Webhooks: cómo viaja la información entre sistemas

Datos de esta lección

Fase	F3 — Conexión con el mundo real y automatización no visual
Módulo	M3.1 — JSON y Webhooks
Puntos de profesionalidad al superar el test	35

Objetivo práctico

Al terminar esta lección habrás recibido y examinado, con una herramienta gratuita real, tu primera petición Webhook, y sabrás explicar qué son la URL, el request, el response, el método POST, el payload y las medidas básicas de autenticación de un Webhook.

Resultado que obtendrás

Una URL de prueba propia en la que habrás recibido al menos una petición real, y la capacidad de leer, con NVDA, los datos que contiene: cabeceras y cuerpo (payload) en formato JSON.

Dónde encaja dentro del sistema final

Esta lección profundiza la pieza que conecta el AGENTE con la AUTOMATIZACIÓN del sistema (introducida como concepto en el módulo 1.2): cuando el Tool Calling del módulo 3.4 dispare una acción real, lo hará exactamente mediante un Webhook como el que vas a examinar aquí.

Conocimientos previos

- El buzón de entrada: qué es un Webhook (módulo 1.2).
- JSON desde cero (lección anterior).

Herramientas necesarias

- Un navegador con NVDA activo.

Posible coste

Esta lección usa exclusivamente herramientas gratuitas de prueba y no genera ningún gasto.

Advertencia de seguridad

La herramienta de prueba de esta lección (webhook.site) genera una URL pública: cualquier persona que conozca esa URL exacta podría ver los datos recibidos en ella durante el tiempo que esté activa. No envíes nunca datos reales de un cliente a esta URL de prueba; usa siempre datos inventados.

Modelo mental

Piensa de nuevo en el buzón de entrada del módulo 1.2: un Webhook es una dirección a la que otro sistema deposita un recado automáticamente. Esta lección te da un buzón de pruebas real y gratuito para ver, con tus propios ojos (o con NVDA), qué aspecto tiene exactamente ese recado cuando llega.

Explicación

Un Webhook completo se compone de varias piezas que ya has visto por separado en lecciones anteriores, ahora unidas:

- **URL:** la dirección exacta a la que se envía el aviso. Cada Webhook tiene la suya, normalmente generada automáticamente por la herramienta que lo recibe (n8n, Make, o la herramienta de pruebas de esta lección).
- **Request (petición):** el mensaje completo que envía el sistema origen, compuesto por un método (normalmente POST), unas cabeceras (headers, con información técnica como el formato de los datos) y un cuerpo (body o payload) con los datos reales, casi siempre en JSON.
- **Payload:** el contenido real del mensaje —los datos del evento—, empaquetado como el JSON que trabajaste en la lección anterior.
- **Response (respuesta):** lo que el sistema receptor devuelve al sistema origen para confirmar que ha recibido la petición, normalmente un código de estado HTTP (por ejemplo, 200 si todo ha ido bien) y, opcionalmente, un cuerpo de respuesta propio.

Sobre la **seguridad** de un Webhook: como la URL puede ser conocida por cualquiera que la consiga, muchos sistemas añaden una capa de autenticación, normalmente un

valor secreto incluido en una cabecera concreta de la petición (una especie de contraseña que sólo conocen el emisor y el receptor legítimos) que el receptor comprueba antes de aceptar los datos como válidos. Vapi y Retell, en el módulo 3.4, permiten configurar este tipo de cabecera secreta en sus Webhooks salientes.

Vocabulario nuevo

Request

Piénsalo así: El sobre completo que llega al buzón, con remite, sello y contenido.

Petición HTTP completa enviada por un sistema a otro, compuesta por método, cabeceras (headers) y cuerpo (body).

Ejemplo: El request que envía Vapi al activar una herramienta incluye, en su cuerpo, los datos de la cita solicitada.

Response

Piénsalo así: El acuse de recibo que se devuelve tras aceptar un recado en el buzón.

Respuesta HTTP que el sistema receptor de un Webhook devuelve al sistema emisor, normalmente con un código de estado.

Ejemplo: Un response con código 200 confirma que n8n ha recibido correctamente los datos de la cita.

Payload

Piénsalo así: El contenido real de la carta dentro del sobre, sin contar el sobre en sí.

Cuerpo de datos de una petición o respuesta HTTP, en este curso casi siempre en formato JSON.

Ejemplo: El payload de un Webhook de confirmación de cita incluye el nombre del cliente, la fecha y la hora.

Cabecera secreta (Webhook)

Piénsalo así: Una contraseña incluida en el propio sobre, que sólo el destinatario legítimo sabe comprobar.

Valor de autenticación incluido en una cabecera HTTP de un Webhook, que el receptor verifica para confirmar que la petición procede realmente del emisor esperado.

Ejemplo: Configurar una cabecera secreta en el Webhook de Vapi evita que alguien pueda enviar datos falsos a tu automatización simplemente conociendo la URL.

Preparación

No necesitas crear ninguna cuenta para esta lección: la herramienta de pruebas genera una URL de un solo uso sin necesidad de registro.

Procedimiento paso a paso

1. **Contexto:** Vas a generar tu propia URL de prueba de Webhook, gratuita y sin necesidad de registro.

   Acción de teclado: Ve a la dirección `webhook.site`. La página genera automáticamente una URL única al cargar.

   Respuesta esperada de NVDA: NVDA debería anunciar un campo de texto de sólo lectura con una dirección similar a «https://webhook.site/» seguida de un identificador largo.

   Qué significa: Esa URL es tu «buzón» de pruebas personal, activo mientras dure la sesión gratuita.

   Acción siguiente: Selecciona y copia esa URL completa con `Ctrl` + `C`.

2. **Contexto:** Vas a enviar una petición real a esa URL para comprobar que funciona, usando otra pestaña del navegador como sistema «emisor» simulado.

   Acción de teclado: Abre una pestaña nueva con `Ctrl` + `T`, pega la URL copiada en la barra de direcciones y pulsa `Enter`.

   Respuesta esperada de NVDA: NVDA debería anunciar el contenido de una página sencilla de confirmación.

   Qué significa: Visitar la URL directamente desde el navegador es, técnicamente, un request de tipo GET, distinto del POST que usará un Webhook real, pero sirve para comprobar que la dirección responde.

   Acción siguiente: Vuelve a la primera pestaña, donde está el panel de

webhook.site.

3. **Contexto:** Vas a examinar la petición recibida en el panel de la herramienta.

   Acción de teclado: En el panel de webhook.site, pulsa H o Tab hasta la lista de peticiones recibidas y abre la más reciente.

   Respuesta esperada de NVDA: NVDA debería anunciar los detalles de la petición: método, cabeceras (headers) y cuerpo (body, vacío en este caso al ser un GET simple).

   Qué significa: Este panel es exactamente el tipo de información que también verás, más adelante, en los registros de n8n o Make cuando reciban un Webhook real de tu agente.

   Acción siguiente: Explora las distintas pestañas o secciones del detalle (cabeceras, cuerpo, información técnica) recorriendo con Tab.

Posibles diferencias de interfaz

Herramientas de prueba de Webhooks como webhook.site cambian su interfaz con cierta frecuencia y pueden aparecer y desaparecer del mercado. El principio estable es siempre el mismo: una URL única, gratuita, que muestra en un panel las peticiones que recibe. Si esta herramienta concreta no está disponible cuando la uses, busca «webhook testing tool» o «webhook tester» para encontrar una alternativa equivalente.

Errores frecuentes y diagnóstico

La petición no aparece en el panel tras visitar la URL.

Cómo localizarlo: Puede deberse a que copiaste la URL de forma incompleta, o a que la sesión de prueba haya caducado.

Solución: Genera una URL nueva recargando la página principal de la herramienta y repite el proceso con la URL completa.

Comprobación final

Confirma que has generado una URL de prueba, has enviado al menos una petición real hacia ella, y has localizado con NVDA, en el panel de resultados, las cabeceras de esa petición. Si lo has conseguido, la comprobación es positiva.

Qué acabas de conseguir

Has visto, con datos reales, la estructura completa de un Webhook: URL, request, cabeceras y payload, la misma estructura que usarán Vapi o Retell para activar tu automatización en el módulo 3.4.

Ejercicio práctico

Investiga, en la documentación de la herramienta que hayas usado, cómo enviar una petición POST con un cuerpo JSON personalizado (por ejemplo, usando el propio formulario de pruebas que suelen incluir estas herramientas) y observa cómo cambia el panel de resultados respecto a la petición GET simple de esta lección.

Resumen de lo imprescindible

- Un Webhook completo tiene URL, request (método, cabeceras, cuerpo) y response.
- El payload es el contenido real de datos, normalmente en JSON.
- Una cabecera secreta permite autenticar que una petición procede realmente del emisor esperado.
- Nunca envíes datos reales de un cliente a una herramienta de pruebas pública.

No necesitas aprender todavía

Todavía no necesitas configurar la cabecera secreta en un Webhook real: se practica en el módulo 3.4 con datos reales del proyecto.

Estoy preparado para continuar si puedo…

- Explicar qué son el request, el response y el payload de un Webhook.
- Generar una URL de prueba y examinar una petición real recibida en ella.
- Explicar para qué sirve una cabecera secreta en un Webhook.

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. Alguien consigue la URL de un Webhook de tu automatización de citas y le envía

datos falsos directamente, sin pasar por tu agente. ¿Qué medida de las explicadas en esta lección está pensada específicamente para evitar que esos datos falsos se acepten como válidos?

   a. Una cabecera secreta que el receptor comprueba antes de aceptar la petición como legítima.
   b. Cambiar el formato del payload de JSON a texto plano.
   c. No existe ninguna medida posible contra este problema.

2. ¿Qué diferencia hay entre el request y el response de un Webhook?

   a. El request es lo que envía el sistema origen (con los datos del evento); el response es lo que devuelve el sistema receptor para confirmar la recepción.
   b. Son exactamente lo mismo, sólo que en distinto idioma técnico.
   c. El response se envía siempre antes que el request.

3. Según la advertencia de seguridad de esta lección, ¿por qué no debes enviar datos reales de un cliente a una herramienta de pruebas pública como webhook.site?

   a. Porque la URL de prueba es pública y cualquiera que la conozca podría ver esos datos mientras la sesión esté activa.
   b. Porque las herramientas de prueba de Webhooks son ilegales en España.
   c. Porque el formato JSON no permite incluir datos reales de personas.

Fuentes

- What is Webhook.site? — Webhook.site (consultado el 2026-08-30)

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