



Implementación de un bot en Telegram programado en Python

Héctor Hugo Moreno Pérez^{1}, Angela Pérez Florentino¹*

¹TecNM-Instituto Tecnológico Superior de Pánuco

*hecktor@itspanuco.edu.mx

RESUMEN

Los bots son programas de computadora que permiten mantener conversaciones con los usuarios de forma automatizada y pueden ejecutar acciones como reservas en hoteles, realizar compras en ecommerce, conocer estados de cuenta, controlar dispositivos de IoT. En el presente trabajo se ha utilizado una tecnología actual de código abierto para la implementación de un bot en el sistema de mensajería Telegram, programando rutinas que permiten la comunicación entre un programa ejecutado de manera local y Telegram.

Palabras claves: bot, Telegram, Python, IoT.

ABSTRACT

Bots are computer programs that allow to hold conversations with users in an automated way and can execute actions such as hotel reservations, make purchases in ecommerce, know account statements, control IoT devices.

In the present work, a current open source technology has been used to implement a bot in the Telegram messaging system, programming routines that allow communication between a locally executed program and Telegram.

Key words: bot, Telegram, Python, IoT.

INTRODUCCIÓN

“En el mundo, el sector financiero y de salud ofrecen sus servicios a una gran cantidad de clientes diariamente y a medida que un servicio es requerido por más clientes, las empresas invierten en mecanismos que les ayuden a mejorar estos servicios. Por ejemplo, en Estados Unidos, Wells Fargo & Co. y Capital Financial Corp., ponen a disposición de sus usuarios bots que permiten a sus clientes obtener información de sus estados de cuenta, ubicación de cajeros automáticos, pagar las tarjetas de crédito, etcétera. En Londres, la consultora Babylon Health desarrolló un bot capaz de dar recomendaciones a los pacientes a partir de sus síntomas”. (Estela & Huerta, 2018).

Aunado a estos hechos, la firma Juniper Research menciona en uno de sus estudios que los bots ahorrarán a las organizaciones cerca de 7.3 mil millones de dólares a nivel mundial para el 2023.

De acuerdo a (Zamora, 2020), en Latinoamérica el impacto de esta tecnología es positivo y un ejemplo de ello es el caso de éxito observado en una cadena de farmacias de Guatemala en donde se implementó un bot de atención. Los usuarios utilizan estos canales de comunicación digital para conocer las ubicaciones de las farmacias, solicitar precios de medicina y hablar con asesores y el 90% de usuarios están contentos.

Según el portal Redacción A21 México no es la excepción, Aeroméxico cuenta con su bot llamado Aerobot, que ha sido la vía de atención para 1 millón de clientes y ha resuelto el 67% de solicitudes de sus clientes.

El despliegue de las soluciones mencionadas se ha realizado con tecnologías propietarias como: Google, Zendesk, Whatsapp. Estas herramientas mantienen mecanismos para poder efectuar la comunicación con otras aplicaciones mediante el consumo de sus APIS. El manejo de estas es complicado y en ocasiones de paga.

Las Apis de Telegram son más sencillas de operar, por tal motivo son el objeto de estudio en este trabajo. Para la programación del bot se eligió el lenguaje de programación

Python por su licencia de código abierto, portabilidad y ser multiplataforma.

METODOLOGÍA

En este apartado se describen las etapas de la implementación del bot.

Etapa 1. Se crea el robot con la ayuda del asistente inteligente de Telegram estableciendo un diálogo a través de comandos:

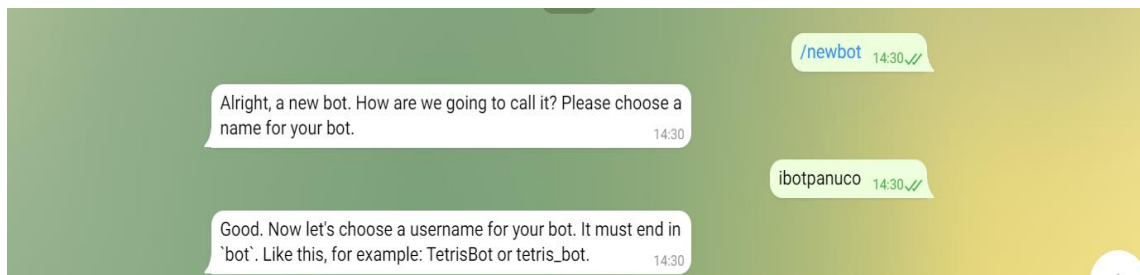


Figura 1. Lanzamiento del comando

El comando /newbot nos permite crear un bot nuevo en el servicio de mensajería Telegram, posteriormente se crea un usuario para obtener la cadena de conexión del API.

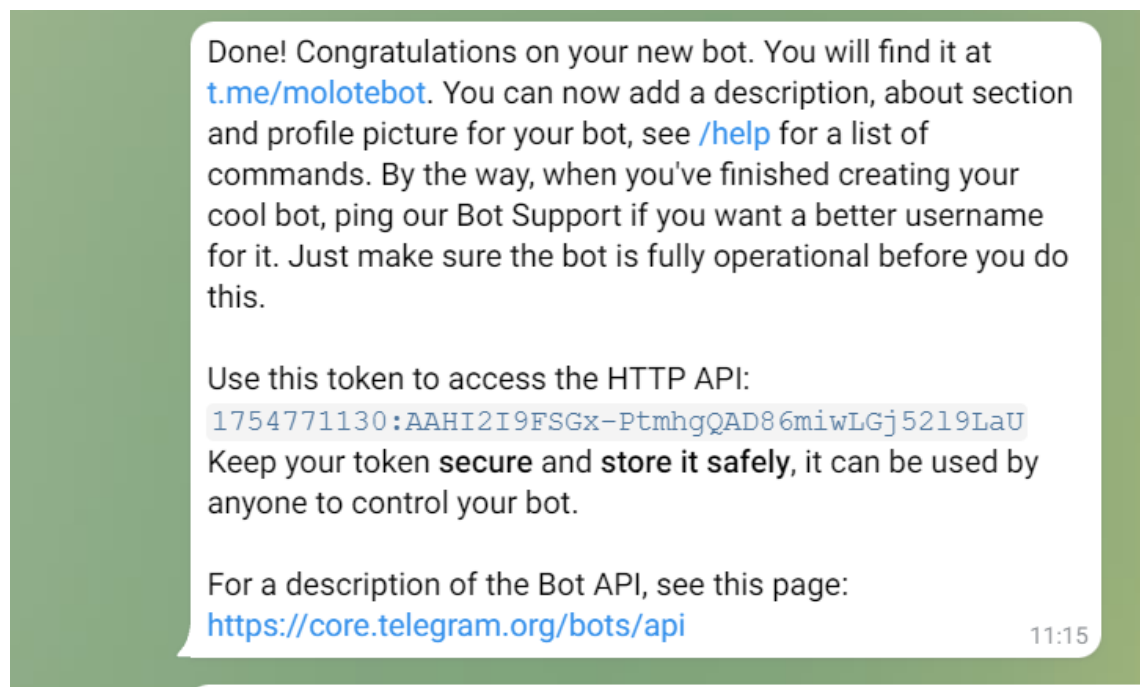
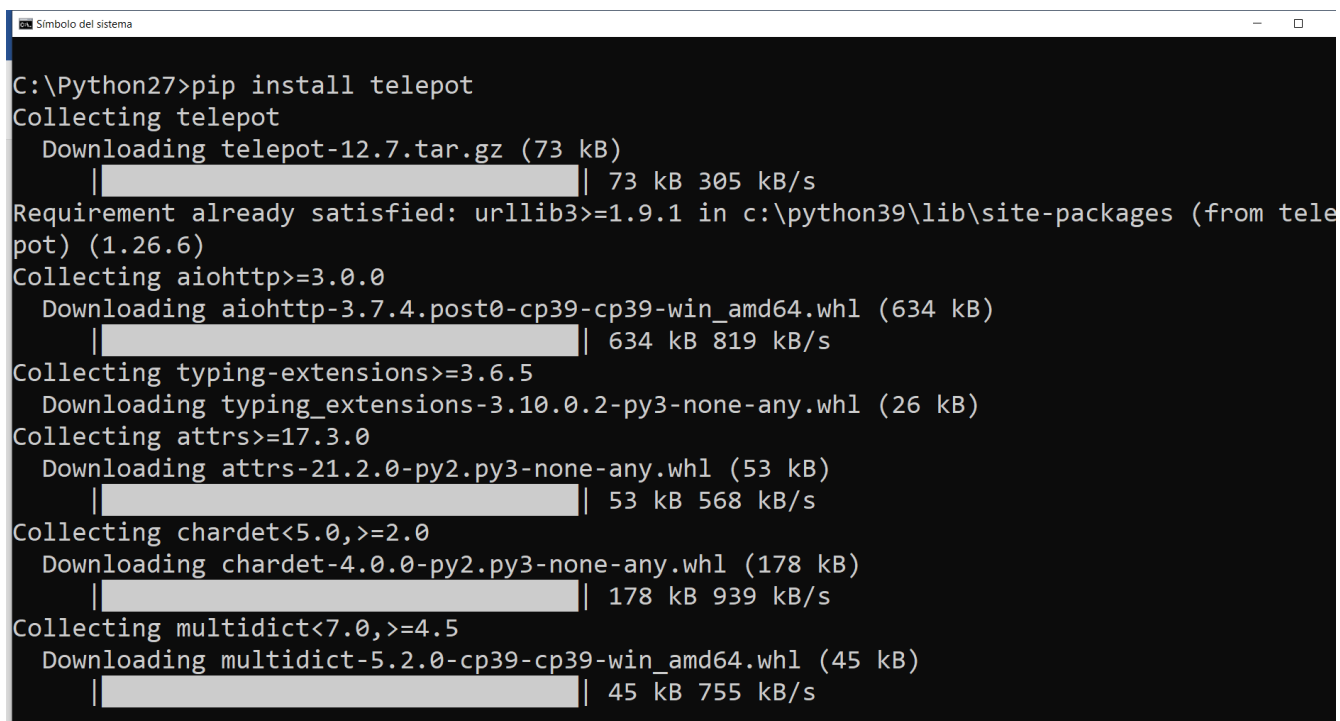


Figura 2. Obtención del token del api.

Contando con el token, ya se puede efectuar la vinculación entre aplicaciones. (Telegram Bot, 2021)

Etapa 2. Creación del vínculo entre el programa de ejecución local y el sistema de mensajería.

La programación de este enlace se realizó con el framework de desarrollo Telepot, que se puede incorporar a Python 27 a través de su instalador de paquetes pip. (Python Framework,2021).



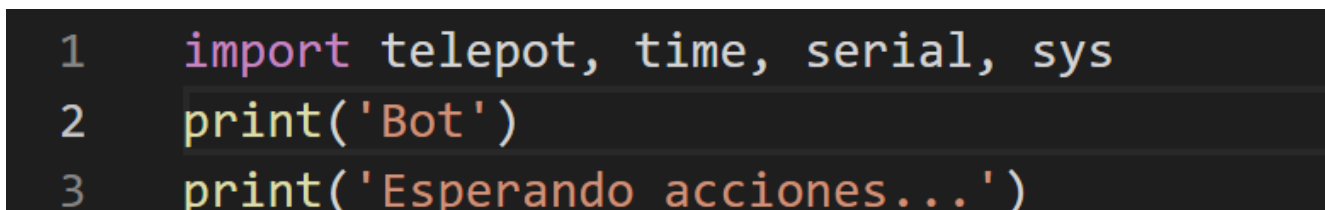
```

C:\Python27>pip install telepot
Collecting telepot
  Downloading telepot-12.7.tar.gz (73 kB)
    |████████████████████████████████████████| 73 kB 305 kB/s
Requirement already satisfied: urllib3>=1.9.1 in c:\python39\lib\site-packages (from telepot) (1.26.6)
Collecting aiohttp>=3.0.0
  Downloading aiohttp-3.7.4.post0-cp39-cp39-win_amd64.whl (634 kB)
    |████████████████████████████████████████| 634 kB 819 kB/s
Collecting typing_extensions>=3.6.5
  Downloading typing_extensions-3.10.0.2-py3-none-any.whl (26 kB)
Collecting attrs>=17.3.0
  Downloading attrs-21.2.0-py2.py3-none-any.whl (53 kB)
    |████████████████████████████████████████| 53 kB 568 kB/s
Collecting chardet<5.0,>=2.0
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
    |████████████████████████████████████████| 178 kB 939 kB/s
Collecting multidict<7.0,>=4.5
  Downloading multidict-5.2.0-cp39-cp39-win_amd64.whl (45 kB)
    |████████████████████████████████████████| 45 kB 755 kB/s

```

Figura 3. Instalación del Framework Telepot..

En el siguiente segmento procedemos a incorporar el framework de trabajo a nuestro programa en Python.



```

1  import telepot, time, serial, sys
2  print('Bot')
3  print('Esperando acciones...')

```

Figura 4. Importación de la librería Telepot.

Se define el método que funciona como controlador de los comandos recibidos desde el bot en telegram.

```

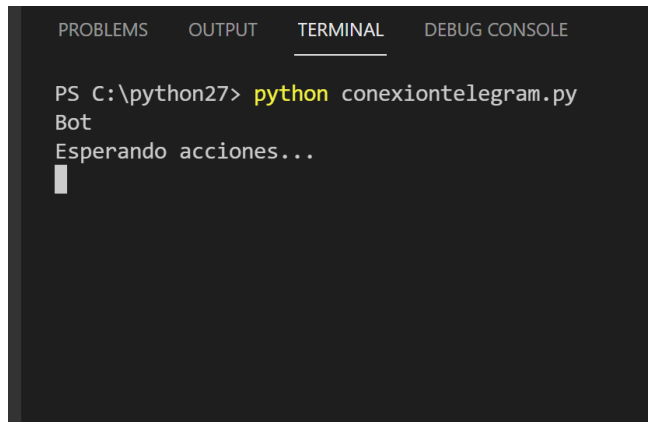
4 def handle(msg):
5     usuario = msg['from']['first_name']
6     content_type, chat_type, chat_id = telepot.glance(msg)
7     if (content_type == 'text'):
8         comando = msg['text']
9         print ('Comando obtenido: %s' % comando)
10
11     if '/start' in comando:
12         bot.sendMessage(chat_id, "Hola , "+"informatica b tardes"
13             +"\n"+"te muestro la lista de acciones que puedo reconocer:"+"\n"
14             +"Encender l1"+" -Enciendo la luz 1"+"+"\n"
15             +"Apagar l1"+" -Apago luz 1"+"+"\n"
16             +"Encender l2"+" -Enciendo la luz 2"+"+"\n"
17             +"start -Te muestro esta lista"")

```

Figura 5. Manejo del comando /start proveniente desde el sistema de mensajería.

Con el método glance del objeto telepot se extrae la información recibida desde el asistente, clasificando el tipo de contenido, el tipo de chat y el id del robot. Para responder al comando se utiliza el método sendMessage, el cual recibe 2 parámetros: el id del chat y la cadena de texto a enviar. (Severance, 2015).

Lanzamiento del programa local, esperando la comunicación proveniente de Telegram.



```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
PS C:\python27> python conexiontelegram.py
Bot
Esperando acciones...
|

```

Figura 6. Ejecución del Programa Local.

Interacción del comando /start enviado desde informaticabot.

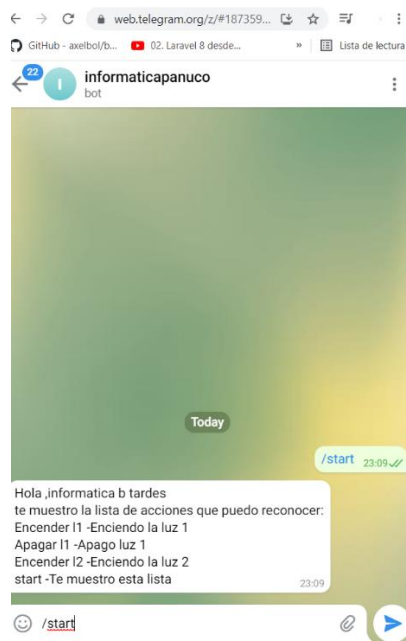


Figura 7. InformaticaBot.

Comando recibido en el ordenador

```
PS C:\python27> python conexiontelegram.py
Bot
Esperando acciones...
Comando obtenido: /start
```

Figura 8. Comando recibido.

El comando [Encender I1] simula la instrucción de encender las luces de un sector, como respuesta obtiene un mensaje confirmando la tarea.



Figura 9. Interacción entre comandos.

En las secciones antes mencionadas se marcaron acciones básicas para la puesta en

marcha de un asistente virtual.

RESULTADOS

El funcionamiento de la comunicación entre las aplicaciones fue exitoso, esto de acuerdo a que se reciben correctamente los datos en el script programado y este a su vez también envía satisfactoriamente las respuestas al bot. Para el desarrollo de esta aplicación se utilizaron herramientas de código abierto siendo Python un lenguaje muy dinámico para la operación de apis.

DISCUSIÓN

En este estudio se presenta la programación de un bot programado en Python, cuya finalidad es generar un método de atención a tareas de manera automática.

Las rutinas que se presentan efectúan una comunicación a través del api de Telegram, programando comandos con funciones personalizadas que pueden ser trasladadas al ámbito de base de datos, Machine Learnig e Internet de las Cosas.

Estos métodos en comparación con los asistentes que crean bots son más personalizables debido a que se tiene el control total de la programación y las herramientas automáticas están cerradas a ciertas tareas. Por citar un ejemplo: con Python podemos programar el puerto serial para que interactúe con dispositivos de IOT y controlarlos a través del bot.

CONCLUSIONES

Con el desarrollo de estos asistentes, se pueden cubrir muchas necesidades de las organizaciones debido a que pueden automatizar algunos de sus procesos sin utilizar recurso humano para operar tareas, todas las acciones las efectuará un programa informático. Este trabajo sienta las bases para que programadores, estudiantes e investigadores puedan usar las rutinas mencionadas, para expandir sus funcionalidades en distintos campos como : Finanzas, Ventas e Inteligencia Artificial.

LITERATURA CITADA

- Estela Quintana, R. & Huerta Barzola, J. (2018). *Chatbot para consultas sobre trámites administrativos en la Municipalidad de Surco*. Universidad Tecnológica del Perú. Recuperado en septiembre 10, 2021. Disponible en: https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/1807/Ronald%20Estela_Juan%20Huerta_Trabajo%20de%20Investigaci%C3%B3n_Bachiller_2018.pdf?sequence=1&isAllowed=y
- Serevance Charles (2015). *Python Para Todos*: Universidad de Michigan
- Telegram Bot. (2021). *Telegram Apis*. Recuperado en Agosto 08, 2021. Disponible en: <https://core.telegram.org/>
- Python Framework (2021). *Python framework for Telegram Bot API*. Recuperado en Agosto 12, 2021. Disponible en: <https://telepot.readthedocs.io/en/latest/>
- Quality Progress (2017). *Chatbot Chatter: More businesses exploring how chatbots can help save money, boost customer service*. Recuperado en septiembre 02, 2021. Disponible en: <http://web.a.ebscohost.com/ehost/detail/detail?vid=4&sid=b50f8a01-727f-4ac9-ac49-724f561c797a%40sessionmgr4007&bdata=JnNpdGU9ZWwhvc3QtbGl2ZQ%3d3d#db=egs&AN=123949087>.
- Woodford, E. (2020, Octubre 03). *Chatbots Vendor Opportunities & Market Forecasts 2020-2024*. Recuperado en septiembre 08, 2021. Juniper Research. Disponible en: <https://www.juniperresearch.com/resources/analytsexpress/july-2017/chatbot-conversations-to-deliver-8bn-cost-saving>.
- Zamora, H. (2019). *Chatbots en Latinoamérica, la clave para la transformación digital*. Recuperado en septiembre 10, 2021 del sitio web: <https://www.fogatagroup.com/noticias/2019/04/20/chatbots-en-latinoamerica-la-clave-para-la-transformacion-digital/>
- Redacción A21. (2020). *Atiende Aeroméxico 1 millón de clientes con chatbot*. Recuperado en septiembre 10, 2021 del sitio web: <https://a21.com.mx/aerolineas/2020/12/02/atiende-aeromexico-1-millon-de-clientes-con-chatbot>