ENGENHARIA DA COMPUTAÇÃO

Programação e IOT

Projeto Final

TINKERCAD – THINGSPEAK – MIT INVENTOR

Gabriel Lucas de Arruda RA:200840;
Paulo Marcos Araujo da Rocha RA:200872;
Lucas Vinicius Tomé RA:200354;

Prof.: André Breda Carneiro, MSc.

Sorocaba / SP

SUMÁRIO

1. Objetivo

O presente relatório tem como objetivo a apresentação do projeto final de programação e IOT, relacionando o projeto as ferramentas de simulação virtual Tinkercad e ThingSpeak. E por conseguinte realizar a criação do app do projeto por meio do Mit Inventor. Nesse aspecto realizar a conectividade e interação de todas no meio virtual.

Diante de tal contexto, a equipe criou um projeto que auxiliará na vida das pessoas o deixando mais informada a respeito da situação meteorológica ao redor da sua residência com auxílio de sensores, e por conseguinte criamos conexões por meio de gráficos no ThingSpeak que será mostrado no APP desenvolvido pela equipe no ambiente virtual Mit Inventor, facilitando atravéis do mesmo a leitura de informações por meio do usuário. Tais utilidades no projeto é a facilidade de ler informações gráficas a respeito da temperatura, umidade e se há lâmpadas ligadas na residência.

2. Materiais utilizados

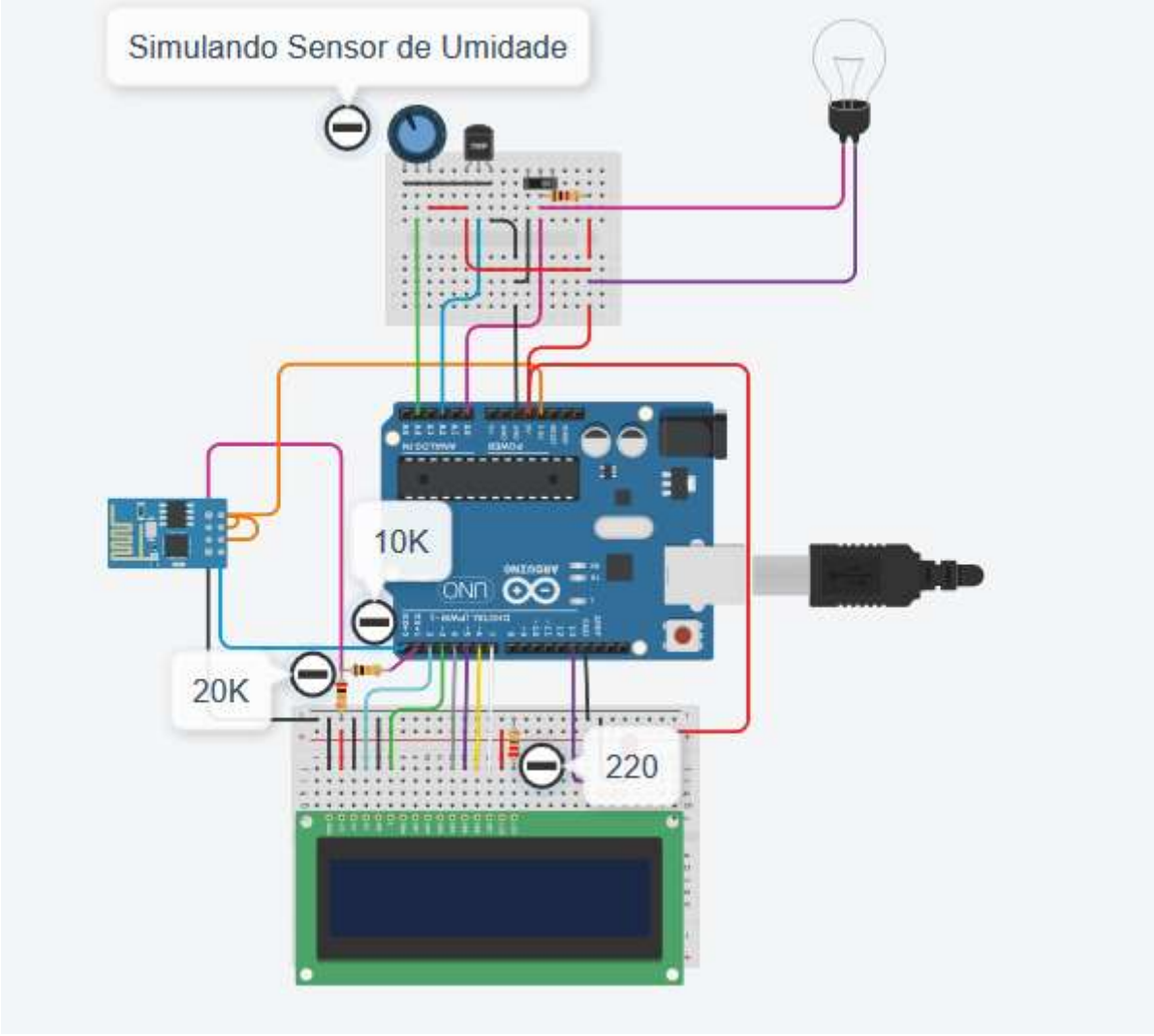
- Computador
- Internet
- Tinkercad
- ThingSpeak
- Mit Inventor

4. Procedimentos

Em primeiro plano, realizamos a montagem no Tinkercad do projeto (Figura 1.0) adicionando o sensor de temperatura [TMP36], um interruptor deslizante para controlar a lâmpada do circuito, resistores, o módulo WIFI [ESP8266], um Display LCD para mostrar a temperatura e umidade, também mostrar no Display [OFF] para

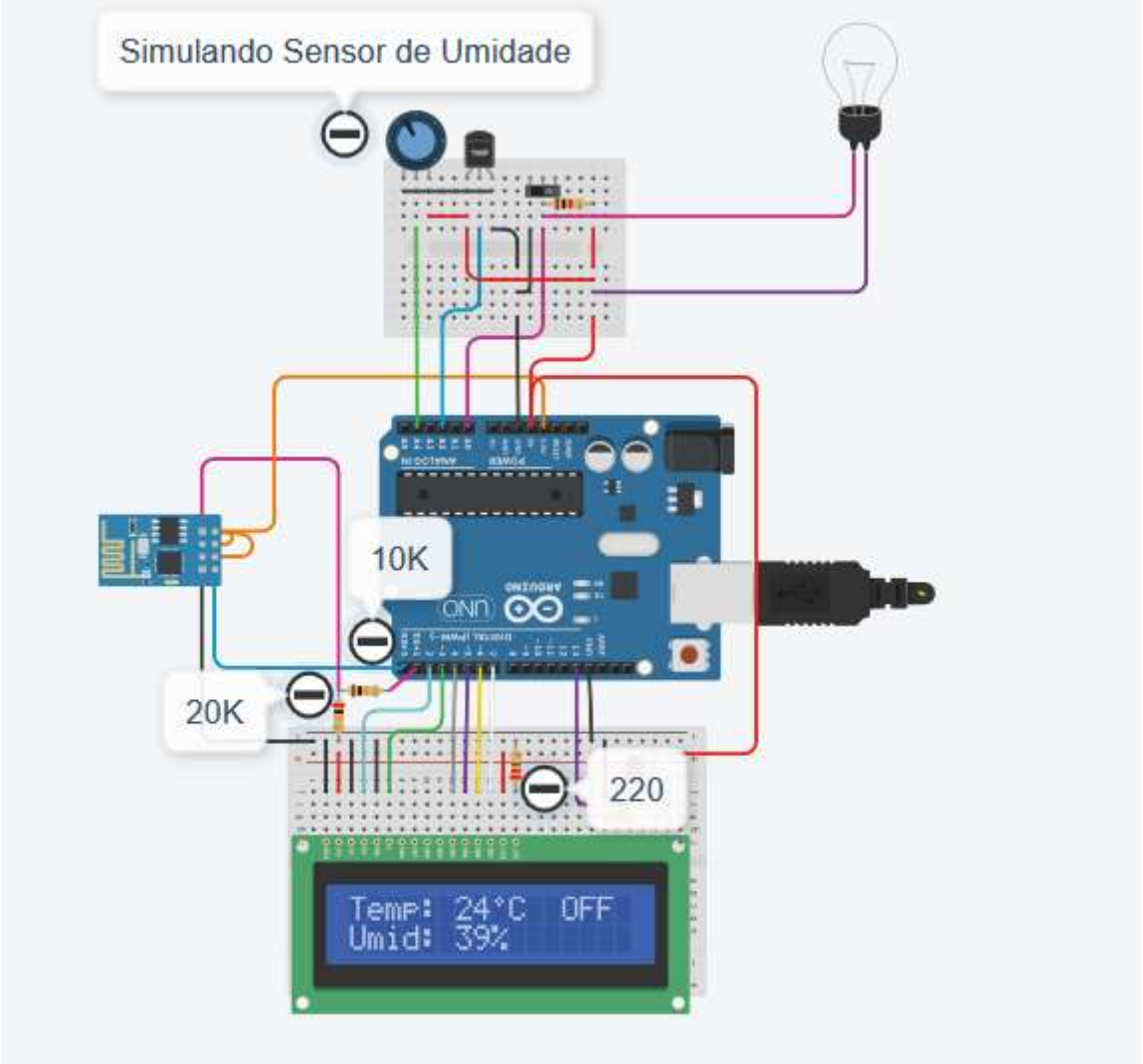
lâmpada desligada (Figura 1.2) ou [ON] para lâmpada ligada (Figura 1.3) e por fim um potenciômetro que simulará um sensor de umidade, visto que a plataforma Tinkercad não possui tal componente eletrônico.

Figura 1.0 - Projeto Montado



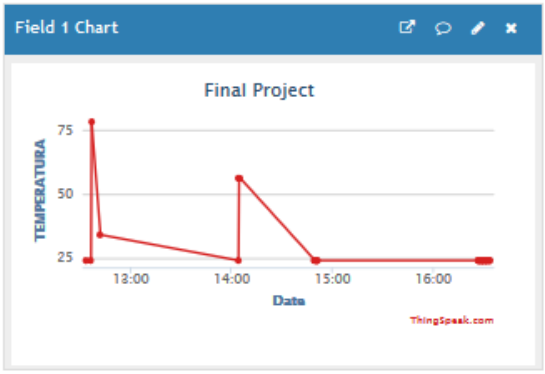
Fonte : https:www.tinkercad.com/

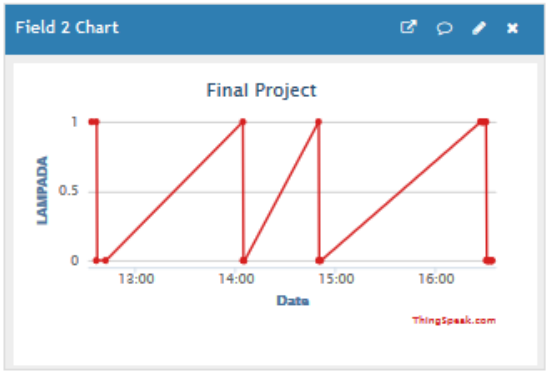
Figura 1.2 – Simulação A



Com o início da simulação observa-se que o circuito extraiu as informações dos sensores e do interruptor, mostrando dessa forma os dados no LCD e o enviando para o ThingSpeak (Figura 1.4) e em seguida enviando para o App desenvolvido pelo grupo (Figura 1.5).

Figura 1.4 – THINGSPEAK





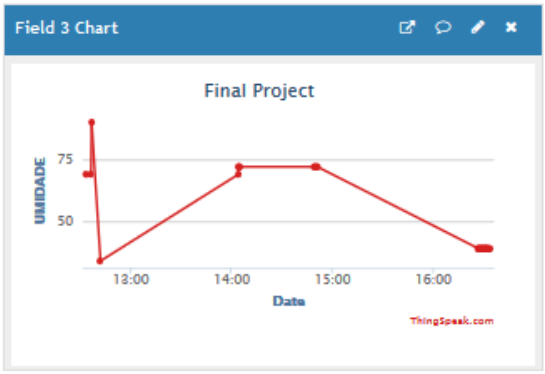


Figura 1.5 - Temperatura

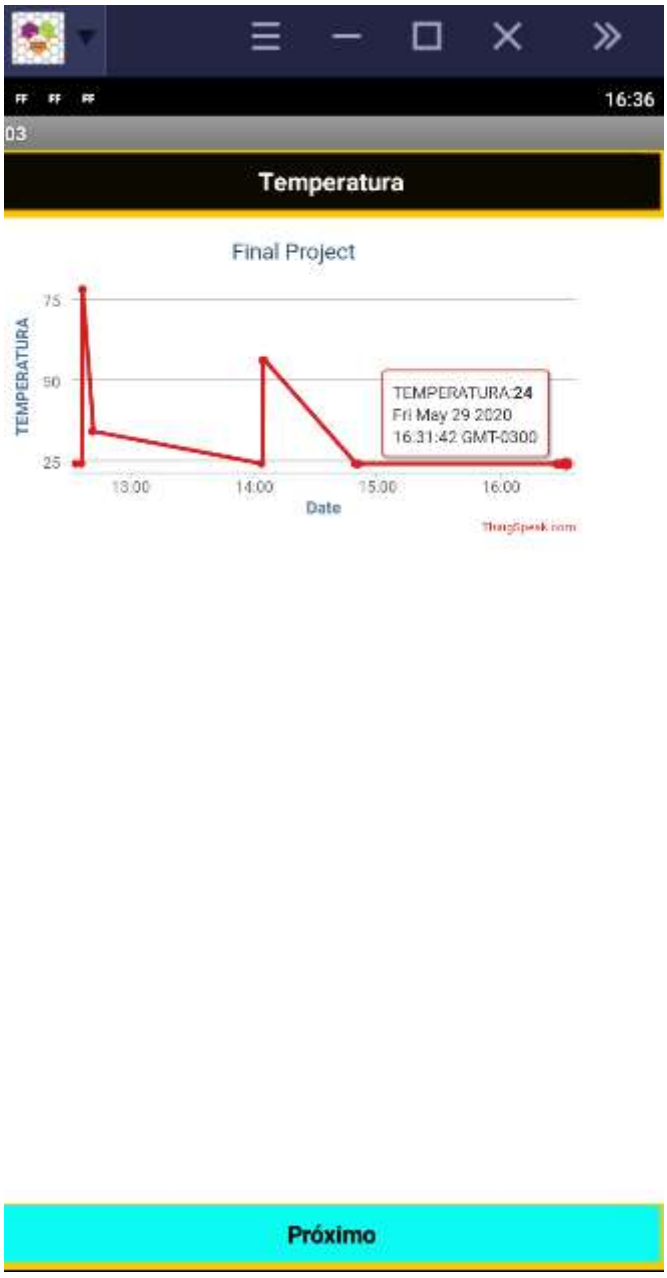


Figura 1.6 – Lâmpada

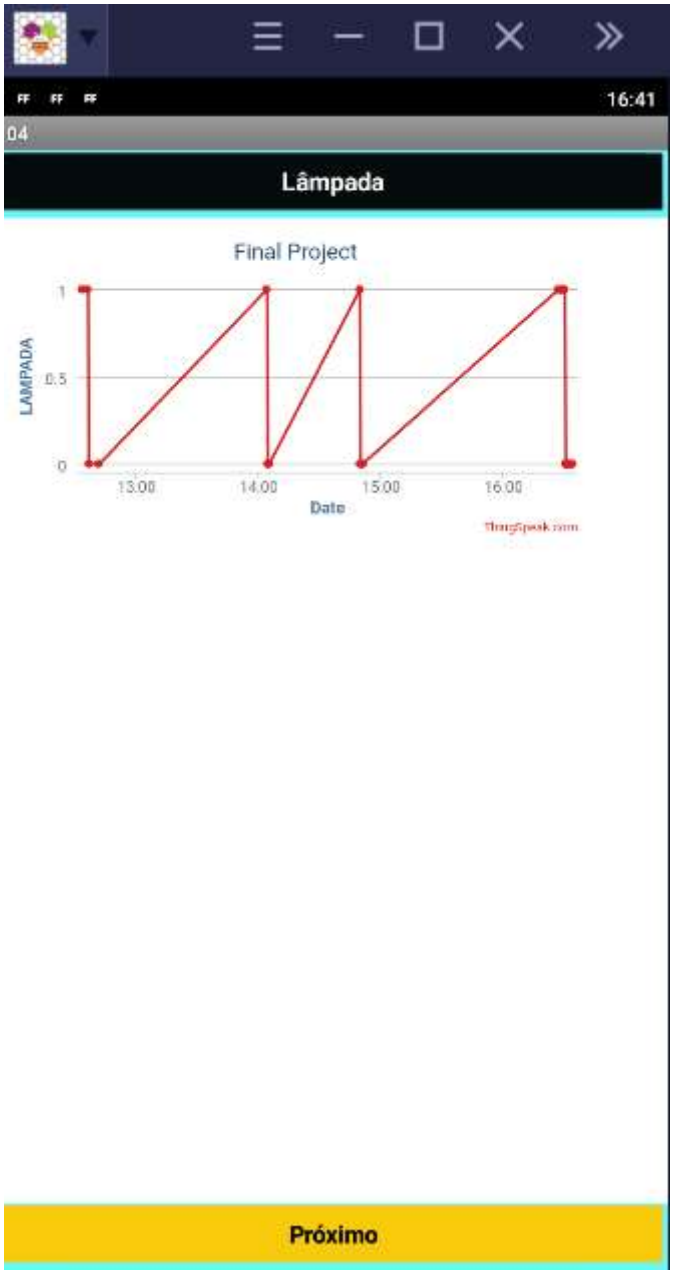
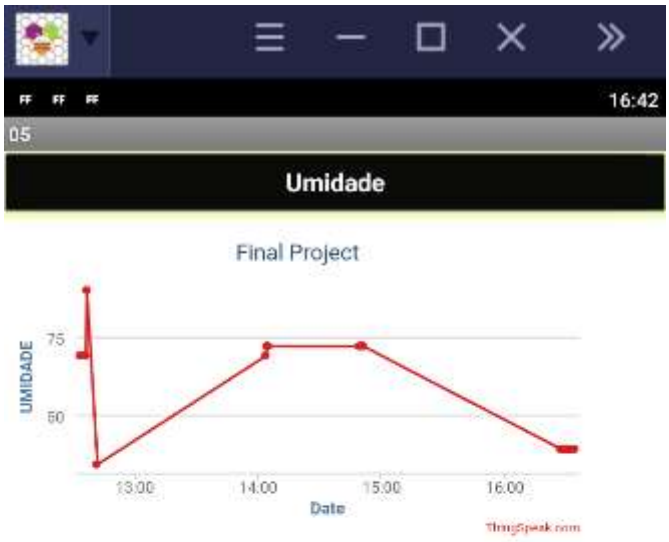
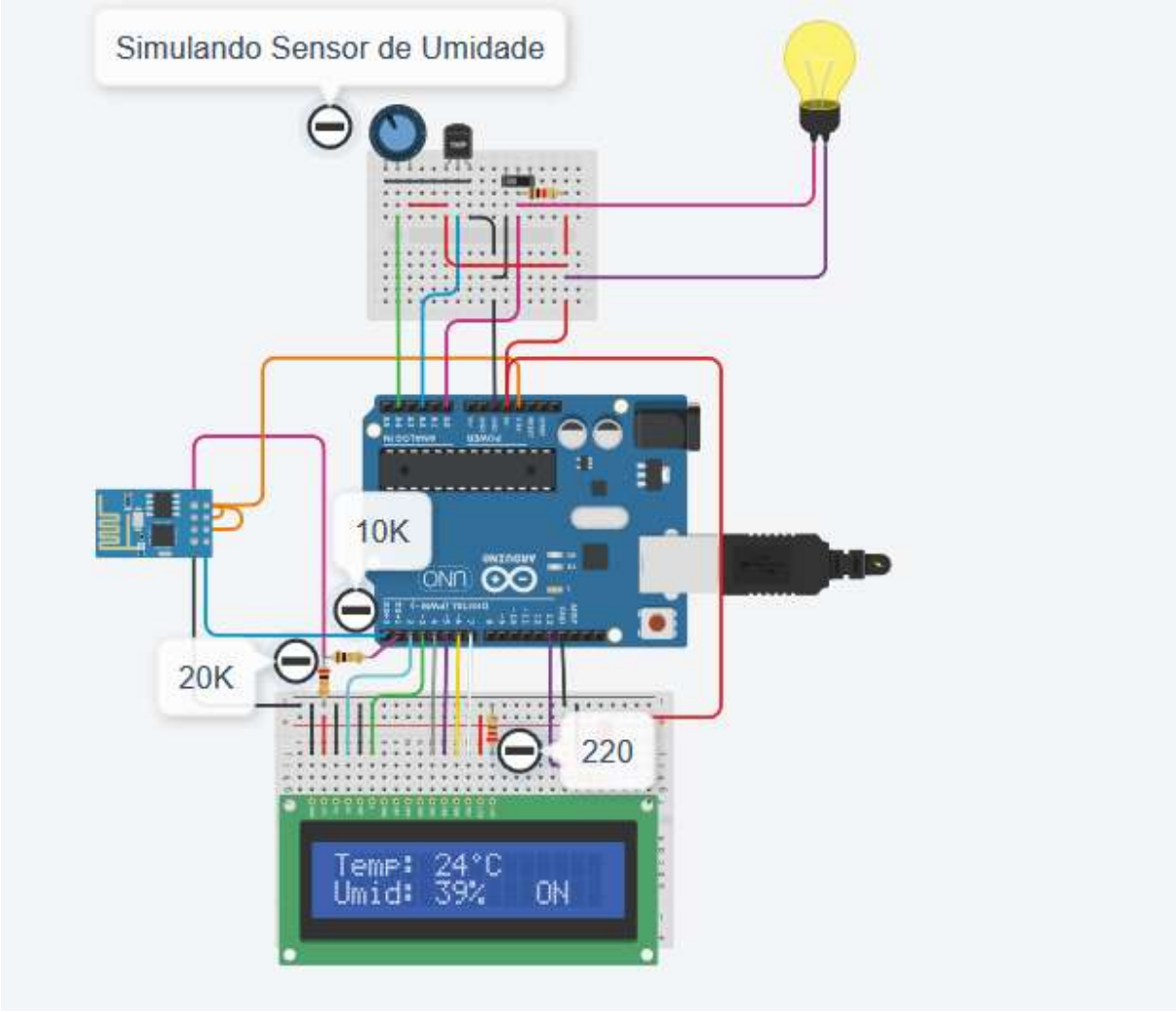


Figura 1.6 – Lâmpada



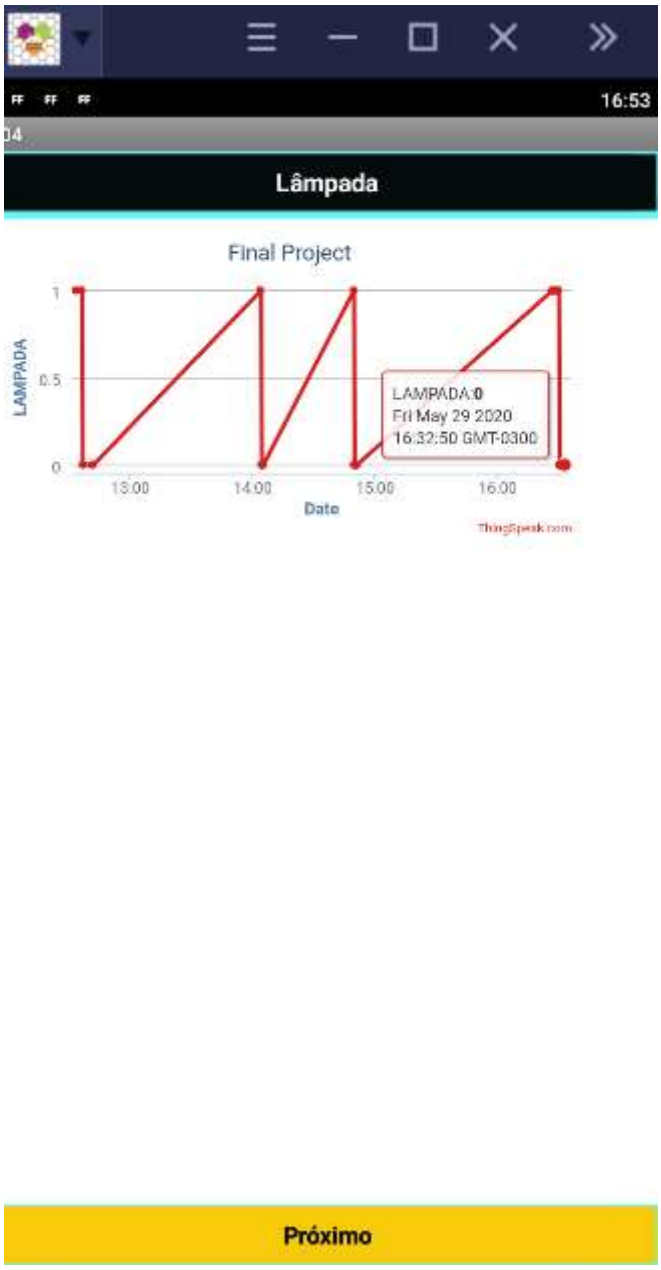
Voltar

Figura 1.3 – Simulação B



Analisando os dados obtidos no Tinkercad, quando a lâmpada está ligada irá aparecer no gráfico o valor igual a 0 (Figura 1.8) e quando a lâmpada estiver desliga irá aparecer o valor igual a 1.

Figura 1.8 – Lâmpada Ligada



```
#include <LiquidCrystal.h>

const int PINO_SENSOR_TEMPERATURA = A2;
const int PINO_SENSOR_NIVEL_DAGUA = A4;
const int PINO_SENSOR_LAMPADA_OPERANDO = A0;

const int PINO_LED_ERRO = 13;

const int PINO_LCD_RS = 2;
const int PINO_LCD_ENABLE = 3;
const int PINO_LCD_DB4 = 4;
const int PINO_LCD_DB5 = 5;
const int PINO_LCD_DB6 = 6;
const int PINO_LCD_DB7 = 7;

const int LCD_NUMERO_LINHAS = 2;
const int LCD_NUMERO_COLUNAS = 16;


LiquidCrystal _lcd(PINO_LCD_RS, PINO_LCD_ENABLE,
                   PINO_LCD_DB4, PINO_LCD_DB5, PINO_LCD_DB6, PINO

String _ssidName     = "Simulator Wifi";
String _ssidPassword = "";
int _tcpHttpPort     = 80;
```

```
26
27
28  String _siteHost    = "api.thingspeak.com";
29  String _siteAPPID   = "WJPE65B0OQBNRXXV";
30  String _siteURIbase = "/update?api_key=" + _siteAPPID;
31  String _siteField1  = "&field1=";
32  String _siteField2  = "&field2=";
33  String _siteField3  = "&field3=";
34
35
36  const int TEMPERATURA_CELSIUS_MINIMA = -40;
37  const int TEMPERATURA_CELSIUS_MAXIMA = 125;
38
39  int _leituraMinima;
40  int _leituraMaxima;
41
42
43  void setup() {
44
45    pinMode(PINO_LED_ERRO, OUTPUT);
46    digitalWrite(PINO_LED_ERRO, LOW);
47
48
```

```
49    pinMode(PINO_SENSOR_TEMPERATURA, INPUT);
50    pinMode(PINO_SENSOR_NIVEL_DAGUA, INPUT);
51    pinMode(PINO_SENSOR_LAMPADA_OPERANDO, INPUT);
52
53    _lcd.begin(LCD_NUMERO_COLUNAS,LCD_NUMERO_LINHAS);
54    _lcd.print("   Final Project");
55    _lcd.setCursor(0,1);
56    _lcd.print("Out oF Home");
57
58    _leituraMinima = 20;
59    _leituraMaxima = 358;
60
61    // Inicializa a comunicação com o ESP8266
62    Serial.begin(115200);
63    Serial.setTimeout(2000);
64    sendCommandTo8266("AT", "OK");     // Confirma que o ESP8266 es
65
66    // Conecta ao Simulator de WiFi usando o comando AT+CWJAP
67    String loginWiFi = "AT+CWJAP=\"" + _ssidName + "\",\"" + _ssid
68    sendCommandTo8266(loginWiFi, "OK");
69
70    // Abre o canal de comunicação TCP com o site usando o comando
71    String acessoTCP = "AT+CIPSTART=\"TCP\",\"" + _siteHost + "\",
72    sendCommandTo8266(acessoTCP, "OK");
```

```
73
74    delay(1000);
75
76    _leituraMinima = 20;
77    _leituraMaxima = 358;
78
79  }
80
81
82  // Atualiza a temperatura a cada 10 segundos
83  void loop() {
84    // Le o sensor de temperatura (precisão 4,88mV)
85    int leituraSensor = analogRead(PINO_SENSOR_TEMPERATURA);
86    // Executa a autocalibração
87    if (leituraSensor > _leituraMaxima)           _leituraMaxima =
88    if (leituraSensor < _leituraMinima)           _leituraMinima =
89    // Converte a leitura do sensor para temperatura Celsius
90    int temperaturaCelsius = map(leituraSensor,
91                                 _leituraMinima, _leituraMaxima,
92                                 TEMPERATURA_CELSIUS_MINIMA, TEMPER
93    String temperaturaCelsiusTexto = numberToString(temperaturaCels
```

```
94
95    // Sensor de nível da caixa d'água
96    int nivelCaixaDagua = analogRead(PINO_SENSOR_NIVEL_DAGUA);
97    int nivelCaixaDaguaPercente = map(nivelCaixaDagua,
98                                      0, 1023, 0, 100);
99    String nivelCaixaDaguaTexto = numberToString(nivelCaixaDaguaPer
100
101   // Sensor do motor
102   bool lampadaLigado = digitalRead(PINO_SENSOR_LAMPADA_OPERANDO);
103   String lampadaLigadoTexto = lampadaLigado ? "1" : "0";
104
105   // Enviar sensores para a nuvem
106   mostraDisplayLCD(temperaturaCelsiusTexto, nivelCaixaDaguaTexto,
107   enviarSensores(temperaturaCelsiusTexto, nivelCaixaDaguaTexto, l
108
109   // Enviar sensores a cada 1 segundo
110   delay(1000);
111 }
112
113 bool enviarSensores(String temperatura, String nivel, String lamp
114   bool sucesso = true;
115
116    // Constrói a requisição HTTP
117   String uriCompleta = _siteURIbase +
```

```
                        _siteField1 + temperatura +
                        _siteField2 + lampada +
                        _siteField3 + nivel;

  String httpPacket = "GET " + uriCompleta + " HTTP/1.1\r\nHost:
  int length = httpPacket.length();

  String tamanhoPacote = "AT+CIPSEND=" + numberToString(length);
  sucesso &= sendCommandTo8266(tamanhoPacote, ">");
  sucesso &= sendCommandTo8266(httpPacket, "SEND OK");

  return sucesso;
}

void mostraDisplayLCD(String temperatura, String nivel, bool lamp

  _lcd.setCursor(0,0);
  _lcd.print("Temp: ");
  _lcd.print(temperatura);
  _lcd.print('\xB2');
  _lcd.print("C  ");
```

```
  if (lampada)          _lcd.print("OFF      ");
  else                  _lcd.print("         ");

  _lcd.setCursor(0,1);
  _lcd.print("Umid: ");
  _lcd.print(nivel);
  _lcd.print("%   ");
  if (!lampada)         _lcd.print("ON     ");
  else                  _lcd.print("         ");
}

bool sendCommandTo8266(String comando, char * aguardar) {
  bool sucesso = false;


  Serial.println(comando);
  Serial.flush();
  delay(50);


```

```
159    if (0 == aguardar[0])                    sucesso = true;
160    else if (Serial.find(aguardar))          sucesso = true;
161    else                                     digitalWrite(PINO_LED_ERI
162      |
163    return sucesso;
164  }
165
166  String numberToString(int valor) {
167    char numero[6];
168    sprintf(numero, "%i", valor);
169    return String(numero);
170  }
171
```

2. Desenvolvimento do APP

Figura 2.0 - MIT inventor

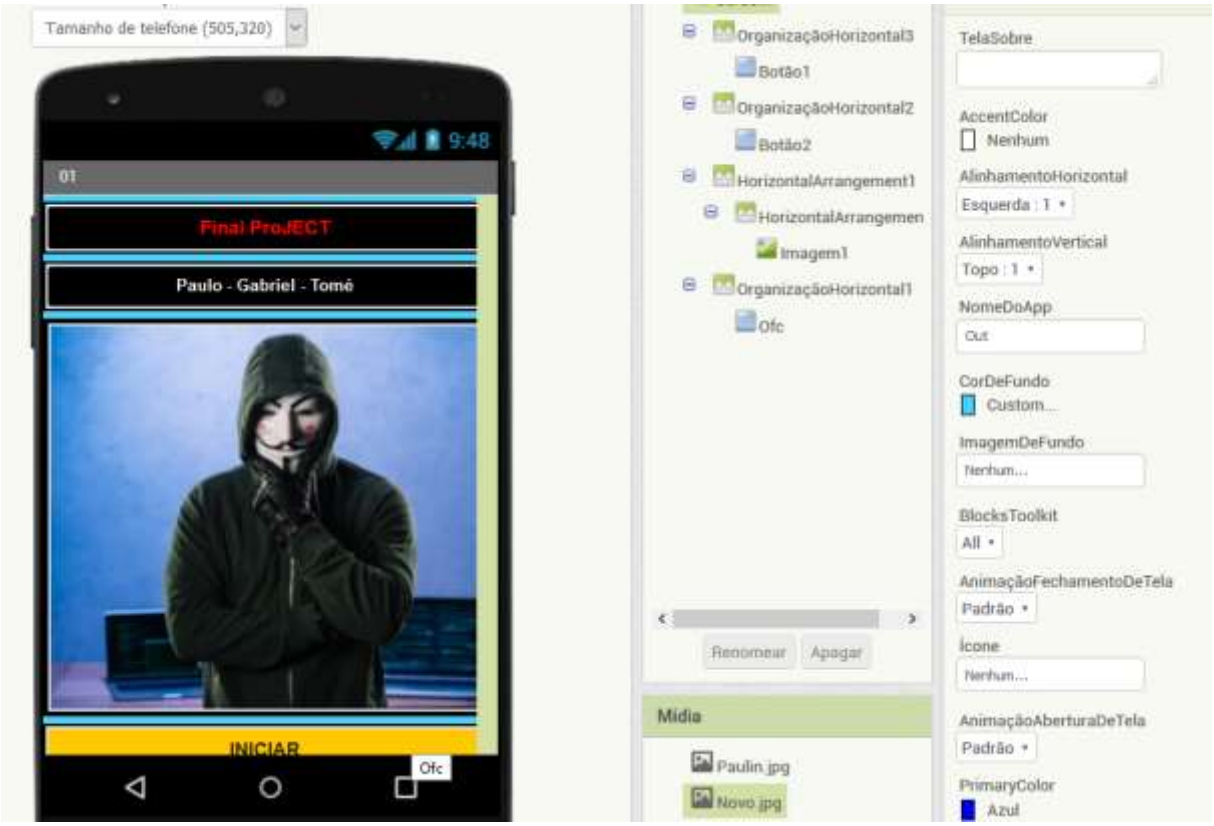
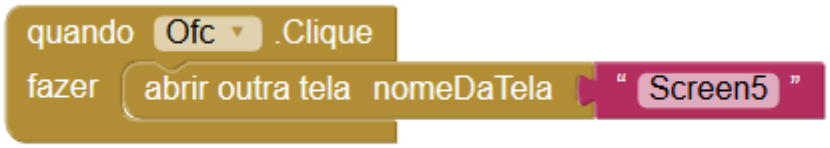


Figura 2.0 – Click Ofc



Quando clicar no Ofc, o app vai acessar a interface Screen5.

Figura 2.1 – Gráfico Temperatura

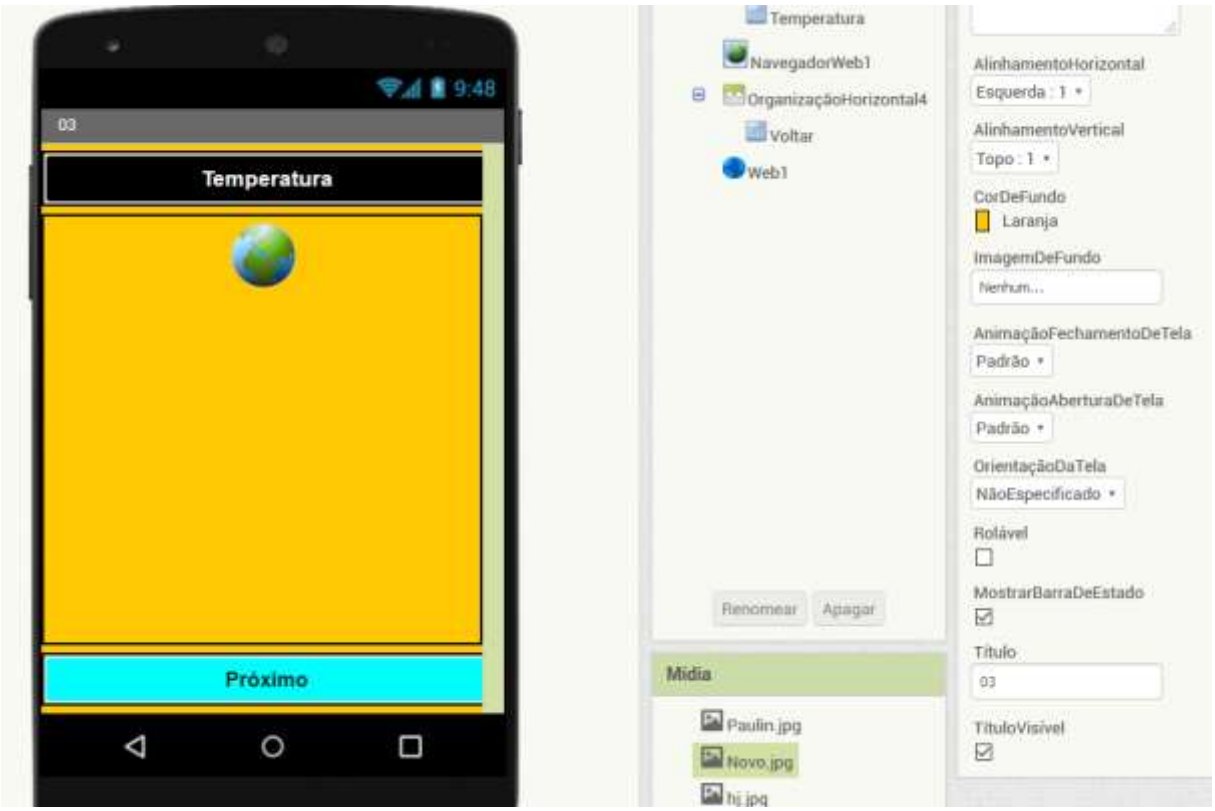
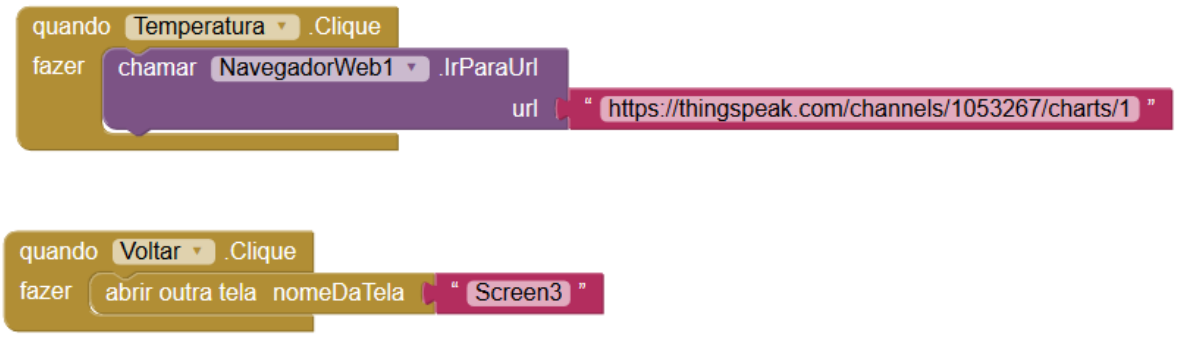


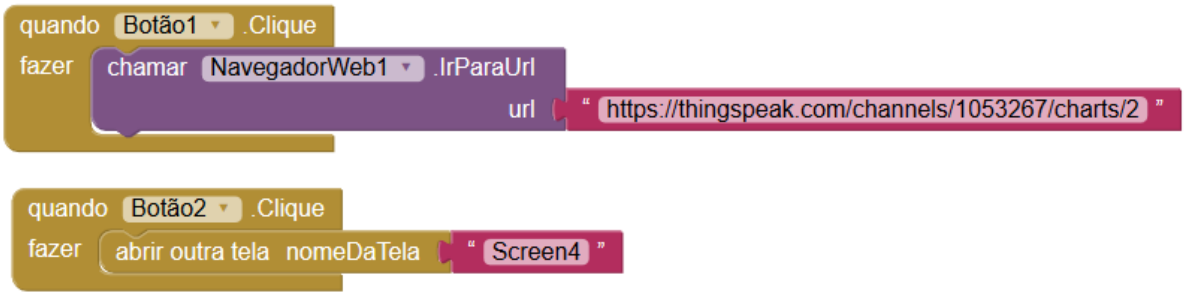
Figura 2 – Click Temperatura



Assim que clicar no Button temperatura, o app irá acessar a URL para buscar o field 1 e dessa maneira mostrar o gráfico da temperatura obtido a partir do ThingSpeak. Em seguida o click no Button voltar irá direcionar a interface Screen3.

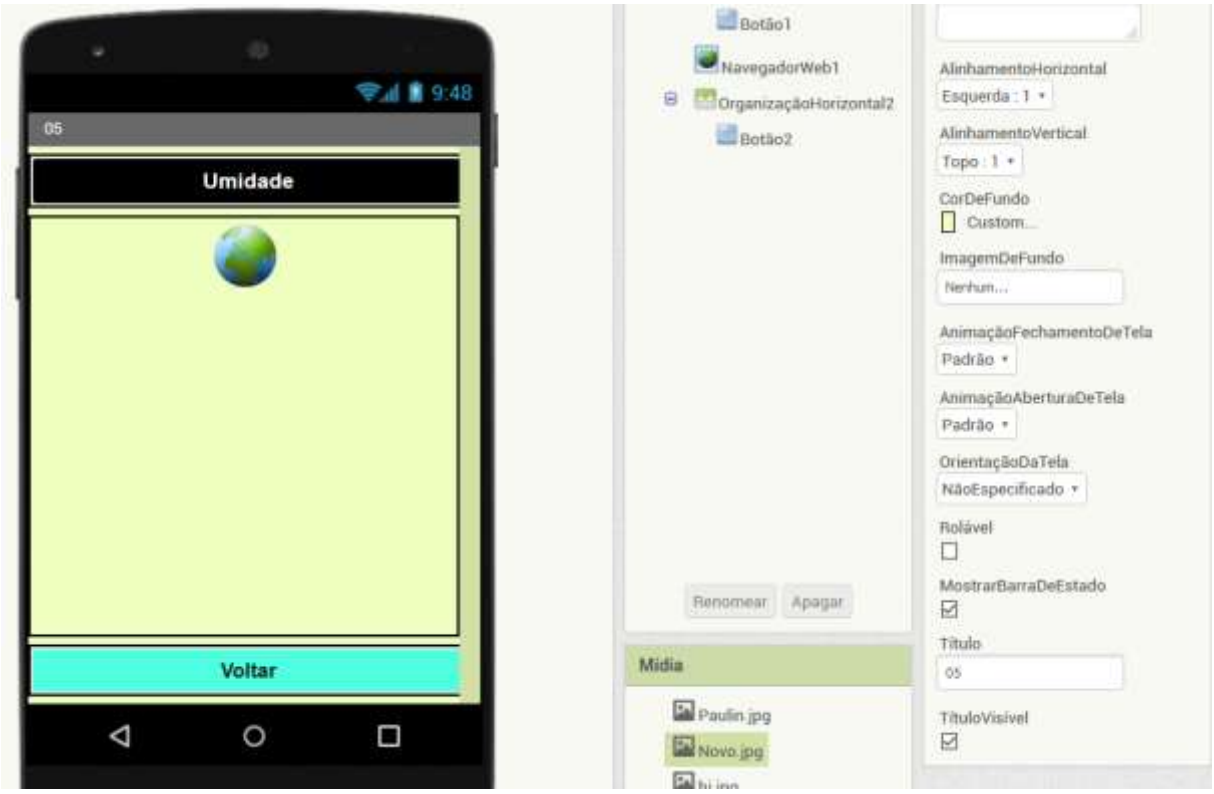
Figura 2.2 – Gráfico Lâmpada

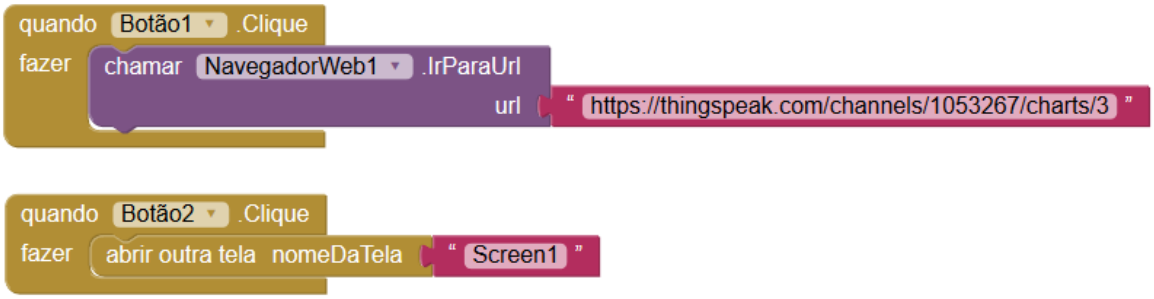
Figura 2.2 – Click Botão1 (Gráfico Lâmpada)



Quando o clicar no Botão1, o aplicativo vai acessar a URL e irá buscar a filed2 e desta forma haverá um retorno e com isso o gráfico da lâmpada será mostrado no app. Em seguida ao haver o click no Botão2 o aplicativo desenvolvido pela equipe vai acessar a Screen4.

Figura 2.3 – Gráfico Umidade





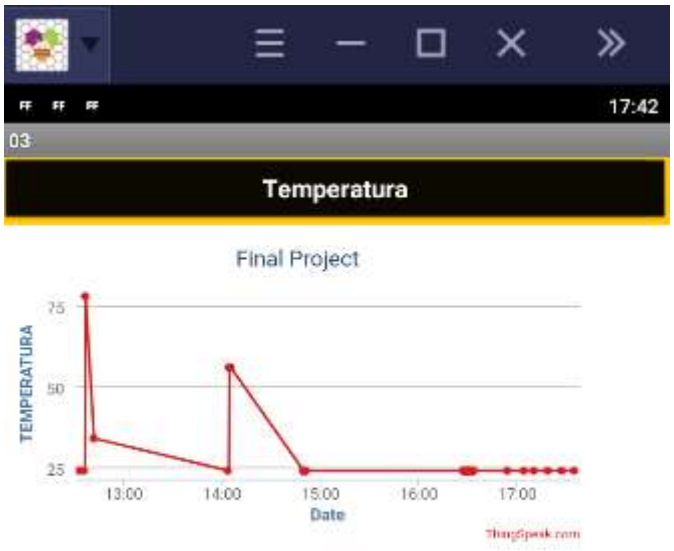
No mesmo segmento dos anteriores quando haver click no Botão1, o app acessará a URL e com isso buscará a field3 e mostrando no aplicativo o gráfico referente a Umidade. Em sequência quando clicar no Botão2 a interface voltará pra a Screen1, recomeçando a operação no aplicativo.

App Mobile 2020

Interface 01

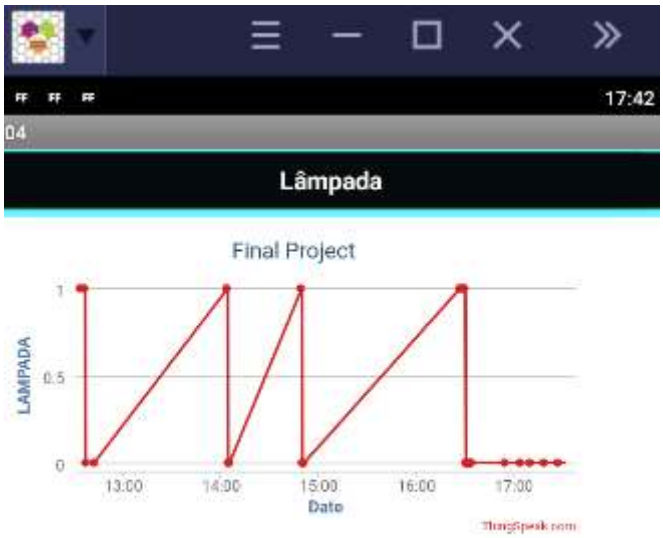
Interface 02

Interface 03



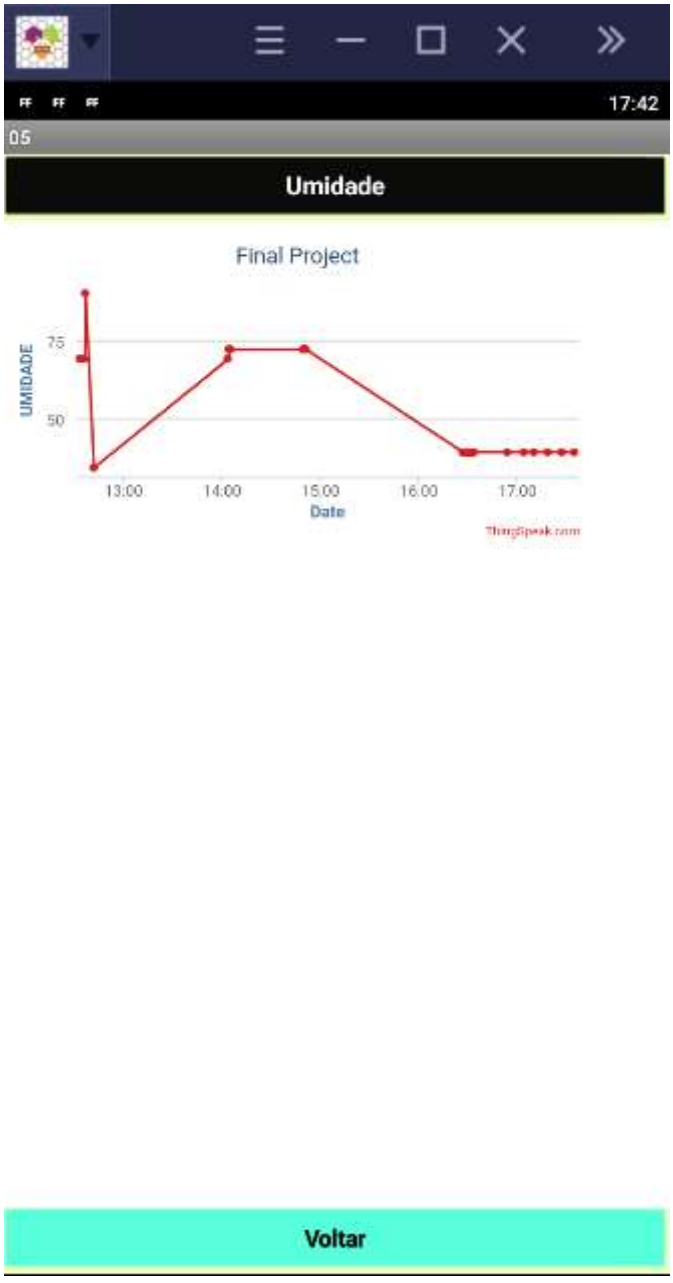
Próximo

Interface 04



Próximo

Interface 05



4. Referências

[1] Dowload do App – http://ai2.appinventor.mit.edu/ode/download/project-output/5542997089714176/Android

[2] Projeto Tinkercad - https://www.tinkercad.com/things/6iAm14kAL8x