

<https://www.hivemq.com/demos/websocket-client/>

<https://wokwi.com/projects/394695923034785793>

```
#include <WebServer_ESP32_ENC.h>
#include <WebServer_ESP32_ENC.hpp>
#include <Key.h>
#include <Keypad.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>
#include <DHT_U.h>
#include <WiFi.h>
#include <PubSubClient.h>
```

```
#define ROW_NUM    4 // Cuatro filas
#define COLUMN_NUM 4 // Cuatro columnas
#define DHTPIN 22    // D27 del ESP27 Dev Module
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

```
// PANTALLA OLED
#define OLED_RESET 16
Adafruit_SSD1306 display(OLED_RESET);
```

```
char keys[ROW_NUM][COLUMN_NUM] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
```

```
const char* WIFI_SSID = "Wokwi-GUEST";
const char* WIFI_PASSWORD = "";
const char* MQTT_CLIENT_ID = "access-control-demo";
const char* MQTT_BROKER = "broker.mqttdashboard.com";
const char* MQTT_USER = "";
const char* MQTT_PASSWORD = "";
const char* MQTT_TOPIC = "access-controlinfohub";
// const char* WIFI_SSID = "AP_asix";
// const char* WIFI_PASSWORD = "AP_asix2023";
// const char* mqttServer = "10.0.110.x";
// const int mqttPort = "1883";
// const char* mqttUser = "sensor";
// const char* mqttPassword = "Woah!";
//
```

```
WiFiClient espClient;
PubSubClient client(espClient);
```

```
byte pin_rows[ROW_NUM] = {19, 18, 5, 17}; // GPIO19, GPIO18, GPIO5, GPIO17 conectados
a los pines de las filas
byte pin_column[COLUMN_NUM] = {16, 4, 0, 2}; // GPIO16, GPIO4, GPIO0, GPIO2
conectados a los pines de las columnas
```

```
Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column, ROW_NUM,
COLUMN_NUM );
```

```
void setup() {
  Serial.begin(9600);
  connectToWiFi();
  client.setServer(MQTT_BROKER, 1883);
  connectToMQTT();
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Dirección 0x3C para 128x32
  display.display();
  display.clearDisplay();
  display.display();
  display.setTextSize(1);
  display.setTextColor(WHITE);
}

int storedPassword = 19921124; // Contraseña previamente almacenada para comparación
int userInput = 0; // Variable para almacenar el número introducido por el usuario
int characterCount = 0;
int contador_fallos = 0;
void connectToWiFi() {
  display.setCursor(0, 0);
  Serial.print("Conectando a WiFi");
  display.println("Conectando a WiFi");
  display.display();
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    display.setCursor(0, 0);
    Serial.print(".");
    delay(100);
  }
  display.setCursor(0, 0);
  Serial.println(" ¡Conectado!");
  display.println(" ¡Conectado!");
  display.display();
}
```

```
void connectToMQTT() {
  display.setCursor(0, 0);
  Serial.print("Conectando al servidor MQTT... ");
  display.println("Conectando al servidor MQTT...");
  display.display();
  while (!client.connected()) {
    if (client.connect(MQTT_CLIENT_ID, MQTT_USER, MQTT_PASSWORD)) {
      display.setCursor(0, 0);
      display.println("Conectado al servidor MQTT!");
      display.display();
    }
  }
}
```

```

    client.subscribe(MQTT_TOPIC);
} else {
    display.setCursor(0, 0);
    display.println("No se pudo conectar al servidor MQTT, reintentando...");
    display.display();
    delay(5000);
}
}
}
}

```

```

void loop() {
    char key = keypad.getKey();
    display.setCursor(0, 0);
    display.println("Escribe la contrase a:\n");
    display.display();
    if (key) {
        if (key >= '0' && key <= '9') {
            display.print(key);
            display.display();
            client.publish(MQTT_TOPIC, "*");
            Serial.print("*");
            display.print("*");
            display.display();
            userInput = userInput * 10 + (key - '0');
            characterCount++;
        } else {
            display.println("\nCaracteres inv alidos");
            display.display();
            delay(1500);
            display.clearDisplay();
            Serial.println("\nCaracteres inv alidos");
            display.println("\nCaracteres inv alidos");
            display.display();
        }
    }
}
}

```

```

if (characterCount == 8) {
    if (userInput == storedPassword) {
        display.println("\n Contrase a correcta!");
        display.display();
        display.clearDisplay();
        Serial.println("\n Contrase a correcta!");
        // Publicar mensaje MQTT de acceso concedido
        client.publish(MQTT_TOPIC, "Acceso Concedido");
    } else {
        contador_fallos++;
        if (contador_fallos <= 3){
            display.println("\nContrase a incorrecta, intenta de nuevo");
            display.display();
            Serial.println("\nContrase a incorrecta, intenta de nuevo");
            delay(1500);
            display.clearDisplay();
        }
    }
}
}

```

```
// Publicar mensaje MQTT de acceso denegado
client.publish(MQTT_TOPIC, "Acceso Denegado");
}
else{
display.println("\n¡ALARMA!");
display.display();
Serial.println("\nSe ha alcanzado el límite de fallos de contraseña");
display.display();
delay(500);
display.clearDisplay();
// Publicar mensaje MQTT de acceso denegado
client.publish(MQTT_TOPIC, "¡ALARMA!");
client.publish(MQTT_TOPIC, "Se ha alcanzado el límite de fallos de contraseña");
}
userInput = 0; // Restablecer el valor de userInput para la próxima entrada
characterCount = 0; // Reiniciar el contador de caracteres
}
}
}
```