



**TRABAJO DE GRADO  
Opción Seminario-Diplomado.**

**Automatización Comercial para Tiendas de Repuestos de Motos**

**Luis David Bedoya Quintero**

**Proyecto presentado como opción de grado Seminario**

**Corporación Universitaria Remington**

**Facultad de Ingenierías**

**Ingeniería de Sistemas**

**Luis Camargo Ortega**

**2026**

## Agradecimientos

El desarrollo de este proyecto no habría sido posible sin el apoyo de varias personas que estuvieron presentes en cada etapa del proceso.

A mi familia, por la paciencia y el respaldo incondicional durante todo el tiempo que demandó este trabajo, su confianza fue el motor que me impulsó a seguir adelante incluso en los momentos más exigentes.

A mi tutor Luis Camargo Ortega, por la orientación, la disponibilidad y las observaciones que permitieron enfocar mejor las ideas y darle solidez académica a este proyecto.

A la Corporación Universitaria Remington y a los docentes de la Facultad de Ingenierías, por los conocimientos compartidos a lo largo de la carrera, que se ven reflejados directamente en la construcción de esta solución.

A la comunidad de desarrolladores de n8n, Anthropic, Supabase y Telegram, cuya documentación pública y recursos abiertos hicieron posible la integración de todas las tecnologías utilizadas en Motofix.

## Tabla de Contenido

Resumen.....	7
Introducción .....	9
Planteamiento del Problema .....	11
Descripción del Problema .....	11
Pregunta de Investigación.....	12
Justificación .....	12
Objetivos.....	13
Objetivo General.....	13
Objetivos Específicos.....	13
Marco Conceptual y Contextual .....	14
Inteligencia Artificial Conversacional .....	14
Bots de Telegram como Canal de Comunicación .....	14
Arquitectura de Agentes con Herramientas .....	15
Vista General del Flujo en n8n.....	15
Tecnologías Utilizadas .....	16
Descripción Detallada de Nodos y Configuración.....	16
Nodo 1: Telegram Trigger.....	17
Nodo 2: Set (Normalización de Datos).....	17

Nodo 3: AI Agent .....	18
Sub-nodo 3.1: Anthropic Chat Model .....	19
Sub-nodo 3.2: Simple Memory .....	20
Nodo 4: detectar_comprobante (Nodo If) .....	21
Nodo 5: enviar_correo_pedido (Gmail) .....	22
Nodo 6: Send a text message (Telegram) .....	23
Herramientas HTTP del AI Agent .....	24
Herramienta 1: validar_cliente .....	24
Herramienta 2: buscar_repuesto .....	25
Herramienta 3: crear_pedido .....	26
Herramienta 4: guardar_sesion .....	27
Herramienta 5: obtener_sesion .....	28
Herramienta 6: registrar_cliente .....	29
Estructura de la Base de Datos (Supabase) .....	30
Tabla: repuestos .....	31
Tabla: motos .....	31
Tabla: clientes .....	32
Tabla: pedidos .....	32
Tabla: sesiones .....	32
Evidencias de Funcionamiento Conversación Real con el Bot .....	33

Inicio de Conversación y Menú Principal.....	33
Registro de Nuevo Cliente .....	34
Búsqueda de Repuesto por Marca y Modelo .....	35
Selección de Repuesto y Confirmación de Pedido .....	35
Instrucciones de Pago .....	36
Confirmación Final .....	37
Ciclo Completo de una Venta .....	39
Credenciales y APIs Integradas al Proyecto .....	40
Resumen General de Componentes .....	41
Conclusiones .....	42
Código Fuente del Proyecto, Flujo n8n (JSON) .....	43
Referencias.....	62

## Lista de Figuras

Figura 1 Vista general del flujo de automatización MotoFix en n8n.....	16
Figura 2 Configuración del nodo Telegram Trigger .....	17
Figura 3 Configuración del nodo Set con los campos mapeados .....	18
Figura 4 Configuración del nodo AI Agent con System Message y herramientas .....	19
Figura 5 Configuración del sub-nodo Anthropic Chat Model (Claude Haiku 4.5) .....	20
Figura 6 Configuración del sub-nodo Simple Memory con Session ID por chat_id.....	21
Figura 7 Configuración del nodo condicional detectar_comprobante .....	22
Figura 8 Configuración del nodo Gmail enviar_correo_pedido .....	23
Figura 9 Configuración del nodo Send a text message (Telegram) .....	24
Figura 10 Configuración de la herramienta validar_cliente (HTTP GET) .....	25
Figura 11 Configuración de la herramienta buscar_repuesto (HTTP GET) .....	26
Figura 12 Configuración de la herramienta crear_pedido (HTTP POST) .....	27
Figura 13 Configuración de la herramienta guardar_sesion (HTTP POST).....	28
Figura 14 Configuración de la herramienta obtener sesión (HTTP GET) .....	29
Figura 15 Configuración de la herramienta registrar cliente (HTTP POST) .....	30
Figura 16 Schema Visualizer de Supabase, vista completa de las 5 tablas.....	31
Figura 17 Inicio de conversación, el bot presenta el menú principal de MotoFix.....	33
Figura 18 Flujo de registro automático de nuevo cliente.....	34
Figura 19 Consulta de repuestos disponibles por marca y modelo de moto.....	35
Figura 20 Selección de repuesto y confirmación del pedido por el cliente .....	36
Figura 21 Bot entrega instrucciones de pago y solicita comprobante .....	37
Figura 22 Confirmación final del pedido, ciclo de venta completado .....	38
Figura 23 Panel de credenciales en n8n, APIs conectadas al proyecto MotoFix.....	40

## Resumen

El presente documento describe el diseño, desarrollo e implementación de Motofix, un sistema automatizado de atención al cliente y gestión de ventas para repuestos de motos, el sistema opera a través de un bot de mensajería en la plataforma Telegram, potenciado por un modelo de inteligencia artificial de Anthropic Claude, integrado con una base de datos relacional PostgreSQL en la nube alojado en la plataforma Supabase mediante flujos de automatización construidos sobre la plataforma N8N.

El problema central que aborda este proyecto es la ineficiencia operativa que enfrentan los negocios de venta de repuestos para motos al gestionar las consultas que realizan los clientes, verifica la disponibilidad de inventario y procesar pedidos de manera manual, esta situación genera demoras, errores humanos y una experiencia poco eficiente para el usuario final.

Como solución, el sistema de automatización comercial con agente de IA para tiendas de repuestos de motos, implementa un agente conversacional inteligente capaz de validar y registrar clientes de forma automática, consultar en tiempo real el inventario de repuestos, generar pedidos de compra, recibir y verificar comprobantes de pago y enviar notificaciones por correo electrónico al personal de la tienda, todo esto ocurre de manera autónoma, sin intervención humana, disponible las 24 horas del día.

La arquitectura del sistema combina herramientas HTTP personalizadas que conectan el agente de IA con el backend, una base de datos con 5 tablas relacionales (repuestos, motos, clientes, pedidos y sesiones) y un flujo de ocho nodos principales en N8N que orquesta el proceso completo desde la recepción del mensaje hasta la confirmación del pedido.

Los resultados demuestran que la automatización del proceso de ventas mediante inteligencia artificial conversacional reduce significativamente los tiempos de respuesta al cliente, elimina errores de registro manual y permite escalar la operación sin incrementar el personal, Motofix representa un modelo replicable para otros negocios del sector de autopartes y servicios similares.

**Palabras clave:** Automatización, inteligencia artificial conversacional, bot de Telegram, n8n, Supabase, gestión de ventas, repuestos para motos, Anthropic Claude, agente de IA, flujo de trabajo automatizado.

## Introducción

En el contexto actual de transformación digital, los negocios de todos los sectores enfrentan la presión de adaptarse a nuevas formas de interacción con sus clientes, el comercio de repuestos para motos, siendo un sector con alta demanda de atención personalizada y consultas técnicas frecuentes, no es la excepción, la necesidad de brindar respuestas rápidas, precisas y disponibles en cualquier horario representa un reto que los modelos tradicionales de atención al cliente difícilmente pueden cubrir de manera sostenible.

El avance de los modelos de lenguaje de gran escala y las plataformas de automatización de flujos de trabajo han abierto nuevas posibilidades para la creación de sistemas inteligentes capaces de simular conversaciones humanas y ejecutar procesos de negocio de forma autónoma, herramientas como Anthropic Claude, N8N, Supabase y Telegram ofrecen hoy en día APIs robustas, bien documentadas y accesibles que permiten construir soluciones de este tipo con recursos limitados.

Este proyecto surge como respuesta a esa oportunidad, desarrollar un asistente virtual inteligente que automatice completamente el flujo de ventas de un negocio de repuestos para motos, desde el primer contacto con el cliente hasta la confirmación del pedido, el sistema utiliza Telegram como canal de comunicación principal, dado que es una de las aplicaciones de mensajería más populares en el contexto latinoamericano y ofrece una API de bots madura y gratuita.

La inteligencia del sistema reside en el AI Agent de N8N, configurado con el modelo de Claude Anthropic, que actúa como núcleo de razonamiento, interpreta los mensajes del usuario en lenguaje natural, decide que acciones ejecutar y coordina las llamadas a los distintos servicios

del backend, la persistencia de datos se gestiona en Supabase, una plataforma de bases de datos PostgreSQL en la nube que expone los datos a través de endpoints REST, facilitando la integración con el agente de IA.

El presente documento tiene como objetivo describir de manera detallada y estructurada todos los componentes del sistema, la arquitectura del flujo de automatización, la configuración de cada nodo, las herramientas disponibles para el agente, la estructura de la base de datos y el ciclo completo de interacción con el usuario final, esta documentación sirve como referencia técnica para la comprensión, mantenimiento y futura escalabilidad del proyecto.

## Planteamiento del Problema

### Descripción del Problema

Los negocios de venta de repuestos para motos, especialmente aquellos de tamaño pequeño y mediano, dependen en gran medida de la atención manual para gestionar consultas de clientes y procesar pedidos, este modelo de operación presenta limitaciones críticas que afectan directamente la competitividad y la experiencia del usuario.

La disponibilidad limitada es una de las restricciones más notorias, pues la atención al cliente está delimitada al horario laboral del personal, lo que genera pérdida de oportunidades de venta en franjas horarias no cubiertas y frustración en clientes que buscan información en cualquier momento del día, a esto se suma el alto volumen de consultas repetitivas, ya que una gran proporción de las interacciones recibidas son preguntas recurrentes sobre disponibilidad de repuestos, precios y compatibilidad con modelos de motos específicos, atender estas consultas de manera manual consume tiempo del personal que podría dedicarse a tareas de mayor valor agregado.

Por otra parte, el registro manual de pedidos, datos de clientes y comprobantes de pago es propenso a errores humanos, duplicados y omisiones que generan problemas operativos y reclamos posteriores, y sin un sistema centralizado resulta difícil hacer seguimiento del estado de los pedidos, el historial de compras de los clientes o el stock disponible en tiempo real, finalmente, a medida que el negocio crece, el volumen de consultas y pedidos aumenta proporcionalmente, lo que obliga a incrementar el personal de atención y eleva los costos operativos de forma directa.

**Pregunta de Investigación**

¿Cómo es posible automatizar el proceso completo de atención al cliente y gestión de ventas de repuestos para motos, garantizando disponibilidad continua, precisión en el registro de datos y una experiencia conversacional natural para el usuario?

**Justificación**

La implementación de un sistema de atención automatizada mediante inteligencia artificial no solo resuelve las limitaciones operativas explicadas, sino que posiciona al negocio en la vanguardia tecnológica del sector, la combinación de herramientas de automatización modernas y modelos de lenguaje avanzados permite ofrecer una experiencia de cliente de calidad superior, reducir costos operativos y capturar ventas que de otro modo se perderían.

## Objetivos

### Objetivo General

Desarrollar e implementar un sistema automatizado de atención al cliente y gestión de ventas de repuestos para motos mediante un bot de Telegram con inteligencia artificial.

### Objetivos Específicos

- Diseñar el flujo de automatización en n8n que integre Telegram, Anthropic Claude y Supabase como plataformas centrales del sistema.
- Implementar un agente de IA capaz de gestionar el ciclo completo de ventas de forma autónoma, desde la validación del cliente hasta la confirmación del pedido.
- Construir una base de datos relacional que soporte las entidades del sistema: clientes, repuestos, motos, pedidos y sesiones.
- Desarrollar los endpoints REST necesarios para la comunicación entre el agente y la base de datos alojada en Supabase.
- Validar el sistema mediante pruebas de flujo completo de compra en entorno real.

## **Marco Conceptual y Contextual**

### **Inteligencia Artificial Conversacional**

La inteligencia artificial conversacional es la tecnología que permite que una máquina mantenga una conversación con una persona, entendiéndola y respondiendo de forma coherente, detrás de esto están los modelos de lenguaje de gran escala (LLM), sistemas entrenados con grandes cantidades de texto que les permiten aprender cómo se expresa el ser humano, para este proyecto decidí usar Claude de Anthropic, un modelo que no solo responde preguntas sino que también ejecuta acciones dentro del sistema según lo que el usuario le pida, lo cual fue clave para que Motofix funcionara de forma autónoma (Anthropic, 2026). Bases de Datos como Servicio (Supabase)

Supabase es una plataforma de backend como servicio (BaaS) construida sobre PostgreSQL, que ofrece base de datos relacional, autenticación, almacenamiento de archivos y funciones serverless a través de APIs REST, su modelo de acceso mediante endpoints HTTP lo hace ideal para ser consumido por agentes de IA que necesitan leer y escribir datos de forma programática (Supabase, 2026).

### **Bots de Telegram como Canal de Comunicación**

Telegram es una aplicación de mensajería con cientos de millones de usuarios en todo el mundo, lo que la convierte en un canal ideal para llegar al cliente sin pedirle que instale nada nuevo, además ofrece una API de bots gratuita y fácil de usar que permite crear asistentes automatizados directamente dentro de la app (Telegram, 2026).

## **Arquitectura de Agentes con Herramientas**

Un agente con herramientas es básicamente un modelo de IA que no solo genera texto, sino que también puede ejecutar acciones concretas, como consultar una base de datos o registrar un pedido, el modelo analiza lo que el usuario pide, decide qué acción tomar y usa el resultado para responder, en Motofix esto se traduce en seis herramientas que Claude puede invocar por su cuenta según lo que necesite en cada momento de la conversación. Descripción General del Proyecto

Motofix es un sistema automatizado de atención al cliente y gestión de ventas de repuestos para motos, el sistema opera a través de un bot de Telegram potenciado por Inteligencia Artificial (Anthropic Claude), capaz de gestionar el ciclo completo de ventas: desde la identificación del cliente, búsqueda de repuestos, generación de pedidos, verificación de comprobantes de pago y notificación final.

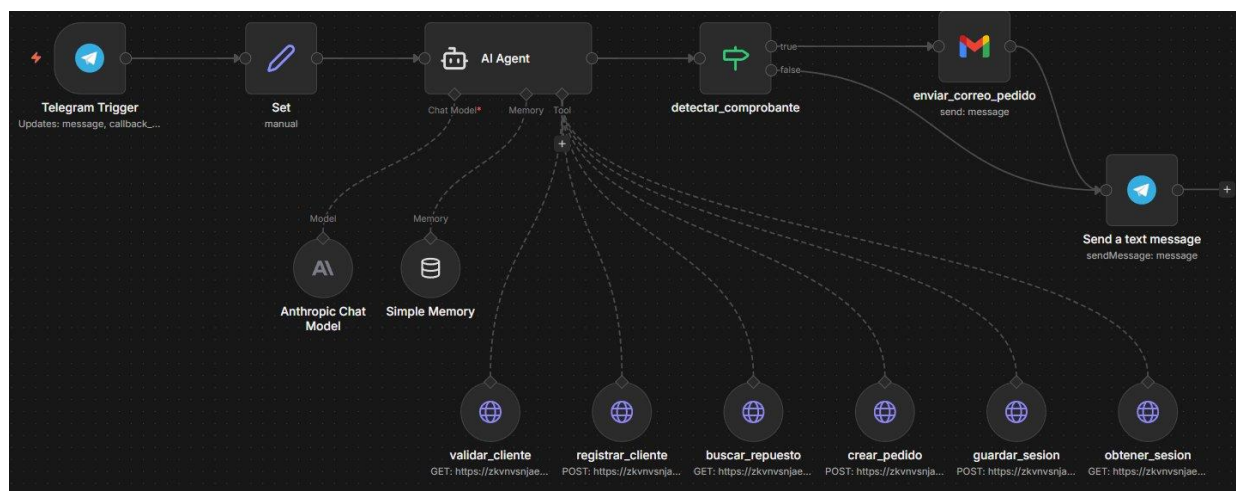
El flujo está construido sobre la plataforma de automatización n8n, integrando múltiples servicios: una base de datos Supabase (PostgreSQL), la API de Claude de Anthropic como motor de inteligencia artificial, Telegram como canal de comunicación con el cliente, y Gmail para el envío de correos de confirmación de pedidos.

### **Vista General del Flujo en n8n**

La siguiente figura muestra la arquitectura completa del flujo de automatización en n8n, con todos sus nodos y conexiones:

**Figura 1**

*Vista general del flujo de automatización MotoFix en n8n*



Fuente: Elaboración propia (2026)

## Tecnologías Utilizadas

El sistema integra seis tecnologías principales que trabajan de forma orquestada, n8n actúa como motor principal de automatización y orquestación del flujo, Anthropic Claude Haiku 4.5 cumple el rol de AI Agent y LLM, Supabase provee el backend con base de datos PostgreSQL en la nube, la API de Telegram Bot funciona como interfaz de mensajería con el usuario final, la API de Gmail gestiona el servicio de correo electrónico para notificaciones, y la comunicación entre n8n y el backend Supabase se realiza a través de HTTP y REST API.

## Descripción Detallada de Nodos y Configuración

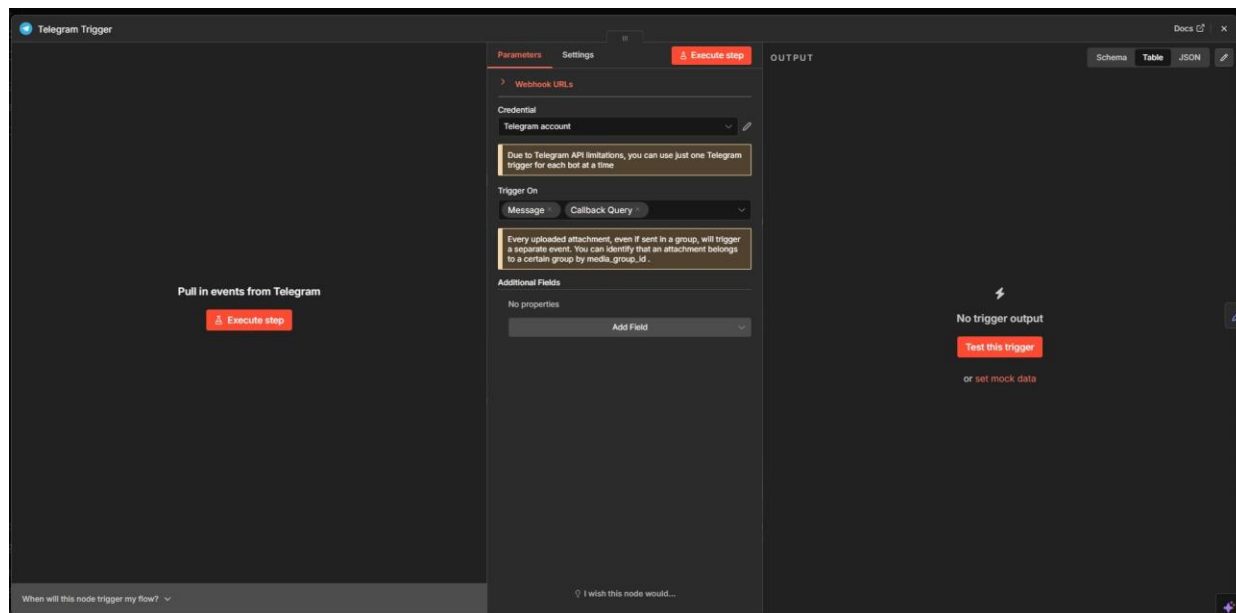
A continuación se describe cada nodo del flujo con su configuración, parámetros y captura de pantalla real de la interfaz de n8n.

### ***Nodo 1: Telegram Trigger***

Este nodo es el punto de entrada del flujo, su función es detectar cada mensaje que llegue al bot de Telegram y pasarlo al resto del sistema, está configurado para escuchar tanto mensajes de texto como respuestas de botones, y genera como salida un objeto con los datos del mensaje recibido, algo importante a tener en cuenta es que Telegram solo permite un trigger activo por bot a la vez, y que cuando el usuario envía una imagen esta llega como un evento separado al del texto.

### **Figura 2**

#### *Configuración del nodo Telegram Trigger*



Fuente: Elaboración propia (2026).

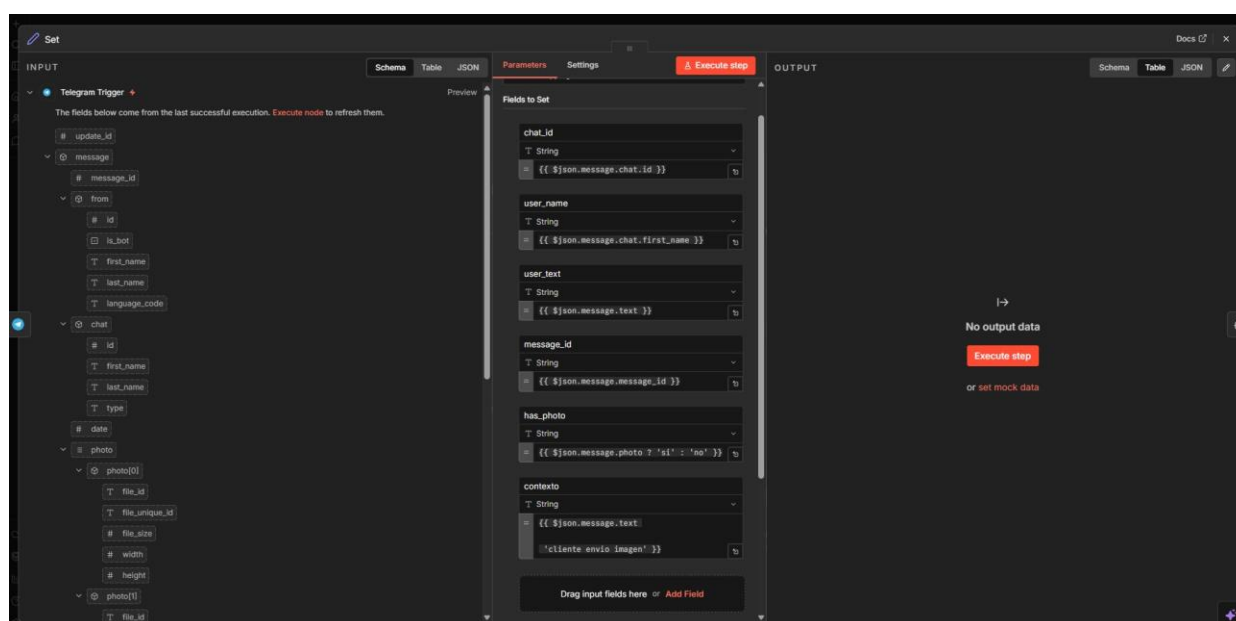
### ***Nodo 2: Set (Normalización de Datos)***

El nodo Set extrae y normaliza los campos del mensaje de Telegram para que el AI Agent los consuma de forma estructurada, opera en modo de transformación de datos (modo manual) y

mapea seis campos: chat\_id (ID único del chat del usuario), user\_name (nombre del usuario en Telegram), user\_text (texto del mensaje enviado por el usuario), message\_id (ID del mensaje para referencia), has\_photo (detecta si el mensaje contiene imagen) y contexto (contexto para el agente cuando no hay texto).

### Figura 3

*Configuración del nodo Set con los campos mapeados*



Fuente: Elaboración propia (2026)

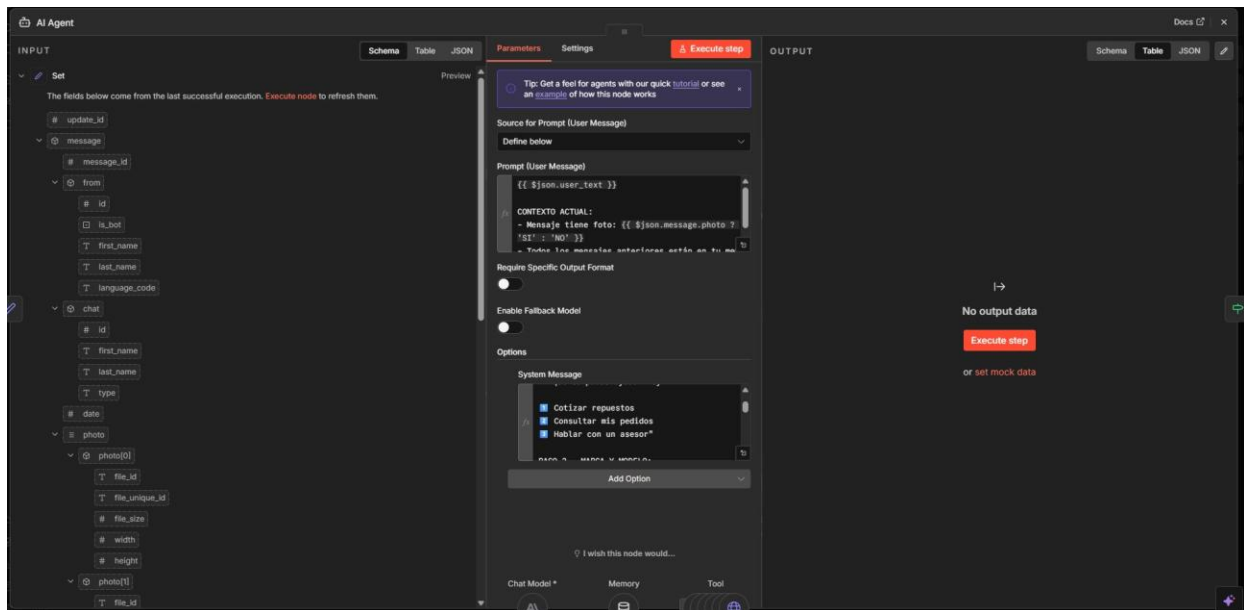
### ***Nodo 3: AI Agent***

Este nodo constituye el núcleo de inteligencia del sistema, es de tipo AI Agent (LangChain / n8n AI) y analiza el mensaje del usuario, mantiene contexto y decide qué herramientas invocar (n8n GmbH, 2026), el prompt de usuario combina el texto del usuario más contexto adicional (has\_photo, historial de memoria), el System Message define el rol del asistente Motofix con el menú inicial de tres opciones (Cotizar repuestos, Consultar mis pedidos,

Hablar con un asesor) y la lógica del flujo de ventas, tiene disponibles seis herramientas HTTP: validar\_cliente, registrar\_cliente, buscar\_repuesto, crear\_pedido, guardar\_sesion y obtener\_sesion.

## Figura 4

*Configuración del nodo AI Agent con System Message y herramientas*



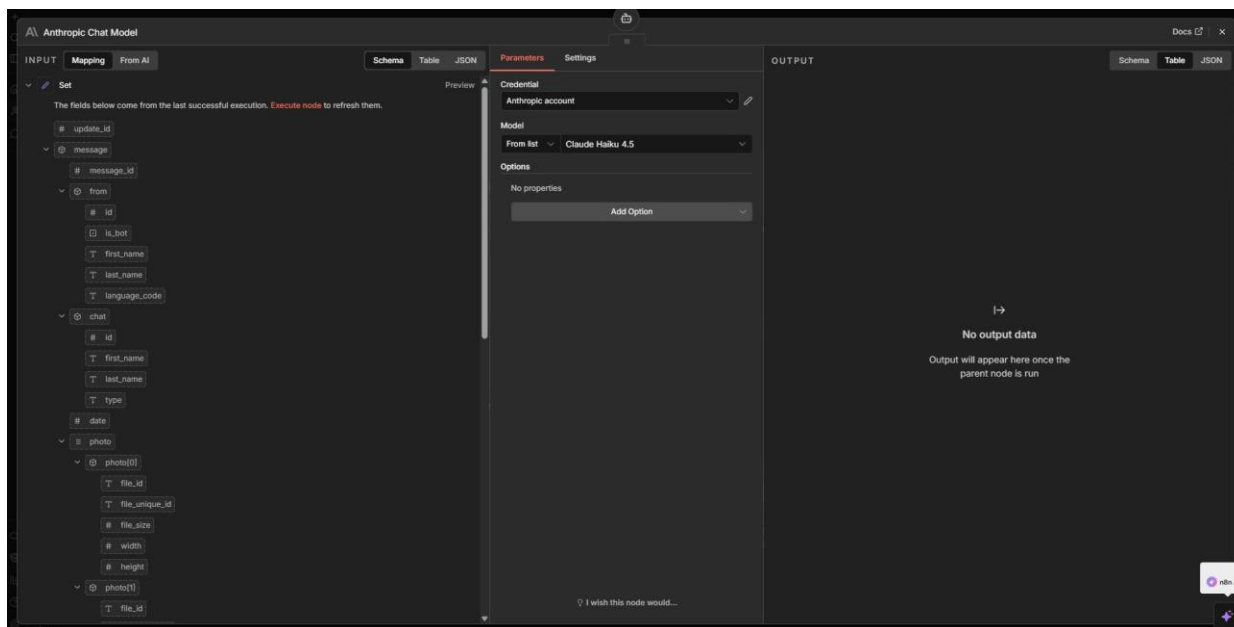
Fuente: Elaboración propia (2026)

### ***Sub-nodo 3.1: Anthropic Chat Model***

Este sub-nodo es el cerebro del agente, conecta el flujo con el modelo Claude Haiku 4.5 de Anthropic mediante una API Key, se encarga de procesar el historial de la conversación y decidir qué responder o qué acción ejecutar en cada momento.

**Figura 5**

*Configuración del sub-nodo Anthropic Chat Model (Claude Haiku 4.5)*



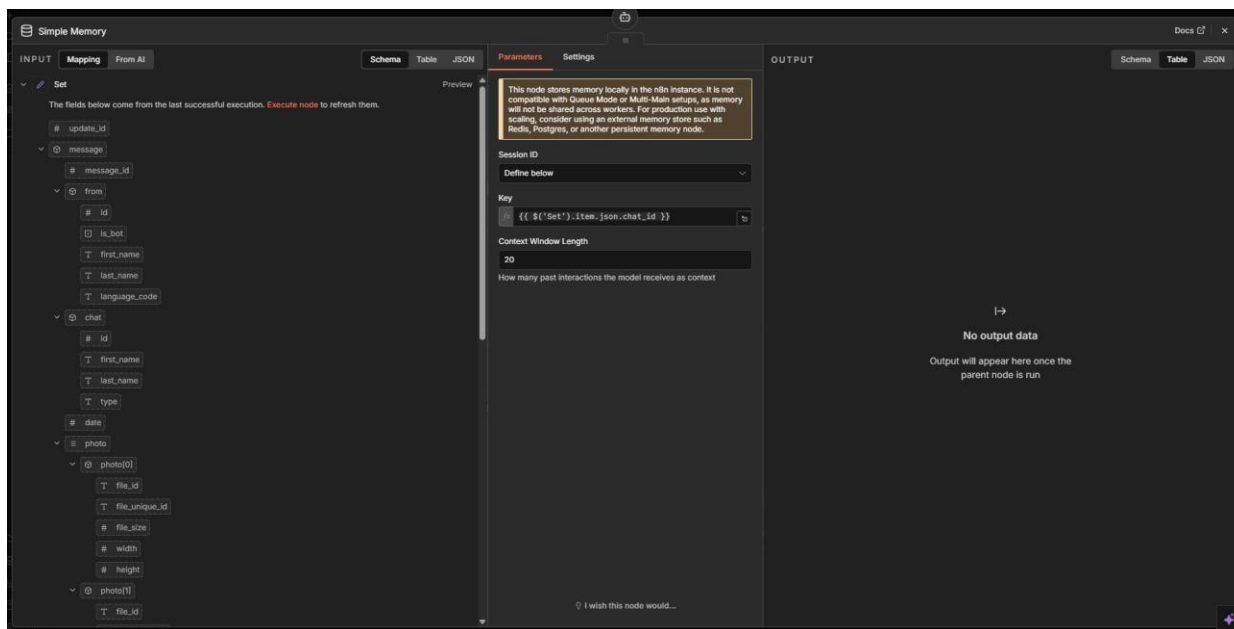
*Fuente: Elaboración propia (2026)*

### ***Sub-nodo 3.2: Simple Memory***

Este sub-nodo le da memoria al agente, guardando el historial de cada conversación identificada por el chat\_id del usuario, está configurado para recordar las últimas 20 interacciones, lo que le permite mantener el contexto a lo largo de toda la sesión, vale aclarar que esta memoria se almacena localmente en n8n, por lo que, si en algún momento el sistema crece y necesita escalar, habría que migrar a una solución de memoria externa.

**Figura 6**

*Configuración del sub-nodo Simple Memory con Session ID por chat\_id*



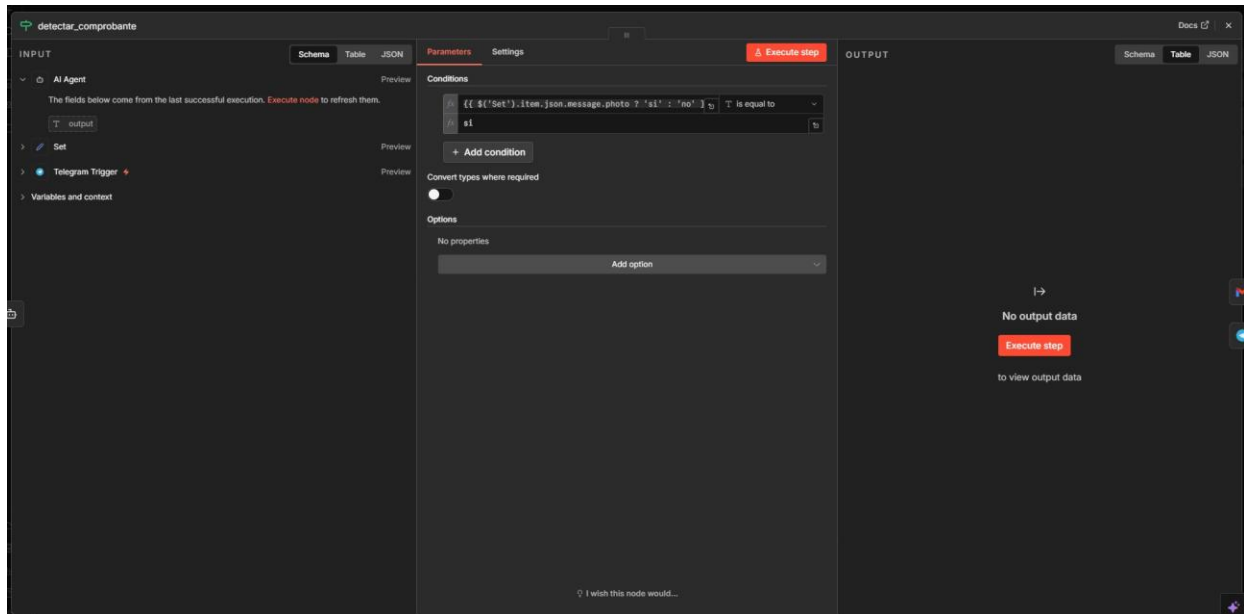
Fuente: Elaboración propia (2026)

#### ***Nodo 4: detectar\_comprobante (Nodo If)***

Este nodo funciona como un semáforo dentro del flujo, revisa si el mensaje del usuario contiene una imagen, si es así asume que es un comprobante de pago y redirige el flujo para enviar el correo de notificación al personal, si no hay imagen simplemente continúa y le responde al usuario por Telegram.

## Figura 7

### Configuración del nodo condicional detectar\_comprobante



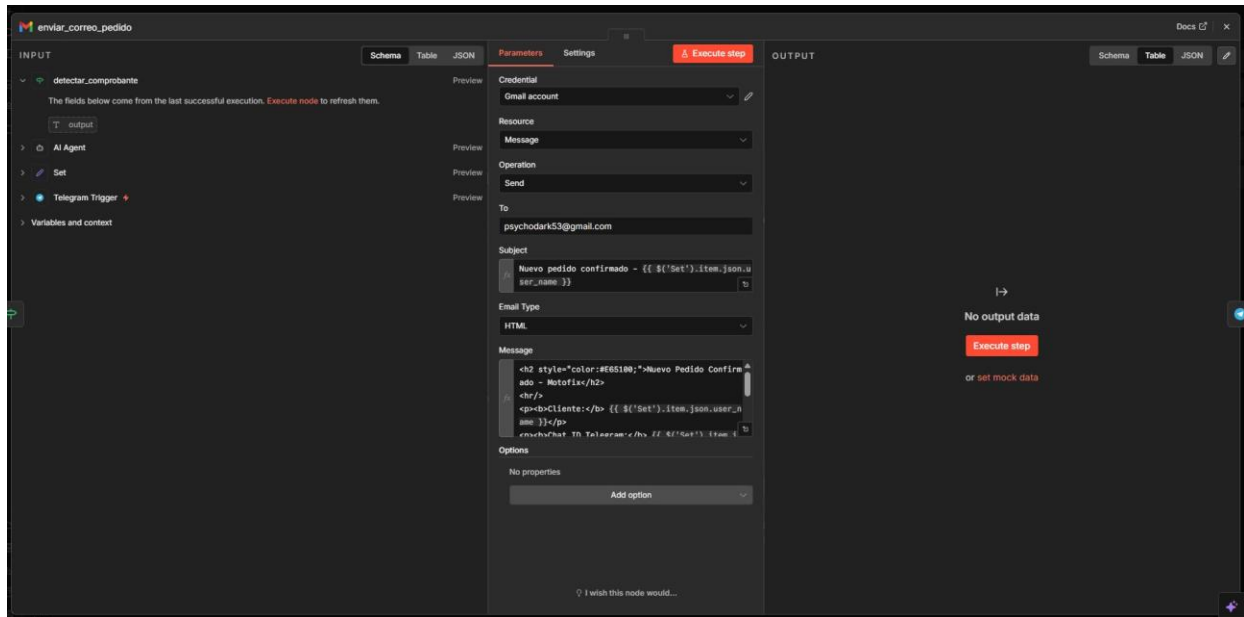
Fuente: Elaboración propia (2026)

### Nodo 5: enviar\_correo\_pedido (Gmail)

Este nodo envía una notificación interna al personal cuando el cliente envía una imagen de comprobante de pago, es de tipo Gmail Send Message, usa la credencial Gmail account (OAuth2), el asunto incluye el nombre del usuario de forma dinámica, y el cuerpo del correo es HTML con los datos del cliente (nombre, Chat ID de Telegram) y el contexto del pedido construido dinámicamente desde el nodo Set.

## Figura 8

### Configuración del nodo Gmail enviar\_correo\_pedido



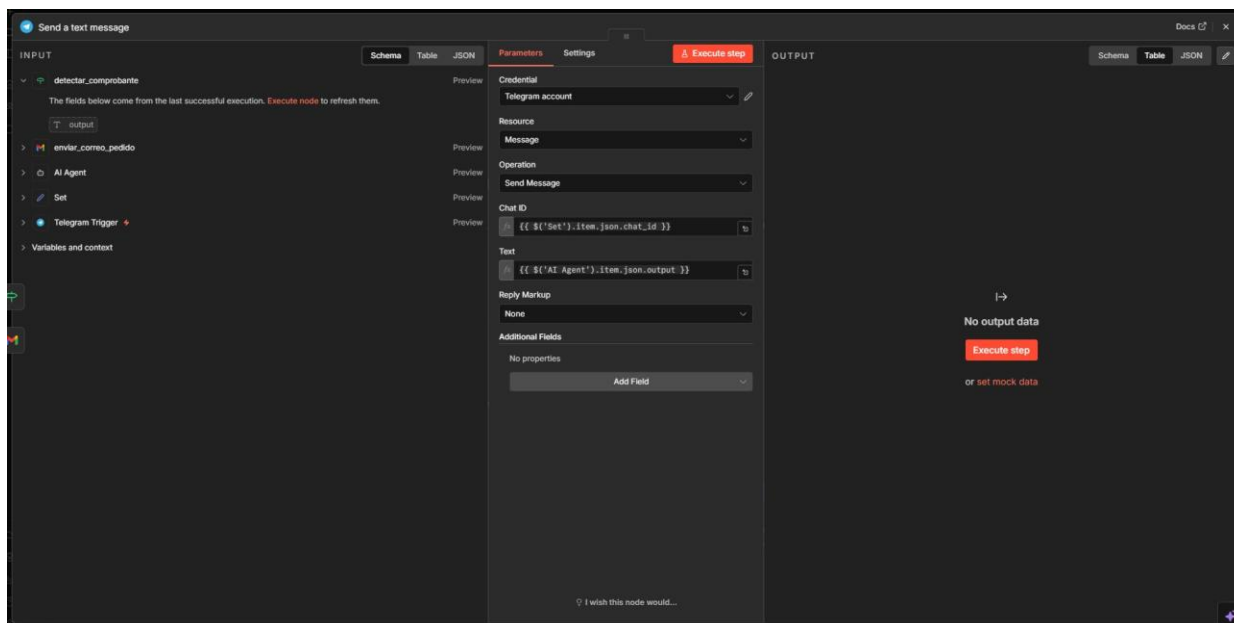
Fuente: Elaboración propia (2026)

### Nodo 6: Send a text message (Telegram)

Este nodo es el punto de salida del flujo, recibe el texto generado por el AI Agent y lo envía al usuario vía Telegram usando el chat\_id capturado al inicio de la conversación, garantizando que la respuesta llegue siempre al chat correcto.

## Figura 9

### Configuración del nodo Send a text message (Telegram)



*Fuente: Elaboración propia (2026)*

## Herramientas HTTP del AI Agent

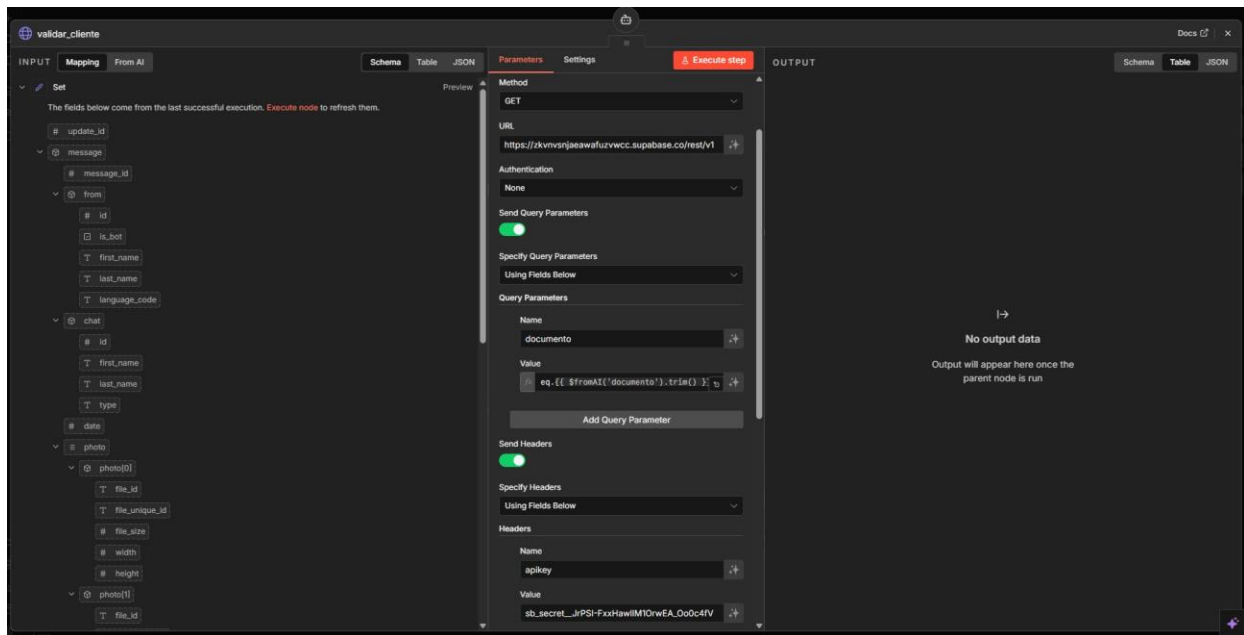
El AI Agent dispone de seis herramientas HTTP que lo conectan directamente con el backend en Supabase a través de la API REST, todas apuntan a la URL base del proyecto motofix-db alojado en Supabase, y son invocadas de forma autónoma por el agente según la lógica de la conversación.

### *Herramienta 1: validar\_cliente*

Esta herramienta verifica si un cliente ya existe en la base de datos usando su número de cédula, es de tipo HTTP GET, opera con parámetro de consulta documento con filtro de igualdad en Supabase, la autenticación se realiza mediante headers con apikey, y retorna los datos del cliente o un arreglo vacío si no existe.

**Figura 10**

*Configuración de la herramienta validar\_cliente (HTTP GET)*



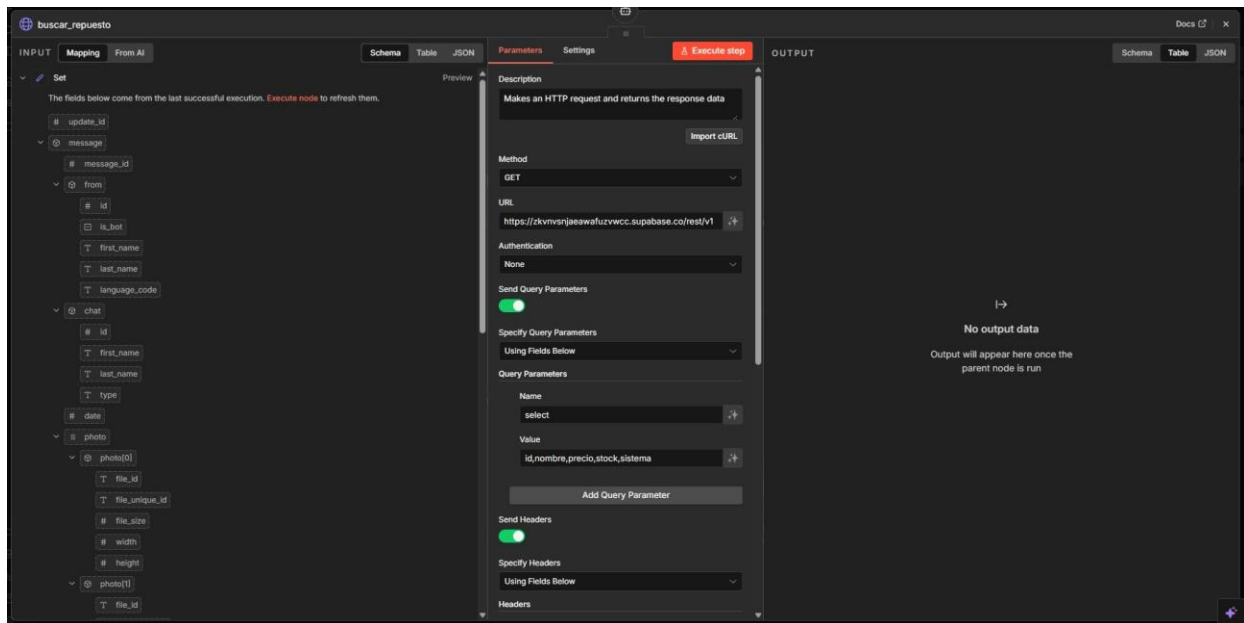
Fuente: Elaboración propia (2026)

### ***Herramienta 2: buscar\_repuesto***

Esta herramienta hace una consulta GET a la base de datos para traer los repuestos disponibles, solicita únicamente los campos necesarios (id, nombre, precio, stock y sistema) para mantener la respuesta liviana, y se autentica mediante los headers requeridos por Supabase.

Figura 11

*Configuración de la herramienta buscar\_repuesto (HTTP GET)*



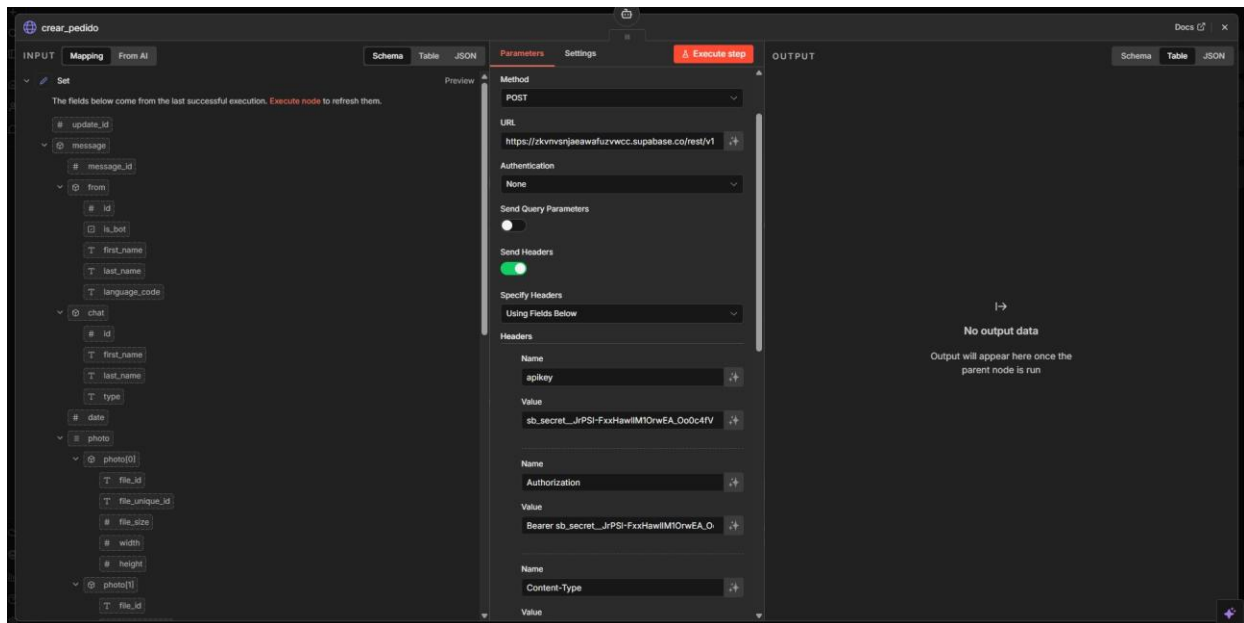
Fuente: Elaboración propia (2026)

### ***Herramienta 3: crear\_pedido***

Esta herramienta crea un nuevo pedido en la base de datos enviando via POST los datos del cliente, el repuesto seleccionado y el total de la compra en formato JSON, autenticándose con los headers requeridos por Supabase.

**Figura 12**

*Configuración de la herramienta crear\_pedido (HTTP POST)*



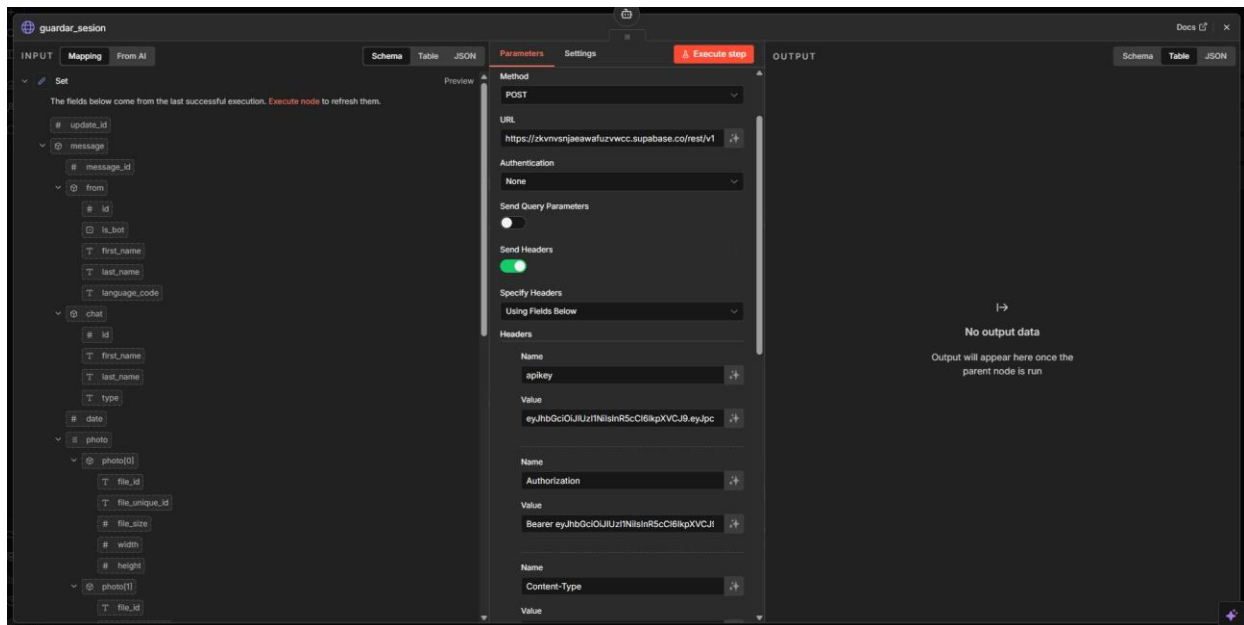
Fuente: Elaboración propia (2026)

#### ***Herramienta 4: guardar\_sesion***

Esta herramienta hace un POST a la tabla sesiones de Supabase guardando el estado actual de la compra (repuesto, precio, marca, modelo y paso del flujo), lo que permite mantener la persistencia de la conversación entre mensajes sin depender únicamente de la memoria del agente.

**Figura 13**

*Configuración de la herramienta guardar\_sesion (HTTP POST)*



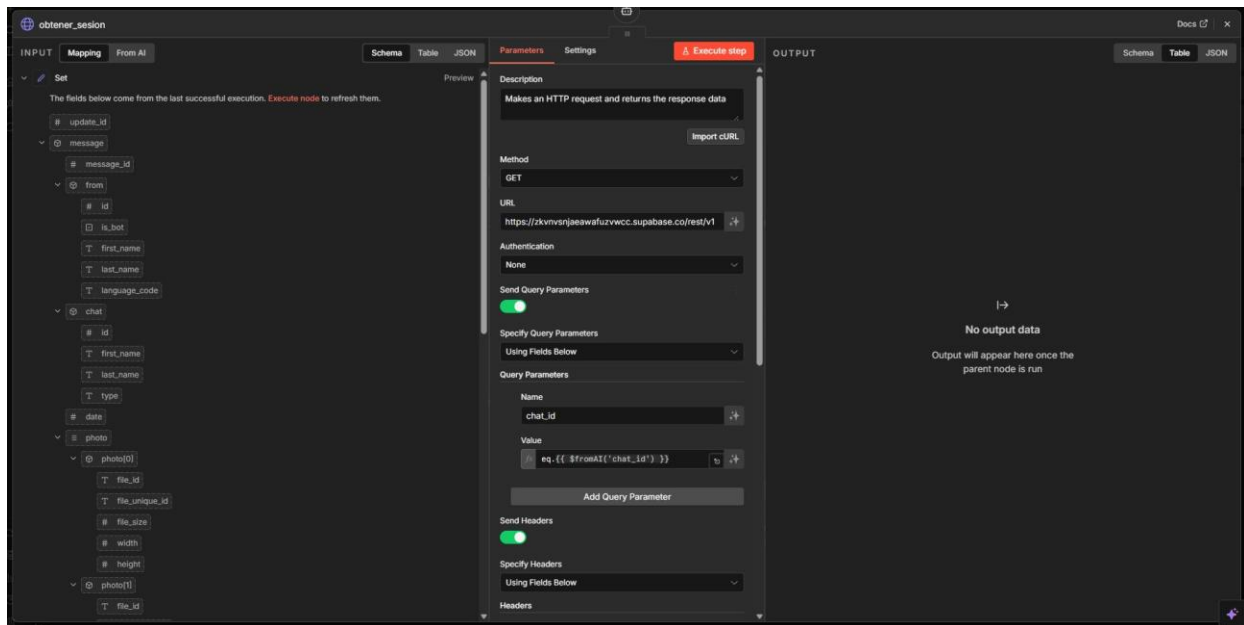
Fuente: Elaboración propia (2026)

### ***Herramienta 5: obtener\_sesion***

Esta herramienta hace un GET a la tabla sesiones filtrando por el chat\_id del usuario, recuperando el estado guardado de la conversación para que el agente pueda retomar el flujo desde donde lo dejó sin perder ningún dato.

**Figura 14**

*Configuración de la herramienta obtener sesión (HTTP GET)*



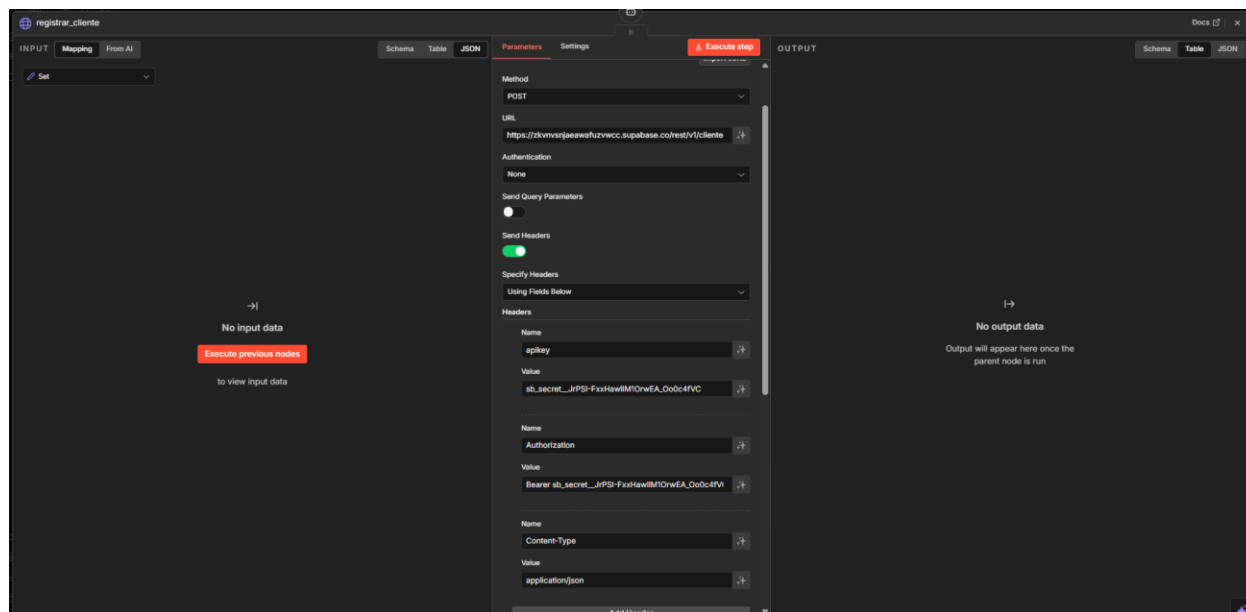
Fuente: Elaboración propia (2026)

### ***Herramienta 6: registrar\_cliente***

Esta herramienta hace un POST para registrar un nuevo cliente en la base de datos cuando el agente detecta que la cédula consultada no existe, enviando en el cuerpo JSON los datos recopilados durante la conversación como nombre, documento, celular, dirección y chat\_id de Telegram.

**Figura 15**

*Configuración de la herramienta registrar cliente (HTTP POST)*



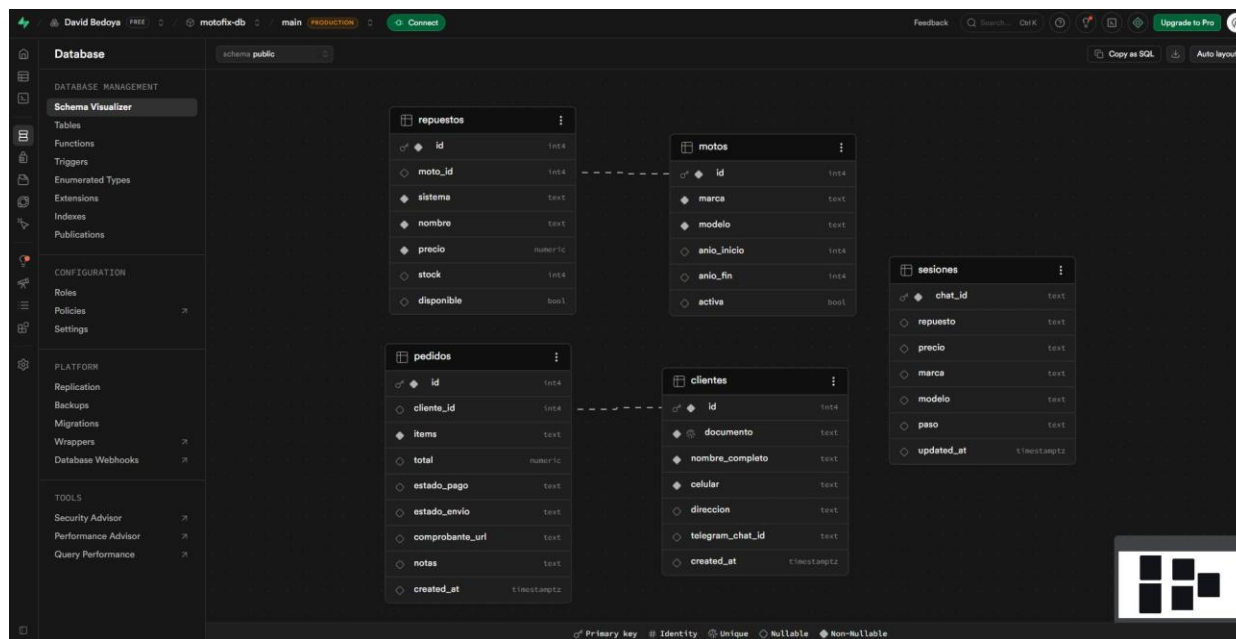
Fuente: Elaboración propia (2026)

## **Estructura de la Base de Datos (Supabase)**

La base de datos del proyecto está alojada en Supabase (PostgreSQL) bajo el proyecto motofix-db, esquema public, contiene cinco tablas principales que cubren todas las entidades del sistema y están relacionadas entre sí mediante claves foráneas.

Figura 16

*Schema Visualizer de Supabase, vista completa de las 5 tablas*



Fuente: Elaboración propia (2026)

### Tabla: repuestos

Esta tabla guarda la lista de repuestos disponibles en el inventario. Sus campos son: id (número único que identifica cada repuesto), moto\_id (número que indica con qué moto es compatible el repuesto), sistema (parte de la moto a la que pertenece el repuesto, como motor o frenos), nombre (nombre del repuesto), precio (valor de venta por unidad), stock (cantidad disponible en inventario) y disponible (indica si el repuesto está listo para la venta).

### Tabla: motos

Esta tabla almacena el catálogo de motos soportadas por el sistema, sus campos principales son la marca, el modelo, el rango de años de fabricación (anio\_inicio y anio\_fin) y un

campo booleano activa que permite habilitar o deshabilitar modelos sin eliminarlos de la base de datos.

**Tabla: clientes**

Esta tabla almacena los datos de los clientes registrados en el sistema, el campo documento es único y funciona como identificador principal del cliente, también guarda nombre completo, celular, dirección y el chat\_id de Telegram vinculado, lo que permite al agente reconocer al cliente en futuras conversaciones sin pedirle sus datos nuevamente.

**Tabla: pedidos**

Esta tabla registra cada pedido generado por el sistema, vinculado a un cliente mediante cliente\_id como clave foránea, almacena los repuestos y cantidades en un campo JSON, el valor total, y maneja dos estados independientes: estado\_pago (pendiente / confirmado / rechazado) y estado\_envio (pendiente / en\_proceso / entregado), además guarda la URL del comprobante de pago cuando el cliente lo envía.

**Tabla: sesiones**

Esta tabla usa el chat\_id de Telegram como clave primaria para persistir el estado de cada sesión activa, almacena el repuesto seleccionado, su precio, la marca y modelo de la moto y el paso actual del flujo de compra, el campo updated\_at permite saber cuándo fue la última interacción y detectar sesiones inactivas.

## Evidencias de Funcionamiento Conversación Real con el Bot

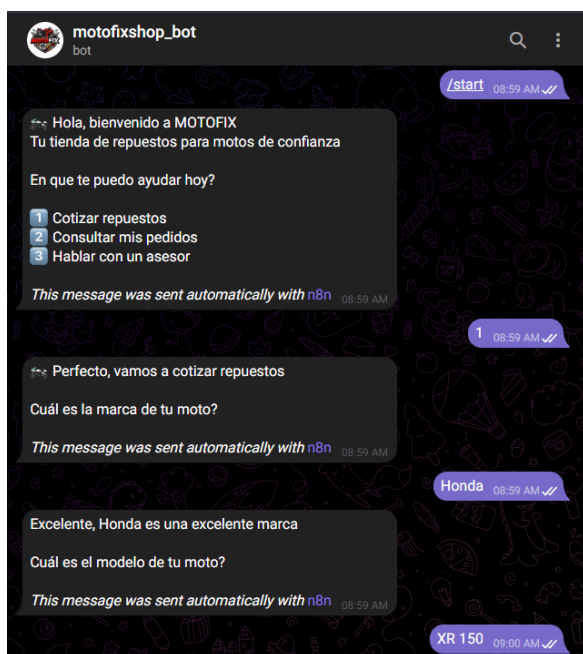
En el presente apartado se presentan capturas de pantalla de conversaciones reales con el bot de Telegram Motofix, demostrando el funcionamiento completo del sistema desde el primer mensaje del usuario hasta la confirmación del pedido.

### Inicio de Conversación y Menú Principal

El usuario inicia la conversación y el bot responde con el menú principal de opciones, demostrando la presentación inicial del sistema con las tres opciones disponibles: Cotizar repuestos, Consultar mis pedidos y Hablar con un asesor.

### Figura 17

*Inicio de conversación, el bot presenta el menú principal de MotoFix*



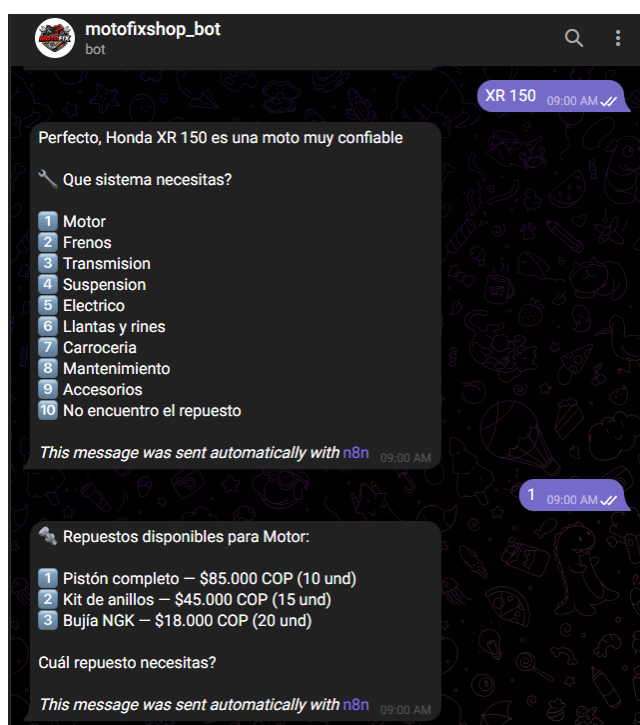
Fuente: Elaboración propia (2026)

## Registro de Nuevo Cliente

Cuando el bot detecta que el usuario no está registrado en el sistema, solicita automáticamente los datos necesarios para crear su perfil: número de documento, nombre completo, celular y dirección, el proceso es guiado y completamente conversacional, sin formularios externos.

### Figura 18

*Flujo de registro automático de nuevo cliente*



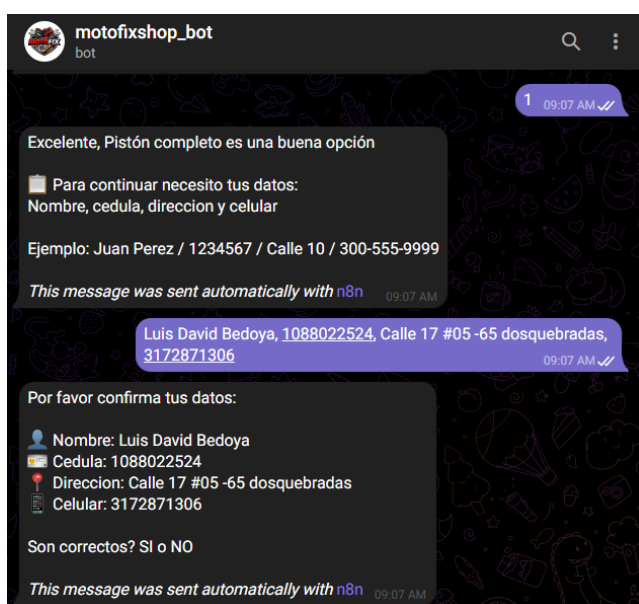
Fuente: Elaboración propia (2026)

## Búsqueda de Repuesto por Marca y Modelo

El usuario indica la marca y modelo de su motocicleta, el agente invoca la herramienta buscar\_repuesto con los parámetros correspondientes y retorna una lista de repuestos disponibles con sus precios, la respuesta es clara, estructurada y generada en lenguaje natural.

### Figura 19

*Consulta de repuestos disponibles por marca y modelo de moto*



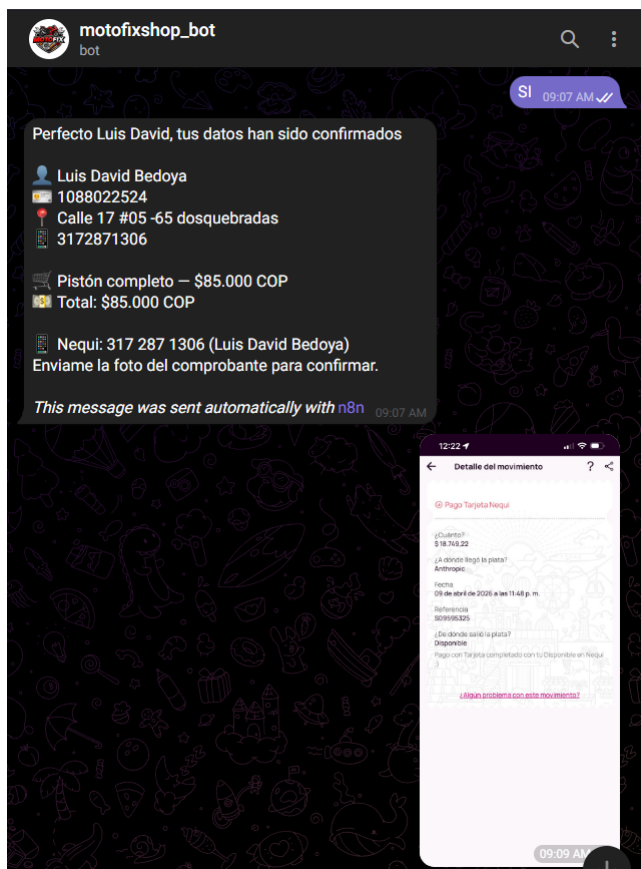
Fuente: Elaboración propia (2026)

## Selección de Repuesto y Confirmación de Pedido

El usuario selecciona el repuesto de su interés, el bot confirma los detalles del pedido (nombre del repuesto, precio, total) y solicita confirmación antes de proceder con la creación formal del pedido en la base de datos, esta doble confirmación evita registros erróneos.

**Figura 20**

*Selección de repuesto y confirmación del pedido por el cliente*



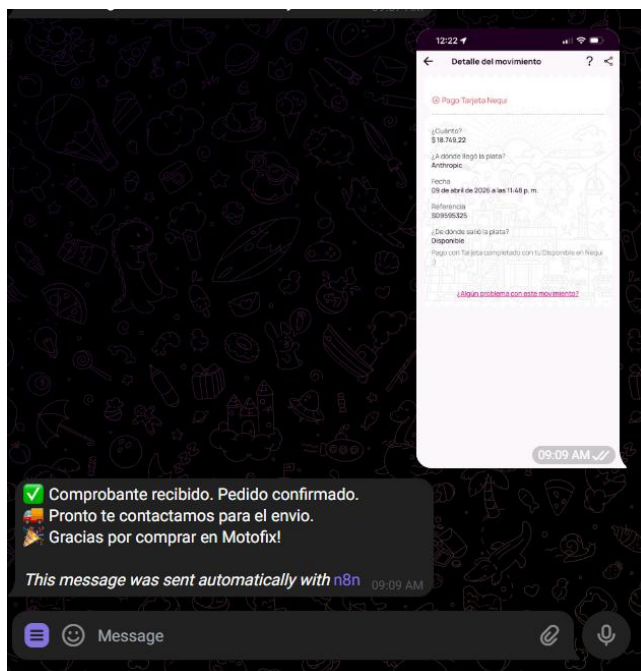
Fuente: Elaboración propia (2026)

## Instrucciones de Pago

Una vez confirmado el pedido, el bot proporciona las instrucciones de pago al cliente, incluyendo el total a cancelar y los métodos disponibles, solicitando el envío del comprobante de pago como imagen.

## Figura 21

*Bot entrega instrucciones de pago y solicita comprobante*



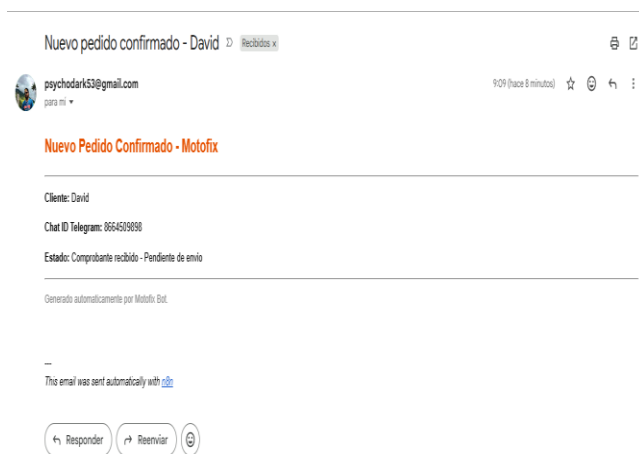
Fuente: Elaboración propia (2026)

## Confirmación Final

Tras el proceso completo, el sistema confirma al cliente que su pedido ha sido registrado exitosamente, si se recibió un comprobante de pago (imagen), se activa automáticamente el flujo de notificación por correo electrónico al equipo interno.

## Figura 22

*Confirmación final del pedido, ciclo de venta completado*



Fuente: Elaboración propia (2026)

## Ciclo Completo de una Venta

El ciclo completo de una transacción en Motofix comprende quince pasos secuenciales que cubren desde la recepción del mensaje inicial hasta la confirmación final al usuario, el paso 1 corresponde al mensaje inicial donde el usuario envía un mensaje al bot de Telegram de Motofix, en el paso 2 el Telegram Trigger captura el evento y extrae chat\_id y texto, el paso 3 es la preparación donde el nodo Set normaliza los datos para el AI Agent, el paso 4 consiste en recuperar la sesión donde el AI Agent invoca obtener\_sesion para verificar si hay una compra en progreso, el paso 5 es la validación del cliente donde el agente solicita el documento al usuario e invoca validar\_cliente.

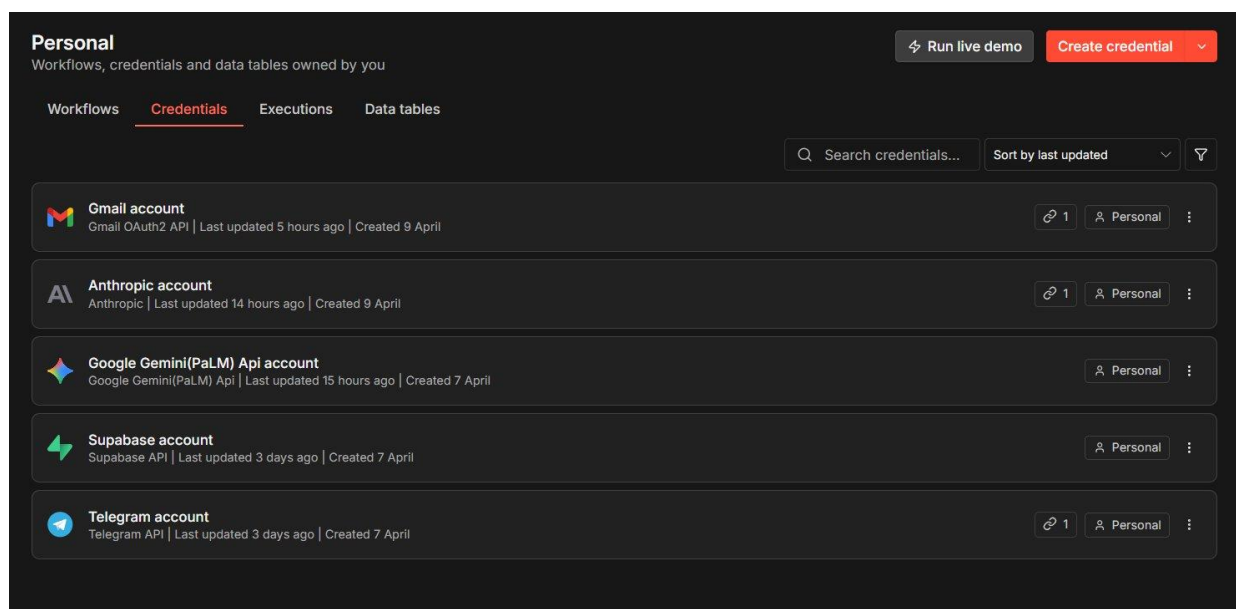
Si el cliente no existe (paso 6a), el agente solicita los datos y ejecuta registrar\_cliente, si el cliente ya existe (paso 6b), el agente lo saluda por nombre y continúa el flujo, en el paso 7 el agente invoca buscar\_repuesto con los criterios indicados por el usuario, en el paso 8 responde con la lista de repuestos disponibles y sus precios, en el paso 9 el usuario confirma el repuesto deseado, en el paso 10 el agente guarda el progreso con guardar\_sesion, en el paso 11 al confirmar la compra se ejecuta crear\_pedido, en el paso 12 el agente indica el total y solicita el comprobante de pago, el paso 13 corresponde a la detección de la imagen donde el nodo detectar\_comprobante evalúa si se recibió una imagen, si hay imagen (paso 14) se activa enviar\_correo\_pedido para notificar al equipo, y en el paso 15 Send a text message envía la confirmación final al usuario por Telegram.

## Credenciales y APIs Integradas al Proyecto

El proyecto Motofix integra cinco servicios externos, cada uno autenticado mediante credenciales configuradas en n8n, todas las credenciales son de ámbito Personal dentro del workspace y están activamente referenciadas por los nodos del flujo.

### Figura 23

*Panel de credenciales en n8n, APIs conectadas al proyecto MotoFix*



Fuente: Elaboración propia (2026)

Gmail account utiliza OAuth2 de Google para el envío de correos de confirmación de pedidos, Anthropic account se autentica mediante API Key para acceder al modelo Claude Haiku 4.5 que alimenta el AI Agent, Google Gemini (PaLM) API es una credencial adicional disponible en el workspace, Supabase account usa API Key y Bearer Token JWT para la conexión con la base de datos PostgreSQL del proyecto motofix-db, y Telegram account usa el Bot Token del bot creado vía @BotFather, tanto en el trigger como en el nodo de respuesta.

## Resumen General de Componentes

El sistema Motofix está compuesto por catorce nodos organizados según su función dentro del flujo, los de entrada y salida se encargan de recibir los mensajes del usuario por Telegram y devolverle la respuesta del agente, los de procesamiento normalizan los datos del mensaje entrante, detectan si se envió una imagen de comprobante y gestionan el envío del correo de notificación al equipo.

El núcleo inteligente lo forman el AI Agent, el modelo Claude Haiku 4.5 y el módulo de memoria, que juntos se encargan de entender al usuario, recordar el contexto y decidir qué hacer en cada paso, finalmente las seis herramientas HTTP son las que conectan al agente con la base de datos, permitiéndole validar y registrar clientes, consultar repuestos, crear pedidos y gestionar el estado de cada sesión.

## Conclusiones

El desarrollo e implementación de Motofix demuestra que es posible construir un sistema de automatización comercial funcional, escalable y de bajo costo utilizando herramientas de código abierto y APIs accesibles, la integración de n8n, Anthropic Claude, Supabase y Telegram en un único flujo de trabajo orquestado permite automatizar completamente el proceso de ventas de repuestos para motos sin necesidad de intervención humana en las operaciones rutinarias.

Los resultados obtenidos durante las pruebas de flujo completo confirman que el agente conversacional es capaz de gestionar de manera autónoma la validación y registro de clientes, la consulta en tiempo real del inventario, la generación de pedidos y la notificación al personal interno mediante correo electrónico, la arquitectura de herramientas HTTP conectadas al AI Agent resulta ser un patrón de diseño robusto y extensible que puede adaptarse a otros sectores de comercio con mínimas modificaciones.

Desde el punto de vista académico, este proyecto demuestra que es posible integrar automatización de flujos de trabajo, inteligencia artificial conversacional y bases de datos en la nube en una solución funcional de bajo costo, lo que lo convierte en un modelo replicable para negocios similares dentro del comercio digital latinoamericano.

## Código Fuente del Proyecto, Flujo n8n (JSON)

A continuación se presenta el código fuente del flujo exportado desde n8n en formato JSON, el archivo contiene la definición de los 14 nodos, sus configuraciones y conexiones, y puede importarse directamente en cualquier instancia de n8n para replicar el sistema.

```
{
  "name": "My workflow",
  "nodes": [
    {
      "parameters": {
        "updates": [
          "message",
          "callback_query"
        ],
        "additionalFields": {}
      },
      "type": "n8n-nodes-base.telegramTrigger",
      "typeVersion": 1.2,
      "position": [
        -864,
        96
      ],
      "id": "062c6d88-2375-4618-b3d7-72cca58f4198",
      "name": "Telegram Trigger",
      "webhookId": "5deb428f-bb72-483c-8d46-4ca2b0fa0b0d",
      "credentials": {
        "telegramApi": {
          "id": "YdSNpXVdu02ha3on",
          "name": "Telegram account"
        }
      }
    }
  ],
  {
```

```

"parameters": {
  "promptType": "define",
  "text": "={{ $json.user_text }}\n\nCONTEXTO ACTUAL:\n- Mensaje tiene foto: {{
$json.message.photo ? 'SI' : 'NO' }}\n- Todos los mensajes anteriores están en tu memoria",
  "options": {
    "systemMessage": "=Eres MotoBot, asistente de MOTOFIX tienda de repuestos para
motos.\nResponde en español con emojis. NUNCA uses ** | - \ \ en los mensajes.\nEl chat_id del
cliente es: {{ $('Set').item.json.chat_id }}\n\nFLUJO ESTRICTO:\n\nPASO 1 - SALUDO:\n\n👋 Hola,
bienvenido a MOTOFIX\nTu tienda de repuestos para motos de confianza\n\nEn que te puedo ayudar
hoy?\n\n1 Cotizar repuestos\n2 Consultar mis pedidos\n3 Hablar con un asesor"\n\nPASO 2 -
MARCA Y MODELO:\nPregunta la marca. Cuando la tenga pregunta el modelo.\nCuando tengas marca Y
modelo llama a guardar_sesion con:\nchat_id, marca, modelo, paso='sistema'\n\nPASO 3 - MENU
SISTEMAS:\n\n👉 Que sistema necesitas?\n1 Motor\n2 Frenos\n3 Transmision\n4 Suspension\n5
Electrico\n6 Llantas y rines\n7 Carroceria\n8 Mantenimiento\n9 Accesorios\n10 No encuentro
el repuesto"\n\nPASO 4 - REPUESTOS:\nLlama a buscar_repuesto con el sistema elegido.\nMuestra
los resultados:\n\n👉 Repuestos disponibles:\n1. [nombre] - ${[precio]} COP ([stock] und)\n2.
[nombre] - ${[precio]} COP ([stock] und)"\n\nCuando el cliente elija un repuesto llama a
guardar_sesion con:\nchat_id, repuesto, precio, paso='datos'\n\nPASO 5 - PEDIR DATOS:\n\n📄 Para
continuar necesito tus datos:\nNombre, cedula, direccion y celular\nEjemplo: Juan Perez / 1234567
/ Calle 10 / 300-555-9999"\n\nPASO 5B - CONFIRMAR DATOS:\nCuando el cliente envíe sus datos
muestra:\n\n"Por favor confirma tus datos:\n👤 Nombre: [nombre]\n📄 Cedula: [cedula]\n📍
Direccion: [direccion]\n📞 Celular: [celular]\nSon correctos? SI o NO"\n\nPASO 6 - CUANDO EL
CLIENTE DIGA SI:\n1. Llama a obtener_sesion con el chat_id para recuperar repuesto y precio\n2.
Llama a validar_cliente con la cedula\n3. Si no existe llama a registrar_cliente\n4. Llama a
crear_pedido\n5. Muestra exactamente:\n\n"Perfecto [nombre], tus datos han sido confirmados\n\n👤
[nombre]\n📄 [cedula]\n📍 [direccion]\n📞 [celular]\n\n🗣️ [repuesto del obtener_sesion] -
${[precio del obtener_sesion]} COP\n👉 Total: ${[precio]} COP\n\n📄 Nequi: 317 287 1306 (Luis David
Bedoya)\nEnviame la foto del comprobante para confirmar.""\n\nPASO 6B - CUANDO EL CLIENTE DIGA
NO:\nPide que corrija sus datos.\n\nPASO 7 - COMPROBANTE:\nCuando llegue una imagen
responde:\n\n✅ Comprobante recibido. Pedido confirmado.\n👉 Pronto te contactamos para el
envio.\n👋 Gracias por comprar en Motofix!"\n\nREGLAS CRITICAS:\n- SIEMPRE usa guardar_sesion
para guardar marca, modelo y repuesto\n- SIEMPRE usa obtener_sesion antes de crear_pedido para
recuperar el repuesto\n- NUNCA repitas pasos ya completados\n- NUNCA inventes precios\n- NUNCA

```

vuelvas a pedir datos ya confirmados\n- Precios en formato \$85.000 COP\n- Una imagen siempre es el comprobante, ir directo al PASO 7"

```

    }
  },
  "type": "@n8n/n8n-nodes-langchain.agent",
  "typeVersion": 3.1,
  "position": [
    -368,
    96
  ],
  "id": "d8dc1e1e-0304-4e74-944e-df63d4b56a32",
  "name": "AI Agent"
},
{
  "parameters": {
    "conditions": {
      "options": {
        "caseSensitive": true,
        "leftValue": "",
        "typeValidation": "strict",
        "version": 3
      },
      "conditions": [
        {
          "id": "aacf6583-13b5-4b64-a848-b5a6a3cede25",
          "leftValue": "={{ $('Set').item.json.message.photo ? 'si' : 'no' }}",
          "rightValue": "=si",
          "operator": {
            "type": "string",
            "operation": "equals"
          }
        }
      ]
    },
    "combinator": "and"
  },
  "options": {}
}

```

```

    },
    "type": "n8n-nodes-base.if",
    "typeVersion": 2.3,
    "position": [
      0,
      96
    ],
    "id": "882d82b5-7bb6-41a6-89ed-5ca23a9cf877",
    "name": "detectar_comprobante"
  },
  {
    "parameters": {
      "sessionIdType": "customKey",
      "sessionKey": "={{ $('Set').item.json.chat_id }}",
      "contextWindowLength": 20
    },
    "type": "@n8n/n8n-nodes-langchain.memoryBufferWindow",
    "typeVersion": 1.3,
    "position": [
      -400,
      384
    ],
    "id": "3d464eb1-64c6-4e80-9451-896602cc1236",
    "name": "Simple Memory"
  },
  {
    "parameters": {
      "method": "POST",
      "url": "https://zkvnvsnjaeawafuzvcc.supabase.co/rest/v1/pedidos",
      "sendHeaders": true,
      "headerParameters": {
        "parameters": [
          {
            "name": "apikey",
            "value": "sb_secret__JrPSI-FxxHawllM1OrwEA_Oo0c4fVC"
          }
        ]
      }
    }
  },

```

```

    {
      "name": "Authorization",
      "value": "Bearer sb_secret__JrPSI-FxxHawllMlOrwEA_Oo0c4fVC"
    },
    {
      "name": "Content-Type",
      "value": "application/json"
    },
    {
      "name": "Prefer",
      "value": "return=representation"
    }
  ]
},
"sendBody": true,
"specifyBody": "json",
"jsonBody": "={\n  \"cliente_id\": \"{{ $fromAI('cliente_id') }}\", \n  \"items\":
\"{{ $fromAI('repuesto') }}\", \n  \"total\": \"{{ $fromAI('total') }}\", \n  \"estado_pago\":
\"pendiente\", \n  \"estado_envio\": \"por_preparar\"\n}",
  "options": {}
},
"type": "n8n-nodes-base.httpRequestTool",
"typeVersion": 4.4,
"position": [
  208,
  576
],
"id": "2444997b-5e7e-4fcf-8263-a1238b27f694",
"name": "crear_pedido"
},
{
  "parameters": {
    "url": "https://zkvnvsnjaeawafuzvwcc.supabase.co/rest/v1/repuestos",
    "sendQuery": true,
    "queryParameters": {
      "parameters": [

```

```

        {
            "name": "select",
            "value": "id,nombre,precio,stock,sistema"
        }
    ]
},
"sendHeaders": true,
"headerParameters": {
    "parameters": [
        {
            "name": "apikey",
            "value": "sb_secret__JrPSI-FxxHawllM1OrwEA_Oo0c4fVC"
        },
        {
            "name": "Authorization",
            "value": "Bearer sb_secret__JrPSI-FxxHawllM1OrwEA_Oo0c4fVC"
        }
    ]
},
"options": {}
},
"type": "n8n-nodes-base.httpRequestTool",
"typeVersion": 4.4,
"position": [
    32,
    576
],
"id": "9ad6d5be-9c00-456d-94db-37f9feba3ae6",
"name": "buscar_repuesto"
},
{
    "parameters": {
        "method": "POST",
        "url": "https://zkvnvsnjaeawafuzvwcc.supabase.co/rest/v1/clientes",
        "sendHeaders": true,
        "headerParameters": {

```

```

    "parameters": [
      {
        "name": "apikey",
        "value": "sb_secret__JrPSI-FxxHawllM1OrwEA_Oo0c4fVC"
      },
      {
        "name": "Authorization",
        "value": "Bearer sb_secret__JrPSI-FxxHawllM1OrwEA_Oo0c4fVC"
      },
      {
        "name": "Content-Type",
        "value": "application/json"
      }
    ]
  },
  "sendBody": true,
  "specifyBody": "json",
  "jsonBody": "={\n  \"nombre_completo\": \"{{ $fromAI('nombre') }}\", \n
  \"documento\": \"{{ $fromAI('documento') }}\", \n  \"celular\": \"{{ $fromAI('celular') }}\", \n
  \"direccion\": \"{{ $fromAI('direccion') }}\", \n  \"telegram_chat_id\": \"{{
$('Set').item.json.chat_id }}\" \n}",
  "options": {}
},
"type": "n8n-nodes-base.httpRequestTool",
"typeVersion": 4.4,
"position": [
  -144,
  576
],
"id": "8740c493-dfd2-475d-ae00-9e96d0a6e4c9",
"name": "registrar_cliente"
},
{
  "parameters": {
    "assignments": {
      "assignments": [

```

```

{
  "id": "3a0b2be9-7b47-415f-a9b4-dbb51878658c",
  "name": "chat_id",
  "value": "={{ $json.message.chat.id }}",
  "type": "string"
},
{
  "id": "df25c67e-8eb2-448a-be83-f8bb0f3e27b8",
  "name": "user_name",
  "value": "={{ $json.message.chat.first_name }}",
  "type": "string"
},
{
  "id": "e3f91a3f-bb5f-4dc5-9f10-f0357803e931",
  "name": "user_text",
  "value": "={{ $json.message.text }}",
  "type": "string"
},
{
  "id": "ab6048b9-ba33-4c8c-b3c2-7fd82f1b59e5",
  "name": "message_id",
  "value": "={{ $json.message.message_id }}",
  "type": "string"
},
{
  "id": "79e839a4-7ab5-4a66-b231-791636ff9083",
  "name": "has_photo",
  "value": "={{ $json.message.photo ? 'si' : 'no' }}",
  "type": "string"
},
{
  "id": "b991a746-8be2-4719-bc15-c24914a2d8a8",
  "name": "contexto",
  "value": "={{ $json.message.text \n\n 'cliente envio imagen' }}",
  "type": "string"
}

```

```

    ]
  },
  "includeOtherFields": true,
  "options": {}
},
"type": "n8n-nodes-base.set",
"typeVersion": 3.4,
"position": [
  -592,
  96
],
"id": "815ba463-147c-4bb1-8a40-a49b896a9ae9",
"name": "Set"
},
{
  "parameters": {
    "chatId": "={{ $('Set').item.json.chat_id }}",
    "text": "={{ $('AI Agent').item.json.output }}",
    "additionalFields": {}
  },
  "type": "n8n-nodes-base.telegram",
  "typeVersion": 1.2,
  "position": [
    544,
    272
  ],
  "id": "2f92a79d-040b-4041-b15f-ad669f70ba1c",
  "name": "Send a text message",
  "webhookId": "c088b9d7-a592-4bae-b767-4ccaf46d3630",
  "credentials": {
    "telegramApi": {
      "id": "YdSNpXVdu02ha3on",
      "name": "Telegram account"
    }
  }
}
},

```

```

{
  "parameters": {
    "sendTo": "psychodark53@gmail.com",
    "subject": "=Nuevo pedido confirmado - {{ $('Set').item.json.user_name }}",
    "message": "=<h2 style=\"color:#E65100;\">Nuevo Pedido Confirmado -
Motofix</h2>\n<hr/>\n<p><b>Cliente:</b> {{ $('Set').item.json.user_name }}</p>\n<p><b>Chat ID
Telegram:</b> {{ $('Set').item.json.chat_id }}</p>\n<p><b>Estado:</b> Comprobante recibido -
Pendiente de envio</p>\n<hr/>\n<p style=\"color:#888;font-size:12px;\">Generado automaticamente
por Motofix Bot.</p>",
    "options": {}
  },
  "type": "n8n-nodes-base.gmail",
  "typeVersion": 2.2,
  "position": [
    320,
    80
  ],
  "id": "3b0b06d0-045b-4346-9d29-9104711917b3",
  "name": "enviar_correo_pedido",
  "webhookId": "b69f4a15-1bc5-408f-8d78-8021ea84de6c",
  "credentials": {
    "gmailOAuth2": {
      "id": "eCVtgdSZ67gJlYjP",
      "name": "Gmail account"
    }
  }
},
{
  "parameters": {
    "url": "https://zkvnvsnjaeawafuzvwcc.supabase.co/rest/v1/clientes",
    "sendQuery": true,
    "queryParameters": {
      "parameters": [
        {
          "name": "documento",
          "value": "=eq.{{ $fromAI('documento').trim() }}"
        }
      ]
    }
  }
}

```

```

    }
  ]
},
"sendHeaders": true,
"headerParameters": {
  "parameters": [
    {
      "name": "apikey",
      "value": "sb_secret__JrPSI-FxxHawllM1OrwEA_Oo0c4fVC"
    },
    {
      "name": "Authorization",
      "value": "Bearer sb_secret__JrPSI-FxxHawllM1OrwEA_Oo0c4fVC"
    }
  ]
},
"options": {}
},
"type": "n8n-nodes-base.httpRequestTool",
"typeVersion": 4.4,
"position": [
  -320,
  576
],
"id": "7ecee384-0fa4-4f1b-98ff-6adf95d9cb2c",
"name": "validar_cliente"
},
{
  "parameters": {
    "model": {
      "__rl": true,
      "value": "claude-haiku-4-5-20251001",
      "mode": "list",
      "cachedResultName": "Claude Haiku 4.5"
    },
  },
  "options": {}
}

```

```

    },
    "type": "@n8n/n8n-nodes-langchain.lmChatAnthropic",
    "typeVersion": 1.3,
    "position": [
      -544,
      384
    ],
    "id": "1c70258c-662a-4097-99e0-bb0ecd59c1b0",
    "name": "Anthropic Chat Model",
    "credentials": {
      "anthropicApi": {
        "id": "w8WtzbyC8gW0DGn1",
        "name": "Anthropic account"
      }
    }
  },
  {
    "parameters": {
      "method": "POST",
      "url": "https://zknvnsnjaeawafuzvwcc.supabase.co/rest/v1/sesiones",
      "sendHeaders": true,
      "headerParameters": {
        "parameters": [
          {
            "name": "apikey",
            "value":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Inprdm52c25qYWVhd2FmdXp2d2NjIiwicm9sZSI6ImFub24iLCJpYXQiOiJlZ3NzU1MzI5NjEsImV4cCI6ImJA5MTEwODk2MX0.2gGnYcKBcdBnFO2Dd31CeV90ux_0zdrOUjHwXYq1gBQ"
          }
        ],
        "name": "Authorization",
        "value": "Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Inprdm52c25qYWVhd2FmdXp2d2NjIiwicm9sZSI6ImFub24iLCJpYXQiOiJlZ3NzU1MzI5NjEsImV4cCI6ImJA5MTEwODk2MX0.2gGnYcKBcdBnFO2Dd31CeV90ux_0zdrOUjHwXYq1gBQ"
      }
    }
  }
}

```

```

    },
    {
      "name": "Content-Type",
      "value": "application/json"
    },
    {
      "name": "Prefer",
      "value": "return=representation,resolution=merge-duplicates"
    }
  ]
},
"sendBody": true,
"specifyBody": "json",
"jsonBody": "={\n  \"chat_id\": \"{{ $fromAI('chat_id') }}\", \n  \"repuesto\":
\"{{ $fromAI('repuesto') }}\", \n  \"precio\": \"{{ $fromAI('precio') }}\", \n  \"marca\": \"{{
$fromAI('marca') }}\", \n  \"modelo\": \"{{ $fromAI('modelo') }}\", \n  \"paso\": \"{{
$fromAI('paso') }}\" \n}",
  "options": {}
},
"type": "n8n-nodes-base.httpRequestTool",
"typeVersion": 4.4,
"position": [
  384,
  576
],
"id": "189e8f4a-8fe0-434a-aefc-dcf3c9ce8aa0",
"name": "guardar_sesion"
},
{
  "parameters": {
    "url": "https://zkvnvsnjaeawafuzvwcc.supabase.co/rest/v1/sesiones",
    "sendQuery": true,
    "queryParameters": {
      "parameters": [
        {
          "name": "chat_id",

```

```

        "value": "=eq.{{ $fromAI('chat_id') }}"
    }
]
},
"sendHeaders": true,
"headerParameters": {
    "parameters": [
        {
            "name": "apikey",
            "value":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Inprdm52c25qYWVhd2FmdXp2d2Nj
jIiwicm9sZSI6ImFub24iLCJpYXQiOiJlE3NzU1MzI5NjEsImV4cCI6MjA5MTEwODk2MX0.2gGnYcKBcdBnFO2Dd3lCeV90ux_0
zdrOUjHwXYq1gBQ"
        },
        {
            "name": "Authorization",
            "value": "Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Inprdm52c25qYWVhd2FmdXp2d2Nj
Iiwicm9sZSI6ImFub24iLCJpYXQiOiJlE3NzU1MzI5NjEsImV4cCI6MjA5MTEwODk2MX0.2gGnYcKBcdBnFO2Dd3lCeV90ux_0z
drOUjHwXYq1gBQ"
        }
    ]
},
"options": {},
},
"type": "n8n-nodes-base.httpRequestTool",
"typeVersion": 4.4,
"position": [
    560,
    576
],
"id": "03989919-5a71-447d-82b1-0ca069546ac2",
"name": "obtener_sesion"
}
],
"pinData": {},

```

```
"connections": {
  "AI Agent": {
    "main": [
      [
        {
          "node": "detectar_comprobante",
          "type": "main",
          "index": 0
        }
      ]
    ]
  },
  "detectar_comprobante": {
    "main": [
      [
        {
          "node": "enviar_correo_pedido",
          "type": "main",
          "index": 0
        }
      ],
      [
        {
          "node": "Send a text message",
          "type": "main",
          "index": 0
        }
      ]
    ]
  },
  "Telegram Trigger": {
    "main": [
      [
        {
          "node": "Set",
          "type": "main",

```

```
        "index": 0
      }
    ]
  ]
},
"Simple Memory": {
  "ai_memory": [
    [
      {
        "node": "AI Agent",
        "type": "ai_memory",
        "index": 0
      }
    ]
  ]
},
"crear_pedido": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"buscar_repuesto": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
}
```

```
]
},
"registrar_cliente": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"Set": {
  "main": [
    [
      {
        "node": "AI Agent",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Send a text message": {
  "main": [
    []
  ]
},
"enviar_correo_pedido": {
  "main": [
    [
      {
        "node": "Send a text message",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
```

```
    }
  ]
]
},
"validar_cliente": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"Anthropic Chat Model": {
  "ai_languageModel": [
    [
      {
        "node": "AI Agent",
        "type": "ai_languageModel",
        "index": 0
      }
    ]
  ]
},
"guardar_sesion": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
}
```

```
    },
    "obtener_sesion": {
      "ai_tool": [
        [
          {
            "node": "AI Agent",
            "type": "ai_tool",
            "index": 0
          }
        ]
      ]
    },
    "active": true,
    "settings": {
      "executionOrder": "v1",
      "binaryMode": "separate"
    },
    "versionId": "2cd815bb-752a-4942-b784-f8e0f5dff102",
    "meta": {
      "templateCredsSetupCompleted": true,
      "instanceId": "e556ec9d6ce7f86e686947f910bfab3f98bd71c4fa221baa5cbd4e1ddc12bee2"
    },
    "id": "P9yMWwMoiQSyoeLP",
    "tags": []
  }
}
```

## Referencias

Anthropic. (2026). Claude API documentation. <https://docs.anthropic.com>

n8n GmbH. (2026). n8n documentation. <https://docs.n8n.io>

Supabase. (2026). Supabase documentation. <https://supabase.com/docs>

Telegram. (2026). Telegram bot API. <https://core.telegram.org/bots/api>