

```

// Microcontrolador - ESP 32 (38 GPIOs)
/*
Esse programa realiza o controle e processamento de dados do sistema de
distribuição de água da Cidade Inteligente.
O sistema é composto por diversas integrações para garantir a eficiência.
Segue algumas instruções básicas de utilização:
1) Conectar ao sistema AP ao dispositivo celular, notebook, etec. Para
isso certifique que o Esp32 esteja alimentado.
2) Acesse as redes Wifi do seu dispositivo e conecte em {ESP32-Manu}
3) Insira a senha {0123456789}
Após esses passos você estará conectado a rede e então pode acessar ao
site.
4) Abra seu navegador google, firefox...
5) Insira no campo de pesquisa a URL {192.168.9.1}
Se tudo estiver funcionando corretamente você será direcionado a página
ManuJurck, tendo acesso remoto aos níveis das caixas de água.
Instruções:
Caso o sistema não esteja funcional é viável resetar o Esp32 através do
botão RES da placa, se o erro persistir pode-se realizar a compilação do
código novamente do Esp32,
pois é provável que um corrompimento no código tenha ocorrido.
Lembre-se que para a compilação de códigos nas plataformas Esp32 é
necessário a preparação da plataforma Arduino IDE. Para mais informações
acesse o manual de usuário.
Caso contrário não será realizada a compilação!!
Segue a programação.
*/

//Bibliotecas -----
-----
#include <Ultrasonic.h> //
Responsável pelos sensores ultrassônicos
#include <WiFi.h> //
Responsavel pelo Wifi
#include <Wire.h> //
Responsável pela comunicação Wire
#include <LiquidCrystal_I2C.h> //
Responsavel pelo LCD
#include <WebServer.h> //
Responsável pelo Servidor Web

//WiFi -----
-----
const char* ssid = "ESP32-Manu"; // Nome da
rede wireless
const char* password = "0123456789"; // Senha da
rede wireless

```

```

//Servidor -----
-----
IPAddress local_ip(192,168,9,1);           // IP do
site
IPAddress gateway(192,168,9,1);           // Gateway
do site
IPAddress subnet(255,255,255,0);         // Endereço
de subnet
WebServer server(80);                   // Porta
padrão do servidor

//LCD -----
-----
LiquidCrystal_I2C lcd(0x27,16,2);

//Variáveis Globais e Definições -----
-----
uint8_t LED1pin = 35;                   // Pino do
led do Botão
bool LED1status = LOW;                 // Estado
inicial do led
//Sensor ultrassônico 1
Ultrasonic ultrasonic_1(23, 33);
int distance_1;
int val_1;
int val1;
//Sensor ultrassônico 2
Ultrasonic ultrasonic_2(3, 19);
int distance_2;
int val_2;
int val2;
//Sensor ultrassônico 3
Ultrasonic ultrasonic_3(18, 5);
int distance_3;
int val_3;
int val3;
//relé
int rele_1 = 17;
int rele_2 = 16;
int rele_3 = 4;

//Funções Auxiliares -----
-----
void setupWiFi();                       //
Responsável por configurar WiFi
void servidor();

//Setup -----
-----

```

```

void setup() {
  //relés
  pinMode(rele_1, OUTPUT);
  pinMode(rele_2, OUTPUT);
  pinMode(rele_3, OUTPUT);
  //leds 1
  pinMode(0, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(15, OUTPUT);
  //leds 2
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(14, OUTPUT);
  //leds 3
  pinMode(27, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(25, OUTPUT);
  pinMode(LED1pin, OUTPUT); //
  Configura o pino como OUTPUT
  setupWiFi(); // Chama
  função setupWiFi()
  servidor(); // Chama
  função servidor
  Serial.begin(115200); // Inicia
  comunicação Serial
  Serial.println("Conexão serial estabelecida"); // Envia
  mensagem Serial
  //LCD
  lcd.init();
  lcd.init();
  lcd.backlight();
  lcd.begin(16,2);
}

```

```

//Loop -----
-----

```

```

void loop() {
  server.handleClient();
  if(LED1status)
  {digitalWrite(LED1pin, HIGH);}
  else
  {digitalWrite(LED1pin, LOW);}
  //sensor 1
  distance_1 = ultrasonic_1.read();
  val_1 = 100 * distance_1 / 100;
  val1 = 100 - val_1;
  Serial.print("Distance in CM: ");
  Serial.println(distance_1);
  delay(500);
}

```

```

    //sensor 2
distance_2 = ultrasonic_2.read();
val_2 = 100 * distance_2 / 100;
val2 = 100 - val_2;
Serial.print("Distance in CM: ");
Serial.println(distance_2);
delay(500);
    //sensor 3
distance_3 = ultrasonic_3.read();
val_3 = 100 * distance_3 / 100;
val3 = 100 - val_3;
Serial.print("Distance in CM: ");
Serial.println(distance_3);
delay(500);
    //atuação caixa 1
    if (distance_1 > 8){
        digitalWrite(0, 1);
        digitalWrite(2, 0);
        digitalWrite(15, 0);
    }
    else if (distance_1 < 8 && distance_1 > 4){
        digitalWrite(0, 0);
        digitalWrite(2, 1);
        digitalWrite(15, 0);
    }
    else{
        digitalWrite(0, 0);
        digitalWrite(2, 0);
        digitalWrite(15, 1);
    }
    //atuação caixa 2
    if (distance_2 > 8){
        digitalWrite(13, 1);
        digitalWrite(12, 0);
        digitalWrite(14, 0);
    }
    else if (distance_2 < 8 && distance_2 > 4){
        digitalWrite(13, 0);
        digitalWrite(12, 1);
        digitalWrite(14, 0);
    }
    else{
        digitalWrite(13, 0);
        digitalWrite(12, 0);
        digitalWrite(14, 1);
    }
    // atuação caixa 3
    if (distance_3 > 8){
        digitalWrite(27, 1);

```

```

        digitalWrite(26, 0);
        digitalWrite(25, 0);
    }
    else if (distance_3 < 8 && distance_3 > 4){
        digitalWrite(27, 0);
        digitalWrite(26, 1);
        digitalWrite(25, 0);
    }
    else{
        digitalWrite(27, 0);
        digitalWrite(26, 0);
        digitalWrite(25, 1);
    }
    // Atuação bomba 1
    if (distance_1 > 8){
        digitalWrite(rele_1, 0);
        lcd.setCursor(0,0);
        lcd.print("B1 ON");
        // SerialBT.print("Bomba_1_ON");
    }
    if (distance_1 <= 3){
        digitalWrite(rele_1, 1);
        lcd.setCursor(0,0);
        lcd.print("      ");}
    // Atuação bomba 2
    if (distance_2 > 8){
        digitalWrite(rele_2, 0);
        lcd.setCursor(7,1);
        lcd.print("B2 ON");
        // SerialBT.print("Bomba_2_ON");
    }
    if (distance_2 <= 3){
        digitalWrite(rele_2, 1);
        lcd.setCursor(7,1);
        lcd.print("      ");
    }
    // Atuação bomba 3
    if (distance_3 > 8){
        digitalWrite(rele_3, 0);
        lcd.setCursor(0,1);
        lcd.print("B3 ON");
        // SerialBT.print("Bomba_3_ON");
    }
    if (distance_3 <= 3){
        digitalWrite(rele_3, 1);
        lcd.setCursor(0,1);
        lcd.print("      ");}
}

```

```

//Config WiFi -----
-----
void setupWiFi() {
    WiFi.softAP(ssid, password);           //
Inicializa WiFi
    WiFi.softAPConfig(local_ip, gateway, subnet);           // Conecta
no Servidor
    delay(500);                               // Espera
0,50s
}

//Setup server -----
-----
void servidor(){
    server.on("/", handle_OnConnect);
    server.on("/led1on", handle_led1on);
    server.on("/led1off", handle_led1off);
    server.onNotFound(handle_NotFound);
    server.begin();
    Serial.println("HTTP server started");
    Serial.println(ssid);
    Serial.println("O dispositivo já pode ser pareado");}

// Conexões do Servidor -----
-----
void handle_OnConnect() {
    LED1status = LOW;
    Serial.println("GPIO4 Status: OFF");
    server.send(200, "text/html",
SendHTML(LED1status,distance_1,distance_2,distance_3));
}
void handle_led1on() {
    LED1status = HIGH;
    Serial.println("GPIO4 Status: ON");
    server.send(200, "text/html",
SendHTML(true,distance_1,distance_2,distance_3));
}
void handle_led1off() {
    LED1status = LOW;
    Serial.println("GPIO4 Status: OFF");
    server.send(200, "text/html",
SendHTML(false,distance_1,distance_2,distance_3));
}
void handle_NotFound(){
    server.send(404, "text/plain", "Not found");
}

//String SendHTML -----
-----

```

```

String SendHTML(uint8_t led1stat,int distance_1,int distance_2,int
distance_3){
    //uint8_t - é usado para variaveis que são lidas apenas como 0 e 1.
    Float - variavel para numeros com virgula
    String ptr = "<!DOCTYPE html> <html>\n";// indicar que o código a
seguir será do tipo HTML
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width,
initial-scale=1.0, user-scalable=no\">\n";
    ptr += "<meta http-equiv=\"refresh\" content=\"2\">\n";//atualiza a
pagina web de dois em dois segundos
    ptr += "<link
href=\"https://fonts.googleapis.com/css?family=Open+Sans:300,400,600\"
rel=\"stylesheet\">\n";
    ptr += "<title>ManuJURCK</title>\n";
    ptr += "<style>html { font-family: Helvetica; display: inline-block;
margin: 0px auto; text-align: center;}\n";
    ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto
30px;} h3 {color: #444444;margin-bottom: 50px;}\n";
    ptr += ".button {display: block;width: 80px;background-color:
#3498db;border: none;color: white;padding: 13px 30px;text-decoration:
none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius:
4px;}\n";
    ptr += ".button-pair { display: flex; flex-direction: row; justify-
content: space-between; }\n";
    ptr += ".button-container { flex: 1; margin: 0 5px; }\n";
    ptr += ".button-on {background-color: #3498db;}\n";
    ptr += ".button-on:active {background-color: #2980b9;}\n";
    ptr += ".button-off {background-color: #34495e;}\n";
    ptr += ".button-off:active {background-color: #2c3e50;}\n";
    ptr += "h2 {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
    ptr += ".side-by-side{display: inline-block;vertical-align:
middle;position: relative;}\n";
    ptr += ".humidity-icon{background-color: #3498db;width: 30px;height:
30px;border-radius: 50%;line-height: 36px;}\n";
    ptr += ".humidity-text{font-weight: 600;padding-left: 15px;font-size:
19px;width: 160px;text-align: left;}\n";
    ptr += ".humidity{font-weight: 300;font-size: 40px;color: #3498db;}\n";
    ptr += ".temperature-icon{background-color: #f39c12;width: 30px;height:
30px;border-radius: 50%;line-height: 40px;}\n";
    ptr += ".temperature-text{font-weight: 600;padding-left: 15px;font-size:
19px;width: 160px;text-align: left;}\n";
    ptr += ".temperature{font-weight: 300;font-size: 40px;color:
#f39c12;}\n";
    ptr += ".superscript{font-size: 17px;font-weight: 600;position:
absolute;right: -20px;top: 15px;}\n";
    ptr += ".data{padding: 10px;}\n";
    ptr += "</style>\n";
    ptr += "</head>\n";
    ptr += "<body>\n";

```

```

ptr += "<div id=\"webpage\">\n";
ptr += "<h1>ESP32 ManuJURCK</h1>\n";
ptr += "<h3>CONTROLE DE ACESSO</h3>\n";
ptr += "<div class=\"data\">\n";
ptr += "<div class=\"side-by-side humidity-icon\">\n";
ptr += "<svg version=\"1.1\" id=\"Layer_2\"
xmlns=\"http://www.w3.org/2000/svg\"
xmlns:xlink=\"http://www.w3.org/1999/xlink\" x=\"0px\" y=\"0px\" \n\";
width=\"12px\" height=\"17.955px\" viewBox=\"0 0 13 17.955\" enable-
background=\"new 0 0 13 17.955\" xml:space=\"preserve\">\n";
ptr += "<path fill=\"#FFFFFF\"
d=\"M1.819,6.217C3.139,4.064,6.5,0,6.5,0s3.363,4.064,4.681,6.217c1.793,2.
926,2.133,5.05,1.571,7.057\n\";
ptr += "c-0.438,1.574-2.264,4.681-6.252,4.681c-3.988,0-5.813-3.107-
6.252-4.681C-0.313,11.267,0.026,9.143,1.819,6.217\"></path>\n";
ptr += "</svg>\n";
ptr += "</div>\n";
ptr += "<div class=\"side-by-side humidity-text\">Nivel - Tanque
(1)</div>\n";
ptr += "<div class=\"side-by-side humidity\">";
ptr += "<distance>\" + String(val1) + \" %</h2>\n";
ptr += "</div>\n";
ptr += "<div class=\"data\">\n";
ptr += "<div class=\"side-by-side humidity-icon\">\n";
ptr += "<svg version=\"1.1\" id=\"Layer_2\"
xmlns=\"http://www.w3.org/2000/svg\"
xmlns:xlink=\"http://www.w3.org/1999/xlink\" x=\"0px\" y=\"0px\" \n\";
width=\"12px\" height=\"17.955px\" viewBox=\"0 0 13 17.955\" enable-
background=\"new 0 0 13 17.955\" xml:space=\"preserve\">\n";
ptr += "<path fill=\"#FFFFFF\"
d=\"M1.819,6.217C3.139,4.064,6.5,0,6.5,0s3.363,4.064,4.681,6.217c1.793,2.
926,2.133,5.05,1.571,7.057\n\";
ptr += "c-0.438,1.574-2.264,4.681-6.252,4.681c-3.988,0-5.813-3.107-
6.252-4.681C-0.313,11.267,0.026,9.143,1.819,6.217\"></path>\n";
ptr += "</svg>\n";
ptr += "</div>\n";
ptr += "<div class=\"side-by-side humidity-text\">Nivel - Tanque
(2)</div>\n";
ptr += "<div class=\"side-by-side humidity\">";
ptr += "<distance>\" + String(val2) + \" %</h2>\n";
ptr += "</div>\n";
ptr += "<div class=\"data\">\n";
ptr += "<div class=\"side-by-side humidity-icon\">\n";
ptr += "<svg version=\"1.1\" id=\"Layer_2\"
xmlns=\"http://www.w3.org/2000/svg\"
xmlns:xlink=\"http://www.w3.org/1999/xlink\" x=\"0px\" y=\"0px\" \n\";
width=\"12px\" height=\"17.955px\" viewBox=\"0 0 13 17.955\" enable-
background=\"new 0 0 13 17.955\" xml:space=\"preserve\">\n";

```



```

    ptr += "<path fill=\"#FFFFFF\"
d=\"M1.819,6.217C3.139,4.064,6.5,0,6.5,0s3.363,4.064,4.681,6.217c1.793,2.
926,2.133,5.05,1.571,7.057\n";
    ptr += "c-0.438,1.574-2.264,4.681-6.252,4.681c-3.988,0-5.813-3.107-
6.252-4.681C-0.313,11.267,0.026,9.143,1.819,6.217\"></path>\n";
    ptr += "</svg>\n";
    ptr += "</div>\n";
    ptr += "<div class=\"side-by-side humidity-text\">Nivel - Tanque
(3)</div>\n";
    ptr += "<div class=\"side-by-side humidity\">";
    ptr += "<distance>" + String(val3) + " %</h2>\n";
    ptr += "</div>\n";
    if(led1stat)
    {ptr += "<div class=\"button-container\"><p>Circuito lampadas (1) <br>
Status : ON</p><a class=\"button button-off\"
href=\"/led1off\">OFF</a></div>\n";}
    else
    {ptr += "<div class=\"button-container\"><p>Circuito lampadas (1) <br>
Status : OFF</p><a class=\"button button-on\"
href=\"/led1on\">ON</a></div>\n";}

    ptr += "</div>\n";
    ptr += "</body>\n";
    ptr += "</html>\n";
    return ptr;
}

```