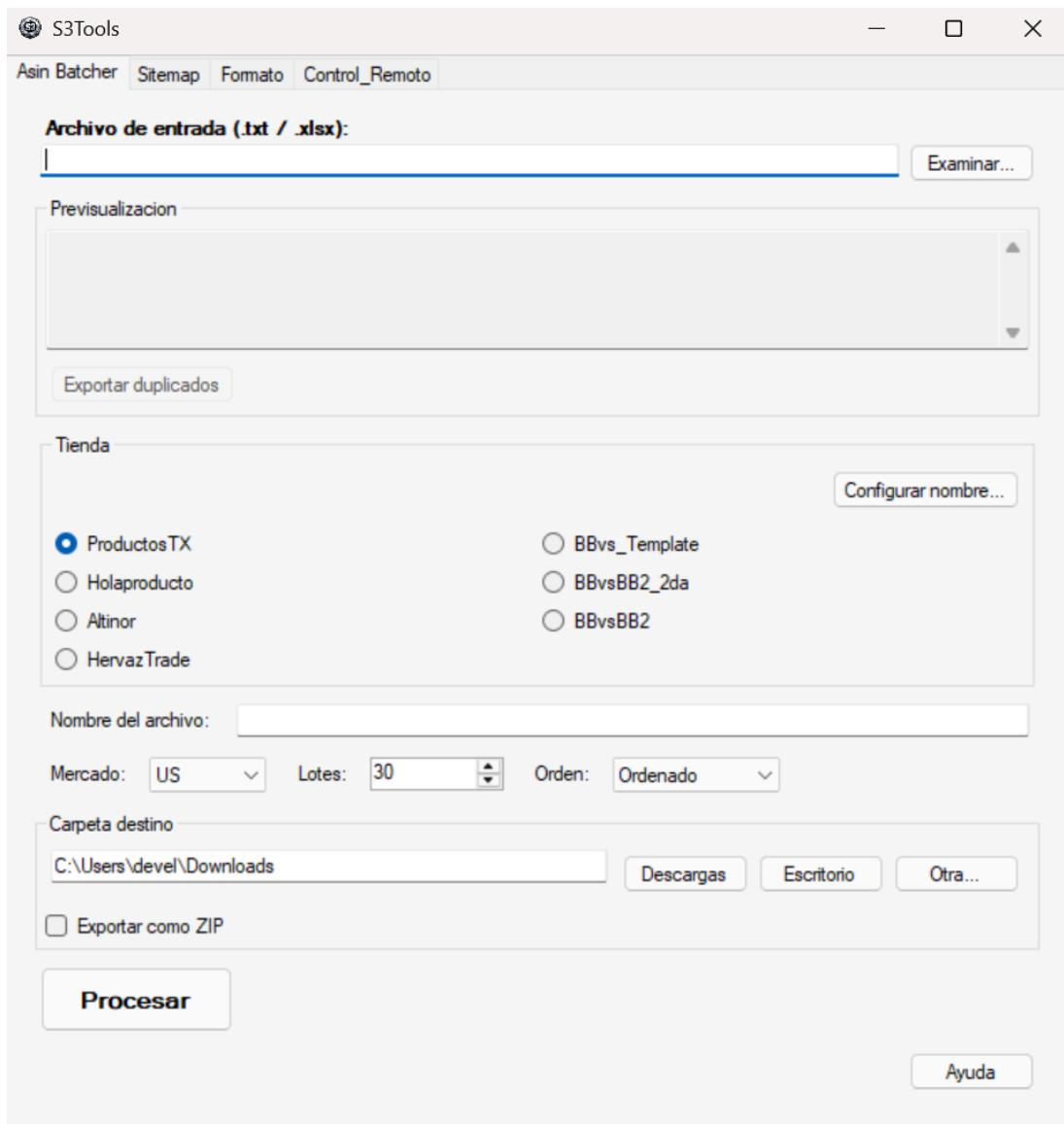


Manual de usuario S3Tools

Base: repositorio [S3Integracion/S3Tools](#) (WinForms + motores Python). La app unifica 3 herramientas: **Asin Batcher**, **Sitemap** y **Formato** para preparar/transformar insumos y salidas de **Web Scraper**.

Nota: en la UI existe una pestaña “Control Remoto”, pero **no tiene lógica implementada actualmente**.



Contenido

1. ¿Qué es S3Tools? y flujo general
 2. Requisitos e instalación (usuario final)
 3. Arquitectura (para entender qué hace “por debajo”)
 4. Conceptos clave (ASIN, lote, start_url, sitemap WebScraper, plantilla)
 5. Asin Batcher (uso + detalle funcional)
 6. Sitemap (uso + detalle funcional)
 7. Formato (uso + detalle funcional)
 8. Persistencia de rutas / “Cargar último lote”
 9. Nombres de salida y saneado de caracteres
 10. Exportar como ZIP
 11. Troubleshooting (problemas típicos y soluciones)
 12. Glosario y buenas prácticas
-

1) ¿Qué es S3Tools y cuál es el flujo típico?

S3Tools es una app de escritorio Windows que centraliza el flujo: **ASIN** → **lotes de URLs** → **sitemaps JSON** → **normalización de columnas** .

Flujos comunes:

- **Flujo A (URLs por lote):**
 1. Asin Batcher genera archivos **.txt** con URLs (con encabezado **start_url**) →
 2. Se usan esos lotes como insumo en tu proceso (por ejemplo, en WebScraper).
 - **Flujo B (Sitemaps WebScraper):**
 1. Asin Batcher genera lotes →
 2. Sitemap convierte esos lotes en **sitemaps JSON** en formato WebScraper usando plantillas .
 - **Flujo C (normalizar exportaciones WebScraper):**
 1. WebScraper exporta **.csv/.xlsx** →
 2. Formato normaliza **las primeras dos columnas** (según plantilla) .
-

2) Requisitos e instalación (usuario final)

Requisitos

- UI: **WinForms (.NET Framework 4.7.2)** .
- Motores:
 - Opción 1 : usar motores empaquetados `.exe` → **no requiere Python instalado** .
 - Opción 2: usar motores `.py` → requiere **Python 3.12** y dependencias según motor:
 - Asin Batcher: `pandas + openpyxl`
 - Sitemap: `openpyxl`
 - Formato: `openpyxl`

Instalación / ejecución

Este programa se distribuye como recurso de la empresa S3 Integración para su uso libre dentro del grupo laboral y uso controlado establecido en la Licencia EULA establecida para uso del programa.

3) Cómo funciona por dentro (para entender errores y rutas)

S3Tools usa:

- **UI** en C# (WinForms).
- **Motores** Python ejecutados como subprocesso, comunicándose por JSON (stdin/stdout) .

El cliente C# resuelve el motor en este orden :

1. **Variable de entorno** (ruta manual).
2. **Motor embebido** como recurso (si existiera).
3. Ejecutable local: `Engines/<Motor>/<script>.exe`.
4. Script local: `Engines/<Motor>/<script>.py` usando `python`.

Variables de entorno soportadas :

- ASIN_BATCHER_ENGINE_PATH
 - SITEMAP_ENGINE_PATH
 - FORMATO_ENGINE_PATH
-

4) Conceptos clave (lo que verás en la UI)

ASIN

Identificador Amazon. El motor valida ASIN como **10 caracteres alfanuméricos** en mayúscula (patrón tipo `[A-Z0-9]{10}`) y “limpia” entradas (trim / mayúsculas / etc.) antes de validar .

Lote (batch)

Un “lote” es un subconjunto de ASINs/URLs. Asin Batcher puede dividir en *N lotes* y luego escribir un archivo por lote .

Encabezado `start_url`

Los `.txt` de salida incluyen una primera línea `start_url` y luego una URL por línea .

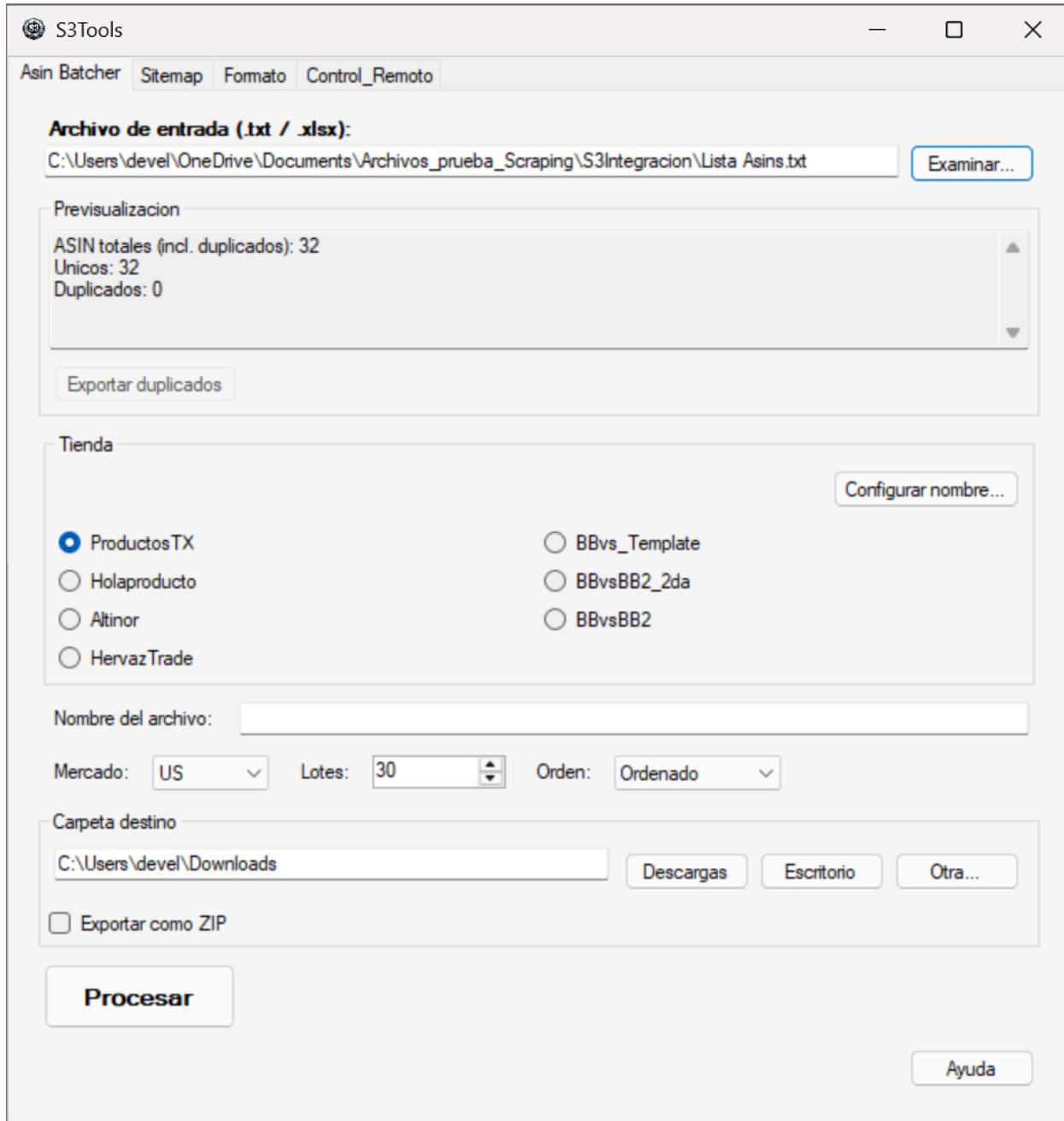
Sitemap (formato WebScraper)

Sitemap genera `.json` usando plantillas “PlantillaSitemaps*.json” (WebScraper) y ajusta dinámicamente `startUrl` + URLs del lote dentro del template .

Plantilla (Auto / Tiendas / BBvs)

- En Sitemap hay plantillas por tipo: Tiendas vs BBvs .
 - En Formato, la plantilla define cómo se renombran/normalizan **las primeras 2 columnas** .
-

5) Módulo: Asin Batcher



5.1 ¿Para qué sirve?

- Carga ASINs desde **TXT/XLSX** (y puede detectar “reporte de inventario” por patrón).

- Elimina duplicados, opcionalmente exporta duplicados.
- Genera lotes de URLs por **mercado** (US/MX) y un archivo `.txt` por lote con encabezado `start_url`.

5.2 Pantalla y campos (qué significa cada cosa)

En la UI se ven (nombres aproximados):

- **Archivo de entrada:** selector con filtro `.txt`, `.xlsx`, `.xls` .
- **Tienda:** opciones (ej.: Amazon, ProductosTX, Altinor, HolaProducto, GameStore, BBvs) .
- **Mercado:** `MX` o `US` .
- **Orden:** `Ordenado`, `Reverso`, `Aleatorio` .
- **Cantidad de lotes:** entero `>= 1` (validado contra la cantidad de ASINs únicos) .
- **Nombre del archivo / base:** texto que afecta el naming de salida y reglas de “saneado” (ver sección 9).

Botones típicos:

- **Procesar**
- **Exportar duplicados** (si existe)
- **Exportar como ZIP** (si existe)
- **Configurar nombre** (prefijos)

[Imagen sugerida] Pestaña “Asin Batcher” con todos los controles.

5.3 Formatos de entrada soportados y cómo se interpretan

A) TXT “simple”

Cada línea se interpreta como un ASIN potencial. El motor limpia y valida; lo no válido se descarta .

B) Excel `.xlsx/.xls`

Lee el **primer** campo/columna que contenga ASINs y aplica limpieza/validación. Para `.xlsx` usa `pandas+openpyxl` si corres en modo `.py` .

C) “Reporte de inventario” (detección automática)

El motor intenta reconocer un reporte por:

- Nombre tipo: `Reporte de inventario_dd-mm-yyyy` o
- Encabezado TSV que contenga `ASIN`.
Si lo detecta, parsea como **tab-separated** y extrae ASINs válidos (ignorando encabezados) .

5.4 Qué hace exactamente al “Procesar”

1. **Extrae ASINs** del archivo (según el tipo de entrada) .
2. Separa:
 - lista de **únicos**
 - lista de **duplicados** (opcional para exportación)
3. Aplica **reordenamiento** según “Orden”:
 - Ordenado: sort asc
 - Reverso: sort desc
 - Aleatorio: shuffle
4. Divide en lotes (chunking) y valida que el número de lotes solicitado no exceda el total de ASINs únicos (también se valida desde la UI) .
5. Convierte cada ASIN a URL según **mercado**:
 - **US** → <https://www.amazon.com/dp/<ASIN>>
 - **MX** → <https://www.amazon.com.mx/dp/<ASIN>>
6. Escribe archivos `.txt` por lote:
 - Primera línea: `start_url`
 - Luego una URL por línea
7. Crea carpeta de salida con timestamp:
 - `AsinBatch_<base>_<ddmmyy>_<hhmm>`
8. Si activas ZIP, comprime y elimina carpeta intermedia .

5.5 Exportar duplicados (qué genera)

Si usas “Exportar duplicados”, se crea un CSV con nombre `duplicates_<timestamp>.csv` y encabezado `asin`, contenido duplicados únicos (la UI invoca este flujo y luego abre la carpeta) .

5.6 Configurar nombre (prefijos / modo naming)

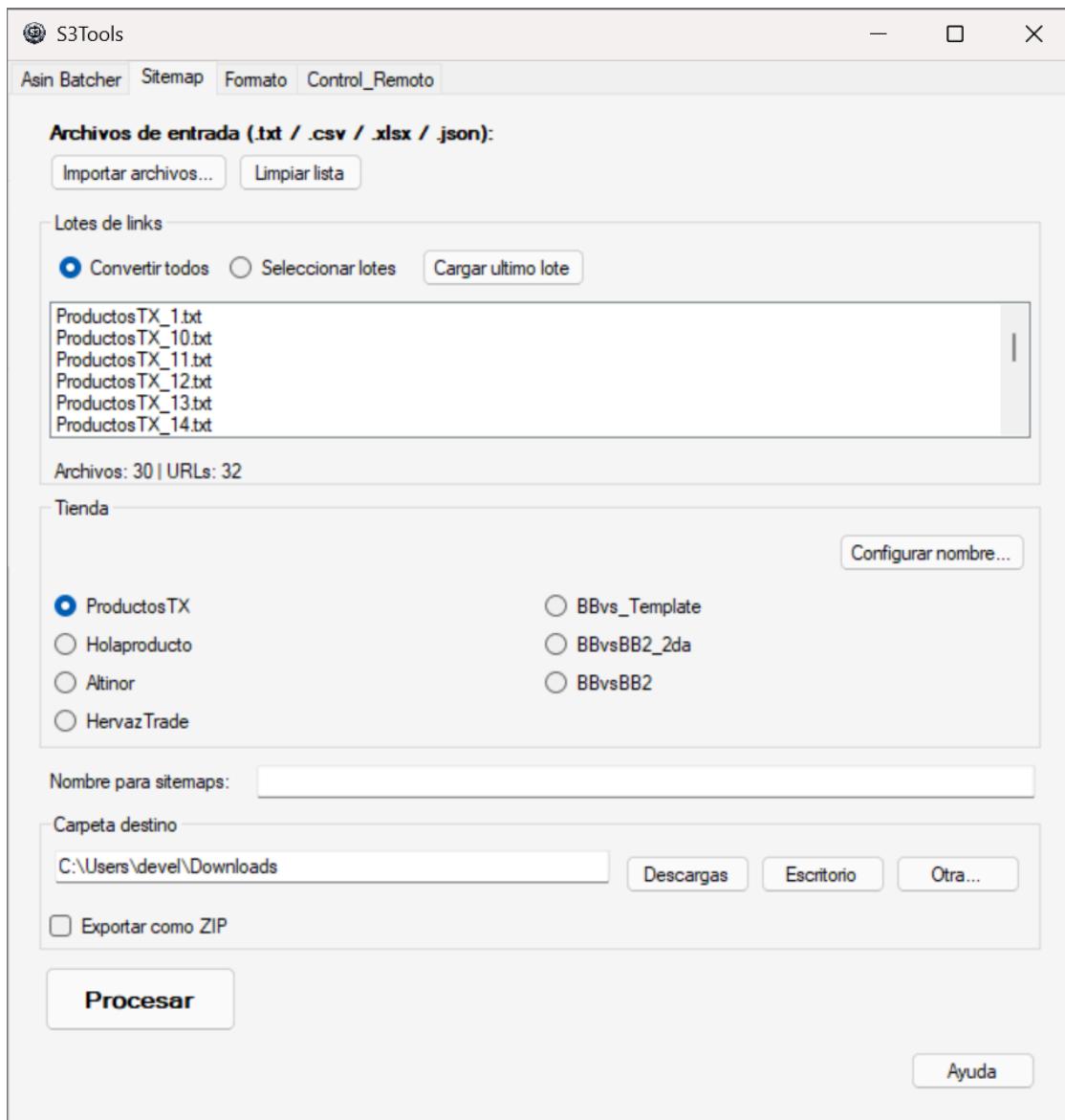
Hay un diálogo de configuración para formar nombres del tipo:

Prefijo1_NombreTienda_Prefijo2_<NombreBase>_<n> (concepto de prefijos + tienda + base) .

Desde UI existe el botón “Configurar nombre” .

Importante: el saneado de caracteres (ver sección 9) se aplica para evitar caracteres inválidos en nombres de archivo .

6) Módulo: Sitemap



6.1 ¿Para qué sirve?

Convierte lotes de URLs (por ejemplo, salidos del Asin Batcher) en **sitemaps JSON** listos para WebScraper usando plantillas .

6.2 Modos de uso en la UI

Normalmente hay 2 formas:

- **Convertir todos**: seleccionas una carpeta y convierte todos los lotes encontrados (p.ej. los `.txt` generados) .
- **Seleccionar lotes**: eliges manualmente uno o más archivos de lote .

Además existe “**Cargar último lote**” que apunta a la última salida del Asin Batcher (ver sección 8).

[Imagen sugerida] Pestaña “Sitemap” mostrando botones Convertir todos / Seleccionar / Cargar último lote.

6.3 Entradas soportadas (motor)

El motor de sitemap acepta múltiples extensiones:

`.txt, .csv, .xlsx, .xls, .json, .xml` .

(En UI puede que el selector muestre un subconjunto; el motor soporta más.)

Caso típico: `.txt` desde Asin Batcher

El motor **elimina** automáticamente la primera línea `start_url` si existe y toma el resto como URLs.

6.4 Tienda, dominio y plantilla

El motor mantiene un mapa tienda → dominio `startUrl` (ej.: Amazon MX/US, ProductosTX, HolaProducto, Altinor, GameStore, BBvs, etc.) .

Y selecciona plantilla así:

- Si el tipo es **BBvs** usa `PlantillaSitemapsBBvs.json`
- En caso contrario usa `PlantillaSitemapsTiendas.json`
(Esto coincide con la guía de README sobre plantillas) .

Luego:

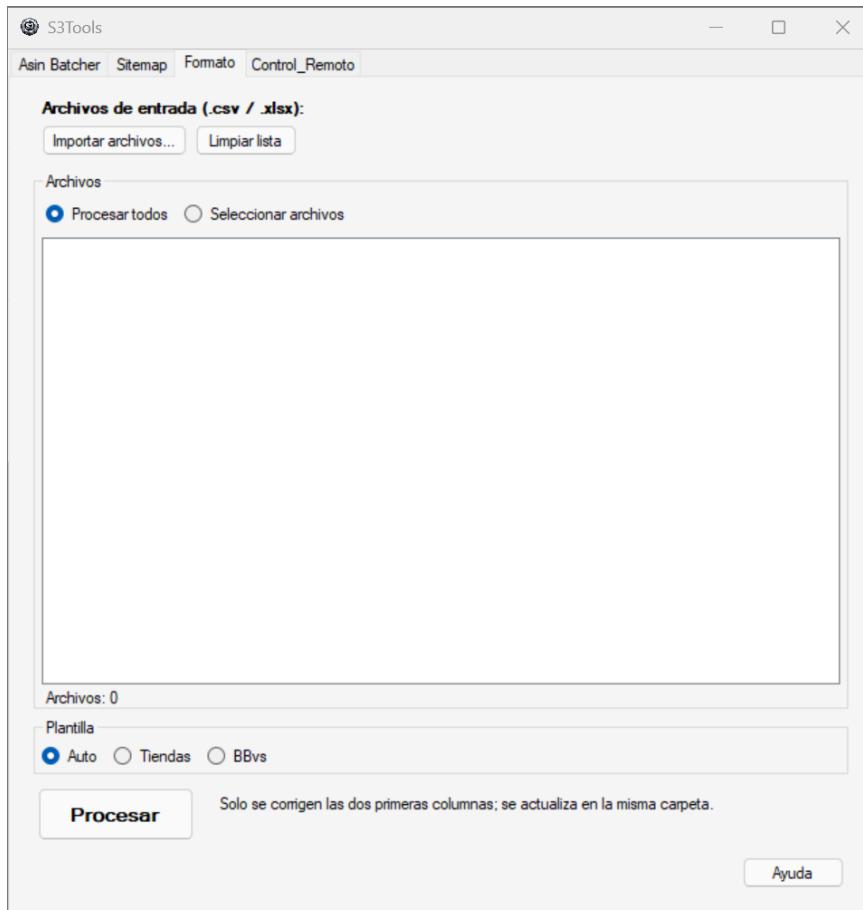
- Reemplaza `template_data["startUrl"][0]` por el dominio real de la tienda .
- Inserta el set de URLs del lote en el JSON resultante .

6.5 Salidas: carpeta, nombres y ZIP

Al procesar:

- Crea carpeta: `SITEMAP_<base_name>_<ddmmyy>_<hhmm>` .
 - Genera un `.json` por lote (y si “Exportar como ZIP” está activo, zippea y borra la carpeta) .
-

7) Módulo: Formato



7.1 ¿Para qué sirve?

Normaliza las **primeras dos columnas** de exportaciones de WebScraper (CSV/XLSX) según una plantilla: **Auto / Tiendas / BBvs** .

7.2 Comportamiento clave: edita “en la misma carpeta”

La UI indica que el archivo **se actualiza en la misma carpeta** y el motor efectivamente **sobrescribe** los archivos procesados (no crea “_fmt” separado) cuando trabaja directamente sobre una ruta dada .

Recomendación operativa: si necesitas preservar el original, copia los archivos a una carpeta “_backup” antes de ejecutar Formato.

7.3 Entradas soportadas

- `.csv, .xlsx, .xls` (por motor) .

7.4 Plantillas (Auto / Tiendas / BBvs) y “Solo primeras dos columnas”

- La plantilla define cómo se renombran/reestructuran las primeras columnas (reglas de normalización) .
 - Si se activa “Solo las primeras dos columnas”, el motor recorta la tabla a 2 columnas antes de aplicar el formato .
-

8) Persistencia de rutas y “Cargar último lote”

S3Tools guarda la última carpeta de salida del Asin Batcher en:

`%LocalAppData%\S3Integracion\last_asin_output_dir.txt` .

Luego, en la pestaña Sitemap, “Cargar último lote” usa esa ruta para ayudarte a ubicar rápidamente los lotes .

9) Nombres de salida y saneado de caracteres (muy importante)

Cuando un nombre viene del usuario (base name / prefijos / tienda), se “sanea”:

- Solo permite `a-zA-Z0-9_()`+- y reemplaza lo demás por `_`.
 - Esto evita errores por caracteres inválidos en Windows.
-

10) Exportar como ZIP (Asin Batcher y Sitemap)

- Si activas ZIP en Asin Batcher: comprime la carpeta de salida y luego la elimina .
- Si activas ZIP en Sitemap: mismo patrón (zip + cleanup) .

Buenas prácticas:

- ZIP ON cuando vas a **enviar** resultados por correo/drive.
 - ZIP OFF si vas a **iterar** (revisar/editar) archivos.
-

11) Troubleshooting (problemas típicos)

11.1 “El motor no responde / no se ejecuta”

- Verifica si está usando `.exe` o `.py` y que exista en `Engines/<Motor>/` .
- Si usas `.py`, confirma Python 3.12 y dependencias (`openpyxl`, `pandas`) .
- Si usas rutas personalizadas, revisa variables de entorno `*_ENGINE_PATH` .

11.2 “Cantidad de lotes inválida”

- No puedes pedir más lotes que ASINs únicos. Se valida en UI y también en motor .

11.3 “Mi archivo no se carga”

- Revisa extensión soportada (Asin: `.txt/.xlsx/.xls` ; Sitemap: varias ; Formato: `.csv/.xlsx/.xls`).
- Cierra Excel si tiene el archivo abierto (bloqueo de escritura es frecuente).

11.4 “Formato me cambió archivos y quería conservar el original”

- Formato actualiza en la misma carpeta . Copia a una carpeta duplicada antes de correr si no quieres que se vea afectado el archivo original.
-

12) Glosario y buenas prácticas

ASIN: identificador Amazon (10 caracteres) .

Lote: subconjunto de ASINs/URLs.

start_url: encabezado requerido en los `.txt` de lotes .

Sitemap (WebScraper): JSON construido desde plantilla + URLs .

Plantilla: define estructura para scraping o normalización.

Buenas prácticas rápidas:

- Nombra outputs con un **base_name** consistente (campaña/fecha) para trazabilidad.
 - Exporta duplicados si el resultado “no cuadra” (te ayuda a ver repetidos) .
 - Mantén una carpeta por corrida; si usas ZIP, conservarás un artefacto único.
-