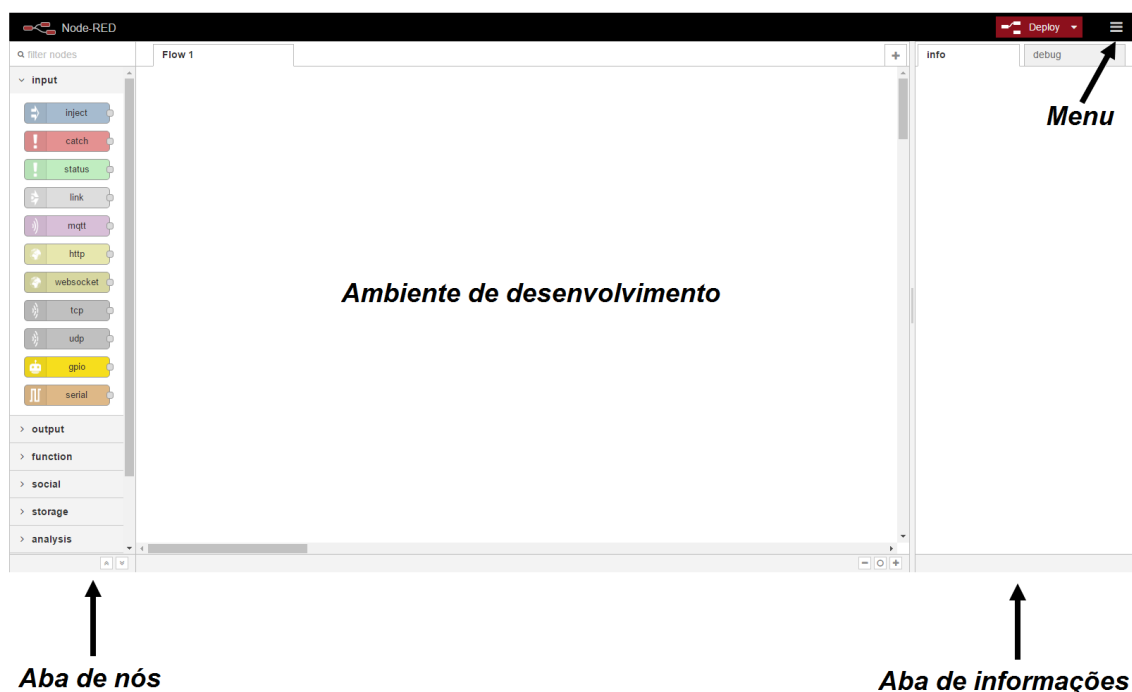
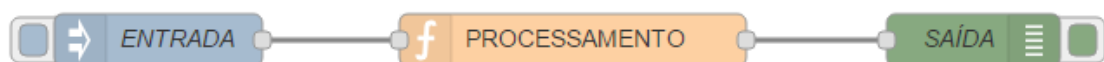


O **Node-RED** tem como função primordial auxiliar na elaboração de aplicações propondo uma maneira mais simples para o usuário que não domina muito as ferramentas de programação. Para a comunicação foi utilizado o Node-RED, que consiste em uma ferramenta bastante versátil, voltada principalmente para o desenvolvimento de aplicações relacionadas ao conceito de Internet de Coisas (IoT). Este recurso utiliza uma abordagem de programação gráfica, ou seja, através do mesmo, torna-se possível elaborar uma aplicação por meio do estabelecimento de conexões entre blocos que possuem códigos predefinidos (conhecidos como nós) para a realização de determinadas tarefas.

## ESTRUTURA DE PROGRAMAÇÃO DO NODE-RED



Na grande maioria das vezes as aplicações são desenvolvidas através da utilização destes três tipos de nós (em alguns casos, apenas dois podem ser necessários), que, quando conectados, geram uma estrutura denominada *fluxo*. Note, que as informações seguem ao longo dos *nós*, portanto, seguem de **nó** para **nó**.



## O que é MQTT

O acrônimo MQTT significa *Message Queuing Telemetry Transport*; um protocolo criado pela IBM em 1999. Esse protocolo muito leve permite a comunicação bi-direcional (não como o protocolo HTTP, mas a seu modo específico) para coleta de dados e

gerenciamento de dispositivos da Internet das Coisas. De modo similar, o gerenciamento de dispositivos complexos é feito com TR-069 e fico meditando a respeito; poderiam esses dispositivos (roteadores, switches, outros equipamentos de telecom). O protocolo utiliza uma arquitetura publish/subscribe, em contraste ao request/response do protocolo web HTTP. Desse modo é possível empurrar mensagens ao client e com uma boa implementação de lógica, fazer um belo gerenciamento dos dispositivos da sua nuvem. Uma outra característica que chama a atenção é a capacidade de buffering que, ainda que haja uma desconexão com o broker por qualquer razão, ao retomar a conexão os clients recebem as informações do tempo de indisponibilidade.

Instalando as bibliotecas do MQTT, você deverá acessar os menus da IDE do Arduino na ordem Sketch->Include Library->Manage Libraries e então escolher a biblioteca que deseja utilizar, digitando “MQTT” na caixa de pesquisa. Eu escolho a biblioteca que aparece em segundo, logo abaixo da biblioteca da Adafruit. No menu File->Examples você deve encontrar o sub-menu CMMC MQTT Connector, onde diversos exemplos estão dispostos.

### **Sensor Ultrassônico HC-SR04**

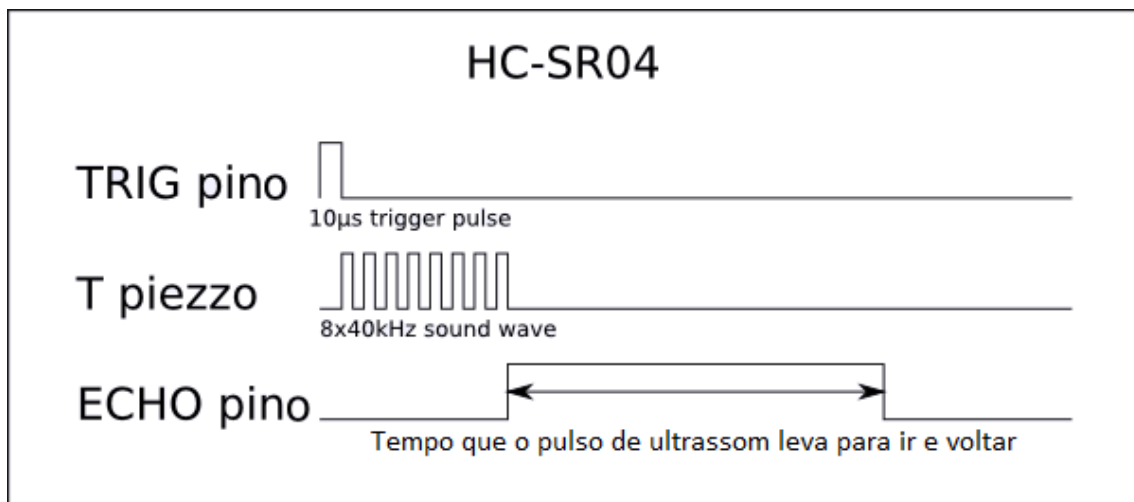
O sensor ultrassônico HC SR04 é amplamente utilizado em aplicações onde se deseja medir distância ou evitar colisões, como na robótica móvel e de reabilitação. Neste tutorial aprenderemos como utilizar o Módulo Sensor HC-SR04 com Arduino. Tudo começa pela emissão de um pequeno pulso sonoro de alta frequência que se propagará na velocidade do som no meio em questão. Quando este pulso atingir um objeto, um sinal de eco será refletido para o sensor. A distância entre o sensor e o objeto pode então ser calculada caso saibamos o tempo entre a emissão e a recepção do sinal, além da velocidade do som no meio em questão. Afigura a seguir exemplifica o processo.

### **Buzzer**

O buzzer é um componente que se baseia no efeito piezoelétrico para emitir sons a partir de uma tensão elétrica em seus terminais, ele funciona a partir do efeito piezoelétrico reverso. O principio desse efeito é o surgimento de uma tensão elétrica a partir de um esforço mecânico. Conectando o buzzer em uma porta digital, basta apenas enviar um sinal em nível lógico alto para fazer o componente apitar. Ele possui dois terminais, um mais comprido que é o positivo e outro mais curto que é o GND.

## Fazendo a leitura do tempo

Para iniciarmos uma medição, o pino Trig, que funciona como gatilho do nosso sensor ultrassom, deve receber um pulso de 5V por pelo menos 10 microssegundos. Isso fará com que o sensor HC-SR04 emita 8 pulsos ultrassônicos em 40kHz (T piezzo) e o pino ECHO, que funcionará como nosso cronômetro, vai para 5V, iniciando assim a espera pelas ondas refletidas. Assim que uma onda refletida for detectada, o pino Echo, que estava em 5V, será alterado para 0V. Desta forma, o período que o pino Echo fica em 5V é igual ao tempo que a onda emitida leva para ir até o obstáculo e voltar.



Por fim, para sabermos quão longe nosso obstáculo está, basta contarmos quanto tempo (T) a tensão no pino Echo ficou em 5V. Em posse desse tempo, sabendo que ele é o dobro do tempo de ida e volta da onda do sensor ultrassônico até o obstáculo, e, considerando a velocidade do som igual a 340,29 m/s, temos:

$$V_{som} = 340,29 \frac{m}{s}$$

$$V_{som} = \frac{\Delta S}{\Delta T} = 340,29 \frac{m}{s}$$

$$\Delta T = T/2$$

$$\frac{\Delta S}{T/2} = 340,29$$

$$\Delta S = \frac{340,29 * T}{2}$$

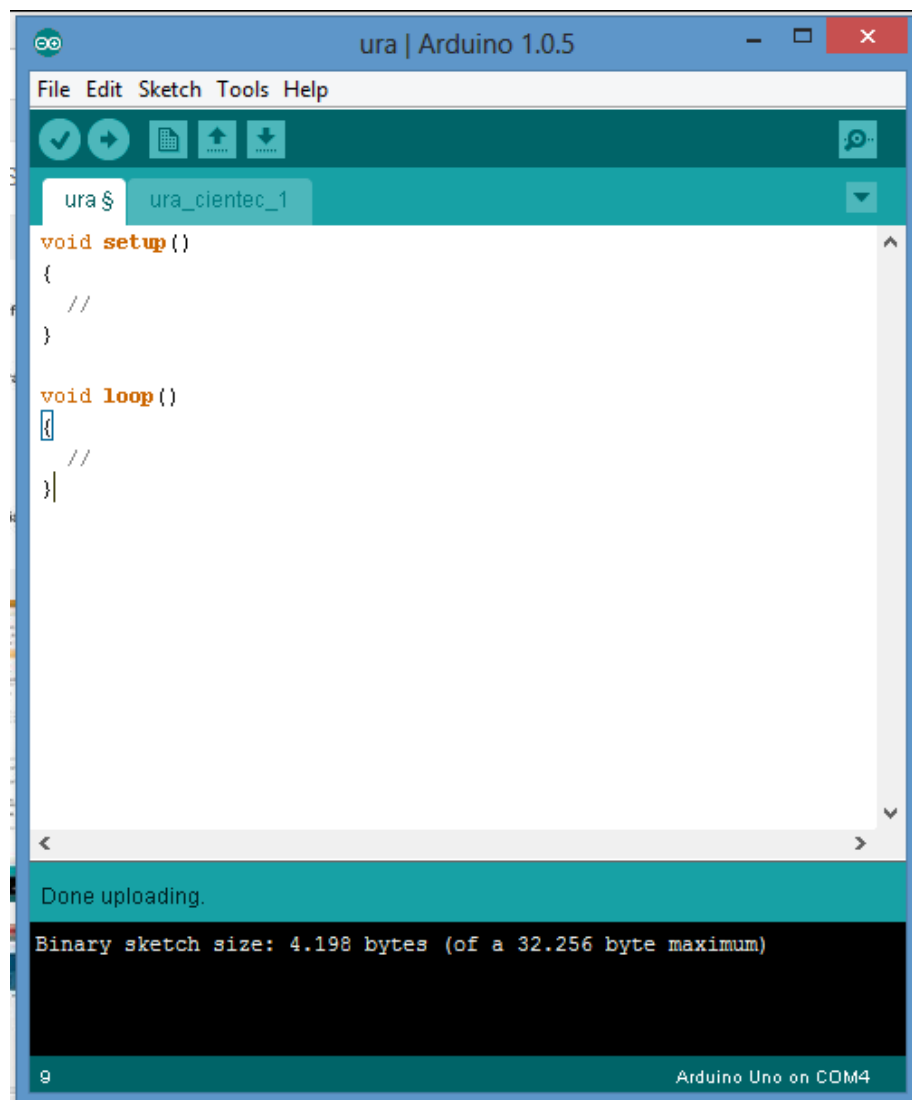
## Arduino

Arduino uno é uma plataforma open-source de prototipagem eletrônica com hardware e software flexíveis e fáceis de usar, destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos. A placa Arduino UNO

possui pinos de entrada e saídas digitais, assim como pinos de entradas e saídas analógicas, abaixo é exibido a pinagem conhecida como o padrão Arduino. A placa pode ser programada através da comunicação serial, pois o microcontrolador vem programado com o bootloader. Dessa forma não há a necessidade de um programador para fazer a gravação (ou upload) do binário na placa. A comunicação é feita através do protocolo STK500. A programação do microcontrolador também pode ser feita através do conector ICSP (in – circuit serial programming) utilizando um programador ATMEL.

## IDE do Arduino

A IDE do Arduino é um ambiente de desenvolvimento integrado. Em outras palavras, é um espaço onde você tem tudo que precisa para programar sua placa baseada nessa plataforma escrevendo seus códigos de maneira satisfatória, rápida e eficiente. A maioria das coisas que você quiser criar, pode ser feita através desse programa.



Como Funciona o Protótipo

Este trabalho teve como objetivo propor um protótipo de uma bengala inteligente que auxilia na deficientes visuais a se locomoverem com melhor qualidade e segurança através da emissão de alertas sonoros e vibratórios diante da detecção de obstáculos e alcançar um custo baixo e para isso foram utilizados como principais componentes o microcontrolador Arduino, um buzzer, um módulo Bluetooth, placa protoboard e o próprio celular do usuário. O Arduino pode se conectar através de uma conexão Bluetooth a diversos dispositivos como notebooks, tablets e celulares, o que permite uma maior variedade de alertas que podem ser emitidos. Pode-se, por exemplo, emitir um sinal vibratório para um dispositivo externo junto ao corpo do usuário como o seu próprio celular. Assim, após a finalização inicial do protótipo da bengala inteligente, os testes de detecção dos obstáculos foram realizados a fim de obter uma melhor calibração na sensibilidade para detecção de obstáculos, o código estável disponível, qualquer um poderá melhorar e expandir o escopo do projeto.

#### Explicação do código

```
const int echoPin = 6; //PINO DIGITAL UTILIZADO PELO HC-SR04
ECHO(RECEBE)

const int trigPin = 7; //PINO DIGITAL UTILIZADO PELO HC-SR04 TRIG(ENVIA)

const int pinoBuzzer = 2; //PINO DIGITAL EM QUE O BUZZER ESTÁ CONECTADO
Ultrasonic ultrasonic(trigPin,echoPin); //INICIALIZANDO OS PINOS

int distancia; //CRIA UMA VARIÁVEL CHAMADA "distancia" DO TIPO INTEIRO

void setup(){
  pinMode(echoPin, INPUT); //DEFINE O PINO COMO ENTRADA (RECEBE)
  pinMode(trigPin, OUTPUT); //DEFINE O PINO COMO SAÍDA (ENVIA)
  pinMode(pinoBuzzer, OUTPUT); //DECLARA O PINO COMO SENDO SAÍDA
}

void loop(){
  hcsr04(); // FAZ A CHAMADA DO MÉTODO "hcsr04()"

  if(distancia <= 70){// SE A DISTÂNCIA ENTRE O OBJETO E O SENSOR
    ULTRASONICO FOR MENOR QUE 30CM, FAZ
    tone(pinoBuzzer,1500);//ACIONA O BUZZER
  }else{//SENÃO, FAZ
    noTone(pinoBuzzer);//BUZZER PERMANECE DESLIGADO
  }
}

//MÉTODO RESPONSÁVEL POR CALCULAR A DISTÂNCIA

void hcsr04(){
  digitalWrite(trigPin, LOW); //SETA O PINO 6 COM UM PULSO BAIXO "LOW"
```

```
delayMicroseconds(2); // DELAY DE 2 MICROSSEGUNDOS
digitalWrite(trigPin, HIGH); //SETA O PINO 6 COM PULSO ALTO "HIGH"
delayMicroseconds(10); // DELAY DE 10 MICROSSEGUNDOS
digitalWrite(trigPin, LOW); //SETA O PINO 6 COM PULSO BAIXO "LOW"
NOVAMENTE

// FUNÇÃO RANGING, FAZ A CONVERSÃO DO TEMPO DE
//RESPOSTA DO ECHO EM CENTÍMETROS E ARMAZENA
//NA VARIÁVEL "distancia"

distancia = (ultrasonic.Ranging(CM)); // VARIÁVEL GLOBAL RECEBE O VALOR
DA DISTÂNCIA MEDIDA

delay(500); //INTERVALO DE 500 MILISSEGUNDOS
}
```