



# Bots de Telegram – Parte 2

Controla tus sistemas desde Telegram

# ¿Que vimos en la Parte 1?

- En la primera parte vimos:
  - Como crear un Bot con BotFather
  - Ejemplos en ShellScript
    - Bot HolaMundo
    - Bot ServerStatus
  - Ejemplo Python
    - Bot ServerStatus

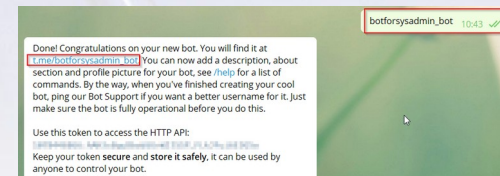
# ¿Que veremos ahora?

- Modificaciones de Bot ServerStatus
  - Para enviar estado del servidor cada X segundos
  - Para comunicarnos con el Bot
    - Nos permitirá comenzar/detener los envíos del estado
  - Ejemplo Python
    - Bot ServerStatus

# Repaso rápido Parte 1



- Creamos Bot con BotFather
  - Obtenemos usuario del Bot
  - Guardamos Token



- Obtenemos nuestro ID con @UserInfoBot



- Programamos Bots

```
#!/bin/bash

TOKEN="Escriba aqui su token" # Rellenar con el Token
ID_RECEPTOR="1552233" #Del estilo 1552233
MENSAJE="HOLA_MUNDO"
URL="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web, incluyendo Token
# Se requiere CURL instalado ("sudo apt-get install curl")

# Envia por Post a la URL dada, al chat con el usuario ID el mensaje
curl -s -X POST $URL -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
```

# Programa Status servidor (1)

- **Status servidor:** partimos del programa realizado en la parte 1.

```
1  #!/bin/bash
2
3  TOKEN="Escriba aqui su token" # Rellenar con el Token
4  ID_RECEPTOR="1552233" #Del estilo 1552233
5  #Genera el mensaje sobre el estado de apache2
6  MENSAJE=`sudo systemctl status apache2 2>&1`
7  URL="https://api.telegram.org/bot\$TOKEN/sendMessage" #URL API Web, incluyendo Token
8
9  # Se requiere CURL instalado ("sudo apt-get install curl")
10 # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
11 curl -s -X POST $URL -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
```

# Programa Status servidor (2)

- **Status servidor bucle:** modificamos el programa para que se repita en bucle.
- Usamos sleep para parar X segundos.
  - sleep 20
    - Para nuestro programa 20 segundos

# Programa Status servidor (3)

- Status Servidor Bucle (1)

```
#!/bin/bash

TOKEN="Escriba aqui su token" # Rellenar con el Token
ID_RECEPTOR="1552233" #Del estilo 1552233
URL="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web, incluyendo Token
# Se requiere CURL instalado ("sudo apt-get install curl")

#Bucle infinito
while :
do
    #Duerme 20 segundos
    sleep 20
    #Genera cada vez el mensaje para comprobar el estado de Apache2
    MENSAJE=`sudo systemctl status apache2 2>&1`
    # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
    curl -s -X POST $URL -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
done
```



# Programa Status servidor (4)

- Status Servidor Bucle (2)

```
#!/bin/bash

TOKEN="Escriba aqui su token" # Rellenar con el Token
ID_RECEPTOR="1552233" #Del estilo 1552233
URL="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web, incluyendo Token
# Se requiere CURL instalado ("sudo apt-get install curl")

#Bucle infinito
while :
do
    #Duerme 20 segundos
    sleep 20
    #Genera cada vez el mensaje para comprobar el estado de Apache2
    MENSAJE=`sudo systemctl status apache2 2>&1`
    # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
    curl -s -X POST $URL -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
done
```



# Programa Status servidor (5)

- **Status servidor bucle activable:** modificamos el programa para que:
  - Nos informe cada X segundos
  - Los envíos se puedan:
    - Inicialmente desactivados
    - Activar (/empezar)
    - Desactivar (/parar)

# Programa Status servidor (6)

- **¿Cómo leemos mensajes de Telegram?**
  - Usamos Web API de Telegram
    - En lugar de “sendMessage” → “getUpdates”
    - Recibimos los datos en formato JSON
    - Interpretando esos datos, decidiremos si se envían o no mensajes.

# Programa Status servidor (7)

- Servidor Status Bucle Activable (1)
- Parte 1

```
#!/bin/bash
```

```
TOKEN="Escriba aqui su token" # Rellenar con el Token
```

```
ID RECEPTOR="1552233" #Del estilo 1552233
```

```
URL_SEND="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web para enviar Mensaje
```

```
URL_UPDATE="https://api.telegram.org/bot$TOKEN/getUpdates" #URL API Web para recibir Mensajes
```

```
#Inicialmente, estado desactivado, no envia notificaciones
```

```
estado="PARADO"
```

# Programa Status servidor (8)

- Servidor Status Bucle Activable (2)
- Parte 2 (1)

```
#Bucle infinito
while :
do
    #Coge los updates y los procesa para obtener la ultima orden recibida por el chat
    orden=$(curl -s -X POST $URL_UPDATE -d chat_id=$ID_RECEPTOR | sed -E 's/.*"text":"?([^,"]*)"?.*\1/' | tail -1)

    #Si la ultima orden recibida es /empezar o /parar, cambia el estado
    if [ $orden = "/parar" ];
    then
        estado="PARADO"
    elif [ $orden = "/empezar" ];
    then
        estado="FUNCIONANDO"
    fi
fi
```

# Programa Status servidor (9)

- Servidor Status Bucle Activable (3)
- Parte 2 (2)

```
#Bucle infinito
while :
do
    #Coge los updates y los procesa para obtener la ultima orden recibida por el chat
    orden=$(curl -s -X POST $URL_UPDATE -d chat_id=$ID_RECEPTOR | sed -E 's/.*"text":"?([^,"]*)"?.*\/\1/' | tail -1)

    #Si la ultima orden recibida es /empezar o /parar, cambia el estado
    if [ $orden = "/parar" ];
    then
        estado="PARADO"
    elif [ $orden = "/empezar" ];
    then
        estado="FUNCIONANDO"
    fi
```

# Programa Status servidor (10)

- Servidor Status Bucle Activable (4)
- Parte 3

```
#Enviamos el mensaje si el estado es funcionando
if [ $estado = "FUNCIONANDO" ];
then
    #Mensaje para obtener estado Apache2
    MENSAJE=`sudo systemctl status apache2 2>&1`
    #Enviamos el mensaje
    curl -s -X POST $URL_SEND -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
fi
#Duerme durante 3 segundos
sleep 3
```



# ¿Otros lenguajes de programación?

- Estos ejemplos pueden realizarse en casi cualquier lenguaje de programación: Java, NodeJS, PHP
- Los ejemplos presentados estan disponibles tanto en ShellScript como en Python en:
  - <https://github.com/sergarb1/TelegramBots>



# Ejemplo con Python

- Servidor Status Bucle Activable - Python (43 líneas)



```
1 #!/usr/bin/python3
2 import subprocess
3 import time
4
5 TOKEN="Escriba aqui su token" # Rellenar con el Token
6 ID_RECEPTOR="1552233" #Del estilo 1552233
7 #Obtenemos el status del servidor Apache2
8 comandoStatus="sudo systemctl status apache2 2>&1"
9
10 URL_SEND="https://api.telegram.org/bot"+TOKEN+"/sendMessage" #URL API Web para enviar
11 URL_UPDATE="https://api.telegram.org/bot"+TOKEN+"/getUpdates" #URL API Web para leer
12
13 #Estado de los envios. Por defecto no se envia nada.
14 estado="PARADO"
15
16 while(True):
17
18     #Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
19     p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
20     (output, err) = p.communicate()
21     ordenEstado=output.decode("utf-8") #Paso de Bytes a String UTF8
22
23     #Si la ultima orden recibida es /empezar o /parar, cambia el estado
24     if ordenEstado==" /empezar":
25         estado="FUNCIONANDO"
26     elif ordenEstado==" /parar":
27         estado="PARADO"
28
29     #Enviamos el mensaje si el estado es funcionando
30     if estado=="FUNCIONANDO":
31         #Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
32         p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
33         (output, err) = p.communicate()
34         MENSAJE=output.decode("utf-8") #Paso de Bytes a String UTF8
35
36         # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
37         comando="curl -s -X POST "+URL_SEND+" -d chat_id="+ID_RECEPTOR+" -d text='"+str(MENSAJE)+"'"
38         #Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
39         p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
40         (output, err) = p.communicate()
41
42     #Paramos 10 segundos
43     time.sleep(10)
```