Alarma Aê – Projeto de alarme com envio de alerta para aplicativo para Smartphone.

Caroline Alves (sma.caroline@ymail.com)
Henry (hpoleselo@gmail.com)
Valdinei França (valdiney.2@hotmail.com)
Victor Correa (senna.correa@gmail.com)

O Alarma Aê é um alarme detector de movimento que envia alertas para um APP caso detecte movimento no local onde está instalado. Foi utilizado neste projeto:

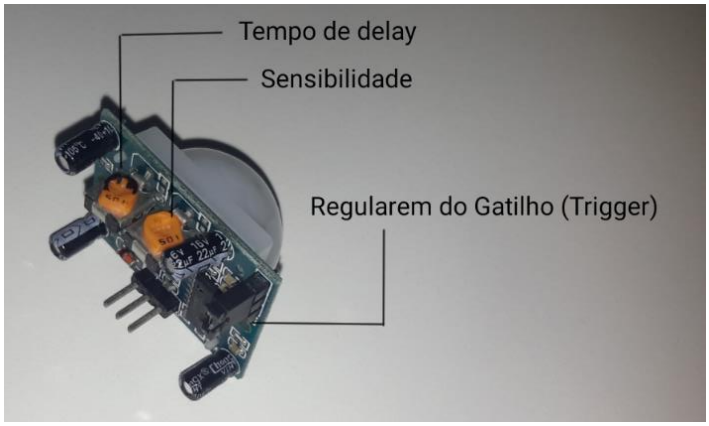
- Placa NodeMcu V3;
- Módulo sensor de movimento PIR para Arduino;
- 2 LED´s;
- 2 Resistores de 220 Ohms;
- Protoboard;
- Jumper;

A placa NodeMcu V3 possui o chip Esp8266 integrado a uma interface usb-Serial e um regulador de tensão de 3.3V. Pode ser programado em linguagem LUA ou através da IDE do Arduino.

O módulo PIR detecta movimentos baseados na variação da luz infravermelha emitida pela radiação do corpo humano e de animais. Quando o sensor detecta o movimento ele emite sinal alto, caso não sinal baixo. Pode ser regulado seu nível de sensibilidade entre 3m a 7m de distância e um tempo de delay entre 3s e 200s.

Outra regulagem do módulo de movimento é a forma do Trigger que após acionado permanece alto até o tempo final do delay programado ou pode ser reativado durante o tempo de duração do delay reiniciando o tempo de duração do sinal alto.

Imagem do sensor de movimento com indicadores dos potenciômetros e do jumper para regulagem.



Funcionamento do Alarme

NodeMcu se conecta à rede local e faz leituras consecutivas do sensor, caso o sensor emita sinal alto o micro controlador se comunica com um servidor http enviando a mensagem de alerta e retorna um código para verificação do êxito da comunicação.

O tempo mínimo entre um envio de alerta e outro pode ser regulado junto ao tempo de delay do sensor. Para cada detecção de movimento um alerta será enviado.

O App AlarmaÊ mostra o dia, o horário e o local que o sensor foi acionado até uma quantidade de movimentações. Através do cadastro o usuário também tem acesso ao resumo (quantidade) de movimentações por dia semana e mês. Permite o cadastro dos usuários que terão acesso a essas informações.

O aparelho possui dois leds como demonstrativo de funcionamento, o led verde permanece acionado para indicar a conexão com a rede Wi-fi, o led vermelho é acionado ao detectar presença e se deliga depois do tempo de delay escolhido. Caso a rede Wi-fi seja desconectada o led verde se desliga e o vermelho liga até a conexão ser reestabelecida. Nesse caso a detecção de movimento para de ser feita até a reconexão. Caso a comunicação com o servidor http não seja feita com sucesso o sistema repete mensagem até obter êxito.

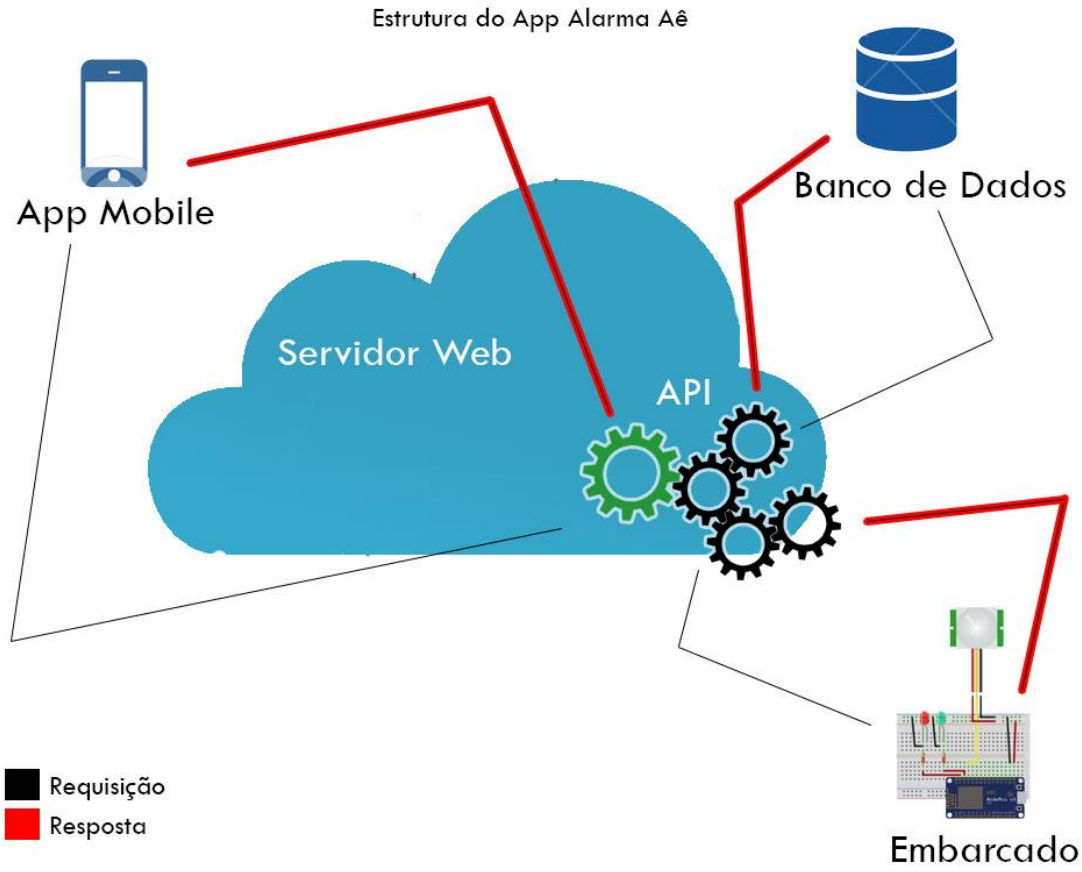
Funcionamento técnico do App

O App foi construído sobre duas camadas conceituais, sendo uma delas a camada web que consiste na criação de uma API escrita em PHP que por sua vez se conecta a uma base de dados MySql e retorna dados no formado Json.

Essa API possui métodos para receber dados por via do verbo http GET que é disparado pelo sistema embarcado direto para a API. Também possui métodos para elaboração de pequenos relatórios que serão apresentados na segunda camada do projeto que trata-se de um App para Android.

O App Android foi escrito com as mesmas tecnologias usadas na web, como HTML5, CSS3 e bastante Javasacript com Jquery juntamente com Ajax para consumir os dados oferecidos pela API. Foi utilizado o framework phonegap-cordova para que de fato a camada mobile possa ser instalada e interpretada no Android.

Estrutura do App:



O App mobile simplesmente consulta a API, que consulta a base de dados e retorna os dados no formato Json. A API pode ser entendida como uma camada de serviço que está disposta a receber estímulos e retorna algo valido. Tanto o App quanto o Embarcado, consultam a API para lhe entregar dados ou para consulta-los.

Veja um exemplo de consulta API por via do App:

```
$(function() {
    $(".banner_cadastrar_email").css('min-height', $(window).height() -135);
    $(".dias_content").hide();

    function load_alertas_por_dia() {
        var url = "http://iot.wifiaqui.com.br/alarme/?presenca=quantidade_alarmes_por_dia";

        $.getJSON(url, function(result) {
            $.each(result, function(i, field) {

                var content = "<div class='emails' href=''><div class='container_body'>";
                    content += " <img src='img/alarm.png' class='pull-left alarm_icon'/>";
                    content += "<h4 class='pull-left numero_dia_semana_grafico'>" + field.quantidade + "</h4>";
                    content += "<h4>" + dias_da_semana(field.data) + "<br><small style='color:#333333'>" + field.data + "<small></h4>";
                    content += "</div></div>";
                    $(".dias_content").append(content);
            });
        }).done(function() {
            $(".carregando").hide('fast');
            $(".dias_content").slideDown('fast');
        });;
    }

    load_alertas_por_dia();

    /*Pega o número do dia para encontrar o dia por Extenso*/
    function dias_da_semana(data_db) {
        var dia = [];
            dia[0] = 'Domingo';
            dia[1] = 'Segunda';
            dia[2] = 'Terça';
            dia[3] = 'Quarta';
            dia[4] = 'Quinta';
            dia[5] = 'Sexta';
            dia[6] = 'Sábado';
        data_db = data_db.split('/');
        var data = new Date(data_db[2], data_db[1]-1, data_db[0]);
        var hoje = data.getDay();
        return dia[hoje];

    }
});
```

A consulta a API retornará um Json como este:

[{"quantidade":"1846","data":"08\/07\/2017"},
{"quantidade":"259","data":"09\/07\/2017"},
{"quantidade":"6","data":"10\/07\/2017"},
{"quantidade":"52","data":"12\/06\/2017"},
{"quantidade":"407","data":"13\/06\/2017"},
{"quantidade":"646","data":"14\/06\/2017"},
{"quantidade":"607","data":"15\/06\/2017"},
{"quantidade":"91","data":"16\/06\/2017"},
{"quantidade":"8","data":"17\/06\/2017"},
{"quantidade":"921","data":"18\/06\/2017"},
{"quantidade":"14","data":"19\/06\/2017"},
{"quantidade":"69","data":"22\/06\/2017"},
{"quantidade":"834","data":"23\/06\/2017"},
{"quantidade":"285","data":"30\/06\/2017"}]

O App interpretará esses dados que são entregues e apresentará de forma elegante na interface de usuário.

Representação visual dos dados retornados pela consulta a API:

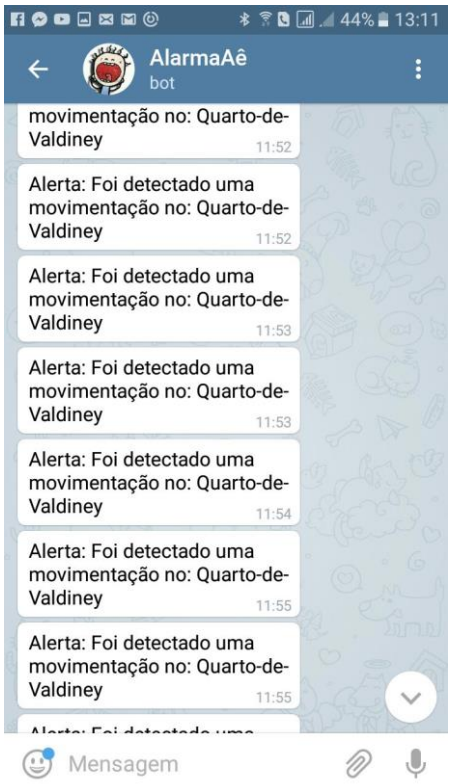
Tela de cadastro de usuário

Tela de Resumos

Alerta via Bot no Telegram:

Também foi criado um BOT, ou seja, um Web Robot com o intuito de alertar o momento que o sensor detectou uma movimentação no ambiente que ele estiver operando. Funciona da seguinte forma: No momento que o embarcado enviar um estilo para a API do Alarma Aê, a mesma consulta a API do telegrama e envia para ela os dados do alerta e a mesma se encarrega de mostrar isso no chat do telegrama.

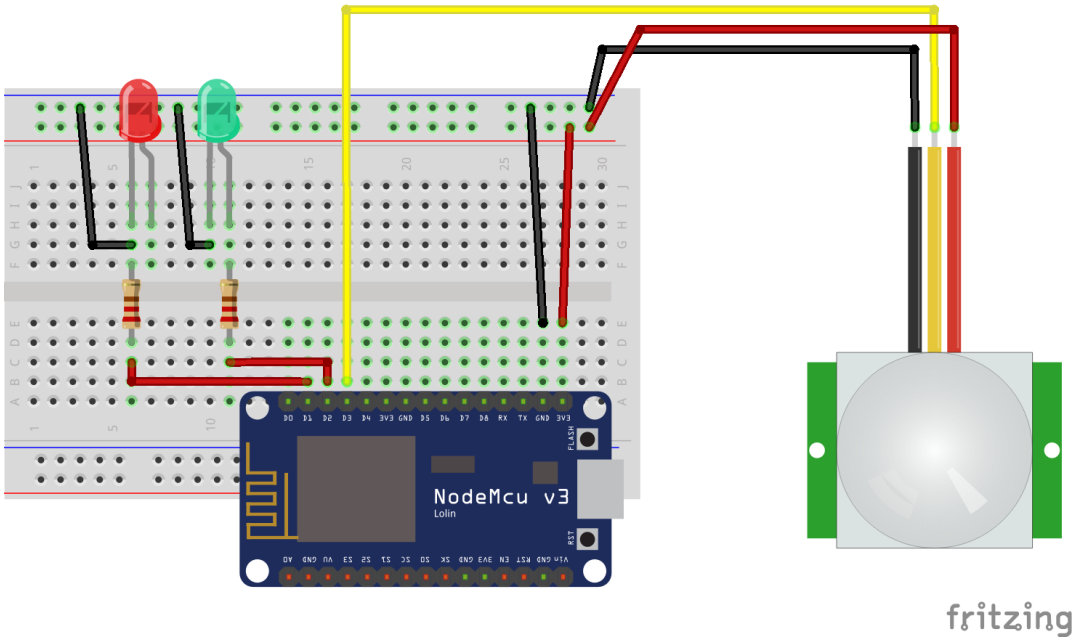
Imagem de disparos no Chat do Telegram:



QR Code para baixar o App

Esquemático do circuito

O sensor PIR possui 3 pinos um vcc um gnd e uma saída (out) que envia o sinal digital alto (1) ou baixo (0). Os leds vermelho e verde foram ligados nas portas D1 e D2 respectivamente com os resistores de 220 ohms. O pino out está ligado na porta D3 do NodeMcu. Neste exemplo o sensor está sendo alimentado pela saída de 3v do NodeMCU, é importante verificar a tensão de alimentação, como também a ordem dos pinos de alimentação e ground, no datasheet do equipamento pois pode haver variações.



Vídeos demonstrando o funcionamento do Alarme

1. Vídeo com um tempo de delay entre as mensagens de aproximadamente 3s.

   https://youtu.be/YH9kAQmtT2k

2. Vídeo com um tempo de delay entre as mensagens de aproximadamente 1 min.

   https://youtu.be/JgFLSofZe1E

3. Vídeo demostrando o comportamento do sistema em caso de desconexão da rede Wi-Fi

   https://youtu.be/_bYXWu3R1f4

O código do embarcado

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

//Dados da conexão Wi-fi

const char* ssid     = "ssid";
const char* password = "senha";

HTTPClient http; // Declara variável para para p HttpClient

int ledVerm = 5;   // Pino ligado ao led vermelho
int ledVerde = 4;   // Pino ligado ao led verde
int sensor = 0;  // Pino ligado ao sensor PIR
int valorSensor = 0;   // Variavel para guardar valor do sensor
int httpCode = 0; // Variável para código de retorno do HttpClient
int aux = 0; // Auxiliar para armazenar o estado anterior do sensor

void setup() {

  Serial.begin(115200);
  delay(10);

  pinMode(ledVerm, OUTPUT);  //Define pino D0 como saida
  digitalWrite(ledVerm, LOW);
  pinMode(ledVerde, OUTPUT);  //Define pino D1 como saida
  digitalWrite(ledVerde, LOW);
  pinMode(sensor, INPUT);   //Define pino D3 como entrada


  WiFi.begin(ssid, password); // Inicia conexão Wifi


  while (WiFi.status() != WL_CONNECTED) { // Aguarda a conexão ser estabelecida

    Serial.println("...");
    delay(1000);
  }

  Serial.println("Conectado");
}

void loop() {

  verificaConexao(); //Chama função para verificar a conexão Wifi

  valorSensor = digitalRead(sensor); // Verifica sinal so sensor
  digitalWrite(ledVerm, valorSensor); //Seta o valor do sensor no Led vermelho
```

```
  if ((valorSensor == 1) && (aux != 1)) { // Verifica o valor do sensor e da variável auxiliar para
não repetir a mesma informação,
                                    // entra no if apenas se o estado anterior do sensor foi 0
    enviaMensagem(); // Chama função para o envio da mensagem para o API

    if (httpCode >= 0) { //httpCode maior > 0 indica que o servidor respondeu corretamente

      aux = 1; // Armazena o estado do sensor
    }

    else { //Caso apresente erro de comunicação com o servidor

      while (httpCode < 0) { //Tentativa de realizar o get até resultado positivo

        verificaConexao(); // Verifica conexão Wifi
        enviaMensagem(); // Repete o envio da mensagem
      }
    }
  }

  else if (valorSensor == 0) {
    aux = 0; // Armazena o estado do sensor
  }
}

void verificaConexao() {

  if (WiFi.status() == WL_CONNECTED) { // Confirma a conexão com a rede Wifi

    digitalWrite(ledVerde, HIGH); // Indica Nodemcu Conectado
  }

  else { // Solicita conexão com a rede Wifi

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) { //Enquanto Wifi não conecta
      digitalWrite(ledVerde, LOW); // Desliga Led Verde
      digitalWrite(ledVerm, HIGH); // Liga Led Vermelho
      Serial.println("...");
      delay(1000);
    }

    Serial.println("Conectado"); //Wifi conectado
    digitalWrite(ledVerm, LOW); //Retorna o estado dos Leds
    digitalWrite(ledVerde, HIGH);
  }
}

int enviaMensagem() {
```

```
  http.begin("http://iot.wifiaqui.com.br/alarme/?presenca=cadastrar&localidade=Nome-do-
local"); //Envia o alerta para a API
  httpCode = http.GET(); //Recebe código para confirmação do get
  http.end(); //Finaliza o Client
  delay(500);
  return httpCode; //Retorna o HttpCode
}
```