EcomLabs

GUÍA PASO A PASO

GA4 y Shopify: cómo cuadrar tus ventas

La checklist para reconciliar GA4 con Shopify, cómo revisar el evento purchase con DebugView, el Pixel Helper de Shopify y Tag Assistant, y un script de Node que cruza tus pedidos con tus compras por transaction_id para listar las que faltan y las duplicadas.

PARA	Tiendas Shopify que miden sus ventas con GA4
PROBADO CON	Node 22.22.3 · node:test · Windows 11 · CSV de ejemplo ilustrativos
ACTUALIZADA	11 de noviembre de 2026
DOCUMENTACIÓN	help.shopify.com/en/manual/reports-and-analytics/discrepancies
ARTÍCULO	ecomlabs.dev/g/tracking
AUTOR	Luis Morales · @luismoralesdev

Índice

Antes de empezar	Qué vas a comparar
Paso 01	Checklist para reconciliar
Paso 02	Revisa el evento purchase en DebugView
Paso 03	Pixel Helper y Tag Assistant
Paso 04	Exporta y cruza pedido por pedido
Paso 05	Lee el resultado

ANTES DE EMPEZAR

Qué vas a comparar

Pedidos de Shopify contra compras de GA4 (el evento `purchase`), pedido por pedido, en días cerrados. No compares sesiones ni tasas de conversión: cada herramienta las define a su manera, y Shopify cambió su medición de sesiones entre el 21 y el 23 de septiembre de 2026. Necesitas el admin de Shopify, permiso de Editor en la propiedad de GA4 y Node 22 (lo probamos con 22.22.3). Nada de esta guía manda datos a Google ni a Shopify: el script trabaja con dos CSV en tu computadora.

PASO 01

Checklist para reconciliar

- [ ] La zona horaria de GA4 (Admin > *Property details*) es la de tu tienda (Shopify: *Settings > General*). Cambiarla en GA4 solo afecta los datos de ahí en adelante.
- [ ] La moneda de GA4 es la de tu tienda; si no, GA4 convierte con el tipo de cambio del día anterior.
- [ ] La tienda no está en modo privado: Shopify avisa que GA4 no rastrea eventos mientras lo esté.
- [ ] Un solo tag manda `purchase` a cada propiedad. Revisa *Settings > Customer events* (app pixels y custom pixels) y el código del tema. Nunca gtag.js y GTM a la vez.
- [ ] El `transaction_id` es el ID del pedido: único, dinámico, nunca vacío ni fijo, sin datos del cliente.
- [ ] Sabes qué base trae `value`. Google recomienda la suma de `price × quantity` sin envío ni impuestos; el tutorial de GTM de Shopify usa `checkout.totalPrice.amount`, que incluye impuestos.
- [ ] No queda medición de compras en additional scripts ni en script tags de la página de estado del pedido: Shopify los apagó en 2025 y 2026.
- [ ] Si usas banner de cookies, sabes en qué regiones está activo y si tu pixel espera el consentimiento.
- [ ] Comparas días de hace al menos 48 horas: GA4 tarda de 24 a 48 horas en procesar y los números cambian mientras tanto.
- [ ] Repites el cruce de la guía después de cada cambio de tags, apps o tema.

PASO 02

Revisa el evento purchase en DebugView

1. Activa el modo de depuración con Tag Assistant (`tagassistant.google.com`) o la vista previa de GTM. Con gtag.js en el tema también sirve `gtag('config', 'G-XXXXXXXXXX', { 'debug_mode': true });`; para apagarlo, quita el parámetro, porque `false` no lo apaga.
2. En GA4, abre Admin > *Data display* > *DebugView* y acepta las cookies del banner: DebugView no muestra los eventos de quien no dio consentimiento a las cookies de Analytics.
3. Haz un pedido de prueba con la pasarela de prueba de Shopify o el modo de prueba de Shopify Payments. Los pedidos de prueba no aparecen en tus reportes.

4. Haz clic en el evento `purchase` para ver sus parámetros: `transaction_id` con el número o el ID de tu pedido, `value`, `currency` e `items`.
5. Recarga la página de gracias. No debe aparecer un segundo `purchase` con otro ID.
6. Si DebugView no muestra tu dispositivo, abre *Reports > Realtime*: la tarjeta de eventos por nombre cubre los últimos 30 minutos.

> **CUIDADO**
>
> GA4 solo recoge el primer `purchase` con un mismo `transaction_id` del mismo usuario e ignora los siguientes, aunque cambien otros valores. Para repetir una prueba, haz otro pedido de prueba: cada uno trae su propio ID.

PASO 03

Pixel Helper y Tag Assistant

Shopify Pixel Helper, para app pixels (como el del canal Google & YouTube) y custom pixels:

1. En el admin, ve a *Settings > Customer events*, abre el menú de opciones del pixel, elige la opción de prueba y haz clic en *Test*.
2. Si aparece «Pixel is awaiting consent», acepta en el banner o haz clic en *Give consent to continue test*.
3. Haz un pedido de prueba y busca `checkout_completed`: un punto verde es que el evento se recibió y su callback corrió; uno rojo, que el callback falló. Expande el diálogo para ver los datos.

El Pixel Helper revisa un pixel a la vez, borra el registro al cambiar de página y no funciona en tiendas headless ni en modo privado.

Tag Assistant, para un Google tag o GTM pegado en el tema: en `tagassistant.google.com`, haz clic en *Add domain*, escribe la URL de tu tienda con `https://` y haz clic en *Connect*. En el encabezado *Google tags found*, tu ID de tag debe aparecer con un ícono verde o azul; compara los tags que encuentra con los de *Customer events*. Si el parámetro `_dbg` rompe la página, desmarca *Include debug signal in the URL*.

> **CUIDADO**
>
> Según el tutorial de GTM de Shopify, la función *Troubleshoot tag* de Tag Assistant no es compatible con custom pixels y no detecta los Google tags de un custom pixel ni los que carga GTM ahí. Para esos, usa el Pixel Helper y DebugView.

PASO 04

Exporta y cruza pedido por pedido

Shopify: en *Orders*, filtra las fechas, haz clic en *Export* y elige CSV. **GA4:** en *Explore*, crea una exploración en blanco, importa la dimensión *Transaction ID* y la métrica *Transactions*, arrástralas a filas y valores, en ese orden, y exporta en CSV con *Export data*. Si aparece el aviso de umbrales, amplía el rango de fechas. Guarda `reconcile.mjs` junto a los dos archivos; no necesita dependencias:

```
// reconcile.mjs: compares a Shopify orders export with a GA4 export of transaction IDs.
// Usage: node reconcile.mjs orders_export.csv ga4-transactions.csv
import { readFileSync } from "node:fs";
import { pathToFileURL } from "node:url";
```

```
// CSV with quoted fields: commas, "" and line breaks inside quotes.
export function parseCsv(text) {
  const rows = [];
  let row = [], field = "", quoted = false;
  for (let i = 0; i < text.length; i++) {
    const c = text[i];
    if (quoted) {
      if (c === '"' && text[i + 1] === '"') { field += '"'; i++; }
      else if (c === '"') quoted = false;
      else field += c;
    } else if (c === '"') quoted = true;
    else if (c === ",") { row.push(field); field = ""; }
    else if (c === "\n") { row.push(field.replace(/\r$/, "")); rows.push(row); row = []; field = ""; }
    else field += c;
  }
  if (field || row.length) { row.push(field.replace(/\r$/, "")); rows.push(row); }
  return rows;
}

const clean = (id) => String(id ?? "").trim().replace(/^#/, "");

export function reconcile(ordersCsv, ga4Csv) {
  // Shopify: one row per line item, so keep the first row of each order (Name).
  const [head, ...rows] = parseCsv(ordersCsv.replace(/^\uFEFF/, ""));
  const col = Object.fromEntries(["Name", "Id", "Source", "Created at", "Payment Method"].map((n) => [n,
head.indexOf(n)]));
  const absent = Object.keys(col).filter((n) => col[n] < 0);
  if (absent.length) throw new Error(`orders export without column(s): ${absent.join(", ")}`);
  const orders = new Map();
  const byId = new Map();
  for (const r of rows) {
    const name = r[col.Name];
    if (!name || orders.has(name)) continue;
    const order = { name, source: r[col.Source], created: r[col["Created at"]], payment: r[col["Payment
Method"]], hits: [] };
    orders.set(name, order);
    // GA4 may hold the order number (1001 or #1001) or the order ID: index both.
    for (const key of [clean(name), clean(r[col.Id])]) if (key) byId.set(key, order);
  }

  // GA4: skip blank lines and # comments before the header; then Transaction ID, Transactions.
  const ga = parseCsv(ga4Csv.replace(/^\uFEFF/, "")).filter((r) => r.join("").trim());
  const header = ga.findIndex((r) => !r[0].startsWith("#"));
  const unknown = [];
  for (const [id, count] of header < 0 ? [] : ga.slice(header + 1)) {
    const n = Number(count);
    if (!Number.isFinite(n)) throw new Error(`GA4 row "${id},${count}": the second column must be
Transactions`);
    const order = byId.get(clean(id));
    if (order) order.hits.push({ id, n });
```

```
    else unknown.push({ id, n });
  }

  const all = [...orders.values()];
  return {
    orders: all.length,
    missing: all.filter((o) => o.hits.length === 0),
    duplicated: all.filter((o) => o.hits.reduce((s, h) => s + h.n, 0) > 1),
    unknown,
  };
}

if (import.meta.url === pathToFileURL(process.argv[1]).href) {
  const [ordersFile, gaFile] = process.argv.slice(2);
  const r = reconcile(readFileSync(ordersFile, "utf8"), readFileSync(gaFile, "utf8"));
  console.log(`Shopify orders: ${r.orders} · missing in GA4: ${r.missing.length} · duplicated:
${r.duplicated.length} · unknown IDs: ${r.unknown.length}`);
  for (const o of r.missing) console.log(`MISSING     ${o.name}  source=${o.source}  payment=${o.payment}
created=${o.created}`);
  for (const o of r.duplicated) console.log(`DUPLICATED  ${o.name}  ${o.hits.map((h) =>
`${JSON.stringify(h.id)}×${h.n}`).join(" + ")}`);
  for (const u of r.unknown) console.log(`UNKNOWN     ${JSON.stringify(u.id)}×${u.n}`);
}
```

Ejecútalo con `node reconcile.mjs orders_export.csv ga4-transactions.csv`, con los nombres de tus archivos. Si falta una columna en la exportación de Shopify o la segunda columna de GA4 no es un número, se detiene con un error que lo dice.

PASO 05

Lee el resultado

La primera línea resume cuántos pedidos faltan, cuántos están duplicados y cuántos IDs de GA4 no son de ningún pedido. Cada línea siguiente empieza con `MISSING`, `DUPLICATED` o `UNKNOWN`; los faltantes traen `source`, `payment` y `created`. El artículo muestra una salida completa con diez pedidos ilustrativos.

LÍNEA	QUÉ SIGNIFICA	QUÉ HACES
`MISSING` con `source` pos o draft order	Venta fuera de la web	Nada: GA4 no la iba a ver
`MISSING` web con pago externo	El cliente no volvió a la página de gracias	Cuéntalas aparte
`MISSING` web cerca de medianoche	Zona horaria distinta	Iguala las zonas
`MISSING` web, el resto	Cookies rechazadas o bloqueadores	Es tu línea base
`DUPLICATED`	Dos tags o dos formatos de ID	Deja un solo tag
`UNKNOWN` vacío o `(not set)`	Compra sin `transaction_id`	Corrige el tag
`UNKNOWN` con otro valor	Otro formato de ID u otro día	Revísalo en DebugView