

Trilha – Internet das Coisas

Prototipando uma solução completa de loT com tecnologias de custo zero: uma abordagem para novatos

André Luiz Maciel Santana (@andrelms91)

Professor (UAM) • Pesquisador (UAM/EP-USP/LSI-TEC) • Creative Learning Fellow (MIT Media Lab)

André Santana

Professor na Universidade Anhembi Morumbi

Engenheiro Mecânico e Mestre em Computação

Doutorando em Engenharia Elétrica na Escola Politécnica da USP

Membro da Rede Brasileira de Aprendizagem Criativa

Vencedor Sinapse da Inovação (2017)

Palestrante TDC, Campus Party, Scratch Conference



Pesquisador no LSITec (IoT eM)

Programa de Especialização Docente - Lemann Center de Stanford

Creative Learning Fellow - MIT Media Lab (2016)

Pesquisador em Tecnologias Aplicadas à Saúde e Educação

Transformações Sociais – Indústria 4.0



















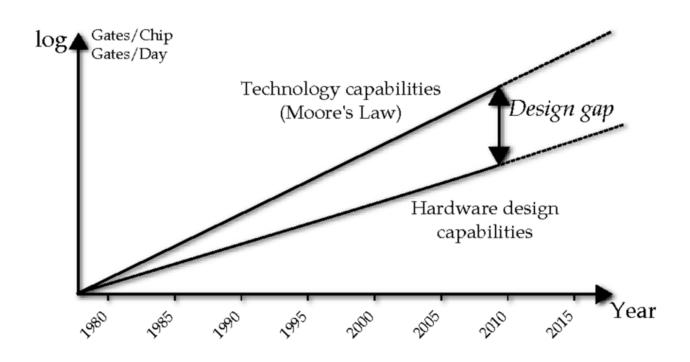


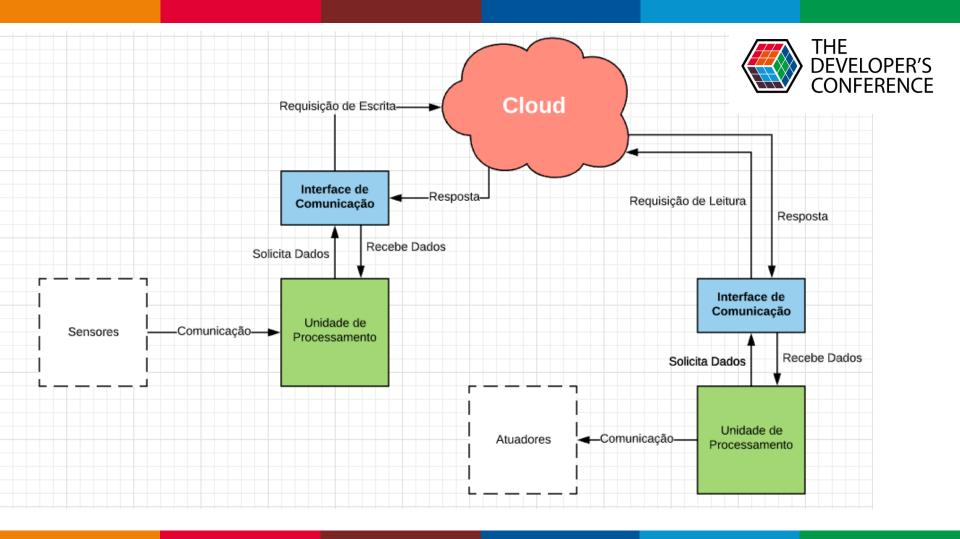
Estimulada pelas **demandas** da **população**, da **indústria** e pelo crescimento e envolvimento de tecnologias da informação;

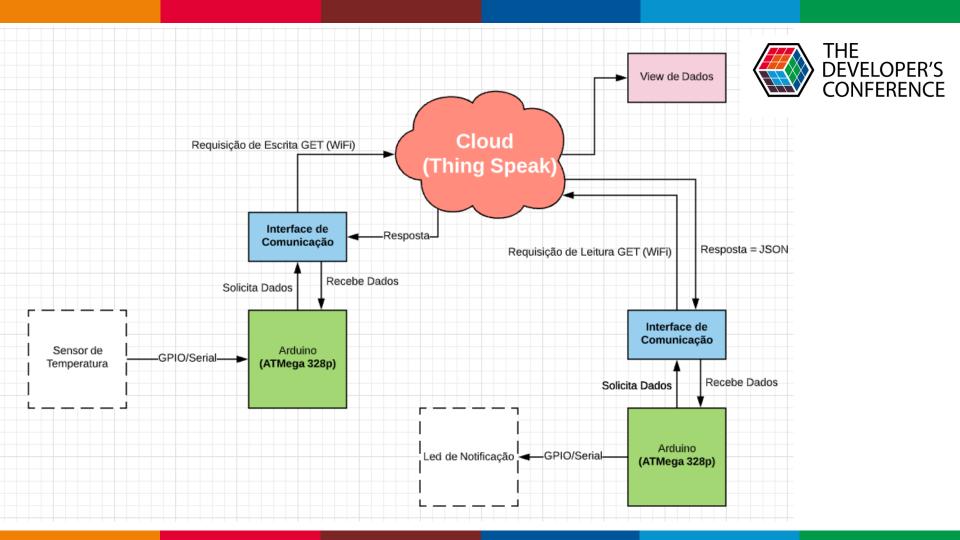
Catalisador frente às inovações sociais e tecnológicas e **inquietações** quanto à **sustentabilidade** deste processo de transformação.

Avanço Tecnológico e o "Design Gap"

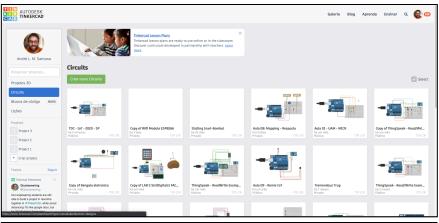








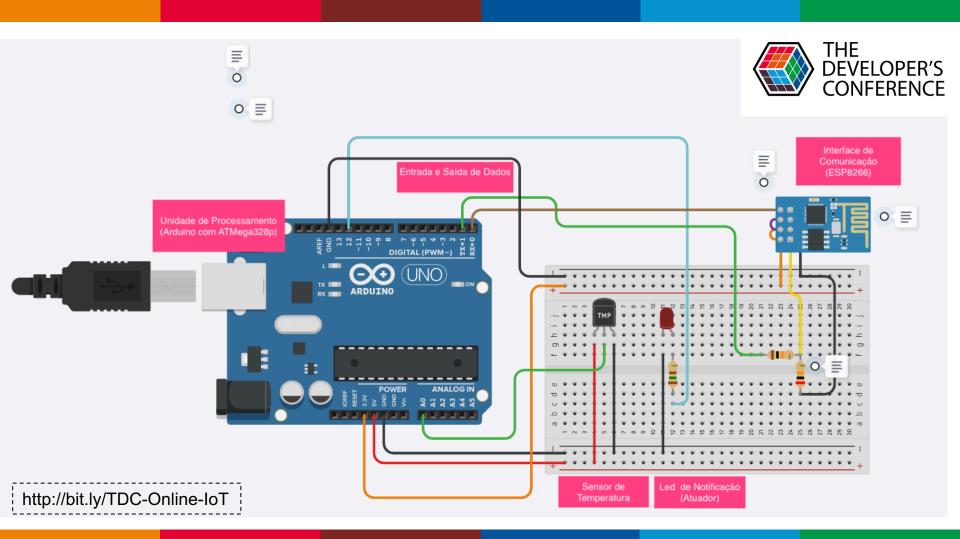
Plataforma: TinkerCad





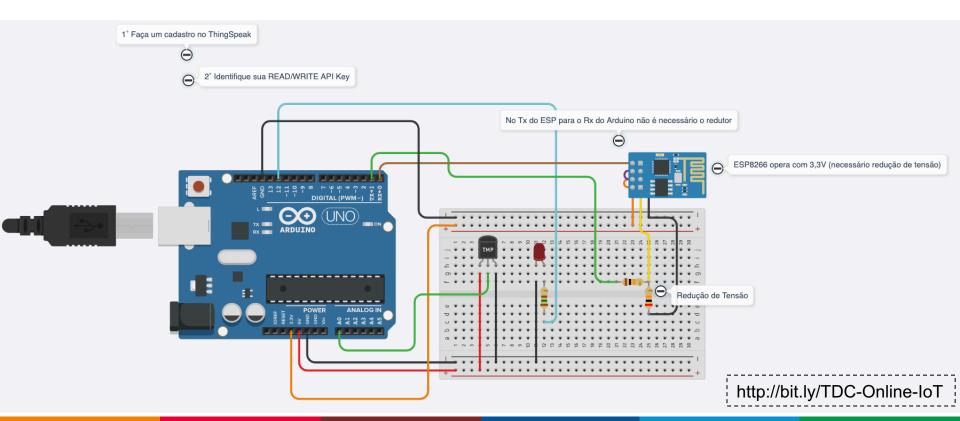


Uma breve demonstração da Plataforma



Implementação: Hardware

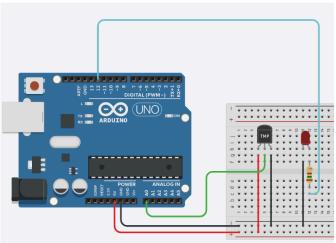




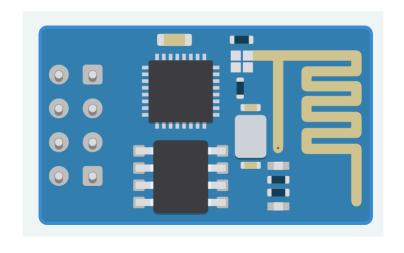
Implementação do Firmware

```
const int GPIO LED = 12;
const int GPIO SENSOR TEMP = A0;
void setup() {
     setupESP8266();
     pinMode(GPIO LED, OUTPUT);
     pinMode(GPIO SENSOR TEMP, INPUT);
void loop() {
     enviaTemperaturaESP8266();
     float temperatura = receberDadosESP8266();
     Serial.print(temperatura);
     acenderLedEmergencia(temperatura);
     delay(10000);
```





ESP8266

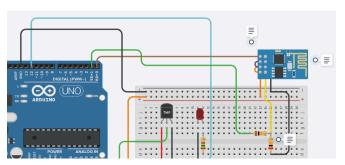


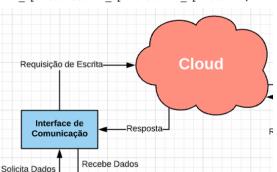
- Uma SOC (System-On-Chip) com Wifi embutido;
- CPU de 80 MHz podendo ir até 160 MHz
- RISC 32 bits
- 32 KBytes de ram (instruções)
- 96 KBytes de ram (dados)
- 64 KBytes de rom (boot)

Implementação do Firmware



```
= "Simulator Wifi"; // SSID para conectar em um WiFi, neste caso, uma rede "Simulada"
   String ssid
   String password = ""; // Este WiFi não possuí senha
   const int httpPort = 80;
                                                      //URL do Serviço de Nuvem (ThingSpeak)
   String apiHost
                         = "api.thingspeak.com";
                                                          //Colar sua key de escrita - substituindo COLAR CHAVE ESCRITA AQUI
   String apiIdWrite = "COLAR CHAVE ESCRITA AQUI";
   String apiRequestUpdate = "/update?api key=";
   String apiField1
                          = "&field1=";
   String uriWrite
                       = apiRequestUpdate + apiIdWrite + apiField1;
12
   String apiPathReadChannel = "/channels/";
   String apiChannelID
                         = "COLAR CHANNEL ID AQUI";
                                                          // Colar seu channel ID - substituindo COLAR CHANNEL ID AQUI
   String apiReadFeed
                          = "/feeds/last.json?api key=";
   String apiIdRead
                     = "COLAR CHAVE LEITURA AQUI";
                                                          //Colar sua key de leitura - substituindo COLAR CHAVE LEITURA AQUI
   String apiReadEnd
18
                      = apiPathReadChannel + apiChannelID + apiReadFeed + apiIdRead + apiReadEnd;
  String uriRead
```



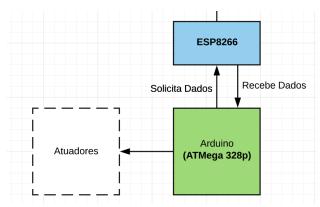


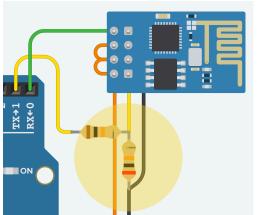
Implementação do Firmware



```
= "Simulator Wifi"; // SSID para conectar em um WiFi, neste caso, uma rede "Simulada"
   String ssid
   String password = ""; // Este WiFi não possuí senha
   const int httpPort
   String apiHost
                          = "api.thingspeak.com";
                                                       //URL do Serviço de Nuvem (ThingSpeak)
   String apiIdWrite = "COLAR CHAVE ESCRITA AQUI";
                                                           //Colar sua key de escrita - substituindo COLAR CHAVE ESCRITA AQUI
   String apiRequestUpdate = "/update?api key=";
   String apiField1
                          = "&field1=":
                       = apiRequestUpdate + apiIdWrite + apiField1;
   String uriWrite
12
   String apiPathReadChannel = "/channels/";
   String apiChannelID
                          = "COLAR CHANNEL ID AQUI";
                                                           // Colar seu channel ID - substituindo COLAR CHANNEL ID AQUI
   String apiReadFeed
                          = "/feeds/last.json?api key=";
   String apiIdRead
                       = "COLAR CHAVE LEITURA AQUI";
                                                           //Colar sua key de leitura - substituindo COLAR CHAVE LEITURA AQUI
   String apiReadEnd
18
                      = apiPathReadChannel + apiChannelID + apiReadFeed + apiIdRead + apiReadEnd;
   String uriRead
```

ESP8266





- Opera como um middleware "Serial Wi-Fi"
- O ESP8266 opera com 3,3V, no entanto, o Arduino enviará 5V. Logo, utilizamos um divisor de tensão (destacado em amarelo) para receber os sinais do Arduino (TX Arduino com RX ESP8266)
- Conectamos o TX do ESP8266 para o RX do Arduino
- Power (fio laranja) conectado com 3,3V volts e associado ao enable e reset do ESP8266.



My Channels



Search by tag	Q
---------------	---

2

New (Chanr			
	Name	TDCSP		
De	scription			
	Field 1	temperatura	✓	
•	Field 2			

- Após criarmos nossa conta, serão exibidos os canais que já criamos. Caso nenhuma canal esteja sendo apresentado, clique em "New Channel" [1]. Se esta tela não aparecer, clique em channels > My Channels.
- Atribua um nome para seu canal e crie as variáveis que serão enviadas pelo microcontrolado ao servidor [2]
- Por fim, salve seu canal [3]

3









Channel Settings

MATLAB Analysis

MATLAB Visualization

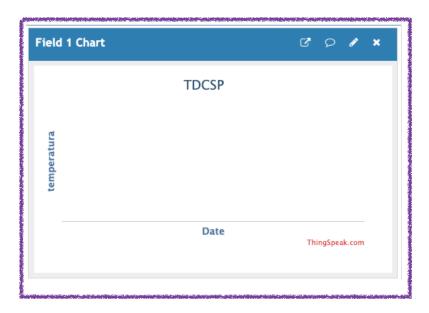
Channel Stats

Created: less than a minute ago

Public View

Entries: 0

Private View



- Em [4] é possível gerar e gerenciar as chaves de acesso para API do ThingSpeak.
- Veja que o gráfico de saída, destacado em roxo, mapeia as variáveis que criamos na etapa anterior. Neste caso, temperatura.



Write API Key COWNLLN6CA5TFK0F Read API Keys Key 9TCTDK1UIZM5CQ0W Note Save Note Delete API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key: Use this key to write data to a channel. If you feel your key has been compromised, click Generate New Write API Key.
- Read API Keys: Use this key to allow other people to view your private channel feeds and charts. Click Generate New Read API Key to generate an additional read key for the channel.
- Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

- Em [5] está disponível sua chave de acesso, que permite a escrita de dados no ThingSpeak
- Já em [6] estão os links de requisição para API. Indicando o host e a chave da API (já inserida)

Implementação: Firmware



```
int setupESP8266() {
 Serial.begin(115200);
 Serial.println("AT");
                                   Testa se está tudo OK
 delay(10);
 if (!Serial.find("OK"))
   return 1:
 Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
 delay(10);
 if (!Serial.find("OK"))
   return 2:
 Serial.println("AT+CIPSTART=\"TCP\",\"" + apiHost + "\"," + httpPort);
 delay(50);
 if (!Serial.find("OK"))
   return 3;
 return 0;
```

Conecta com Access Point e retorna resultado com SSID e status da conexao.

Inicia conexão com um cliente (modo multiplas conexões)

Id, type, addr, port

Comandos AT

http://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/

http://bit.ly/TDC-Online-IoT

Implementação: Firmware



```
void enviaTemperaturaESP8266() {
                                                                       Faz a leitura da GPIO conectada ao sensor e altera
                                                                                     os intervalos
  int temp = map(analogRead(GPIO SENSOR TEMP), 20, 358, -40, 125);
                                                                        map(value, fromLow, fromHigh, toLow, toHigh)
  // Construindo a requisição, tipo GET com HTTP
  String httpPacket = "GET " + uriWrite + String(temp) + " HTTP/1.1\r\nHost: " + apiHost + "\r\n\r\n";
  int length = httpPacket.length();
  // Enviando tamanho da mensagem por Serial para ESP8266
  Serial.print("AT+CIPSEND=");
                                                                Define tamanho dos dados que serão
  Serial.println(length);
                                                                            enviados
  delay(10);
                                                                             length
  // Enviando requisicao/mensagem para ESP8266
  Serial.print(httpPacket);
```

Comandos AT

delay(10);

87

88

89 90

92

93 94

95

96

98

99

100

101

102 103 104

http://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/

if (!Serial.find("SEND OK\r\n")) return;

http://bit.ly/TDC-Online-IoT

```
float receberDadosESP8266(){
 String httpPacket3 = "GET" + uriRead + " HTTP/1.1\r\nHost: " + _apiHost + "\r\n\r\n";
 int length3 = httpPacket3.length();
 Serial.print("AT+CIPSEND=");
 Serial.println(length3);
 delay(10);
 Serial.print(httpPacket3);
 delay(10);
 while(!Serial.available())
   delay(5);
 String saida = "";
 if (Serial.find("\r\n\r\n")){
       delay(5);
       unsigned int i = 0;
       if (!Serial.find("\"field1\":")){}
       while (i<60000) {
           if(Serial.available()) {
                  int c = Serial.read();
                  if (c == '.')
                      break:
                  if (isDigit(c))
                    saida += (char)c;
            i++;
   return saida.toFloat();
```

```
THE DEVELOPER'S CONFERENCE
```

```
"channel": {
    "id": 1123326,
    "name": "TDCSP",
    "latitude": "0.0",
    "longitude": "0.0",
    "field1": "temperatura",
    "created at": "2020-08-24T23:45:38Z",
    "updated at": "2020-08-24T23:45:38Z",
    "last entry id": 2
},
"feeds": [
        "created at": "2020-08-24T23:46:38Z",
        "entry id": 1,
        "field1": "24"
    },
        "created at": "2020-08-25T02:17:39Z",
        "entry id": 2,
        "field1": "24"
```





Twitter/Linkedin/Facebook

@andrelms91

Email

andrelms91@gmail.com

OBRIGADO!

