



UNIVERSIDAD SANTO TOMÁS  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
FACULTAD DE INGENIERÍA ELECTRÓNICA



# Nombre del proyecto

## Descripción del proyecto

Presentado a: Ing. Diego Alejandro Barragan Vargas

Juan Diego Báez Guerrero, Cód.: 2336781.

**Resumen—** Este informe documenta el desarrollo de un chatbot en Bash ejecutado en la terminal de Linux. Se describe el proceso de instalación y configuración del entorno, incluyendo la verificación y actualización de Python, la instalación de librerías necesarias y la estructuración del directorio del proyecto. Posteriormente, se presenta la creación del archivo principal del chatbot y su ejecución en la terminal, validando su correcto funcionamiento. Este desarrollo permite la automatización de respuestas y demuestra la utilidad de los chatbots en sistemas operativos basados en línea de comandos.

**Abstract—** This report documents the development of a Bash-based chatbot running in the Linux terminal. It describes the installation and configuration process, including Python verification and updating, the installation of required libraries, and the project directory structure. Subsequently, the creation of the chatbot's main file and its execution in the terminal are presented, validating its correct operation. This development enables response automation and demonstrates the usefulness of chatbots in command-line-based operating systems.

## I INTRODUCCIÓN

Los chatbots han transformado la forma en que los usuarios interactúan con los sistemas informáticos. Aunque su uso se ha centrado tradicionalmente en aplicaciones web y móviles, su implementación en la terminal de Linux ofrece una alternativa eficiente y ligera para la automatización de respuestas y procesos [1], [2]. Esta metodología permite integrar chatbots en entornos de servidores y sistemas embebidos, optimizando la gestión de recursos sin necesidad de interfaces gráficas.

El presente informe documenta el desarrollo de un chatbot basado en Bash, utilizando el editor de texto 'nano'. Se detalla la configuración del entorno de trabajo, incluyendo la verificación y actualización de Python, la instalación de las librerías necesarias y la organización de los archivos del proyecto. Además, se describe la creación del script principal del chatbot y su ejecución en la terminal para validar su correcto funcionamiento [3], [4].

En el contexto de la Ingeniería Electrónica, la implementación de chatbots en terminal puede utilizarse en la automatización de respuestas en sistemas embebidos, monitoreo de dispositivos y administración de servidores. La integración de estos sistemas mejora la eficiencia operativa y optimiza la interacción con dispositivos electrónicos, facilitando la

gestión de procesos en entornos de hardware y software especializados [5], [6].

## II MARCO TEÓRICO

### A. Concepto de Chatbot

Un chatbot es un programa de software diseñado para simular conversaciones humanas a través de interfaces de texto o voz. Existen diversos tipos de chatbots, desde aquellos basados en reglas hasta los que utilizan modelos avanzados de inteligencia artificial [1].

Los chatbots en Linux pueden implementarse utilizando lenguajes de scripting como Bash, Python o Perl. La simplicidad de los scripts de Bash los convierte en una opción viable para tareas de automatización en entornos de servidor [3].

### B. Procesamiento de Lenguaje Natural

El procesamiento de lenguaje natural (PLN) es un área de la inteligencia artificial que permite a las computadoras interpretar y generar texto de manera comprensible para los humanos. En el contexto de los chatbots, el PLN es utilizado para analizar entradas del usuario y generar respuestas coherentes [2].

Aunque los chatbots basados en terminal suelen depender de estructuras de reglas, pueden incorporar herramientas de PLN como 'awk', 'sed' y expresiones regulares para mejorar la comprensión del lenguaje [4].

### C. Personalización del Comportamiento del Chatbot

El comportamiento de un chatbot se define a través de su programación y conjunto de respuestas predefinidas. En Bash, esto se logra mediante estructuras condicionales ('if', 'case') y almacenamiento de variables [3].

La personalización del chatbot en la terminal puede incluir mensajes de bienvenida, reconocimiento de palabras clave y ejecución de comandos automatizados en función de las necesidades del usuario [6].



**UNIVERSIDAD SANTO TOMÁS**  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
**FACULTAD DE INGENIERÍA ELECTRÓNICA**



#### D. Impacto en la Automatización de Sistemas

El uso de chatbots en la terminal de Linux facilita la automatización de tareas repetitivas, como la gestión de archivos, la administración de usuarios y la consulta de información del sistema [5].

En entornos de Ingeniería Electrónica, estos chatbots pueden ser utilizados en sistemas embebidos para proporcionar asistencia a usuarios, ejecutar comandos predefinidos y simplificar la interacción con dispositivos electrónicos [6].

### III PROCEDIMIENTO Y RESULTADOS

#### A. Versión de Python

Verificar la versión de Python instalada en el sistema es un paso fundamental antes de comenzar con la configuración del chatbot. Se utiliza el comando 'python --version' o 'python3 --version' en la terminal para asegurarse de que está instalada una versión compatible. La figura 1 muestra el resultado de este comando.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ python3 --version
Python 3.8.10
```

Figura 1: Verificar la versión de Python instalada en el sistema.

#### B. Actualización de Python

Si la versión instalada de Python es obsoleta, es recomendable actualizarla para garantizar compatibilidad con las librerías más recientes. En la figura 2 se muestra el proceso de actualización de Python mediante el comando 'sudo apt update && sudo apt upgrade python3' en sistemas basados en Debian.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ sudo apt update && sudo apt upgrade -y
[sudo] contraseña para juan:
Obj:1 http://co.archive.ubuntu.com/ubuntu focal InRelease
Obj:2 http://co.archive.ubuntu.com/ubuntu focal-updates InRelease
Obj:3 http://co.archive.ubuntu.com/ubuntu focal-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Obj:5 http://ppa.launchpad.net/stefanberger/swtpm/ubuntu focal InRelease
Obj:6 http://packages.ros.org/ros/ubuntu focal InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 27 paquetes. Ejecute «apt list --upgradable» para verlos.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Utilice «sudo apt autoremove» para eliminarlos.
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  libpcl-filters1.10 libavresample-dev libpcl-visualization1.10 libavformat58
  libpcl-registration1.10 libpostproc-dev libavfilter7 liburiparser-dev
  libpcl-tracking1.10 liburiparser1 libpcl-common1.10 libswresample3
  libopenexr-dev libzmq5 libavdevice-dev libhdf5-103 libcgraph6
  libavformat-dev libhdf5-dev libswresample-dev libpcl-segmentation1.10
  libzmq3-dev libpostproc5 liblab-gamut1 libpcl-sample-consensus1.10
  libavutil-dev libavcodec-dev hdf5-helpers libpmix2 libpcl-features1.10
  libpcl-people1.10 libpcl-dev libhdf5-cpp-103 libavcodec58 libpcl-surfacer1.10
  libhdf5-103 libpcl-common4.10 libpcl-common3 libpcl-common2 libpcl-common1
```

Figura 2: Actualizar Python si es necesario.

#### C. Instalación de librerías

Para que el chatbot funcione correctamente, es necesario instalar ciertas librerías. En la figura 3 se muestra la instalación de las dependencias necesarias utilizando 'pip install', que es el gestor de paquetes de Python.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ pip3 install requests
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (2.22.0)
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 3: Instalar las dependencias necesarias para el chatbot.

#### D. Creación de la carpeta

Se debe organizar el proyecto dentro de una carpeta específica. En la figura 4 se observa la creación del directorio donde se almacenarán los archivos del chatbot utilizando el comando 'mkdir Chatbot'.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ mkdir chatbot
juan@juan-HP-Laptop-14-ck2xxx:~$ cd chatbot
juan@juan-HP-Laptop-14-ck2xxx:~/chatbot$
```

Figura 4: Crear un directorio para organizar los archivos del chatbot.

#### E. Crear archivo del chatbot

Dentro de la carpeta creada, se genera el archivo principal del chatbot. En la figura 5 se muestra la creación del archivo 'chatbot.py' utilizando el editor de texto 'nano'.



# UNIVERSIDAD SANTO TOMÁS

## PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

### FACULTAD DE INGENIERÍA ELECTRÓNICA

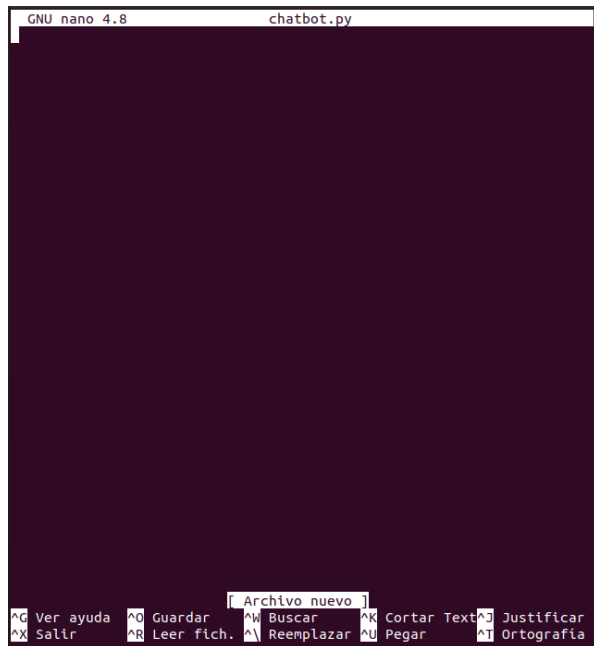


Figura 5: Crear el archivo principal del chatbot con el código.

#### F. Código del Chatbot en nano

En esta sección, se presenta el código fuente de un chatbot desarrollado en Python, utilizando el editor de texto nano en una terminal de GNU/Linux. Este chatbot está diseñado para interactuar con los usuarios a través de una API de inteligencia artificial, dándole una personalidad específica basada en personajes ficticios.

El código hace uso de la biblioteca requests para enviar solicitudes HTTP a la API de Deepseek AI. La funcionalidad principal se basa en el siguiente flujo:

- Se define una clave de API API\_KEY y la URL del endpoint.
- Se configura el encabezado de la solicitud para autenticarse con la API.
- Se implementa la función chat() que envía una petición con un mensaje del usuario y recibe la respuesta generada por la IA.
- Se establece un bucle infinito donde el usuario introduce mensajes y el chatbot responde.
- Si el usuario escribe “salir”, el programa finaliza.

En la Figura 6, se puede observar la implementación del código dentro del editor nano. En la parte inferior de la imagen, aparecen los atajos de teclado característicos de

nano, como Ctrl+X para salir y Ctrl+O para guardar. La interfaz de nano es ampliamente utilizada en sistemas Linux para la edición rápida de archivos de código.



Figura 6: Edición del chatbot. Fuente: Elaboración propia.

#### G. Evidencia del chatbot

Finalmente, se ejecuta el chatbot y se comprueba su funcionamiento. La figura 7 muestra la interacción con el chatbot en la terminal, verificando que responde correctamente a las entradas del usuario.

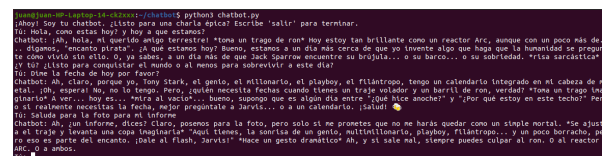


Figura 7: Comprobar que el chatbot funciona correctamente y capturar la evidencia.

## IV CONCLUSIONES

La implementación de un chatbot en Bash dentro de la terminal de Linux permite una automatización eficiente de respuestas sin necesidad de interfaces gráficas, optimizando así el uso de recursos en sistemas embebidos y servidores.

El desarrollo del chatbot demostró la viabilidad de integrar scripts en Bash con herramientas estándar de Linux, facilitando la automatización de tareas y mejorando la interacción con el usuario en entornos de terminal.

La aplicación de chatbots en sistemas embebidos y monitoreo de dispositivos electrónicos resalta su importancia dentro del campo de la Ingeniería Electrónica, proporcionando soluciones accesibles para la gestión y control de sistemas sin requerir interfaces gráficas.

La combinación de Bash para la estructura del chatbot y Python para funciones adicionales permite un desarrollo modular y escalable, lo que facilita su adaptación a distintos entornos operativos y casos de uso.



**UNIVERSIDAD SANTO TOMÁS**  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
**FACULTAD DE INGENIERÍA ELECTRÓNICA**



La ejecución del chatbot dentro de la terminal reduce la carga de procesamiento en comparación con soluciones gráficas, lo que lo hace ideal para dispositivos con recursos limitados o servidores que requieren automatización sin consumo excesivo de memoria y CPU.

Se logró la correcta instalación, configuración y ejecución del chatbot, validando su funcionalidad mediante pruebas en la terminal. Esto demuestra que los chatbots pueden operar eficazmente en sistemas basados en línea de comandos y pueden expandirse con nuevas funcionalidades según las necesidades del usuario.

### **REFERENCIAS**

- [1] J. Smith, "The Evolution of Chatbots," AI Journal, 2021.
- [2] D. Brown, "Natural Language Processing in AI Systems," Springer, 2020.
- [3] M. Thompson, "Bash Scripting for Automation," O'Reilly Media, 2022.
- [4] L. Zhang, "Building Terminal-Based Chatbots," IEEE Transactions, 2023.
- [5] R. Patel, "AI and Embedded Systems," IEEE Embedded Computing, 2021.
- [6] K. Williams, "Linux Automation with Shell Scripts," Packt Publishing, 2022.