



Centro Federal de Educação Tecnológica de Minas Gerais
Unidade Contagem

Relatório final de atividades

Editais do programa institucional de bolsas de Iniciação Científica Júnior

Tecnologia assistiva para deficientes visuais

Aluna: Ana Clara Amorim Andrade
Orientador: Nelson Alexandre Estevão

Contagem,
Março de 2017

Sumário

1. Resumo.....	3
2. Introdução.....	4
2.1. Conceito de Tecnologia Assistiva.....	4
2.2. Categorias Tecnologia Assistiva.....	4
2.3. Tecnologia Assistiva para deficientes visuais.....	6
3. Revisão Literária.....	8
4. Metodologia.....	12
4.1. Bengala Microcontrolada.....	12
4.2. Aplicativo SecondEye.....	18
4.3. Interação aplicativo SecondEye com ESP8266.....	23
4.4. Interação ESP8266 com bengala eletrônica.....	28
5. Resultados.....	31
6. Referências Bibliográficas.....	32

1. Resumo

O objetivo deste trabalho é criar recursos que auxiliem na vida de deficientes visuais, através de pesquisas na área da tecnologia assistiva.

Com base nas pesquisas realizada nota-se extrema dificuldade desses deficientes nos âmbitos da leitura, escrita, locomoção, o que impossibilita a total inclusão na sociedade. Para resolver este problema é necessário que cada vez mais recursos sejam investidos na área de pesquisa de TA.

Alguns recursos tecnológicos já estão disponíveis no mercado e através dos mesmos nota-se uma facilidade ao acesso ao celular, por meio do recurso "Acessibilidade" do Android. Através disso decidiu-se que o projeto iria ser sobre um aplicativo, que melhoraria a questão da locomoção dos deficientes, promovendo maior inclusão social.

Palavras chaves: tecnologia assistiva, deficientes visuais, inclusão social.

2. Introdução

2.1 Conceito de Tecnologia Assistiva

A tecnologia assistiva é um termo que veio para auxiliar um grupo de pessoas na sociedade, podendo ser definido como "uma ampla gama de equipamentos, serviços, estratégias e práticas concebidas e aplicadas para minorar os problemas encontrados pelos indivíduos com deficiências" (*Cook e Hussey • Assistive Technologies: Principles and Practices • Mosby – Year Book, Inc., 1995*).

Surgiu nos Estados Unidos em 1988, como Assistive Technology, sendo parte do elemento jurídico dentro da legislação norte-americana conhecida como Public Law 100-407 e foi renovado em 1998 como Assistive Technology Act de 1998 (P.L. 105-394, S.2432). Sendo que ele compõe, juntamente com outras leis, o ADA - American with Disabilities Act, que é uma lei de direitos civis que proíbe a discriminação baseada na deficiência.

No Brasil, o Comitê de Ajudas Técnicas - CAT, instituído pela PORTARIA N° 142, DE 16 DE NOVEMBRO DE 2006 propõe o seguinte conceito para a tecnologia assistiva: "Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social" (*ATA VII - Comitê de Ajudas Técnicas (CAT) - Coordenadoria Nacional para Integração da Pessoa Portadora de Deficiência (CORDE) - Secretaria Especial dos Direitos Humanos - Presidência da República*).

2.2 Categorias Tecnologia Assistiva

Os recursos de Tecnologia Assistiva (TA) são divididos em categorias de acordo com a finalidade a quais se destinam.

Existem várias classificações de TA que foram surgindo com finalidades distintas, a que segue abaixo foi escrita em 1998 por José Tonolli e Rita Bersch, sendo esta uma classificação com finalidade didática, levando em conta a existência de recursos e serviços de cada categoria; foi desenhada com base em outras classificações utilizadas em bancos de dados de TA e especialmente a partir da formação dos autores no Programa de Certificação em Aplicações da Tecnologia Assistiva – ATACP da California State University Northridge, College of Extended Learning and Center on Disabilities. Recentemente esta classificação foi utilizada pelo Ministério da Fazenda; Ciência, Tecnologia e Inovação e pela Secretaria Nacional de Direitos Humanos da Presidência da República na publicação da Portaria Interministerial N° 362, de 24 de Outubro de 2012 que trata sobre a linha de crédito subsidiado para aquisição de bens e serviços de Tecnologia Assistiva destinados às pessoas com deficiência e sobre o rol dos bens e serviços.

Categorias de TA	
Auxílios para a vida diária e vida prática	Materiais e produtos que favorecem desempenho autônomo e independente em tarefas rotineiras ou facilitam o cuidado de pessoas em situação de dependência de auxílio, nas atividades como se alimentar, cozinhar, vestir-se, tomar banho e executar necessidades pessoais.
CAA - Comunicação Aumentativa e Alternativa	Destinada a atender pessoas sem fala ou escrita funcional ou em defasagem entre sua necessidade comunicativa e sua habilidade em falar, escrever e/ou compreender
Recursos de acessibilidade ao computador	Conjunto de hardware e software especialmente idealizado para tornar o computador acessível a pessoas com privações sensoriais (visuais e auditivas), intelectuais e motoras. Inclui dispositivos de entrada (mouses, teclados e acionadores diferenciados) e dispositivos de saída (sons, imagens, informações táteis).
Sistemas de controle de ambiente	Através de um controle remoto as pessoas com limitações motoras, podem ligar, desligar e ajustar aparelhos eletroeletrônicos como a luz, o som, televisores, ventiladores, executar a abertura e fechamento de portas e janelas, receber e fazer chamadas telefônicas, entre outros projetos.
Projetos arquitetônicos para acessibilidade	Projetos de edificação e urbanismo que garantem acesso, funcionalidade e mobilidade a todas as pessoas, independentemente de sua condição física e sensorial.
Órteses e próteses	Próteses são peças artificiais que substituem partes ausentes do corpo. Órteses são colocadas junto a um segmento corpo, garantindo-lhe um melhor posicionamento, estabilização e/ou função.
Adequação Postural	Seleção de recursos que garantam posturas alinhadas, estáveis, confortáveis e com boa distribuição do peso corporal.

Auxílios de mobilidade	Equipamentos ou estratégia utilizada na melhoria da mobilidade pessoal, como bengalas, muletas, andadores, carrinhos.
Auxílios para ampliação da função visual e recursos que traduzem conteúdos visuais em áudio ou informação tátil.	São exemplos: Auxílios ópticos, lentes, lupas manuais e lupas eletrônicas; os softwares ampliadores de tela. Material gráfico com texturas e relevos, mapas e gráficos táteis, software OCR em celulares para identificação de texto informativo, etc.
Auxílios para melhorar a função auditiva e recursos utilizados para traduzir os conteúdos de áudio em imagens, texto e língua de sinais.	Auxílios que incluem vários equipamentos (infravermelho, FM), aparelhos para surdez, sistemas com alerta tátil-visual, celular com mensagens escritas e chamadas por vibração, entre outros.
Mobilidade em veículos	Acessórios que possibilitam uma pessoa com deficiência física dirigir um automóvel.
Esporte e Lazer	Recursos que favorecem a prática de esporte e participação em atividades de lazer.

Tabela 2.2.1, disponível em:

http://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf

Para desenvolver esses recursos, muitas vezes são necessários profissionais de diversas áreas, para se dedicar ao estudo da necessidade do deficiente, da produção do projeto e da verificação da eficácia do mesmo. Portanto, podem aparecer profissionais como: fisioterapeutas, fonoaudiólogos, educadores, psicólogos, enfermeiros, médicos, engenheiros, arquitetos, designers e técnicos de muitas outras especialidades.

2.3 Tecnologia Assistiva para deficientes visuais

De acordo com dados do IBGE (2010) do total da população brasileira, 23,9% (45,6 milhões de pessoas) declararam ter algum tipo de deficiência. Entre as deficiências declaradas, a mais comum foi a visual, chegando a 35,7 milhões de pessoas, concluindo-se que 18,8% dos entrevistados afirmam ter dificuldades para enxergar, mesmo com óculos ou lentes de contato. Em segundo lugar ficou a deficiência motora, com 7% dos brasileiros, a deficiência auditiva (5,1%) e a deficiência mental ou intelectual foi declarada por mais de 2,6 milhões de brasileiros.

De acordo com o Decreto nº 3.298/99 e o Decreto nº 5.296/04, conceitua-se como deficiência visual:

- Cegueira – na qual a acuidade visual é igual ou menor que 0,05 no melhor olho, com a melhor correção óptica;
- Baixa Visão – significa acuidade visual entre 0,3 e 0,05 no melhor olho, com a melhor correção óptica;
- Os casos nos quais a somatória da medida do campo visual em ambos os olhos for igual ou menor que 60°;
- Ou a ocorrência simultânea de quaisquer das condições anteriores.

Segundo dados do IBGE de 2010, no Brasil, das mais de 6,5 milhões de pessoas com alguma deficiência visual:

- 528.624 pessoas são incapazes de enxergar (cegos);
- 6.056.654 pessoas possuem baixa visão ou visão subnormal (grande e permanente dificuldade de enxergar);

Outros 29 milhões de pessoas declararam possuir alguma dificuldade permanente de enxergar, ainda que usando óculos ou lentes.

De acordo com essas estatísticas surge a necessidade de projetos/produtos que auxiliem esses deficientes, visando melhorar a qualidade de vida dos mesmos.

3. Revisão Literária

Visto a enorme dificuldade desses deficientes, surge a necessidade da criação de produtos e de pesquisas nesta área. Para decidir qual seria o objetivo específico do projeto BIC, será feita uma revisão literária sobre tecnologia assistiva para deficientes visuais e este é o assunto tratado neste tópico.

Após realizada as pesquisas foram encontrados, no site *Casadaptada*, alguns produtos, como:

- BlindTool

É um aplicativo gratuito disponível na plataforma Android, foi criado pelo cientista da computação Joseph Cohen, pesquisador da Universidade de Massachusetts.

O aplicativo reconhece objetos, através de uma rede neural artificial capaz de reconhecer as semelhanças dos objetos que o usuário do aplicativo deve apontar através da câmera, com as imagens armazenadas no banco de dados, buscando semelhanças.

- Be my eyes

É um aplicativo que permite que pessoas com problemas visuais possam ler etiquetas, bulas de remédio, contas, etc. Funciona da seguinte forma, ao se cadastrar pode-se escolher a opção voluntário ou a de quem precisa de ajuda, quem assume a posição de voluntário passa a receber imagens e verbalizá-las para os deficientes visuais.

Disponível gratuitamente no sistema IOS.

- Color ID

O aplicativo recebe as imagens da câmera do celular e através disso verbaliza as cores que estão sendo visualizadas, ajudando pessoas com baixa visão a saberem qual a cor da roupa que estão vestindo, por exemplo.

Este aplicativo está disponível gratuitamente para Iphone e Android.

- Ariadne GPS

É um GPS desenvolvido especificamente para cegos, basta passar a mão no mapa que o aplicativo verbaliza o local em que a pessoa se encontra e oferece coordenadas para chegar ao destino, sendo também capaz de dizer em qual parada de ônibus ele deve descer e o celular vibra caso seja preciso atravessar um cruzamento.

Disponível na plataforma IOS, em vários idiomas, por US\$ 5,99.

- UBook

É uma audioteca com mais de 1000 títulos no catálogo, o plano de assinatura é no valor mensal de R\$ 18,90, sendo uma ideia muito parecida com serviços de streaming de filmes e ótima para quem tem dificuldades de leitura.

Pode ser acessada pelo computador e pelo smartphone Ios ou Android.

- CPqD Alcance

É um aplicativo de celular que visa facilitar o acesso do deficiente visual ao mesmo, é basicamente um guia, com narração automática da tela e com auxílio para quase todas as funções do celular.

É um aplicativo brasileiro disponível gratuitamente para Android.

Além desses aplicativos o próprio smartphone possui a opção de acessibilidade, que pode ser facilmente encontrada nas configurações, como mostra as imagens abaixo:



Figura 3.1- Print de tela de celular mostrando o recurso de acessibilidade

Esse recurso possui integração para deficientes visuais e além disso para deficiência auditiva e motora.



Figura 3.2- Print de tela de celular mostrando as categorias do recurso de acessibilidade

Após acionar o recurso da visão, o celular passa a ser todo verbalizado, a cada clique surge um áudio explicando a opção que foi selecionada, se mostrando eficaz na interação com deficientes visuais, ademais possui recursos para quem não possui a visão 100% comprometida.



Figura 3.2- Print de tela de celular com a assistente de voz ativada

Após a pesquisa, nota-se que o recurso do Google gera a acessibilidade dos deficientes visuais aos smartphones que possuem o sistema Android, portanto, conclui-se que a criação de um aplicativo, seria algo em que não se encontraria dificuldade de acesso por parte desses deficientes.

4. Metodologia

4.1 Bengala Microcontrolada

No ano de 2016, foi criado, com parceria com Júlia Fagundes Gomes e Nayôto Saulo, e apresentado na exposição do META a Bengala Eletrônica Microcontrolada, que deu início a essa pesquisa.

A bengala foi criada como forma de melhorar a caminhada de deficientes visuais em ambientes desconhecidos, visando ajudar na detecção de objetos pelo caminho, sem a necessidade do toque, através do sensor ultrassônico.

Este produto, é composto de:

- 1 sensor ultrassônico Hc Sr04
- 1 microcontrolador Arduino Uno
- 1 Bateria recarregável
- 2 motores vibracall

Para começar a explicação de seu funcionamento, é importante entender o princípio de funcionamento do sensor ultrassônico.

Este sensor é o detector de objetos da bengala eletrônica, ele utiliza sinais ultrassônicos (40Khz, que é uma frequência acima da capacidade de audição do ouvido humano, que é de 20Khz) para detectar obstáculos. Podendo medir distâncias entre 2cm e 4m.

Ele possui 4 pinos: VCC (5V), Trigger, Echo e GND. Sendo ideal para trabalhar com o Arduino, pois consome baixa corrente (15mA).

Seu funcionamento pode ser dividido em três etapas:

1. É enviado um sinal com duração de 10 us (microsegundos) ao pino trigger, indicando que a medição terá início
2. Automaticamente, o módulo envia 8 pulsos de 40 KHz e aguarda o retorno do sinal pelo receptor
3. Caso haja um retorno de sinal (em nível HIGH), determinamos a distância entre o sensor e o obstáculo utilizando a seguinte equação:

$$Distancia = (pulso\ em\ nível\ alto \times velocidade\ do\ som\ (340m/s)) / 2$$

Sendo essas etapas mostradas graficamente abaixo:

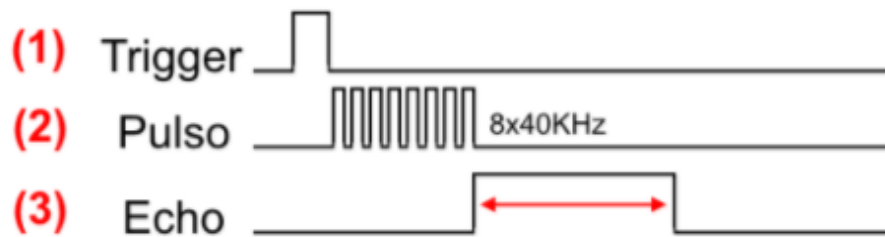


Figura 4.1.1- Diagrama de tempo HC-SR04

Para melhor compreensão, podemos simplificar o funcionamento, que ocorre basicamente da seguinte forma, um sinal ultrassônico é enviado pelo pino Trigger (emissor), o sensor aguarda o retorno do sinal pelo pino Echo (Receptor), com base nesse tempo é calculada a distância em que o objeto se encontra.

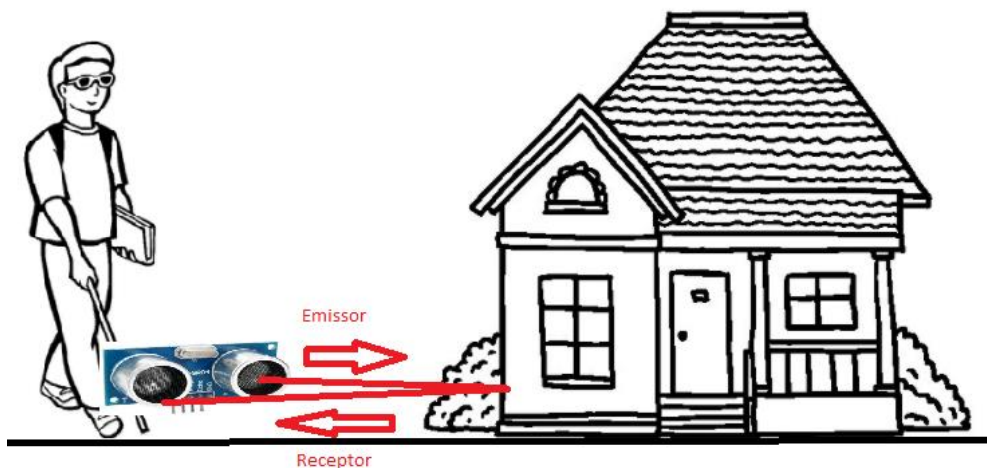


Figura 4.1.2- Princípio de funcionamento do sensor ultrassônico.

Por seguinte, a explicação do cérebro do processo, o microcontrolador Arduino Uno, tudo que acontece é comandado por ele.

O Arduino Uno é uma placa de microcontrolador baseado no ATmega328. Tem 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; A sua programação é feita com o Software Arduino e ele é conectado ao computador via USB.



Figura 4.1.3- Microcontrolador Arduino UNO.

A alimentação do Arduino é feita por uma bateria recarregável de celular (power bank), a escolha foi devido ao baixo custo desta bateria, pela facilidade de recarregar a mesma, que pode ser recarregada com carregadores de celulares, rapidamente e pela durabilidade, esta bateria consegue alimentar o Arduino praticamente o dia todo.

```
#include <Ultrasonic.h>

#define pino_trigger 12
#define pino_echo    13
#define pino_pwm     3
#define pino_forte   11
```

Figura 4.1.4- Início da programação da Bengala Eletrônica

O primeiro passo é incluir a biblioteca *Ultrasonic.h* que faz as leituras dos comandos do sensor ultrassônico, posteriormente são definidas os nomes de cada pino do Arduino que será usado e na linha *Ultrasonic* quais pinos serão responsáveis por receber os dados do Sensor, que é então ligado diretamente aos mesmos.

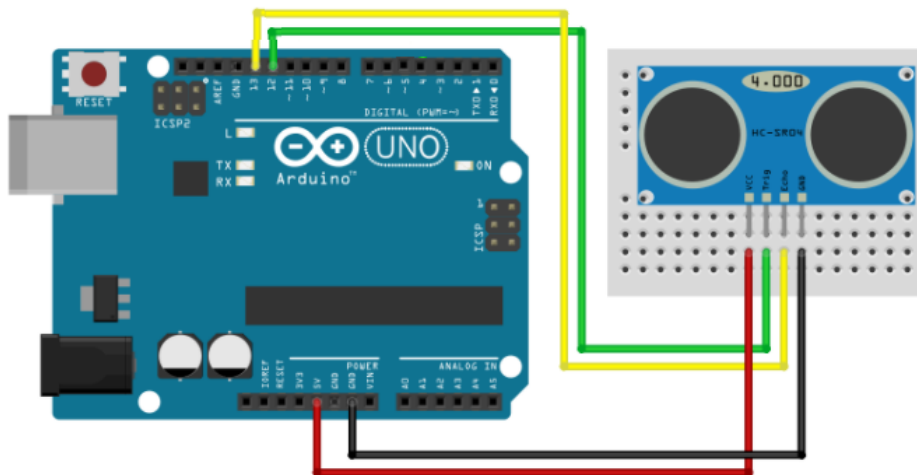


Figura 4.1.5- Conexão Arduino e HC-SR04

A conexão entre esses dois componentes é feita como mostra a figura acima.

A programação da Bengala foi dividida em: "void setup" e "void loop". O void setup é uma configuração inicial, que é feita somente 1 vez quando o Arduino é ligado, enquanto o void loop fica acontecendo o tempo todo enquanto a bengala está ligada. Isso gera uma diferença nos comandos que são colocados em cada um, que ficará mais clara abaixo:

```
void setup()
{
  Serial.begin(9600);
  Serial.print("ANA CLARA AMORIM");
  Serial.print("\n\n");
  Serial.print("BENGALA ELETRONICA MICROCONTROLADA");
  Serial.print("\n\nLendo dados do sensor...\n\n");
}
```

Figura 4.1.6- Setup da programação do Arduino.

Nessa imagem são colocados comandos que só precisam de ser rodados uma única vez, como o *Serial.begin* que define a taxa de dados em bits por segundo (baud) para transmissão de dados em série, os comandos de *Serial.print* servem apenas para mostrar o texto na tela de teste.

```

void loop()
{
  float cm;
  long microsec = ultrasonic.timing();

  cm = ultrasonic.convert(microsec, Ultrasonic::CM);

  Serial.print("Distancia em cm: ");
  Serial.print(cm);

  delay(500); //tempo para estabilizar leitura

```

Figura 4.1.7- Loop da programação do Arduino

O *void Loop* já possui uma quantidade maior de comandos, que serão divididos nessa imagem e nas que seguem abaixo. São definidas duas variáveis e que a leitura feita pelo sensor ultrassônico seja convertida para centímetros, em *Serial.print* também é definido que essa distância em centímetros apareça no monitor serial do Arduino, até aí esses comandos são para os testes para verificar a eficácia do programa. Enquanto o comando *delay*, define que a leitura dos dados do sensor será feita de 500 em 500ms.

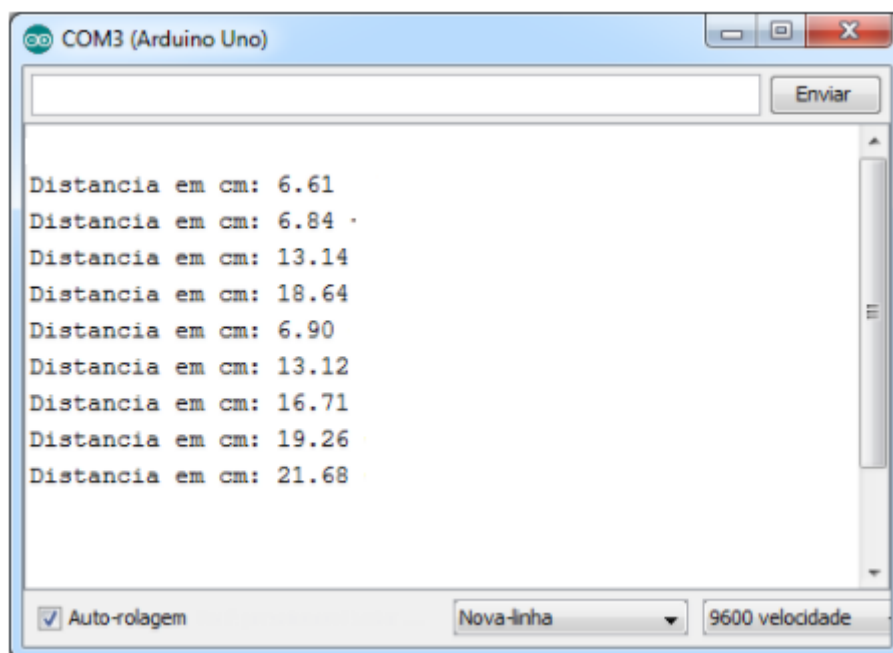


Figura 4.1.8- Monitor Serial do Arduino

Através desse monitor foram realizados os testes para saber se os dados recebidos pelo sensor batiam com a realidade e os resultados foram satisfatórios.


```

if(cm<50){
    analogWrite(pino_pwm, 183);
    analogWrite(pino_forte, 183);
    delay(300);
}

if((cm>=51)&&(cm<80)){
    analogWrite(pino_pwm, 133);
    analogWrite(pino_forte, 133);
    delay(300);
}

if((cm>=81)&&(cm<100)){
    analogWrite(pino_pwm, 103);
    analogWrite(pino_forte, 0);
    delay(300);
}

if(cm>=100.1){
    analogWrite(pino_pwm, 0);
    analogWrite(pino_forte, 0);
}
}

```

Figura 4.1.9- Parte final da programação da Bengala Eletrônica

Na parte final da programação são definidas as condições que os motores da bengala serão ativados e em qual intensidade. Para a programação foram usados dois pinos PWM (Pulse-Width Modulation), em que é possível definir a intensidade com que o motor será ativado, dessa forma, quanto mais próximo do obstáculo mais forte os motores serão ativados.

Para distâncias maiores ou iguais a 100.1cm os motores ficam desativados, por isso o valor 0 em AnalogWrite. Quando a distância está entre 81 e 100 cm ele ativa somente um motor, com o valor de 103, esse valor significa 2V. Assim acontece para cada circunstância em que as distâncias se encontram, e cada vez o motor fica mais forte, para saber o valor da tensão que está indo para cada motor, basta fazer uma regra de 3, sabendo que o valor 255 equivale a 5V. Em nenhum momento os motores foram ativados com 255, pois o valor de tensão máximo que suportam é 3,6V.

Então o funcionamento da bengala ocorre desta maneira, o sensor ultrassônico reconhece a distância dos obstáculos e envia esse dado para o microcontrolador Arduino uno, que como foi visto anteriormente, realiza a leitura dos dados recebidos e ativa os motores de acordo com a distância dos obstáculos, para distâncias maiores que 1 metro, nada acontece, para que tudo isso aconteça é necessário que o Arduino esteja conectado a fonte de alimentação, que é o Power Bank.

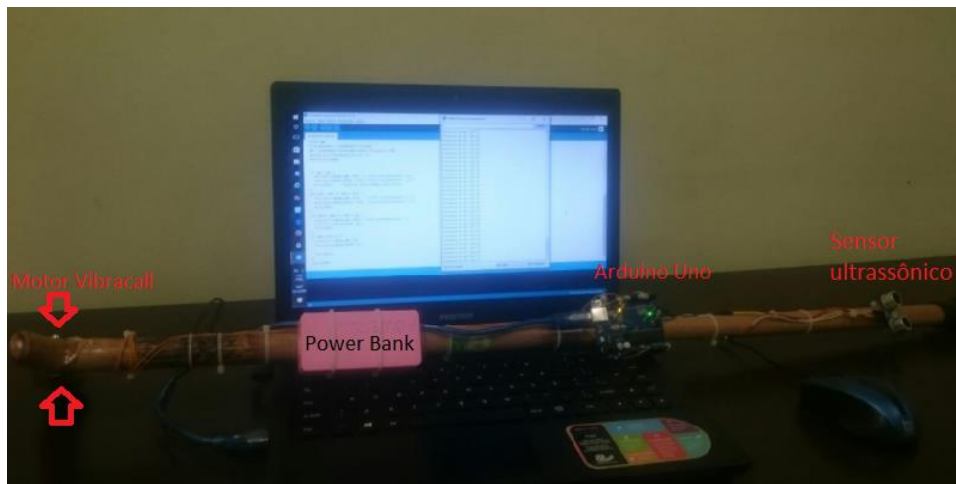


Figura 4.1.10- Bengala microcontrolada e ao fundo software Arduino.

Com esse projeto em mãos e a proposta de montar um aplicativo, pensou-se em algo que pudesse unir os projetos, e dessa forma inicia-se a elaboração do aplicativo SecondEye.

4.2 SecondEye

O SecondEye é um aplicativo, em apk, que foi pensado exclusivamente para a bengala microcontrolada, consiste em uma tela que inicialmente está toda vermelha e que em qualquer lugar que se clica nesta tela a torna verde. Essa troca de cores significa que um sinal foi enviado para a bengala, por meio da rede Wifi e faz com que os motores vibracall vibrem e com o passar do tempo ela passa a emitir um som (os dispositivos usados para receber o sinal Wifi e para emitir um som serão explicados no próximo tópico). A utilidade do aplicativo é para quando o deficiente visual esquecer onde deixou a bengala, ao apertar o botão, a vibração começa, pois se essa pessoa está em casa o som da vibração já pode tornar possível de encontrar a bengala, mas caso isso não aconteça o buzzer é acionado e faz com que ela seja encontrada.

Para o desenvolvimento deste aplicativo foi utilizado o site AppInventor, que é uma plataforma do MIT, Instituto de Tecnologia de Massachusetts. O AppInventor possui duas janelas para a criação do aplicativo, uma para o design, enquanto a outra é para a programação que é feita em blocos.

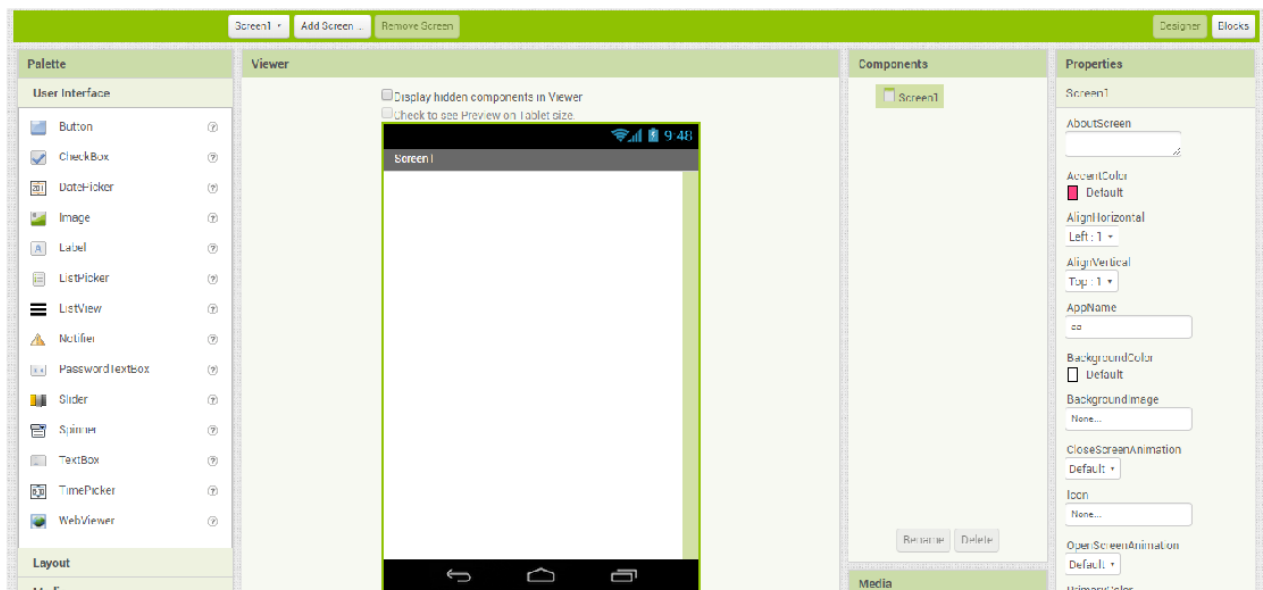


Figura 4.2.1- Área de design do AppInventor

Nesta janela é criado todo o design do aplicativo, a localização dos botões, o texto e são selecionados os recursos que serão utilizados, para este aplicativo foi usado o recurso WEB e TinyDB1 e 1 botão que cobre toda a tela. E note que apesar da tela ser de uma cor só e não possuir nenhuma gravura, são adicionadas duas imagens, sino desligado e sino ligado, a razão será explicada posteriormente.

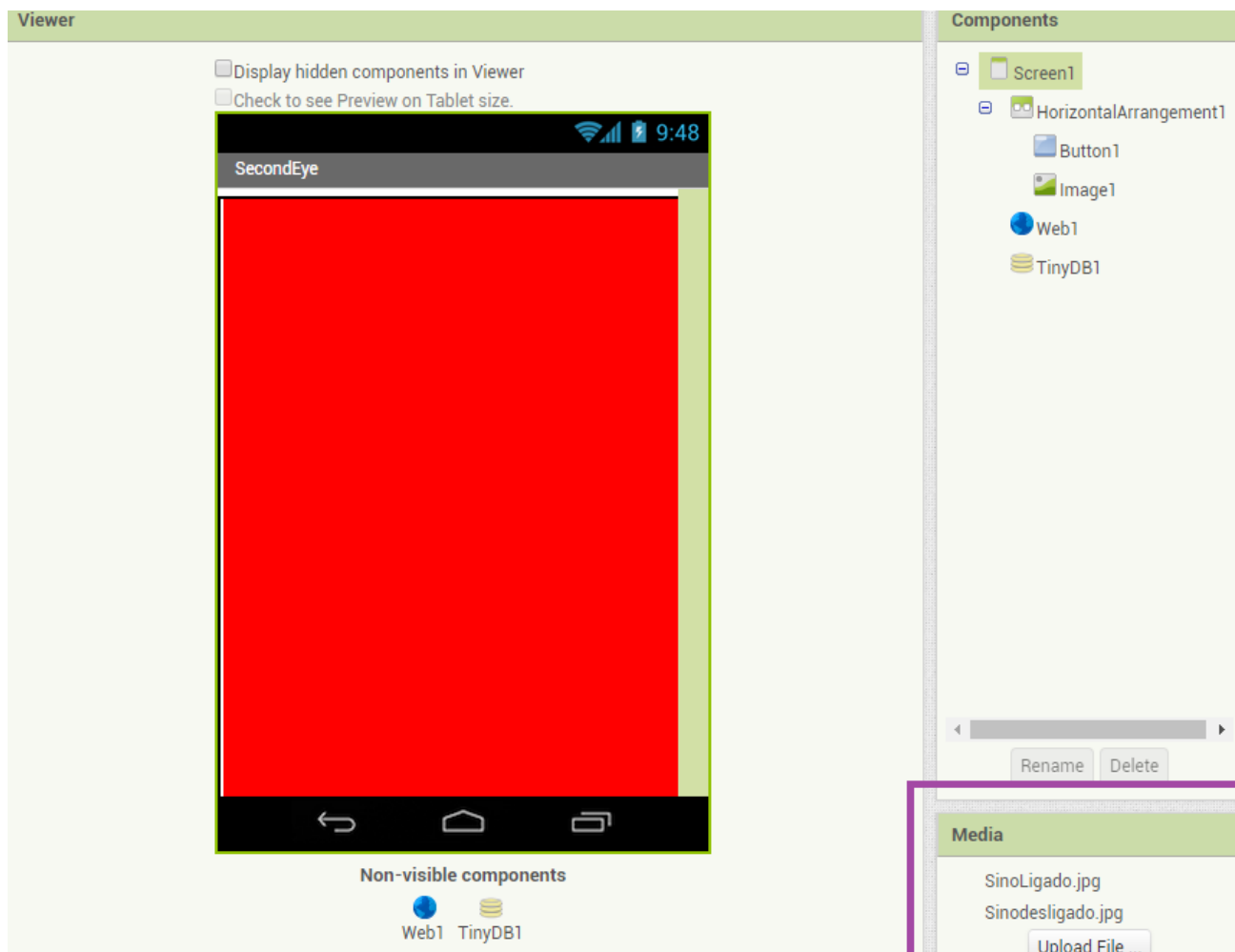


Figura 4.2.2- Design do SecondEye

Na outra janela é feita toda a programação do aplicativo, é decidido a função do botão, a mudança de cores na tela e a forma como ele interage com a web. Como mostra a figura abaixo:

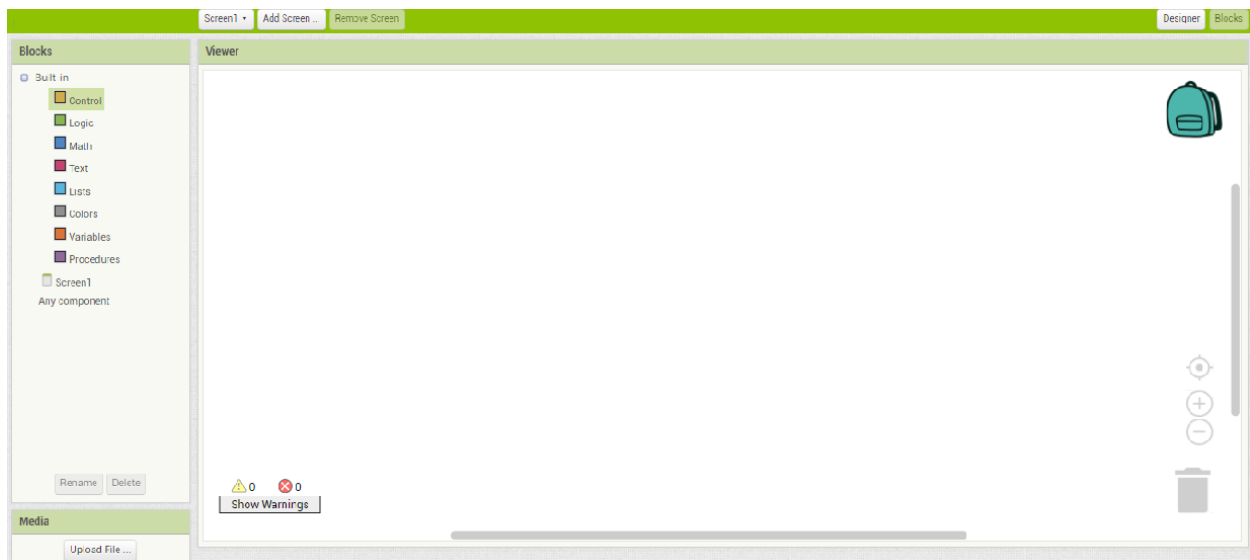


Figura 4.2.3- Área de programação do AppInventor

A esquerda ficam todos os recursos que foram selecionados na área de design e os comandos de programação.

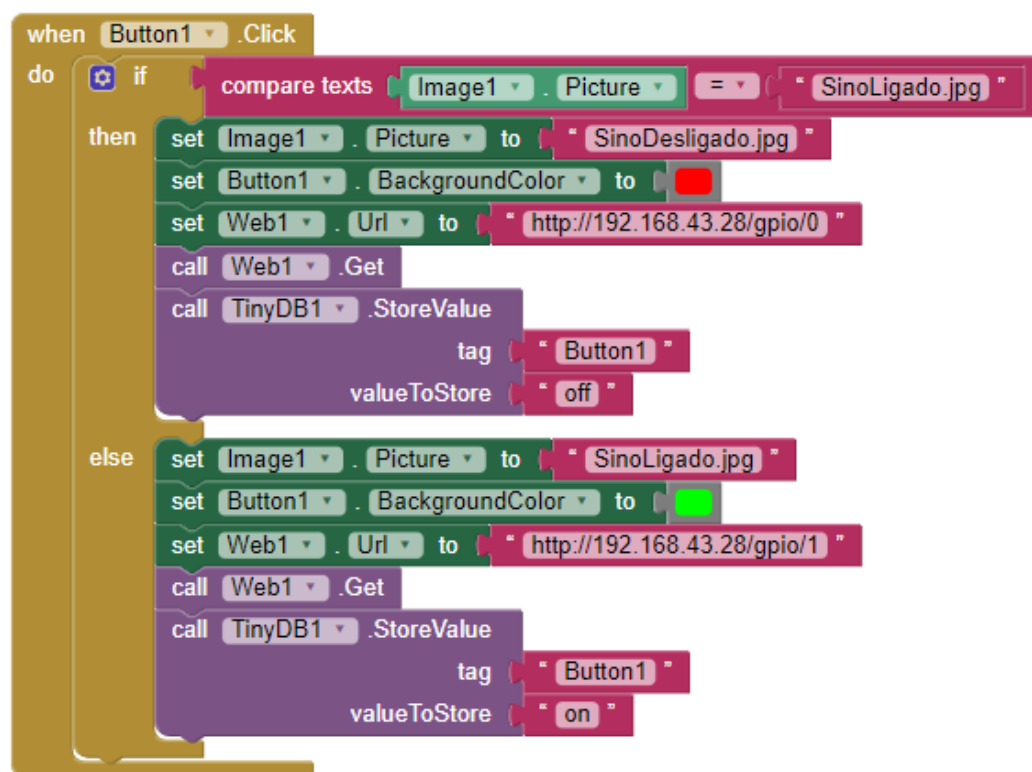


Figura 4.2.4- Parte da programação do SecondEye.

A figura 4.2.4 corresponde a parte da programação do aplicativo. O bloco caramelo representa que o programa somente irá entrar nesse bloco quando o botão 1 for clicado, que é o botão que

corresponde a parte inteira da tela. A partir do momento em que ele entra no bloco ele irá fazer um teste, testando aquelas imagens que foram citadas na figura 4.2.1, se a imagem 1 for igual a do sino ligado ele entra no bloco do "if then" que troca a imagem para a do sino desligado, passa a tela para a cor vermelha e manda o endereço de ip/gpio/0 que manda o nível lógico 0 para a entrada do Arduino, tudo isto acontece nas barras que começam com "set", caso contrário, ou seja, se a imagem 1 for igual a imagem do sino ligado, ele troca a imagem 1 para a do sino desligado, passa a tela para a cor verde e manda para o ESP o endereço de ip/gpio/1 que corresponde a mandar nível 1 para a entrada do Arduino, novamente tudo isso acontece nos blocos de "set".

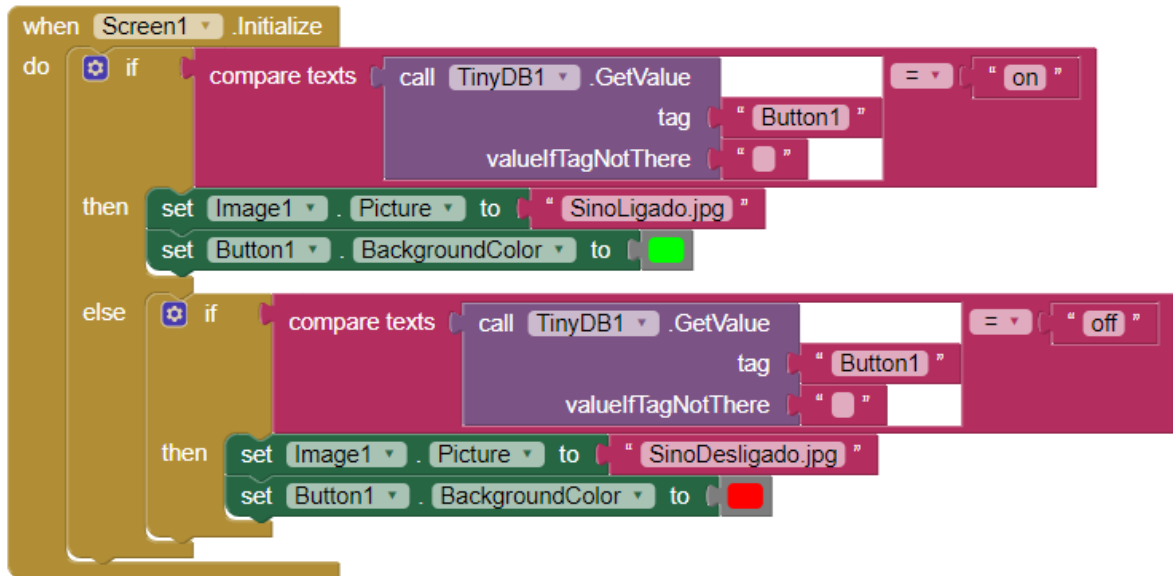


Figura 4.2.5- Parte da programação do SecondEye.

Os blocos de "call" da figura 4.2.4 interagem com os blocos dessa figura, 4.2.5, que é responsável pela inicialização do aplicativo. Os blocos de "call" guardam no recurso "TinyDB1" o estado que o aplicativo se encontrava quando ele foi fechado, isto é, se ele estava com a tela verde quando ele for aberto novamente ele será aberto com a tela verde, o mesmo vale para a vermelha. Esse recurso é de extrema importância, pois se quando o aplicativo foi fechado o buzzer estava ligado, sem o recurso de memória do aplicativo o buzzer não se desligaria automaticamente com a pessoa clicando 1 vez na tela.

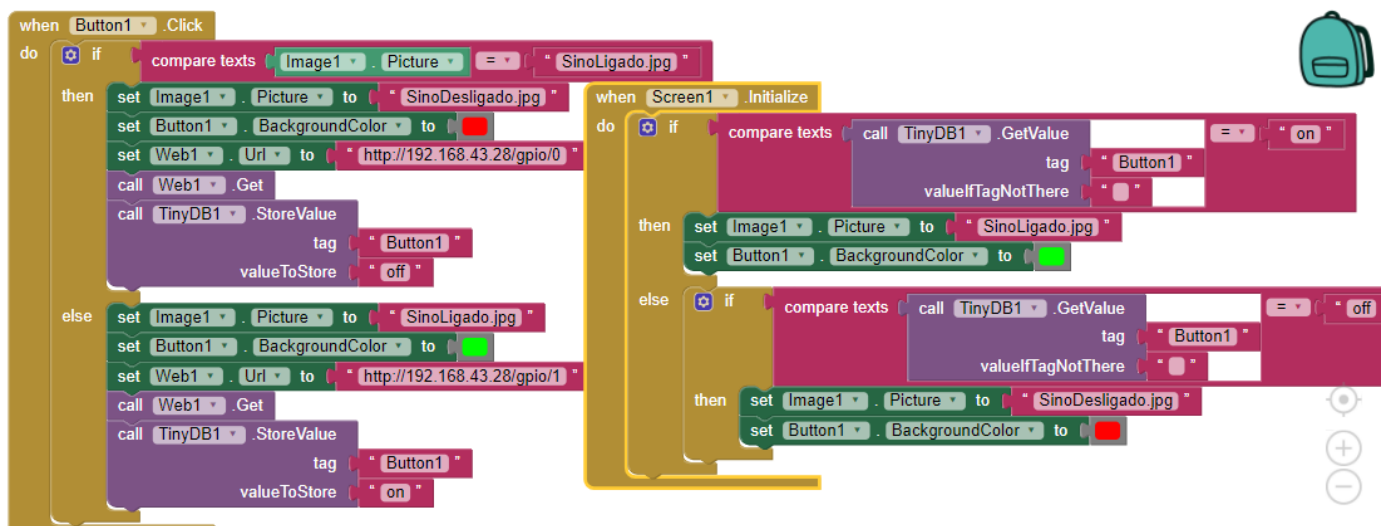


Figura 4.2.6- Programação completa do aplicativo.

4.3 Interação aplicativo SecondEye com ESP8266

Com o aplicativo e a bengala eletrônica funcionando, torna-se necessário a construção de um projeto para que eles possam interagir. O aplicativo foi criado para a comunicação WiFi, um meio de conseguir fácil comunicação com o Arduino Uno, utilizando esse recurso, é o ESP 8266, outra vantagem dele é seu baixo custo.

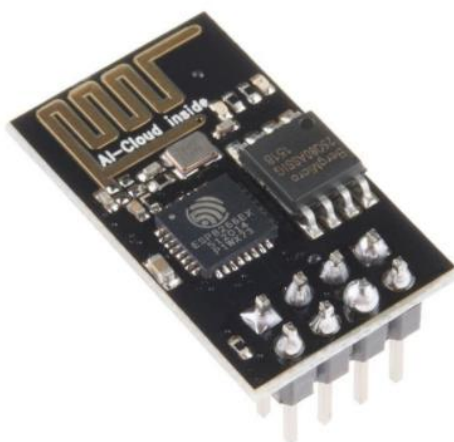


Figura 4.3.1- ESP 8266

A dificuldade que surge com o ESP é que ele suporta uma tensão de alimentação de até 3,3V e as saídas do Arduino, tirando o VCC que tem as opções de 3.3V e de 5V, todas possuem 5V e isso queimaria os pinos do ESP na hora de realizar sua programação. Porém, a solução deste problema é simples, basta montar um divisor de tensão da saída do Arduino, em vez de ligar o ESP diretamente.

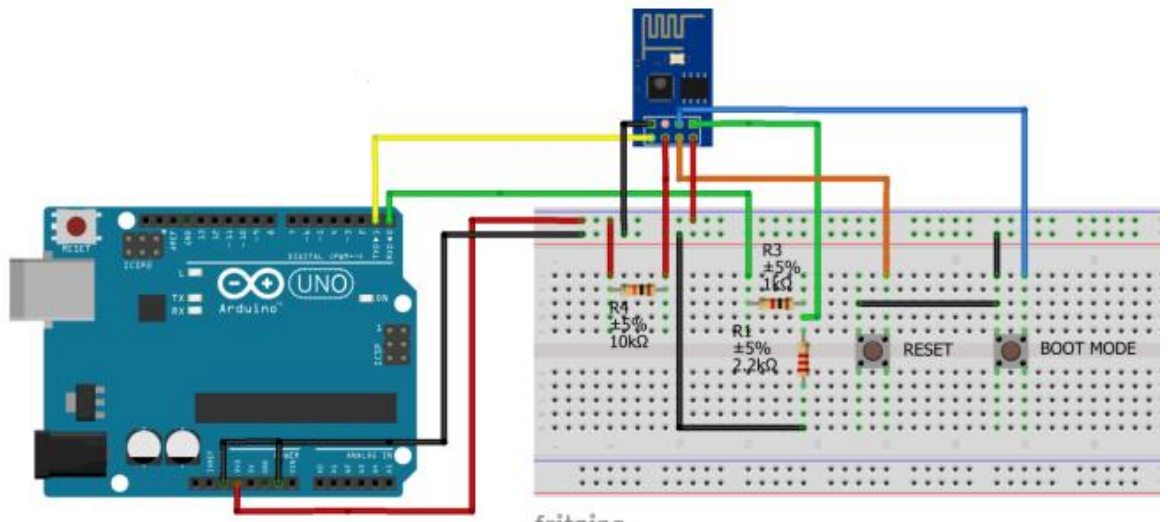


Figura 4.3.2- Esquema de ligação para programar ESP8266

Esse esquema de ligação permite que a linguagem de programação do ESP e do Arduino sejam a mesma e que o módulo WiFi seja programado diretamente pela IDE do Arduino com a inclusão de algumas bibliotecas, como será mostrado abaixo.

Configuração da IDE do Arduino para programação do ESP:

- Verificar se a IDE do Arduino é acima da 1.6.5 pois essas bibliotecas só estão disponíveis a partir dessa versão.
- Ao inicializar a IDE do Arduino clique em arquivo, preferências,

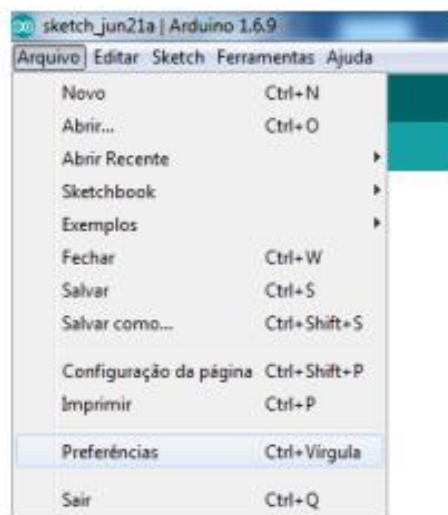


Figura 4.3.3- Barra de ferramentas da IDE Arduino.

- Será aberta uma nova janela e a URL: http://arduino.esp8266.com/versions/2.4.0/package_esp8266com_index.json deve ser copiada na janela "URL adicionais de gerenciamento de placa".

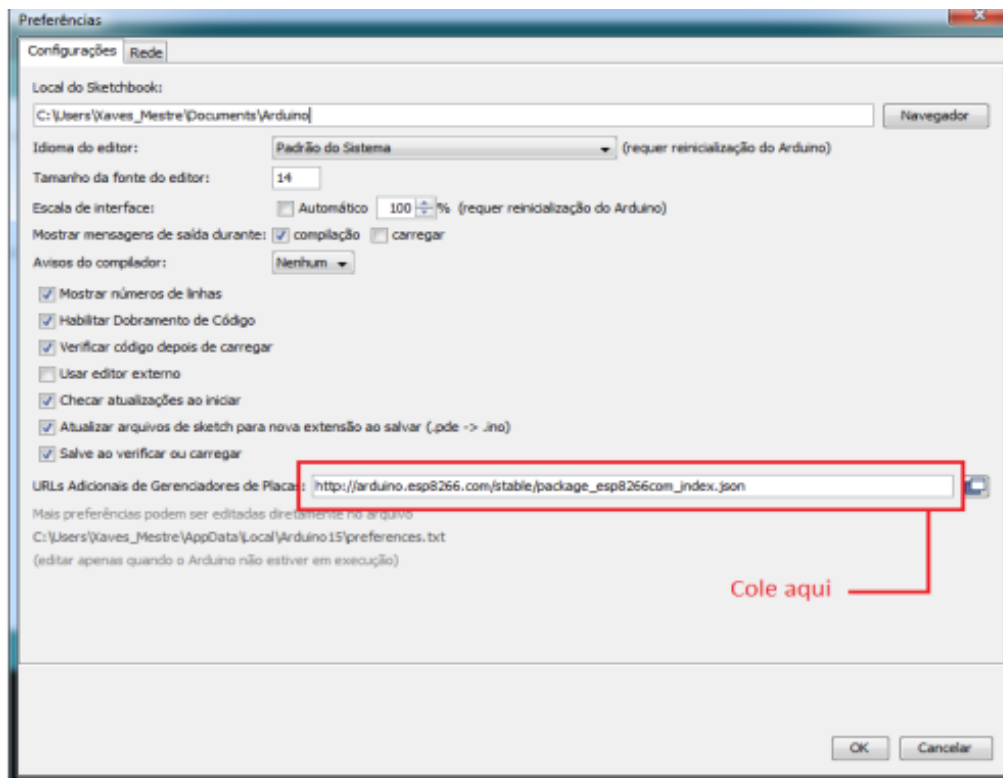


Figura 4.3.4- Janela de preferências IDE Arduino.

- Posteriormente clique em Ferramentas/Placa:/Gerenciador de placas.

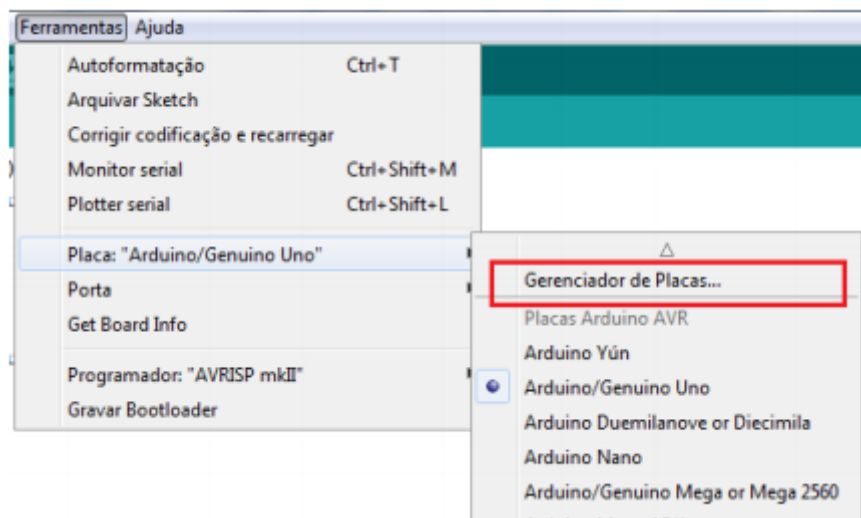


Figura 4.3.5- Ferramentas IDE Arduino.

- Na janela de Gerenciador de Placas que se abrirá, basta digitar ESP8266 no campo de busca e localizar os pacotes para o ESP8266 mostrado na figura 4.3.6.

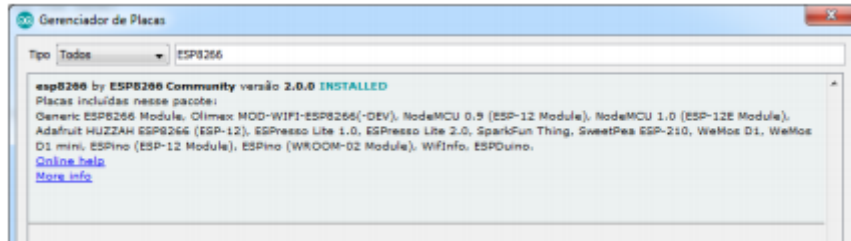


Figura 4.3.6- Pacotes ESP8266 .

•Por fim, clique novamente em Ferramentas / Placa / e localize os módulos ESP8266 já instalados. Como o módulo utilizado é o ESP01 a escolha do módulo é “Generic ESP8266 Module”

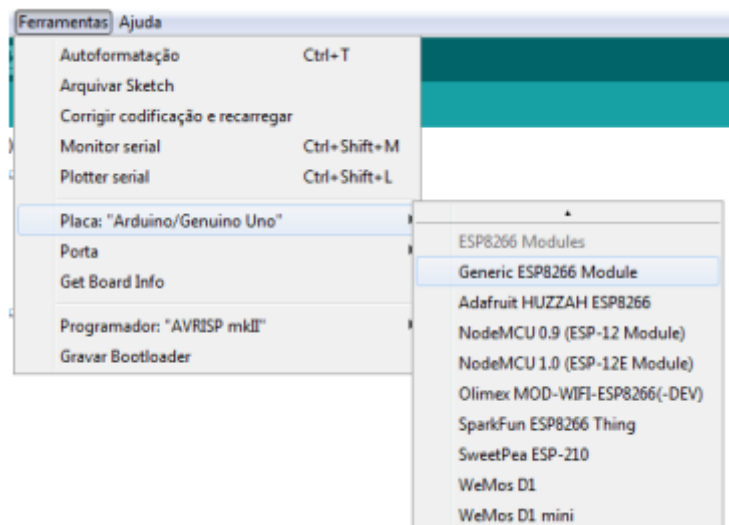
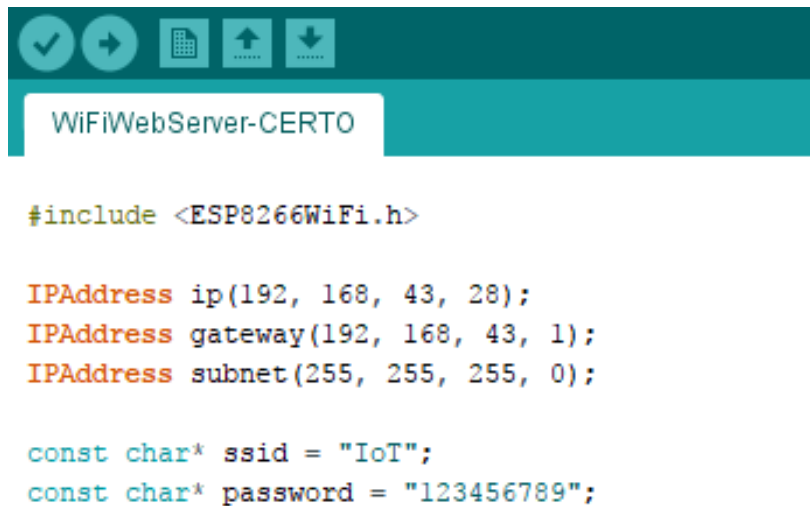


Figura 4.3.7- Seleção do módulo ESP8266

Após realizar essas configurações basta colocar o ESP 8266 no modo gravação, mantendo o botão RESET pressionado enquanto aperta uma única vez o botão BOOT MODE.

Seguindo todos esses passos o ESP fica pronto para receber a programação, para que a programação funcione é necessário a existência de uma conexão WiFi, então foi estabelecida uma rede WiFi por meio do celular, porém verificou-se que sempre que essa rede era desligada o endereço IP mudava e pro aplicativo SecondEye funcionar é necessário que o IP seja fixo. Essa configuração de Ip foi realizada pela IDE do Arduino no programa que o ESP recebeu.

Abaixo serão explicadas as partes chaves da programação do ESP, alguns aspectos como configuração da velocidade das portas, delays, etc, não serão detalhados.



```
#include <ESP8266WiFi.h>

IPAddress ip(192, 168, 43, 28);
IPAddress gateway(192, 168, 43, 1);
IPAddress subnet(255, 255, 255, 0);

const char* ssid = "IoT";
const char* password = "123456789";
```

Figura 4.3.8- Linhas iniciais da programação do ESP

Na figura acima, as linhas que começam com IPAddress são as responsáveis pela configuração do IP fixo e a linha de const char são responsáveis por conectar o ESP a rede, colocando em ssid o nome da rede e em password a senha da rede.

```
void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare 2
  pinMode(2, OUTPUT);    // Initialize the 2 pin as an output
  digitalWrite(2, LOW);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.config(ip, gateway, subnet);
  WiFi.begin(ssid, password);
```

Figura 4.3.9- Parte da programação do ESP

O "pinMode" configura o pino 2 do ESP como saída, é este pino que enviará dados para o Arduino e o digitalWrite define que ele começará no estado "Low", enviando o sinal 0 para o Arduino. Já o comando WiFi.config, configura como será feita a conexão, que são com os endereços que definimos na imagem 4.3.8 e o WiFi.begin inicializa a conexão.

```

int val;
if (req.indexOf("/gpio/0") != -1)
    val = 0;
else if (req.indexOf("/gpio/1") != -1)
    val = 1;
else {
    Serial.println("invalid request");
    client.stop();
    return;
}

// Set GPIO2 according to the request
digitalWrite(2, val);

```

Figura 4.3.10- Recorte da programação do ESP

Nesta parte é definido uma variável "val" através da primeira linha. No tópico 4.2, na figura 4.2.4 é explicado que o aplicativo manda um endereço de IP para o ESP, esse endereço de IP é testado, caso ele termine com "gpio/0" a variável *val* assume o valor 0, da mesma forma, se ele termina com "gpio/1" a variável *val* assume o valor 1. Caso o final do endereço IP não seja nenhum desses, nada acontece, aparecendo na tela de testes um "invalid request".

Após esses testes o pino 2 recebe o valor da variável *val*, através do digitalWrite e esse pino que faz o contato direto com o Arduino.

4.4 Interação ESP8266 com bengala eletrônica.

Para que o ESP 8266 interagisse com o Arduino, foi preciso fazer algumas alterações no código original da bengala eletrônica. O primeiro passo foi realizar a conexão física entre o Arduino e o ESP8266, para isso o TX e o RX do Arduino nos equivalentes no ESP, a alimentação do ESP deve ser no 3,6V do Arduino e o GND no GND do Arduino, para fazer a troca de sinais o pino 2 do ESP é conectado ao pino 8, que passa a receber o sinal do ESP e no pino 9 é conectado um buzzer, responsável pelo sinal sonoro que a bengala passará a emitir. Dessa forma, a conexão física foi feita e nada da conexão original foi alterada.

A partir dessa ligação e com as alterações no código que serão explicadas posteriormente, o ESP passa a receber o endereço IP enviado do aplicativo, a partir do endereço recebido ele envia para o pino 8 do Arduino um sinal em 0 ou em 1, a partir desse sinal o pino 9 e o pino 3 emitem uma saída.

```

#include <Ultrasonic.h>

#define pino_trigger  12
#define pino_echo     13
#define pino_pwm      3
#define pino_forte    11
#define digital 8
#define modo 9

```

Figura 4.4.1- Definição de variáveis

Note que agora foram acrescentadas duas novas variáveis o pino 8, que foi nomeado de digital e o pino 9, que recebeu o nome de modo.

```

void setup()
{
  Serial.begin(9600);
  Serial.print("ANA CLARA AMORIM");
  Serial.print("\n\n");
  Serial.print("BENGALA ELETRONICA MICROCONTROLADA");
  Serial.print("\n\nLendo dados do sensor...\n\n");
  pinMode(8, INPUT);
  pinMode(9, OUTPUT);
}

```

Figura 4.4.1- Setup adaptado para interagir com o aplicativo.

Já no *void setup* foram adicionados duas novas linhas, de *pinMode*, que define o pino 8 como entrada (INPUT) e o pino 9 como saída (OUTPUT).

```

if(cm>=100.1){                                     //DESATIVA VIBRAÇÕES
analogWrite(pino_pwm, 0);                           //133: 0V
analogWrite(pino_forte, 0);                         //133: 0V
}

```

```

    if(
        int val = digitalRead(digital); // le o valor na entrada
    if (val == HIGH)
    {
        analogWrite(pino_pwm, 183);
        delay(500);
        delay(500);
        digitalWrite(modo,HIGH); //Liga o BUZZER
    }
    else
    {
        digitalWrite(modo, LOW); // Desliga BUZZER
    }
}

```

Figura 4.4.2- Adaptações no void Loop do programa da bengala eletrônica.

A última adaptação realizada foi dentro do void Loop, essa adaptação faz um teste com o sinal recebido pelo pino 8 (digital), se esse sinal estiver no nível lógico alto um motor da bengala eletrônica será ativado para que a pessoa através do som da vibração consiga encontrar a bengala, caso ela não encontre no intervalo de 1000ms o pino 9 (modo) passará a ter nível lógico '1', fazendo com que o buzzer seja ligado juntamente com os motores até que a bengala seja encontrada. Após a pessoa encontrar a bengala ela vai até o aplicativo e aperta a tela novamente, fazendo com que o sinal enviado ao Arduino agora passe a ser do nível lógico '0', nesse momento o pino 9 (modo) passa a emitir 0V de saída, desligando o buzzer e os motores são desligados após passar seu tempo de delay.

5. Resultados

No findar das pesquisas e ao analisarmos os resultados obtidos com o projeto, nota-se que a área de tecnologia assistiva, apesar de estar crescendo, precisa de mais projetos para assegurar a total inclusão da sociedade que precisa da mesma. Por isso, as pessoas que tem acesso à tecnologia, como estudantes, cientistas, engenheiros, técnicos devem buscar sempre ajudar, fazendo com que, dessa forma, os projetos fiquem cada vez mais eficazes.

A Bengala eletrônica em conjunto com o aplicativo SecondEye é um projeto que busca essa inclusão, buscando uma melhoria na locomoção dos deficientes visuais. Nos testes o projeto se provou eficaz, na identificação de obstáculos, com o HC-Sr04, tendo sido ajustados o tempo certo de cada atualização de dados de distância, para que os obstáculos sejam avisados a tempo, da mesma forma acontece com os testes do aplicativo, com o ESP 8266, que mostra uma resposta rápida.

6. Referências bibliográficas

1. SARTORETTO, Mara Lúcia; BERSCH, Rita. **O que é Tecnologia Assistiva?** 2017. Disponível em: <<http://www.assistiva.com.br/tassistiva.html>>. Acesso em: 08 mar. 2017.
2. BERSCH, Rita. **INTRODUÇÃO À TECNOLOGIA ASSISTIVA: ASSISTIVA • TECNOLOGIA E EDUCAÇÃO.** 2017. Disponível em: <http://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf>. Acesso em: 14 mar. 2017.
3. DESCONHECIDO. **A Tecnologia Assistiva e as pessoas cegas ou com baixa visão.** 2011. Disponível em: <http://www.contagem.pucminas.br/pitane/index.php?option=com_content&view=article&id=87:a>. Acesso em: 14 mar. 2017
4. DESCONHECIDO. **23,9% dos brasileiros declaram ter alguma deficiência, diz IBGE.** 2012. Disponível em: <<http://g1.globo.com/brasil/noticia/2012/04/239-dos-brasileiros-declaram-ter-alguma-deficiencia-diz-ibge.html>>. Acesso em: 14 mar. 2017.
5. OLIVEIRA, Hamilton. **7 TECNOLOGIAS INOVADORAS PARA DEFICIENTES VISUAIS.** 2016. Disponível em: <<http://www.casadaptada.com.br/2016/11/7-tecnologias-inovadoras-para-deficientes-visuais/>>. Acesso em: 22 mar. 2017.
6. THOMSEN, Adilson. **Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino.** 2011. Disponível em: <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Acesso em: 11 abr. 2017.
7. BARBACENA, Prof. Dr. Ilton; XAVIER, Antonio; FALQUETO, Alessandro. **IDE Arduino para o módulo ESP8266-01.** Disponível em: <http://iltonluiz.vhost.ifpb.edu.br/ainventor/PDFs/ESP8266/Tutorial_02_-_IDE_Arduino_+_ESP8266.pdf>. Acesso em: 24 maio 2017.