

Universidad Tecnológica Nacional
Facultad Regional Tucumán

Asignatura:

Técnicas Digitales II

Proyecto:

*Implementación de red IoT para monitoreo de
temperatura de piscina hogareña*

Alumnos: Esperanza Facundo.
Fernández Nahuel.

Profesor: Ing. Mansilla, Rubén Darío.

INTRODUCCIÓN

Este proyecto surge de la necesidad de sensar la temperatura de una piscina y obtener datos para de esta manera tener un control y realizar un monitoreo de la misma con cualquier celular o computadora que cuente con conexión a una red Wifi.

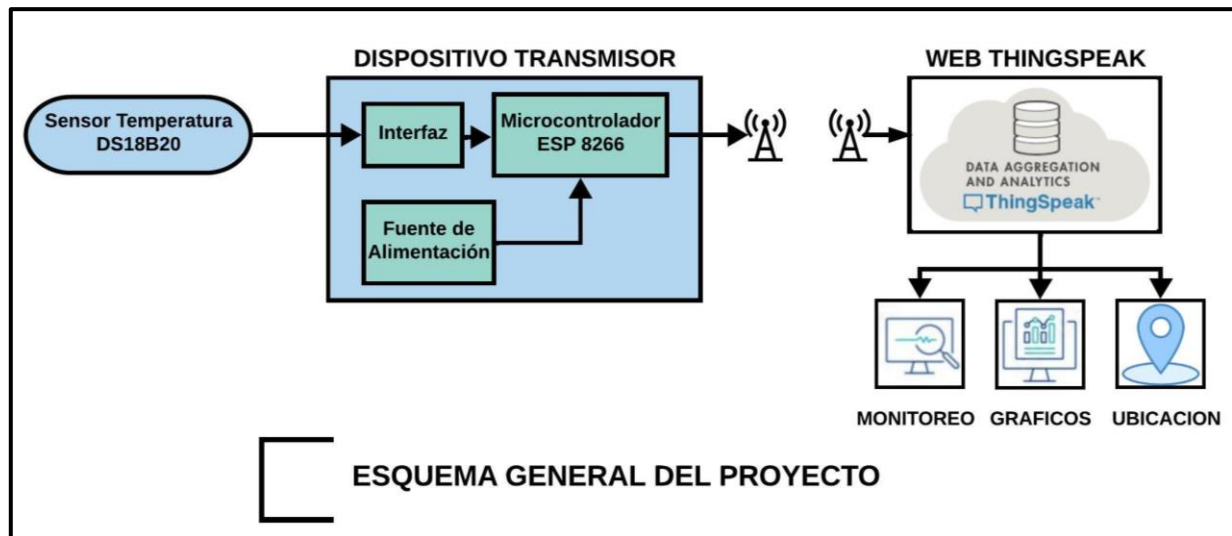
OBJETIVO DEL PROYECTO

La idea de este proyecto es obtener datos de la temperatura del agua de una piscina hogareña, donde para lograr tal fin y haciendo uso de lo aprendido en la asignatura, será necesario que implementemos una red IoT (Internet of Things – Internet de las cosas).

Para ello vamos a utilizar el microcontrolador ESP8266 que se encargará de dotar de conexión Wifi a nuestro proyecto, conjuntamente con el sensor de temperatura digital DS18B20 los cuales estarán integrados en una placa Protoboard, todo esto ubicado dentro de un gabinete adecuado.

Ya que la finalidad es monitorear los datos medidos por el sensor, haciendo uso de la plataforma ThingSpeak, implementaremos una base de datos donde se guardarán las lecturas de temperatura del agua para luego ser visualizadas usando los "fields" de la plataforma y también generando gráficas de la temperatura con respecto al tiempo, que nos permita ver la evolución de la misma en un determinado intervalo de tiempo.

DIAGRAMA ESQUEMATICO DEL PROYECTO



ELEMENTOS UTILIZADOS

Para la implementación de nuestro proyecto fueron necesarios los siguientes elementos:

- Protoboard 80 puntos
- Regulador 3.3 V
- Resistencia 4.7 K Ω
- Microcontrolador ESP8266
- Sensor de Temperatura DS18B20
- Capacitores cerámicos 100nF
- Gabinete estanco 10x10
- Baterías 18650 x 2
- Interruptor

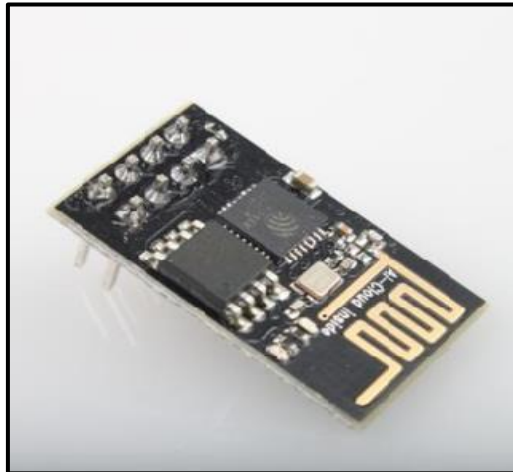
Entre los elementos utilizados más importantes tenemos:

Microcontrolador ESP8266

Como módulo de comunicación que nos permita conformar la red IoT y establecer la comunicación con la nube (Thingspeak) se eligió al módulo wifi ESP-01s desarrollado por la compañía Ai-Thinker. Este módulo está basado en el microcontrolador tipo SoC ESP8266, el cual es un microcontrolador de ultra baja potencia de 32 bits.

Este módulo brinda a nuestro proyecto una opción económica y completa que hace posible realizar todas las funciones y procesamiento relacionados con el envío de datos y conexión a la nube al procesador integrado en el módulo.

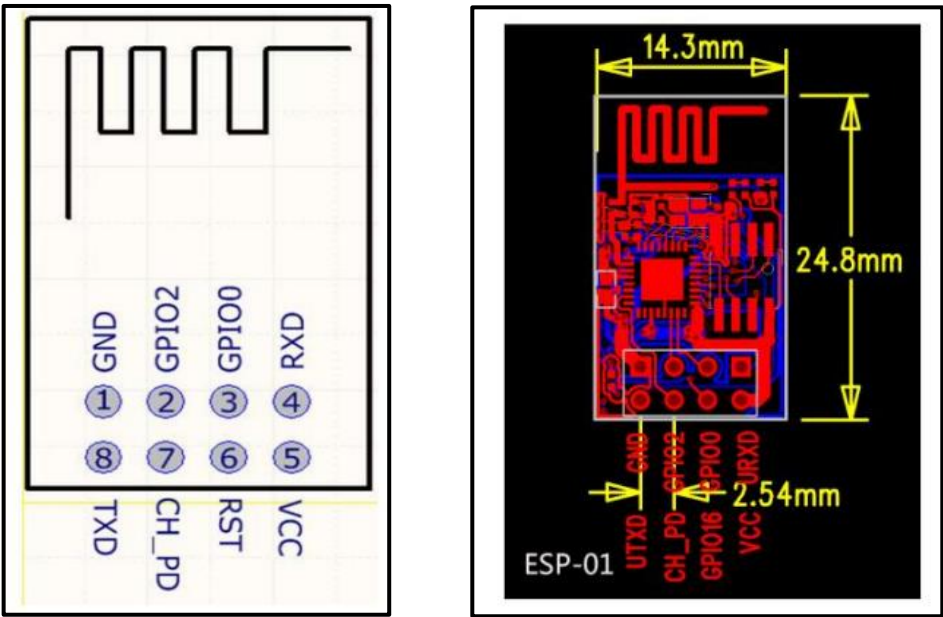
Físicamente, este módulo presenta un tamaño bastante reducido (14,3mm x 24,8mm) en comparación a otros módulos de la misma compañía, necesitando así un mínimo requerimiento de espacio para poder ser empleado. Posee una antena on-board y ocho pines de conexión.



Especificaciones Técnicas

- Modelo: ESP-01.
- SoC: ESP8266.
- Voltaje de Alimentación: 3.3V DC.
- Voltaje de Entradas/Salidas: 3.3V DC.
- CPU: Tensilica Xtensa LX3 (32 bit).
- Frecuencia de Reloj: 80MHz/160MHz.
- Instruction RAM: 32KB.
- Data RAM: 96KB.
- Memoria Flash Externa: 4MB.
- Pines Digitales GPIO: 4.
- Protocolo: 802.11 b/g/n.
- Wi-Fi Direct (P2P), soft-AP.
- Stack de Protocolo TCP/IP integrado.
- PLLs, reguladores, DCXO y manejo de poder integrados.
- Potencia de salida de +19.5dBm en modo 802.11b.
- Corriente de fuga menor a 10uA.
- El procesador integrado de 32-bit puede ser utilizado como procesador de aplicación
- Buses de comunicación: SDIO 1.1/2.0, SPI, UART.
- STBC, 1x1 MIMO, 2x1 MIMO
- Consumo de potencia Standby < 1.0mW.

Pines y dimensiones



Descripción de los pines

NO.	Pin Name	Function
1	GND	GND
2	GPIO2	GPIO,Internal Pull-up
3	GPIO0	GPIO,Internal Pull-up
4	RXD	UART0,data received pin RXD
5	VCC	3.3V power supply (VDD)
6	RST	1) External reset pin, active low 2) Can loft or external MCU
7	CH_PD	Chip enable pin. Active high
8	TXD	UART0,data send pin RXD

Sensor de Temperatura DS18B20

El sensor digital de Temperatura **DS18B20** incluye internamente toda la electrónica necesaria: sensado, acondicionamiento, conversión analógica a digital e interfaz digital. Permite conocer la temperatura utilizando solo 1 cable para datos (1-Wire). Permite utilizar simultáneamente varios sensores en el mismo bus ya que cada sensor tiene un identificador único de fábrica.

Entre sus aplicaciones más comunes está el sensado de temperatura en ambientes de edificios, aire acondicionado, maquinaria, control y monitoreo de procesos industriales.



Especificaciones técnicas

- Voltaje de operación: 3.0V – 5.5V DC
- Rango de medición: -55°C hasta +125°C (-67°F a +257°F)
- Precisión en el rango de -10°C hasta +85°C: $\pm 0.5^{\circ}\text{C}$.
- Resolución ADC seleccionable de 9-12 bits
- Cables: Rojo (+VCC), Blanco (DATA 1-Wire), Negro (GND)
- Protocolo 1-Wire, solo necesita 1 pin para comunicarse
- Identificación única de 64 bits
- Cubierta de acero inoxidable de alta calidad, previene la oxidación de la sonda
- Sonda a prueba de agua
- Longitud de cable: 1m
- Dimensiones sonda: D5mm*L50mm
- Peso: 23 gramos

FUNCIONAMIENTO DEL SISTEMA DESARROLLADO

El funcionamiento del sistema desarrollado es el siguiente:

Al apretar la tecla de encendido, situada en la parte superior del mismo, el dispositivo inicia el proceso de conexión a la red Wifi previamente especificada en el código fuente (en nuestro caso la de nuestro celular), una vez que logra cumplir este objetivo y de manera automática, inicia el proceso de medición de la temperatura del agua.

El encargado de llevar a cabo el mismo, es el sensor digital de temperatura DS18B20, el cual ya sumergido en nuestra piscina y orden mediante de nuestro microcontrolador ESP8266 comienza a recolectar las mediciones y a enviarlas al ESP8266 haciendo uso del protocolo de comunicación One-Wire.

Una vez que nuestro microcontrolador recibe las lecturas, se encarga de almacenarlas en la plataforma ThingSpeak, en la cual las mismas se guardan en una base de datos. Esta plataforma también nos brinda facilidades como la de visualizar en tiempo real los datos recogidos por nuestro dispositivo, visualizar en un mapa la ubicación exacta desde donde están siendo enviadas las lecturas, así como también gráficos de la temperatura en función del tiempo y realizar el procesamiento de nuestros datos para generar distintos tipos de graficos.

El código fuente de nuestro proyecto es el siguiente:

CODIGO FUENTE

```
// Inicialización de las librerías
#include<ESP8266WiFi.h>
#include<OneWire.h>
#include<DallasTemperature.h>

// Definición de la contraseña de API desde ThingSpeak
StringapiKey = "Q2K6JYS9W5JRDk57";

// Definición de los datos de la red wifi a utilizar
constchar *ssid = "Redmi Note 13 Pro 5G";
constchar *pass = "1234567890";
constchar* server = "api.thingspeak.com";

#defineONE_WIRE_BUS2

OneWireoneWire(ONE_WIRE_BUS);

DallasTemperatureensors(&oneWire);

WiFiClientclient;
```

```

// Inicio del programa principal
void setup()
{
  Serial.begin(115200);
  delay(10);

  // Inicicion de la conexión a la red wifi
  WiFi.begin(ssid, pass);

  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
  }

}

// Lazo de repetición
void loop()
{

  // Inicio del proceso de medición
  sensors.requestTemperatures();
  float tempC = sensors.getTempCByIndex(0);

  // Subida de los datos a la nube de ThingSpeak
  if(client.connect(server,80))
  {
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(tempC);

    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

  }
  client.stop();

  // Intervalo entre mediciones
  delay(15000);
}

```

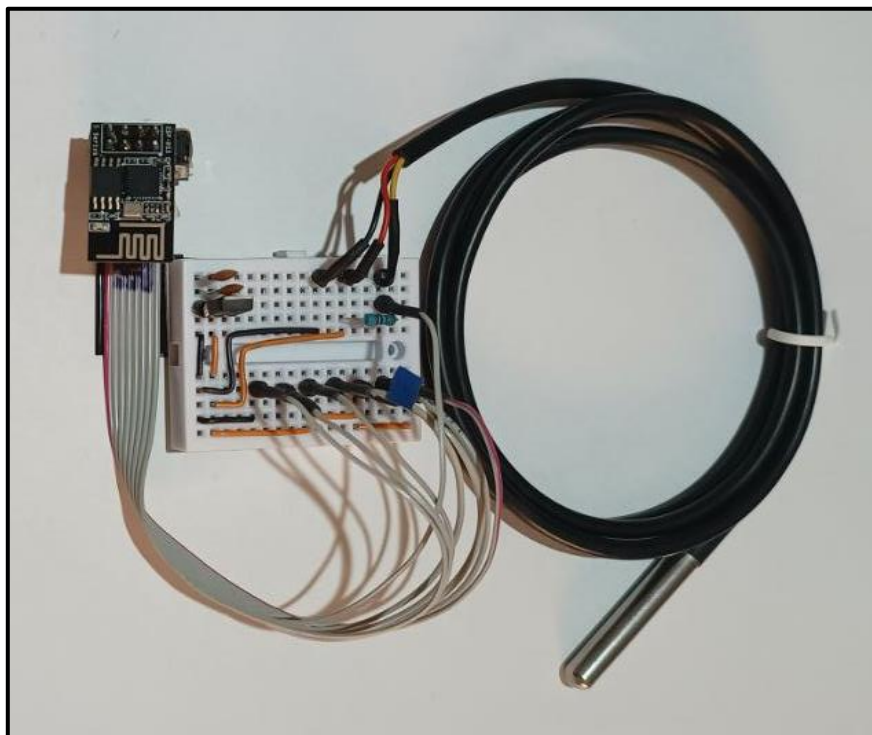

Descripción del código fuente:

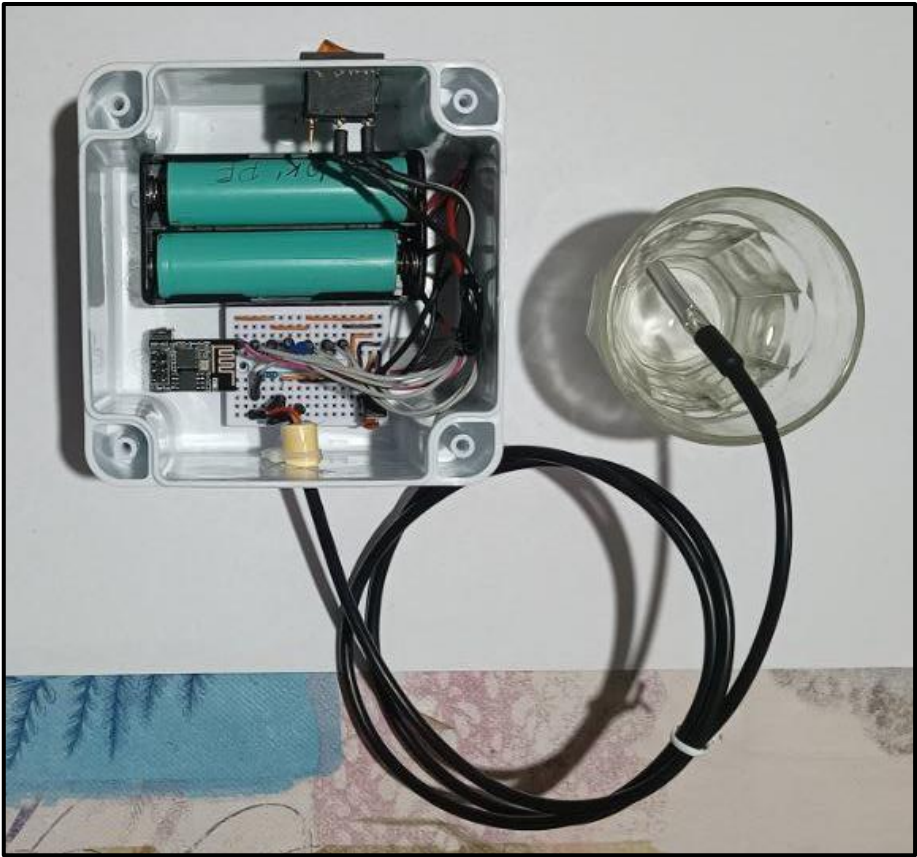
El programa comienza con la inicialización de las librerías necesarias que vamos a ocupar en nuestro código, luego se definen variables de suma importancia como ser "StringApiKey" que nos permite el acceso a la cuenta previamente generada en la plataforma ThingSpeak, los datos de la red Wifi, que en nuestro caso es la de nuestro celular y el pin de comunicación "OneWire" para el sensor.

A continuación, inicia el proceso de conexión a la red wifi, es necesario que este paso se complete para poder avanzar al siguiente punto. Si la conexión a la red wifi fue exitosa, el microcontrolador solicita los datos sensados, una vez que están disponibles, comienza la parte más importante del funcionamiento, con la carga de los datos sensados a la base de datos de la plataforma ThingSpeak y por último se genera un "delay" que corresponde al tiempo entre lecturas que realiza el sensor.

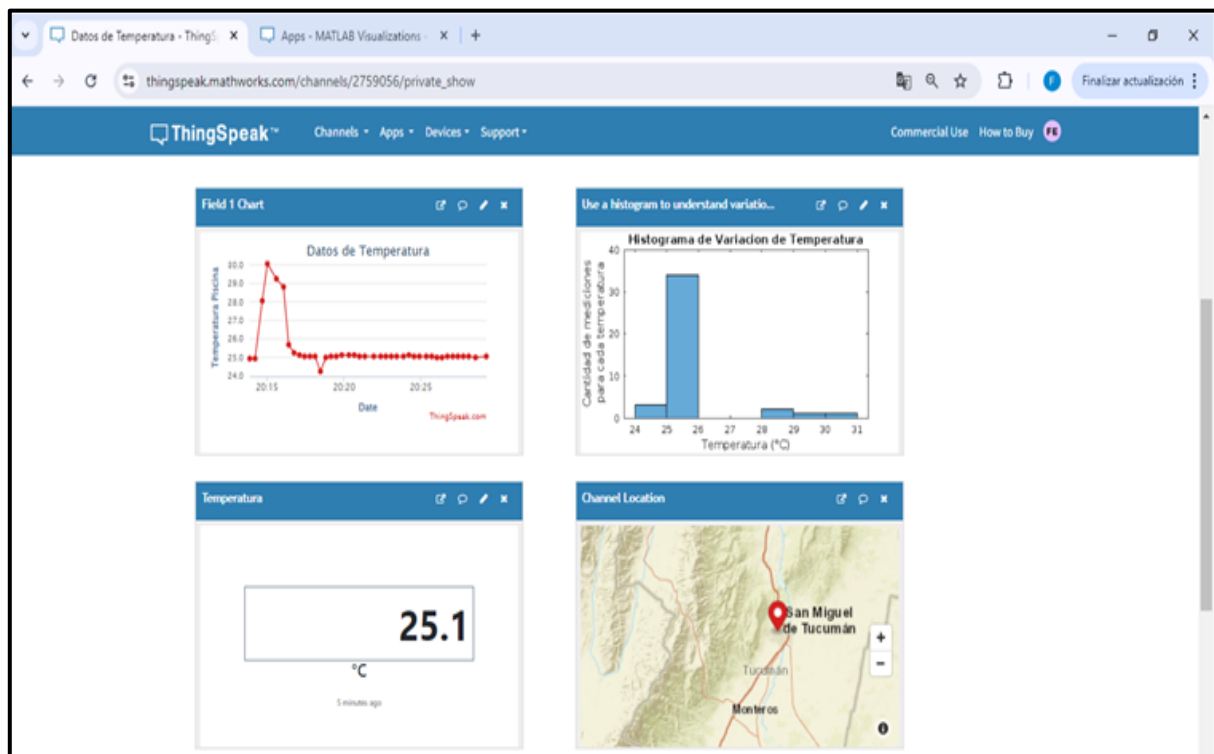
DISPOSITIVO TERMINADO

A continuación, podemos ver diferentes vistas del dispositivo terminado tales como el circuito ya montado en la protoboard y luego cuando ya está ubicado dentro de la caja estanco que lo protege de la intemperie, donde podemos ver las diferentes partes que lo conforman.





Entorno ThingSpeak



En la imagen anterior podemos observar el entorno de la plataforma ThingSpeak, donde se muestran cuatro recuadros los cuales se denominan campos. Los mismos son seleccionados por nosotros ya que consideramos que para la aplicación del proyecto son necesarios.

El cuadro que contiene la gráfica número 1 (vértice superior izquierdo) nos muestra los datos de temperatura en función del tiempo, estas muestras son tomadas en intervalos programados previamente, por razones prácticas los mismos serán tomados cada 15 segundos aproximadamente, cabe aclarar que durante las mediciones fuimos variando la temperatura a propósito para demostrar la funcionalidad del dispositivo, esto se logró sumergiendo el sensor en agua a temperatura ambiente y con hielo, es por ello que en la gráfica aparecen saltos de temperatura amplios.

La segunda gráfica (vértice superior derecho) nos muestra un histograma generado mediante los datos tomados de la gráfica uno, que nos informa la cantidad de muestras tomadas por el dispositivo que se encuentran en un determinado rango de temperaturas. El mismo está desarrollado mediante un código de Matlab que se muestra a continuación.

MATLAB Code

```

1 % Read temperature for the last 10 hours from a ThingSpeak channel and
2 % visualize temperature variations using the MATLAB HISTOGRAM function.
3
4 % Channel 12397 contains data from the MathWorks Weather Station, located
5 % in Natick, Massachusetts. The data is collected once every minute. Field
6 % 4 contains temperature data.
7 % Channel ID to read data from
8 readChannelID = 2759056;
9
10 % Temperature Field ID
11 TemperatureFieldID = 1;
12
13 % Channel Read API Key
14 % If your channel is private, then enter the read API
15 % Key between the '' below:
16 readAPIKey = 'KU04EDF75BN81DOC';
17
18 % Get temperature data from field 4 for the last 10 hours = 10 x 60
19 % minutes. Learn more about the THINGSPEAKREAD function by going to
20 % the Documentation tab on the right side pane of this page.
21
22 tempC = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID,...
23 'NumMinutes',10*60, 'ReadKey',readAPIKey);
24
25 histogram(tempC);
26 xlabel('Temperatura (°C)');
27 ylabel('Cantidad de mediciones\nnewline para cada temperatura');
28 title('Histograma de Variacion de Temperatura');

```

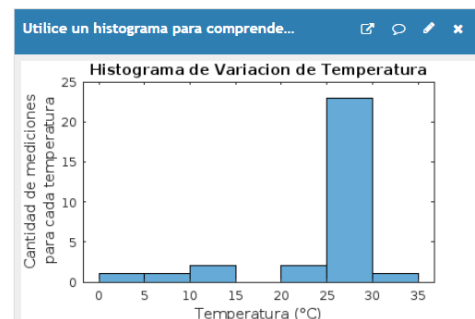
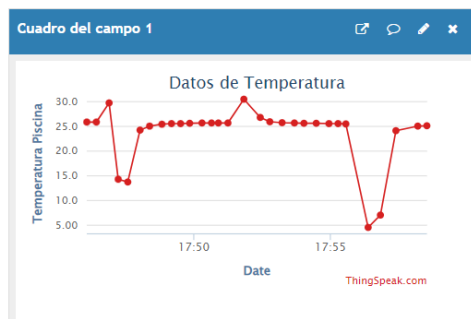
La tercera gráfica (vértice inferior izquierdo) nos muestra el último dato de temperatura sensado. Por último, en la cuarta posición (vértice inferior derecho) y no menos importante observamos la ubicación de la piscina en el mapa de la ciudad de Tucumán, la cual podemos seleccionar en el entorno de ThingSpeak. Otro ejemplo de datos sensados:

Estadísticas del canal

Creado: hace 17 días

Última entrada: Hace aproximadamente una hora

Entradas: 30



Enlace a GitHub

<https://github.com/facundo36121/Proyecto-Digitales-Esperanza-Fern-ndez>