



FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS

ASIGNATURA:

INTERNET DE LAS COSAS IoT

GUÍA DE LABORATORIO: TEORÍA

PRÁCTICA DE LABORATORIO N.º 4

MENSAJERÍA INSTANTÁNEA: TELEGRAM

*Este material de apoyo académico
se hace para uso exclusivo de los alumnos
de la Universidad de Lima y en concordancia
con lo dispuesto por la legislación sobre
los derechos de autor: Decreto Legislativo 822*

PRÁCTICA DE LABORATORIO N.º 4

MENSAJERÍA INSTANTÁNEA: TELEGRAM

1. OBJETIVOS

- Comprensión de las Universal Telegram Bot.
- Creación de un Bot en Telegram
- Realizar un correcto cableado entre el módulo y el ESP32.
- Encendido de diversos LED y lecturas de sensores vía Telegram.

2. FUNDAMENTO TEÓRICO

2.1 Telegram

Telegram es una aplicación de mensajería enfocada en la velocidad y seguridad, actualmente tiene más de 700 millones de usuarios activos mensuales y es una de las 10 apps más descargadas del mundo. Se puede usar Telegram en teléfonos inteligentes, tablets e incluso computadoras, se tiene una app nativa para macOS y una app universal de escritorio para Windows, macOS y Linux. Se puede usar también Telegram Web. La API de Telegram es abierta para desarrolladores, en caso de que se quiera construir una aplicación propia para otras plataformas.

Todos los chats y grupos en Telegram son privados entre sus participantes. No se procesa ninguna solicitud relacionada a ellos. Pero los packs de stickers, canales públicos, grupos públicos y bots en Telegram, están disponibles públicamente.



Imagen 1: Icono oficial de Telegram

2.2 API Telegram

Telegram ofrecen dos tipos de API para desarrolladores. La API Bot le permite crear fácilmente programas que usan mensajes de Telegram para una interfaz. La API de Telegram y TDLib le permiten crear sus propios clientes de Telegram personalizados.

La API Bot le permite conectar bots a nuestro sistema. Los Telegram Bots son cuentas especiales que no requieren un número de teléfono adicional para configurar. Estas cuentas sirven como interfaz para el código que se ejecuta en algún lugar de su servidor.

Para usar esta API no necesita saber nada sobre cómo funciona el protocolo de cifrado MTProto: el servidor intermediario se encargará de todo el cifrado y la comunicación con la API de Telegram por cada usuario. Se comunica con este servidor a través de una interfaz HTTPS simple que ofrece una versión simplificada de la API de Telegram.

La API de Telegram y TDLib permite crear sus propios clientes de Telegram personalizados. Está 100% abierto para todos los desarrolladores que deseen crear aplicaciones de Telegram en la plataforma de Telegram. Cada usuario se siente libre de estudiar el código fuente abierto de las aplicaciones



Imagen 2: Bot de Telegram botfather

3. CONFIGURACIONES

3.1 CHAT BOT Telegram

Una vez tengamos instalado y configurado el Telegram en nuestro teléfono, Tablet, etc. Debemos de buscar el botfather, este contacto es un bot pre-configurado por Telegram para la creación de bots. Debemos de buscar el contacto con el check azul para poder configurar, tal como se muestra en la imagen.

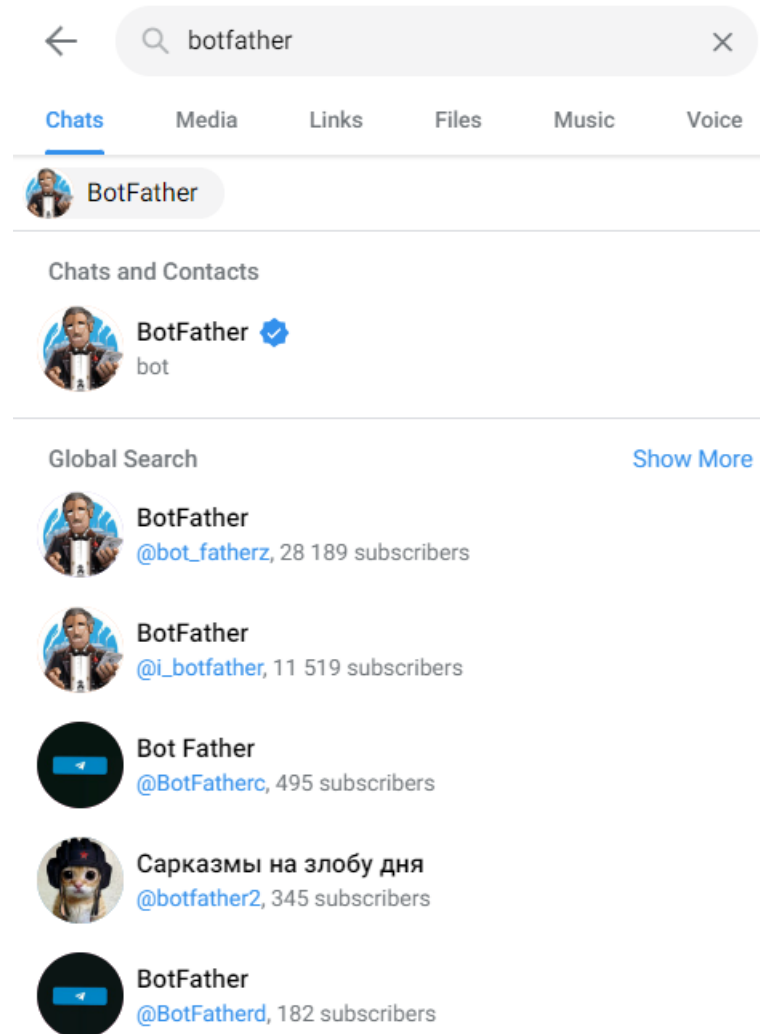


Imagen 3: Buscando el contacto del botfather

Una vez estemos en el chat, debemos de escribir el comando /start con este comando se mostrarán todas las opciones disponibles como la creación y edición de los bots:

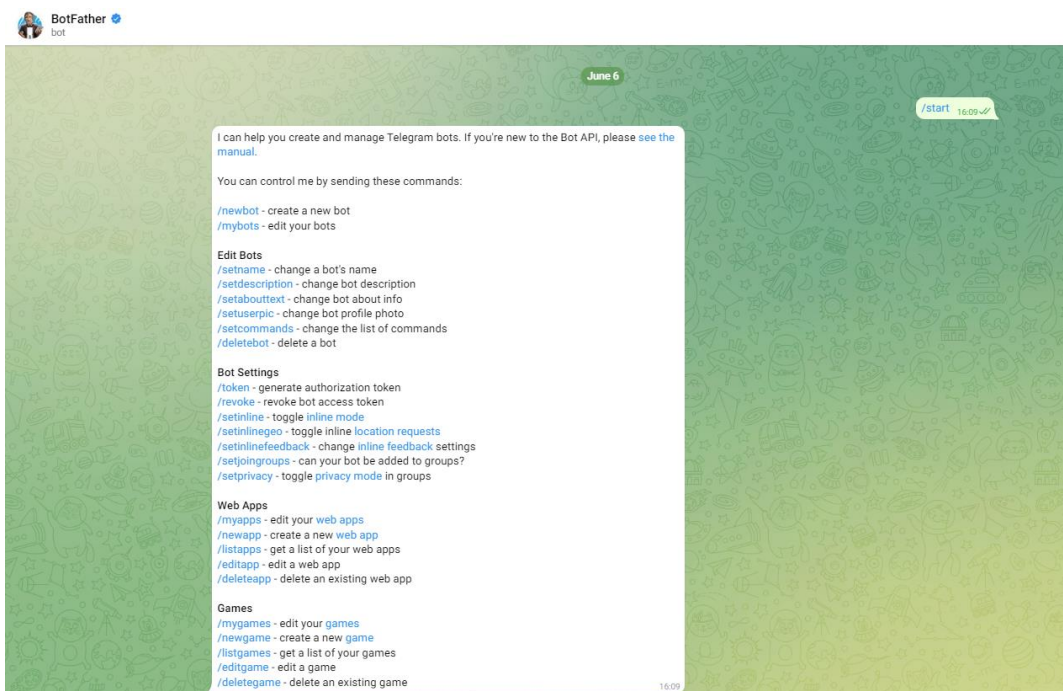


Imagen 4: Opciones de edición y creación de bots

En nuestro caso queremos crear un bot, por lo que el comando a elegir es /newbot. El siguiente paso es darle un nombre al bot, para la creación de esta guía se uso el nombre Prueba1 pero cada alumno es libre de colocarle el nombre que guste, si se tiene planeado usar el Telegram para el trabajo de investigación, se debe decidir el nombre del bot de manera grupal. Finalmente se debe de crear un nombre único al bot, este nombre debe de terminar con la palabra bot, en caso que exista el nombre el mismo sistema de lo hará saber, para esta guía se usó el nombre Brueba1bot por ende este nombre no se podrá usar, cada uno asignará su bot. A continuación una imagen referencial:



Imagen 5: Asignación del nombre del bot

Una vez creado el bot, el botfather devolverá un mensaje con el API del bot, ES DE EXTREMA IMPORTANCIA MANTENER ESTE API DE MANERA PRIVADA, si se comparte el API se podrá suplantar la identidad de la persona, al finalizar el trabajo se recomienda eliminar el bot desde el mismo botfather. El mensaje es similar a la siguiente imagen, por motivos de seguridad se borra gran parte de la API.

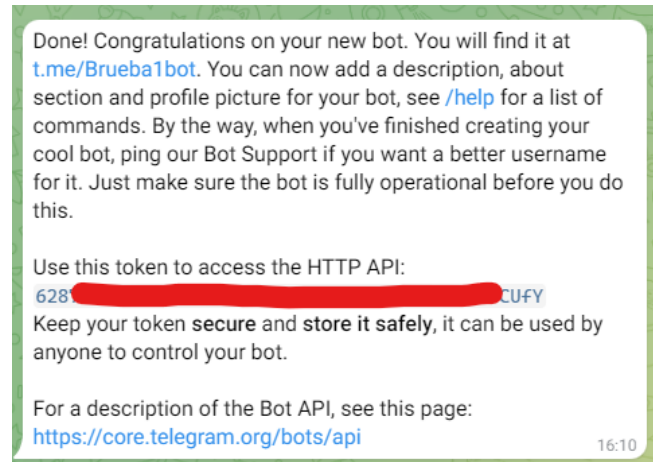


Imagen 6: API del bot creado

Una vez creado el bot, debemos de configurar el ID del bot. El id de usuario es un número único para cada chat, grupo y usuario que le permite a Telegram identificar usuarios y chats. En este caso, cualquiera que tenga el enlace al bot puede interactuar con él. Para prevenir accesos no autorizados vamos a realizar una encriptación el bot usando dicho id. Haciendo esto, cada vez que el microcontrolador reciba un mensaje del bot revisará si es nuestro usuario quien manda el mensaje, y solo ejecutará el comando cuando seamos nosotros.

En el Telegram debemos de buscar el IDBot para poder conocer el ID, una vez dentro del IDBot escribimos el comando /getid y automáticamente se entregará el ID del bot, tal como se muestra en la imagen:



Imagen 7: ID del usuario

Ahora que ya tenemos configurado el chatbot, el siguiente paso es configurar el ESP32 para poder realizar la comunicación entre nuestro microcontrolador y el servicio de mensajería.

3.2 Configuración ESP32

El primer paso es instalar la librería de Telegram. Usaremos la librería Universal de Telegram creada por Brian Logh que provee una interfaz fácil de usar para la API de Telegram. Dicha librería se adjunta con la guía en formato .zip en el IDE de Arduino se debe de incluir la librería siguiendo la siguiente ruta: Sketch > Include Library > Add.ZIP Library... Y listo. Para leer todos los detalles de la librería puedes revisar la documentación en GitHub. De manera adicional se debe de instalar la librería ArduinoJson, para instalar la librería de Json necesitas buscarla desde el administrador de librerías de Arduino. Abre Sketch > Include Library > Manage Libraries y busca ArduinoJson. Selecciona la versión más reciente e instálala.

Incluimos las librerías necesarias para el correcto funcionamiento de nuestro código:

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
```

Como usamos la librería WiFi, debemos de conectarnos a una red WiFi abierta, para ello definimos nuestra red y contraseña:

```
// Reemplazar con los datos de tu red wifi
const char* ssid = "REDWIFI";
const char* password = "PASSWORD";
```

El siguiente paso es el más importante, debemos de introducir las credenciales de nuestro bot: el API y su ID, lo definimos con el siguiente código:

```
//Reemplazar con los datos del telegram
#define token_Bot "TOKENPERSONAL" //Token de Telegram BOT se obtienen desde Botfather en telegram
#define ID_Chat "IDPERSONAL" //El ID se obtiene de (IDBot) en telegram no olvidar hacer click en iniciar
```

El último paso en la definición de las variables es crear un cliente seguro, para poder encriptar nuestros mensajes y definir nuestro bot Telegram:

```
WiFiClientSecure secured_client;
UniversalTelegramBot bot(token_Bot, secured_client);
```

Procedemos a configurar el ESP32 dentro de la subrutina void setup. Lo primero es habilitar el monitor serial y conectarnos a la red WiFi, tal como los laboratorios anteriores de WiFi


```
Serial.begin(115200);
Serial.print("Conectando a la red wifi... ");
Serial.println(ssid);
//Seteo de la red Wifi
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT);
while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
}
Serial.println("");
Serial.println("Conectado a la red wifi!!!");
Serial.print("Dirección ip: ");
Serial.println(WiFi.localIP()); //Imprimimos la dirección ip local
```

Cuando ya se conecte a la red WiFi, podemos enviar un mensaje a nuestro chatbot usando el siguiente comando:

```
bot.sendMessage(ID_Chat, "Sistema preparado para usar telegram!", ""); //Enviamos un mensaje a telegram para informar que el sistema está listo
```

Este mensaje es solo para poder asegurarnos que llegan los mensajes adecuadamente. Si se siguió la guía nos debería de llegar un comando a nuestro Telegram con el mensaje que configuramos de manera manual.

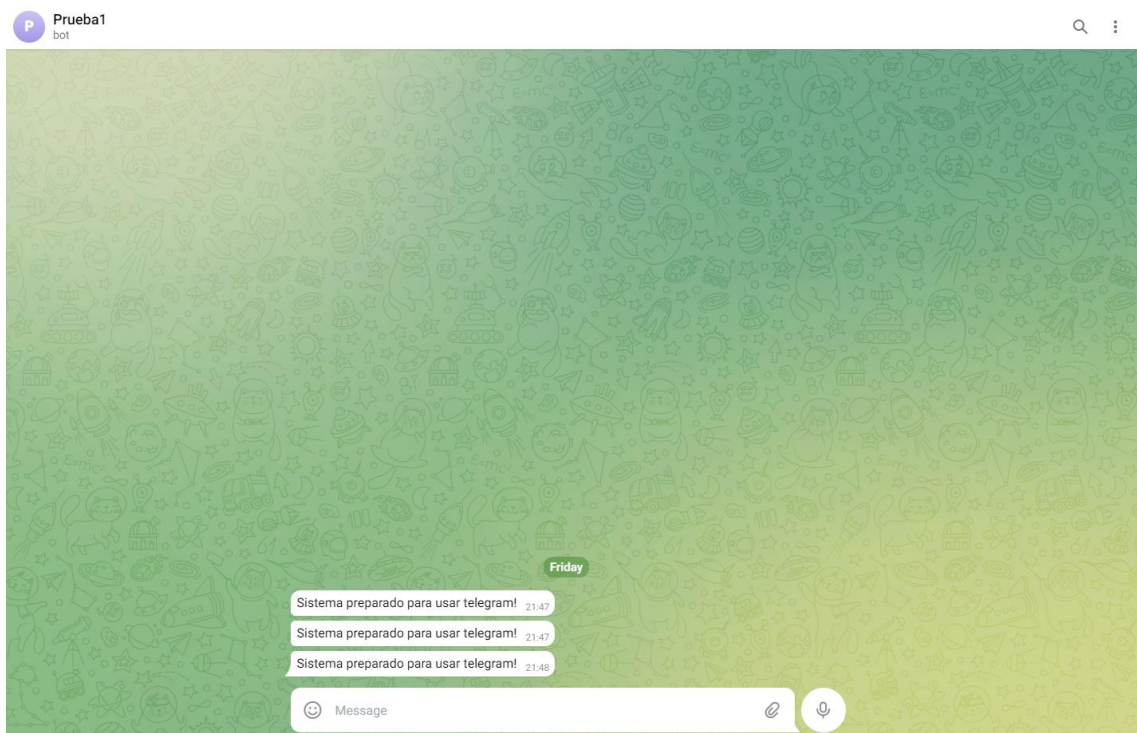


Imagen 8: Chat de telegram

Los siguientes pasos se revisarán directamente de códigos ejemplos mostrados por el profesor.