

Documentación -Proyecto Final

Programación III

Luisa Maria Reyes

LINK VIDEO TUTORIAL

https://www.canva.com/design/DAGXrkvHpvE/r3eO1Jlf9URz8bnr4jBJig/edit?utm_content=DAGXrkvHpvE&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Este documento explica el proceso completo para crear un chatbot en Telegram utilizando **n8n**. Incluye los pasos para instalar y configurar las herramientas necesarias, así como la implementación del flujo en n8n.

1. Objetivos del proyecto

El chatbot responderá mensajes enviados por usuarios en Telegram y podrá realizar consultas dinámicas como:

- Buscar información en fuentes externas.
 - Proporcionar respuestas personalizadas utilizando inteligencia artificial.
 - Recuperar información de archivos locales, si es necesario.
-

2. Herramientas necesarias

1. **n8n**: Herramienta de automatización para crear flujos de trabajo.
 2. **Docker**: Plataforma para contenerizar y desplegar n8n.
 3. **ngrok**: Para exponer n8n al público y recibir webhooks de Telegram.
 4. **Bot de Telegram**: Creado a través de BotFather.
-

3. Pasos detallados

Paso 1: Crear un bot en Telegram

1. **Abrir BotFather**:
 - En la aplicación de Telegram, busca a "BotFather" y comienza un chat.
2. **Crear un nuevo bot**:

- Escribe el comando `/newbot`.
 - BotFather pedirá un nombre para tu bot (visible para los usuarios) y un nombre de usuario único (debe terminar en "bot").
3. **Obtener el token del bot:**
- Una vez creado, BotFather proporcionará un token (una cadena larga de caracteres). Este token es necesario para conectar tu bot con n8n.
 - **Guárdalo en un lugar seguro.**
-

Paso 2: Instalar n8n

A. Usando Docker

1. **Prepara el archivo `docker-compose.yml`:**
 - Crea un archivo `docker-compose.yml` en tu máquina. Este archivo definirá cómo se ejecutará n8n en Docker.
2. **Ejecuta n8n:**

Abre una terminal en la carpeta donde se encuentra el archivo `docker-compose.yml` y ejecuta:

Copiar código

```
docker-compose up -d
```

-
- Esto iniciará n8n en el puerto `5678`. Accede a la interfaz web de n8n en <http://localhost:5678>.

B. Instalación sin Docker (opcional)

1. Si prefieres no usar Docker, puedes instalar n8n con Node.js:
 - Instala Node.js y npm.

Ejecuta:

Copiar código

```
npm install -g n8n
```

-

Luego, inicia n8n con:

Copiar código

```
n8n
```

-
-

Paso 3: Configurar ngrok

1. **Descargar e instalar ngrok:**

- Descarga ngrok desde su [sitio oficial](#) y sigue las instrucciones de instalación.

2. **Exponer n8n al público:**

Ejecuta ngrok en el puerto **5678** para que n8n sea accesible desde una URL pública:

yaml

Copiar código

```
ngrok http 5678
```

- - Copia la **URL HTTPS** que genera ngrok (por ejemplo, <https://abc123.ngrok.io>).
-

Paso 4: Configurar el webhook de Telegram

1. **Establecer el webhook:**

Abre un navegador y pega la siguiente URL:

bash

Copiar código

```
https://api.telegram.org/bot<TU_BOT_TOKEN>/setWebhook?url=<NGROK_URL>/webhook/telegram
```

- - Reemplaza **<TU_BOT_TOKEN>** con el token de tu bot.
 - Reemplaza **<NGROK_URL>** con la URL generada por ngrok.

2. **Verificar el webhook:**

Abre otra URL para confirmar que el webhook está configurado correctamente:

bash

Copiar código

```
https://api.telegram.org/bot<TU_BOT_TOKEN>/getWebhookInfo
```

-
-

Paso 5: Crear el flujo en n8n

A. Nodo Telegram Trigger

1. **Añadir Telegram Trigger:**

- Este nodo capturará los mensajes enviados al bot.
- Configura las credenciales del bot (token).
- Selecciona el tipo de actualización (**message**).

B. Nodo de lógica (If/Switch)

1. **Añadir un nodo If:**
 - Úsalo para manejar diferentes tipos de mensajes (comandos específicos o consultas generales).
 - Por ejemplo:
 - Si el mensaje es `/start`, responde con un mensaje de bienvenida.
 - Si es otro mensaje, envíalo a procesamiento adicional.

C. Nodo de integración (opcional)

1. **Añadir herramientas externas:**
 - **SerpAPI:** Busca información en la web.
 - **AI Agent (OpenAI):** Genera respuestas personalizadas usando inteligencia artificial.

D. Nodo Telegram Send Message

1. **Enviar respuestas al usuario:**
 - Configura este nodo para enviar las respuestas generadas por el flujo.
-

Paso 6: Probar el flujo

1. Activa el flujo en n8n.
 2. Envía mensajes al bot de Telegram.
 3. Verifica que el flujo procese los mensajes y envíe respuestas automáticamente.
-

Paso 7: Despliegue y mantenimiento

1. **Mantener n8n corriendo:**
 - Usa Docker para asegurar que n8n esté siempre disponible.
 2. **Actualizar ngrok (si es necesario):**
 - La URL de ngrok cambia cada vez que lo reinicias. Actualiza el webhook de Telegram si esto ocurre.
-

8. Resumen de nodos usados en n8n

1. **Telegram Trigger:** Captura mensajes enviados al bot.
2. **If/Switch:** Maneja diferentes tipos de mensajes.
3. **AI Agent/SerpAPI (opcional):** Procesa las consultas del usuario.
4. **Telegram Send Message:** Responde al usuario.