



**Universidad autónoma de baja california**

**Ingeniería en computación**

**Internet de las cosas**

**Lab-taller: ThingSpeak**

**Aguilar Noriega Leocundo**

**Garcia Chaves erik**

**20 de febrero del 2026**

## Descripción:

Crear un programa que suba datos en a la nube de **ThingSpeak** se deben de crear 2 canales dentro de la plataforma en donde por medio del ESP y por medio de su API el ESP32 pueda subir 2 datos a 2 labels de la plataforma que se estarán monitoreando,

## Desarrollo:

Para la poder subir datos a la plataforma **ThingSpeak** es necesario es crear una cuenta de **MathLab** y una vez iniciado, en el menú de arriba nos aparecerán varias opciones, nosotros debemos elegir la opción de **channels**, creamos nuestro canal

Cuando este se cree, nos dará varias llaves, pero por ahora solo nos interesa 1 llave, la cual es la que usaremos para escribir dentro de la nube, estas se encontraran dentro de la pestania **API KEYS**

Mediante el protocolo HTTP y la petición **GET** podremos escribir con la API a la nube de ThingSpeak

Código que envía los datos.

```
// envio de datos
void thingspeak_send(float field1, float field2) {
    char url[256];
    sprintf(url, sizeof(url),
            "http://api.thingspeak.com/update?api_key=%s&field1=%.*f&field2=%.*f",
            TS_API_KEY, field1, field2);

    esp_http_client_config_t config = {
        .url = url,
        .method = HTTP_METHOD_GET,
    };

    esp_http_client_handle_t client = esp_http_client_init(&config);
    esp_err_t err = esp_http_client_perform(client);

    if (err == ESP_OK) {
        int status = esp_http_client_get_status_code(client);
        ESP_LOGI(TAG, "HTTP Status: %d | field1: %.2f | field2: %.2f", status, field1, field2);
    } else {
        ESP_LOGE(TAG, "Error en request: %s", esp_err_to_name(err));
    }

    esp_http_client_cleanup(client);
}
```

Dentro de la plataforma vamos a poder ver nuestras 2 tablas con los 3 números randoms que se están mandando cada minuto:

```
I (3265) wifi:dp: 1, bi: 102400, li: 3, scale listen interval from 307200 us to 307200 us
I (3285) wifi:AP's beacon interval = 102400 us, DTIM period = 1
I (6485) esp_netif_handlers: sta ip: 192.168.1.93, mask: 255.255.255.0, gw: 192.168.1.254
I (6485) ThingSpeak: IP obtenida: 192.168.1.93
I (6485) ThingSpeak: WiFi conectado exitosamente
I (6485) ThingSpeak: Enviando -> field1: 1.73 | field2: 0.00
I (6885) ThingSpeak: HTTP Status: 200 | field1: 54.00 | field2: 37.00
I (25925) wifi:<ba-add>idx:0 (ifx:0, d8:6d:17:90:8f:48), tid:0, ssn:5, winSize:64
```

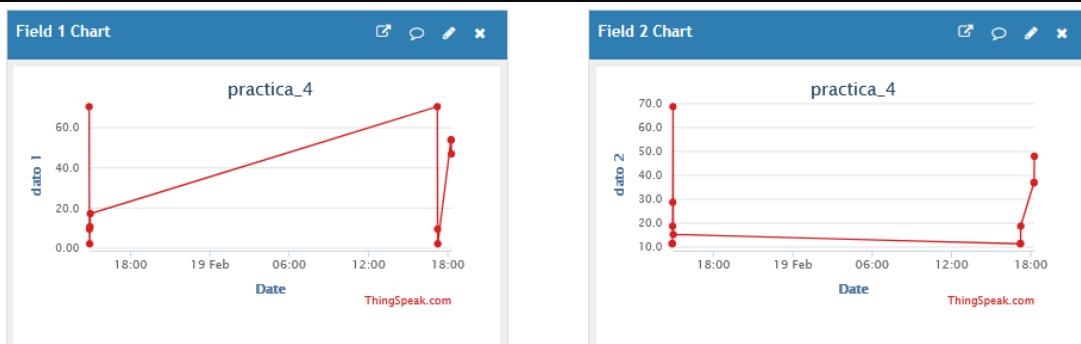
Vemos como se están actualizando los datos dentro de la plataforma

### Channel Stats

Created: [a day ago](#)  
Last entry: [about a minute ago](#)  
Entries: 11



```
I (6485) ThingSpeak: WiFi conectado exitosamente
I (6485) ThingSpeak: Enviando -> field1: 1.73 | field2: 0.00
I (6885) ThingSpeak: HTTP Status: 200 | field1: 54.00 | field2: 37.00
I (25925) wifi:<ba-add>idx:0 (ifx:0, d8:6d:17:90:8f:48), tid:0, ssn:5, winSize:64
I (66885) ThingSpeak: Enviando -> field1: 1.73 | field2: 0.00
I (67305) ThingSpeak: HTTP Status: 200 | field1: 47.00 | field2: 48.00
```



## **Conclusión:**

En la práctica se pudo ver un ejemplo de como poder monitorear datos, en este caso solo se están enviando datos randoms, pero es un ejemplo de algo que puede ser real, para la implementación de un proyecto real, como el monitorear la humedad y presentarlo en gráficos, para muy posiblemente un análisis de datos o para entrenamiento de algún agente de IA, las posibilidades son muy amplias el solo mandar datos a una nube, por lo que lo aprendido fue mucho y pudimos poner en prácticas los conocimientos de protocolos HTTP/HTTPS