



Pluva

¿Todas las mañanas te preguntas si coger el paraguas o no?

¿Y si mediante un LED pudieras saber de un vistazo qué tiempo va a hacer sin sacar el móvil de tu bolsillo?



@adirael

angel.alonso
lucia.gomez



@malakatonez

@tecnilogica.com

Arquitectura

ESP8266

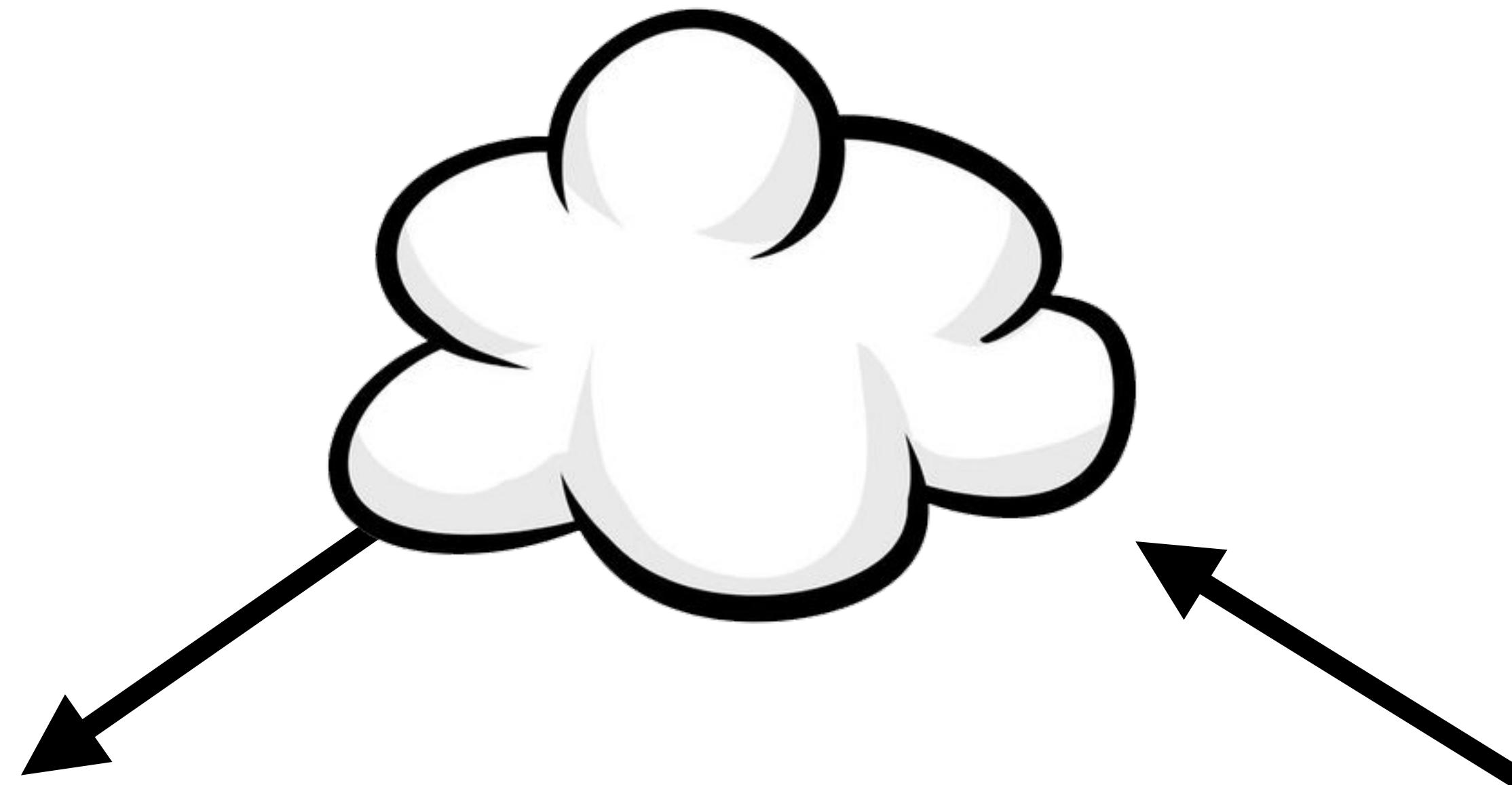
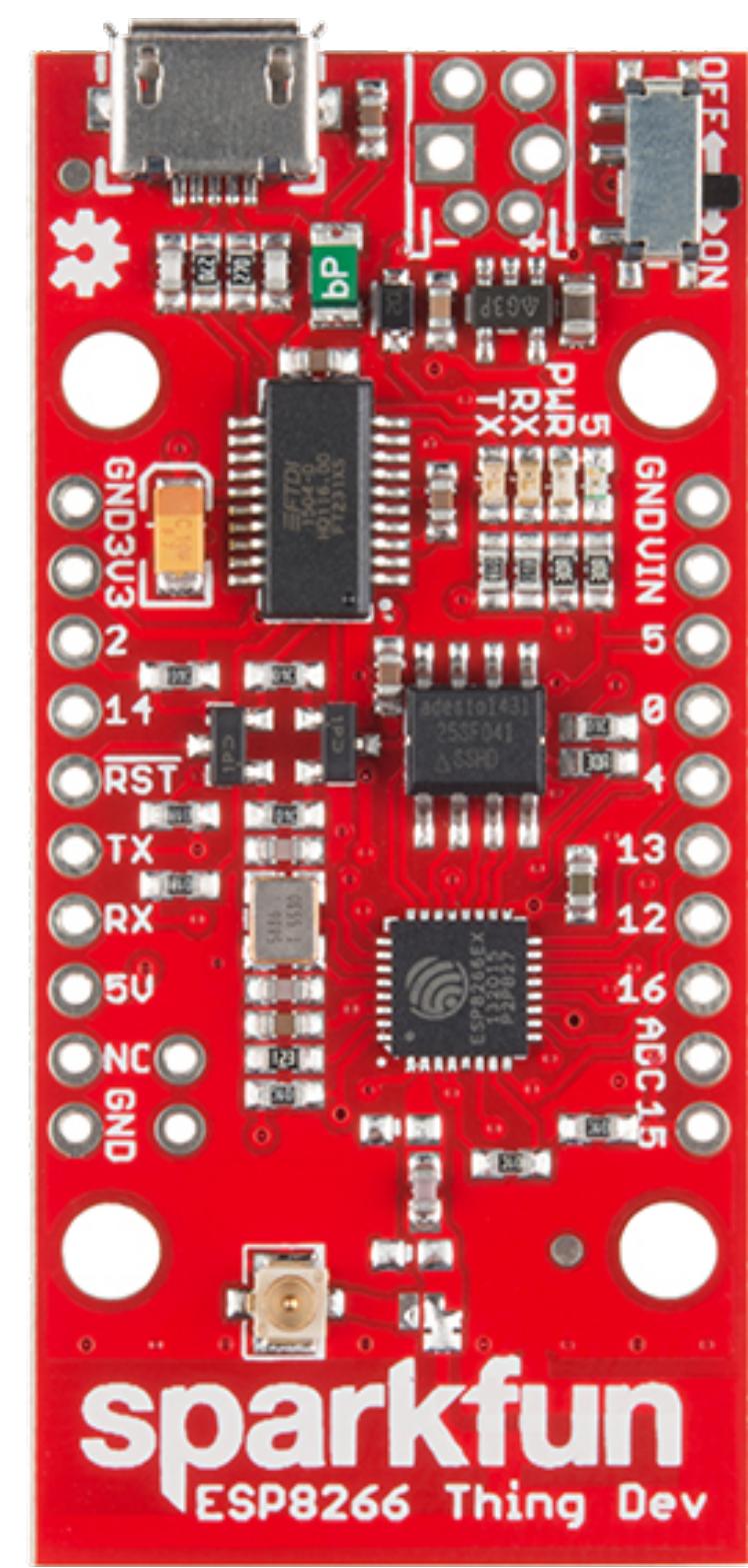
- IEEE 802.11 b/g/n Wi-Fi
- TCP/IP Stack
- 32-bit RISC CPU 80 MHz
- 64 KiB + 96 KiB de RAM
- 1 pixel = 3 bytes de RAM
- 16 pines de GPIO

Servidor web

- Diferenciación de clientes
- API del tiempo de Yahoo
- Versión demo en PHP

Configurador

- JAVA
- Aplicación Android
- Conecta con Servidor web



Make it rain!

Implementación en cuatro etapas

1. Los colores del arcoíris (v1.0)
2. El Internet del Thing (v1.1)
3. Papers, please (v1.2)
4. ¿Qué tiempo hará hoy? (v1.3)

<https://github.com/tecnilogica/pluva/tags>

Los colores del arcoíris

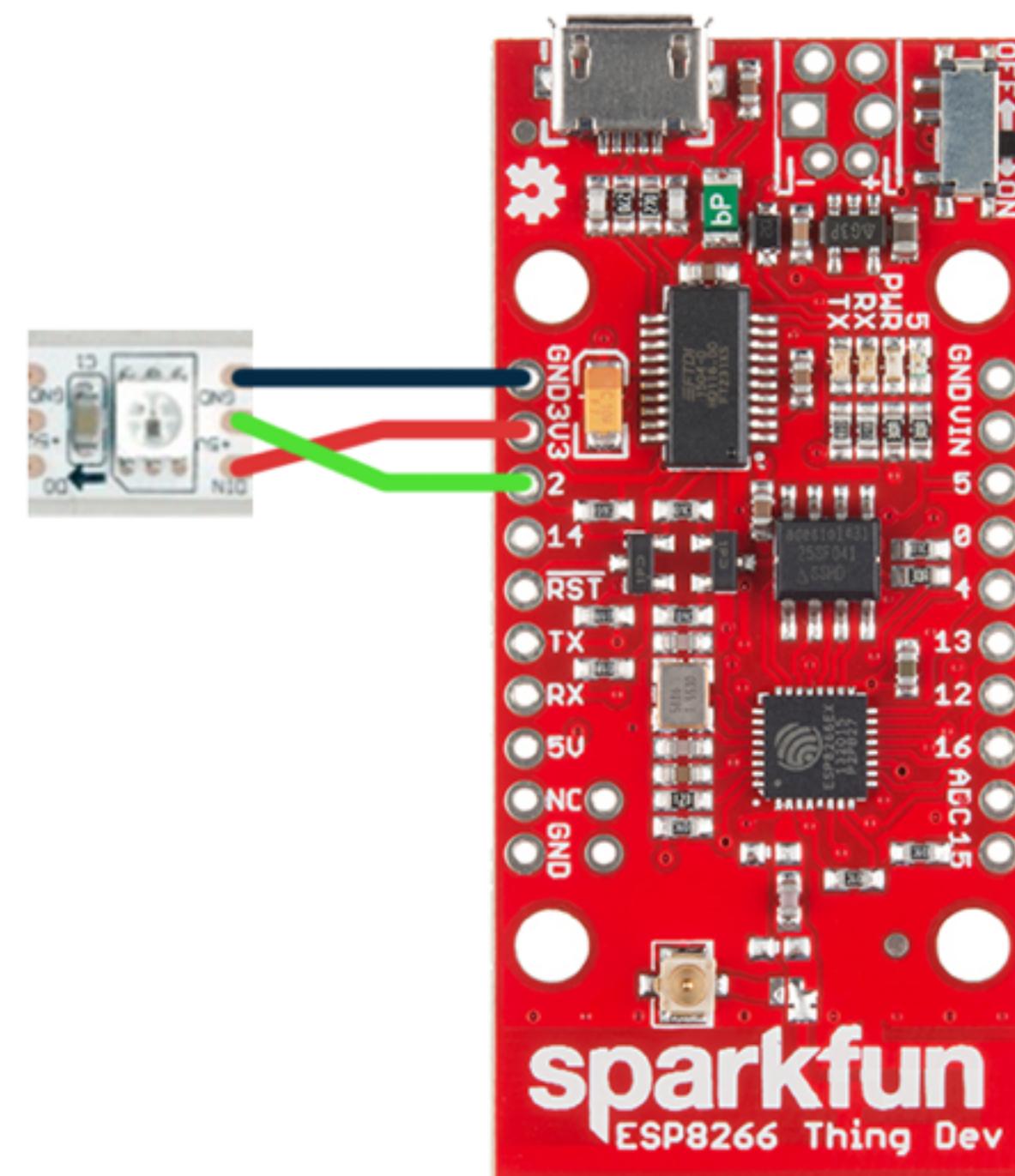


```
#01 // Librería Adafruit NeoPixel para controlar el LED
#02 #include <Adafruit_NeoPixel.h>
#03
#04 // Declaramos los pines que vamos a utilizar
#05 const int PIXEL_PIN = 2;
#06
#07 // Inicializamos la tira de LEDs
#08 Adafruit_NeoPixel strip = Adafruit_NeoPixel(1,
#09                                     PIXEL_PIN,
#10                                     NEO_GRB + NEO_KHZ800);
#11
#12 // Definimos los colores que vamos a utilizar
#13 const uint32_t rojo = strip.Color(0xFF, 0x00, 0x00);
#14 const uint32_t azul = strip.Color(0x00, 0xBF, 0xFF);
#15
#16 void setup() {
#17     initHardware();
#18 }
#19
```

```
#20 void loop() {  
#21   cycleColors();  
#22 }  
#23  
#24 void initHardware() {  
#25   strip.begin();  
#26   strip.setBrightness(255);  
#27   strip.show();  
#28 }  
#29  
#30 void cycleColors() {  
#31   int delayTime = 4000;  
#32  
#33   strip.setPixelColor(0, rojo);  
#34   strip.show();  
#35   delay(delayTime);  
#36  
#37   strip.setPixelColor(0, azul);  
#38   strip.show();
```

```
#39    delay(delayTime);  
#40 }  
#41
```

Guía de conexión



```
#01 // Librería Adafruit NeoPixel para controlar el LED
#02 #include <Adafruit_NeoPixel.h>
#03+#include <ESP8266WiFi.h>
#04
#05 // Declaramos los pines que vamos a utilizar
#06 const int PIXEL_PIN = 2;
#07+const int LED_PIN = 5;
#08
#09+// Configuración de la conexión
#10+const char WiFiSSID[] = "Pluva";
#11+const char WiFiPSK[] = "rainmaker3000";
#12+
#13 // Inicializamos la tira de LEDs
#14 Adafruit_NeoPixel strip = Adafruit_NeoPixel(1,
#15                                     PIXEL_PIN,
#16                                     NEO_GRB + NEO_KHZ800);
#17
#18 // Definimos los colores que vamos a utilizar
#19 const uint32_t rojo = strip.Color(0xFF, 0x00, 0x00);
```

```
#20 const uint32_t azul = strip.Color(0x00, 0xBF, 0xFF);
#21
#22 void setup() {
#23     initHardware();
#24+    connectWiFi();
#25 }
#26
#27 void loop() {
#28     cycleColors();
#29 }
#30
#31+void connectWiFi() {
#32+    byte ledStatus = LOW;
#33+
#34+    WiFi.mode(WIFI_STA);
#35+    WiFi.begin(WiFiSSID, WiFiPSK);
#36+
#37+    while (WiFi.status() != WL_CONNECTED) {
#38+        digitalWrite(LED_PIN, ledStatus);
```

```
#39+     ledStatus = (ledStatus == HIGH) ? LOW : HIGH;
#40+
#41+     delay(100);
#42+ }
#43+
#44+ digitalWrite(LED_PIN, LOW);
#45+
#46
#47 void cycleColors() {
#48     int delayTime = 4000;
#49
#50     strip.setPixelColor(0, rojo);
#51     strip.show();
#52     delay(delayTime);
#53
#54     strip.setPixelColor(0, azul);
#55     strip.show();
#56     delay(delayTime);
#57 }
```

```
#58
#59 void initHardware() {
#60+ pinMode(LED_PIN, OUTPUT);
#61+ digitalWrite(LED_PIN, HIGH);
#62+
#63 strip.begin();
#64 strip.setBrightness(255);
#65 strip.show();
#66 }
```



```
#01 // Librería Adafruit NeoPixel para controlar el LED
#02 #include <Adafruit_NeoPixel.h>
#03 #include <ESP8266WiFi.h>
#04
#05 // Declaramos los pines que vamos a utilizar
#06 const int PIXEL_PIN = 2;
#07 const int LED_PIN = 5;
#08
#09 // Configuración de la conexión
#10 const char WiFiSSID[] = "Pluva";
#11 const char WiFiPSK[] = "rainmaker3000";
#12
#13+// Variables para guardar la MAC
#14+String clientMac = "";
#15+uint8_t mac[WL_MAC_ADDR_LENGTH];
#16
#17 // Inicializamos la tira de LEDs
#18 Adafruit_NeoPixel strip = Adafruit_NeoPixel(1,
#19                                     PIXEL_PIN,
```

```
#20                                     NEO_GRB + NEO_KHZ800);  
#21  
#22 // Definimos los colores que vamos a utilizar  
#23 const uint32_t rojo = strip.Color(0xFF, 0x00, 0x00);  
#24 const uint32_t azul = strip.Color(0x00, 0xBF, 0xFF);  
#25  
#26 void setup() {  
#27     initHardware();  
#28     connectWiFi();  
#29 }  
#30  
#31 void loop() {  
#32     cycleColors();  
#33 }  
#34  
#35 void connectWiFi() {  
#36     byte ledStatus = LOW;  
#37  
#38     WiFi.mode(WIFI_STA);
```

```
#39 WiFi.begin(WiFiSSID, WiFiPSK);
#40
#41 while (WiFi.status() != WL_CONNECTED) {
#42   digitalWrite(LED_PIN, ledStatus);
#43   ledStatus = (ledStatus == HIGH) ? LOW : HIGH;
#44   delay(100);
#45 }
#46
#47
#48+ // Guardamos la MAC del ESP en un String
#49+ WiFi.macAddress(mac);
#50+ for (int i = 0; i < WL_MAC_ADDR_LENGTH; ++i) {
#51+   clientMac += String(mac[i], 16);
#52+ }
#53
#54   digitalWrite(LED_PIN, LOW);
#55 }
#56
#57 void cycleColors() {
```

```
#58     int delayTime = 4000;
#59
#60     strip.setPixelColor(0, rojo);
#61     strip.show();
#62     delay(delayTime);
#63
#64     strip.setPixelColor(0, azul);
#65     strip.show();
#66     delay(delayTime);
#67 }
#68
#69 void initHardware() {
#70     pinMode(LED_PIN, OUTPUT);
#71     digitalWrite(LED_PIN, HIGH);
#71
#73     strip.begin();
#74     strip.setBrightness(255);
#75     strip.show();
#76 }
```



```
#01 // Librería Adafruit NeoPixel para controlar el LED
#02 #include <Adafruit_NeoPixel.h>
#03 #include <ESP8266WiFi.h>
#04
#05 // Declaramos los pines que vamos a utilizar
#06 const int PIXEL_PIN = 2;
#07 const int LED_PIN = 5;
#08
#09 // Configuración de la conexión
#10 const char WiFiSSID[] = "Pluva";
#11 const char WiFiPSK[] = "rainmaker3000";
#12
#13 // Variables para guardar la MAC
#14 String clientMac = "";
#15 uint8_t mac[WL_MAC_ADDR_LENGTH];
#16
#17+// Datos de conexión con el servidor
#18+const char host[] = "pluva.tecnologica.com";
#19+const int port = 80;
```

```
#20+// Tag donde vienen los datos del tiempo
#21+const String code = "</forecast>";
#22+
#23+// Tiempo en ms entre cada peticion
#24+const unsigned long sleepTime = 60000;
#25+
#26 // Inicializamos la tira de LEDs
#27 Adafruit_NeoPixel strip = Adafruit_NeoPixel(1,
#28                                     PIXEL_PIN,
#29                                     NEO_GRB + NEO_KHZ800);
#21
#22 // Definimos los colores que vamos a utilizar
#23 const uint32_t rojo = strip.Color(0xFF, 0x00, 0x00);
#24 const uint32_t azul = strip.Color(0x00, 0xBF, 0xFF);
#25
#26 void setup() {
#27     initHardware();
#28     connectWiFi();
#29 }
```

```
#30+    cycleColors();  
#31 }  
#32  
#33 void loop() {  
#34+  WiFiClient client;  
#35+  
#36+  if (!client.connect(host, port)) {  
#37+    return;  
#38+  }  
#39+  
#40+  String url = "/forecast.php?u=" + clientMac;  
#41+  client.print(String("GET ") + url + " HTTP/1.1\r\n" +  
#42+    "Host: " + host + "\r\n" +  
#43+    "Connection: close\r\n\r\n");  
#44+  
#45+  delay(500);  
#36+  
#37+  while(client.available()){  
#38+    String line = client.readStringUntil('\r');
```

```
#49+
#50+    // Comprobamos si la línea actual termina con el tag </forecast>
#51+    if (line.endsWith(code)) {
#52+        // Sacamos el valor que viene entre <code></code>
#53+        String setColor = line.substring(code.length(), line.indexOf(code));
#54+
#55+        // Ponemos el pixel del color que corresponda
#56+        setPixelToWeather(setColor.toInt());
#57+    }
#58+}
#59+delay(sleepTime);
#60+}
#61
#62void connectWiFi() {
#63    byte ledStatus = LOW;
#64
#65    WiFi.mode(WIFI_STA);
#66    WiFi.begin(WiFiSSID, WiFiPSK);
#67
```

```
#68  while (WiFi.status() != WL_CONNECTED) {
#69    digitalWrite(LED_PIN, ledStatus);
#70    ledStatus = (ledStatus == HIGH) ? LOW : HIGH;
#71    delay(100);
#72  }
#73
#74
#75 // Guardamos la MAC del ESP en un String
#76 WiFi.macAddress(mac);
#77 for (int i = 0; i < WL_MAC_ADDR_LENGTH; ++i) {
#78   clientMac += String(mac[i], 16);
#79 }
#80
#81 digitalWrite(LED_PIN, LOW);
#82 }
#83
#84 void cycleColors() {
#85   int delayTime = 4000;
#86 }
```

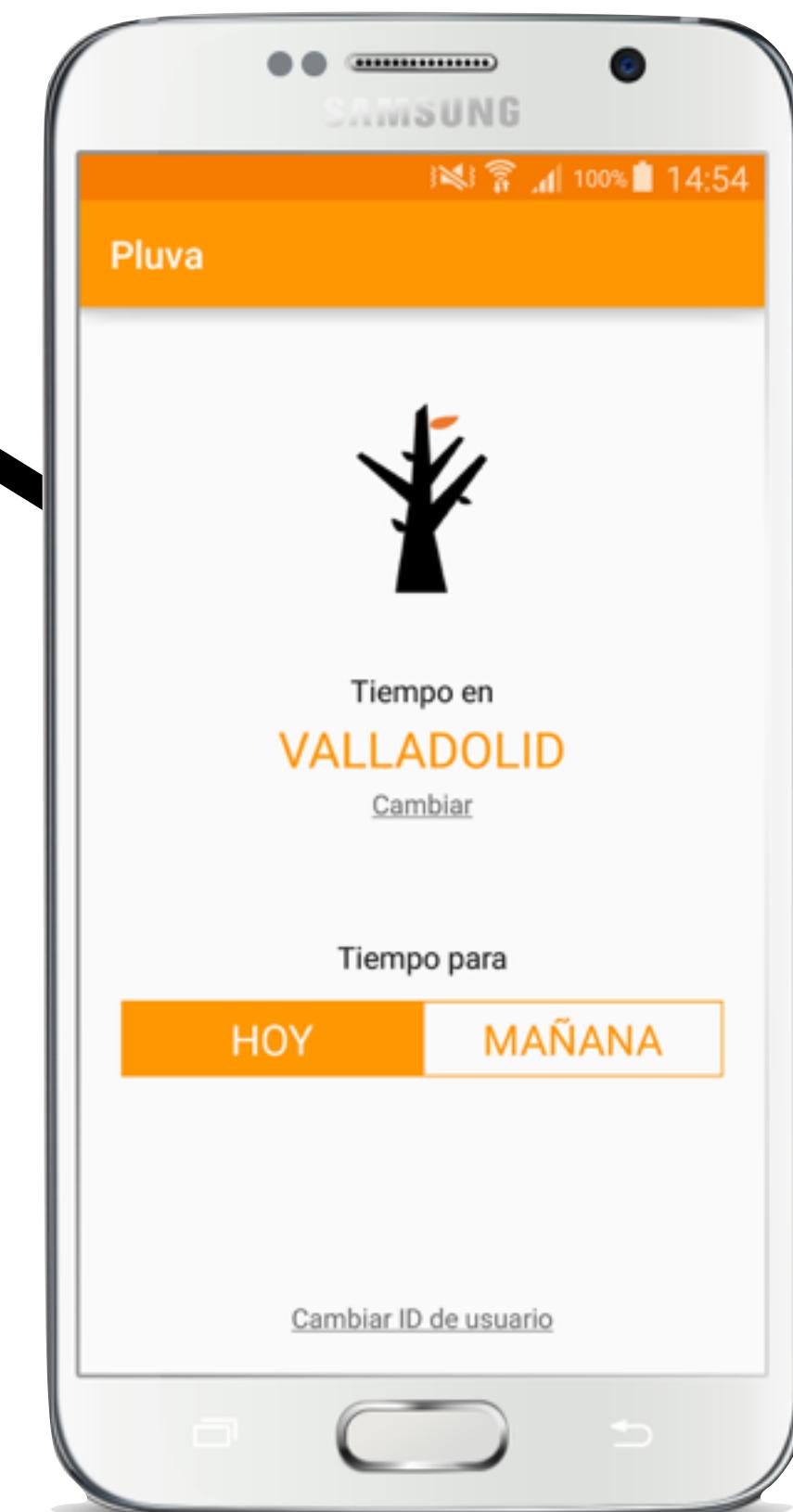
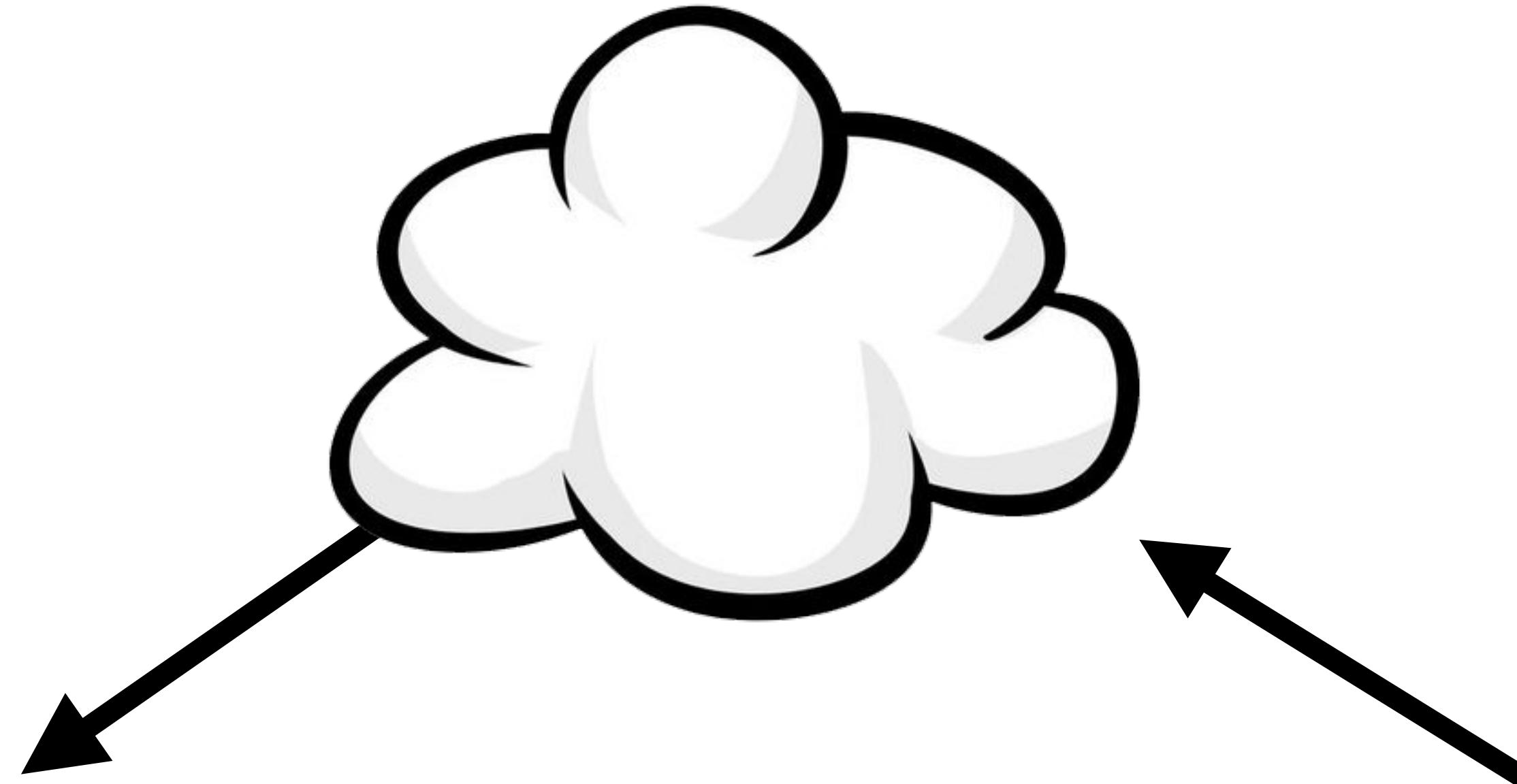
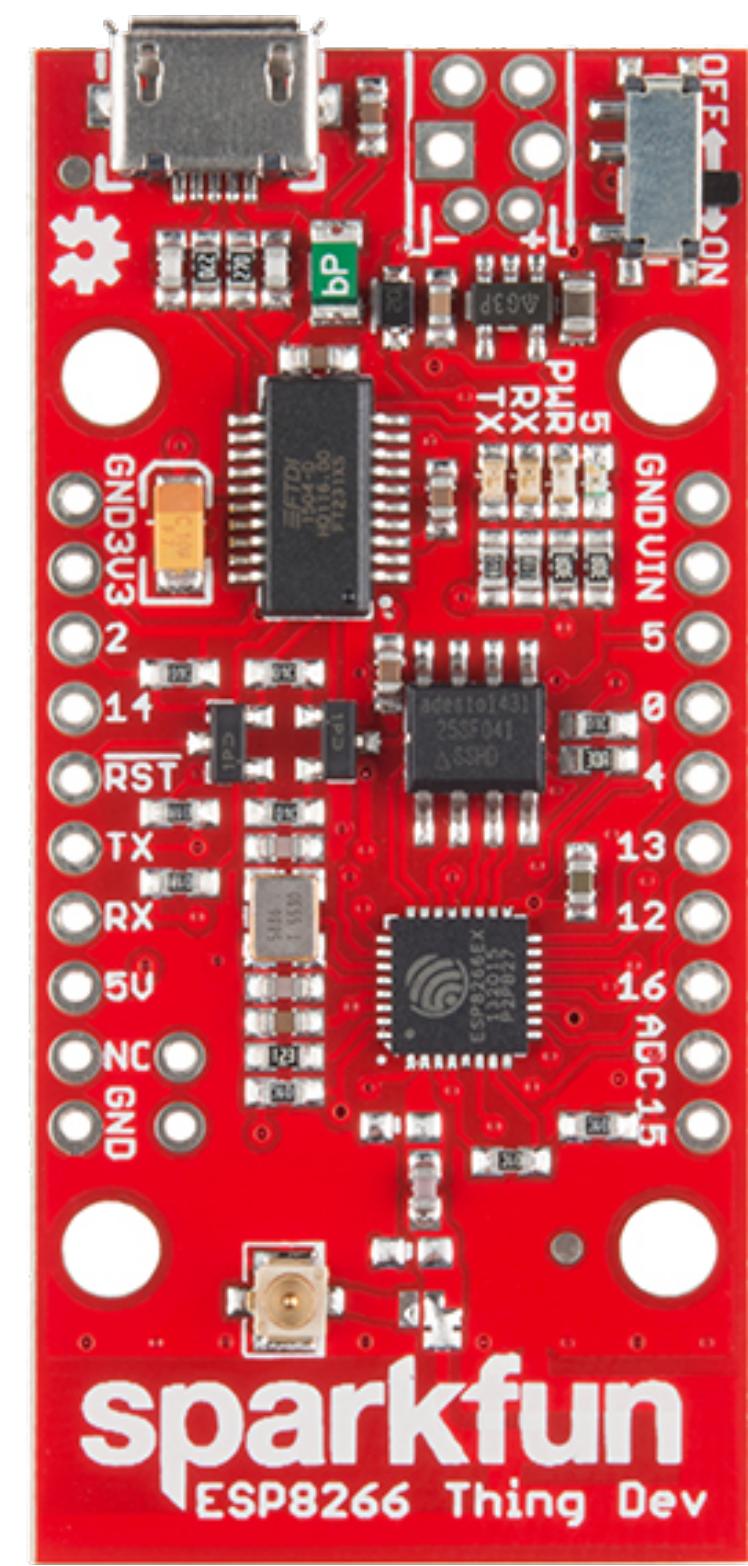
```
#087     strip.setPixelColor(0, rojo);
#088     strip.show();
#089     delay(delayTime);
#090
#091     strip.setPixelColor(0, azul);
#092     strip.show();
#093     delay(delayTime);
#094 }
#095
#096 void initHardware() {
#097     pinMode(LED_PIN, OUTPUT);
#098     digitalWrite(LED_PIN, HIGH);
#099
#100    strip.begin();
#101    strip.setBrightness(255);
#102    strip.show();
#103 }
#104
#105// Cambia el color del pixel dependiendo del clima recibido
```

```
#106+void setPixelToWeather(int weather) {  
#107+    switch (weather) {  
#108+        case 1:  
#109+        case 2:  
#110+        case 3:  
#111+            strip.setPixelColor(0, rojo);  
#112+            break;  
#113+        case 4:  
#114+        case 5:  
#115+            strip.setPixelColor(0, azul);  
#116+            break;  
#117+    }  
#118+  
#119+    strip.show();  
#120+}
```



 tecnológica

Configurador: App Android



Configurador

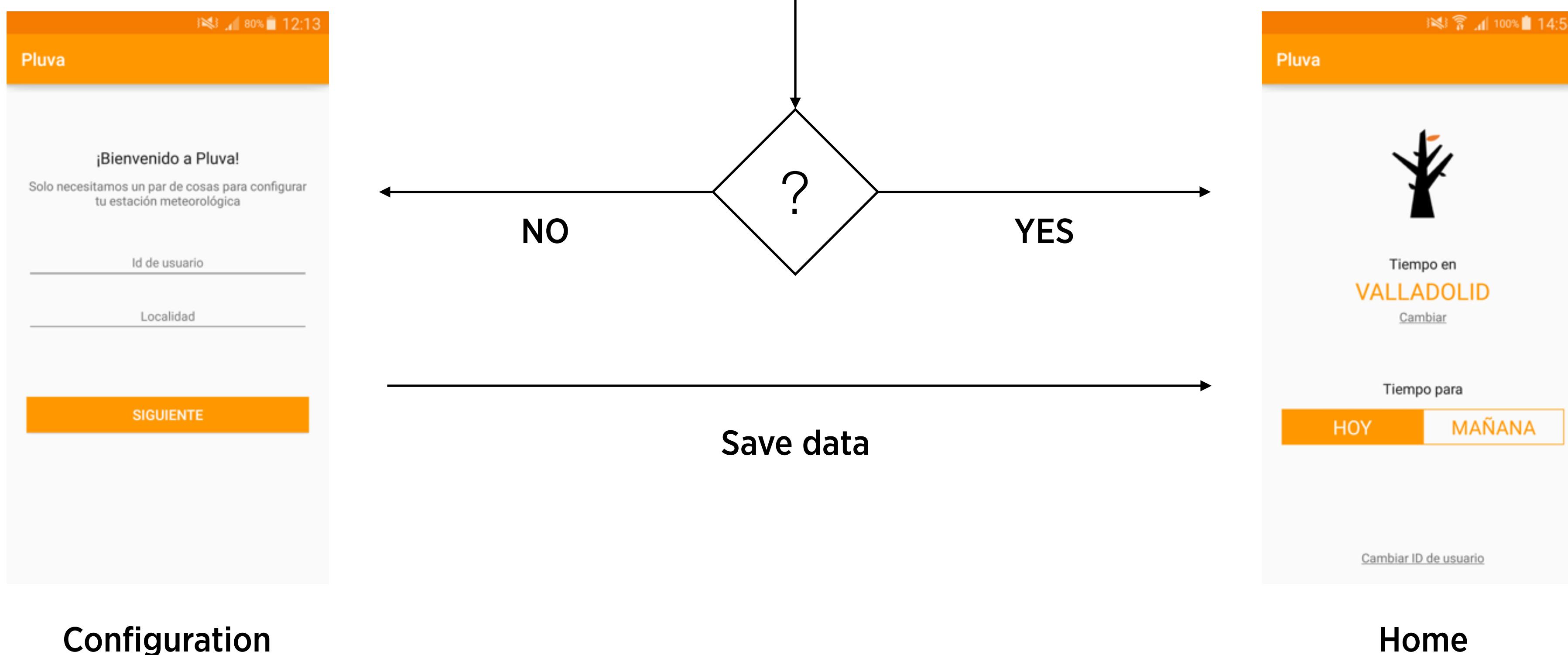
Implementación en tres etapas

1. Estructura y front (v2.0)
2. Guardando en preferencias (v2.1)
3. Con llamadas al API (v2.2)

<https://github.com/tecnilogica/pluva/tags>

Estructura de la app

Main



Preferencias

- SharedPreferences: Pares clave-valor
- Datos en el dispositivo

```
public static String PREF_USER_ID = "PREF_USER_ID";
public static String PREF_LOCATION = "PREF_LOCATION";
public static String PREF_DAY = "PREF_DAY";
```

Llamando al API

- Retrofit vs Volley
- Interface con Llamadas + Clase ApiCall
- 2 Llamadas: hoy y mañana
- Retorno comprobando localización

¡Muchas gracias!

angel.alonso@tecnilogica.com
lucia.gomez@tecnilogica.com