pois no momento ela está sendo alimentada através da porta USB do computador que controla o sistema.

Poderá haver a confecção de uma caixa apropriada para proteção das placas e conectores que compõem o sistema, bem como a possibilidade de abertura automática da tampa por meio de sensores indutivos e capacitivos dependendo do modelo de lixeira escolhido para instalação do sistema.

Pode ser ainda instalado, junto aos *LEDs* um sistema de aviso sonoro com sensor de aproximação para que o deficiente visual possa ter a certeza de que esta perto do recipiente correto, pois conforme a sua aproximação da lixeira o sinal sonoro torna-se mais intenso.

Além disso, é possível a instalação de um leitor de código de barras que facilitaria no descarte de todas as embalagens de produtos industrializados que contivessem tal código. Assim, quando uma pessoa aproximasse da lixeira o sensor PIR acionaria a placa de Gravação de Voz, de onde se poderia ouvir uma mensagem informando a possibilidade de auxílio de descarte por meio de um leitor de código de barras. A identificação do tipo de produto pode ser realizada por meio de acesso a um Banco de Dados que foi modelado e apresentado nesse trabalho.

## 8 CONCLUSÕES

O projeto foi desenvolvido através de *software* e *hardware Arduino*. Isso nos permitiu desenvolver, juntamente com outros componentes eletrônicos, um controlador lógico programável que atuasse diretamente no controle de todo o sistema de sensores, *LEDs*, *Buzzers*, placa de mensagem e placa de comando de voz, e que também que auxiliasse a pessoa com deficiência no descarte em lixeiras de coleta seletiva.

O sistema pode identificar a presença do usuário por meio de sensor infravermelho e iniciar todo o processo de interação através da placa de mensagem de voz, de onde são informados ao usuário os passos para início de todo o processo de descarte. Ao reconhecer o comando dado pelo usuário, o sistema irá apresentar através do *LED* de auto brilho e de um sinal sonoro, o local exato para a finalização do processo.

A escolha de lixeiras transparentes e de *LEDs* de auto brilho colaborou para o auxílio as pessoas com visão parcialmente comprometida, já que uma vez escolhida a opção de descarte o *LED* irá acender dentro da própria lixeira e sua luz de auto brilho acenderá por completo iluminando todo o recipiente. O *LED* de auto brilho estará instalado, para demonstração, na parte interna e sua luz será refletida por todo o interior da lixeira. O projeto desenvolvido permite ainda sua instalação em lixeiras convencionais independente do tamanho e do volume.

Foi desenvolvido um projeto não apenas de uma simples lixeira para material reciclável, mas uma lixeira interativa que se comunicará com as pessoas através de sensores e comandos de voz.

Portanto, o projeto visa atuar de forma educativa em que adultos e principalmente crianças seriam estimuladas desde os primeiros anos de vida ao hábito da reciclagem, a conscientização e bem o respeito ao meio ambiente.

## REFERÊNCIAS

SANTOS e col. Projeto R<sup>3</sup>- Reciclar, Reduzir e Reutilizar - Faculdade de Zootecnia e Engenharia de Alimentos / Departamento de Engenheiro de Alimentos – USP. 2011. Disponível em: [http://www.usp.br/ccps/gestaoindustrial/projetos\\_finais/projeto\\_r3.pdf](http://www.usp.br/ccps/gestaoindustrial/projetos_finais/projeto_r3.pdf)

NUNES e col. AVEI – Ambiente Virtual de Ensino Inteligente - Universidade Regional Integrada - RS. 2001. - Disponível em; <<http://www.urisan.tcche.br/~avei/4.pdf>>

ULRICH e col. The GuideCane - Applying Mobile Robot - Technologies to Assist the Visually Impaired - Department of Mechanical Engineering at the University of Michigan. 2001. Disponível em:<<http://web.eecs.umich.edu/~johannb/paper72.pdf>>

ROSA - Bengala de apoio a cegos com detecção de buracos – Departamento de Electrónica, Telecomunicações e Informática - Universidade de Aveiro Portugal. 2009. Disponível em:  
<<http://www.ieeta.pt/~vieira/Curriculum/MyPublicPublications/Rosa2009.pdf>>

FERNANDES. Dissertação de Mestrado em Informática - Orientação de Cegos usando Visão por Computador - Universidade De Trás-Os-Montes e Alto Douro – Portugal 2010. Disponível em:  
<[http://repositorio.utad.pt/bitstream/10348/587/1/msc\\_hrmfernandes.pdf](http://repositorio.utad.pt/bitstream/10348/587/1/msc_hrmfernandes.pdf)>

PEREIRA e col. - Substituição Sensorial para auxílio à Mobilidade de Deficientes Visuais via Eletro Estimulação Tátil - Politécnica da USP. 2004. Disponível em:<[http://www.gpec.ucdb.br/artigos/pereira\\_iberdiscap2004.pdf](http://www.gpec.ucdb.br/artigos/pereira_iberdiscap2004.pdf)>

ARDUINO – Disponível em Site Oficial: <<http://arduino.cc/en/>> - Acesso em 15 mar, 2013.

Sasaki R. K. In: VIVARTA, Veet (coord.). Mídia e deficiência. Brasília: Andi/Fundação Banco do Brasil, 2003, p. 160-165. Acesso em 15 mar, 2013.

CONAMA – Disponível em: Site Oficial: <http://www.mma.gov.br/port/conama> - Acesso em 28 mar, 2013.

## APÊNDICE A

A seguir é apresentado o código para programação do *LED / Buzzer*.

```
int buzzerGreenPin = 13; //ligar led verde e o sinal sonoro no pino 13
int buzzerYellowPin = 12; //ligar led amarelo e o sinal sonoro no pino 12
int buzzerRedPin = 11; //ligar led vermelho e o sinal sonoro no pino 11
int buzzerBluePin = 10; //ligar led azul e o sinal sonoro no pino 10
byte com = 0; //comandos recebidos pelo modulo de voz

void setup()
{
  Serial.begin(9600);
  //definição das LEDs e do sinal sonoro como saída
  pinMode(buzzerGreenPin, OUTPUT);
  pinMode(buzzerYellowPin, OUTPUT);
  pinMode(buzzerRedPin, OUTPUT);
  pinMode(buzzerBluePin, OUTPUT);
  delay(2000);
  //comandos de configuração do modulo de voz
  Serial.write(0xAA);
  Serial.write(0x37); // configura o módulo para responder em modo compacto
  delay(1000);
  //comandos de configuração do modulo de voz
  Serial.write(0xAA);
  Serial.write(0x21); //importa o grupo 1
  //pre definição do estado das LEDs e do sinal sonoro
  digitalWrite(buzzerGreenPin,LOW);
  digitalWrite(buzzerYellowPin,LOW);
  digitalWrite(buzzerRedPin,LOW);
  digitalWrite(buzzerBluePin,LOW);
}

void loop() //inicio do programa principal função de recorrência
```

```

{

while(Serial.available())
{
com = Serial.read(); //recebe o byte referente ao comando
switch(com)
{

//a cada comando pelo módulo o Aduino irá ligar o sinal sonoro e piscar as LEDs
case 0x11:
tone(buzzerGreenPin,3000); //ligando o buzzer com uma frequencia de 3000 hz
delay(1250); //tempo que o buzzer fica ligado
noTone(buzzerGreenPin); //desligando o buzzer
delay(500); //tempo que o buzzer fica desligado
tone(buzzerGreenPin,3000);
delay(1250);
noTone(buzzerGreenPin);
delay(500);
tone(buzzerGreenPin,3000);
delay(1250);
noTone(buzzerGreenPin);
delay(500);
tone(buzzerGreenPin,3000);
delay(1250);
noTone(buzzerGreenPin);
delay(500);
tone(buzzerGreenPin,3000);
delay(1250);
noTone(buzzerGreenPin);
break;

case 0x12:
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);

```

```
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
delay(500);
tone(buzzerYellowPin,3000);
delay(1250);
noTone(buzzerYellowPin);
break;

case 0x13:
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
```

```
tone(buzzerRedPin,3000);  
delay(1250);  
noTone(buzzerRedPin);  
break;  
  
case 0x14:  
tone(buzzerBluePin,3000);  
delay(1250);  
noTone(buzzerBluePin);  
delay(500);  
tone(buzzerBluePin,3000);  
delay(1250);  
noTone(buzzerBluePin);  
delay(500);  
tone(buzzerBluePin,3000);  
delay(1250);  
noTone(buzzerBluePin);  
delay(500);  
tone(buzzerBluePin,3000);  
delay(1250);  
noTone(buzzerBluePin);  
delay(500);  
tone(buzzerBluePin,3000);  
delay(1250);  
noTone(buzzerBluePin);  
break;  
}  
}  
}
```

## APÊNDICE B

A seguir é apresentado o código fonte contendo toda a programação do sistema.

```
/*declarando as variaveis de presença e ISD1820*/
int PIRPin = 9;           //Pino Out do PIR(Leitura de valores)
int PIR = 0;             //Variável que representa o valor fornecido pelo sensor PIR
byte pinoPlayE = 2;      //pino ligado ao playe do ISD1820

/*declarando as variaveis dos buzzers */
int buzzerGreenPin = 13; //ligar led verde e o sinal sonoro no pino 13
int buzzerYellowPin = 12; //ligar led amarelo e o sinal sonoro no pino 12
int buzzerRedPin = 11; //ligar led vermelho e o sinal sonoro no pino 11
int buzzerBluePin = 10; //ligar led azul e o sinal sonoro no pino 10
byte com = 0; //comandos recebidos pelo modulo de voz

void setup()
{

//sensor
pinMode(PIRPin, INPUT);

//player
pinMode(pinoPlayE, OUTPUT);

Serial.begin(9600);
//definição das leds e do sinal sonoro como saída
pinMode(buzzerGreenPin, OUTPUT);
pinMode(buzzerYellowPin, OUTPUT);
pinMode(buzzerRedPin, OUTPUT);
pinMode(buzzerBluePin, OUTPUT);
delay(2000);
```



```

//comandos de configuração do modulo de voz
Serial.write(0xAA);
Serial.write(0x37); // configura o módulo para responder em modo compacto
delay(1000);
//comandos de configuração do modulo de voz
Serial.write(0xAA);
Serial.write(0x21); //importa o grupo 1
//pre definição do estado das leds e do sinal sonoro
digitalWrite(buzzerGreenPin,LOW);
digitalWrite(buzzerYellowPin,LOW);
digitalWrite(buzzerRedPin,LOW);
digitalWrite(buzzerBluePin,LOW);
}

//procedimento para ligar os leds e sinal sonoro
void ligabuzz(char buzzer, int indice) {
  int i = 0;

  for(int i=0; i < indice ; i++) {
    switch(buzzer) {
      case 'G' :
        tone(buzzerGreenPin,3000); //ligando o buzzer com uma frequencia de 3000 hz
        delay(1250); //tempo que o buzzer fica ligado
        noTone(buzzerGreenPin); //desligando o buzzer
        delay(500);
        break;

      case 'Y':
        tone(buzzerYellowPin,3000);
        delay(1250);
        noTone(buzzerYellowPin);
        delay(500);
        break;

      case 'R':

```

```

tone(buzzerRedPin,3000);
delay(1250);
noTone(buzzerRedPin);
delay(500);
break;

case 'B':
tone(buzzerBluePin,3000);
delay(1250);
noTone(buzzerBluePin);
delay(500);
break;
}
}
}

//procedimento para habilitar os leds e sinais sonoros qdo for falado o tipo de produto
void serial(){

while(Serial.available())
{
com = Serial.read(); //recebe o byte referente ao comando
switch(com)
{
//a cada comando pelo módulo o Aduino irá ligar o sinal sonoro e piscar as leds
case 0x11:
ligabuzz('G',5);
break;

case 0x12:
ligabuzz('Y',5);
break;

case 0x13:
ligabuzz('R',5);

```

```

break;

case 0x14:
  ligabuzz('B',5);
break;

  }
}
}

//procedimento para habilitar o player quando for ativado o sensor
void presenca(){

  PIR = digitalRead(PIRPin);  // Faz leitura do OUT do senso PIR e joga o valor lido na variável: "PIR"
  if (PIR == HIGH) {          // Testa se a variável "PIR" é alta(5v), caso SIM: executa uma tarefa Caso
  NÃO pular para else
    digitalWrite(pinoPlayE,HIGH); //toca o sinal de alerta no ISD1820
    delay(100);
  }
  digitalWrite(pinoPlayE,LOW); //desliga aviso de alerta no ISD1820
  delay(500);

}

void loop() //inicio do programa principal função de recorrência
{
  presenca();
  digitalWrite(PIRPin,LOW);
  delay(1000);
  serial();
}

```

## **FOLHA DE APROVAÇÃO DO PROJETO**

RENATO ALEXANDRE PAVIN  
GILMAR CORRÊA DE OLIVEIRA  
WELLINGTON PIMENTEL DA SILVA

### **SISTEMA PARA AUXILIAR PESSOAS COM DEFICIÊNCIA VISUAL NO DESCARTE CORRETO DE ITENS EM LIXEIRAS DE COLETA SELETIVA**

Monografia apresentada à Diretoria dos Cursos de Informática da Universidade Nove de Julho como requisito parcial para obter a Graduação de Bacharel em Ciência da Computação, sob a orientação da Prof<sup>a</sup>. Dra. Andréia Miranda Domingues.

Data: \_\_\_\_/\_\_\_\_/\_\_\_\_

\_\_\_\_\_  
Assinatura do professor orientador

OBSERVAÇÕES: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_