

JSON desde cero: leerlo con NVDA sin miedo

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	30

Objetivo práctico

Al terminar esta lección sabrás leer con soltura, usando NVDA, un JSON anidado con varios niveles y con arrays de objetos, del tipo que verás en los Webhooks reales de la Fase 3, sin necesidad de escribir JSON complejo a mano.

Resultado que obtendrás

Capacidad de leer sin ayuda el ejemplo 3 (objeto anidado) y el ejemplo 4 (array de objetos) del recurso «Ejemplos de JSON comentados», los mismos formatos que recibirás en los Webhooks del módulo 3.4 (Tool Calling).

Dónde encaja dentro del sistema final

Este módulo abre la Fase 3: conectar el sistema con el mundo real. El JSON es el «papel» en el que viaja toda esa información —Webhooks, respuestas de API, configuración de automatizaciones—, por lo que esta lección profundiza en la introducción ya hecha en el módulo 1.2.

Conocimientos previos

- El formulario con casillas: qué es JSON (módulo 1.2).
- El buzón de entrada: qué es un Webhook (módulo 1.2).

Herramientas necesarias

- Un navegador con NVDA activo.
- El recurso «Ejemplos de JSON comentados» (ya descargado en el módulo 1.2, o descárgalo ahora desde Recursos).

Posible coste

Esta lección es conceptual y no requiere ningún gasto.

Modelo mental

Piensa de nuevo en el formulario de papel con casillas del módulo 1.2, pero ahora imagina un formulario dentro de una carpeta, y dentro de esa carpeta, varias fichas idénticas, una por cada cita del día. Eso es exactamente un objeto anidado (la carpeta con datos generales) que contiene un array de objetos (las fichas repetidas de cada cita).

Explicación

En el módulo 1.2 aprendiste el patrón básico de JSON: pares clave-valor, objetos entre llaves, listas (arrays) entre corchetes. Esta lección trabaja con los dos ejemplos más avanzados del recurso «Ejemplos de JSON comentados»: un objeto anidado (un objeto dentro de otro objeto, como los datos de un cliente dentro de la confirmación de una cita) y un array de objetos (una lista donde cada elemento es, a su vez, un objeto completo con sus propias casillas, como ocurre al consultar todas las citas de un día).

La estrategia para leer estos JSON más complejos con NVDA sin perderse es fijarse en la **indentación** (los espacios al principio de cada línea): cuanta más indentación tiene una línea, más «dentro» está de un objeto o de un array concreto. Si pierdes la cuenta de en qué nivel estás, vuelve a la línea con la llave o el corchete de apertura más cercano hacia arriba con menos indentación: ese es el «contenedor» de lo que estás leyendo.

Este formato exacto —objetos anidados y arrays de objetos— es el que recibirás en los Webhooks reales del módulo 3.4, cuando el agente confirme una cita y el sistema de automatización te devuelva los datos completos del cliente y de la cita en una sola respuesta estructurada.

Vocabulario nuevo

Indentación

Piénsalo así: Los distintos niveles de sangría de un índice de documento, que muestran qué apartado está dentro de cuál.

Espacios en blanco al principio de una línea de código que muestran visualmente su

nivel de anidación dentro de una estructura.

Ejemplo: En el ejemplo 3 del recurso de JSON, las casillas de "cliente" están indentadas un nivel más que la clave "cliente" que las contiene.

Objeto anidado

Piénsalo así: Una carpeta dentro de otra carpeta.

Objeto JSON (delimitado por llaves) que aparece como valor de una clave dentro de otro objeto.

Ejemplo: En {"cliente": {"nombre": "Juan"}}, el objeto {"nombre": "Juan"} está anidado dentro de la clave "cliente".

Preparación

Ten abierto el recurso «Ejemplos de JSON comentados» (descárgalo desde [Recursos](#) si todavía no lo hiciste en el módulo 1.2), concretamente los ejemplos 3 y 4.

Procedimiento paso a paso

1. **Contexto:** Vas a leer el ejemplo 3 del recurso (Webhook de confirmación de cita), que contiene un objeto anidado.

Acción de teclado: Con el archivo abierto, sitúate al principio del bloque de código del «Ejemplo 3» y avanza línea por línea con la flecha abajo, fijándote en la indentación de cada línea.

Respuesta esperada de NVDA: NVDA leerá cada línea completa, incluyendo los espacios en blanco al principio si tienes activada la lectura de indentación, o puedes contarlos visualmente si usas también magnificación.

Qué significa: Las líneas "nombre" y "telefono" están más indentadas que "cliente": eso confirma que son casillas del objeto anidado dentro de "cliente", no casillas del objeto principal.

Acción siguiente: Identifica de la misma forma qué casillas pertenecen al objeto anidado "cita".

2. **Contexto:** Vas a leer el ejemplo 4 (varias citas en una sola respuesta), que

combina un array con objetos anidados dentro.

Acción de teclado: Localiza la clave "citas" y, con la flecha abajo, cuenta cuántas veces aparece el patrón de apertura y cierre de llaves dentro de los corchetes de "citas".

Respuesta esperada de NVDA: NVDA leerá cada llave y corchete conforme avanzas, línea por línea.

Qué significa: Cada bloque entre llaves dentro de los corchetes de "citas" es un elemento distinto del array: en este ejemplo, dos citas distintas con la misma estructura de casillas cada una.

Acción siguiente: Cuenta cuántas casillas tiene cada cita individual (hora, cliente, servicio) y confirma que ambos elementos del array siguen la misma estructura.

Errores frecuentes y diagnóstico

Perder la cuenta de a qué objeto pertenece una casilla en un JSON con varios niveles de anidación.

Cómo localizarlo: Es habitual en JSON con tres o más niveles si se intenta leer de memoria en lugar de apoyarse en la indentación.

Solución: Vuelve a la línea con menos indentación inmediatamente anterior: esa línea es el contenedor directo de lo que estás leyendo.

Comprobación final

A partir del ejemplo 4 del recurso, responde en voz alta: ¿cuántas citas hay en total, y cuál es el servicio de la segunda? Si respondes correctamente (dos citas; la segunda es «urgencia de fontanería») leyendo el JSON con NVDA sin ayuda, la comprobación es positiva.

Qué acabas de conseguir

Puedes leer con soltura el formato exacto de JSON que recibirás en los Webhooks reales del resto de la Fase 3, incluidos objetos anidados y arrays de objetos.

Ejercicio práctico

Escribe a mano (en un editor de texto simple) un JSON con un array de dos elementos, donde cada elemento sea un objeto con al menos dos casillas, siguiendo el patrón del ejemplo 4. No hace falta que sea perfecto: el objetivo es practicar el patrón, no la sintaxis exacta.

Resumen de lo imprescindible

- La indentación (los espacios al principio de cada línea) indica el nivel de anidación de cada casilla.
- Un objeto anidado es un objeto completo dentro de otra casilla; un array de objetos es una lista donde cada elemento es, a su vez, un objeto con sus propias casillas.
- Este es el formato exacto que recibirás en los Webhooks reales de Tool Calling (módulo 3.4).

No necesitas aprender todavía

Todavía no necesitas escribir JSON complejo a mano para un caso real: el módulo 3.4 proporciona los archivos ya preparados.

Estoy preparado para continuar si puedo...

- Leer con NVDA un JSON con objetos anidados, siguiendo la indentación.
- Leer con NVDA un array de objetos y contar cuántos elementos contiene.
- Explicar la diferencia entre un objeto anidado y un array de objetos.

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. En un JSON, encuentras una línea con más espacios de indentación al principio que la línea anterior que contiene una llave de apertura. ¿Qué indica normalmente ese aumento de indentación?
 - a. Que esa línea pertenece a un nivel más profundo de anidación, dentro del objeto abierto en la línea anterior.
 - b. Que hay un error de formato en el JSON.

- c. Que esa línea no tiene ninguna relación con el resto del documento.
2. Un Webhook de n8n te envía un JSON con la clave "citas" seguida de corchetes que contienen tres bloques entre llaves. ¿Qué representa cada uno de esos tres bloques?
- a. Cada bloque es un elemento distinto del array: en este caso, tres citas diferentes, cada una con su propia estructura de casillas.
 - b. Los tres bloques son, en realidad, tres nombres alternativos para la misma cita.
 - c. Es un error, porque un array sólo puede contener un elemento.
3. ¿Por qué esta lección insiste en que sepas leer JSON con objetos anidados y arrays de objetos, en lugar de sólo JSON simple de una casilla?
- a. Porque ese es exactamente el formato real que recibirás en los Webhooks de Tool Calling del módulo 3.4, no un ejemplo simplificado sin relación con el resto del curso.
 - b. Porque es un requisito de certificación oficial de JSON.
 - c. Porque el JSON simple de una casilla ya no se usa en ningún sistema real.

Fuentes

- [Introducing JSON — JSON.org \(consultado el 2026-08-30\)](#)

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