

Tool Calling: cómo el agente pide ayuda a otros sistemas

Datos de esta lección

Fase	F3 — Conexión con el mundo real y automatización no visual
Módulo	M3.4 — Tool Calling
Puntos de profesionalidad al superar el test	30

Objetivo práctico

Al terminar esta lección entenderás con precisión el mecanismo completo de Tool Calling: cómo el LLM decide activar una herramienta, qué datos le pide al alumno durante la conversación, cómo esos datos viajan hasta tu automatización, y qué formato de respuesta espera cada plataforma para poder seguir hablando con naturalidad.

Resultado que obtendrás

Un modelo mental completo y verificado del ciclo LLAMADA → AGENTE → SOLICITA CITA → TOOL CALL → WEBHOOK → N8N/MAKE → CALENDARIO → RESPUESTA → CONFIRMACIÓN HABLADA, que aplicarás en las dos lecciones siguientes con Vapi+n8n y con Retell+Make.

Dónde encaja dentro del sistema final

Este módulo conecta, de forma literal, todas las piezas que has construido en fases anteriores: AGENTE (Fase 2), AUTOMATIZACIÓN (módulo 3.3) y CALENDARIO (módulo 3.2). Tool Calling es el mecanismo técnico exacto que las une.

Conocimientos previos

- El mensajero interno de la oficina: Tool Calling como concepto (módulo 1.2, lección «El flujo completo»).
- Webhooks (módulo 3.1).
- n8n o Make ya configurados con el Blueprint del curso (módulo 3.3).

Herramientas necesarias

- Un navegador con NVDA activo.

Posible coste

Esta lección es conceptual y no requiere ningún gasto adicional a las cuentas ya creadas.

Modelo mental

Piensa en un telefonista que, en mitad de una llamada, necesita comprobar la disponibilidad real antes de prometer una cita. No inventa un hueco: descuelga el teléfono interno, pregunta al departamento de agenda, espera la respuesta, y sólo entonces se lo confirma a quien llama. Tool Calling es exactamente ese teléfono interno.

Explicación

Cuando configuras un LLM con una herramienta (tool/function), le estás dando, además del prompt, una descripción técnica de una acción que puede solicitar: un nombre (por ejemplo, `reservar_cita`), una descripción en lenguaje natural de para qué sirve (que el LLM usa para decidir cuándo activarla), y un **esquema de parámetros** en JSON que especifica qué datos necesita pedir a quien llama antes de poder ejecutarla (por ejemplo, nombre, teléfono, servicio, fecha y hora deseadas).

Durante la conversación, el LLM no ejecuta la acción directamente: sólo detecta que hacen falta esos datos, los pide de forma natural dentro de la conversación («¿me confirmas tu nombre y qué día te vendría bien?»), y cuando los tiene todos, genera una **llamada a la herramienta** (tool call) con esos datos ya estructurados en JSON. La plataforma de orquestación (Vapi o Retell) envía entonces esos datos mediante un Webhook a la URL que configuraste (tu automatización de n8n o Make), exactamente como practicaste en el módulo 3.3.

Tu automatización procesa la petición (en este curso, crea el evento en el calendario) y **debe responder con un formato JSON concreto**, específico de cada plataforma: Vapi espera un objeto con un array `results`, cada uno con un `toolCallId` (que identifica exactamente qué llamada se está respondiendo) y un `result` con el texto de resultado; Retell espera simplemente un código de estado entre 200 y 299 y un cuerpo (texto o JSON) que se convierte en la información que el LLM usa para continuar hablando. En ambos casos, esa respuesta es lo que permite al agente decir,

con naturalidad, «Perfecto, tu cita ha quedado confirmada para el jueves a las diez», en lugar de quedarse en silencio o inventar una confirmación sin haberla comprobado.

Vocabulario nuevo

Esquema de parámetros (JSON Schema)

Piénsalo así: La lista de casillas que un formulario exige rellenar antes de poder tramitarlo.

Definición en JSON de qué datos necesita una herramienta, con su nombre, tipo y si son obligatorios.

Ejemplo: La herramienta «reservar_cita» puede exigir, como parámetros obligatorios, nombre, telefono y fecha_hora_inicio.

toolCallId

Piénsalo así: El número de resguardo de un ticket, que permite emparejar la respuesta con la solicitud exacta que la originó.

Identificador único que Vapi asigna a cada llamada a una herramienta, y que la respuesta del Webhook debe incluir para que el sistema sepa a qué solicitud corresponde.

Ejemplo: Si el agente activa dos herramientas seguidas, cada una tendrá su propio toolCallId para no confundir sus respuestas.

Preparación

No se requiere preparación técnica. Ten a mano tu documento de configuración técnica del módulo 1.1.

Procedimiento paso a paso

1. **Contexto:** Vas a repasar, con tus propias palabras y usando NVDA para releer secciones concretas, el ciclo completo de Tool Calling antes de implementarlo.

Acción de teclado: Vuelve a la lección «El flujo completo» del módulo 1.2 (usa el Buscador con el término «Tool Calling») y localiza, con **Ctrl** + **F**, el párrafo

que describe el mecanismo.

Respuesta esperada de NVDA: NVDA debería anunciar el texto resaltado por la búsqueda del navegador.

Qué significa: Confirmas que el concepto introducido en el módulo 1.2 y el mecanismo técnico exacto de esta lección son la misma idea, ahora con más detalle.

Acción siguiente: Vuelve a esta lección y continúa.

2. **Contexto:** Vas a examinar el formato exacto de respuesta que Vapi espera, comparándolo con el que genera tu Blueprint de n8n del módulo 3.3.

Acción de teclado: Abre el Blueprint «n8n-webhook-cita-calendario.json» descargado en el módulo 3.3 con un editor de texto, y localiza el nodo «Responder al agente» y su campo «responseBody».

Respuesta esperada de NVDA: NVDA leerá el contenido del archivo JSON línea a línea, igual que practicaste en el módulo 3.1.

Qué significa: Verás que la respuesta actual del Blueprint es un JSON sencillo, no todavía el formato específico que exige Vapi (con results y toolCallId); ajustarlo es exactamente la tarea de la siguiente lección.

Acción siguiente: Continúa con la siguiente lección de este módulo, donde adaptarás esta respuesta.

Errores frecuentes y diagnóstico

El agente pide un dato, lo recibe, pero nunca llega a ejecutar la herramienta.

Cómo localizarlo: Puede deberse a que el esquema de parámetros marca ese dato como opcional cuando debería ser obligatorio, o a que la descripción de la herramienta no deja claro al LLM cuándo debe activarla.

Solución: Revisa el esquema de parámetros y la descripción de la herramienta, no la automatización: si la herramienta nunca llega a activarse, el problema está antes del Webhook.

La herramienta se ejecuta correctamente (aparece la cita en el

calendario) pero el agente se queda en silencio o repite la pregunta.

Cómo localizarlo: El problema está casi siempre en el formato de la respuesta del Webhook: si no coincide exactamente con lo que la plataforma espera (por ejemplo, falta el `toolCallId` en Vapi), el agente no puede interpretar el resultado aunque la acción se haya completado.

Solución: Compara la respuesta real de tu automatización con el formato exacto documentado de la plataforma (Vapi o Retell) y corrige la estructura JSON.

Comprobación final

Explica en voz alta, con tus propias palabras, qué pasos ocurren exactamente entre que el LLM decide que hace falta reservar una cita y que el agente confirma esa cita en voz alta a quien llama. Si tu explicación incluye el esquema de parámetros, el Webhook y el formato de respuesta, la comprobación es positiva.

Qué acabas de conseguir

Tienes el modelo mental técnico completo de Tool Calling, necesario para implementar con criterio los dos casos completos de las siguientes lecciones: Vapi + n8n y Retell + Make.

Ejercicio práctico

Diseña, sobre papel o en un documento de texto, el esquema de parámetros (nombre de cada dato y si es obligatorio) que necesitaría una herramienta «cancelar_cita», distinta de «reservar_cita», pensando en qué datos mínimos hacen falta para identificar sin ambigüedad qué cita cancelar.

Resumen de lo imprescindible

- Una herramienta (tool) se define con un nombre, una descripción y un esquema de parámetros en JSON.
- El LLM detecta cuándo activar la herramienta y pide, dentro de la conversación, los datos que le faltan.
- La respuesta del Webhook debe seguir el formato exacto que exige cada plataforma: Vapi (`results + toolCallId`) o Retell (código 200-299 y cuerpo con el resultado).
- Si la herramienta nunca se activa, el problema está en el esquema/descripción; si se activa pero el agente no sabe interpretar el resultado, el problema está en

el formato de la respuesta.

No necesitas aprender todavía

Todavía no necesitas configurar la herramienta dentro del panel de Vapi o Retell: eso se hace paso a paso en las dos lecciones siguientes.

Estoy preparado para continuar si puedo...

- Explicar el ciclo completo de Tool Calling, desde la decisión del LLM hasta la confirmación hablada.
- Diferenciar el formato de respuesta exacto que exige Vapi del que exige Retell.
- Diagnosticar si un fallo de Tool Calling está en el esquema de parámetros o en el formato de respuesta.

Test de comprobación

Para registrar el resultado y obtener Puntos de profesionalidad, realiza este test desde la versión web de la lección. Las opciones se muestran aquí sin señalar la respuesta correcta.

1. El agente crea correctamente la cita en Google Calendar (puedes verla en el calendario real), pero durante la llamada se queda en silencio en lugar de confirmarla en voz alta. ¿Qué es lo más probable, según esta lección?
 - a. La automatización ha respondido con un formato JSON que no coincide con el que espera la plataforma (por ejemplo, falta el toolCallId en Vapi), por lo que el agente no puede interpretar el resultado aunque la acción se haya completado.
 - b. El número de Twilio ha dejado de funcionar.
 - c. El STT ha transcrito mal la petición inicial.
2. Configuras una herramienta «reservar_cita» pero, durante varias llamadas de prueba, el agente nunca llega a activarla, ni siquiera cuando el cliente pide claramente una cita. ¿Dónde tiene más sentido buscar el problema primero?
 - a. En la descripción de la herramienta y su esquema de parámetros, que son los que guían al LLM sobre cuándo y cómo activarla.
 - b. En la configuración del calendario de Google.
 - c. En el número de teléfono de Twilio.
3. ¿Para qué sirve exactamente el toolCallId en la respuesta que envías a Vapi?

- a. Para identificar a qué llamada concreta de herramienta corresponde esa respuesta, especialmente si se activa más de una herramienta a la vez.
- b. Para identificar al cliente que ha llamado.
- c. Para autenticar la cuenta de Twilio.

Fuentes

- [Introduction to Tools — Vapi \(consultado el 2026-08-30\)](#)
- [Function Calling Overview — Retell AI \(consultado el 2026-08-30\)](#)

Última actualización de este contenido: **2026-08-30**.