SGA Lince · Imar

Propuesta de cambios para el correo de Sergio Guarnido del 25/11/2026 («RE: Notas revisión SGA Lince»)
Preparado el 25/11/2026 por Atalaya-Motor para Agustín Guardeño · **Estado: solo análisis, no se ha cambiado nada**

> **Qué es este documento.** Explica cómo haría cada cambio en el SGA Lince de Imar, después de leer el código que está en marcha. **No se ha tocado ningún fichero, ajuste ni base de datos**, y no se ha probado nada contra el SAP de Imar (desde el entorno no se llega). Nada de esto está publicado ni lo ve Imar.

Resumen

#	Petición	Dónde se toca	A quién afecta	Esfuerzo	Depende de Imar
1	Postventa: disponible de los almacenes 01 y 04	lib/imar.js, lib/sap/documentos.js, lib/procesos.js	Solo Imar	Medio	Confirmar si el 04 lleva ubicaciones
2	Llevar a carros: sacar también del 04	lib/imar.js, lib/imar-carros.js	Solo Imar	Medio	Lo mismo
3	Postventa: filtros «Disponibles» y «Todas» en las líneas	publico/terminal/app.js	Solo Imar (detrás de `imar-postventa`)	Bajo	No
4	Postventa: leer el código de barras del pedido de destino	rutas/imar.js, lib/staging.js, app.js	Solo Imar	Medio	Qué lleva ese código de barras
5	Botón de imprimir el albarán (Crystal «Albarán de Venta (EC)», 3 copias, idioma)	lib/imar.js, rutas/imar.js, app.js	Solo Imar	Alto	Versión de SAP, impresora
6	Embalaje desde el pedido (pedidos cerrados con algún albarán abierto)	lib/procesos.js, rutas/imar.js, app.js, lib/imar.js	Solo Imar (proceso nuevo)	Alto	Validar el criterio
7	Error «Falta configurar la ruta de las fotos del packing»	Ajuste `ajustes.imar.fotosRuta` + montaje	Solo Imar	Bajo	Carpeta de destino
8	Bultos estándar (cajas, tubos, cajones) leyendo su etiqueta	SAP (tipos de embalaje) + app.js, contabiliza.js	Solo Imar	Medio	Dar de alta los tipos
9	Ficha técnica: tarda unos 15 s	rutas/api.js, lib/imar.js	Todos (mejora común) + Imar	Medio	Medir en producción
10	Ficha técnica: el disponible no suma lo solicitado	rutas/api.js, app.js	Todos, o solo Imar (a decidir)	Bajo	No
—	Service Layer que se cae (correo de las 10:44)	Servidor SAP de Imar, no el SGA	—	—	Su informática (Aldaro)

> **Regla multicliente.** Todo lo de `lib/`, `rutas/` y `publico/` es común. Los cambios del 1 al 8 van detrás de `esImar(t)`, de un proceso con `soloTenants: ['imar']` (`imar-postventa`, `imar-carros`) o de un ajuste de `ajustes.imar`, así que no cambian nada a Lafermu ni a demo. Los puntos 9 y 10 están en la ficha de artículo, que es común: lo señalo en cada uno.

1. Picking postventa: disponible de los almacenes 01 y 04

Qué pasa hoy

- La lista de pedidos usa `imar.disponiblePedidos` (`lib/imar.js`). Su consulta `SGA_IMAR_DISP_PED` cruza cada línea del pedido con `OITW` **solo del almacén de la línea** (`W."WhsCode" = T1."WhsCode"`, que en Imar es el 01).
- Dentro del pedido, el «Disponible» y la ubicación sugerida de cada línea salen de `sugerirUbicaciones` (`lib/sap/documentos.js`), que también mira solo `l.almacen`.
- La entrega se crea con `WarehouseCode = almacén del apunte` (`contabiliza.documentoComercial`). Si la pieza se coge de una ubicación del 04, la línea de la entrega puede ir al 04 (SAP lo permite en una entrega desde pedido).

Cómo lo haría

- **Ajuste nuevo** `ajustes.imar.almacenesPostventa`, por defecto `["01","04"]`, leído en `imar.config()` y añadido a las claves de `scripts/imar-config.js`.
- **Lista de pedidos**: en `disponiblePedidos`, cambiar el JOIN a `W."WhsCode" IN ('01','04')` (con los almacenes del ajuste), traer `T1."LineNum"` y sumar el `OnHand` por línea en JS. El validador del Service Layer de Imar no admite `SUM`, por eso ya se suma en JS. Cambiar también el nombre de la consulta guardada (p. ej. `SGA_IMAR_DISP_PED2`) para que no choque con la antigua.
- **Líneas del pedido**: en `sugerirUbicaciones`, solo si `proceso.id === 'imar-postventa'`, pedir el stock de cada almacén del ajuste (`stock.consultarVarios` por almacén, como ya hace), sumarlo en `l.stock` y sugerir la ubicación con más cantidad entre los dos. Añadir a la línea `l.stockPorAlmacen = {"01": x, "04": y}` para pintarlo.
- **Recoger del 04**: dar a `imar-postventa` la opción `almacenPorUbicacion: true` (ya existe para entrada y salida), para que el almacén del apunte salga de la ubicación leída. Hay que revisar el desglose del terminal (`app.js`, hacia la línea 4585), porque con documento base hoy se usa el almacén de la línea. Habría que dejar que la ubicación leída sea de cualquiera de los almacenes del ajuste. `staging.exigirStock` ya comprueba el stock con `apunte.almacen`, así que vale tal cual.

> **A confirmar con Imar:** si el almacén 04 («Almacén TIP») lleva ubicaciones. Si no las lleva, su stock no está en `OIBQ`: habría que leerlo de `OITW` y, al coger del 04, apuntar sin ubicación. Y si el disponible debe restar lo comprometido por otros pedidos (hoy se usa el stock físico, como en la lista actual).

2. Llevar a carros: sacar también del almacén 04

Qué pasa hoy

- `imar.config()` tiene `almacenOrigen: '01'` (un solo almacén). `imar-carros.construir` lee el stock por ubicación con `SQL_ORIGEN ... WHERE T0."WhsCode" = :almacen`, y `recoger` apunta siempre `almacen: c.almacenOrigen`.
- El traspaso ya admite un almacén de origen distinto en cada línea (`FromWarehouseCode`), así que un carro con piezas del 01 y del 04 se puede traspasar en un solo documento.

Cómo lo haría

- Ajuste nuevo `almacenesOrigen: ["01","04"]`, manteniendo `almacenOrigen` como el primero para no romper lo que lo usa.
- `SQL_ORIGEN`: traer también `T0."WhsCode"` y lanzar una consulta por almacén y juntarlas (más seguro que `IN` con el validador del Service Layer). Guardar el almacén en cada fila de `origen`.
- En `armar` (reparto de lo «por llevar»), recorrer las ubicaciones de los dos almacenes: primero las del 01 y luego las del 04 (orden a confirmar con Imar). Cada pick lleva su `almacen`.
- En `recoger`, usar `fila.almacen` en vez de `c.almacenOrigen`. La clave que agrupa las líneas del traspaso ya incluye el almacén.
- En «otros almacenes» (línea ~482) excluir los dos almacenes de origen, no solo el 01.
- En pantalla, mostrar el almacén junto a la ubicación de origen para que el operario sepa a cuál ir.

3. Postventa: filtros «Disponibles» y «Todas» en las líneas

Qué pasa hoy

La pantalla de líneas (`pantallaLineas`, `app.js` ~3696) es **la misma para los treinta procesos** y pinta «Pendientes · n» / «Todas · n».

Cómo lo haría (solo para `imar-postventa`)

- Si `p.id === 'imar-postventa'`, el segmento pasa a «**Disponibles · n**» (por defecto) y «**Todas · n**». En el resto de procesos se queda como está.
- **Disponibles**: líneas sin completar que no son inventariables (`l.inventariable === false`: PORTES., horas…) o que tienen `l.stock > 0` (con el stock de 01+04 del punto 1). Si hay stock pero no llega, se muestra con la marca «Parcial: hay x».
- **Todas**, por este orden: primero las disponibles, luego las pendientes sin stock (marca roja «Sin stock») y al final las completas en verde («Completa»; el estilo `preparada` ya existe).
- Mensaje propio cuando no hay ninguna: «No hay piezas disponibles para albaranar. Toca «Todas» para ver las que faltan.»

Este fichero se sirve en cuanto se guarda (el service worker va primero a la red): no hace falta reiniciar, pero sí recargar la página del terminal.

4. Postventa: leer el código de barras del pedido de destino

Cómo lo haría (como en Llevar a carros)

- Al terminar de recoger (o pieza a pieza, a elegir), el terminal dice «Deja las piezas junto al pedido 1904789 y lee su etiqueta». El operario lee el código de barras del pedido.
- Ruta nueva `POST /api/imar/postventa/:transaccion/pedido` en `rutas/imar.js`. Compara lo leído con el `DocNum` del pedido base, aceptando el número solo o con prefijo (p. ej. `/^(?:P[\s-]*)?(\d{1,9})$/`, como hace `carros.leerOf` con «OF»). Si coincide, guarda `cabecera.pedidoLeido = true` en la transacción. Si no, avisa: «Ese es el pedido X, no el 1904789».
- Solo para `imar-postventa` y con un ajuste `exigirLeerPedidoPostventa` (por defecto true, igual que `exigirLeerOfCarro`), no dejar contabilizar sin esa lectura. El botón «Contabilizar» queda desactivado con el texto «Lee la etiqueta del pedido».
- Recomiendo una sola lectura al final (un pedido = un sitio), salvo que en postventa las piezas de un pedido se dejen en varios sitios.

A confirmar con Imar: qué etiqueta o documento tiene el pedido de destino y qué lleva su código de barras (¿el nº de pedido? ¿con prefijo?). Que nos manden una foto, como hicieron con la etiqueta de la OF.

5. Imprimir el albarán tras contabilizar

Qué pasa hoy

El SGA no imprime ningún informe de Crystal: tras contabilizar sale `pantallaHecho` con el documento creado.

Cómo lo haría

- **Que el PDF lo genere el propio SAP** con la plantilla de Crystal «Albarán de Venta (EC)». Las versiones recientes de SAP Business One para HANA permiten exportar un informe de Crystal a PDF por el servicio de informes que acompaña al Service Layer (por ejemplo, `/rs/v1/ExportPDFData`, con el código de la plantilla de `RDOC` y el `DocEntry`). **Hay que comprobar en el servidor de Imar que su versión lo tiene y cómo se llama**: no lo he podido verificar desde aquí.
- Ajustes: `ajustes.imar.plantillaAlbaran` (código de la plantilla en `RDOC`, que se busca una vez por su nombre) y `copiasAlbaran: 3`.
- Ruta nueva `GET /api/imar/albaran/:entry/pdf` en `rutas/imar.js` (ese fichero ya responde 404 a los demás clientes). Pide el PDF a SAP y lo devuelve con `Content-Type: application/pdf`. Se anota en el registro (`bd.anotar`).
- En `pantallaHecho`, solo con `esImar()` y si el documento creado es una entrega (`DeliveryNotes`): botón «**Imprimir albarán (3 copias)**», que abre el PDF.
- **Idioma**: si hoy la plantilla sale en inglés o en español según el cliente al imprimir desde SAP, debería salir igual cuando SAP la genere con la misma plantilla y el mismo documento. Hay que comprobarlo con un cliente de cada idioma.
- **Las 3 copias**: desde un PC o una PDA se abre el cuadro de imprimir del sistema y hay que poner 3 copias a mano (o el PDF lleva las tres copias seguidas). Para que salga solo, sin cuadro, habría que imprimir desde el servidor directamente en una impresora de red de Imar a través del agente de conexión (como se hizo con la carpeta de fotos). Es más trabajo y depende de su informática.

> **A decidir:** (a) PDF en pantalla y el operario imprime las 3 copias (rápido, sin montar nada más), o (b) impresión directa en una impresora de red de Imar (hay que dar acceso a su impresora por el agente). Y confirmar la versión de SAP de Imar para exportar el Crystal.

6. Embalaje desde el pedido

Qué pasa hoy

- El proceso `packing` (común a todos) parte de un albarán (`DeliveryNotes`). Los bultos se guardan en el propio albarán (`DocumentPackages`, `contabiliza.guardarBultos`). Con Imar, `POST /api/embalaje/:entry` exige una foto por bulto (`imar.exigirFotos`), que se guardan en `<fotosRuta>/<día>/<albarán>-<caja>/`.

Cómo lo haría (sin tocar el embalaje de los demás)

- **Proceso nuevo solo de Imar**, `imar-embalaje-pedido` (`soloTenants: ['imar']`). Se da a los operarios en Equipo en lugar del «Embalaje» común, así que el `packing` de Lafermu y demo no cambia.
- **Lista**: consulta `SGA_IMAR_EMB_PEDIDOS` con los pedidos de venta (`ORDR`, sin filtrar por estado, porque en SAP salen cerrados) que tengan **algún albarán abierto**:

```
SELECT DISTINCT T2."DocEntry", T2."DocNum", T2."CardCode", T2."CardName", T2."DocDueDate"
FROM ODLN T0 INNER JOIN DLN1 T1 ON T1."DocEntry" = T0."DocEntry"
```

```
INNER JOIN ORDR T2 ON T2."DocEntry" = T1."BaseEntry" AND T1."BaseType" = 17
WHERE T0."DocStatus" = 'O' AND T0."CANCELED" = 'N'
```

con filtro por nº de pedido y por cliente, y primero el de entrega más antigua.

- **Pantalla**: la misma de embalaje (`pantallaEmbalaje`), pero con las líneas de todos los albaranes abiertos de ese pedido, cada una marcada con su albarán.
- **Guardar**: reparte los bultos por albarán y llama a `guardarBultos` en cada uno. Si un bulto lleva líneas de dos albaranes del mismo pedido, se guarda con el mismo nº de paquete en los dos (SAP guarda los paquetes por documento).
- **Fotos**: pasarían a ir por pedido (`<día>/P<pedido>-<caja>`), y «Fotos del packing» (`/api/imar/fotos`) buscaría con los dos formatos para no perder las antiguas. La alternativa es guardar la foto en la carpeta de cada albarán del bulto.

> **A confirmar con Imar:** que «albarán relacionado abierto» es el criterio bueno (albarán abierto = todavía sin facturar) y si un bulto puede llevar piezas de dos albaranes del mismo pedido.

7. «Falta configurar la ruta de las fotos del packing (fotosRuta de Imar)»

No es un fallo del código: falta el ajuste. En los ajustes de Imar (consultados solo leyendo) solo está `fotosArticuloRuta = /mnt/imar-fotos`. `fotosRuta` no se ha configurado nunca (tampoco estaba en la copia de antes del paso a producción), y `imar.raizFotos()` da ese mensaje cuando falta.

- Pedir a Imar en qué carpeta de su red quieren las fotos del packing (p. ej. una carpeta `LINCE` en un servidor suyo) y un usuario con permiso de escritura.
- Montarla en el servidor del SGA por el agente de conexión, igual que `\\otservi\sap_imagenes` (servicio SMB del agente, entrada en `fstab`, credenciales en `/etc/samba/`), por ejemplo en `/mnt/imar-packing`. Esta tiene que montarse **con permiso de escritura**.
- Configurar con `node scripts/imar-config.js fotosRuta=/mnt/imar-packing`. Lo hace quien tenga acceso al servidor. Hay que comprobar si la configuración se lee en cada petición; si no, reiniciar `sga-eci`.
- Mientras tanto se podría apuntar a una carpeta local del servidor, pero entonces las fotos quedarían en nuestro servidor y no en el de Imar.

8. Bultos estándar leyendo su etiqueta

Cómo lo haría

- **En SAP** (lo hace Imar, o nosotros con su permiso): dar de alta sus embalajes en *Tipos de embalaje* (tabla `OPKG`): 4 cajas, 2 tubos, 3 cajones… con sus medidas y su peso. Así viven en SAP y no en el SGA.
- **Etiqueta**: cada tipo con un código que no sea solo un número (p. ej. `CAJA-1`, `TUBO-2`, `CAJON-3`), para no confundirlo con el nº de bulto, que hoy se lee como número.
- **Terminal**: en la pantalla de embalaje, si lo leído coincide con un tipo de embalaje (se leen de SAP por el Service Layer y se guardan un rato), se crea el siguiente bulto con ese tipo y se muestran sus medidas y su peso. También un desplegable para elegirlo sin pistola.
- **Guardar**: `guardarBultos` ya manda `Type: b.tipo`. Se añadiría el peso (`TotalWeight`) del tipo, o el que corrija el operario.
- Va detrás de `esImar()` o dentro del proceso nuevo del punto 6.

A confirmar con Imar: la lista de embalajes con medidas y peso, y si el peso del bulto es el de la caja vacía o hay que pesarlo.

9. Ficha técnica: tarda unos 15 segundos

Desde el entorno no se puede medir, porque no llegamos al SAP de Imar. Por el código, la ficha (`GET /api/articulo/:codigo/ficha`, `rutas/api.js`) espera a todo esto antes de contestar:

- El artículo con `ItemWarehouseInfoCollection`, el grupo, los almacenes y el stock por ubicación (`stock.consultar`).
- **La imagen de SAP** (`imagenDe`): si el artículo tiene `Picture` y la carpeta de imágenes de SAP no está montada, la pide al Service Layer (`ItemImages`), que en Imar puede tardar o fallar.
- Las fotos web: **las descarga enteras** solo para quitar las repetidas.
- **Después**, no a la vez que lo demás, `imar.archivosArticulo`: busca plano y foto haciendo `readdir` de la carpeta compartida `/mnt/imar-fotos` por la VPN. El índice se guarda 5 minutos; cuando caduca, la siguiente ficha espera a que se vuelva a leer toda la carpeta.

Cómo lo haría

1. **Medir primero**: dejar unos días en el log el tiempo de cada paso (artículo, stock, imagen, fotos web e Imar) para ver qué se come los 15 s.
2. Meter `imar.archivosArticulo` dentro del `Promise.all` para que vaya a la vez que lo demás (hoy va después).
3. No esperar a la imagen de SAP ni a las fotos web: devolver solo sus direcciones (`/api/articulo/:codigo/imagen` ya existe) y que el terminal las cargue aparte. La ficha sale con los datos y la foto aparece un momento después.
4. Índice de la carpeta de fotos de Imar: renovarlo en segundo plano (se sirve el viejo mientras se lee el nuevo), como ya hace Llevar a carros con sus datos.

A quién afecta: los pasos 1 y 3 están en la ficha común y afectan a todos (para bien). Los pasos 2 y 4 solo cambian lo de Imar. Si se prefiere no tocar lo común, se pueden hacer solo el 2 y el 4.

10. Ficha técnica: el disponible no tiene en cuenta lo solicitado

Sergio tiene razón. En SAP, **Disponible = En stock − Comprometido + Solicitado**. Con sus ejemplos:

Artículo	Stock	Comprometido	Solicitado	SGA hoy	SAP
30.020012093A	317	33	4	284	288
90.007400112A	9	14	4	−5	−1

El SGA calcula `stock − comprometido` en dos sitios de la ficha:

- `publico/terminal/app.js` (~2654): `kpi(r.total.stock - r.total.comprometido, 'disponible')` → sumar `r.total.pedidoProveedor`.
- `rutas/api.js` (~354): `fila.disponible = fila.stock - fila.comprometido` → sumar `fila.pedido`.

Lo mismo pasa en `lib/sap/stock.js` (líneas ~277 y ~310), en la consulta de stock. La pantalla ATP (`stock.atp`) ya lo calcula bien.

> **A quién afecta:** la ficha es **común a todos los clientes**. Es la definición de SAP, así que lo lógico sería corregirlo para todos, pero **no se ha pedido así**: hay que decidir si va para todos o solo para Imar (con `esImar`). Antes de tocar `stock.js` hay que revisar quién usa ese `disponible` (sugerencias,

> avisos de falta de stock), porque ahí puede convenir seguir sin sumar lo solicitado: lo pedido al proveedor todavía no se puede coger de la estantería.

Aparte: el Service Layer que se cae (correo de las 10:44)

No es cosa del SGA. El Service Layer corre en el servidor SAP de Imar (SUSE + HANA), donde no tenemos acceso de root. El 06/11 se arrancó a mano desde el «Service Layer Controller». Lo que hay que pedir a su informática (Aldaro):

- Que los servicios del Service Layer arranquen solos (`systemctl enable`) y se reinicien si fallan (`Restart=on-failure` en el servicio de systemd).
- Que revisen sus logs para ver por qué se para hoy sin corte de luz (memoria, licencia, reinicio de HANA…).
- Por nuestra parte: el SGA ya muestra un mensaje claro cuando da 502 y se recupera solo cuando vuelve. Se podría añadir un aviso por correo a ECINSA cuando el Service Layer de un cliente lleve más de X minutos dando 502.

Cómo se pondría en marcha (cuando se apruebe)

1. Copia previa en `imar/copias/AAAA-MM-DD-correo-sergio-25-11/` de: `lib/imar.js`, `lib/imar-carros.js`, `lib/procesos.js`, `lib/sap/documentos.js`, `lib/staging.js`, `lib/sap/contabiliza.js`, `rutas/imar.js`, `rutas/api.js`, `publico/terminal/app.js` y `scripts/imar-config.js`.
2. Cambios en bloques pequeños y revisables, en este orden: 7 (ajuste) → 10 → 1+2+3 → 9 → 4 → 8 → 6 → 5.
3. `node --check` de cada `.js` tocado.
4. Copiar los ficheros como quedan a `imar/cambios/`, actualizar `imar/actual/` y apuntar la entrada en `imar/CAMBIOS.md`.
5. Reiniciar `sga-eci` (lo hace la persona desde Atalaya o en calculadora: `sudo -u fisio pm2 restart sga-eci`), comprobando antes que no haya nadie trabajando ni transacciones en estado `contabilizando`.
6. Probar en IMAR_PROD: primero solo leyendo (listas, disponibles, ficha) y, con permiso de Imar, una entrega de postventa real de principio a fin.

Cómo volver atrás: restaurar los ficheros de la copia previa y reiniciar `sga-eci`. Los ajustes nuevos de `ajustes.imar` se quitan con `node scripts/imar-config.js clave=`.

Preguntas para Imar

1. ¿El almacén 04 lleva ubicaciones? ¿De cuál se coge primero, del 01 o del 04?
2. ¿Qué lleva el código de barras del pedido de destino? (Foto de una etiqueta.)
3. ¿En qué carpeta de su red van las fotos del packing, y con qué usuario?
4. La lista de sus embalajes estándar con medidas y peso, y si los damos de alta nosotros en SAP.
5. Para imprimir: ¿PDF en el terminal y lo imprimen ellos, o impresión directa en una impresora de red? ¿Qué versión exacta de SAP tienen?
6. Embalaje por pedido: ¿un bulto puede llevar piezas de dos albaranes del mismo pedido?