

Integraciones de SIMPLAT

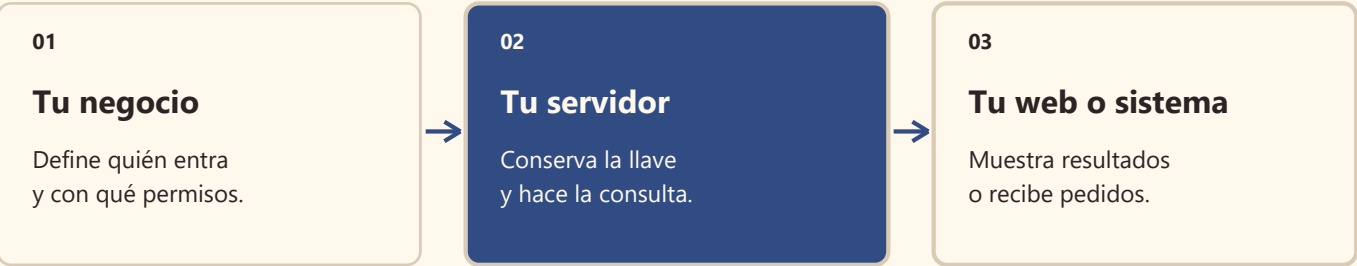
Manual visual para el propietario, sus clientes y quien conecta un sistema.

Versión 1.0 · 13 de octubre de 2026 · API v1

Puedes consultar esta guía sin una suscripción a ChatGPT. La integración depende de tu plan de SIMPLAT, del servidor de tu sistema y de una llave válida.

Este manual describe el código disponible de SIMPLAT. Los ejemplos son ficticios; no son ventas, clientes ni pedidos reales. La aceptación completa con cuentas y sincronización en un entorno aislado sigue pendiente.

De entender la conexión a dejarla documentada.



RUTA DEL NEGOCIO / capítulos 01-06
RUTA TÉCNICA / capítulos 07-16

01. Elige el camino que necesitas

Si solo quieres vender mediante una página de SIMPLAT, empieza en **Mi tienda**. Esa opción ya utiliza el flujo de pedidos del producto y no requiere que programes una conexión con la API.

Si tienes una página web, una tienda o un sistema propio, su **servidor** puede consultar la API. La llave identifica un negocio y sus permisos. No es una contraseña para entrar a la aplicación.

Necesidad	Camino	Quién lo prepara
Configurar la tienda de SIMPLAT	Mi tienda, dentro de la app	El negocio
Mostrar un catálogo en otra web	API de productos + servidor de esa web	El desarrollador de la web
Recibir pedidos de otro sistema	API de pedidos + confirmación dentro de SIMPLAT	El desarrollador y el negocio
Llevar ventas a un reporte externo	API de ventas, de solo lectura	Quien prepara el reporte

No hay un botón universal que conecte cualquier aplicación. Esta versión no incluye conectores oficiales de Shopify, WooCommerce o Zapier, ni webhooks publicados. Una herramienta que admita HTTP puede adaptarse a la API; hay que configurar y probar esa adaptación.



02. Qué se necesita antes de empezar

1. Un negocio con la función **API para integraciones**, incluida en el plan Empresarial.
2. Acceso de **propietario** de ese negocio para administrar las llaves. Ser co-dueño o empleado no concede ese permiso.
3. Un servidor o servicio de tu sistema externo capaz de hacer peticiones HTTPS y guardar un secreto de forma privada.
4. Internet. La API consulta la información del servidor; los cambios de un dispositivo que aún no se han subido no aparecen por esa vía.
5. Para ensayar escrituras, un negocio y un entorno aislados identificados expresamente para pruebas.

Dirección base: <https://www.simplatgestion.online/api/v1>

Documentación de rutas: abre esa dirección sin llave. Devuelve información técnica, no datos de un negocio.

Pantalla de llaves: entra a tu cuenta, elige el negocio correcto y abre **Integraciones**. También puedes abrir /integraciones después de iniciar sesión.

Si la pantalla dice que solo el propietario administra las llaves, pide al propietario que prepare la conexión. Si indica que la API no está habilitada, reintenta la consulta y contacta a studio@simplatgestion.online; no cambies permisos ni ejecutes SQL para sortear el aviso.

01

Propietario

Elige el negocio.
Crea y revoca llaves.
Decide los permisos.

02

Desarrollador

Configura el servidor.
Protege la llave.
Maneja los errores.

03

Negocio

Revisa los pedidos.
Confirma en la app.
Documenta la entrega.

Una cuenta de co-dueño no sustituye el permiso del propietario para administrar llaves.

03. Crear y conservar una llave

En **Integraciones > Nueva llave**:

1. En **¿Para qué es?**, escribe un nombre reconocible: Catálogo de mi web o Pedidos del sistema externo.
2. En **Vence en**, elige 90 días, un año o sin vencimiento. Anota la fecha elegida y quién renovará la conexión.
3. Marca únicamente los permisos necesarios. Para un catálogo, empieza con **Leer productos**.
4. Pulsa **Crear llave**. Copia el valor completo cuando aparezca.
5. Guárdalo en la configuración privada del servidor externo. La lista posterior solo muestra un prefijo; el valor completo se muestra una vez.

Una llave pertenece a un negocio. Cambiar de negocio en tu sesión no cambia el negocio de una llave ya creada. Para otro negocio, su propietario debe crear otra llave.

Hay un máximo configurado de **10 llaves activas por negocio**. No crees una por cada visitante o cada pedido. Usa nombres por sistema y conserva un inventario de conexiones.

Si pierdes el valor completo, crea una llave de reemplazo y revoca la anterior. No se puede recuperar desde la lista.

Nueva llave

Representación ilustrativa del formulario; no contiene un secreto.

1 ¿Para qué es?

Catálogo de mi web

2 Vence en

90 días / un año

3 Permisos: Leer productos

Marca solo lo necesario para esa conexión.

4 **Crear llave** > copiar una vez > guardar en el servidor

04. Permisos: concede lo que hace falta

Permiso	Permite	Uso habitual
productos:leer	Catálogo, precio y existencias	Catálogo de una web
clientes:leer	Contactos y datos de identificación del cliente	Sincronización de contactos autorizada
ventas:leer	Ventas y devoluciones; líneas opcionales	Reportes externos
pedidos:leer	Pedidos, líneas y estado	Seguimiento
pedidos:escribir	Crear un pedido pendiente	Recibir pedidos externos

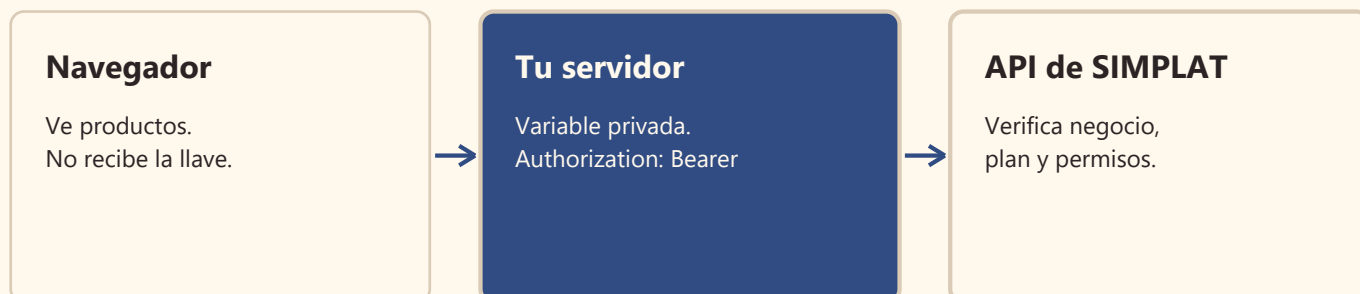
Para mostrar productos no necesitas leer clientes. Los clientes incluyen datos personales y documentos de identificación: concede ese permiso únicamente para un uso definido y limita quién puede ver los resultados.

La API de productos no entrega el costo ni el margen. Un producto digital lleva stock: null: significa que no aplica una existencia física, no que esté agotado.

La consulta de negocio comprueba la llave sin pedir un permiso adicional. Todas las demás consultas exigen el permiso de su recurso.

El plan se comprueba en cada llamada. Una llave creada antes de un cambio de plan puede dejar de funcionar si el negocio ya no tiene la función API.

05. Dónde colocar la llave



RESPUESTA AL CLIENTE

Tu servidor devuelve solo los campos necesarios. Mantiene el secreto fuera del HTML, de las URLs y del código público.

El navegador del cliente pide el catálogo a **tu servidor**. Tu servidor añade la llave y consulta SIMPLAT. Después devuelve al navegador solo lo que la página necesita mostrar.

La cabecera de la llamada privada es:

```
Authorization: Bearer VALOR_COMPLETO_DE_TU_LLAVE
```

No coloques la llave en HTML, JavaScript que se descarga al navegador, una app pública, una URL, un repositorio, una captura o un campo que cualquier visitante pueda leer. Variables como `NEXT_PUBLIC_*` son públicas y no sirven para este secreto.

La llave de integración no es la clave `service_role` de Supabase ni un certificado de Hacienda. Un cliente externo no necesita esos secretos.

En un constructor de páginas usa una función de servidor o una conexión HTTP privada. Un widget que solo ejecuta JavaScript en el navegador no basta. El servidor evita además depender de permisos CORS del navegador.

Si entregas la configuración a otra persona, usa un medio privado apropiado y comparte solo la llave de ese sistema. En este manual y en el kit no hay llaves reales.

06. Tu primera comprobación, sin crear datos

Primero consulta **GET /negocio**. Verifica el nombre e identificador del negocio y la lista alcances. Si corresponde a otro negocio, detente y reemplaza la llave.

En PowerShell de Windows, la entrada oculta evita escribir el secreto como parte del comando guardado en el historial:

```
$simTokenSeguro = Read-Host "Llave SIMPLAT" -AsSecureString
$env:SIMPLAT_TOKEN = [System.Net.NetworkCredential]::new(
    "", $simTokenSeguro
).Password
$simHeaders = @{ Authorization = "Bearer $env:SIMPLAT_TOKEN" }
Invoke-RestMethod -Method Get -Headers $simHeaders -Uri (
    "https://www.simplatgestion.online/api/v1/negocio"
)
# Al terminar las consultas de esta sesión:
Remove-Item Env:\SIMPLAT_TOKEN
Remove-Variable simHeaders, simTokenSeguro
```

Respuesta ilustrativa; los valores serán los de tu negocio:

```
{
  "ok": true,
  "alcances": ["productos:leer"],
  "restantes": 599,
  "datos": {
    "id": "11111111-1111-4111-8111-111111111111",
    "nombre": "Negocio de ejemplo", "moneda": "USD",
    "iva": 13, "pais": "SV", "plan": "Empresarial"
  }
}
```

Una lectura autenticada actualiza el uso de la llave; no crea ventas ni pedidos. No publiques una respuesta con contactos o cifras privadas para pedir soporte.

07. Un catálogo funcionando en tu computadora

El kit incluye un servidor de ejemplo sin dependencias externas. Necesita **Node.js 24 o posterior**. Es una base didáctica; no sustituye el alojamiento, la protección y la operación de tu web comercial.

1. Descomprime el kit en una carpeta propia. Abre una terminal en esa carpeta.
2. Ejecuta el modo de ejemplo, que no usa una llave ni llama a SIMPLAT:

```
node servidor-catalogo.mjs --demo
```

3. Abre `http://127.0.0.1:3210`. Verás productos ficticios y la indicación de modo de ejemplo.
4. Para una consulta real de solo lectura, copia `configuracion.env.example` como `.env`. Completa `SIMPLAT_TOKEN` en ese archivo privado y ejecuta:

```
node --env-file=.env servidor-catalogo.mjs
```

5. El servidor comprueba el negocio antes de pedir productos. Revisa los datos y detén el proceso con **Ctrl+C** al terminar.

El servidor del kit solo escucha en tu computadora, no publica tu web en internet, no envía pedidos y no guarda contactos. Filtra los productos no publicados en tienda y devuelve una lista reducida de campos.

Para llevarlo a tu sitio, entrega el ejemplo a quien administra su servidor: debe adaptar el alojamiento, el dominio y sus protecciones. La llave sigue en la configuración privada. El kit no trae un puente público que permita consultar ventas o clientes a cualquier visitante.

SIN LLAVE / SIN LLAMADAS A SIMPLAT

Terminal + navegador

```
node servidor-catalogo.mjs --demo
Abre http://127.0.0.1:3210
```

Cuando domines el ejemplo, configura una lectura real desde tu servidor privado.

08. Consultar productos y recorrer las páginas

GET /productos necesita productos: leer.

Parámetro	Significado
limite	Tamaño de página: 100 por defecto, máximo 500
desde	Desplazamiento: 0, 100, 200...; no es una fecha
categoria	Nombre de categoría; coincidencia exacta
inactivos=1	Incluye productos archivados

```
GET /api/v1/productos?limite=100&desde=0
Authorization: Bearer VALOR_COMPLETO_DE_TU_LLAVE
```

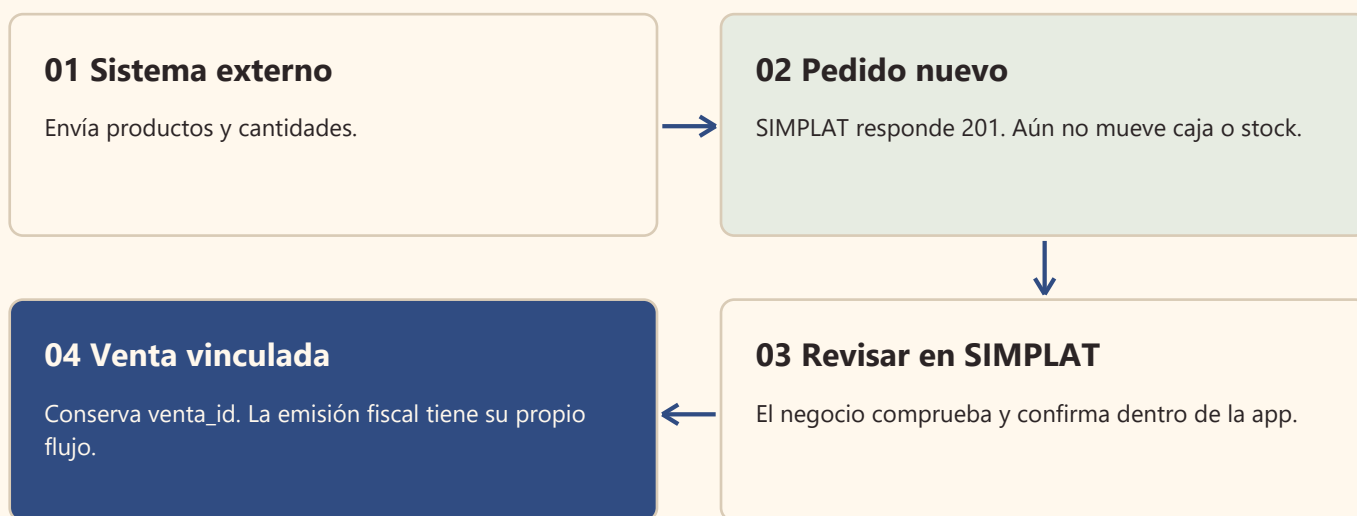
La respuesta contiene ok, datos, total, limite, desde y restantes. datos es una lista. Cada producto incluye id, nombre, categoria, precio, unidad, codigo, descripcion, stock, digital, unidades_por_paquete, activo, en_tienda y created_at.

Avanza sumando el limite recibido a desde. Termina cuando datos esté vacío, tenga menos elementos que limite, o hayas alcanzado un total numérico. Si total es null, no lo interpretes como cero.

Para una vitrina pública filtra activo y en_tienda. La API autenticada puede devolver productos activos que el negocio no publicó en la tienda; la web externa decide qué muestra.

Las páginas por desplazamiento no forman una instantánea bloqueada. Si el catálogo cambia durante la lectura, puede haber repetidos o movimientos entre páginas. Deduplica por id, conserva una fecha de actualización y no prometas stock reservado.

09. Recibir pedidos: el recorrido completo



Pedido recibido, venta confirmada y DTE aceptado son resultados diferentes.

POST /pedidos necesita `pedidos:escribir`. Crea un pedido con estado nuevo. No reserva stock, cobra, registra una venta ni emite DTE.

Dentro de **Pedidos** en SIMPLAT, el negocio revisa el pedido y lo confirma mediante el flujo de la aplicación. En consultas posteriores, un `venta_id` identifica que ese pedido se convirtió en una venta. El estado de preparación o entrega no acredita por sí solo un documento aceptado por Hacienda.

Los precios y nombres los obtiene el servidor de SIMPLAT. Envía identificadores y cantidades, no un precio calculado por el navegador. Para empezar, usa productos base; la API no publica rutas de catálogo de variantes o modificadores.

Si tu tienda cobra mediante otro proveedor, la integración debe conciliar ese cobro con el pedido y el procedimiento de SIMPLAT. Esta API no incorpora un pago electrónico ni confirma automáticamente que se recibió dinero.

Prueba primero las lecturas. Ensayo POST solo en el entorno aislado acordado: un pedido en producción es una escritura real, aunque todavía no mueva caja o inventario.

10. Forma del pedido y respuesta

Petición ficticia para preparar; no la ejecutes contra un negocio real como ensayo:

```
{
  "cliente_nombre": "Persona de ejemplo",
  "entrega": "retiro",
  "nota": "Referencia externa DEMO-001",
  "items": [{
    "producto_id": "22222222-2222-4222-8222-222222222222",
    "cantidad": 2
  }]
}
```

Campo	Regla actual
cliente_nombre	Obligatorio; hasta 80 caracteres
cliente_telefono	Opcional; hasta 30 caracteres
entrega	retiro o domicilio; no admite mesa por esta API
cliente_direccion	Obligatoria para domicilio; hasta 240 caracteres
cliente_id	Opcional; debe pertenecer al mismo negocio
nota	Hasta 400 caracteres; puedes guardar una referencia externa
items	De 1 a 40 líneas; cantidad positiva, máximo 500 por línea

La implementación conserva como máximo 40 líneas y limita cantidades a 500. Prepara esos límites en el sistema externo; no dependas de recortes silenciosos. Los productos no válidos se descartan; si no queda ninguno, se rechaza el pedido.

Una creación aceptada devuelve **HTTP 201**, `datos.id`, `numero`, `estado`, `total`, `items` y `descartadas`, además de `aviso` y `restantes`. Compara cantidades, líneas y total de la respuesta antes de informar al cliente. `descartadas` se refiere a las líneas de entrada retenidas, no certifica que una petición de más de 40 líneas se haya conservado completa.

El número del pedido no es un número de factura o DTE. La API no crea ni empareja una ficha de cliente por teléfono.

11. Cuando una respuesta se pierde

Se perdió la respuesta al POST

Conserva contenido, referencia y hora. Marca el envío como pendiente.



Encontraste el pedido

Vincula su id. No vuelvas a crear el mismo envío.



No sabes el resultado

Revisión manual antes de reenviar. La ausencia no está demostrada.

Si el sistema externo agota la espera después de enviar un pedido, no sabes todavía si SIMPLAT lo guardó. Marca ese envío como **pendiente de comprobar** y conserva la referencia, la hora y el contenido enviado.

Consulta **GET /pedidos** con `pedidos:leer` y revisa los pedidos recientes y su nota. La referencia externa ayuda a investigar, pero no es una restricción única ni evita duplicados en el servidor.

Esta versión no documenta una clave de idempotencia para POST. Mandar una cabecera `Idempotency-Key` no añade esa garantía. No configures un reintento automático de pedidos como si fueran consultas GET.

Si encuentras el pedido, vincula su `id` al registro externo. Si no puedes determinar el resultado, solicita revisión antes de volver a enviarlo. Un error de conexión o un HTTP 500 tampoco prueba por sí solo que no hubo una escritura.

Puedes reintentar lecturas con espera creciente y un máximo de intentos. Los errores de llave o permisos requieren corregir la configuración; repetirlos no ayuda.

12. Ventas, clientes y seguimiento

Consulta	Filtros y detalles
GET /ventas	desde_fecha, hasta_fecha en YYYY-MM-DD; límites inclusivos
GET /ventas	tipo=venta excluye devoluciones; por defecto las incluye
GET /ventas	items=1 con páginas de hasta 100 ventas para incluir líneas
GET /clientes	buscar busca por nombre; inactivos=1 incluye archivados
GET /pedidos	estado filtra por el estado exacto; devuelve líneas

Todos admiten `limite` y `desde`. Para ventas con líneas usa `limite=100` o menos; una página mayor puede devolver un aviso y no incluir sus líneas.

Las devoluciones llevan `tipo`: `devolucion`, `totales negativos` y `ref_venta_id`. No las conviertas en ventas positivas. `importada` distingue el historial traído de otro sistema: evita sumarlo dos veces al consolidar datos.

El catálogo de clientes incluye teléfono, correo, dirección, NIT/NRC/DUI y otros datos de identificación; no devuelve deuda ni límite de fiado. Usa ese permiso solo para el propósito autorizado. No publiques su respuesta como si fuera un catálogo.

Los pedidos incluyen `cliente`, `estado`, `venta_id`, `fechas`, `motivo` e `items`. La API no incorpora rutas `PATCH/DELETE` para cambiar su estado desde fuera. La operación se completa en SIMPLAT.

Para seguimiento, combina consultas periódicas moderadas y almacenamiento por `id` en tu servidor. La API no publica webhooks para avisar cada cambio y no garantiza que una lectura contenga cambios aún pendientes de sincronizar desde un dispositivo.

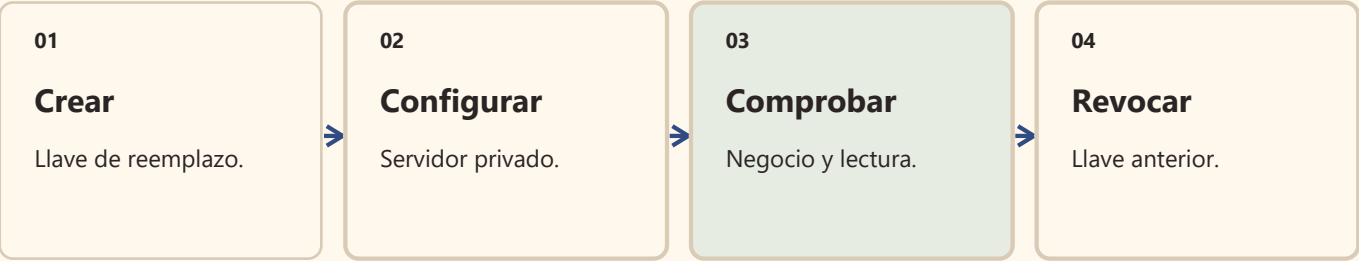
13. Errores que puedes resolver

HTTP / error	Qué significa	Próximo paso
401 / no_token	Falta la llave en la cabecera	Revisa Authorization en el servidor
401 / token_invalido	Formato incorrecto, llave inexistente, vencida o revocada	Sustituye la llave; no pruebes valores al azar
403 / sin_alcance	Falta el permiso del recurso	El propietario crea una llave adecuada
403 / plan	La función API no está disponible para el plan	Comprueba el plan del negocio
400 / payload	El cuerpo no es JSON válido	Corrige Content-Type y el JSON
400 / faltan_datos, falta_direccion, sin_items	Falta nombre, dirección de domicilio o líneas válidas	Revisa el pedido antes de enviar
400 / cliente_no_existe	El cliente indicado no pertenece a ese negocio	Usa un id del negocio correcto
429 / limite	Se alcanzó el límite configurado de llamadas	Reduce consultas y espera
429 / muchos_pedidos	Tope configurado de pedidos por negocio	Detén nuevos envíos automáticos
500 / error	La operación no pudo completarse correctamente	En POST, comprueba primero su resultado
503 / unavailable	Configuración o servicio de API no disponible	Conserva la configuración y contacta soporte

El límite configurado es **600 llamadas por hora por llave** y **60 pedidos nuevos por hora por negocio**, incluidos otros canales que llegan al mismo buzón. Si existe una cabecera Retry-After, respétala; no todas las respuestas 429 la incluyen. El kit no intenta gastar la cuota para probar el límite.

La forma habitual del error es ok: false, error y mensaje; algunos rechazos generales de seguridad pueden tener otra forma. Comprueba primero HTTP y maneja respuestas que no sean JSON.

14. Renovar, revocar y dejarlo documentado



Ante una exposición, prioriza revocar y recuperar la conexión con una llave nueva.

Renovación planificada: crea una llave de reemplazo con los permisos necesarios, actualiza la configuración privada del servidor, comprueba GET /negocio y una lectura del recurso, y después revoca la anterior en **Tus llaves**.

Si la llave se expuso, revócala cuanto antes y reemplázala. Las llamadas que se autenticquen después ya no podrán utilizarla; la revocación no borra datos que un sistema externo haya descargado ni deshace operaciones anteriores.

Cuando retires el acceso de una persona, revisa también las llaves que tenían sus sistemas. Quitar a una persona de la app no elimina automáticamente los secretos de servidores externos.

No hay una opción para editar los permisos de una llave existente. Sustitúyela para cambiar su alcance o su vencimiento.

Ficha que debe conservar el propietario, sin el secreto:

Dato	Completar
Sistema y responsable	Nombre de la web, proveedor y contacto
Negocio	Nombre e identificador comprobados
Llave	Nombre y prefijo visible; nunca su valor completo
Permisos y propósito	Recursos estrictamente necesarios
Vencimiento	Fecha y responsable de renovar
Última comprobación	Fecha, lectura comprobada y resultado
Recuperación	Cómo detener el sistema y revocar la llave

15. Entrega al cliente y comprobación final

El propietario puede preparar permisos, crear la llave y probar una lectura con esta guía. Si la web externa no tiene una función de servidor preparada, todavía requiere desarrollo o configuración de su proveedor.

Antes de dar una conexión por lista:

1. GET /negocio identifica el negocio esperado.
2. El sistema recibe los permisos previstos y maneja su ausencia.
3. Se recorren todas las páginas y se conservan identificadores, nulos y devoluciones.
4. La llave no aparece en los archivos, solicitudes o respuestas que recibe un visitante.
5. El cliente comprende que un pedido recibido se confirma dentro de SIMPLAT.
6. POST, reintentos, cambios de estado y sincronización se prueban en un entorno aislado antes de utilizarlos con operaciones reales.
7. Quedan anotados vencimiento, responsable y procedimiento para detener la conexión.

Ayuda para soporte: envía el nombre del sistema, endpoint, HTTP, clave de error, hora aproximada y referencia externa no sensible. Oculta Authorization, tokens, contactos, identificadores fiscales y datos de clientes en cualquier captura.

Soporte: studio@simplatgestion.online.

La guía sirve sin el agente ni una membresía a ChatGPT. La operación depende de tu cuenta de SIMPLAT, del servicio externo y de quienes administren su servidor. Ninguna guía sustituye comprobar una conexión real.

16. Referencia técnica y archivos del kit

Archivo	Para qué sirve
servidor-catalogo.mjs	Catálogo local; modo ficticio o lecturas reales con llave privada
cliente-simplat.mjs	Cliente de solo lectura con paginación y control de respuestas
comprobar.mjs	Comprueba negocio y recurso desde la terminal
configuracion.env.example	Nombres de variables; sin una llave real
coleccion-postman.json	Peticiones de lectura y pedido desactivado por defecto
pedido.example.json	Forma de un pedido ficticio para adaptar en QA
README.md	Inicio rápido y límites del ejemplo

Los ejemplos usan módulos estándar de Node.js y peticiones HTTP. No instales paquetes desconocidos para que una llave funcione. En la colección de Postman, el token está vacío y el envío de pedidos bloquea el dominio principal y exige habilitar expresamente un entorno aislado de pruebas; no guarda ni entrega un secreto real.

Fuentes de esta versión: contratos y pantallas de SIMPLAT revisados el 13 de octubre de 2026: `packages/core/src/api.ts`, `/api/v1`, `/api/tokens`, rutas negocio/productos/clientes/ventas/pedidos, pantalla Integraciones y `lib/tienda/pedido.ts`. Código de referencia de esta revisión: `a7aa418`; la guía no certifica una conexión autenticada en producción.

Documentación de SIMPLAT: <https://www.simplatgestion.online/api/v1>

Node.js 24, variables de entorno: <https://nodejs.org/docs/latest-v24.x/api/cli.html#--env-filefile>

Postman, importar una colección:

<https://learning.postman.com/docs/getting-started/importing-and-exporting/importing-data/>

Al cambiar la API, compara primero la documentación de rutas y conserva esta versión para saber qué integración configuraste. Una nueva ruta o garantía debe documentarse y verificarse antes de incorporarla a un sistema externo.