



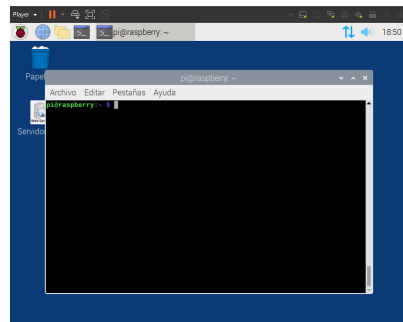
Universidad Politécnica de Madrid

Uso del IoT para construir tú mismo un hogar digital

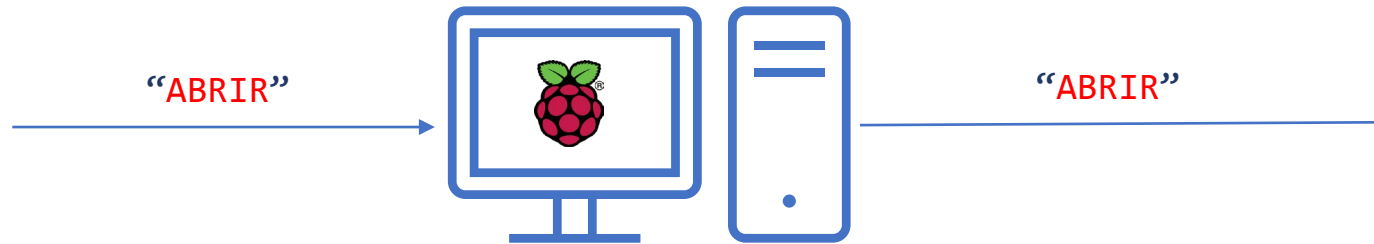
Recepción de mensajes MQTT mediante ESP8266

Iván Pau

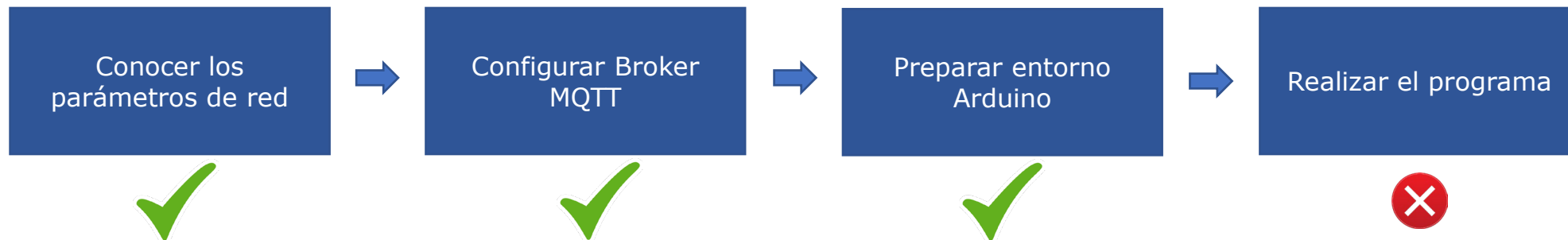
Escenario



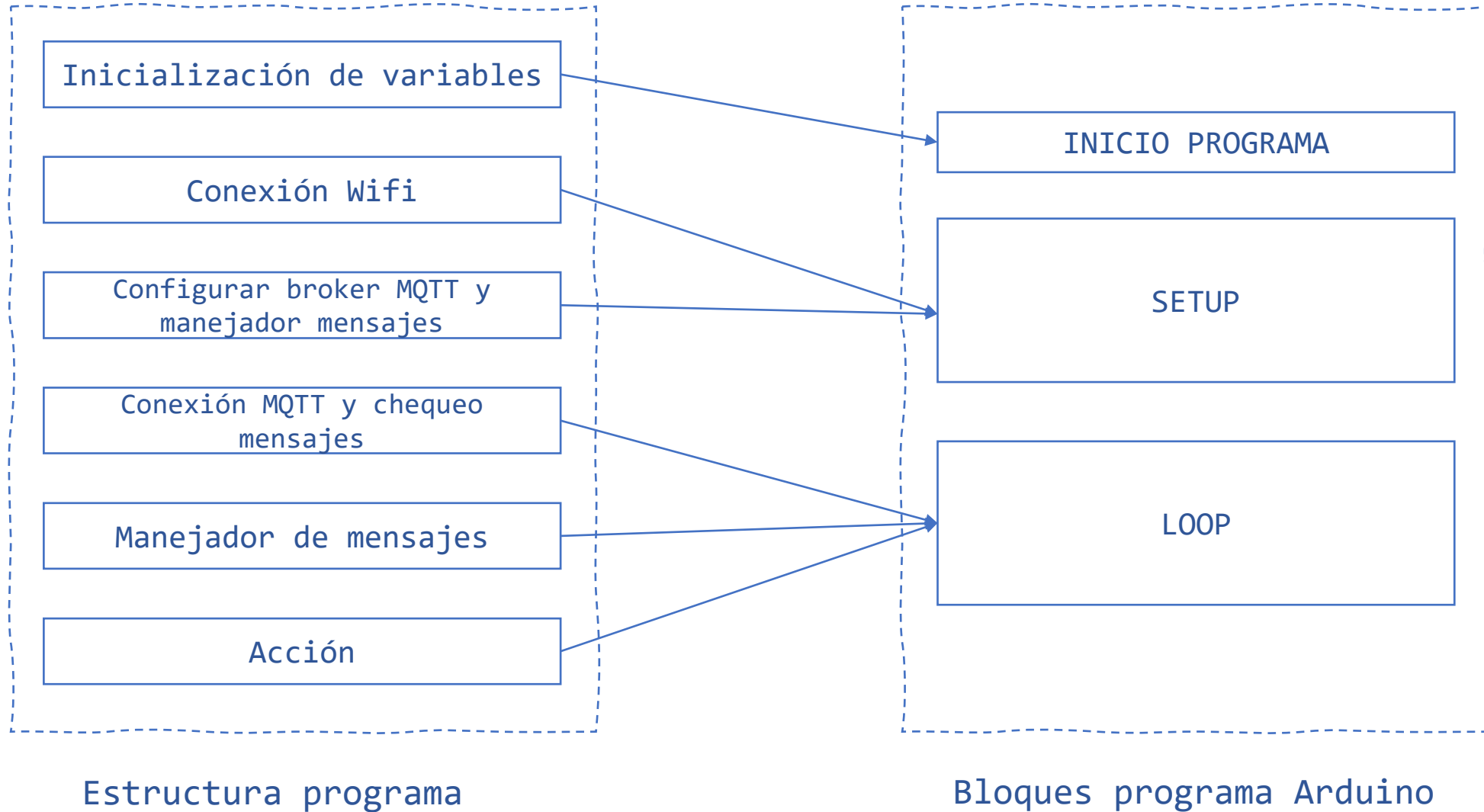
Publicando en:
casa/puerta/accion



Suscrito a:
casa/puerta/accion



Programa de recepción



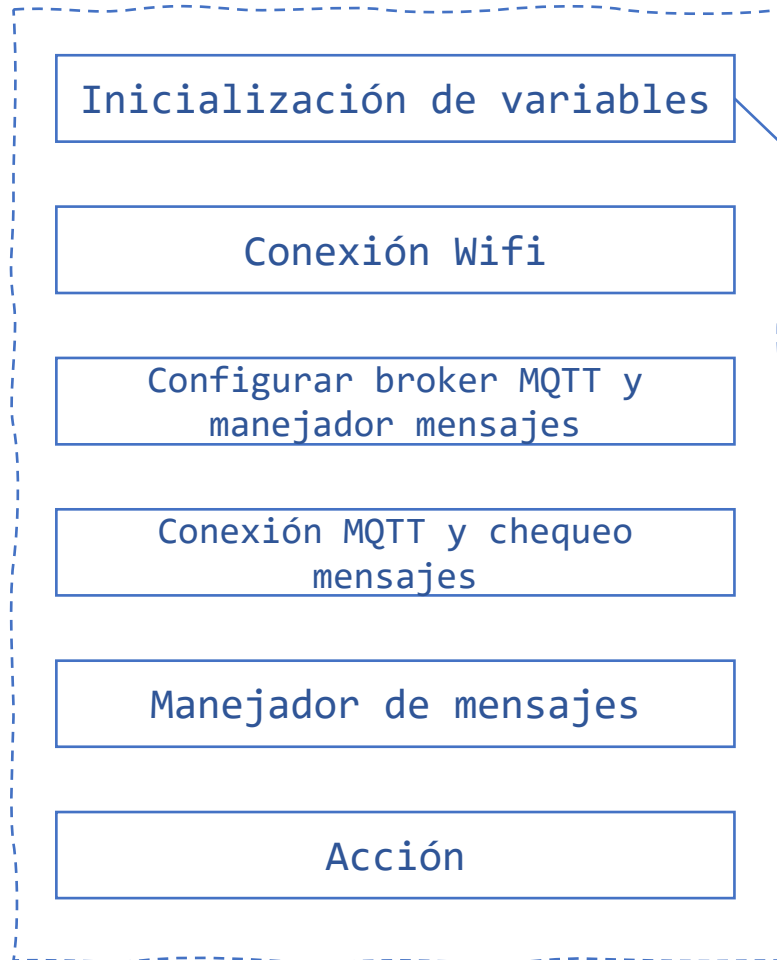
Programa de recepción



POLITÉCNICA



Universidad Politécnica de Madrid



Estructura programa

```
#include <ESP8266WiFi.h> // Biblioteca para esp8266
#include <PubSubClient.h> // Biblioteca para el cliente MQTT

// Constantes programa
#define MSG_BUFFER_SIZE  (50)
#define NUMERO_REINTENTOS (5)
#define TIEMPO_RECONEXION (5000)
#define OPCION_ABRIR "ABRIR"
#define OPCION_BLOQUEAR "BLOQUEAR"
#define TOPIC_SUBSCRIPCION "casa/puerta/accion"

// Parametros de conexión Wifi
const char* ssid = "LabIoT1";
const char* password = "XXXXXXXXXX";
// IP de la ESP8266
IPAddress wifiIP(192, 168, 1, 50);
// Máscara de red
IPAddress wifiNET(255, 255, 255, 0);
// Dirección IP del encaminador
IPAddress wifiON(192, 168, 1, 1);
//IP del broker
IPAddress mqtt_server (192, 168, 1, 145);

// Variables globales
WiFiClient clienteWIFI;
PubSubClient clienteMQTT(clienteWIFI);
```

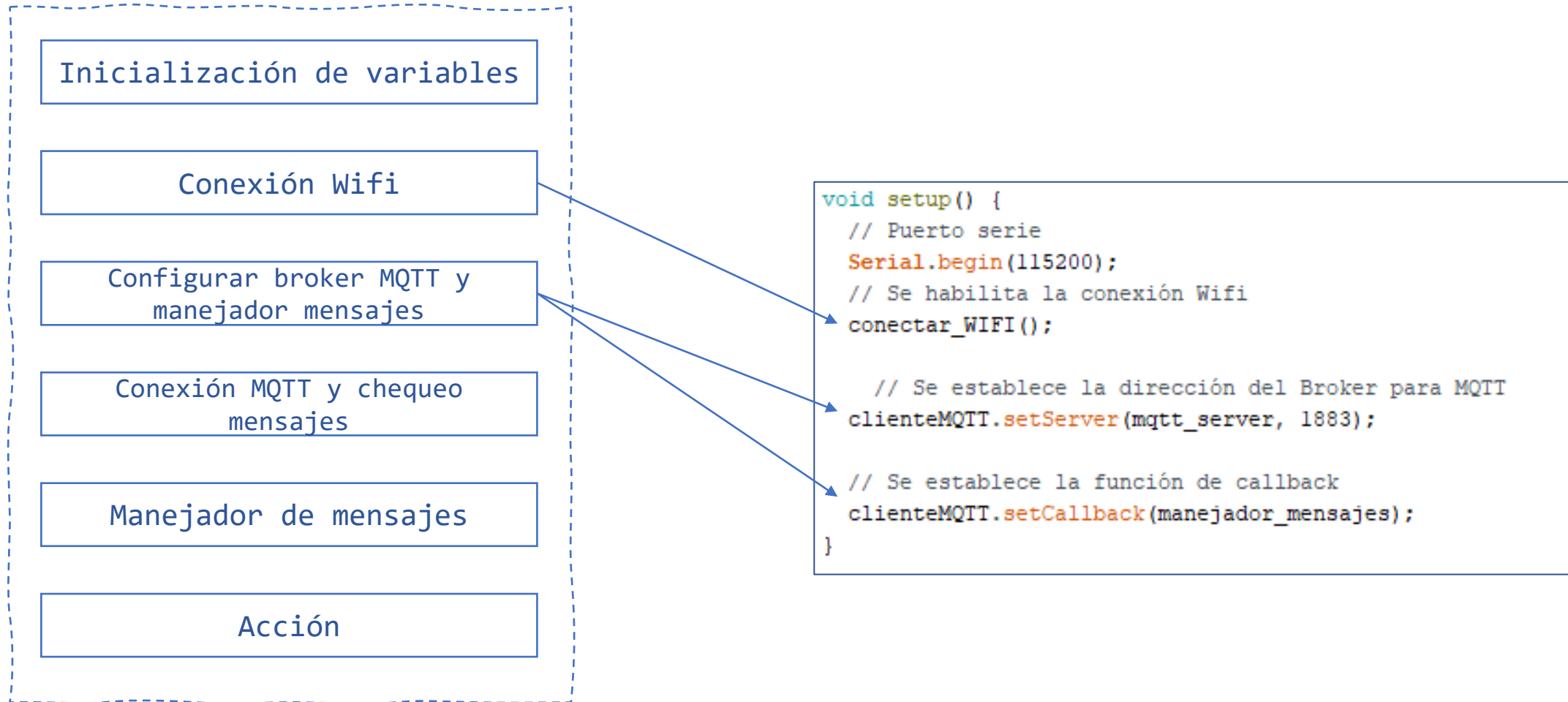
Programa de recepción



POLITÉCNICA



Universidad Politécnica de Madrid



Estructura programa

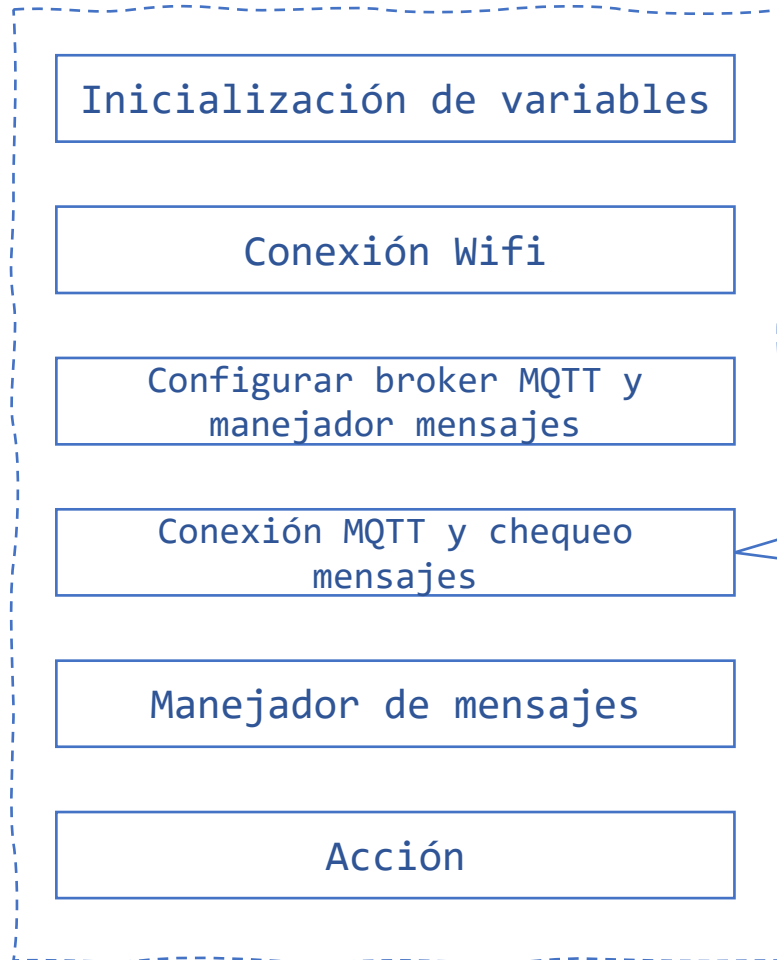
Programa de recepción



POLITÉCNICA



Universidad Politécnica de Madrid



```
void loop() {  
  
    // Se asegura de que existe conexión con el broker  
    if (!clienteMQTT.connected()) {  
        conectar_MQTT();  
    }  
  
    // Se comprueban mensajes entrantes  
    clienteMQTT.loop();  
  
}
```

Estructura programa

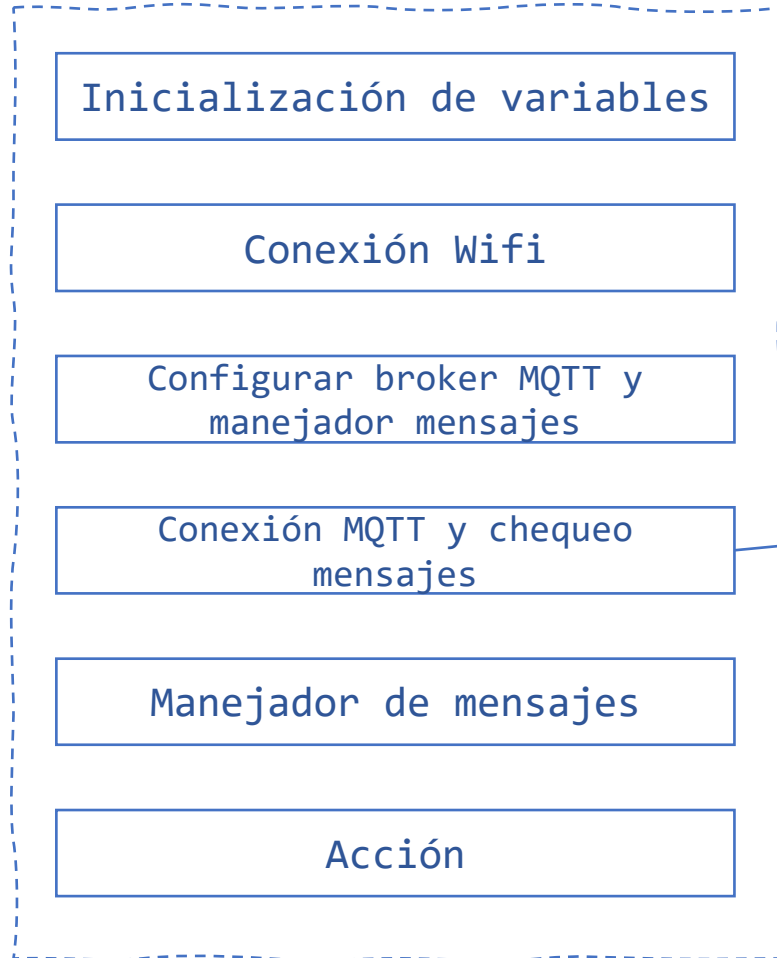
Programa de recepción



POLITÉCNICA



Universidad Politécnica de Madrid



Estructura programa

```
void conectar_MQTT() {  
  
    for (int i=0 ; i<NUMERO_REINTENTOS ; i++) {  
  
        // Creación de un ID de cliente  
        String clientId = "esp32Client-1";  
  
        // Intento de conexión  
        if (clienteMQTT.connect(clientId.c_str())) {  
            Serial.println("Conectado");  
  
            clienteMQTT.subscribe(TOPIC_SUBSCRIPCION);  
  
            i=NUMERO_REINTENTOS;  
  
        } else {  
  
            Serial.print("Error al conectar. Resultado: ");  
            Serial.print(clienteMQTT.state());  
            Serial.println(". Se volverá a intentar en 5 segundos. ");  
  
            delay(TIEMPO_RECONEXION);  
  
        }  
    }  
}
```

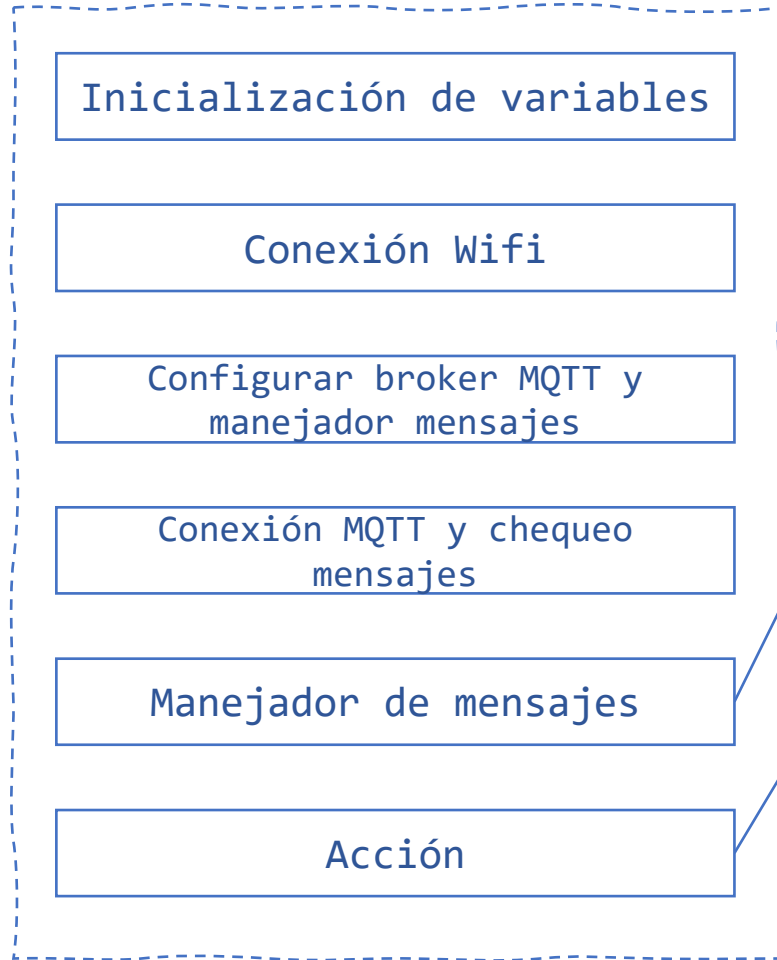
Programa de recepción



POLITÉCNICA



Universidad Politécnica de Madrid



```
void manejador_mensajes (char *canal, byte *mensaje, unsigned int longitud) {  
    Serial.print("Ha llegado un mensaje para el canal: ");  
    Serial.println(canal);  
    tratamiento_mensaje (mensaje, longitud);  
}  
  
void tratamiento_mensaje (byte *mensaje, unsigned int longitud) {  
  
    char bufferTemporal [MSG_BUFFER_SIZE];  
    strncpy (bufferTemporal, (char *) mensaje, longitud);  
    bufferTemporal [longitud] = '\0';  
  
    Serial.print("Mensaje recibido: ");  
    Serial.println(bufferTemporal);  
  
    if (strcmp(bufferTemporal, OPCION_ABRIR) == 0) {  
        Serial.println("Se procede a abrir la puerta.");  
    } else if (strcmp(bufferTemporal, OPCION_BLOQUEAR) == 0) {  
        Serial.println("Se procede a cerrar la puerta.");  
    } else {  
        Serial.println("*** ERROR ***. Formato del mensaje recibido erróneo.");  
    }  
  
    Serial.println("-----");  
}
```

Estructura programa



POLITÉCNICA

mooob
Universidad Politécnica de Madrid

