# Especificación Técnica del Proyecto: ScrapSAE

## 1. Resumen Ejecutivo

**ScrapSAE** es una solución de software diseñada para ejecutarse de forma local en el mismo entorno donde reside **Aspel SAE**. Su función principal es actuar como un integrador inteligente que extrae información de productos de múltiples proveedores mediante técnicas de *web scraping*, procesa y enriquece los datos usando Inteligencia Artificial, y los sincroniza de forma bidireccional entre el inventario local (SAE) y una tienda en línea externa mediante Webhooks. La arquitectura prioriza la seguridad al no requerir apertura de puertos entrantes y utilizar una base de datos de paso (SQLite) para validación de datos.

## 2. Requerimientos Funcionales (RF)

### RF1: Orquestador Multi-Sitio y Multi-Configuración

El sistema debe ser capaz de gestionar perfiles independientes para cada proveedor.

* **Detalle:** Cada perfil de sitio debe incluir la URL base, selectores de navegación (CSS/XPath) y reglas de comportamiento (ej: si requiere login).
* **Flexibilidad:** Debe permitir añadir nuevos proveedores sin necesidad de recompilar todo el código, preferiblemente mediante archivos de configuración o tablas en la BD local.

### RF2: Planificador de Tareas Dinámico (Scheduler)

Permitir la ejecución automatizada basada en tiempo.

* **Detalle:** Se debe poder configurar que el "Proveedor A" se analice cada lunes a las 3:00 AM y el "Proveedor B" diariamente a las 11:00 PM.
* **Control:** Debe permitir ejecuciones manuales inmediatas desde el dashboard para pruebas o urgencias.

### RF3: Extracción Robusta (Web Scraping)

Capacidad de navegar y leer sitios web complejos.

* **Detalle:** Uso de navegadores automatizados para renderizar JavaScript, manejar paginación, scroll infinito y descarga de imágenes.
* **Bajas Automáticas:** Si un producto registrado previamente ya no se encuentra en el sitio tras N intentos, el sistema debe marcarlo para descontinuación.

### RF4: Procesamiento y Categorización con IA

Uso de modelos de lenguaje (LLM) para la limpieza de datos.

* **Detalle:** La IA debe recibir el texto bruto extraído y transformarlo en una ficha técnica estructurada (Nombre, Marca, Modelo, Características). Aunque también debe mantener en la base de datos la información del producto tal como aparece en la página del proveedor para que se conserve la referencia exacta para hacer comparaciones. Aunque debe hacer lo posible por obtener el sku para tener un valor llave del producto.
* **Mapeo de Categorías:** La IA debe asignar el producto a una "Línea de Producto" de SAE basándose en el catálogo de categorías predefinido.

### RF5: Almacenamiento Local de Paso (Staging)

Garantía de integridad de datos antes de la carga final.

* **Detalle:** Toda la información extraída debe guardarse primero en una base de datos SQLite local.
* **Estado de Validación:** Los productos permanecerán en estado "Pendiente" hasta que el sistema (o un humano) valide que la información es coherente.

### RF6: Integración con Aspel SAE (ODBC)

Escritura y lectura directa en el ERP.

* **Detalle:** Actualización de la tabla INVE01 (descripciones, precios, existencias y campos libres) utilizando el driver ODBC de Firebird o SQL Server.
* **Seguridad:** El sistema debe verificar que la conexión sea estable antes de iniciar un proceso de actualización masiva.

### RF7: Notificación Externa vía Webhooks

Sincronización con la tienda en línea.

* **Detalle:** Tras cada cambio exitoso en SAE, el sistema debe disparar una petición HTTPS POST a la API de la tienda en línea con el SKU, precio, stock y estado de disponibilidad.

## 3. Requerimientos No Funcionales (RNF)

### RNF1: Seguridad Perimetral

* **Aislamiento:** El sistema no debe exponer servicios hacia internet. Toda la comunicación con la IA y con la Tienda en Línea debe ser mediante peticiones salientes protegidas por TLS 1.2+.

### RNF2: Gestión de Errores y Reintentos

* **Resiliencia:** Implementar políticas de reintento (Exponential Backoff) para el scraping y las llamadas a la API de IA en caso de fallos de red.
* **Logs:** Registro detallado de cada paso en un archivo diario rotativo para depuración.

### RNF3: Eficiencia de Recursos

* **Modo Headless:** Los navegadores de scraping deben correr sin interfaz gráfica para minimizar el impacto en la memoria RAM del servidor de SAE.

## 4. Stack Tecnológico y Organización de Datos

### Frameworks y Herramientas

* **Backend:** .NET 8/9 Worker Service (Servicio de Windows).
* **Scraping:** Playwright for .NET.
* **IA:** OpenAI API SDK / Semantic Kernel.
* **Base de Datos Local:** SQLite (Archivo scrapsae\_staging.db).
* **Acceso a Datos:** Dapper o Entity Framework Core (para SQLite) y System.Data.Odbc (para SAE).
* **Frontend Local:** React + Vite (Dashboard administrativo alojado localmente).

### Estructura de Datos (Organización)

1. **Config\_Sites:** Almacena URLs, horarios (Cron expressions) y credenciales encriptadas.
2. **Staging\_Products:** Tabla temporal con campos: SKU\_Source, Raw\_Data, AI\_Processed\_JSON, Status (Pending, Synced, Error).
3. **Category\_Mapping:** Tabla que relaciona categorías de proveedores con claves de línea de SAE.
4. **Sync\_Logs:** Historial de envíos de Webhooks y resultados de la sincronización.