

***Envío de frases positivas dende  
App IoT MQTT Panel con Node-  
Red a unha matriz de puntos  
conectada con ESP32 e Arduino  
Nano***

# ÍNDICE

<b>Introdución.</b> .....	<b>3</b>
<b>Funcionamento.</b> .....	<b>4</b>
<b>Compoñentes da montaxe.</b> .....	<b>11</b>
<b>Software empregado.</b> .....	<b>12</b>
Programa dos Microcontroladores. ....	12
Programación do Arduino Nano. ....	12
Programación da ESP32: .....	15
<b>Bibliografía.</b> .....	<b>18</b>

# Introdución.

Recoller, automatizar, dixitalizar e conectar datos industriais (OT) a sistemas IT, pode ser unha tarefa complexa dada a falta de estándares, múltiples formatos de datos e protocolos propietarios.

Gracias ás ferramentas Open Source temos ó noso alcance un conxunto de ferramentas hardware e software, así como documentación, que permiten a calquer empresa, dende unha peme a unha gran multinacional, dixitalizar os seus procesos e democratizar o IoT e a Industria 4.0.

A través de Node-RED podemos traballar de maneira visual mostrando gráficamente o fluxo da información e as relacións e función que conectan varios dispositivos á vez, tanto de *hardware* como de servizos de Internet. Este sistema de representación pode axudar a visualizar gráficamente o fluxo da información de maneira que se poda programar sin escribir.

Nesta práctica realizaremos a codificación de dous microcontroladores, Arduino Nano e ESP32, os cales enviarán a través do porto serie unha serie de mensaxes aleatorias almacenadas no microcontrolador a unha matriz de puntos. Ademais, a aplicación permitirá a recepción de mensaxes a través dunha app móbil ( IoT MQTT Panel), que permitirá a súa saída por pantalla no momento da recepción.

# Funcionamento.

1. Un microcontrolador ESP32 de 38 pins será o encargado de almacenar unha serie de frases que serán enviadas aleatoriamente a través do porto serie cara outro microcontrolador ( Arduino Nano ).
2. Creamos unha matriz de frases en diferentes idiomas.

```
main.cpp x PIO Home platformio.ini
src > main.cpp > loop()
1 #include <Arduino.h>
2
3 // Matriz de frases aleatorias en diferentes idiomas
4 const char* frases[][4] = {
5     {"I hope you have a great day", "Espero que tengas un buen dia", "Espero que tenha"},
6     {"Who does not risk, does not win", "Quien no arriesga, no gana", "Quen non arrisc"},
7     {"You are worth more than you think", "Vales mas de lo que crees", "Vales mais do"},
8     {"Diamonds are formed under pressure", "Los diamantes se forman bajo presion", "O"},
9     {"Success is my future", "El exito es mi futuro", "O exito e o meu futuro"},
10    {"Do not live so that your presence is noticed, but so that your absence is felt", "No vivas pa"},
11    {"Nothing is like you, that is your power", "Nadie es como tu y ese es tu poder"},
12    {"Grapes must be trampled to make wine", "Las uvas deben ser pisoteadas para hacer vino"}
}
```

3. Xeramos a semente para a xeración aleatoria.

```
unsigned long currentDay = 0;

void setup() {
    Serial.begin(9600);
    randomSeed(analogRead(A0)); // Inicializa la semilla aleatoria
    currentDay = millis() / (24 * 60 * 60 * 1000); // Calcula el día actual
}

void loop() {
    unsigned long currentMillis = millis();
    unsigned long newDay = currentMillis / (24 * 60 * 60 * 1000); // Calcula el día actual basado en el tiempo tra

    if (newDay != currentDay) {
        currentDay = newDay;
        int index = random(sizeof(frases) / sizeof(frases[0])); // Genera un nuevo índice aleatorio para el nuevo día
    }
}
```

4. Cargamos a frase aleatoria no porto serie que será o que ó envío cara ó segundo microcontrolador ( Arduino Nano ).

```
Serial.println(String(frase_en));
delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
Serial.println(String(frase_es));
delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
Serial.println(String(frase_gal));
delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
Serial.println(String(frase_por));
delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
```

5. Un microcontrolador Arduino Nano recibirá unha serie de frases a través do porto serie e enviará ditas frases a unha matriz de puntos para a súa visualización.
6. Definición do número de elementos da matriz de puntos e desprazamento da mensaxe sobre as matrices.

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Define the number of devices we have in the chain and the hardware interface
// NOTE: These pin numbers will probably not work with your hardware and may
// need to be adapted
#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 8
#define CLK_PIN D5
#define DATA_PIN D7
#define CS_PIN D8

// HARDWARE SPI
MD_Parola P = MD_Parola(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);

uint8_t scrollSpeed = 50; // default frame delay value
textEffect_t scrollEffect = PA_SCROLL_LEFT;
textPosition_t scrollAlign = PA_LEFT;
uint16_t scrollPause = 2000; // in milliseconds

```

7. Defínese o buffer que compartiran tanto a función do porto serie coma a de desprazamento.

```

// Global message buffers shared by Serial and Scrolling functions
#define BUF_SIZE 250
char curMessage[BUF_SIZE] = { "" };
char newMessage[BUF_SIZE] = { "Esperando por unha mensaxe....." };
bool newMessageAvailable = true;

void readSerial(void)
{
    static char *cp = newMessage;

    while (Serial.available())
    {
        *cp = (char)Serial.read();
        if ((*cp == '\n') || (cp - newMessage >= BUF_SIZE-2)) // end of message character or full buffer
        {
            *cp = '\0'; // end the string
            // restart the index for next filling spree and flag we have a message waiting
            cp = newMessage;
            newMessageAvailable = true;
        }
        else // move char pointer to next position
            cp++;
    }
}

```

8. Establécese a configuración wifi e a conexión co broker MQTT.

```
// WiFi and MQTT settings
const char* ssid = "movil";
const char* password = "12345678";
const char* mqtt_server = "test.mosquitto.org";
const int mqtt_port = 1883;
const char* mqtt_topic = "ULAB/mensaxe1";
const char* mqtt_topic_out = "ULAB/mensaxe1_recibido";
WiFiClient wifiClient;
PubSubClient client(wifiClient);

void reconnect()
{
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("ESP8266Client"))
    {
      Serial.println("connected");
      client.subscribe(mqtt_topic);
    }
    else
    {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" retrying in 5 seconds");
      delay(5000);
    }
  }
}
```

```

void callback(char* topic, byte* payload, unsigned int length)
{
    Serial.print("Message received [");
    Serial.println(topic);

    // Clear the current message
    memset(newMessage, 0, sizeof(newMessage));

    // Copy the received message to the new message buffer
    for (int i = 0; i < length && i < BUF_SIZE - 1; i++)
    {
        newMessage[i] = (char)payload[i];
    }
    String respuesta = "Mensaje recibido en la pantalla ";
    client.publish(mqtt_topic_out, respuesta.c_str());
    // Print the message to the serial monitor
    Serial.println(newMessage);
    delay(100);
    Serial.println('\n');
    delay (5000);

    // Update the new message flag
    newMessageAvailable = true;
}

```

## 9. Configuración do porto serie e establecemento da conexión.

```
void setup()
{
  Serial.begin(115200);
  Serial.print("\n[Parola Scrolling Display]\nType a message for the scrolling display\nEnd message line with a

#if USE_UI_CONTROL
  uiDirection.begin();
  uiInvert.begin();
  pinMode(SPEED_IN, INPUT);

  doUI();
#endif // USE_UI_CONTROL

  P.begin();
  P.displayText(curMessage, scrollAlign, scrollSpeed, scrollPause, scrollEffect, scrollEffect);

  // Connect to Wi-Fi
  Serial.print("\nConnecting to Wi-Fi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected");

  // Connect to MQTT broker
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
}
```

## 10. Envío das mensaxes.

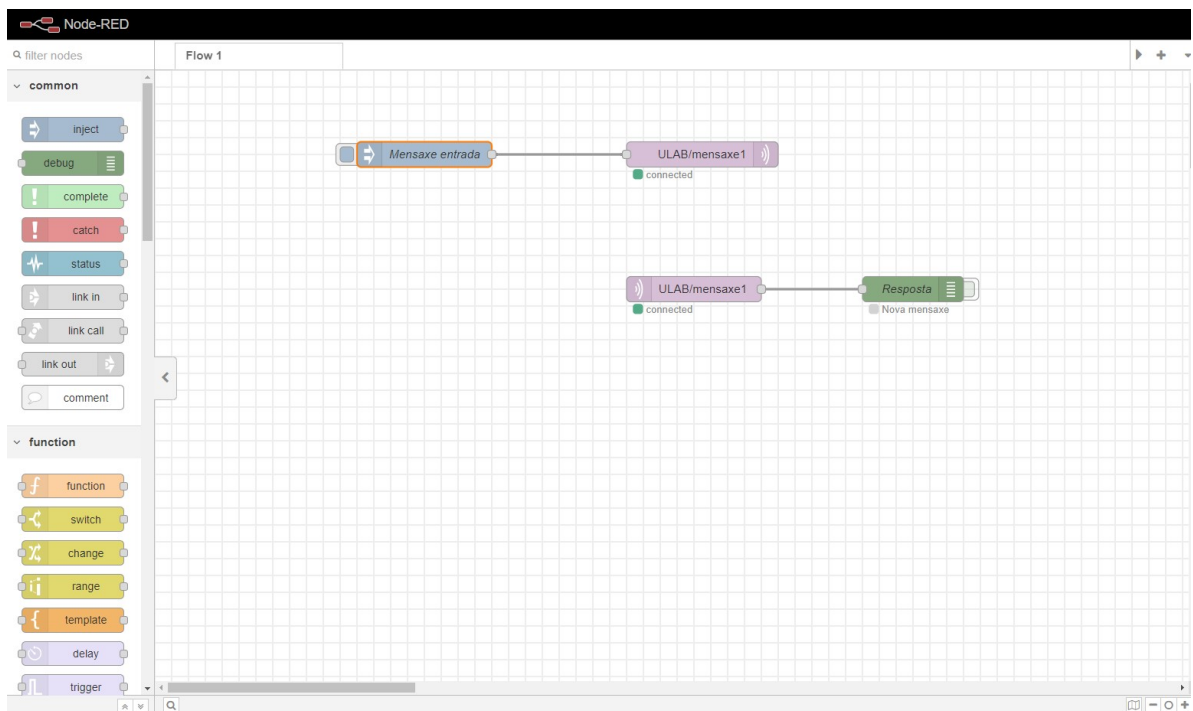
```
void loop()
{
#if USE_UI_CONTROL
  doUI();
#endif // USE_UI_CONTROL

  if (!client.connected())
  {
    reconnect();
  }
  client.loop();

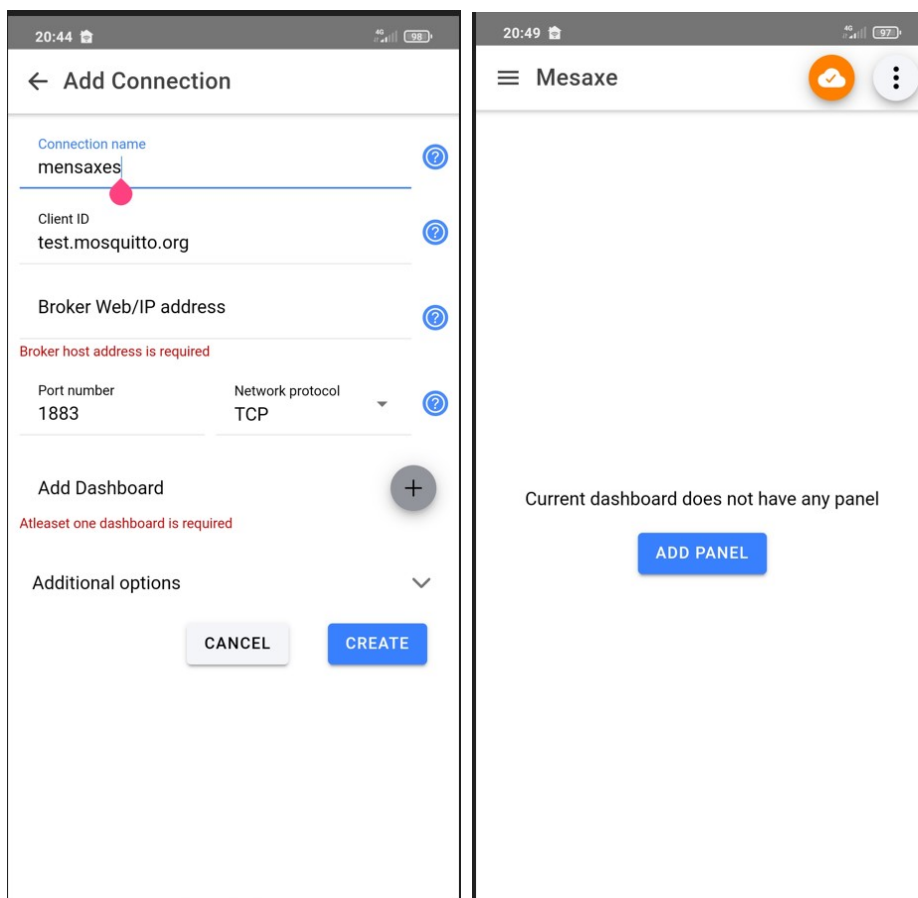
  if (P.displayAnimate())
  {
    if (newMessageAvailable)
    {
      strcpy(curMessage, newMessage);
      newMessageAvailable = false;
    }
    P.displayReset();
  }
}
```



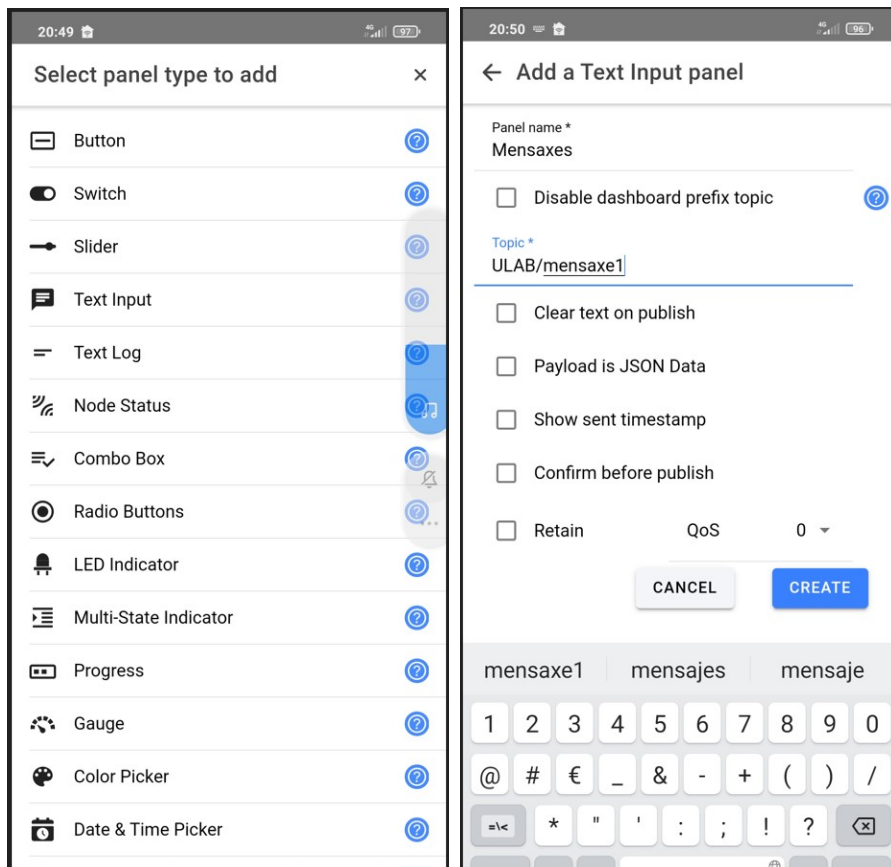
## 11. Programación en Node-Red do servizo MQTT.



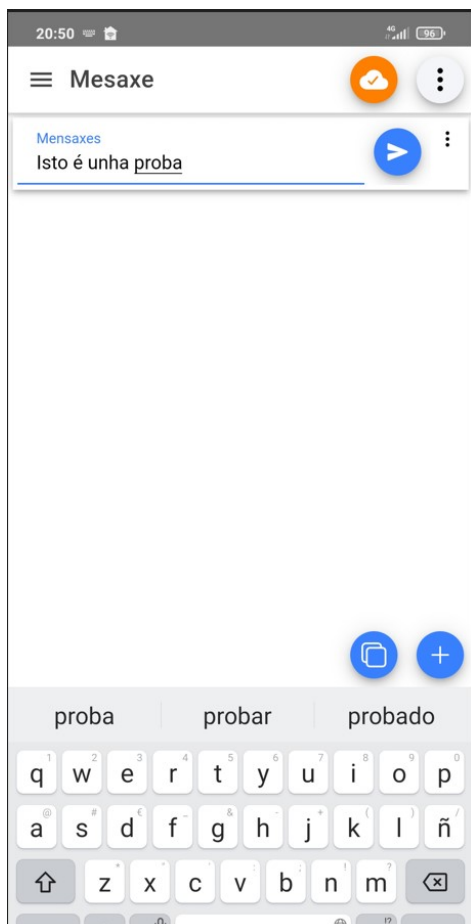
## 12. Configuración da app móbil para o envío de mensaxes. Descargamos do Play Store da App IoT MQTT Panel e procedemos á súa configuración.



13. Seleccionamos un panel de entrada de texto.



14. Xa podemos enviar as mensaxes que sairán na nosa matriz.





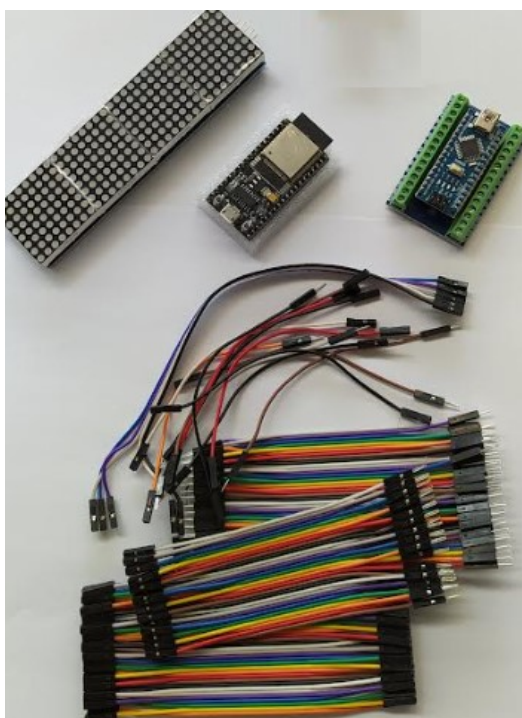
## Compoñentes da montaxe.

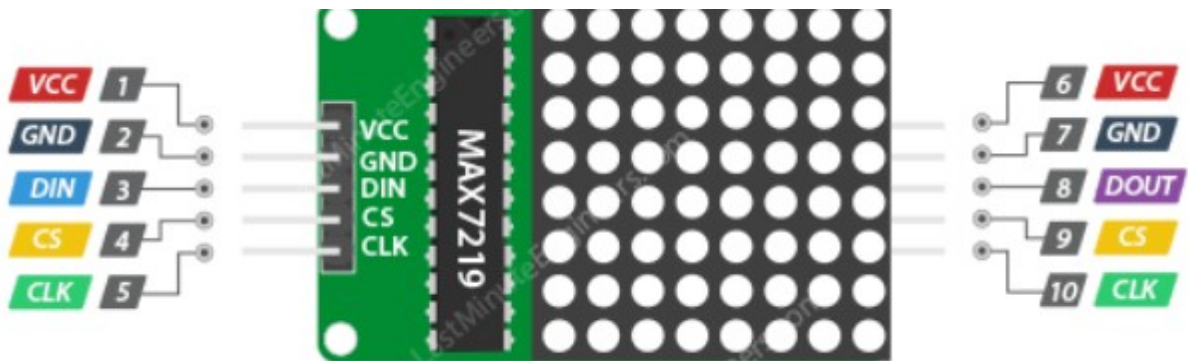
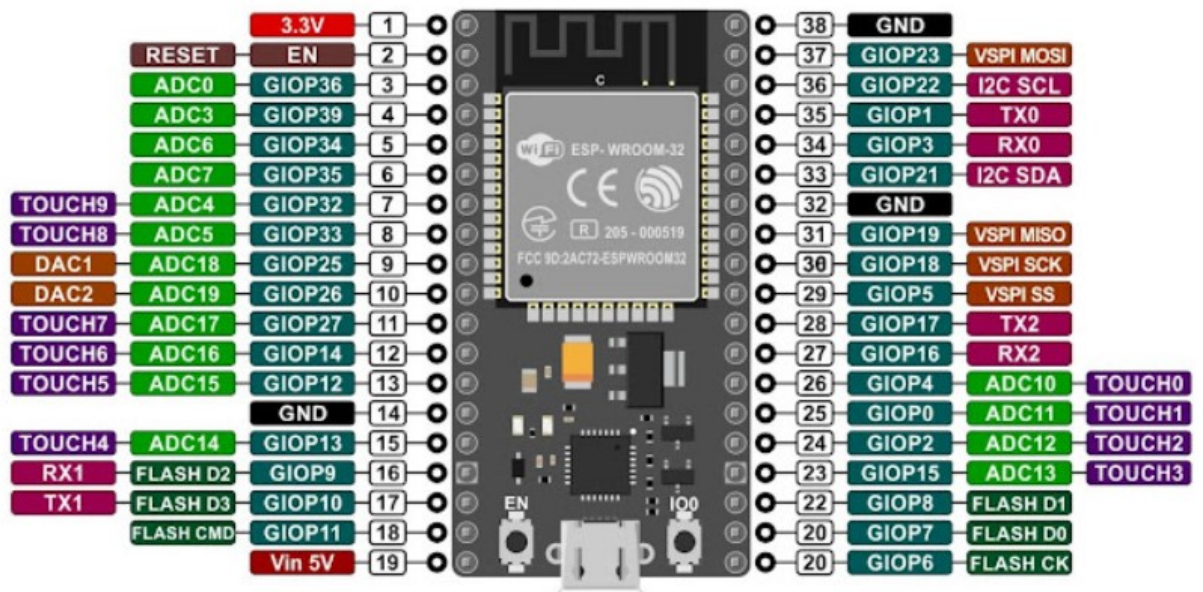
Placa de desenvolvemento ESP32 de 38 pines.

Módulo matriz de puntos MAX7219.

Arduino Nano con zócalo de parafusos.

Kit de cables de conexión.





## Software empregado.

Empregouse o software Visual Studio Code coa extensión PlatformIO.

Node-Red.

App IoT MQTT Panel.

## Programa dos Microcontroladores.

### Programación do Arduino Nano.

```
#include <MD_Parola.h>
#include <MD_MAX72xx.h>
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Define the number of devices we have in the chain and the hardware interface
// NOTE: These pin numbers will probably not work with your hardware and may
// need to be adapted
```

```

#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 8
#define CLK_PIN D5
#define DATA_PIN D7
#define CS_PIN D8

// HARDWARE SPI
MD_Parola P = MD_Parola(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);

uint8_t scrollSpeed = 50; // default frame delay value
textEffect_t scrollEffect = PA_SCROLL_LEFT;
textPosition_t scrollAlign = PA_LEFT;
uint16_t scrollPause = 2000; // in milliseconds

// Global message buffers shared by Serial and Scrolling functions
#define BUF_SIZE 250
char curMessage[BUF_SIZE] = { "" };
char newMessage[BUF_SIZE] = { "Esperando por unha mensaxe....." };
bool newMessageAvailable = true;

void readSerial(void)
{
    static char *cp = newMessage;

    while (Serial.available())
    {
        *cp = (char)Serial.read();
        if ((*cp == '\n') || (cp - newMessage >= BUF_SIZE-2)) // end of message character or full buffer
        {
            *cp = '\0'; // end the string
            // restart the index for next filling spree and flag we have a message waiting
            cp = newMessage;
            newMessageAvailable = true;
        }
        else // move char pointer to next position
            cp++;
    }
}

// WiFi and MQTT settings
const char* ssid = "movil";
const char* password = "12345678";
const char* mqtt_server = "test.mosquitto.org";
const int mqtt_port = 1883;
const char* mqtt_topic = "ULAB/mensaxe1";
const char* mqtt_topic_out = "ULAB/mensaxe1_recibido";
WiFiClient wifiClient;
PubSubClient client(wifiClient);

void reconnect()
{
    while (!client.connected())
    {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP8266Client"))
        {
            Serial.println("connected");
            client.subscribe(mqtt_topic);
        }
        else
        {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" retrying in 5 seconds");
            delay(5000);
        }
    }
}

```

```

}
}

void callback(char* topic, byte* payload, unsigned int length)
{
  Serial.print("Message received [");
  Serial.println(topic);

  // Clear the current message
  memset(newMessage, 0, sizeof(newMessage));

  // Copy the received message to the new message buffer
  for (int i = 0; i < length && i < BUF_SIZE - 1; i++)
  {
    newMessage[i] = (char)payload[i];
  }
  String respuesta = "Mensaje recibido en la pantalla ";
  client.publish(mqtt_topic_out, respuesta.c_str());
  // Print the message to the serial monitor
  Serial.println(newMessage);
  delay(100);
  Serial.println('\n');
  delay (5000);

  // Update the new message flag
  newMessageAvailable = true;
}

void setup()
{
  Serial.begin(115200);
  Serial.print("\n[Parola Scrolling Display]\nType a message for the scrolling display\nEnd message line with a
  newline");

#ifdef USE_UI_CONTROL
  uiDirection.begin();
  uiInvert.begin();
  pinMode(SPEED_IN, INPUT);

  doUI();
#endif // USE_UI_CONTROL

  P.begin();
  P.displayText(curMessage, scrollAlign, scrollSpeed, scrollPause, scrollEffect, scrollEffect);

  // Connect to Wi-Fi
  Serial.print("\nConnecting to Wi-Fi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected");

  // Connect to MQTT broker
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
}

void loop()
{
#ifdef USE_UI_CONTROL
  doUI();
#endif // USE_UI_CONTROL

```

```

if (!client.connected())
{
    reconnect();
}
client.loop();

if (P.displayAnimate())
{
    if (newMessageAvailable)
    {
        strcpy(curMessage, newMessage);
        newMessageAvailable = false;
    }
    P.displayReset();
}
}

```

## Programación da ESP32:

```

#include <Arduino.h>

// Matriz de frases aleatorias en diferentes idiomas
const char* frases[][4] = {
    {"I hope you have a great day", "Espero que tengas un buen dia", "Espero que tenhas",
    un bo día", "Espero que tenha um bom dia"},
    {"Who does not risk, does not win", "Quien no arriesga, no gana", "Quen non arrisca",
    ca, non ganha", "Quem nao arrisca nao ganha"},
    {"You are worth more than you think", "Vales mas de lo que crees", "Vales mais do",
    que cres", "Voce vale mais do que pensa"},
    {"Diamonds are formed under pressure", "Los diamantes se forman bajo presion", "Os",
    diamantes formanse baixo presion", "Os diamantes sao formados sob pressao"},
    {"Success is my future", "El exito es mi futuro", "O exito e o meu futuro",
    "O sucesso e o meu futuro"},
    {"Do not live so that your presence is noticed, but so that your absence is felt", "No vivas",
    para que tu presencia se note, sino para que tu ausencia se sienta", "Non vivas para que se note a",
    tua presenza, senon para que se sinta a tua ausencia", "Nao viva para que sua presença seja notada,",
    mas para que sua ausencia seja sentida"},
    {"Nothing is like you, that is your power", "Nadie es como tu y ese es tu poder",
    "Niguen e coma ti e ese e o teu poder", "Ningem e como voce e esse e o seu poder"},
    {"Grapes must be trampled to make wine", "Las uvas deben ser pisoteadas para hacer vino",
    "Hai que pisar as uvas para facer vinho", "Uvas devem ser pisadas para fazer vinho"},
    {"Better than yesterday but worse than tomorrow", "Mejor que ayer, peor que mañana",
    "Mellor que onte, peor que maña", "Melhor que ontem, pior que amanha"},
    {"Life is about getting up", "La vida se trata de levantarse", "A vida e levantarse",
    "A vida e sobre se levantar"},
    {"Sweat has its reward", "El sudor tiene su recompensa", "A suor ten a sua recom-",
    pensa", "O suor tem a sua recompensa"},
    {"Make each day your masterpiece", "Haz de cada dia tu obra maestra", "Fai de cada",
    dia a tua obra maestra", "Faca de cada dia sua obra prima"},
    {"Hard times never last but hard people do", "Los tiempos dificiles nunca duran pero la gente",
    dura si", "Os tempos duros nunca duran pero a xente dura si", "Os tempos duros",
    nunca duram mas a gente sim"},
    {"The best way to predict the future is to invent it", "La mejor forma de predecir el futuro es",
    crearlo", "A mellor maneira de predicir o futuro e crealo", "A melhor forma de",
    predecir o futuro e crearlo"},
    {"Grey days are colour too", "Los dias grises tambien son colores", "Os dias grises",
    tamen son cores", "Os dias cinzas tambem sao cores"},
    {"Fight for your dreams", "Lucha por tus sueños", "Loita polos teus sonhos",
    "Luta pelos teus sonhos"},
    {"There is a tear for every laugh", "Hay una lagrima por cada risa", "Hai unha",
    bagoa por cada risa", "Ha uma lagrima por cada risada"},
    {"With sweat everything is achieved", "Con sudor todo se consigue", "Con suor todo",
    se consigue", "Com suor tudo se consegue"}
}

```

```

{"Cold is psychological", "El frio es psicologico", "O frio e psicoloxico", "O frio e psicologico"},
{"Happy mind active body", "Mente feliz cuerpo activo", "Mente feliz corpo ativo"},
{"Turn your wounds into wisdom", "Transforma tus heridas en sabiduria", "Transforma as tuas feridas en sabiduria", "Transforme suas feridas em sabedoria"},
{"The best way to predict the future is creating it", "La mejor forma de predecir el futuro es creandolo", "A mellor maneira de predicir o futuro e creandoo", "A melhor maneira de prever o futuro e cria-lo"},
{"The way to success is the attitude", "El camino al exito es la actitud", "O caminho ao éxito e a actitude", "O caminho para o sucesso é a atitude"},
{"Live the moment", "Vive el momento", "Vive o momento", "Viva o momento"},
{"Do not stop smiling", "No dejes de sonreir", "Non deixes de sorrir", "Nao deixes de sorrir"},
{"The most difficult opponent is in your head", "El rival mas dificil esta en tu cabeza", "O rival mais dificil esta na tua cabeza", "O rival mais dificil estar na sua cabeca"},
{"Do it and if you are afraid do it with fear", "Hazlo y si te da miedo hazlo con miedo", "Faino e se che da medo faino con medo", "Faca e se tiver medo faca com medo"},
{"Success is the sum of small efforts repeated day after day", "El exito es la suma de pequenos esfuerzos repetidos dia tras dia", "O exito e a suma de pequenos esforzos repetidos dia tras dia", "Sucesso e a soma de pequenos esforcos repetidos dia apos dia"},
{"Passion for what you do will make you understand that limits are mental", "La pasion por lo que haces te hara entender que los limites son mentales", "A paixon polo que fas farache entender que os limites son mentais", "A paixao pelo que faz vai te fazer entender que os limites sao mentais"},
{"Every day is a new opportunity to change your life", "Cada dia es una nueva oportunidad para cambiar tu vida", "Cada dia e unha nova oportunidade para cambiar a tua vida", "Cada dia e uma nova oportunidade para mudar sua vida"},
{"Dont stop until you are proud of your work", "No pares hasta estar orgulloso de tu trabajo", "Non o deixes ata estar orgulloso do teu traballo", "Nao pare ate se orgulhar do seu traballo"},
{"Your goals are bigger than your fears", "Tus metas son mas grandes que tus miedos", "As tuas metas son mais grandes que os teus medos", "Seus objetivos sao maiores que seus medos"},
{"Always do your best what you grow now you will reap later", "Da siempre lo mejor de ti lo que plantas ahora lo cosecheras mas tarde", "Fai sempre o mellor que podes o que plantas agora colle-ras mais tarde", "Faca sempre o seu melhor o que planta agora colhera depois"},
{"Stop thinking that you cannot achieve it", "Deja de creer que no puedes lograrlo", "Deixa de crer que non podes conseguilo", "Deixa de pensar que voce nao pode conseguir"},
{"Focus on where you want to go to and not on what you fear", "Centrate hacia donde quieras ir y no en lo que temas", "Centrate cara onde queiras ir e non no que temes", "Foca onde voce queira ir e nao no que voce teme"},
{"Yes, you can", "Si, vosotros podeis", "Si, vos podedes", "Sim, voces podem"},
{"Whatever you have decided to do do it with all your might", "Cualquier cosa que hayas decidido hacer hazla con todas tus fuerzas", "Calquera cousa que teñas decidido facer faina con todas as tuas forzas", "Qualquer coisa que tenhas decidido fazer faca com todas as suas forcas"},
{"If you only know how to do one thing do it perfectly", "Si solo sabes hacer una cosa hazla a la perfeccion", "Se so sabes facer unha cousa faina a perfeccion", "Se voce so sabe fazer uma coisa faz ella com perfeicao"},
{"There is always a point of light in every storm cloud", "Siempre hay un punto de luz en cada nube de tormenta", "Sempre hai un punto de luz en cada nube de tormenta", "Ha sempre um ponto de luz em cada nuvem de tempestade"},
{"All can be achieved", "Todo se consigue", "Todo se consegue"},
{"Cold is a way of life", "El frio es una forma de vida", "O frio e unha forma de vida", "O frio e uma forma de vida"},
{"The worst rival is in your head", "El peor rival esta en tu cabeza", "O peor rival esta na tua cabeza", "O peor rival esta na sua cabeca"},
{"Dont be afraid to be alone goals are personal", "No temas estar solo , las metas son personales", "Non temas estar so as mentas son persoais", "Nao temas estar sozinho as metas sao pessoais"}

```



```

{"No matter how slow you go if you do not stop          ", "No importa que tan lento vayas mientras no te
detengas          ", "Non importa que tan lento vaias mentras non te deteñas          ", "Nao importa
quao lento voce va enquanto nao te detenhas          "},
{"Dont count the days make the days count          ", "No cuentes los dias haz que los dias cuenten
          ", "Non contes os dias fai que os dias conten          ", "Nao conte os dias faca que os dias contem
          "},
{"I know the change you want to see in the world          ", "Se el cambio que quieres ver en el mundo
          ", "Sei o cambio que quieres ver no mundo          ", "Seja a mudanca que voce quer ver no mundo
          "},
{"You live every day you die once          ", "Se vive todos los dias y se muere una vez
          ", "Vivese todos os dias e morrese unha vez          ", "Se vive todos os dias e se morre uma vez
          "},
{"Every day is a new opportunity to change your life          ", "Cada dia es una nueva oportunidad para
cambiar tu vida          ", "Cada dia e unha nova oportunidade para cambiar a tua vida
          ", "Cada dia uma nova oportunidade para mudar a tua vida          "},
{"We trust you          ", "Confiamos en vosotros          ", "Confiamos en vos
          ", "Confiamos em voces          "
};
unsigned long currentDay = 0;

void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(A0)); // Inicializa la semilla aleatoria
  currentDay = millis() / (24 * 60 * 60 * 1000); // Calcula el día actual
}

void loop() {
  unsigned long currentMillis = millis();
  unsigned long newDay = currentMillis / (24 * 60 * 60 * 1000); // Calcula el día actual basado en el tiempo
  transcurrendo

  if (newDay != currentDay) {
    currentDay = newDay;
    int index = random(sizeof(frases) / sizeof(frases[0])); // Genera un nuevo índice aleatorio para el nuevo día

    const char* frase_en = frases[index][0]; // Frase en inglés
    const char* frase_es = frases[index][1]; // Frase en español
    const char* frase_gal = frases[index][2]; // Frase en galego
    const char* frase_por = frases[index][3]; // Frase en portugués

    Serial.println(String(frase_en));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
    Serial.println(String(frase_es));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
    Serial.println(String(frase_gal));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
    Serial.println(String(frase_por));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
  }
  else {
    const char* frase_en = frases[0][0]; // Frase en inglés
    const char* frase_es = frases[0][1]; // Frase en español
    const char* frase_gal = frases[0][2]; // Frase en galego
    const char* frase_por = frases[0][3]; // Frase en portugués

    Serial.println(String(frase_en));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
    Serial.println(String(frase_es));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
    Serial.println(String(frase_gal));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
    Serial.println(String(frase_por));
    delay(10000); // Espera 10 segundos antes de mostrar la siguiente frase
  }

  delay(300000); // Espera 5 minutos antes de mostrar la siguiente frase
}

```

# Bibliografía.

Desarrollo de aplicaciones IoT en la nube para Arduino y ESP8266. Tomás Domínguez Mínguez. Ed. Marcombo.