

Mapeo de Campos - SIESA ERP ↔ Kong WMS/RFID

Índice de Mapeos

1. Productos: SIESA → Kong
 2. Clientes/Terceros: SIESA → Kong
 3. Bodegas: SIESA → Kong
 4. Órdenes de Compra: SIESA → Kong
 5. Órdenes de Venta: SIESA → Kong
 6. Movimientos de Inventario: Kong → SIESA
-

Productos: SIESA → Kong

Entidad SIESA: Items/Productos

API: Consulta de Items

Entidad Kong: SKUs

Endpoint: `POST /inventory/skus/`

Mapeo de Campos

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio
f120_id_item	integer	-	-	No sincronizar	No
f120_referencia	string(50)	external_id	string	Directo	Sí

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio
f120_descripcion	string(200)	name	string	Directo	Sí
f120_descripcion_comercial	string(200)	display_name	string	Si null, usar f120_descripcion	Sí
f120_id_grupo	integer	group_external_id	string	Convertir a string	Sí
f120_codigo_barras	string(50)	ean	string	Directo, si null usar ""	No
f120_unidad_medida	string(10)	properties.unidad_medida	string	En properties	No
f120_ind_estado	integer	is_active	boolean	1 → true, 0 → false	Sí
f120_peso	decimal	properties.peso	decimal	En properties	No
f120_volumen	decimal	properties.volumen	decimal	En properties	No
Todos los campos	-	properties	object	Campos adicionales	No

Función de Transformación

```
javascript
```

```
function transformarProductoSIESAAaKong(productoSIESA) {
    return {
        external_id: productoSIESA.f120_referencia,
        group_external_id: String(productoSIESA.f120_id_grupo),
        name: productoSIESA.f120_descripcion,
        display_name: productoSIESA.f120_descripcion_comercial || productoSIESA.f120_descripcion,
        ean: productoSIESA.f120_codigo_barras || "",
        is_active: productoSIESA.f120_ind_estado === 1,
        properties: {
            unidad_medida: productoSIESA.f120_unidad_medida,
            siesa_id: productoSIESA.f120_id_item,
            peso: productoSIESA.f120_peso,
            volumen: productoSIESA.f120_volumen
        }
    };
}
```

Validaciones

javascript

```
function validarProducto(sku) {  
    const errores = [];  
  
    if (!sku.external_id || sku.external_id.trim() === "") {  
        errores.push("external_id es obligatorio");  
    }  
  
    if (!sku.name || sku.name.trim() === "") {  
        errores.push("name es obligatorio");  
    }  
  
    if (!sku.group_external_id) {  
        errores.push("group_external_id es obligatorio");  
    }  
  
    if (typeof sku.is_active !== "boolean") {  
        errores.push("is_active debe ser boolean");  
    }  
  
    return errores;  
}
```

Clientes/Terceros: SIESA → Kong

Entidad SIESA: Terceros

API: Consulta de Terceros

Entidad Kong: Customers

Endpoint: `POST /customers/customers/`

Mapeo de Campos

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio	Notas
f200_id_tercero	string(20)	external_id	string	Directo	Sí	Código tercero
f200_razon_social	string(200)	name	string	Primer nombre	Sí	Split espa
f200_razon_social	string(200)	last_name	string	Resto del nombre	No	Split espa
f200_email	string(100)	email	string	Directo o generar	Sí	Validar forn
f200_nit	string(20)	identification	string	Directo	Sí	Número iden
f200_tipo_identificacion	string(5)	type_identification	string	Mapear (ver tabla)	Sí	Tipo
f200_telefono	string(20)	properties.telefono	string	En properties	No	Teléfono
f200_direccion	string(200)	properties.direccion	string	En properties	No	Dirección
f200_ind_cliente	integer	properties.es_cliente	boolean	1 → true	No	Es cliente
f200_ind_proveedor	integer	properties.es_proveedor	boolean	1 → true	No	Es proveedor

Mapeo de Tipos de Identificación

SIESA (f200_tipo_identificacion)	Kong (type_identification)
NIT	nit
CC	cc
CE	ce
PAS	pasaporte

SIESA (f200_tipo_identificacion)	Kong (type_identification)
TI	ti
DNI	dni

Función de Transformación

javascript

```
function transformarTerceroSIESAAaKong(terceroSIESA) {
    // Separar nombre completo
    const nombres = terceroSIESA.f200_razon_social.trim().split(/\s+/);
    const primerNombre = nombres[0] || terceroSIESA.f200_id_tercero;
    const apellido = nombres.slice(1).join(' ') || "";

    // Generar email si no existe
    const email = terceroSIESA.f200_email ||
        `${terceroSIESA.f200_id_tercero}@temp.local`;

    return {
        external_id: terceroSIESA.f200_id_tercero,
        name: primerNombre,
        last_name: apellido,
        email: email,
        identification: terceroSIESA.f200_nit,
        type_identification: mapearTipoIdentificacion(terceroSIESA.f200_tipo_identificacion),
        properties: {
            razon_social_completa: terceroSIESA.f200_razon_social,
            telefono: terceroSIESA.f200_telefono,
            direccion: terceroSIESA.f200_direccion,
            es_cliente: terceroSIESA.f200_ind_cliente === 1,
            es_proveedor: terceroSIESA.f200_ind_proveedor === 1,
            siesa_id: terceroSIESA.f200_id_tercero
        }
    };
}

function mapearTipoIdentificacion(tipoSIESA) {
    const mapeo = {
        'NIT': 'nit',
        'CC': 'cc',
        'CE': 'ce',
    }
}
```

```

'PAS': 'pasaporte',
'TI': 'ti',
'DNI': 'dni'
};

return mapeo[tipoSIESA.toUpperCase()] || 'cc';
}

```

Bodegas: SIESA → Kong

Entidad SIESA: Bodegas

API: Consulta de Bodegas

Entidad Kong: Locations

Endpoint: `POST /inventory/locations/`

Mapeo de Campos

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio	Notas
f110_id_bodega	string(10)	external_id	string	Directo	Sí	Código bodega
f110_descripcion	string(100)	name	string	Directo	Sí	Nombre bodega
f110_id_co	integer	properties.company_id	integer	En properties	No	Centro operación
f110_ind_activo	integer	is_active	boolean	1 → true	Sí	Estado

Función de Transformación

javascript

```
function transformarBodegaSIESAAKong(bodegaSIESA) {  
    return {  
        external_id: bodegaSIESA.f110_id_bodega,  
        name: bodegaSIESA.f110_descripcion,  
        location_type: 1, // Warehouse type (predefinido en Kong)  
        is_active: bodegaSIESA.f110_ind_activo === 1,  
        properties: {  
            siesa_id_co: bodegaSIESA.f110_id_co,  
            siesa_bodega_id: bodegaSIESA.f110_id_bodega  
        }  
    };  
}
```

Órdenes de Compra: SIESA → Kong

Entidad SIESA: Órdenes de Compra

API: Consulta de Órdenes de Compra

Entidad Kong: Purchase Orders

Endpoint: [POST /operations/purchase-orders/](#)

Mapeo de Campos - Cabecera

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio	Notas
f430_consec_docto	string(20)	external_id	string	Directo	Sí	Número d OC

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio	Notas
f430_id_tercero	string(20)	requester_external_id	string	Directo	Sí	Proveedor
f430_id_bodega	string(10)	destination_external_id	string	Directo	Sí	Bodega destino
f430_fecha_entrega	date	expected_date	datetime	ISO 8601	No	Fecha esperada
f430_notas	string	properties.notas	string	En properties	No	Observaci

Mapeo de Campos - Líneas

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio	Notas
f431_id_item	string(50)	sku_external_id	string	Directo	Sí	Código producto
f431_cantidad	decimal	quantity	integer	Redondear	Sí	Cantidad esperada
f431_nro_registro	integer	line_number	integer	Directo	No	Número de línea

Función de Transformación

javascript

```
function transformarOCSIESAAaKong(ocSIESA, lineasSIESA) {
    return {
        external_id: ocSIESA.f430_consec_docto,
        requester_external_id: ocSIESA.f430_id_tercero,
        destination_external_id: ocSIESA.f430_id_bodega,
        expected_date: ocSIESA.f430_fecha_entrega ?
            new Date(ocSIESA.f430_fecha_entrega).toISOString() : null,
        lines: lineasSIESA.map(linea => ({
            sku_external_id: linea.f431_id_item,
            quantity: Math.round(linea.f431_cantidad),
            line_number: linea.f431_nro_registro
        })),
        properties: {
            siesa_tipo_docto: ocSIESA.f430_id_tipo_docto,
            siesa_notas: ocSIESA.f430_notas
        }
    };
}
```

Órdenes de Venta: SIESA → Kong

Entidad SIESA: Remisiones/Órdenes de Venta

API: Consulta de Remisiones

Entidad Kong: Store Orders

Endpoint: `POST /operations/store-orders/`

Mapeo de Campos - Cabecera

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio	Notas
f350_consec_docto	string(20)	external_id	string	Directo	Sí	Número remisión
f450_id_bodega_salida	string(10)	source_external_id	string	Directo	Sí	Bodega origen
f350_id_tercero	string(20)	destination_external_id	string	Directo	Sí	Cliente destino
f350_id_tercero	string(20)	requester_external_id	string	Directo	Sí	Cliente solicita
f350_fecha	date	expected_date	datetime	ISO 8601	No	Fecha entrega
f350_fecha	date	arrival_date	datetime	ISO 8601	No	Fecha llegada

Mapeo de Campos - Líneas

Campo SIESA	Tipo	Campo Kong	Tipo	Transformación	Obligatorio	Notas
f470_id_item	string(50)	sku_external_id	string	Directo (referencia)	Sí	Código producto
f470_cant_base	decimal	quantity	integer	Redondear	Sí	Cantidad a despachar

Función de Transformación

javascript

```
function transformarRemisionSIESAaKong(remisionSIESA, lineasSIESA) {
    return {
        external_id: remisionSIESA.f350_consec_docto,
        source_external_id: remisionSIESA.f450_id_bodega_salida,
        destination_external_id: remisionSIESA.f350_id_tercero,
        requester_external_id: remisionSIESA.f350_id_tercero,
        expected_date: remisionSIESA.f350_fecha ?
            new Date(remisionSIESA.f350_fecha).toISOString() : null,
        arrival_date: null,
        lines: lineasSIESA.map(linea => ({
            sku_external_id: linea.f470_id_item,
            quantity: Math.round(parseFloat(linea.f470_cant_base))
        })),
        properties: {
            siesa_tipo_docto: remisionSIESA.f350_id_tipo_docto,
            siesa_consecutivo: remisionSIESA.f350_consec_docto,
            siesa_notas: remisionSIESA.f350_notas
        }
    };
}
```

Movimientos de Inventario: Kong → SIESA

Entidad Kong: Moves, Packings, Audits

Endpoints: </inventory/moves/>, </operations/packings/>, </inventory/audits/>

Entidad SIESA: Documento de Inventario

API: Conector Documentolv (POST)

Mapeo Genérico - Cabecera del Documento

Campo Kong	Tipo	Campo SIESA	Tipo	Transformación	Obligatorio	Notas
-	-	F_CIA	string	Hardcode "1"	Sí	Código compañía
-	-	f350_id_co	string	Hardcode "1"	Sí	Centro operación
move_type	string	f350_id_tipo_doco	string	Ver tabla tipos	Sí	ENT/SAL/AJU/TRA
-	-	f350_consec_doco	string	Hardcode "AUTO"	Sí	Consecutivo auto
closed_at	datetime	f350_fecha	date	Formato YYYY-MM-DD	Sí	Fecha movimiento
reference	string	f350_notas	string	Incluir en notas	No	Referencia externa
-	-	f350_ind_estado	string	Hardcode "2"	Sí	Confirmado
move.id	integer	f450_doco_alterno	string	"KONG-XXX-{id}"	No	Referencia Kong

Mapeo por Tipo de Movimiento

Entrada (RECEIVING)

Campo Kong	Campo SIESA	Valor
move_type: "RECEIVING"	f350_id_tipo_doco	"ENT"
-	f450_id_concepto	"1"
destination_location.external_id	f450_id_bodega_entrada	Directo
-	f450_id_bodega_salida	""

Salida (SHIPPING)

Campo Kong	Campo SIESA	Valor
move_type: "SHIPPING"	f350_id_tipo_doco	"SAL"
-	f450_id_concepto	"2"
source_location.external_id	f450_id_bodega_salida	Directo
-	f450_id_bodega_entrada	""

Ajuste (ADJUSTMENT)

Campo Kong	Campo SIESA	Valor
difference > 0	f350_id_tipo_doco	"AJU"
-	f450_id_concepto	"3" (positivo) o "4" (negativo)
location.external_id	f450_id_bodega_entrada	Si positivo
location.external_id	f450_id_bodega_salida	Si negativo

Transferencia (TRANSFER)

Campo Kong	Campo SIESA	Valor
move_type: "TRANSFER"	f350_id_tipo_doco	"TRA"
-	f450_id_concepto	"5"
source_location.external_id	f450_id_bodega_salida	Directo
destination_location.external_id	f450_id_bodega_entrada	Directo

Mapeo - Líneas de Movimiento

Campo Kong	Tipo	Campo SIESA	Tipo	Transformación	Obligatorio	Notas
-	-	F_CIA	string	Hardcode "1"	Sí	Código compañía

Campo Kong	Tipo	Campo SIESA	Tipo	Transformación	Obligatorio	Notas
sku.external_id	string	f470_id_item	string	Directo	Sí	Código producto
location.external_id	string	f470_id_bodega	string	Directo	Sí	Bodega
-	-	f470_id_unidad_medida	string	Hardcode "UN"	Sí	Unidad medida
quantity	integer	f470_cant_base	string	Convertir a string	Sí	Cantidad
(índice)	-	f470_nro_registro	string	Índice + 1	Sí	Número línea
notes	string	f470_notas	string	Directo	No	Observaciones

Funciones de Transformación

Transformar Move a Documento SIESA

```

javascrip

```

```

function transformarMoveKongASIESA(move, lines) {
    // Determinar tipo de documento
    const tipoMap = {
        'RECEIVING': { tipo: 'ENT', concepto: '1', usaBodegaEntrada: true },
        'SHIPPING': { tipo: 'SAL', concepto: '2', usaBodegaSalida: true },
        'TRANSFER': { tipo: 'TRA', concepto: '5', usaAmbas: true }
    };

    const config = tipoMap[move.move_type];

    const documento = {
        Inicial: [{ F_CIA: "1" }],
        Documentos: [
            {
                F_CIA: "1",
                f350_id_co: "1",
                f350_id_tipo_docto: config.tipo,
                f350_consec_docto: "AUTO",
                f350_fecha: new Date(move.closed_at).toISOString().split('T')[0],
                f350_ind_estado: "2",
                f350_notas: `Kong Move ${move.id} - ${move.reference} `,
                f450_id_concepto: config.concepto,
                f450_docto_alterno: `KONG-MOVE-${move.id}`
            }],
        Movimientos: [],
        Final: [{ F_CIA: "1" }]
    };

    // Configurar bodegas según tipo
    if(config.usaBodegaEntrada) {
        documento.Documentos[0].f450_id_bodega_entrada = move.destination_location.external_id;
        documento.Documentos[0].f450_id_bodega_salida = "";
    } else if(config.usaBodegaSalida) {
        documento.Documentos[0].f450_id_bodega_entrada = "";

```

```

documento.Documentos[0].f450_id_bodega_salida = move.source_location.external_id;
} else if (config.usaAmbas) {
    documento.Documentos[0].f450_id_bodega_salida = move.source_location.external_id;
    documento.Documentos[0].f450_id_bodega_entrada = move.destination_location.external_id;
}

// Agregar líneas
documento.Movimientos = lines.map((line, index) => ({
    F_CIA: "1",
    f470_id_item: line.sku.external_id,
    f470_id_bodega: config.usaBodegaSalida ?
        move.source_location.external_id :
        move.destination_location.external_id,
    f470_id_unidad_medida: "UN",
    f470_cant_base: String(line.quantity_received || line.quantity),
    f470_nro_registro: String(index + 1),
    f470_notas: `SKU: ${line.sku.name} - Cantidad: ${line.quantity_received || line.quantity}`
}));
```

return documento;

}

Transformar Ajuste a Documento SIESA

javascript

```
function transformarAjusteKongASIESA(auditLine, saldoSIESA) {
    const diferencia = auditLine.physical_quantity - saldoSIESA.saldo_cantidad;
    const esPositivo = diferencia > 0;

    return {
        Inicial: [{ F_CIA: "1" }],
        Documentos: [
            F_CIA: "1",
            f350_id_co: "1",
            f350_id_tipo_docto: "AJU",
            f350_consec_docto: "AUTO",
            f350_fecha: new Date().toISOString().split('T')[0],
            f350_ind_estado: "2",
            f350_notas: `Ajuste Auditoría Kong ${auditLine.audit} - Físico: ${auditLine.physical_quantity}, Contable: ${saldoSIESA.saldo_cantidad}`,
            f450_id_concepto: esPositivo ? "3" : "4",
            f450_id_bodega_entrada: esPositivo ? auditLine.location.external_id : "",
            f450_id_bodega_salida: esPositivo ? "" : auditLine.location.external_id,
            f450_docto_alterno: `KONG-ADJ-${auditLine.audit}-${auditLine.sku.external_id}`
        ],
        Movimientos: [
            F_CIA: "1",
            f470_id_item: auditLine.sku.external_id,
            f470_id_bodega: auditLine.location.external_id,
            f470_id_unidad_medida: "UN",
            f470_cant_base: String(Math.abs(diferencia)),
            f470_nro_registro: "1",
            f470_notas: `Conteo RFID - ${esPositivo ? 'Sobrante' : 'Faltante'} detectado`
        ],
        Final: [{ F_CIA: "1" }]
    };
}
```

Reglas de Conversión de Datos

Fechas

```
javascript

// SIESA espera formato: YYYY-MM-DD
function formatearFechaParaSIESA(fecha) {
    if (!fecha) return null;
    return new Date(fecha).toISOString().split('T')[0];
}

// Kong espera formato ISO 8601
function formatearFechaParaKong(fecha) {
    if (!fecha) return null;
    return new Date(fecha).toISOString();
}
```

Cantidades

```
javascript

// SIESA requiere cantidades como string
function formatearCantidadParaSIESA(cantidad) {
    return String(cantidad);
}

// Kong usa integers
function formatearCantidadParaKong(cantidad) {
    return Math.round(parseFloat(cantidad));
}
```

Booleanos

```
javascript

// SIESA usa integers (1/0)
function booleanASIESA(valor) {
    return valor ? 1 : 0;
}

// SIESA a Kong
function siesaABoolean(valor) {
    return valor === 1 || valor === "1";
}
```

Validaciones de Integridad

Validar Documento SIESA antes de enviar

```
javascript
```

```
function validarDocumentoSIESA(documento) {
    const errores = [];

    // Validar estructura
    if (!documento.Inicial || !documento.Documentos ||
        !documento.Movimientos || !documento.Final) {
        errores.push("Estructura incompleta");
        return { valido: false, errores };
    }

    const doc = documento.Documentos[0];
    const movs = documento.Movimientos;

    // Validar campos obligatorios cabecera
    const obligatoriosCabecera = [
        'F_CIA', 'f350_id_co', 'f350_id_tipo_docto',
        'f350_consec_docto', 'f350_fecha', 'f350_ind_estado',
        'f450_id_concepto'
    ];

    obligatoriosCabecera.forEach(campo => {
        if (!doc[campo]) {
            errores.push(`Campo obligatorio faltante: ${campo}`);
        }
    });
}

// Validar bodegas según tipo
if (doc.f350_id_tipo_docto === 'ENT' && !doc.f450_id_bodega_entrada) {
    errores.push("Entrada requiere bodega de entrada");
}
if (doc.f350_id_tipo_docto === 'SAL' && !doc.f450_id_bodega_salida) {
    errores.push("Salida requiere bodega de salida");
}
```

```
if(doc.f350_id_tipo_doco === 'TRA' &&
  (!doc.f450_id_bodega_entrada || !doc.f450_id_bodega_salida)) {
  errores.push("Transferencia requiere ambas bodegas");
}

// Validar movimientos
if(movs.length === 0) {
  errores.push("Debe haber al menos un movimiento");
}

movs.forEach((mov, idx) => {
  if(!mov.f470_id_item) {
    errores.push(`Línea ${idx + 1}: Item es obligatorio`);
  }
  if(!mov.f470_cant_base || parseFloat(mov.f470_cant_base) <= 0) {
    errores.push(`Línea ${idx + 1}: Cantidad inválida`);
  }
  if(!mov.f470_id_unidad_medida) {
    errores.push(`Línea ${idx + 1}: Unidad de medida obligatoria`);
  }
});

return {
  valido: errores.length === 0,
  errores
};
}
```

Tabla de Referencia Rápida

Tipos de Documento SIESA

Código	Descripción	Campos Requeridos
ENT	Entrada	f450_id_bodega_entrada
SAL	Salida	f450_id_bodega_salida
AJU	Ajuste	f450_id_bodega_entrada (+) o f450_id_bodega_salida (-)
TRA	Transferencia	f450_id_bodega_salida + f450_id_bodega_entrada

Conceptos Comunes

Código	Descripción	Tipo Documento
1	Entrada directa	ENT
2	Salida directa	SAL
3	Ajuste positivo	AJU
4	Ajuste negativo	AJU
5	Transferencia	TRA

Nota: Los códigos de concepto pueden variar según configuración de cada implementación SIESA.