



Manual de Webhooks & API

Versión 1.0 | Febrero 2026

Confidencial — Uso interno y para clientes autorizados

Índice de Contenidos

- 1.** Introducción al sistema de Webhooks
- 2.** Guía para el Administrador (Telsim)
 - 2.1** Arquitectura del sistema
 - 2.2** Estructura de la base de datos
 - 2.3** Trigger SQL — cómo funciona
 - 2.4** Mantenimiento y monitoreo
- 3.** Guía para el Cliente (Usuario)
 - 3.1** Qué es un webhook
 - 3.2** Cómo configurarlo en Telsim
 - 3.3** Estructura del payload JSON
 - 3.4** Cómo probarlo
- 4.** Ejemplos de integración
 - 4.1** Node.js / Express
 - 4.2** Python / Flask
 - 4.3** Make.com
 - 4.4** n8n — Paso a paso completo
 - 4.5** Zapier — Paso a paso
 - 4.6** Make.com — Paso a paso
 - 4.7** Respuesta esperada del servidor
- 5.** Solución de problemas

1. Introducción

Telsim cuenta con un sistema de webhooks en tiempo real que permite notificar automáticamente a sistemas externos cada vez que un SMS es recibido por el gateway físico (Sim Vault).

La notificación ocurre en milisegundos gracias a un trigger SQL nativo en Supabase que usa la extensión pg_net, eliminando la necesidad de polling o módulos adicionales en Make.



Velocidad: El webhook se dispara en menos de 10ms tras la recepción del SMS.

Flujo del sistema



2. Guía para el Administrador

2.1 Arquitectura del sistema

El sistema funciona completamente dentro de Supabase sin depender de Make para el envío del webhook. Make sigue siendo el encargado de recibir el SMS desde Sim Vault y guardarlos en la base de datos. Una vez guardado, Supabase se encarga del resto.

👉 **Importante:** Make NO consume créditos adicionales para el envío de webhooks. El trigger SQL es gratuito e ilimitado.

2.2 Estructura de la base de datos

Tabla: users

Los webhooks de los clientes se almacenan en la tabla users con estas columnas:

Columna	Tipo	Descripción
user_webhook_url	text	URL donde se enviará el POST con el SMS
webhook_is_active	boolean	true = activo, false = desactivado
webhook_provider	text	Tipo de proveedor (ej: custom_api)
api_key	uuid	Clave secreta del cliente para autenticar

2.3 Trigger SQL — Cómo funciona

Este es el código SQL activo en producción. Se ejecuta automáticamente en cada INSERT en sms_logs:

```
CREATE OR REPLACE FUNCTION notify_user_webhook()
RETURNS trigger AS $$

DECLARE
    webhook_url text;
    webhook_active boolean;
    phone text;

BEGIN
```

```

SELECT user_webhook_url, webhook_is_active
INTO webhook_url, webhook_active
FROM users WHERE id = NEW.user_id;

-- Obtener número de teléfono real del Sim Vault
SELECT phone_number INTO phone
FROM slots WHERE slot_id = NEW.slot_id;

IF webhook_active = true AND webhook_url IS NOT NULL THEN
    PERFORM net.http_post(
        url := webhook_url,
        headers := '{"Content-Type": "application/json"}'::jsonb,
        body := json_build_object(
            'event', 'sms.received',
            'from', NEW.sender,
            'content', NEW.content,
            'verification_code', NEW.verification_code,
            'service', NEW.service_name,
            'received_at', NEW.received_at,
            'phone_number', phone
        )::jsonb
    );
END IF;
RETURN NEW;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;

```

2.4 Activar webhook para un cliente

Para activar o desactivar el webhook de un cliente desde el SQL Editor:

```

-- ACTIVAR
UPDATE users
SET user_webhook_url = 'https://url-del-cliente.com/webhook',
    webhook_is_active = true
WHERE id = 'UUID-DEL-USUARIO';

```

```

-- DESACTIVAR
UPDATE users
SET webhook_is_active = false
WHERE id = 'UUID-DEL-USUARIO';

```

2.5 Monitoreo

Para verificar si los webhooks están siendo enviados correctamente, consulta la tabla de requests de pg_net:

```
SELECT id, status_code, created, url  
FROM net._http_response  
ORDER BY created DESC  
LIMIT 20;
```

 **Exitoso:** status_code = 200. Si ves 4xx o 5xx, la URL del cliente es incorrecta o su servidor rechazó el request.

3. Guía para el Cliente

3.1 ¿Qué es un webhook?

Un webhook es una URL de tu sistema que Telsim llamará automáticamente cada vez que llegue un SMS a tu número. En lugar de consultar Telsim cada cierto tiempo, Telsim te avisa a ti en tiempo real.



Analogía: Es como una alarma. En lugar de revisar si llegó un mensaje cada 5 minutos, Telsim te llama directamente cuando llega.

3.2 Cómo configurarlo en Telsim

Sigue estos pasos para activar tu webhook:

1. Inicia sesión en tu cuenta de Telsim
2. Ve a Ajustes > API & Webhooks
3. Ingresa la URL de tu endpoint (debe ser HTTPS)
4. Activa el toggle 'Webhook activo'
5. Guarda los cambios



Requisito: Tu URL debe ser pública y accesible desde internet. No funciona con localhost.

3.3 Estructura del payload JSON

Cuando llegue un SMS, tu endpoint recibirá un POST con este cuerpo JSON:

```
{  
  "event": "sms.received",  
  "from": "ENTEL",  
  "content": "Tu código de verificación es 847291",  
  "verification_code": "847291",  
  "service": "GoDaddy",  
  "phone_number": "+56 9 3444 9937",  
}
```

```
        "received_at": "2026-02-23T19:18:22.913814+00:00"
    }
```

Campo	Tipo	Descripción
event	string	Siempre 'sms.received'
from	string	Remitente del SMS (número o nombre)
content	string	Texto completo del mensaje
verification_code	string null	Código numérico detectado, si existe
service	string null	Servicio detectado (Google, WhatsApp, etc.)
phone_number	string	Número de teléfono que recibió el SMS
received_at	ISO 8601	Fecha y hora exacta de recepción

3.4 Cómo probar tu webhook

Para probar sin tener tu servidor listo, puedes usar webhook.site:

6. Ve a <https://webhook.site> y copia tu URL única
7. Pégala en Telsim > Ajustes > Webhook URL
8. Espera un SMS entrante o pide al soporte que envíe uno de prueba
9. Verifica que aparece el JSON en webhook.site

4. Ejemplos de Integración

4.1 Node.js / Express

```
const express = require('express');
const app = express();
app.use(express.json());

app.post('/webhook/telsim', (req, res) => {
  const { event, from, content, verification_code } = req.body;
  console.log('SMS recibido de:', from);
  console.log('Código:', verification_code);
  // Tu lógica aquí...
  res.status(200).json({ ok: true });
});
```

4.2 Python / Flask

```
from flask import Flask, request, jsonify
app = Flask(__name__)

@app.route('/webhook/telsim', methods=['POST'])
def receive_sms():
    data = request.json
    code = data.get('verification_code')
    print(f'Código recibido: {code}')
    return jsonify({'ok': True}), 200
```

4.3 Make.com / Zapier

Puedes usar la URL de webhook de Make o Zapier directamente como tu Webhook URL en Telsim. El payload JSON llegará automáticamente y podrás mapearlo a cualquier acción.

4.4 n8n — Paso a paso

n8n es una herramienta de automatización self-hosted o en la nube compatible con el webhook de Telsim sin configuración adicional.

Configuración del nodo Webhook en n8n:

10. En tu instancia n8n, crea un nuevo Workflow
11. Agrega un nodo Webhook (trigger) como primer nodo
12. En 'HTTP Method' selecciona POST
13. En 'Path' escribe: telsim-sms (o el nombre que prefieras)
14. Cambia el modo a 'Production URL' y copia la URL generada
15. Pega esa URL en Telsim > Ajustes > Webhook URL y activa el toggle
16. Activa el Workflow en n8n (botón 'Active' arriba a la derecha)

Ejemplo de flujo completo en n8n:

Este flujo recibe el SMS, extrae el código y lo envía por correo:

```
[Webhook Telsim] → [IF: verification_code existe?]
                           ↓ Sí
                           [Send Email / Telegram / Slack]
                           ↓ No
                           [Set: registrar SMS informativo]
```

Acceder a los campos en n8n:

```
{{ $json.verification_code }}    // El código numérico
{{ $json.phone_number }}        // Número que recibió el SMS
{{ $json.from }}                // Remitente (ej: Google, ENTEL)
{{ $json.content }}             // Texto completo del SMS
{{ $json.service }}              // Servicio detectado
{{ $json.received_at }}          // Fecha y hora ISO 8601
```

4.5 Zapier — Paso a paso

Zapier puede recibir el webhook de Telsim usando el trigger 'Webhooks by Zapier'.

17. Crea un nuevo Zap en Zapier
18. Como trigger elige 'Webhooks by Zapier' > 'Catch Hook'
19. Copia la URL de webhook generada por Zapier
20. Pégala en Telsim > Ajustes > Webhook URL
21. Espera un SMS de prueba para que Zapier detecte la estructura

22. Agrega las acciones que necesites (Gmail, Sheets, Slack, etc.)

4.6 Make.com — Paso a paso

En Make puedes usar el módulo Webhooks para recibir los SMS de Telsim.

23. Crea un nuevo Scenario en Make

24. Como módulo inicial selecciona Webhooks > Custom Webhook

25. Crea el webhook y copia la URL generada

26. Pégala en Telsim > Ajustes > Webhook URL

27. Envía un SMS de prueba — Make detectará automáticamente la estructura del JSON

28. Conecta los módulos siguientes: Google Sheets, Gmail, Telegram, etc.

4.7 Respuesta esperada de tu servidor

Tu servidor DEBE responder con HTTP 200 para que Telsim considere el envío exitoso. Telsim no reintenta si falla. Cualquier código diferente a 2xx se considera error.

```
// ✓ Respuesta correcta
HTTP/1.1 200 OK
{ "ok": true }
```

```
// ✗ Respuesta incorrecta – Telsim marcará error
HTTP/1.1 500 Internal Server Error
```

5. Solución de Problemas

Problema	Solución
No llega el webhook	Verifica que webhook_is_active = true en Supabase y que la URL es correcta
Error 404 o 502	Tu servidor no está corriendo o la ruta no existe. Prueba primero con webhook.site
El código llega null	El SMS no tenía un código numérico detectable. Es normal para SMS informativos
Llegan webhooks duplicados	Revisa si tienes múltiples triggers activos en Supabase
La URL no acepta el POST	Asegúrate de que tu servidor responde con status 200. Telsim no reintenta en caso de error
Quiero desactivar temporalmente	Cambia webhook_is_active = false en tu perfil o desde el panel

¿Necesitas ayuda? Contacta soporte en [soporte@telsim.io](mailto:support@telsim.io)