Jano.

Checklist: tu primer agente de IA en n8n

Introducción

Con este checklist vas a poder armar y revisar un agente como el del video: con su modelo, una memoria separada por persona, herramientas y un subflujo, atendiendo en Telegram. Es la primera versión que le mandás a tu cliente para que la pruebe. Tenelo abierto mientras armás el tuyo y repasalo entero antes de publicar y antes de mandársela.

Importante

La API key de OpenAI y el token de tu bot son contraseñas. En n8n van solo dentro de las credenciales: no los muestres en pantalla ni los pegues en un chat. Si alguna se llega a ver, generá una nueva y dejá sin efecto la anterior.

Sección 1: Las piezas del agente

1. **Chat Model** conectado. Es la única pieza obligatoria: sin modelo, el agente no corre.
2. En OpenAI Chat Model, **Model** en gpt-4.1 y **Sampling Temperature** en 0 para atender clientes. Si elegís un modelo que razona antes de responder, n8n te muestra **Reasoning Effort**: ajustá ese y no agregues temperatura, porque esos modelos la rechazan.
3. **Simple Memory** con **Session ID** en **Define below** y, en Key, un dato único por persona. En Telegram es el id del chat: {{ $('Telegram Trigger').item.json.message.chat.id }}. Nunca el id del mensaje ni el de la ejecución: cambian en cada mensaje y el agente no recuerda nada. Simple Memory alcanza para la versión de prueba; en producción, la memoria va en una base de datos (video de agentes avanzado).
4. **Context Window Length** en 10. Son los intercambios que recuerda: más alto, recuerda más y cada mensaje sale más caro.
5. **Prompt (User Message)** en Define below, con el mensaje entre comillas: Mensaje del usuario: "{{ $json.message.text }}".
6. **System Message** con tu prompt: Rol, Tarea, Herramientas e Importante (el modelo está al final de este documento). Si el prompt lleva la fecha, el campo va en modo **Expression**: Hoy es {{ $now.setLocale('es').toFormat("cccc d 'de' LLLL 'de' yyyy, HH:mm") }}.
7. La zona horaria del workflow, en los tres puntos de arriba, **Settings** y **Timezone**. Si no la elegís, n8n usa la de su instalación, que por defecto es la de Nueva York.
8. Cada herramienta con un nombre claro y sin tildes ni ñ (el nombre del nodo es el que ve el agente) y una **Description** que diga qué hace y qué datos necesita.

Sección 2: Qué nodo usar

1. **Nodo de OpenAI (Message a model):** para tareas de una sola vuelta, como redactar un presupuesto o clasificar un mensaje. No tiene memoria: cada vez que corre, arranca de cero. Si conectás otro nodo después, el texto está en {{ $json.output[0].content[0].text }}. Los tutoriales viejos usan message.content, que en esta versión no existe. También va al lado de un agente, para una tarea fija: por ejemplo, partir la respuesta en mensajes cortos, algo que pedírselo al agente en el prompt no siempre alcanza.
2. **AI Agent:** para conversar con una persona y acordarse de lo que hablaron.
3. **Herramienta (nodo redondo):** cuando el agente decide si la usa y cuándo.
4. **Paso (nodo cuadrado):** cuando algo tiene que pasar siempre, como mandar la respuesta por Telegram.
5. **Subflujo:** cuando la acción tiene varios pasos o una cuenta que tiene que dar exacta. El agente decide cuándo pedirla; la hace siempre el mismo workflow.

Sección 3: Un subflujo como herramienta

1. Workflow nuevo con el disparador **When Executed by Another Workflow**, en **Define using fields below**, con cada dato que va a recibir y su tipo (String, Number). Nombres sin tildes, sin ñ y sin espacios: son las claves que completa el modelo.
2. La respuesta del subflujo es lo que sale del **último nodo que corrió**: armá ahí los datos que necesita el agente.
3. Las reglas del negocio van en el subflujo: para el stock, un **If** antes de calcular (con **Convert types where required**) y, en la otra salida, un mensaje para el agente: "no hay stock, ofrecé otro producto".
4. Si usaste datos de prueba (set mock data), sacalos con **Unpin** antes de conectarlo.
5. **Publicá el subflujo, y de nuevo cada vez que lo cambies.** Cuando el agente atiende de verdad, usa la última versión publicada; en las pruebas desde el editor, la última guardada. n8n te deja publicar el agente con el subflujo sin publicar, pero en producción la herramienta falla.
6. En el agente: **Call n8n Workflow Tool**, renombrado con el nombre de la herramienta, con su Description, el workflow elegido en **Workflow** y cada dato en **Let the model define this parameter** (n8n lo guarda como una expresión fromAI). Si los datos no aparecen, tocá **Refresh**.
7. Para revisar qué pasó: en **Logs**, abrí la herramienta y tocá **View sub-execution**. En el primer nodo del subflujo ves los datos que armó el modelo.

Sección 4: Telegram de punta a punta

1. Creá el bot con **BotFather**: /newbot, un nombre y un usuario que termine en "bot". El token que te devuelve va en la credencial de Telegram, en **Access Token**.
2. Tu n8n tiene que tener HTTPS: Telegram no le manda mensajes a una dirección sin HTTPS.
3. Disparador de Telegram **On message**. Tocá **Listen for test event** y mandale un mensaje a tu bot: con esos datos de prueba arrastrás el id del chat y el texto al resto de los nodos.
4. **Send a text message** después del agente: en Chat ID, el id del chat del disparador; en Text, el output del agente; en Additional Fields, **Append n8n Attribution** desactivado.

5. Probá con **Execute workflow** antes de publicar. Después de publicar, probá solo desde Telegram, y si cambiás algo, volvé a publicar.
6. Un bot, un workflow publicado: Telegram le manda los mensajes de cada bot a una sola dirección.
7. Los audios, las fotos y los stickers no traen texto: al agente le llega el mensaje vacío. Si tu cliente recibe audios, hace falta un paso que los pase a texto antes del agente.

Sección 5: Antes de publicar y de mandársela al cliente

1. El subflujo está publicado, con su último cambio.
2. La memoria usa el id de cada chat, no un valor fijo.
3. El prompt pide texto plano, sin asteriscos ni guiones bajos.
4. Si el prompt usa la fecha: System Message en Expression y la zona horaria del negocio.
5. Probaste cinco mensajes: un saludo, una cotización, una pregunta de seguimiento ("¿y si fueran 20?"), un producto que no existe y una pregunta fuera de tema. Para repetirlos en automático cada vez que lo cambies, usá el sistema de testing del video 11: la carpeta casos trae agente-telegram.md.
6. La respuesta llega sin la frase "This message was sent automatically with n8n".
7. Lo probó alguien más, además de vos. Cada cosa que sale al cliente se prueba antes.
8. Le mandaste al cliente el link del bot (t.me/ y el usuario del bot), cinco cosas para probar y qué hacer si algo no le gusta: una captura del chat y lo que hubiera querido que diga.
9. Con su OK, se pasa a su canal: cambiás el disparador y el nodo que manda la respuesta; el agente queda igual.

Sección 6: Errores y cómo salir

Ordenados por pieza: el modelo (1 a 3), la memoria (4 a 7), el prompt (8 y 9), las herramientas y el subflujo (10 a 14) y Telegram (15 a 18).

1. No node connected to required input "Chat Model"

Falta conectar el modelo al agente. Tocá el + de Chat Model y agregá uno.

2. You exceeded your current quota

Tu cuenta de OpenAI no tiene crédito (n8n lo muestra también como "Insufficient quota detected"). Cargalo en platform.openai.com, en Billing. La suscripción a ChatGPT no cuenta: la API se paga aparte.

3. Unsupported parameter: 'temperature'

Elegiste un modelo que razona antes de responder, y esos no aceptan temperatura. Sacá Sampling Temperature de las opciones y, si hace falta, ajustá Reasoning Effort.

4. No session ID found

La memoria quedó en Connected Chat Trigger Node y tu disparador no manda ese dato. Pasala a Define below y usá el id del chat.

5. Key parameter is empty

La expresión de Key quedó vacía. Arrastrá el id del chat desde el disparador, así la expresión nombra ese nodo aunque en el medio agregues otros.

6. El bot le cuenta a una persona lo que habló otra

El Session ID es el mismo para todos: un valor fijo, o una expresión que quedó en modo Fixed y n8n lee como texto. Usá el id del chat, en modo Expression.

7. El bot se olvida de golpe

Simple Memory vive dentro de n8n: se borra si n8n se reinicia o si la persona pasa más de una hora sin escribir. Para un cliente, usá una memoria guardada en una base de datos (video 15).

8. No prompt specified

El agente quedó en Connected Chat Trigger Node, que espera el chat de prueba de n8n. Con otro disparador, elegí Define below en Source for Prompt (User Message).

9. El agente no sabe qué día es, o se equivoca de día

El System Message quedó en Fixed, así que al modelo le llegan las llaves tal cual, o la zona horaria del workflow no es la del negocio. Pasalo a Expression y revisá Settings → Timezone.

10. None of your tools were used in this run

El agente respondió sin usar herramientas. Mejorá la Description y decile en el prompt cuándo usarla ("Usala siempre que…"). En Logs confirmás si la usó.

11. Received tool input did not match expected schema

Al modelo le faltó un dato de la herramienta. Decile en el prompt qué usar cuando falta, como la moneda en dólares si el cliente no la dice.

12. Workflow is not active and cannot be executed

El agente publicado llamó a un subflujo sin publicar. Publicá el subflujo.

13. The workflow did not return a response

El subflujo no llegó a su último nodo con datos: por ejemplo, el producto no está en la planilla o llegó escrito distinto. El filtro de Google Sheets compara el texto exacto.

14. The column "producto" could not be found

El nombre de la columna del filtro no coincide con el de la planilla. Escribilo igual, con las mismas mayúsculas.

15. can't parse entities

La respuesta traía un asterisco o un guion bajo suelto y Telegram lo quiso leer como formato. Pedí texto plano en el prompt.

16. An HTTPS URL must be provided for webhook

Tu n8n no tiene HTTPS, o no está en uno de los puertos que acepta Telegram (443, 80, 88 u 8443). Configurá el dominio con HTTPS (video 2) y volvé a probar.

17. El bot publicado dejó de responder

Probaste con Execute workflow después de publicar, y Telegram le mandó los mensajes al editor. Despublicá el workflow y volvé a publicarlo.

18. Cambiaste el prompt y el bot responde igual

Cuando atiende de verdad, el bot usa la última versión publicada. Tocá Publish.

Sección 7: Palabras clave

Simple Memory
Memoria que guarda la conversación dentro de n8n.

Session ID
El identificador de cada conversación. La memoria se guarda separada por ese dato.

System Message
Tus instrucciones fijas: viajan iguales en cada mensaje.

Expression
El modo de un campo en el que n8n reemplaza lo que va entre llaves dobles.

Subflujo
Un workflow que otro workflow llama, le pasa datos y recibe una respuesta.

Sección 8: Tu prompt base

Cómo usarlo: cambiá el negocio, las tareas y las herramientas por las de tu cliente y pegalo en el System Message. Los nombres entre comillas, iguales a los de los nodos. Escribilo en el registro del agente: si habla de vos, el prompt va en voseo.

```
# Rol
Sos el asistente virtual de la Ferretería El Tornillo. Atendés a los clientes por Telegram.

# Tarea
- Cotizá los productos de la ferretería.
- Respondé dudas generales sobre herramientas y materiales.
- Respondé en español rioplatense, en dos o tres oraciones.
- Escribí en texto plano: sin asteriscos, sin negritas y sin listas con viñetas.

# Herramientas
## "cotizar"
Usala siempre que el cliente pregunte un precio o quiera saber cuánto le sale un pedido. Necesitás el producto, la cantidad y la moneda: si no dice la moneda, usá dólares (USD). Cuando termine, respondé con el total que te devuelve y aclarale si tuvo descuento.
## "Wikipedia"
Usala siempre que el cliente tenga una duda general sobre una herramienta o un material, por ejemplo qué es un nivel de burbuja. Buscá en inglés, porque esta herramienta consulta la Wikipedia en inglés, y respondé en español.

# Importante
- Nunca inventes precios: usá solo lo que devuelve "cotizar". Si "cotizar" no encuentra el producto, decile al cliente que no lo tenemos en la lista.
- Nunca respondas consultas que no tengan que ver con la ferretería. Si te preguntan otra cosa (por ejemplo, la distancia a la Luna), decilo con buena onda y volvé a lo tuyo.
- Si te preguntan si sos una persona o una IA, decí con naturalidad que sos el asistente virtual de la Ferretería El Tornillo.
```

Descripción de la herramienta "cotizar" (va en el campo Description del nodo):

```
Cotiza la compra de un producto de la ferretería. Productos (pasá el nombre tal cual, en singular): martillo, destornillador, taladro, serrucho, pinza, llave inglesa, cinta aislante, nivel, candado y lija. La moneda va en código de tres letras, por ejemplo USD, ARS, MXN o COP. Devuelve el precio unitario, el descuento y el total en esa moneda.
```

Ejemplo de resultado: un cliente escribe "¿Cuánto me salen 12 martillos en pesos argentinos?". El agente llama a "cotizar" con producto martillo, cantidad 12 y moneda ARS, y responde con el total del subflujo, con el 10 % de descuento por ser 10 o más.

Clases filtradas de n8n · Jano Llosa · @one.jano · jano.my. Usalo libre en tus proyectos y con tus clientes; si lo compartís, nombrá la fuente.