

Caso completo: Retell + Make reservando una cita

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	55

Objetivo práctico

Al terminar esta lección tendrás una Custom Function real configurada en tu agente de Retell, conectada mediante Webhook con el escenario de Make del módulo 3.3, extendido con un módulo de Google Calendar, y probada con una llamada de prueba real.

Resultado que obtendrás

Un ciclo completo funcional: el agente de Retell pide los datos de una cita, ejecuta la función, Make crea el evento en Google Calendar, y el agente confirma la cita en voz alta a partir de la respuesta recibida.

Dónde encaja dentro del sistema final

Esta lección implementa, con Retell y Make, el segundo de los dos casos obligatorios de Tool Calling de este curso (el primero, Vapi + n8n, se implementó en la lección anterior).

Conocimientos previos

- Tool Calling: cómo el agente pide ayuda a otros sistemas (módulo 3.4).
- Ensamblaje del agente en Retell AI (módulo 2.3).
- Automatización accesible con Make (módulo 3.3), con el escenario del Blueprint ya activo.

Herramientas necesarias

- Un agente de Retell ya creado (módulo 2.3).
- El escenario de Make del módulo 3.3, activo, con su URL de Webhook.
- Un navegador con NVDA activo.

Posible coste

Esta lección no genera coste adicional a las cuentas ya creadas. Las llamadas de prueba consumirán minutos y tokens según lo explicado en el módulo 2.1.

Modelo mental

El mismo botón de marcación rápida hacia el departamento de agenda del módulo anterior, ahora instalado en la centralita de Retell en lugar de en la de Vapi: el objetivo es idéntico, el procedimiento concreto cambia ligeramente.

Explicación

Según la documentación oficial de Retell AI verificada el 30 de agosto de 2026, las Custom Functions se añaden directamente dentro de la configuración de cada agente, en su sección «Functions», con el botón «+ Add» y la opción «Custom Function». Cada función requiere un nombre único (con guion bajo, por ejemplo reservar_cita), una descripción clara de cuándo debe usarse, el método HTTP (POST por defecto), la URL pública del endpoint (tu Webhook de Make; Retell bloquea direcciones locales o privadas por seguridad, así que la URL debe ser pública, como lo es la de Make), y el esquema de parámetros.

A diferencia de Vapi, Retell no exige un formato de respuesta JSON tan estricto: basta con devolver un código de estado entre 200 y 299, con un cuerpo (texto o JSON) que se convierte en información que el LLM usa para seguir hablando; si el cuerpo es JSON, además puedes extraer campos concretos mediante notación de puntos como «variables de respuesta» reutilizables en la conversación. Retell ofrece también dos ajustes de comportamiento útiles para una llamada real: «Talk While Waiting» (el agente puede seguir hablando mientras la función se ejecuta, útil si la automatización tarda unos segundos) y «Talk After Action Completed» (el agente continúa hablando de forma natural una vez recibida la respuesta).

El escenario de Make del módulo 3.3, tal y como está, registra la cita en una hoja de cálculo. En esta lección lo extiendes añadiendo un módulo de «Google Calendar» (buscándolo por nombre, como practicaste en el módulo 3.3), para que, además del

registro, la cita quede creada en el calendario real.

Vocabulario nuevo

Custom Function (Retell)

Piénsalo así: El botón de marcación rápida instalado en la centralita de Retell.

Herramienta configurada directamente en un agente de Retell, con nombre, descripción, método HTTP, URL y esquema de parámetros, que activa un Webhook durante la conversación.

Ejemplo: La Custom Function «reservar_cita» de Retell envía los datos de la cita al Webhook de Make.

Variable de respuesta (Retell)

Piénsalo así: Anotar, tras la llamada al departamento de agenda, el dato concreto de la respuesta que interesa recordar durante el resto de la conversación.

Campo extraído de la respuesta JSON de una Custom Function, mediante notación de puntos, reutilizable en el resto de la conversación.

Ejemplo: Extraer cita.hora_confirmada como variable de respuesta permite que el agente la mencione más adelante sin volver a consultarla.

Preparación

Ten a mano la URL de tu escenario de Make (módulo 3.3) y el acceso a tu agente de Retell (módulo 2.3).

Procedimiento paso a paso

1. **Contexto:** Vas a añadir un módulo de Google Calendar a tu escenario de Make, ampliando el Blueprint del módulo 3.3.

Acción de teclado: En tu escenario de Make, localiza el botón «+» para añadir un módulo nuevo después del módulo de Google Sheets. Escribe «Google Calendar» en el buscador de aplicaciones y selecciona la acción «Create an Event».

Respuesta esperada de NVDA: NVDA debería anunciar los resultados de la búsqueda de texto y, al seleccionar, el formulario de configuración del nuevo módulo.

Qué significa: Este módulo se añade mediante un buscador de texto estándar, no mediante arrastre, siguiendo el criterio de accesibilidad del módulo 3.3.

Acción siguiente: Conecta tu cuenta de Google Calendar (OAuth, módulo 3.2), selecciona tu calendario, y mapea los campos de fecha, hora y título usando los datos ya interpretados por el módulo «json:ParseJSON» anterior.

2. **Contexto:** Vas a crear la Custom Function en tu agente de Retell.

Acción de teclado: En el Dashboard de Retell, abre tu agente y localiza la sección «Functions». Activa «+ Add» y selecciona «Custom Function».

Respuesta esperada de NVDA: NVDA debería anunciar el formulario: nombre, descripción, método HTTP, URL, y editor de parámetros.

Qué significa: Este formulario es un formulario HTML estándar.

Acción siguiente: Rellena el nombre (reservar_cita), la descripción, elige el método POST, y pega la URL del Webhook de tu escenario de Make.

3. **Contexto:** Vas a definir el esquema de parámetros y el comportamiento de habla del agente durante la ejecución.

Acción de teclado: En el editor de parámetros, añade los campos nombre, telefono, servicio, fecha_hora_inicio y fecha_hora_fin como propiedades de tipo texto, con sus descripciones. Después, localiza los interruptores «Talk While Waiting» y «Talk After Action Completed».

Respuesta esperada de NVDA: NVDA debería anunciar cada interruptor y su estado actual (activado/desactivado).

Qué significa: Activar «Talk While Waiting» evita silencios largos si Make tarda unos segundos en responder.

Acción siguiente: Guarda la función y confirma en el prompt del agente una instrucción explícita sobre cuándo debe usarla.

Posibles diferencias de interfaz

Los nombres exactos de los campos pueden variar ligeramente entre versiones de Retell y de Make. El principio estable es siempre el mismo: una función con nombre, descripción, método HTTP, URL y esquema de parámetros en Retell, conectada a un escenario de Make con un módulo que ejecuta la acción real.

Errores frecuentes y diagnóstico

Retell rechaza la URL del Webhook con un error de conexión.

Cómo localizarlo: Retell bloquea explícitamente direcciones localhost o de red privada por seguridad; la URL debe ser pública.

Solución: Confirma que estás usando la URL pública real de tu escenario de Make, no una dirección de pruebas local.

La cita se registra en la hoja de cálculo pero no aparece en Google Calendar.

Cómo localizarlo: El módulo de Google Calendar puede no estar correctamente mapeado, o puede haberse añadido en un orden incorrecto dentro del escenario.

Solución: Revisa, en el historial de ejecuciones de Make, que el módulo de Google Calendar se ejecutó después del de interpretación del JSON, y que sus campos de fecha y hora están correctamente mapeados.

Comprobación final

Realiza una llamada de prueba real pidiendo una cita con datos inventados, y confirma que el agente responde en voz alta confirmando la cita, y que esa cita aparece tanto en tu hoja de registro como en tu Google Calendar real. Si ambas cosas ocurren, la comprobación es positiva.

Qué acabas de conseguir

Tienes el segundo caso completo de Tool Calling de este curso funcionando de extremo a extremo: Retell pidiendo datos, Make creando la cita real, y el agente confirmándolo con naturalidad. Con los dos casos completados, dominas Tool Calling en las dos combinaciones principales de plataformas de este curso.

Ejercicio práctico

Activa «Talk While Waiting» si no lo habías hecho, añade una frase de espera («Dame un momento mientras confirmo tu cita») y comprueba en una llamada de prueba que el agente la dice mientras Make procesa la petición.

Resumen de lo imprescindible

- Las Custom Functions de Retell se configuran directamente dentro de cada agente, en su sección Functions.
- Retell exige sólo un código de estado 200-299 y un cuerpo de respuesta, más flexible que el formato estricto de Vapi.
- El módulo de Google Calendar se añade a un escenario de Make buscándolo por nombre, sin necesidad de conocer su identificador técnico.
- «Talk While Waiting» evita silencios incómodos si la automatización tarda unos segundos en responder.

No necesitas aprender todavía

Todavía no necesitas gestionar casos límite (silencios, interrupciones, cancelaciones): se trabajan en la Fase 4.

Estoy preparado para continuar si puedo...

- Añadir un módulo de Google Calendar a un escenario de Make buscándolo por nombre.
- Crear una Custom Function en Retell con su esquema de parámetros y URL.
- Configurar el comportamiento de habla del agente durante la ejecución de una función.

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. Configuras la Custom Function de Retell apuntando a una URL local de pruebas en tu propio ordenador y Retell rechaza la conexión. ¿Por qué ocurre esto, según la documentación oficial?
 - a. Retell bloquea por seguridad las direcciones localhost o de red privada; la

- URL debe ser pública, como la de Make.
- b. Retell no admite Custom Functions en ningún caso.
 - c. El método HTTP elegido es incorrecto.
2. ¿Qué diferencia principal hay entre el formato de respuesta que exige Vapi (lección anterior) y el que exige Retell (esta lección)?
- a. Vapi exige un objeto JSON estricto con results y toolCallId; Retell sólo exige un código de estado 200-299 y un cuerpo con el resultado, más flexible.
 - b. Ambas exigen exactamente el mismo formato, sin ninguna diferencia.
 - c. Retell exige un formato más estricto que Vapi.
3. Quieres evitar que el agente se quede en silencio mientras Make procesa la petición durante varios segundos. ¿Qué ajuste de Retell resuelve directamente ese problema?
- a. «Talk While Waiting», que permite al agente seguir hablando mientras la función se ejecuta.
 - b. Cambiar el método HTTP de POST a GET.
 - c. Aumentar la cuota de créditos de ElevenLabs.

Fuentes

- [Integrate any system with custom function — Retell AI \(consultado el 2026-08-30\)](#)

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