

CONFIGURACIÓN DE LA PLATAFORMA IoT BLYNK

I- INTRODUCCIÓN

Las plataformas y frameworks IoT, nos permiten crear proyectos IoT de forma rápida y sencilla a través de teléfonos inteligentes (smartphones) con diferentes sistemas operativos (iOS/Android) y en algunos casos también controlar desde el navegador. Algunas de las opciones disponibles en la actualidad son:



Fig. 1 - Ejemplos de plataformas y frameworks IOT

BLYNK

La plataforma IoT Blynk permite:

- Monitoreo y control de dispositivos en una aplicación o en la web
- Configurar notificaciones
- Actualizar el firmware del dispositivo por aire (OTA)
- Proporcionar credenciales de Wi-Fi a dispositivos
- Compartir dispositivos con otros usuarios
- Administrar roles y permisos
- Gestionar organizaciones

Blynk consta de cuatro componentes principales:

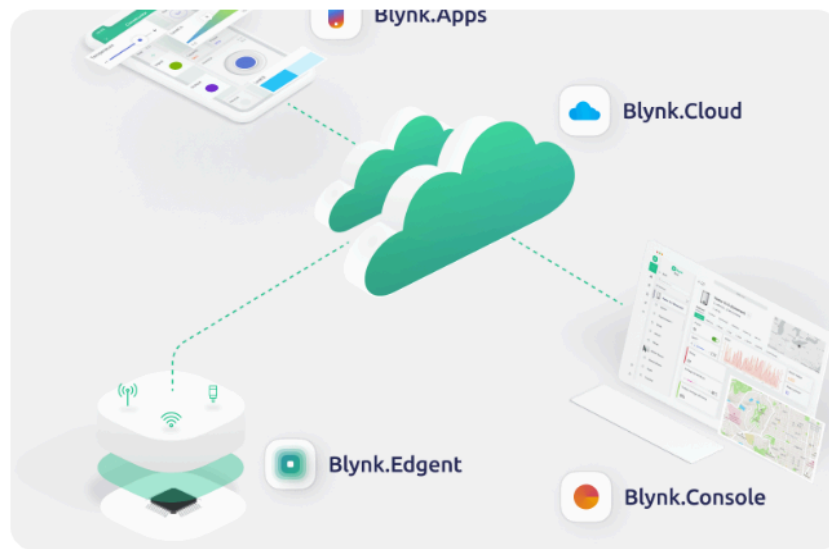


Fig. 2 - Esquema de funcionamiento de Blynk

- 1) **Blynk.CLOUD: servidor** que envía datos de forma segura entre los dispositivos y las aplicaciones.
- 2) **Blynk.APPS: aplicaciones móviles** para iOS (Sistema Operativo de iPhone) y Android. Permiten crear una interfaz de usuario para dispositivos sin codificación, la cual puede ser compartida con usuarios.
- 3) **Blynk.EDGENT:** incluye **librerías** que permiten la comunicación de dispositivos con la nube. Trae provisionamiento dinámico de WiFi y actualizaciones OTA (Over The Air). **Asegura hardware compatible en todo momento.**
- 4) **Blynk.CONSOLE: aplicación web** que permite configurar, conectar y supervisar los dispositivos. Permite además análisis de datos, actualizaciones de firmware OTA y gestión de usuarios y organizaciones.

Ventajas

- Facilidad de uso
- Rapidez de puesta en marcha
- Comunidad de respaldo
- Mucha información en la web
- Tiene actualizaciones y sigue funcionando en equipos instalados
- Permite generar interfaces gráficas “arrastrando” Widgets existentes

Nota: los Widgets son pequeñas aplicaciones o programas que interactúan con servicios e información de internet y nos proveen información visual.

Desventajas

- Dependencia de un tercero
- Versión gratuita limitada
- Gráfico de variables pago
- Gestión de usuarios pago
- Almacenamiento de más de un mes pago
- Más de 10 widgets pago
- Escalón de precios muy grande para pocos beneficios extra
- Desde una misma cuenta no se puede ser usuario y administrador de distintos proyectos al mismo tiempo

Modos de funcionamiento

Blynk opera en dos Modos:

- **Modo Desarrollador:** permite configurar los dispositivos para especificar su comportamiento.
- **Modo Usuario:** permite monitorear y controlar los dispositivos, pero no permite modificar ninguna configuración.

Protocolo

Los **dispositivos** que se conectan a Blynk pueden utilizar WiFi, Ethernet, Cellular (GSM) o Satellite. Los datos son transferidos mediante:

- Protocolo Blynk (vía Blynk.Edgent o librería Blynk)
- MQTT
- HTTP(s)

Datastream

Es una forma de estructurar datos que fluyen regularmente dentro y fuera del dispositivo. Se utilizan para intercambiar datos con sensores, actuadores, microcontroladores, etc., por telemetría. Los datos pueden ser:

- **Integer:** números enteros positivos y negativos
- **Double:** números con decimales positivos y negativos
- **String:** cadenas de caracteres

A cada dato se le debe asignar un **CANAL**, por ejemplo un **PIN VIRTUAL**. Los canales son una forma de intercambiar datos entre el hardware y la aplicación Blynk, sirviendo como el medio para intercambiar datos. Los pines virtuales no tienen representación física. La información se envía como cadenas y no existen límites prácticos en los datos que se pueden enviar. Sin embargo, existen ciertas limitaciones en el lado del hardware cuando se trata de números. Por ejemplo, la variable de tipo entero (int) en Arduino es de 16 bits, lo que permite representar enteros entre -32768 a 32767.

Template

Los Templates almacenan las configuraciones de los dispositivos. Se pueden crear múltiples dispositivos individuales desde el mismo Template, facilitando el manejo de todos ellos desde un mismo lugar. Cada template está compuesto por:

- **Data Streams** (canales para transferir datos entre dispositivos)
- Interfaz de usuario de la **aplicación móvil**
- Interfaz de usuario del **panel web**
- Configuración del **ciclo de vida de la conexión**
- Configuración de **Metadata**
- **Eventos** y **notificaciones**

Dashboard

El Dashboard nos permite visualizar datos de un dispositivo o controlar el dispositivo a distancia mediante widgets. Los widgets pueden organizarse en el panel mediante una o más pestañas (dependiendo de la versión).

II. CONFIGURACIÓN DE BLYNK

1. Crear una cuenta en Blynk

Para crear una cuenta ingresar a la página <https://blynk.io/> e ingresar a *START FREE*

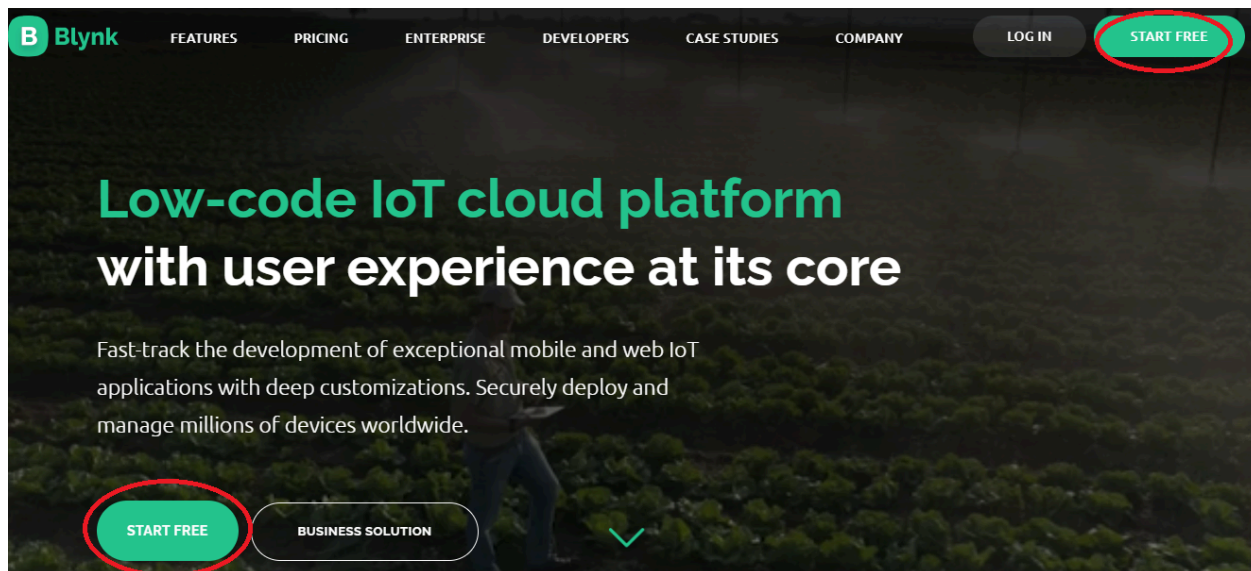


Fig. 3 -Creación de cuenta en Blynk

Luego completar con un mail y aceptar términos y condiciones.

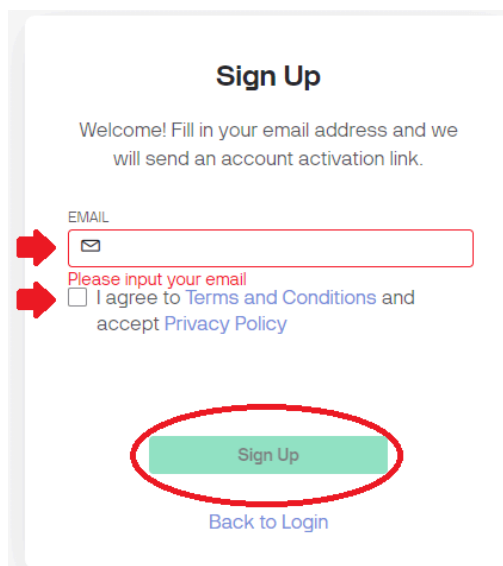


Fig. 4 - Registro en Blynk

Una vez que te hayas registrado recibirás en tu casilla de correo electrónico un nuevo mail para crear contraseña (Create Password).

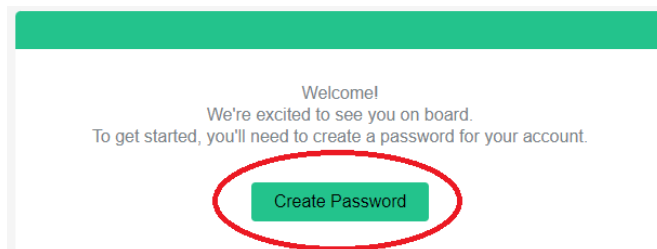


Fig. 5 - Creación de contraseña

Crea tu Password e ingresa desde la página de Blynk en "Log In" con tu usuario (mail) y contraseña.

2. Creación de un nuevo TEMPLATE

Una vez que ingresas a la plataforma, para comenzar un nuevo desarrollo, debemos crear un nuevo template. Para ello ingresamos en el menú de la izquierda a **"Developer Zone"** → **"My templates"** → **"New Template"**

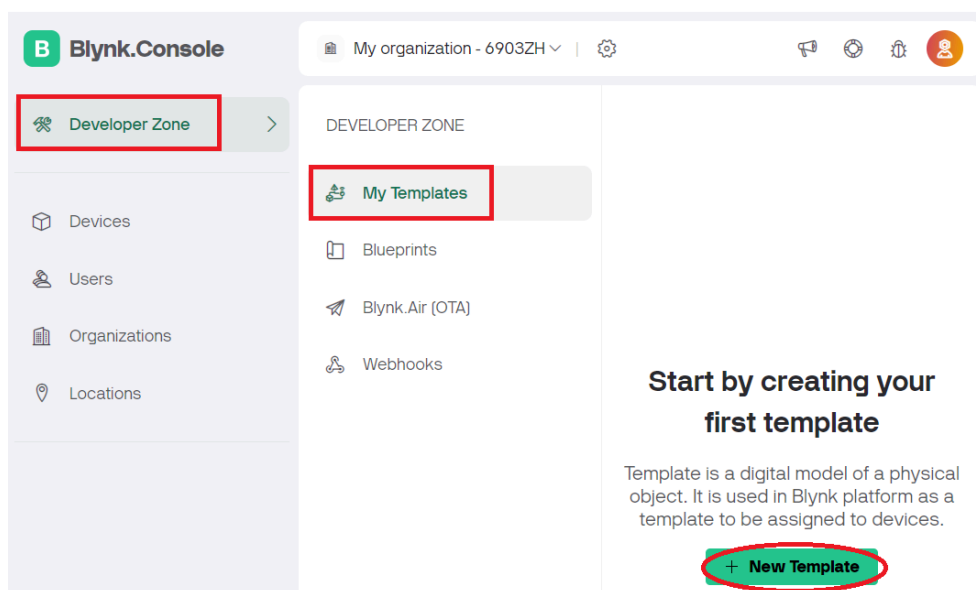
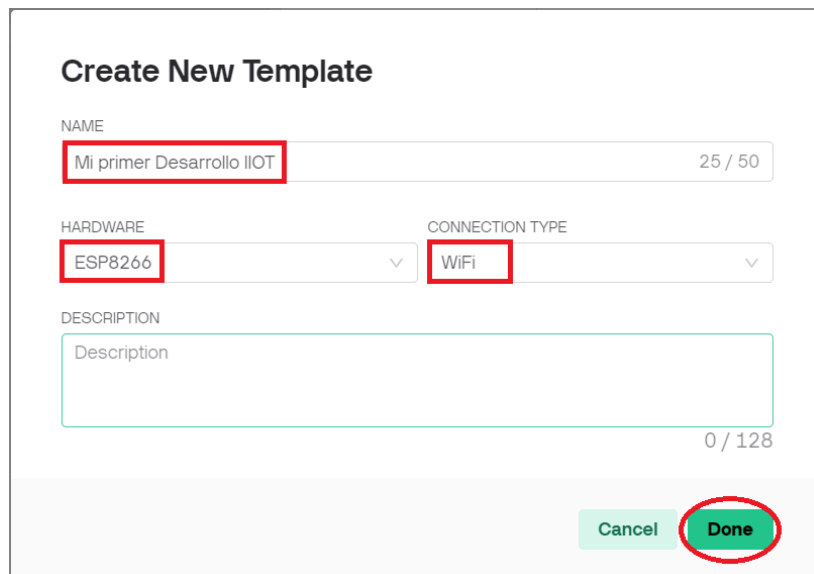


Fig. 6 - Creación de nuevo Template

Luego debemos asignarle al nuevo template un nombre, el hardware y el tipo de conexión. Nuestro Hardware es **ESP8266** y el tipo de conexión **WiFi**.



Create New Template

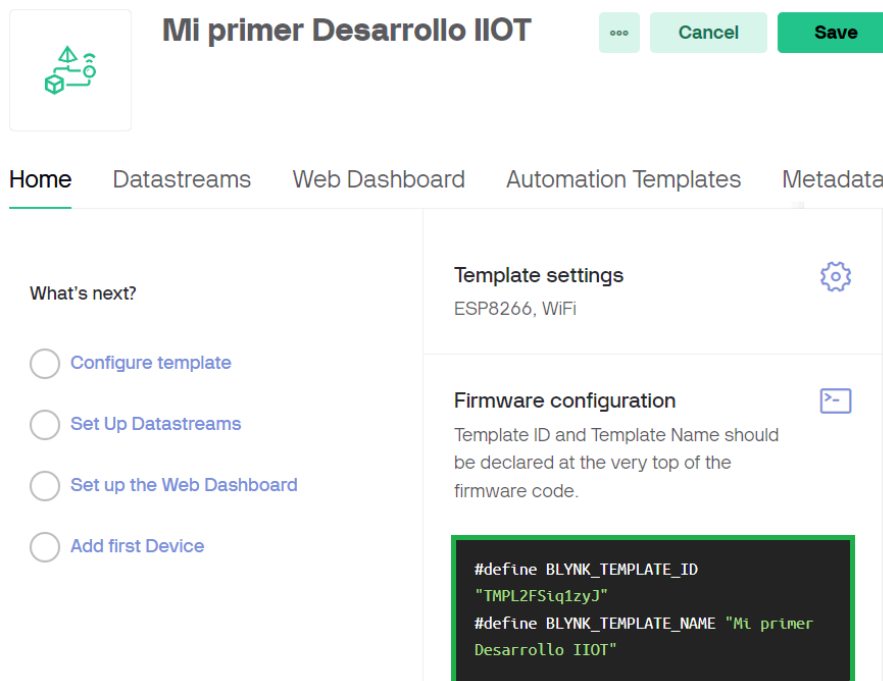
NAME
 25 / 50

HARDWARE CONNECTION TYPE

DESCRIPTION
 0 / 128

Fig. 7 - Configuración del Template

Una vez creado nuestro template, Blynk nos muestra sus CREDENCIALES, formadas por un nombre (TEMPLATE_NAME) y un identificador (TEMPLATE_ID). Esta información la necesitaremos luego en el código de nuestro ESP8266.



Mi primer Desarrollo IIOT

Home Datastreams Web Dashboard Automation Templates Metadata

What's next?

- ☐ Configure template
- ☐ Set Up Datastreams
- ☐ Set up the Web Dashboard
- ☐ Add first Device

Template settings ESP8266, WiFi

Firmware configuration

Template ID and Template Name should be declared at the very top of the firmware code.

```
#define BLYNK_TEMPLATE_ID  
"TMPL2FSiq1zyJ"  
#define BLYNK_TEMPLATE_NAME "Mi primer  
Desarrollo IIOT"
```

Fig. 8 - Credenciales del Template

Luego ingresamos a Datastreams, para comenzar a cargar las **variables virtuales** de nuestro Template.

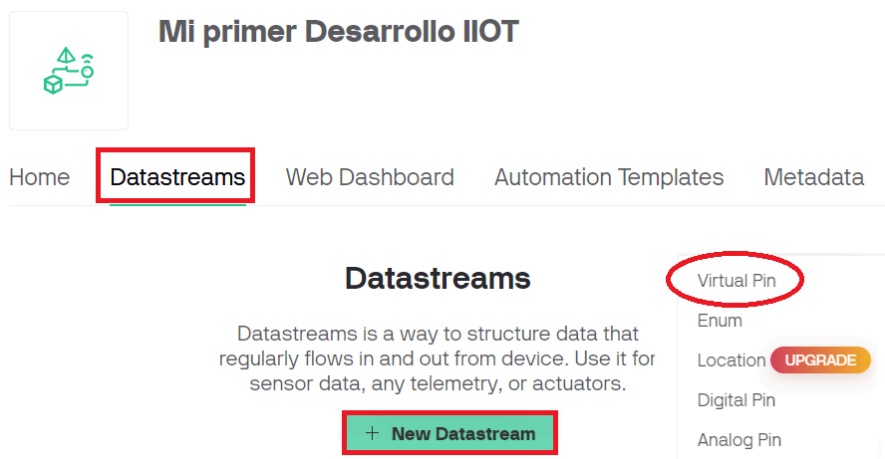




Fig. 9 - Creación de Pines Virtuales dentro del Template

Ejemplo para SWITCH en V0 (activará LED de nuestra Placa)

- **Nombre:** SWITCH (a elección del programador).
- **Pin virtual:** V0 (a elección del programador). Deberá coincidir con el declarado en el código del ESP8266.
- **Tipo de datos:** Integer (entero). En esta aplicación es suficiente con una variable que pueda representar los dos estados (ON y OFF)
- **Mín./Máx.:** 0/1 (a elección del programador). Se pueden elegir otros valores que luego deberán ser interpretados en el código del ESP8266.

Virtual Pin Datastream

General Expose to Automations

NAME		ALIAS	
	SWITCH	LED	
PIN		DATA TYPE	
V0		Integer	
UNITS			
None			
MIN	MAX	DEFAULT VALUE	
0	1	0	

Cancel
Create

Fig. 10 - Configuración Pin Virtual de nuestro SWITCH

Nota: el valor de DEFAULT es en el cual inicializa. Seleccionamos 0, para que el dispositivo comience en estado apagado (según lo programado en el ESP8266).

Ejemplo para TEMPERATURA en V1 (leerá T° de nuestra sonda DS18B20)

- **Nombre:** Temperatura (a elección del programador).
- **Pin virtual:** V1 (a elección del programador). Deberá coincidir con el declarado en el código del ESP8266
- **Tipo de datos:** Double (para números decimales positivos y negativos).
- **Mín./Máx.:** -55 °C a 125 °C (rango de T° de nuestra sonda)

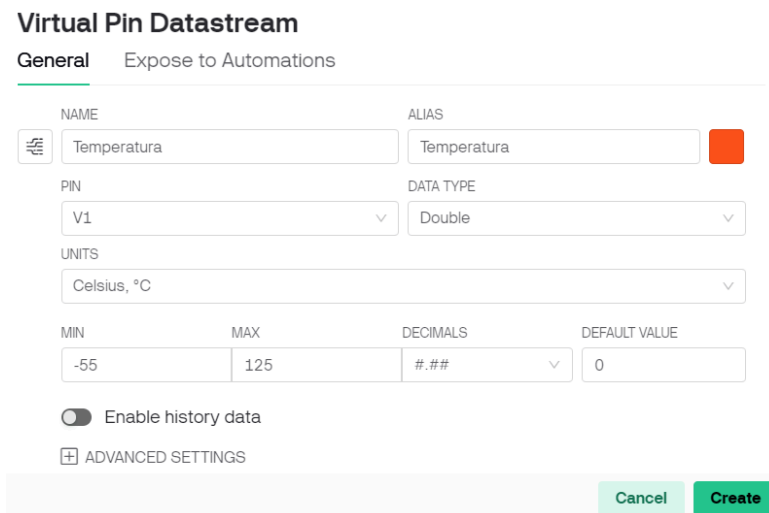


Fig. 11 - Configuración Pin Virtual de TEMPERATURA

Nota: en configuraciones avanzadas podemos elegir gradientes de colores para la representación visual, en función de la T°.

Arrastra Widgets a tu Dashboard. Por ejemplo, podemos agregar un tipo “switch” para V0 y un tipo “Gauge” para V1.

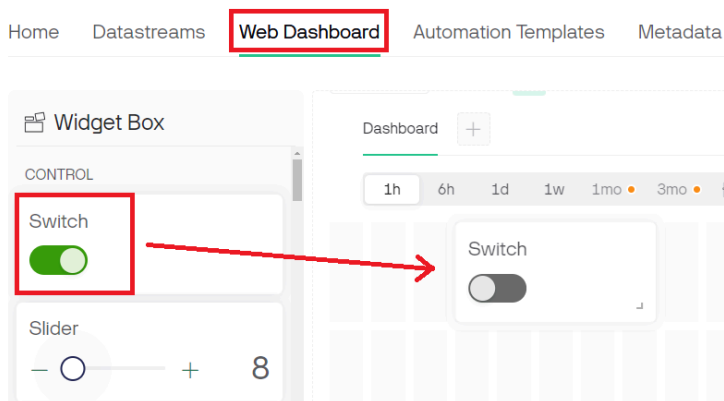


Fig. 12 - Arrastra los Widgets a tu Dashboard

A continuación, configura cada Widget (desde el ícono de engranaje).

Switch Settings ⓘ

TITLE (OPTIONAL)

Datastream

▼ 🗑️

ON VALUE OFF VALUE 🎨

☐ Show on/off labels

☐ Hide widget name

Cancel Save

Fig. 13 - Configuración de nuestro Switch

Gauge Settings

TITLE (OPTIONAL)

Datastream

▼ 🗑️

☒ Override Datastream's Min/Max fields

MIN MAX

LEVEL COLOR

☒ Change color based on value

🎨 ↩ ✖

Cancel Save

Fig. 14 - Configuración de nuestro Gauge

Ahora guardamos los cambios para crear nuestro primer dispositivo IoT.

Creación de un dispositivo IOT en Blynk

Una vez configurado el Template y creado nuestro Dashboard con Datastreams, podemos crear uno o múltiples dispositivos en Blynk, para poder enviar y recibir información con nuestro hardware a través de la nube.

Al ingresar en nuevo dispositivo (+ New Device), debemos seleccionar la opción de template y seleccionamos el template que configuramos previamente.

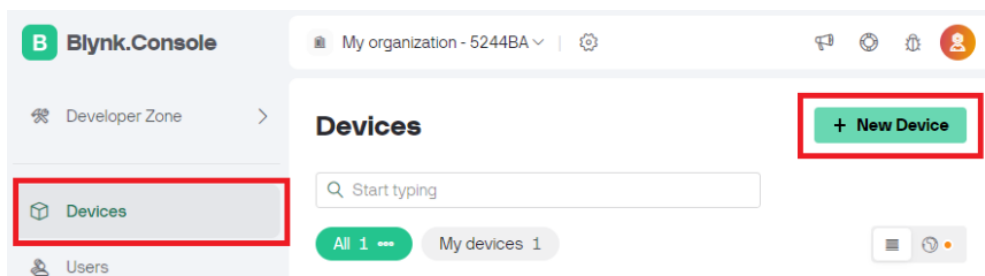
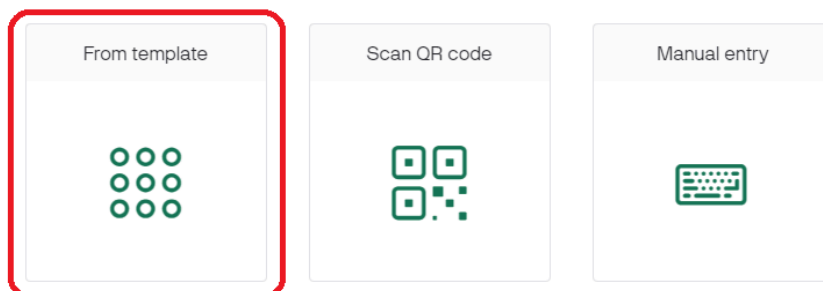


Fig. 15 - Creación de nuestro primer Dispositivo

New Device

Choose a way to create new device



Point on the cards to see instructions

Fig. 16 - Dispositivo creado a partir de nuestro Template

New Device

Create new device by filling in the form below

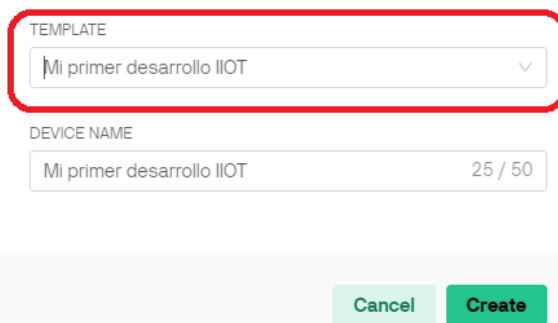
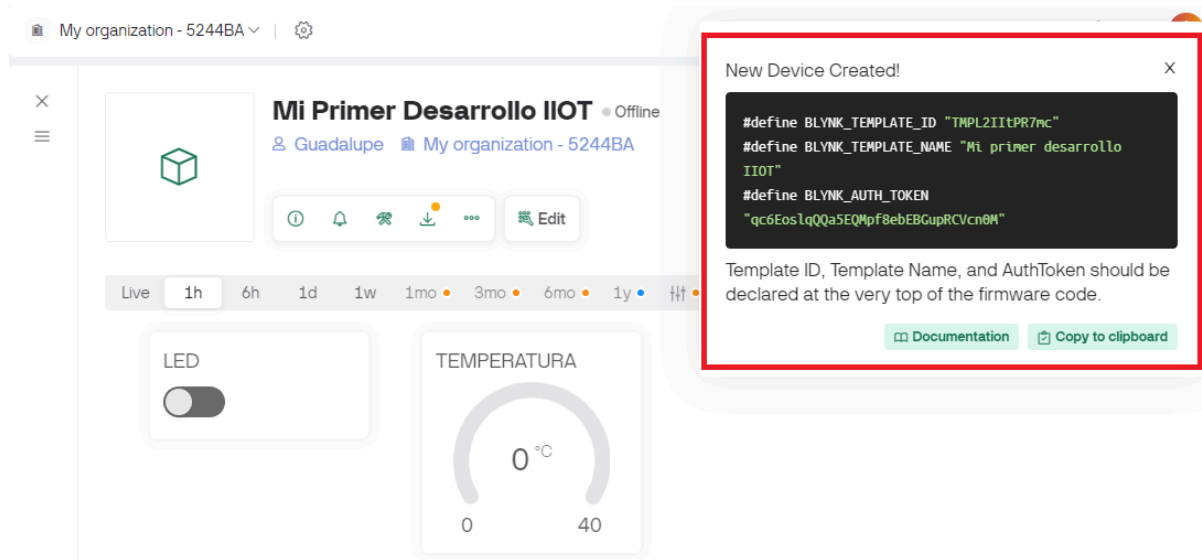


Fig. 17 - Dispositivo creado a partir de nuestro Template

Al crear un nuevo dispositivo vamos a obtener un **TOKEN** único para cada dispositivo, el cual debemos incluir en nuestro código junto con el TEMPLATE ID y el TEMPLATE NAME.

Es decir, todos los dispositivos creados a partir del mismo Template compartirán las credenciales del mismo pero tendrán su identificación única a través del TOKEN.



vo creado a partir de nuestro Template, con su TOKEN único

Vinculación del dispositivo Blynk con Placa Entrenadora TR3BOL (ESP8266)

Para vincular nuestro Dispositivo en Blynk con nuestro hardware (ESP8266), debemos incluir en el inicio del código creado en nuestro entorno de programación (Arduino IDE), el TEMPLATE ID, el TEMPLATE NAME y el AUTH TOKEN.

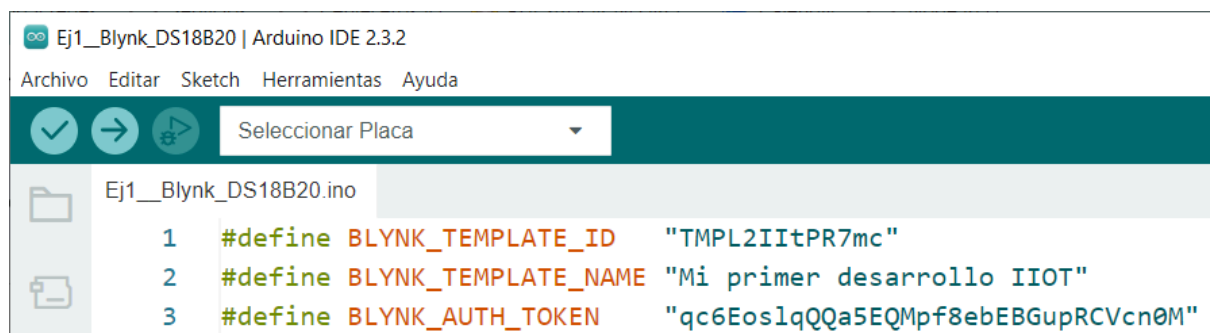


Fig. 19 - Credenciales y TOKEN de nuestro Dispositivo en Arduino IDE

RECORDAR QUE EL TOKEN ES ÚNICO PARA CADA DISPOSITIVO, por lo que en general, será diferente para cada programador. Solo en el caso de que deseemos compartir el mismo Dispositivo con otros usuarios, utilizaremos el mismo TOKEN.

A continuación debemos instalar la librería Blynk en ArduinoIDE.

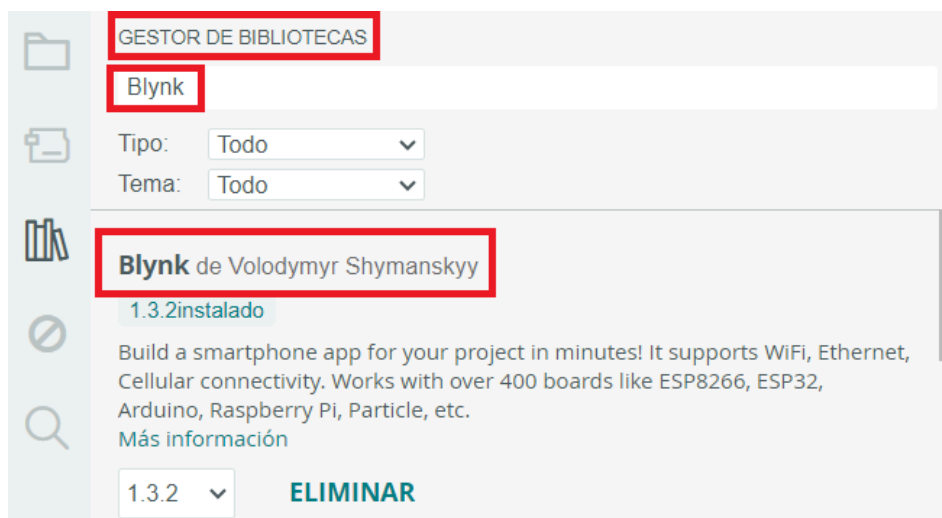


Fig. 20 - Librería de Blynk de Volodymyr Shymanskyi

Luego, podemos ver ejemplos de implementación en la siguiente página.

<https://examples.blynk.cc/>

Debemos elegir la Placa y el tipo de Conexión. Luego, podemos buscar distintos ejemplos de implementación y adaptarlos a nuestra necesidad.

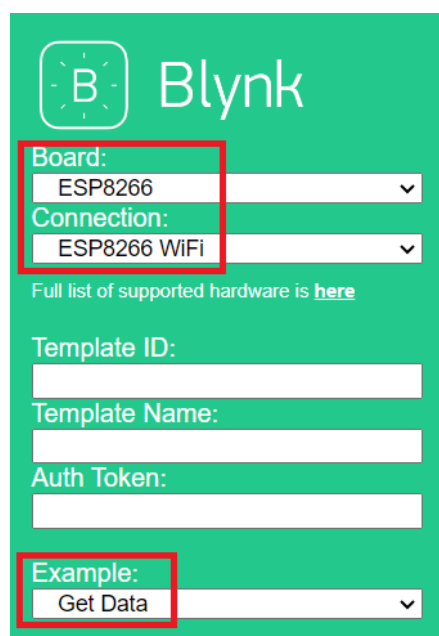


Fig. 21 - Ejemplo en obtención de datos desde Blynk hacia el hardware

Por ejemplo, en “**Get Data**” tenemos código de cómo **OBTENER** datos (desde Blynk hacia nuestra placa) y en “**Push Data**” tenemos código de cómo **ENVIAR** datos (desde nuestra placa hacia Blynk).

Vemos por ejemplo que se pueden recibir datos con la siguiente estructura:

```
// This function will be called every time Slider Widget
// in Blynk app writes values to the Virtual Pin 1
BLYNK_WRITE(V1)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  Serial.print("V1 Slider value is: ");
  Serial.println(pinValue);
}
```

Fig. 22 - Código de ejemplo de obtención de datos

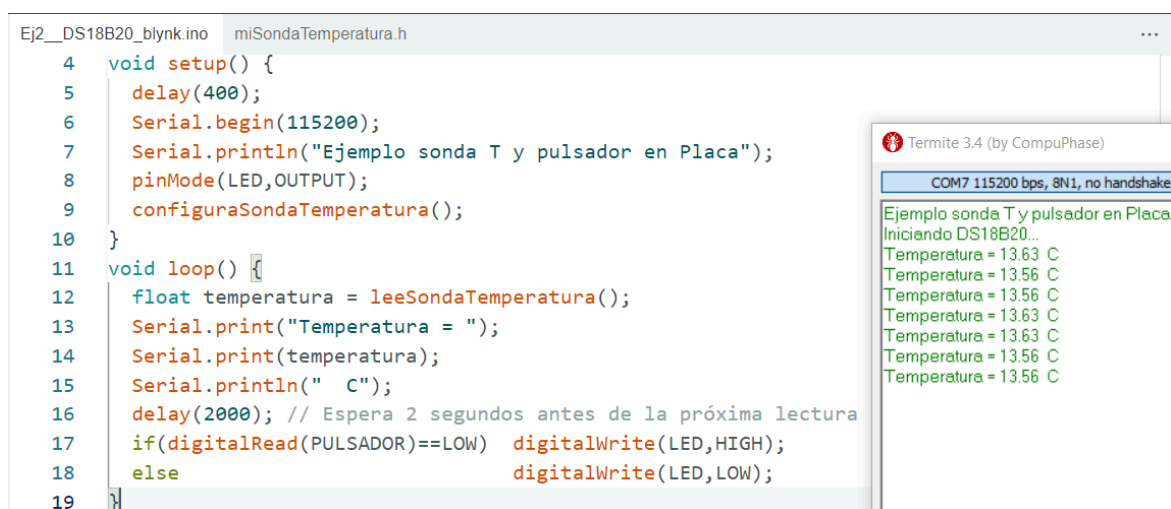
Mientras que se pueden enviar datos con la siguiente instrucción:

```
// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V5, millis() / 1000);
```

Fig. 23 - Código de ejemplo de obtención de datos

Notar que la función de escritura recibe dos parámetros. El primero es el Pin Virtual (en nuestro caso V1 para enviar la T°) y el segundo es el contenido del mensaje (en nuestro caso la variable de T° real, la cual deberemos reemplazar en el código por el argumento del ejemplo).

Veamos un ejemplo de implementación sencillo en el cual partimos del siguiente código base, que lee la T° del sensor DS18B20 y prende o apaga el led en función de un pulsador de la nuestra placa. La T° la muestra por puerto serie.



```
Ej2_DS18B20_blynk.ino  miSondaTemperatura.h  ...
4 void setup() {
5   delay(400);
6   Serial.begin(115200);
7   Serial.println("Ejemplo sonda T y pulsador en Placa");
8   pinMode(LED,OUTPUT);
9   configuraSondaTemperatura();
10 }
11 void loop() {
12   float temperatura = leeSondaTemperatura();
13   Serial.print("Temperatura = ");
14   Serial.print(temperatura);
15   Serial.println(" C");
16   delay(2000); // Espera 2 segundos antes de la próxima lectura
17   if(digitalRead(PULSADOR)==LOW) digitalWrite(LED,HIGH);
18   else digitalWrite(LED,LOW);
19 }
```

Termite 3.4 (by CompuPhase)

COM7 115200 bps, 8N1, no handshake

Ejemplo sonda T y pulsador en Placa
Iniciando DS18B20...
Temperatura = 13.63 C
Temperatura = 13.56 C
Temperatura = 13.56 C
Temperatura = 13.63 C
Temperatura = 13.63 C
Temperatura = 13.56 C
Temperatura = 13.56 C

Fig. 24 - Código de ejemplo de obtención de datos

Ahora modificamos el código para que, en lugar de leer el pulsador de la placa detecte el Switch de Blynk. Y además de mostrar la T° por puerto serie, la envía hacia Blynk.

```
Ej2_DS18B20_blynk.ino  miSondaTemperatura.h

1  // Credencial de mi Template
2  #define BLYNK_TEMPLATE_ID "TMPL2FSiq1zyJ"
3  #define BLYNK_TEMPLATE_NAME "Mi primer Desarrollo IIOT"
4  //TOKEN de mi Dispositivo
5  #define BLYNK_AUTH_TOKEN "Q83PUivL2HCUFSmxsLa3h_HUMcZGigJ9"
6
7  #include <ESP8266WiFi.h>
8  #include <BlynkSimpleEsp8266.h>
9  #include "miSondaTemperatura.h"
10 #define LED      D8
11 #define PULSADOR D0
12 // Credencial de mi red WiFi
13 char ssid[] = "TR3BOL";
14 char pass[] = "12345678";
15 BLYNK_WRITE(V0){
16     int pinValue = param.asInt();
17     if(pinValue==1) digitalWrite(LED, HIGH);
18     else             digitalWrite(LED, LOW);
19 }
20 void setup() {
21     delay(400);
22     Serial.begin(115200);
23     Serial.println("Ejemplo sonda T y pulsador en Placa");
24     pinMode(LED,OUTPUT);
25     configuraSondaTemperatura();
26     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
27 }
28 void loop() {
29     float temperatura = leeSondaTemperatura();
30     Serial.print("Temperatura = ");
31     Serial.print(temperatura);
32     Serial.println(" C");
33     Blynk.virtualWrite(V1,temperatura);
34     Blynk.run();
35     delay(2000); // Espera 2 segundos antes de la próxima lectura
36 }
```

Fig. 25 - Código de ejemplo de obtención de datos

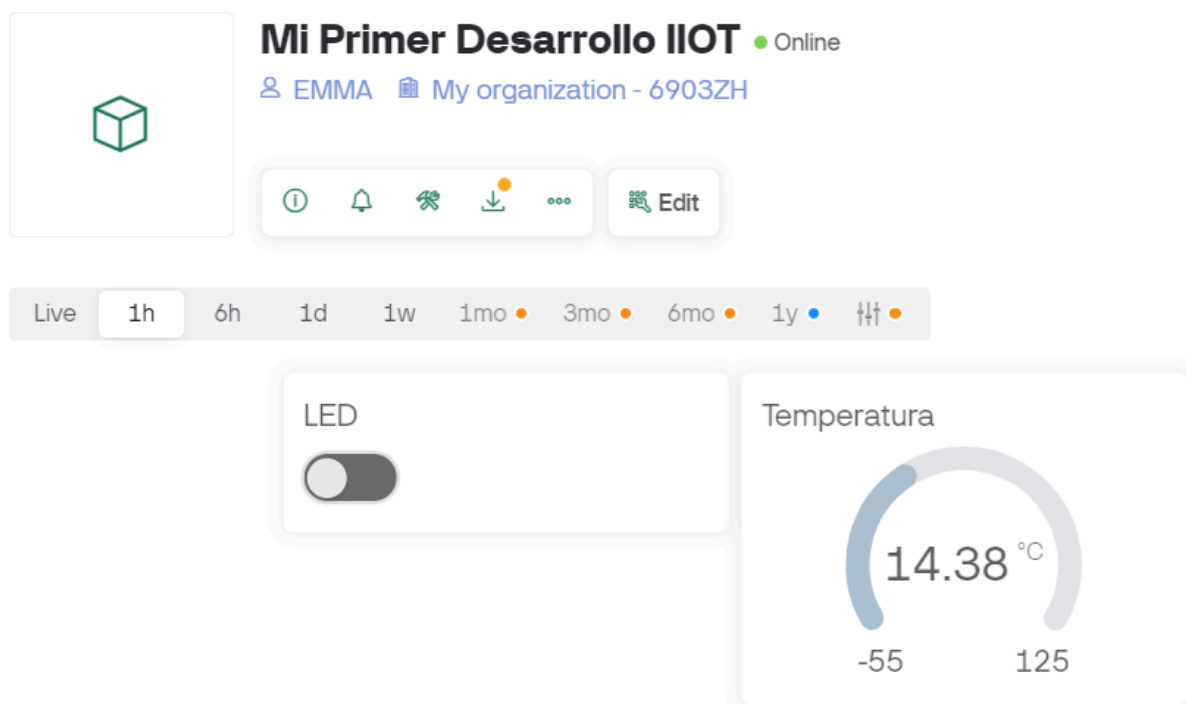


Fig. 26 - Interfaz en Blynk

Creación de una App móvil

Para crear la aplicación desde un celular, primero debes descargar Blynk IoT en tu dispositivo de preferencia. Por ejemplo, desde la PlayStore de tu celular Android.

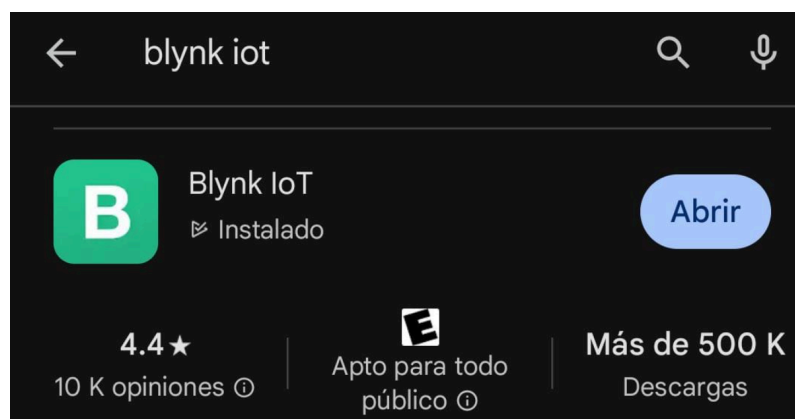


Fig. 27 - Blynk IoT en Android

Luego debes ingresar con tus credenciales y crear tus dispositivos de manera similar que en la Web. En la interfaz de celulares normalmente hay más opciones de visualización que en la Web.