

Daniel Felipe	41584945
Dário Teles de Oliveira	41582391
Ramon Cardoso	41582802

**SmartCane**

## Sumário

Contextualização e descrição da proposta.....	3
Especificação de hardware.....	3
Arduino uno R3.....	3
Sensor de Distância Ultrassônico HC-SR04.....	3
Micro Motor c/ Pêndulo 12V / 4600 RPM.....	4
Bengala comum.....	4
Jumpers.....	4
Tinkercad.....	5
Circuito.....	5
Descrição da Física.....	6
Descrição hardware.....	7
Arduino UNO.....	7
Sensor de Distância Ultrassônico HC-SR04.....	8
Micro Motor c/ Pêndulo 12V / 4600 RPM.....	8
Código Arduino.....	9
Node-Red.....	10

## Contextualização e descrição da proposta

Segundo dados do IBGE de 2010, no Brasil, haviam mais de 6,5 milhões de pessoas com alguma deficiência visual, sendo um pouco mais de 500 mil pessoas cegas. Por conta disso, para poderem se locomoverem fazem o uso de bengalas ou de cães guias.

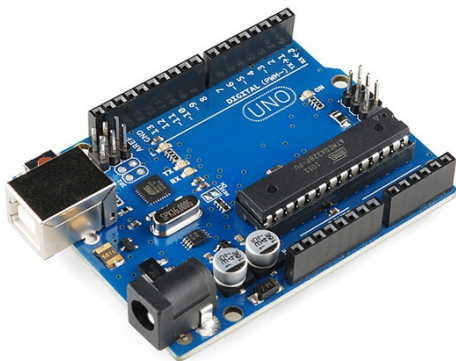
A bengala, hoje em dia, é o instrumento que é utilizado para identificação para os indivíduos com deficiência visual, porém a sua utilização pode ser complicada dependendo da situação e pode haver dificuldades na identificação de obstáculos. Em contrapartida, há os cães guia que ajudam ainda mais os indivíduos no dia a dia, porém a obtenção de um cão guia é algo difícil. No Brasil, há apenas 160 cães guias e apenas 60 novos são treinados por ano (Estadão, 2016) e preparar um cão guia custa em média R\$ 30.000 e demora cerca de 2 anos (Globo News, 2011).

Observando essa deficiência está sendo elaborado uma bengala inteligente (Smart Cane), uma bengala comum que faz uso de internet das coisas (IOT). Será inserido em cima de uma bengala comum, um arduino conectado a um sensor de distância e conforme o obstáculo for se aproximando (6 cm), um micro motor será acionado vibrando a bengala desta forma alertando o deficiente, ao mesmo tempo será enviado uma mensagem por um bot com uma mensagem alertando que um objeto foi detectado.

## Especificação de hardware

Será utilizado os seguintes componentes no desenvolvimento do projeto:

### Arduino uno R3



### Sensor de Distância Ultrassônico HC-SR04



**Micro Motor c/ Pêndulo 12V / 4600 RPM**



**Bengala comum**



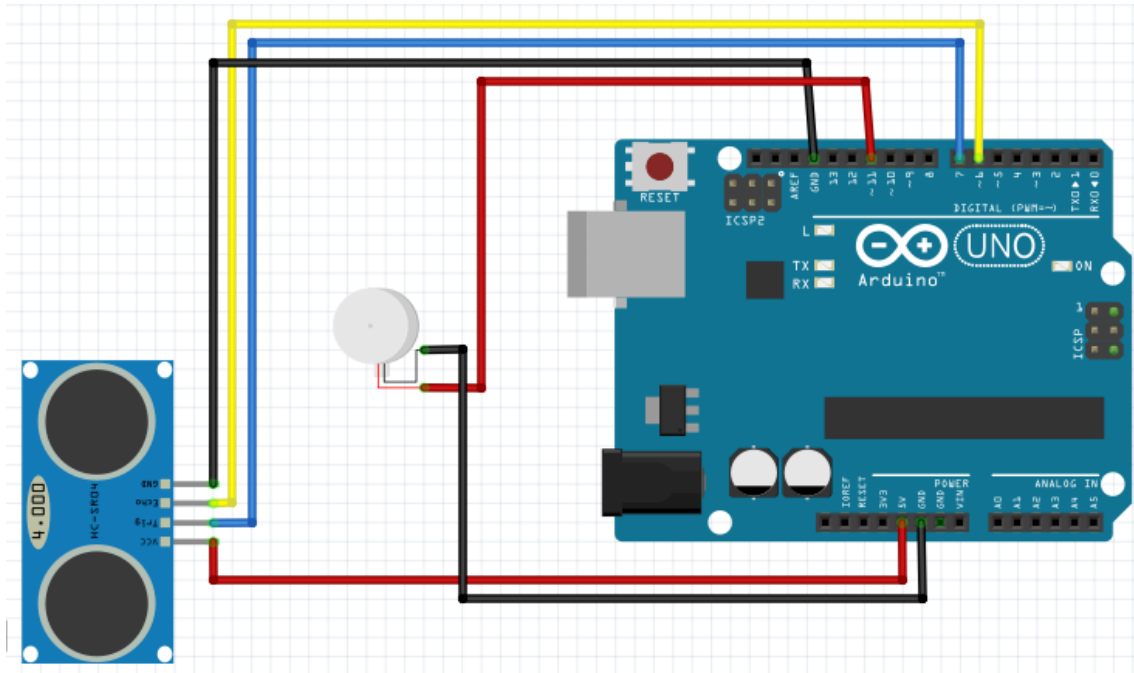
**Obs.:** Foi utilizado uma barra de ferro para simular uma bengala.

**Jumpers**

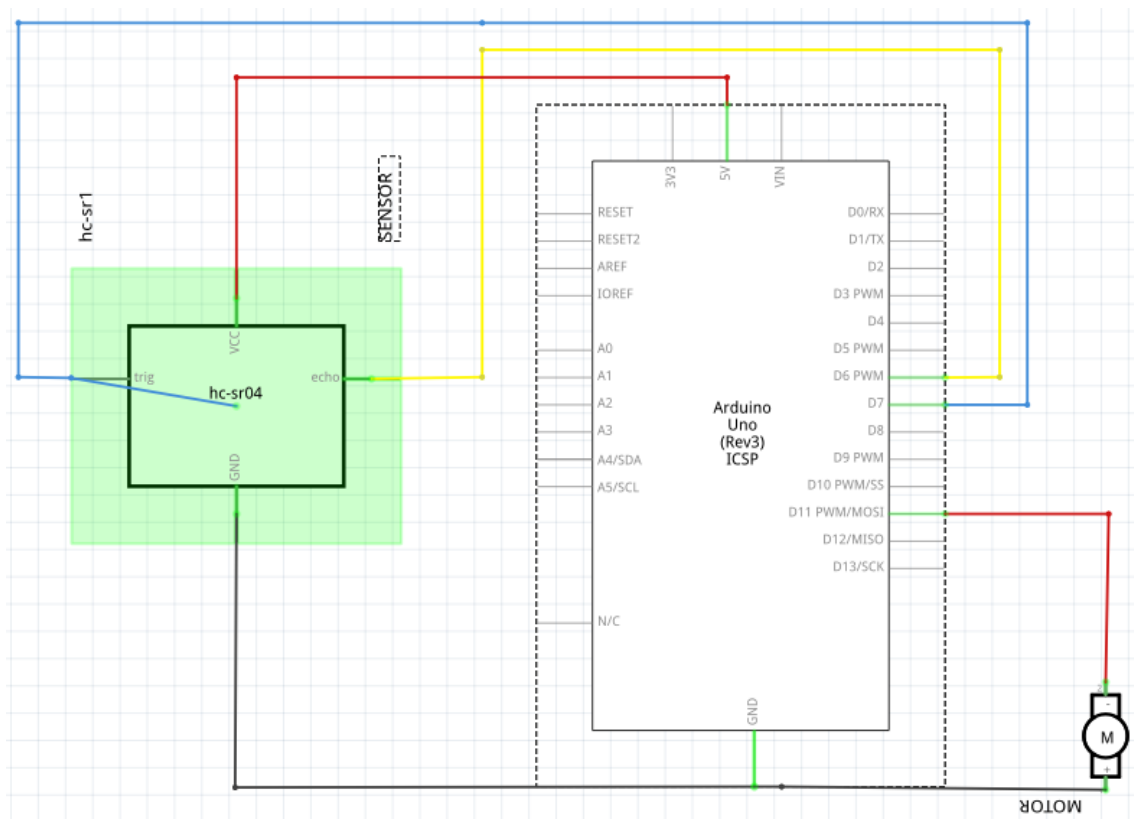


**Obs.:** As imagens são meramente ilustrativas.

## Tinkercad



## Circuito



## Descrição da Física

Foi utilizado uma barra de ferro com o tamanho de 1,05 m com um apoio da bengala feita manualmente de madeira de 3 cm.

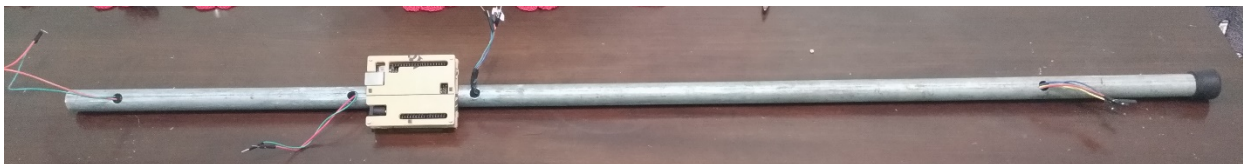


*Figura 1 – Apoio*

Foram feitos **4 furos** na barra de ferro para que fosse possível passar os jumpers, nas seguintes posições:

1. **5 cm (Micromotor)**
2. **10 cm (Arduino)**
3. **15 cm (Arduino)**
4. **90 cm (Sensor Ultrassônico)**

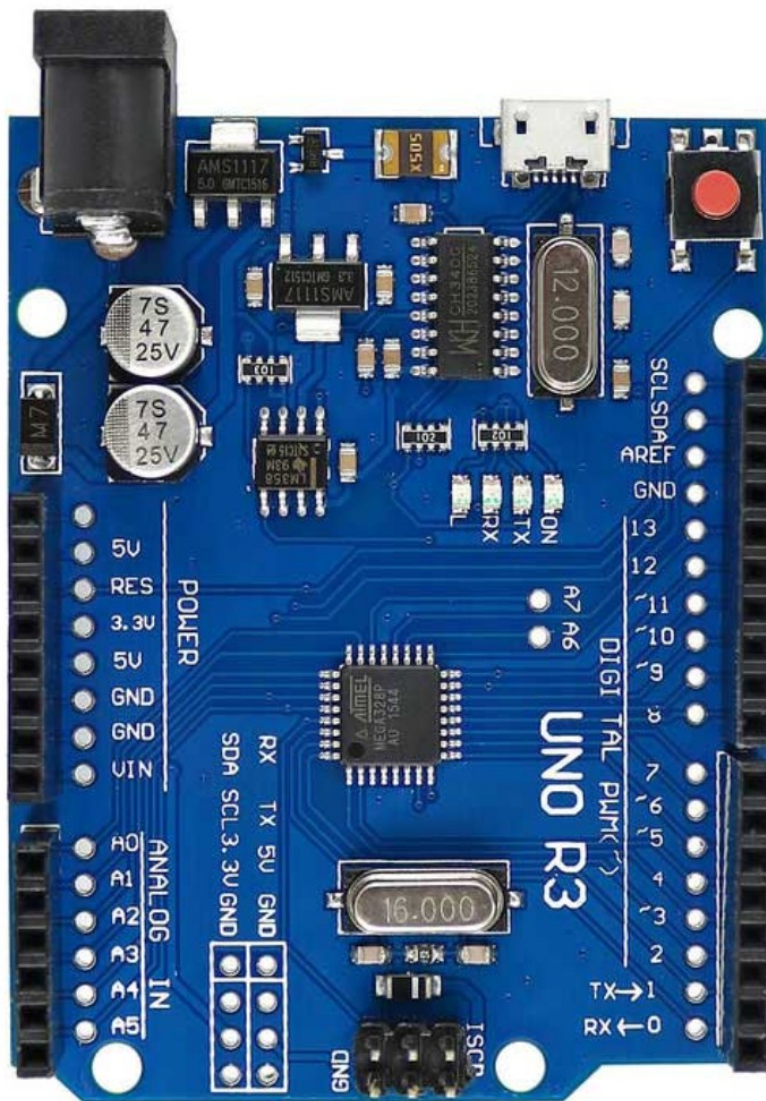
O Arduino foi parafusado em uma case para que pudesse ser colado com silicone na barra de ferro. Seu centro foi colado em **33 cm** em relação a parte superior do ferro. O resultado ficou da seguinte forma:



*Figura 2 -Bengala sem conexões*

## Descrição Hardware

### Arduino UNO



VCC ULTRASONIC

- MICRO MOTOR

→ GND ULTRASONIC

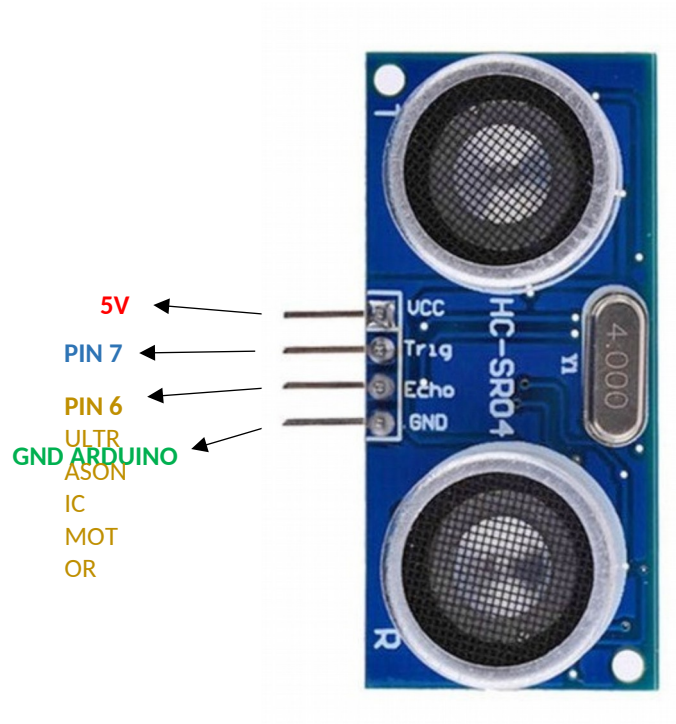
→ +MICRO MOTOR

→ TRIGGER ULTRASONIC

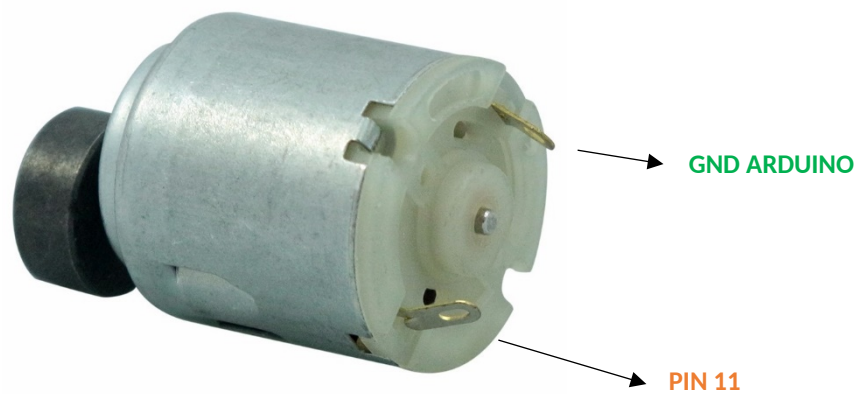
→ ECHO ULTRASONIC  
ULTRASONIC  
MOTOR



## Sensor de Distância Ultrassônico HC-SR04



## Micro Motor c/ Pêndulo 12V / 4600 RPM



Contudo devidamente ligado, ficou da seguinte forma:





**Figura 3 - Bengala com conexões**

## **Código Arduino**

Para adicionar o código no Arduino foi utilizado sua própria IDE, [download](#).



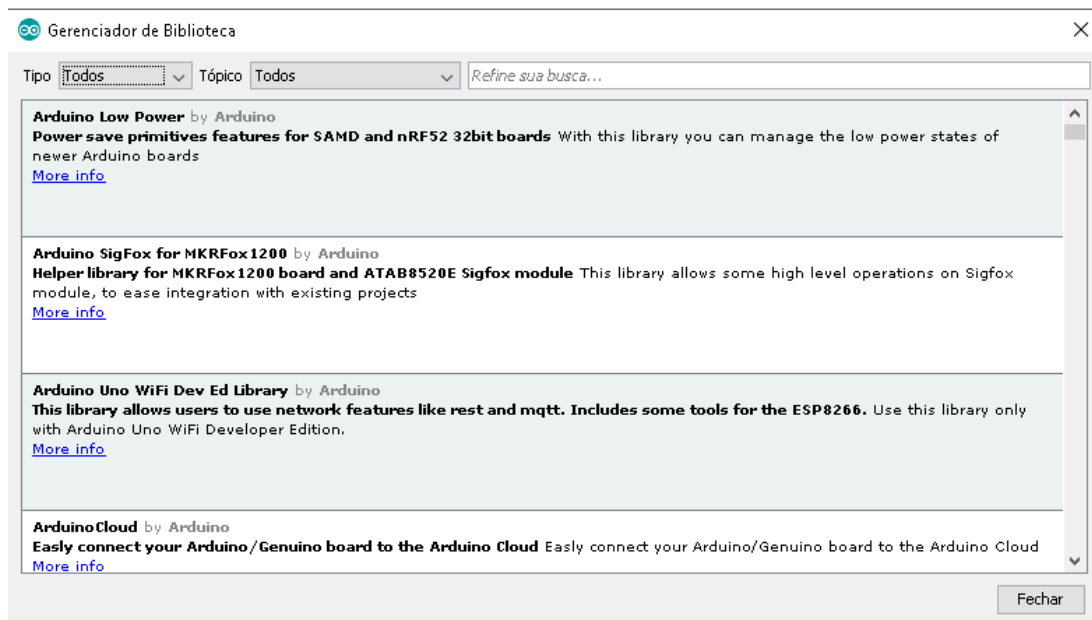
**Figura 4 - Arduino IDE**

Foi utilizado duas bibliotecas no projeto, são elas:

**Firmata** - A biblioteca Firmata implementa o protocolo Firmata para comunicação com o software no computador host. Isso permite que você grave um firmware personalizado sem ter que criar seu próprio protocolo e objetos para o ambiente de programação que você está usando.

**Ultrasonic** - Trabalha com o módulo de ultrassom de maneira simples e leve. Compatível com os módulos HC-SR04, Ping))) e sensor Seeed Studio. Esta biblioteca tem como objetivo a eficiência de recursos e simplificar o acesso aos dados.

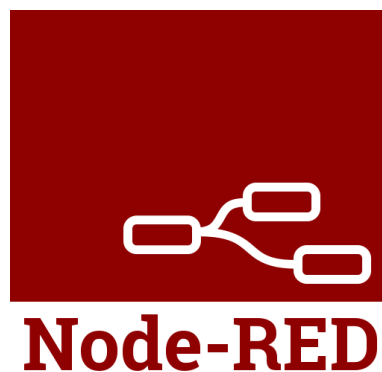
Ambos as bibliotecas foram adicionadas na IDE pelo gerenciado de bibliotecas **Sketch > Incluir Biblioteca > Gerenciar Bibliotecas...**



**Figura 5 - Gerenciador de biblioteca Arduino**

Código completo <CAMINHO>

## Node-Red



**Figura 6 - Node-Red IDE**

Foi utilizado o Node-Red ([Saiba mais](#)) para programar a “Internet das coisas” no projeto. Foi programado para quando um objeto estiver próximo do sensor de distância e o motor for acionado, no mesmo momento será enviado uma mensagem para o telegram através de um BOT chamado SmartCane\_Bot.

Para configurar o Node-Red é necessário realizar o download e configuração do NodeJS ([Download/Configuração](#)). Após ser baixado e configurado, é necessário executar:

Abra o Prompt de comando e Digite Node-Red, após a inicialização acesse o servidor localhost:1880.

```
node-red
See https://github.com/yagop/node-telegram-bot-api/issues/319. module.js:652:30
29 May 01:07:43 - [warn] rpi-srf : Ignoring Raspberry Pi specific node
29 May 01:07:43 - [warn] -----
29 May 01:07:43 - [warn] [node-red/rpi-gpio] Info : Ignoring Raspberry Pi specific node
29 May 01:07:43 - [warn] [node-red/tail] Not currently supported on Windows.
29 May 01:07:43 - [warn] -----
29 May 01:07:43 - [info] Settings file : \Users\Daniel\.node-red\settings.js
29 May 01:07:43 - [info] User directory : \Users\Daniel\.node-red
29 May 01:07:43 - [warn] Projects disabled : editorTheme.projects.enabled=false
29 May 01:07:43 - [info] Flows file : \Users\Daniel\.node-red\flows_DESKTOP-662IH0E.json
29 May 01:07:43 - [info] Server now running at http://127.0.0.1:1880/
29 May 01:07:43 - [warn] -----

Your flow credentials file is encrypted using a system-generated key.

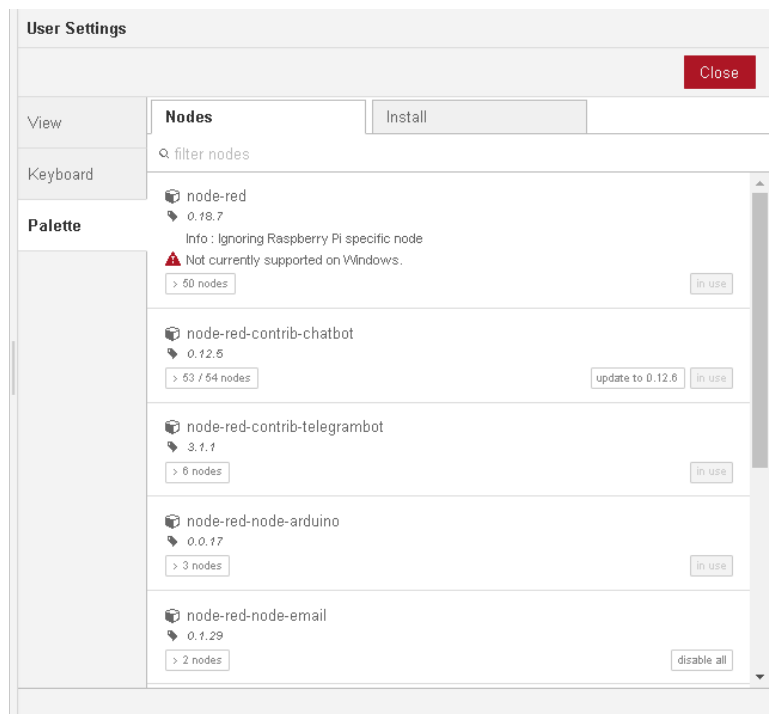
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

29 May 01:07:43 - [info] Starting flows
29 May 01:07:43 - [error] [arduino-board:9b429007.875ff] TypeError: Cannot read property 'on' of null
29 May 01:07:43 - [error] [arduino-board:68031305.664b8c] TypeError: Cannot read property 'on' of null
29 May 01:07:43 - [info] Started flows
```

**Figura 7 - CMD conexão com o servidor**

Ao acessar a página o servidor é necessário instalar duas bibliotecas. **Menu > Manage palette > Install**



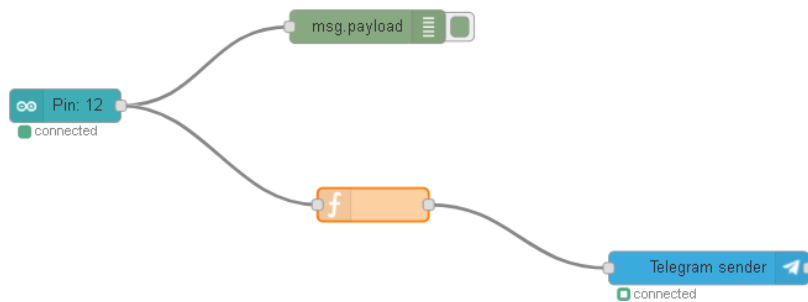
**Figura 8 - Gerenciador de bibliotecas Node-Red**

Foi utilizado duas bibliotecas no node-red, são elas:

**node-red-contrib-telegrambot** - Este pacote contém um receptor e um nó emissor que atuam como um bot do telegram. A única coisa necessária é o token que pode ser recuperado pelo bot de telegrama, o @botfather. <https://core.telegram.org/bots>

**node-red-node-arduino** - Um nó Node-RED para se comunicar com um Arduino através do padrão firmata.

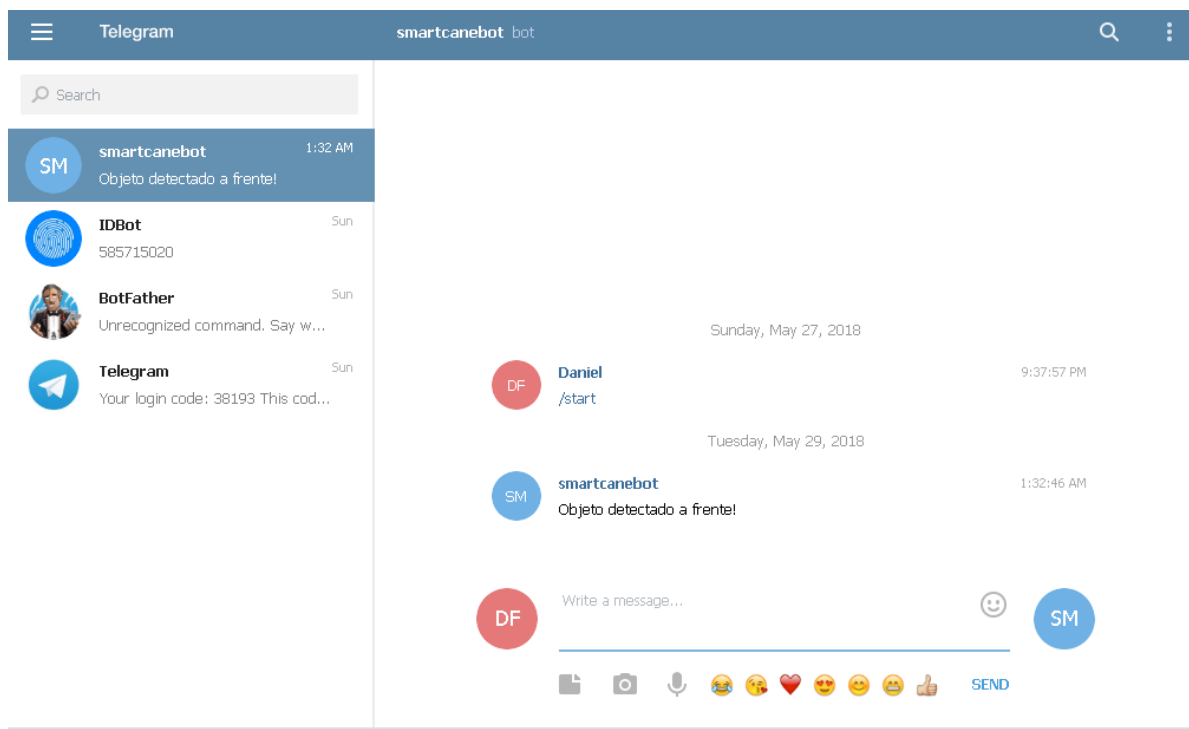
Foi node-red foi montado da seguinte forma:



**Figura 9 - Ligações Node-Red**

- ▮ **Arduino IN**
- ▮ **Function**
- ▮ **Output Debug**
- ▮ **Telegram Sender**

Com o Arduino conectado e o deploy realizado, quando o sensor for acionado o bot enviará uma mensagem automaticamente:



**Figura 10 - Mensagem via telegram**

