



# Bots de Telegram – Parte 1

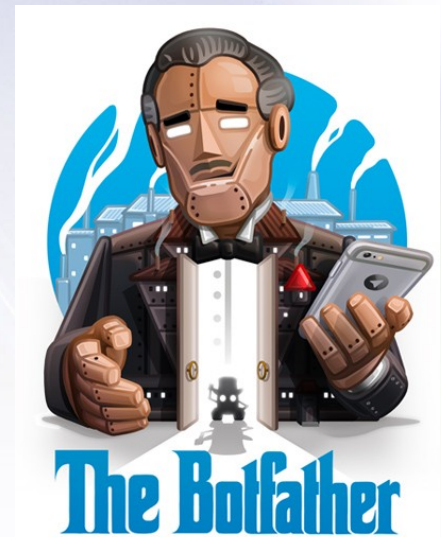
Controla tus sistemas desde Telegram

# ¿Que es Telegram?

- Servicio de mensajería instantánea
- Ventajas:
  - Multiplataforma (PC, Móvil, Tablet)
  - Grupos privados/públicos
    - No se comparte número de teléfono
  - Canales unidireccionales.
  - Bots con “BotFather”

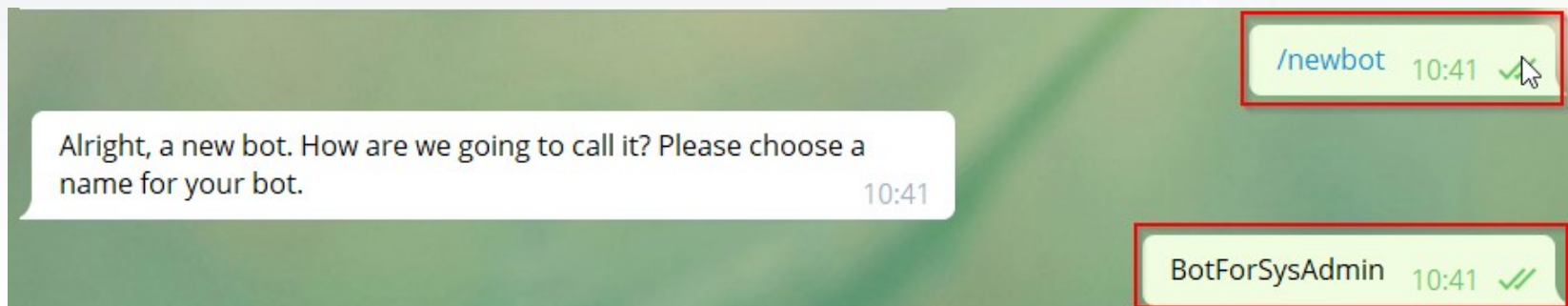
# BotFather

- BotFather
  - Herramienta para crear Bots
  - Unos pocos pasos y listo
  - Accedemos mediante @BotFather  
<https://t.me/BotFather>



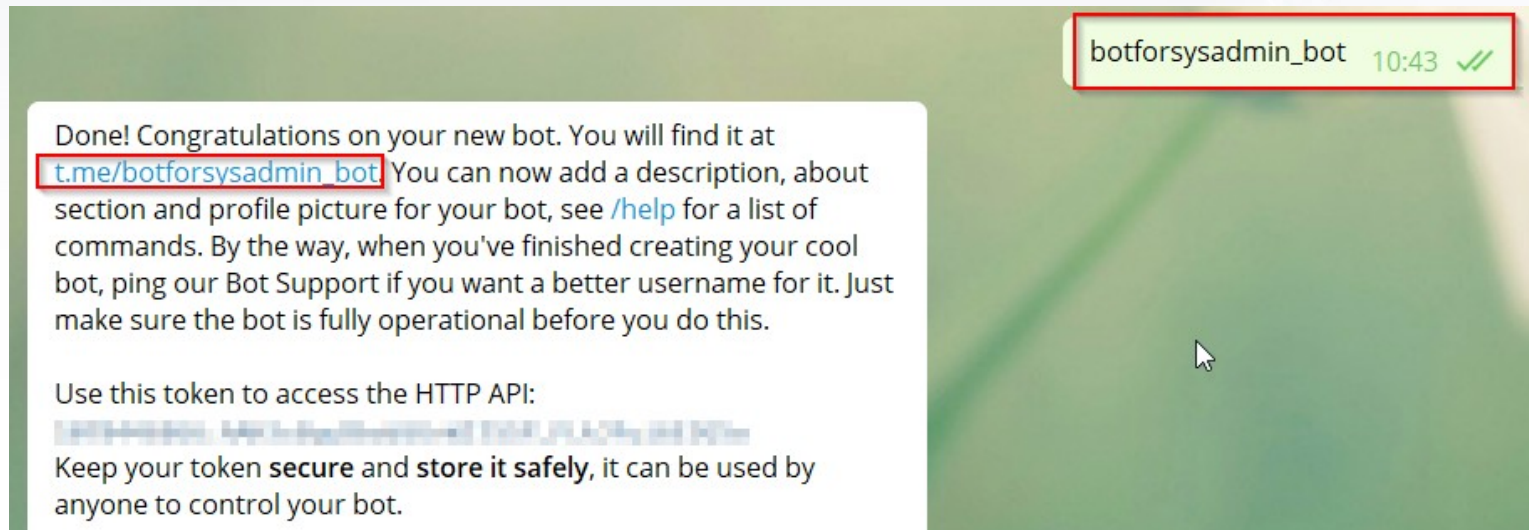
# Configurando BotFather (1)

- Si es la primera vez que accedemos a BotFather, le damos a START
- Escribimos la orden `/newbot` y damos nombre al Bot



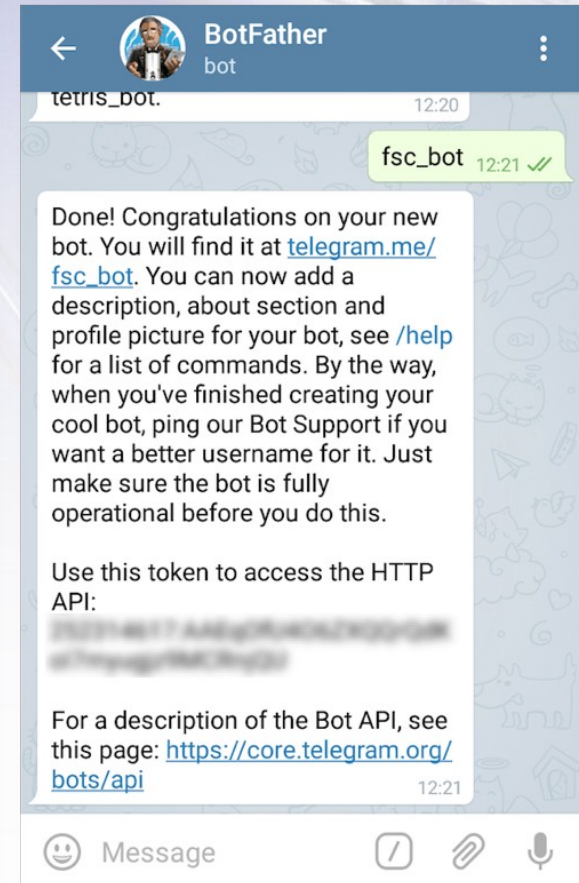
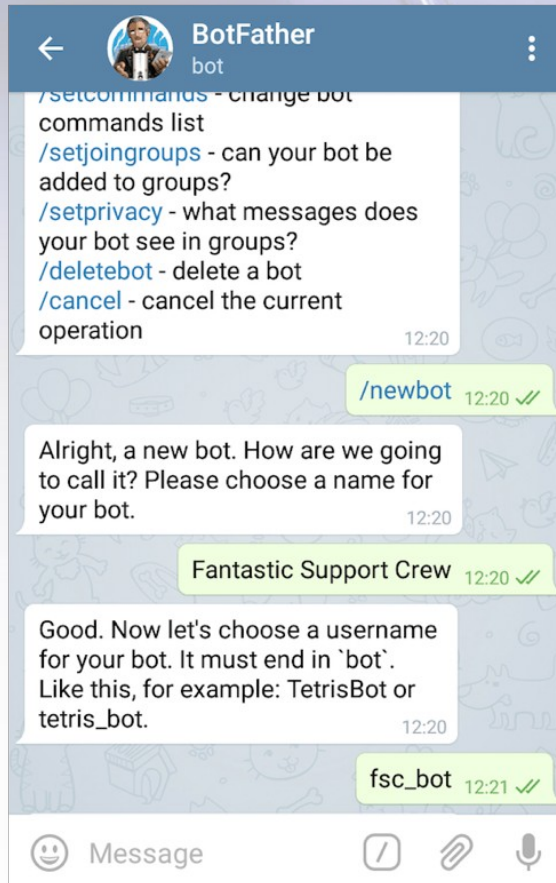
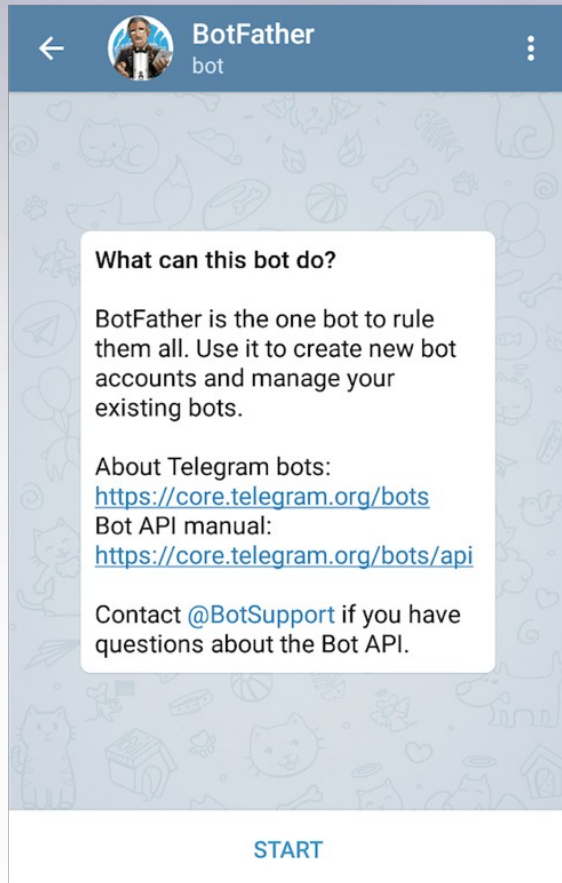
# Configurando BotFather (2)

- Ahora damos un Username al Bot. Debe ser único y acabar en `_bot`. Tras ello obtendremos:
  - URL de acceso al Bot
  - Token para acceder a la API (Debe ser secreto!)



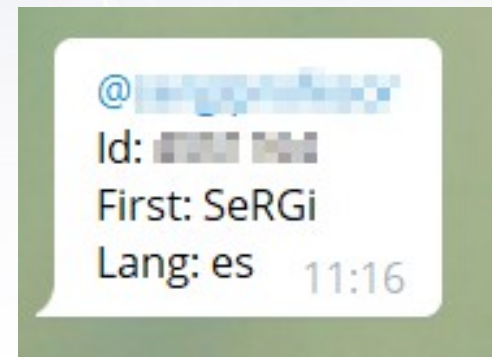


# Configurando BotFather (3)



# Configurando BotFather (4)

- Para obtener ID única Usuarios y/o Canales/Grupos.
  - Grupo/Canal: @ChannelIdBot
  - Usuarios: @UserInfoBot
    - **Necesitaremos nuestra ID**



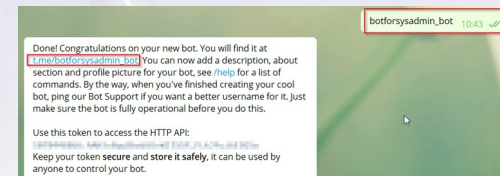
# Configurando BotFather (5)

- La primera vez que accedemos a nuestro propio bot, le damos a  `START`
  - Obligatorio para que el Bot nos pueda escribir.
- Los ejemplos utilizados disponibles en:
  - <https://github.com/sergarb1/TelegramBots>

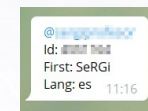


# Resumen primeros pasos

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



- Obtenemos nuestro ID con @UserInfoBot
- Todo listo para programar nuestro Bot



# Cuestiones antes de empezar (1)

- **¿Donde se ejecuta nuestro Bot?**
  - Donde queramos: nuestra máquina, un servidor...
  - Único requisito Internet para comunicarse con Telegram
- **¿En que lenguaje debemos programar el Bot?**
  - En cualquiera que nos guste.
  - Ejemplos en ShellScript y Python.

# Cuestiones antes de empezar (2)

- **¿Como se comunica el Bot con Telegram?**
  - Usando una Web API
  - Nos apoyamos en la herramienta CURL
    - “sudo apt-get install curl”
    - CURL permite realizar envios GET/POST a una URL mediante línea de comandos.

# Nuestro primer programa (1)

- **Hola mundo:** Bot que envía el mensaje “Hola mundo” a un usuario.
- Ejemplo en ShellScript
- Necesitamos
  - Token
  - ID usuario receptor
  - Mensaje a enviar

# Nuestro primer programa (2)

- Hola mundo (1)

```
#!/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"
```



# Nuestro primer programa (3)

- Hola mundo (2)

```
#!/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"
```

# Nuestro primer programa (4)

- Hola mundo (3)

```
#!/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:** Bot que envía estatus del servicio apache2.
- Orden para obtener estatus de Apache2
  - “sudo systemctl status apache2 2>&1”
    - 2>&1 hace que la salida de error se redirija a la salida estándar

# Programa Status servidor (2)

- Servidor Status – Primera versión (11 líneas)

```
#!/bin/bash

TOKEN="Escriba aqui su token" # Rellenar con el Token
ID_RECEPTOR="1552233" #Del estilo 1552233
#Genera el mensaje sobre el estado de apache2
MENSAJE=`sudo systemctl status apache2 2>&1`
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"
```

# ¿Otros lenguajes de programación?

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



# Ejemplo con Python

- Servidor Status Python

```
#!/usr/bin/python3
import subprocess

TOKEN="Escriba aqui su token" # Rellenar con el Token
ID_RECEPTOR="1552233" #Del estilo 1552233
#Obtenemos el status del servidor Apache2
comandoStatus="sudo systemctl status apache2 2>&1"
#Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
(output, err) = p.communicate()

#Pasamos output que esta en Bytes a String
MENSAJE=output.decode("utf-8")
print(MENSAJE)
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
comando="curl -s -X POST "+URL+" -d chat_id="+ID_RECEPTOR+" -d text='"+str(MENSAJE)+"'"
#Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
(output, err) = p.communicate()
```



# Bibliotecas útiles para Bots Telegram

- Los ejemplos aquí vistos se han programado el Bot usando la Web API directamente.
- Colección de bibliotecas para facilitar la tarea
  - <https://core.telegram.org/bots/samples>
- Bibliotecas destacadas
  - **Python** <https://github.com/python-telegram-bot/python-telegram-bot>
  - **NodeJS** <https://github.com/yagop/node-telegram-bot-api>
  - **PHP** <https://github.com/tg-bot-api/bot-api-base>
  - **Java** <https://github.com/rubenlagus/TelegramBots>