

Projeto Câmara Fria

Objetivo do projeto:

O objetivo do projeto é trazer dados via aplicação *mobile* de temperatura do freezer qual o usuário possa ter controle onde estiver. Seja para negócios ou residencial.

A câmara fria tem como seu objetivo trazer acesso em tempo real via aplicativo a temperatura da mesma. Com uma UI(*user interface*) minimalista e resiliente, pode-se obter a temperatura de uma geladeira e ou freezer. Além disso, existe também uma interface física que pode alegar o estado da temperatura visualmente.

Seu funcionamento:

É possível realizar a detecção de temperatura e com *leds*, realizar toda a reação visual, trazendo uma interface de interação visual. Por meio do aplicativo, também pode-se obter acesso aos dados, agora também tendo uma interação digital.

Quando a temperatura sobe, é acionado o *led* vermelho, demonstrando que a temperatura está alta e, quando baixa, o *led* verde é acionado mostrando que a temperatura atual é gelada. Para satisfazer a câmara fria, é também acionado um *led* de lâmpada quando está ativa no ambiente.

Já o sensor de temperatura, deve estar sempre em contato com a câmara fria pois é por meio deste que irá detectar a temperatura e levar os dados ao hardware (microcontrolador). Através da programação do mesmo, ocorre o tratamento desses dados a fim de satisfazer a aplicação *mobile*, levando todas as informações necessárias como, qual *led* ligou, qual temperatura está, se está adequado ou não, se está frio ou quente.

Modo de operação:

Para que aconteça essa detecção de temperatura e toda a proposta previamente abordada, é usado um microcontrolador, no caso deste projeto, o Arduino Uno que, irá receber todos os dados detectados por periféricos a ele conectados e enviar os dados via porta serial para o *node-red* realizar a análise dos mesmos. Os periféricos anteriormente citados, seriam componentes como sensor de temperatura, responsável por detectar a temperatura da câmara e, *leds*, responsáveis por trazer a informação visual a quem usa.

Ainda que se chame câmara fria, o medidor de temperatura além de pegar temperaturas mínimas, também pega as máximas. E com o envio desses dados ao microcontrolador Arduino Uno, é possível trabalhar através do *software* NodeRed toda a obtenção dessa informação e também manipulação da mesma através da criação de dashboards que serão integradas ao aplicativo em questão. Com o auxílio de *leds*, é possível obter visão sobre o estado da temperatura (quente ou frio) e, também para iluminar

o interior da câmara, facilitando seu uso.

- Programação:

No NodeRed, foi feito um fluxo de nodos que compõem basicamente a base de funcionamento de todo o sistema. Dentro desse fluxo, foram dispostos elementos para conectar arduino e fazer a leitura via porta serial, que faz a conexão com os outros componentes como as leds e o sensor. Também, foram inseridos elementos para a conexão com o firebase, e-mail e uma planilha de dados.

Primeiramente, os dados do sensor de temperatura provenientes do Arduino são convertidos e modificados para apresentar e gravar esses dados. De acordo com o valor, esses dados são enviados via serial para o Arduino acender as leds e mostrar visualmente ao usuário o estado da temperatura. Outro dado em questão, que é possível a interação do usuário, é o ato de acender ou apagar a luz no ambiente do projeto. Isso foi simulado com uma led amarela.

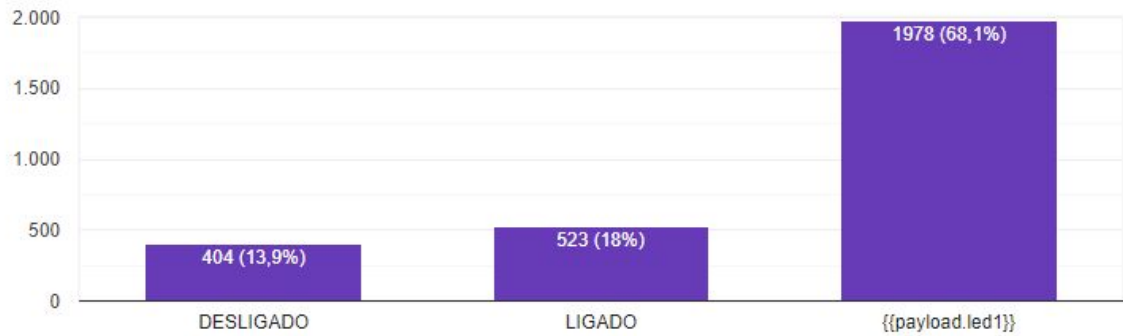
Na imagem abaixo, é possível ver os atributos com seus respectivos valores sincronizados com o node-red, por meio a url. O atributos foram nomeados da seguinte forma: lâmpada indica o valor da iluminação do ambiente (led amarela), led1 indica o valor da led de indicação do status da temperatura abaixo do limite (led verde), led2 indica o valor da led de indicação do status da temperatura acima do limite (led vermelha), e a temperatura indica o valor proveniente do sensor de temperatura.



Em seguida, é possível visualizar os gráficos e valores obtidos por meio da integração, nas imagens abaixo do resultado dos formulários e planilha.

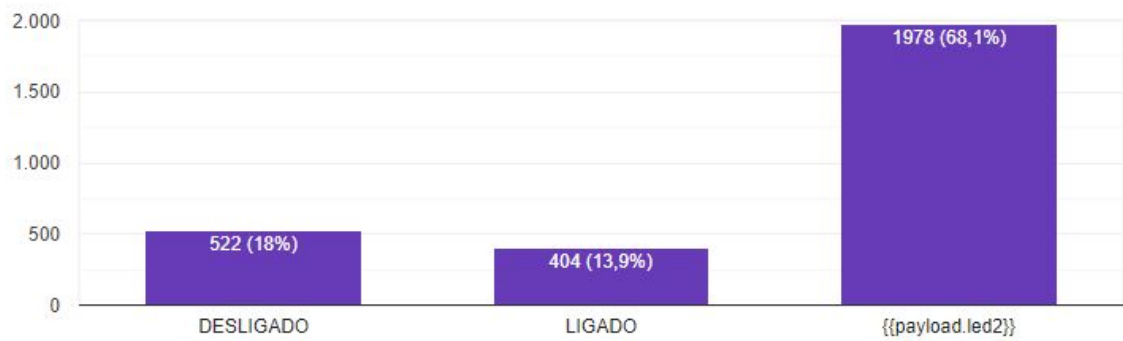
led1

2.905 respostas



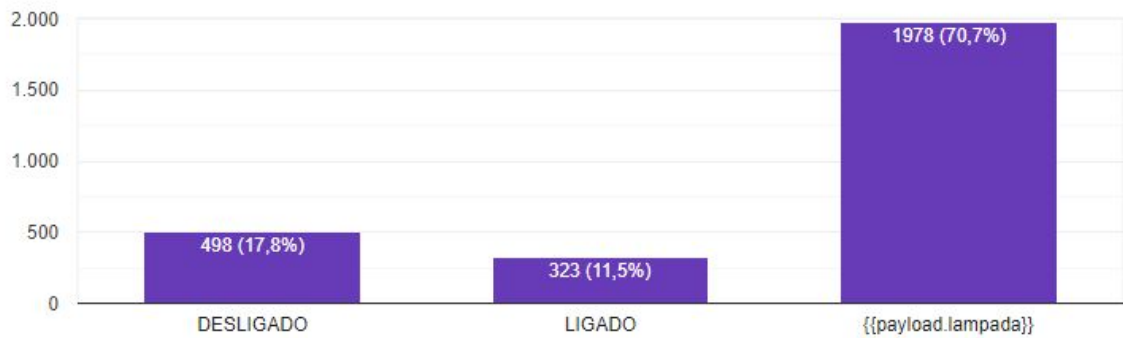
led2

2.904 respostas



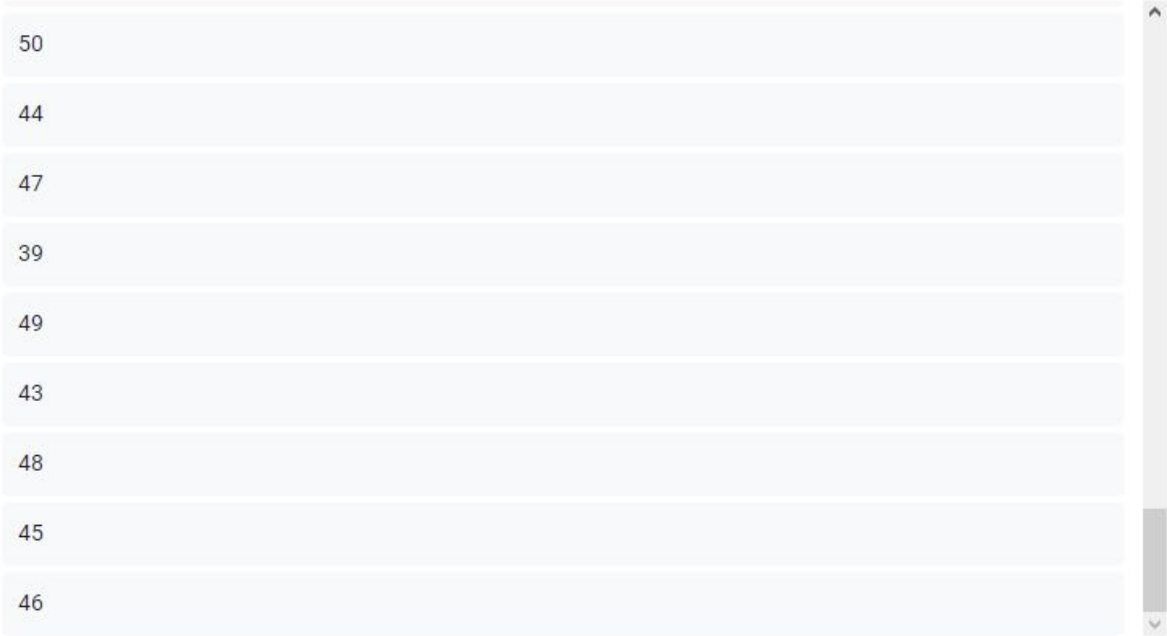
lampada

2.799 respostas



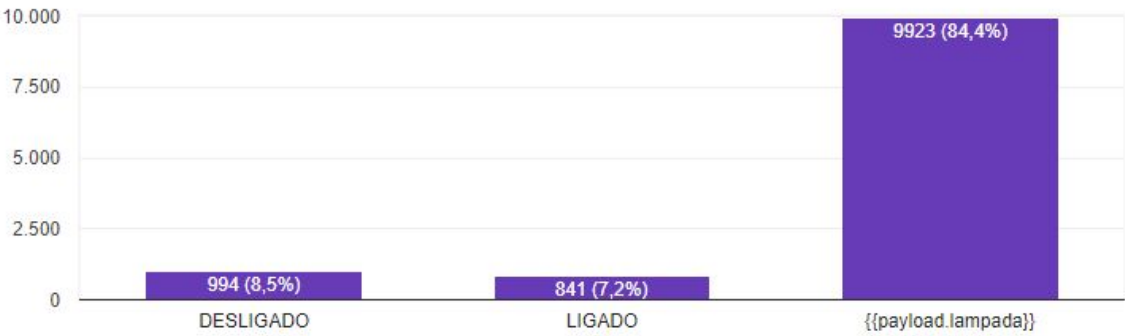
temperatura

3.650 respostas



lampada

11.758 respostas



Formulário Câmara Fria (respostas) ☆ 📄 ☁

Arquivo Editar Ver Inserir Formatar Dados Ferramentas Formulário Complementos Ajuda

100% R\$ % .0 .00 123 Padrão (Ari... 10 B I S A 🔍 📊 📅

	A	B	C	D	E
1	Carimbo de data/hora	led1	led2	temperatura	lampada
73	12/09/2020 23:33:07	LIGADO	DESLIGADO		8 LIGADO
74	12/09/2020 23:33:08	LIGADO	DESLIGADO		6 LIGADO
75	12/09/2020 23:33:16	LIGADO	DESLIGADO		5 LIGADO
76	12/09/2020 23:33:17	LIGADO	DESLIGADO		8 LIGADO
77	12/09/2020 23:33:17	LIGADO	DESLIGADO		9 LIGADO
78	12/09/2020 23:33:19	LIGADO	DESLIGADO		8 LIGADO
79	12/09/2020 23:33:20	LIGADO	DESLIGADO		11 LIGADO
80	12/09/2020 23:37:50	LIGADO	DESLIGADO		11 LIGADO
81	12/09/2020 23:42:02	LIGADO	DESLIGADO		11 LIGADO
82	12/09/2020 23:48:55	LIGADO	DESLIGADO		11 LIGADO
83	12/09/2020 23:49:02	LIGADO	DESLIGADO		12 LIGADO
84	12/09/2020 23:49:02	LIGADO	DESLIGADO		9 LIGADO
85	12/09/2020 23:49:14	LIGADO	DESLIGADO		6 LIGADO
86	12/09/2020 23:49:15	LIGADO	DESLIGADO		9 LIGADO
87	12/09/2020 23:49:15	LIGADO	DESLIGADO		10 LIGADO

Foi desenvolvido um app mobile com Xamarin para o monitoramento dos dados do ambiente de escolha do usuário conforme citado anteriormente. Esses dados são capturados em tempo real da ferramenta Firebase, assim é possível ter controle sobre os dados de temperatura, e se esse valor superou o valor máximo que o ambiente deve ter ou não. Também, é possível acionar e desligar a luz do local ao clicar no ícone de lâmparina na parte superior da interface.

Os dados foram capturados em formato json e convertidos para serem apresentados na tela, assim como na requisição para alterar o valor de acionamento e o desligamento da lâmpada do ambiente em questão.

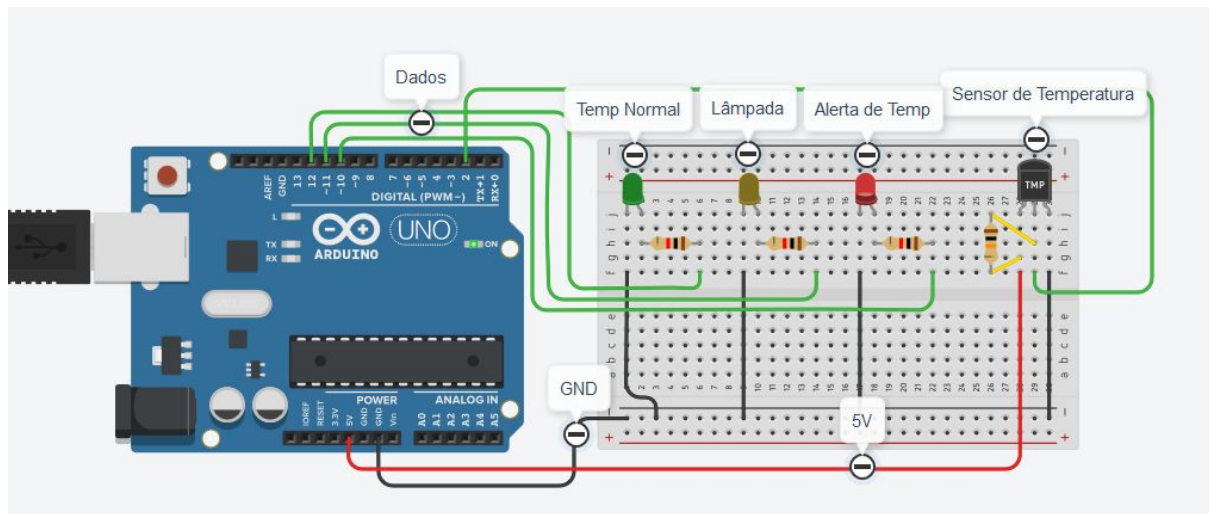
Esquemático de montagem:

Na esquemática abaixo, é possível notar como cada componente deve estar conectado para satisfazer seu propósito. Na mesma é ilustrado o arduino como fonte alimentando a placa de prototipação pela saída de 5V, além do GND via Jumpers.

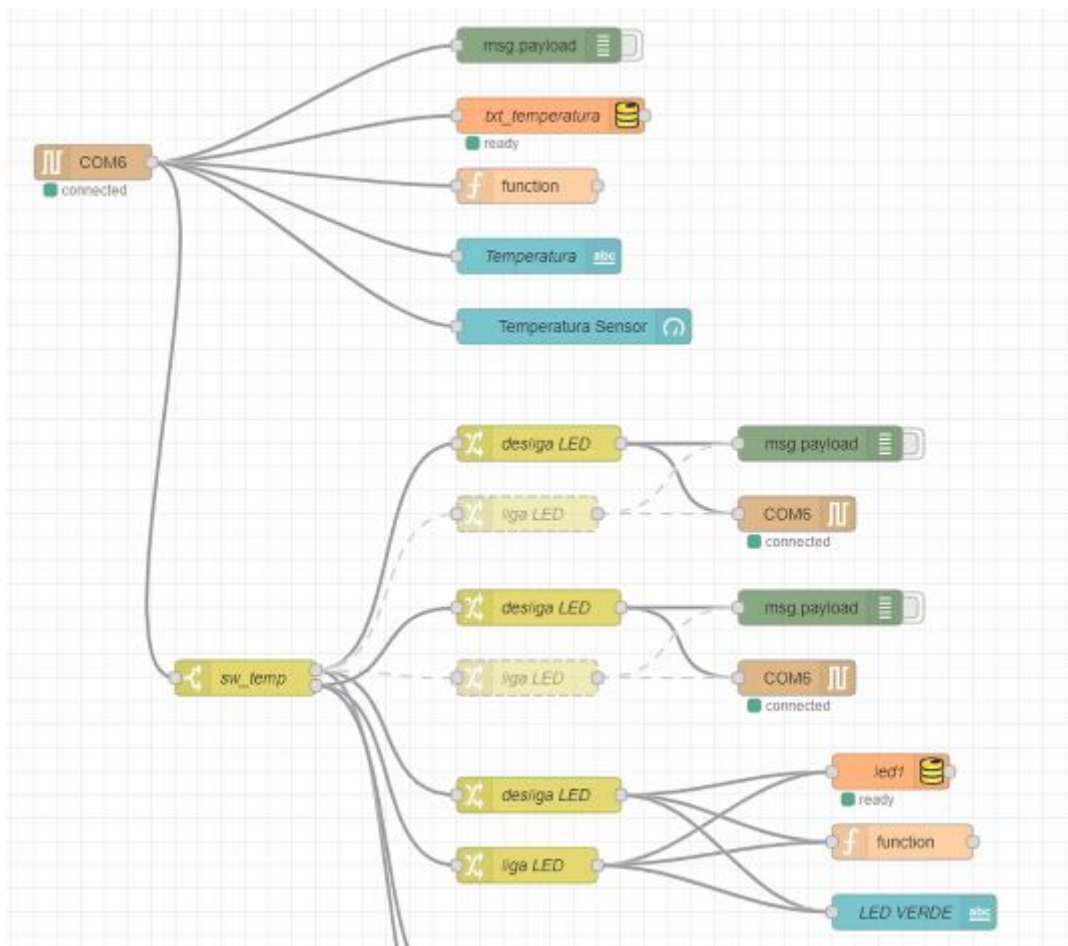
Realizando ligações pela própria placa de prototipação, é possível distribuir a energia para o sensor de temperatura, assim como atribuir o GND (*ground*) ao componente. Além das ligações necessárias para ligar o componente sensor, também é necessário conectá-lo ao I/O do Arduino Uno, na imagem abaixo, é ilustrado que o cabo de dados em verde é ligado em uma das pontas na porta 2 (*digital 2*), uma porta digital, e na outra até o

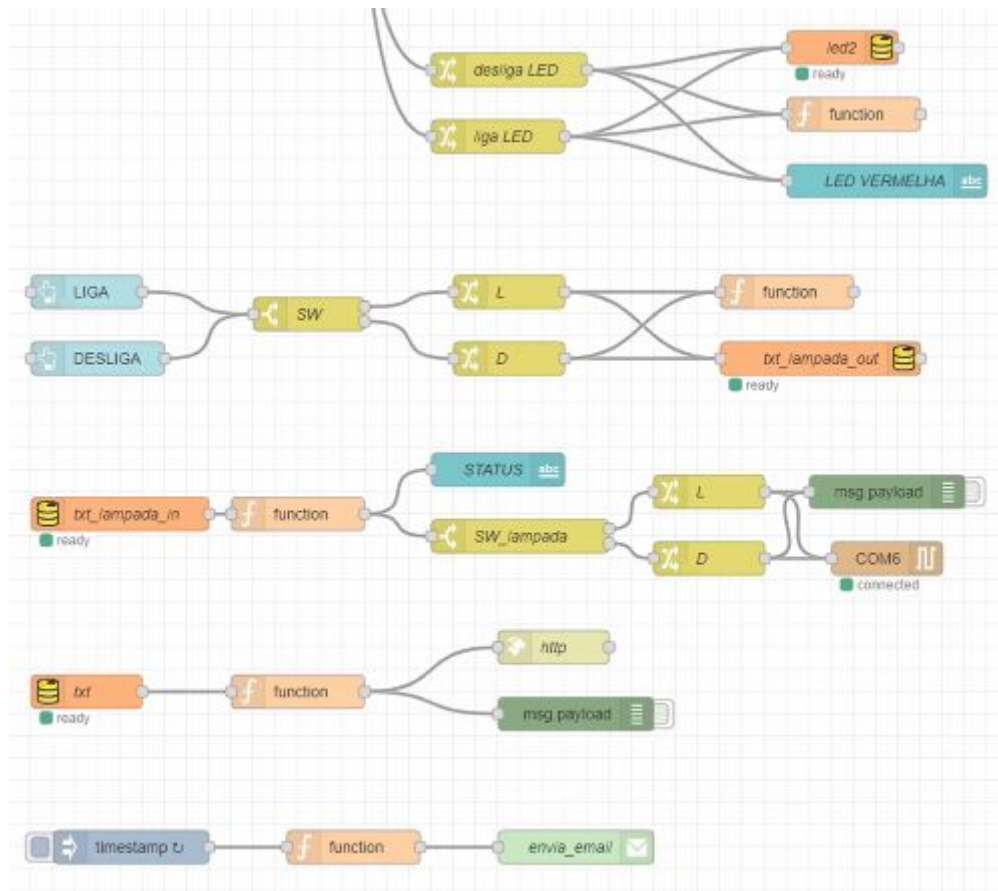
pino que irá liberar os dados no sensor. Conectando também um resistor de 220Ω entre a entrada de 5V e a saída de dados do sensor para garantir integridade aos dados.

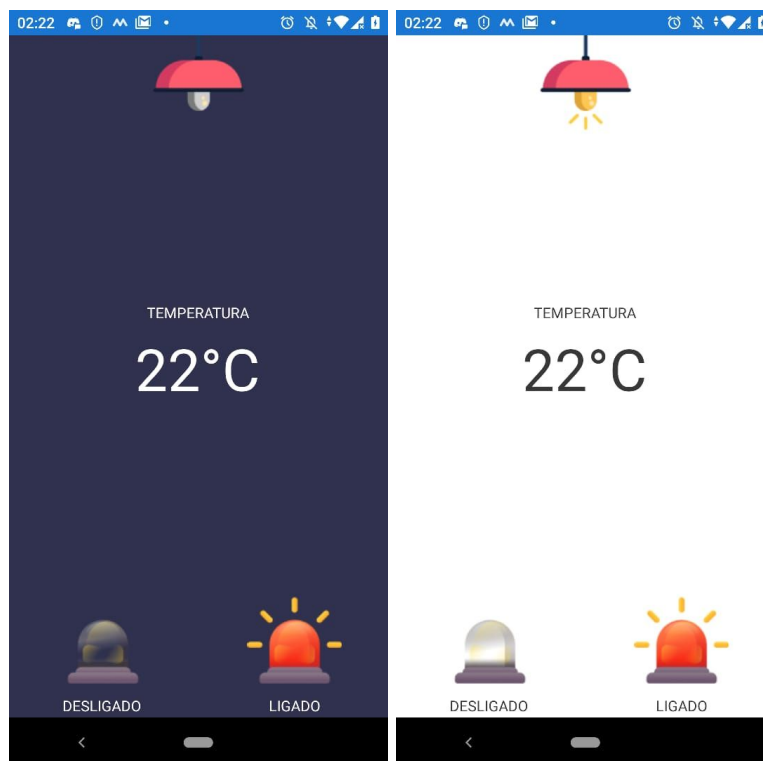
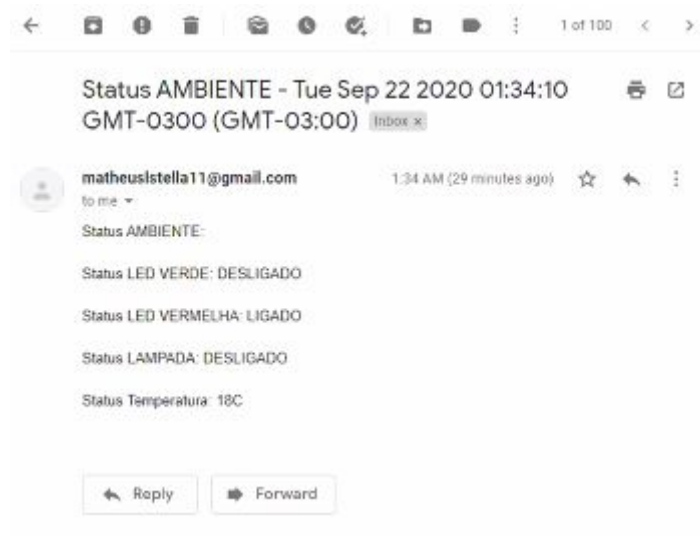
Para ligar os *leds*, expande-se a conexão do GND por jumpers conectado à resistores de 220Ω no *Catodo* e, n o *Anodo*, cabos de conexão direta de dados pelas I/O 12,11 e 10 do Arduino Uno que também irá mandar carga para o mesmo.

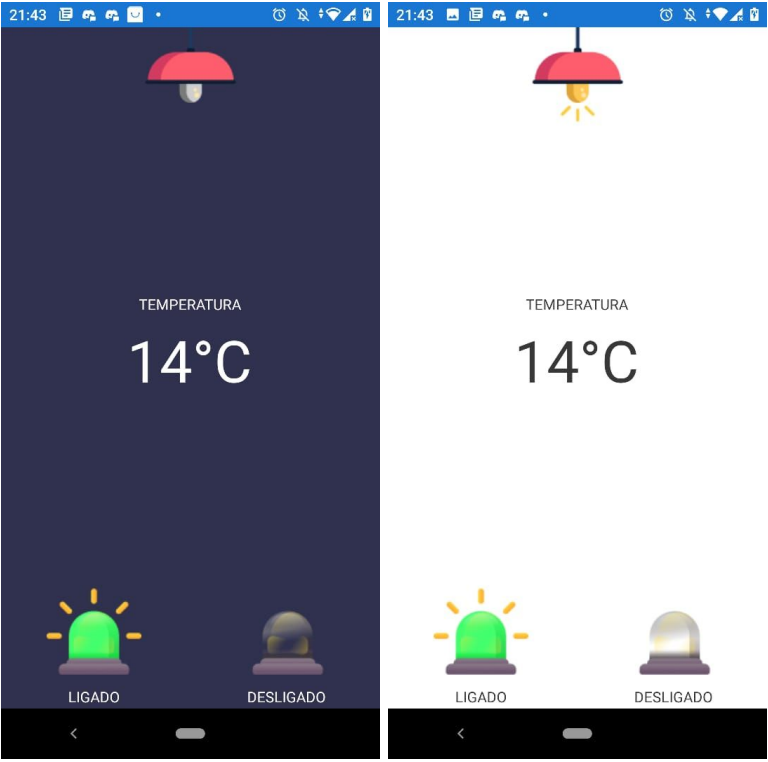


- Fotos do projeto:









```

#include <DHT.h>
#define DHTPIN 2    // what pin we're connected to
#define DHTTYPE DHT11 // DHT 11 (AM2302)
// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);

int Lvermelha = 10;
int Lverde = 12;
int Lamarela = 11;
String a;
float temp; //Stores temperature value

void setup()
{
  Serial.begin(9600);
  dht.begin();

  pinMode(Lvermelha, OUTPUT);
  pinMode(Lverde, OUTPUT);
  pinMode(Lamarela, OUTPUT);
}
String templ;
void loop()
{
  temp = dht.readTemperature();
  templ = String(temp);
  Serial.println(templ);
  delay(200); //Delay 2 sec.

  while (Serial.available() > 0 ) {
    a = Serial.readString(); // read the incoming data as string
    if (a == "1") {
      digitalWrite(Lvermelha, 1);
      digitalWrite(Lverde, 0);
    } else if (a == "2") {
      digitalWrite(Lvermelha, 0);
      digitalWrite(Lverde, 1);
    }

    if (a == "3") {
      digitalWrite(Lamarela, 1);
    } else if (a == "4") {
      digitalWrite(Lamarela, 0);
    }
  }
}

```

