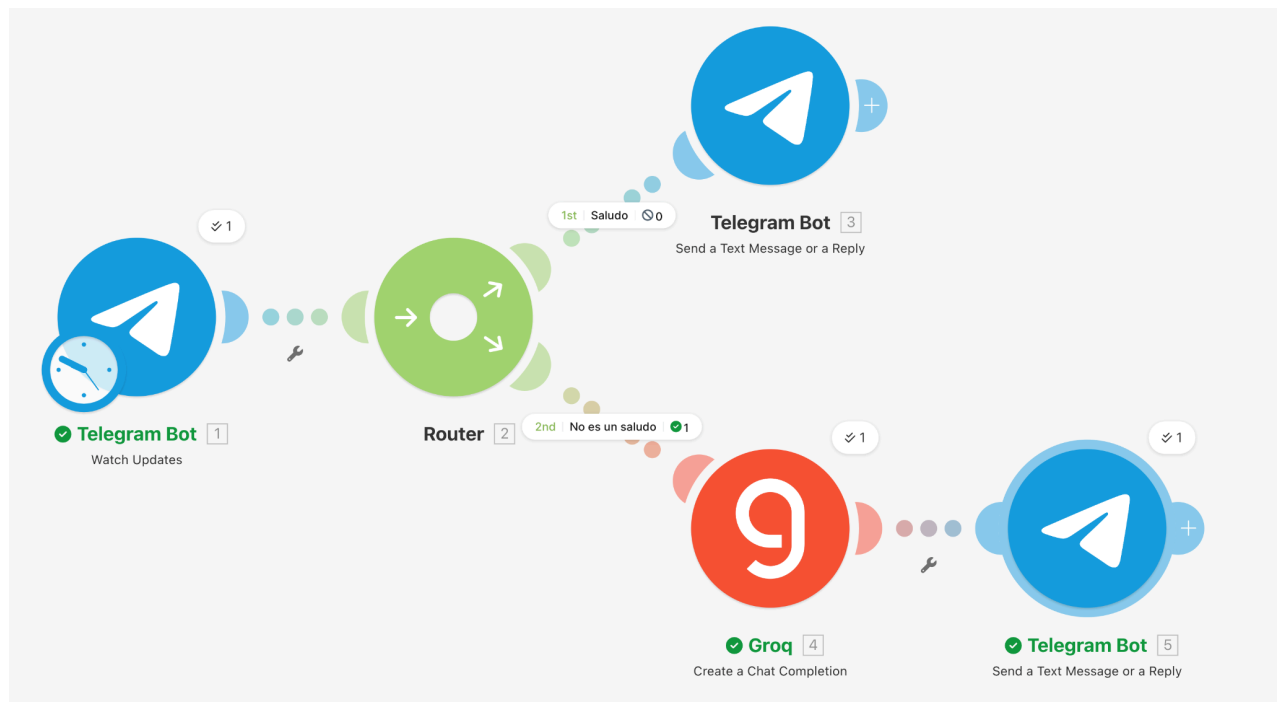


# Guía paso a paso: crear un bot de Telegram conectado a Make.com con Groq

## Objetivo

Esta guía explica cómo crear un bot de Telegram que reciba mensajes mediante webhook en Make, los enrute con un Router y responda de dos formas: una respuesta fija para un saludo y una respuesta generada por Groq para cualquier otro mensaje.[cite:11][cite:18]

## Flujo que se va a construir



Según el escenario mostrado, el flujo en Make usa estos módulos y esta lógica:[image:1]

1. **Telegram Bot — Watch Updates** para recibir los mensajes entrantes del bot en tiempo real.[cite:11][cite:18]
2. **Router** para separar los mensajes en dos caminos según una condición de filtro.[cite:7]
3. **Telegram Bot — Send a Text Message or a Reply** para contestar si el usuario escribe un saludo.[cite:11]
4. **Groq — Create a Chat Completion** para generar una respuesta si no es un saludo.[cite:11]

5. **Telegram Bot — Send a Text Message or a Reply** para devolver al usuario la respuesta generada por Groq.[cite:11]

## Qué hace cada módulo

| Orden | Módulo  | Función   |
|-------|---|---|
| 1     | Telegram Bot - Watch Updates                  | Recibe el update del usuario mediante el webhook gestionado por Make.[cite:11][cite:18] |
| 2     | Router  | Decide por qué rama seguirá el escenario según filtros definidos por ti.[cite:7]        |
| 3     | Telegram Bot - Send a Text Message or a Reply | Envía una respuesta corta cuando detectas un saludo.[cite:11]                           |
| 4     | Groq - Create a Chat Completion               | Genera una respuesta con IA para mensajes que no sean saludo.[cite:11]                  |
| 5     | Telegram Bot - Send a Text Message or a Reply | Devuelve al chat la respuesta creada por Groq.[cite:11]                                 |

## Paso 1: crear el bot en Telegram

1. Abre Telegram y busca @BotFather, que es el asistente oficial para crear bots.[cite:15]
2. Envía el comando /newbot.[cite:9][cite:12]
3. Escribe un nombre para el bot y después un nombre de usuario terminado en bot.[cite:12]
4. Copia el token que devuelve BotFather, porque Make lo pedirá al conectar Telegram.[cite:11][cite:12]

## Paso 2: crear el escenario en Make

1. Crea un escenario nuevo en Make.[cite:7]
2. Añade el módulo **Telegram Bot > Watch Updates** como disparador inicial.[cite:11]
3. Crea la conexión pegando el token del bot.[cite:11]
4. Guarda la configuración del módulo para que Make quede listo para recibir updates del bot.[cite:11]

### **Paso 3: cómo entra el webhook**

Telegram permite entregar actualizaciones a una URL HTTPS usando el método `setWebhook`.<sup>[cite:16]</sup><sup>[cite:18]</sup>

Cuando usas **Watch Updates** en Make, esa recepción se resuelve a través de la integración de Make, por lo que el escenario actúa como receptor del webhook sin que tengas que montar tu propio servidor.<sup>[cite:11]</sup><sup>[cite:24]</sup>

### **Paso 4: añadir el Router**

Después de **Watch Updates**, añade un **Router** para dividir el flujo en dos ramas.<sup>[cite:7]</sup>

En la imagen se ve una rama etiquetada como Saludo y otra como No es un saludo, así que la lógica del escenario depende de filtros colocados en cada salida del Router.<sup>[image:1]</sup>

### **Paso 5: configurar la rama de saludo**

En la primera rama, añade el módulo **Telegram Bot > Send a Text Message or a Reply**.<sup>[cite:11]</sup>

Esta rama debe tener un filtro que detecte mensajes como hola, buenas o hello, y cuando se cumpla, el bot enviará una respuesta fija como “Hola, ¿en qué puedo ayudarte?”.<sup>[image:1]</sup>

### **Ejemplo de filtro para saludo**

Puedes crear un filtro con una condición basada en el texto entrante del mensaje, por ejemplo comprobando si el texto contiene hola o coincide con un listado de saludos.<sup>[image:1]</sup>

La idea es que esta rama responda rápido sin pasar por Groq cuando el mensaje solo sea una apertura de conversación.<sup>[image:1]</sup>

### **Paso 6: configurar la rama que usa Groq**

En la segunda rama, añade **Groq > Create a Chat Completion**.<sup>[image:1]</sup>


Esta rama debe ejecutarse solo cuando el filtro indique que el mensaje no es un saludo, tal como aparece en la rama No es un saludo del escenario.<sup>[image:1]</sup>

### **Paso 7: pasar el mensaje del usuario a Groq**

Dentro del módulo de Groq, usa como entrada el texto recibido desde Telegram para que el modelo genere una respuesta.<sup>[image:1]</sup>

Lo habitual es mapear el campo de texto del mensaje del usuario dentro del prompt o del array de mensajes del módulo de chat completion.<sup>[image:1]</sup>

## Ejemplo de instrucción para Groq

Puedes usar una instrucción del estilo: “Responde en español de forma breve y útil al siguiente mensaje del usuario: {{texto del mensaje}}”.





Así consigues que la salida de Groq ya venga lista para devolverla por Telegram.

## Paso 8: devolver la respuesta generada a Telegram




Después de Groq, añade un segundo módulo **Telegram Bot > Send a Text Message or a Reply**.

En ese módulo, mapea el `chat_id` recibido en el update original y como contenido del mensaje usa el texto generado por Groq.


## Paso 9: campos que normalmente se mapean

| Campo de origen                                     | Se usa en                              | Propósito   |
|---|--|---|
| <code>message.chat.id</code> del update de Telegram | Ambos módulos de respuesta de Telegram | Indica a qué chat hay que contestar.   |
| <code>message.text</code> del update de Telegram    | Router y módulo de Groq                | Permite evaluar el filtro y generar la respuesta.   |
| Salida textual de Groq                              | Último módulo de Telegram              | Se envía al usuario como respuesta final.    |

## Paso 10: prueba del escenario

1. Ejecuta el escenario con **Run once** en Make.
2. Envía a tu bot un mensaje tipo `hola` y comprueba que entra por la rama de saludo.
3. Envía después una pregunta normal y comprueba que pasa por Groq y vuelve la respuesta al chat.

## Estructura final del escenario

La estructura final queda así:

- Telegram Bot — Watch Updates.
- Router.
- Rama 1: filtro Saludo → Telegram Bot — Send a Text Message or a Reply.

- Rama 2: filtro No es un saludo → Groq — Create a Chat Completion → Telegram Bot — Send a Text Message or a Reply.

## Recomendaciones de configuración

- Usa respuestas cortas en la rama de saludo para no gastar llamadas innecesarias a IA.[image:1]
- Añade instrucciones claras en Groq para controlar tono, idioma y longitud de la respuesta.[image:1]
- Si el bot solo debe responder a ciertos usuarios o grupos, añade filtros extra en el Router o antes de Groq.[cite:7][cite:18]

## Problemas habituales

| Problema   | Causa probable  | Solución  |
|--|---|---|
| Telegram recibe mensajes pero el escenario no responde | El escenario no está ejecutándose o no está activo.[cite:7]                               | Ejecuta <b>Run once</b> para probar y luego activa el escenario.[cite:7]            |
| La rama de saludo no se ejecuta                        | El filtro no coincide con el texto real del mensaje.[image:1]                             | Ajusta la condición del filtro y prueba variantes como hola, buenas o hey.[image:1] |
| Groq no genera salida útil                             | El prompt es demasiado ambiguo.[image:1]  | Añade una instrucción más concreta sobre idioma, tono y objetivo.[image:1]          |
| Telegram no devuelve la respuesta de Groq              | El último módulo no está mapeando bien el chat_id o el texto de salida.[cite:18][image:1] | Revisa el mapeo del chat y el campo exacto de respuesta generado por Groq.[image:1] |

## Implementación recomendada para esta guía

Para este caso concreto, la mejor forma de explicarlo es como un bot con dos comportamientos: saludo automático y respuesta inteligente.[image:1]

Esa estructura encaja exactamente con el escenario de la imagen y sigue el patrón de recepción instantánea de Telegram en Make mediante webhook.[cite:11][cite:18]