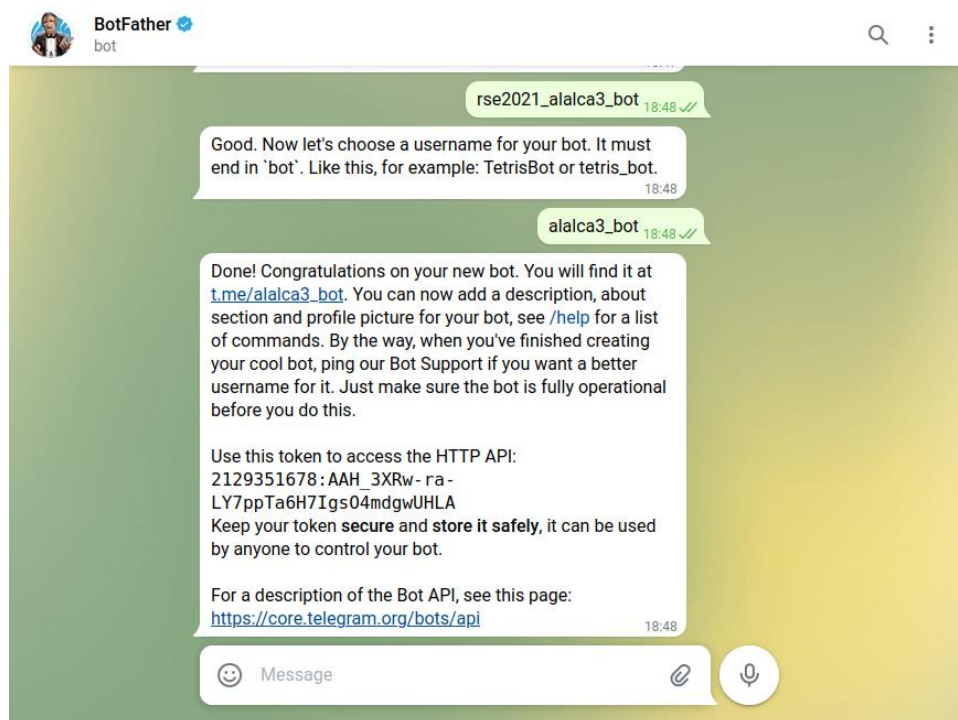


RSE 2021-2022 / Alejandro Albert Casañ

Memoria práctica 5

1. Comprueba que el Bot funciona.

Una vez hemos instalado el paquete de python-telegram-bot y tenemos a nuestra disposición el código de la práctica, creamos el bot mediante un chat con BotFather que nos irá haciendo preguntas y finalmente nos dará el token par acceder a la API de Telegram.



Teniendo ya el token, inicializamos los parámetros que usaremos para el bot y para recibir los datos del TTN en el código de la práctica.

```
#Parámetros de configuración de TTN
config = {
    "Broker": "eul.cloud.thethings.network",
    "Username": "lopys2ttn@ttn",
    "Password": "NNSXS.A55Z2P4YCH2RQ70NQVFX2IPMPJQLXAPKQSWQ.A5AB4GALMW623GZMJEWNIVRQSMRMZF4CHDBTTEQYR",
    "Topic": "v3/+/devices/#"
}

#Parámetros de configuración del bot
user = "alalca3_bot"
bot = "rse2021_alalca3_bot"
token = "2129351678:AAH_3XRw-ra-LY7ppTa6H7Igs04mdgwUHLA"
```

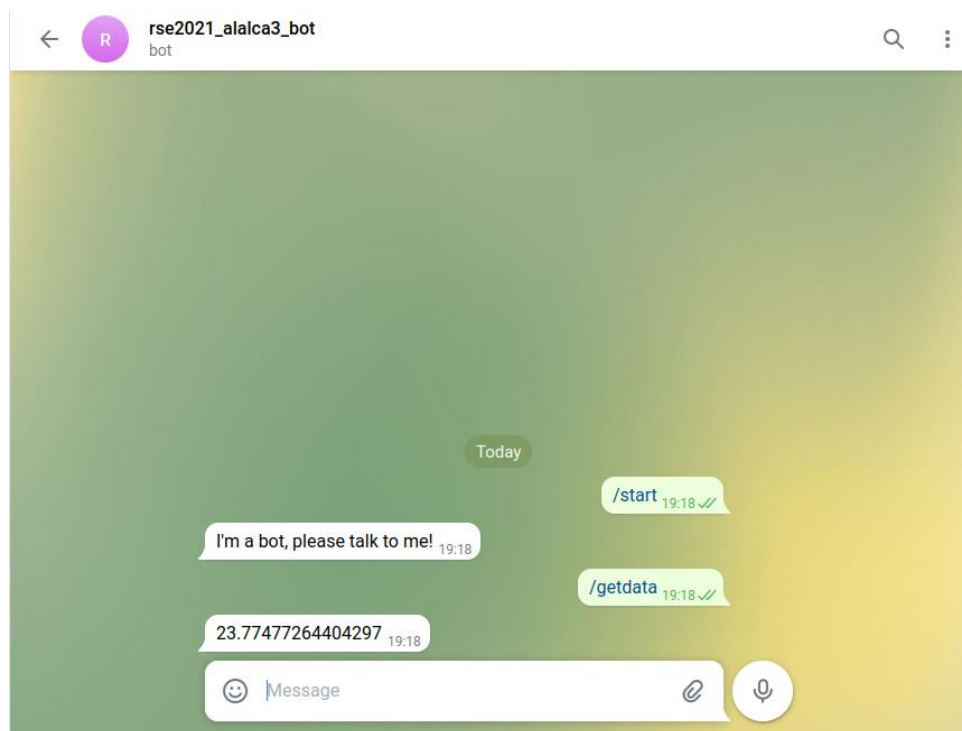
Más tarde, los utilizamos para conectarnos al TTN y a la API de Telegram.

```
client = mqtt.Client()

client.on_connect = on_connect
client.on_message = on_message
client.username_pw_set(
    config["Username"],
    password=config["Password"]
)
client.connect(config["Broker"], port=1883, keepalive=60)
client.loop_start()

updater = Updater(token=token, use_context=True)
dispatcher = updater.dispatcher
```

Una vez hecho, podemos ejecutar el script y probar el funcionamiento del bot.



El bot funciona correctamente.

2. Modifica el Bot para que pueda devolver el valor de temperatura, humedad y luminosidad en función de la petición.

A continuación, vamos a realizar las modificaciones oportunas:

En primer lugar, crearemos unas variables globales que nos servirán para llevar los registros deseados. Estas variables reemplazarán a r_value.

```
#Variables para la recogida de datos
help_value = "/gettemp - Valor de la temperatura \n /
temp_value = "VOID"
lum_value = "VOID"
hum_value = "VOID"
```

A continuación, haremos que cuando se reciba el mensaje se actualicen las variables anteriores.

```
def on_message(client, userdata, msg):
    global temp_value
    global lum_value
    global hum_value
    print("msg received with topic: {} and payload: {}".format(
        msg.topic, str(msg.payload)))

    if (msg.topic == "v3/lopy2ttn@ttn/devices/lopy4sense/up"):
        themsg = json.loads(msg.payload.decode("utf-8"))
        dpayload = themsg["uplink_message"]["decoded_payload"]

        print("@%s >> temp=%.3f hum=%.3f lux=%.3f" %
              (time.strftime("%H:%M:%S"), dpayload["temperature"],
               dpayload["lux"], dpayload["humidity"]))

        temp_value = dpayload["temperature"]
        lum_value = dpayload["lux"]
        hum_value = dpayload["humidity"]
```

Más tarde, crearemos las rutinas que se seguirán cuando el usuario nos envíe los distintos comandos. Cada rutina enviará el contenido de la variable global del parámetro que representa.

```
def gettemp(update, context):
    context.bot.send_message(chat_id=update.effective_chat.id,
                             text=str(temp_value))

def gethum(update, context):
    context.bot.send_message(chat_id=update.effective_chat.id,
                             text=str(hum_value))

def getlum(update, context):
    context.bot.send_message(chat_id=update.effective_chat.id,
                             text=str(lum_value))
```

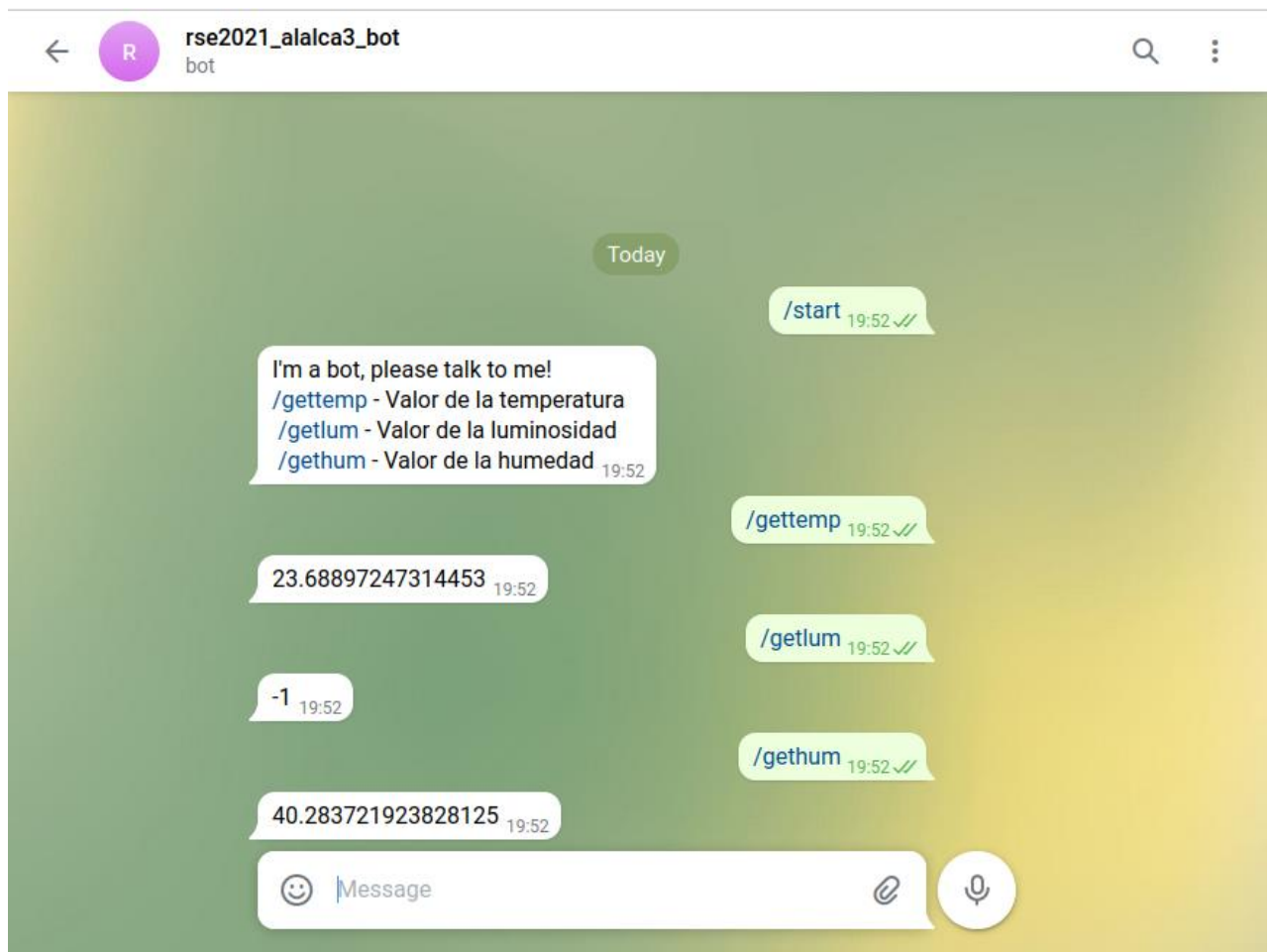
Por último, solo queda asignar esas rutinas como los manejadores que responderán a sus respectivos comandos.

```
gettemp_handler = CommandHandler('gettemp', gettemp, pass_args=False)
dispatcher.add_handler(gettemp_handler)

getlum_handler = CommandHandler('getlum', getlum, pass_args=False)
dispatcher.add_handler(getlum_handler)

gethum_handler = CommandHandler('gethum', gethum, pass_args=False)
dispatcher.add_handler(gethum_handler)
```

Ahora solo queda probar el resultado:



Obtenemos el resultado esperado, así que, podemos dar por concluido el ejercicio. El código corresponde al fichero lab5.py

Código de la práctica

El código de los ficheros de la práctica será adjuntado en el momento de la entrega, pero también se puede encontrar en [mi repositorio de GitHub de la asignatura](#).