

PRD — Módulo Inventario

Propósito: Este documento define, con detalle suficiente para desarrollo, QA y despliegue, los requisitos funcionales y no funcionales de una herramienta de Inventario orientada a **compras a proveedores** y **operación de inventario/almacén**, con capacidades comparables a un sistema empresarial moderno, **sin** incorporar contabilidad/finanzas, emisión tributaria ni gestión de proyectos.

1. Alcance y visión

1.1. Visión del producto

Construir una herramienta única para gestionar el ciclo completo de abastecimiento y control de inventario: desde la necesidad interna hasta la recepción física y disponibilidad de stock, incluyendo trazabilidad, auditoría y reposición operativa.

1.2. Objetivos

1. **Operar sin planillas** el flujo end-to-end: necesidad → solicitud → cotización → orden → recepción → ubicación → stock disponible.
2. Mantener **exactitud de inventario** mediante un libro mayor (ledger) auditable de movimientos.
3. Soportar operación **multi-bodega** con ubicaciones (bins) y trazabilidad por lote/serie cuando aplique.
4. Reducir quiebres y sobrestock con **reglas de reposición** y sugerencias accionables.
5. Proveer reportes operativos y KPIs para compras y bodega.

1.3. No objetivos (exclusiones explícitas)

- No contabilidad: sin asientos, cuentas por pagar/cobrar, pagos, conciliaciones.
- No emisión de documentos tributarios.
- No gestión de proyectos (WBS, órdenes de trabajo de proyecto, control de avance).
- No planificación avanzada de producción (MRP completo) en MVP; se permitirá reposición por reglas.

1.4. Definiciones

- **PR (Purchase Requisition / Solicitud de Compra):** documento interno que formaliza una necesidad.
 - **RFQ (Request for Quotation / Solicitud de Cotización):** consulta a proveedores para obtener ofertas.
 - **PO (Purchase Order / Orden de Compra):** orden formal hacia un proveedor.
 - **GRN (Goods Receipt Note / Recepción):** documento de recepción física contra PO.
 - **Ledger de inventario:** registro inmutable de movimientos que determina stock.
 - **Stock on hand:** stock físico registrado.
 - **Stock disponible:** stock liberado y no reservado.
-

2. Usuarios, roles y permisos

2.1. Personas

1. **Operador de bodega:** recibe, ubica, mueve, cuenta.
2. **Comprador:** gestiona PR, RFQ, PO, seguimiento y devoluciones.
3. **Aprobador/Jefatura:** aprueba PR/PO, variaciones, ajustes y cierres.
4. **Planificador de reposición:** define parámetros y ejecuta sugerencias de compra.
5. **Administrador:** configura maestros, permisos, integraciones.

2.2. Modelo de permisos (RBAC)

- Permisos por **acción** (crear, editar, aprobar, anular, postear), por **entidad** (PR/RFQ/PO/GRN/ movimientos), por **alcance** (empresa, bodega, ubicación, categoría de SKU).
- Soporte para **aprobaciones por umbrales** (monto, categoría, criticidad).
- Restricciones de **segregación**: quien recibe no necesariamente ajusta; quien compra no necesariamente libera cuarentena.

2.3. Auditoría

Toda operación que afecte documentos o stock debe registrar:

- Usuario, rol, timestamp, IP/opcional, acción, entidad afectada, valores antes/después, motivo y adjuntos si aplica.

3. Requisitos no funcionales (NFR)

3.1. Seguridad

- Cifrado en tránsito (TLS) y reposo.
- RBAC y mínimo privilegio.
- Registro de auditoría inmutable.
- MFA/SSO opcional.

3.2. Rendimiento y escalabilidad

- Búsqueda y navegación de documentos < 1s en escenarios típicos (hasta 50k SKUs, 100k movimientos/mes).
- Inserción de movimientos de inventario con latencia baja (append-only).
- Generación de reportes por agregación incremental o snapshots.

3.3. Disponibilidad y resiliencia

- Tolerancia a fallos con colas/eventos internos.
- Idempotencia en operaciones críticas (postear GRN, postear ajustes, integraciones).

3.4. Usabilidad

- Flujos “warehouse-first”: minimizar clics en recepciones, transferencias y conteos.
- Mobile-first o modo escáner (fase posterior, pero arquitectura preparada).

4. Arquitectura funcional del dominio

4.1. Principio rector: Inventario basado en ledger

- El stock **no** se edita directamente.
- El stock se calcula como suma de movimientos válidos (ledger) por (empresa, bodega, ubicación, SKU, lote/serie, estado).
- Se permiten **snapshots** para performance, derivados del ledger.

4.2. Cadena documental (trazabilidad)

- PR → RFQ → Quote(s) → PO → GRN → Putaway/Movimientos → Stock.
 - Cada documento mantiene referencias a su origen y destino.
-

5. Modelo de datos (entidades, relaciones y reglas)

Esta sección es suficiente para derivar un modelo relacional o event-sourced. Se indican claves, relaciones y restricciones.

5.1. Entidades base

5.1.1. Organización

- **Company**(company_id, name, timezone, currency_default, status)
- Reglas: multi-empresa opcional; todo objeto pertenece a company_id.

5.1.2. Almacenes

- **Warehouse**(warehouse_id, company_id, code, name, status)
- **Zone**(zone_id, warehouse_id, code, name, type)
- **Location/Bin**(location_id, zone_id, code, barcode, type, capacity, status)
- Reglas:
- Código único por warehouse.
- Location.type: receiving, storage, picking, quarantine, returns, staging.

5.1.3. Unidades de medida

- **UoM**(uom_id, code, name)
- **UoMConversion**(from_uom_id, to_uom_id, factor)
- Reglas: factor > 0; conversión definida por SKU o global.

5.1.4. Productos

- **Item/SKU**(item_id, company_id, sku_code, name, type, category_id, base_uom_id, tracking_mode, shelf_life_days, status)
- type: stock, no_stock, service
- tracking_mode: none, lot, serial
- **ItemCategory**(category_id, company_id, code, name, parent_id)
- **ItemAttribute**(attribute_id, name, data_type)

- **ItemAttributeValue**(item_id, attribute_id, value)
- Reglas:
- sku_code único por company.
- Si tracking_mode=serial, qty por serial siempre 1.

5.1.5. Proveedores

- **Supplier**(supplier_id, company_id, legal_name, trade_name, tax_id_optional, status, default_currency, default_lead_time_days, rating)
- **SupplierContact**(contact_id, supplier_id, name, email, phone, role)
- **SupplierItem**(supplier_item_id, supplier_id, item_id, supplier_sku_code, MOQ, multiple_qty, lead_time_days, price, currency, incoterm_optional, status)
- Reglas:
- Un supplier puede tener múltiples registros por item para diferentes condiciones (con vigencias vNext).

5.2. Compras

5.2.1. Solicitud de compra (PR)

- **PurchaseRequisition**(pr_id, company_id, requester_user_id, warehouse_id, needed_date, status, justification, created_at)
- **PRLine**(pr_line_id, pr_id, item_id, requested_qty, uom_id, required_date, notes, status)
- **ApprovalFlow**(flow_id, company_id, object_type, rule_json)
- **ApprovalInstance**(approval_id, object_type, object_id, step, approver_user_id, decision, decision_at, comment)
- Reglas:
- status: draft → submitted → approved/rejected → sourced → closed
- No se puede crear RFQ/PO desde PR no aprobada (salvo configuración).

5.2.2. RFQ y cotizaciones

- **RFQ**(rfq_id, company_id, source_pr_id_optional, buyer_user_id, status, due_date, notes)
- **RFQLine**(rfq_line_id, rfq_id, item_id, qty, uom_id, target_date)
- **RFQSupplier**(rfq_supplier_id, rfq_id, supplier_id, status, sent_at)
- **Quote**(quote_id, rfq_supplier_id, currency, validity_date, lead_time_days, total_estimated, status)
- **QuoteLine**(quote_line_id, quote_id, rfq_line_id, unit_price, qty_offered, notes)
- Reglas:
- status RFQ: draft → sent → closed
- Quote: received → evaluated → awarded/declined

5.2.3. Orden de compra (PO)

- **PurchaseOrder**(po_id, company_id, supplier_id, buyer_user_id, warehouse_id, status, currency, promised_date_optional, version, created_at)
- **POLine**(po_line_id, po_id, item_id, qty_ordered, uom_id, unit_price, qty_received, status, notes)
- **POVersionLog**(po_version_id, po_id, version, changed_by, changed_at, diff_json, reason)
- Reglas:
- status: draft → sent → confirmed → partially_received → received → closed/cancelled
- Cambios que afecten qty/price/fechas requieren control de versión y permisos.
- qty_received <= qty_ordered (salvo tolerancias configurables con flags y auditoría).

5.3. Recepción y WMS

5.3.1. Recepción (GRN)

- **GoodsReceipt**(grn_id, company_id, po_id, warehouse_id, received_by_user_id, received_at, status, notes)
- **GRNLine**(grn_line_id, grn_id, po_line_id, item_id, qty_received, uom_id, discrepancy_type, discrepancy_notes)
- **GRNTracking**(grn_tracking_id, grn_line_id, lot_code_optional, serial_code_optional, expiry_date_optional, qty)
- Reglas:
- status: draft → posted (impacta stock) → cancelled (reversa movimientos)
- tracking_mode de SKU obliga registro de lot/serial según configuración.

5.3.2. Putaway

- **PutawayTask**(task_id, grn_id, warehouse_id, status, created_at)
- **PutawayLine**(task_line_id, task_id, item_id, qty, from_location_id, to_location_id, tracking_ref_optional)
- Reglas:
- MVP: asignación manual con sugerencia por ubicación preferida.
- vNext: reglas de almacenamiento por categoría/capacidad/rotación.

5.3.3. Movimientos e inventario

- **InventoryLedgerEntry**(ile_id, company_id, timestamp, source_type, source_id, warehouse_id, from_location_id_optional, to_location_id_optional, item_id, tracking_ref_optional, qty_delta, inventory_status, reason_code, posted_by_user_id)
- qty_delta positivo para entradas, negativo para salidas.
- inventory_status: available, quarantine, damaged, reserved
- **TrackingRef** puede ser:
- lote: (item_id, lot_code)
- serie: (item_id, serial_code)
- Reglas:
- Stock = SUM(qty_delta) agrupado por dimensiones.
- No permitir saldo negativo si regla de no-negative está activa (configurable por bodega).

5.3.4. Ajustes

- **InventoryAdjustment**(adj_id, company_id, warehouse_id, status, requested_by, approved_by_optional, reason, created_at)
- **AdjustmentLine**(adj_line_id, adj_id, location_id, item_id, tracking_ref_optional, qty_new, qty_old_calculated, variance, notes)
- Reglas:
- Aprobación requerida sobre umbral de varianza/monto operativo.
- Postear ajuste genera ILE con motivo.

5.4. Inventarios físicos (conteos)

- **CycleCountPlan**(plan_id, company_id, warehouse_id, method, schedule_rule, status)
- method: ABC, random, location-based
- **CycleCountTask**(count_id, plan_id_optional, warehouse_id, status, assigned_to, created_at)

- **CountLine**(count_line_id, count_id, location_id, item_id, tracking_ref_optional, expected_qty, counted_qty, variance, status)
- Reglas:
- Si $\text{abs}(\text{variance}) > \text{tolerancia}$: requiere recount.
- Cierre del conteo puede generar ajuste (con aprobación).

5.5. Reposición

- **ReplenishmentPolicy**(policy_id, company_id, item_id, warehouse_id, min_qty, max_qty, safety_stock, lead_time_days, review_period_days, status)
- **ReplenishmentSuggestion**(sugg_id, company_id, warehouse_id, item_id, suggested_qty, rationale_json, status, created_at)
- Reglas:
- $\text{suggested_qty} = \text{max_qty} - (\text{on_hand} + \text{in_transit} - \text{reserved})$ cuando $\text{on_hand} \leq \text{min_qty}$.
- in_transit incluye líneas de PO abiertas.

5.6. Calidad y devoluciones

- **QualityHold**(qh_id, company_id, grn_line_id, status, reason, created_at, released_by_optional)
- **ReturnToVendor**(rtv_id, company_id, supplier_id, warehouse_id, status, created_at)
- **RTVLine**(rtv_line_id, rtv_id, item_id, qty, tracking_ref_optional, reason)
- Reglas:
- Cuarentena bloquea disponibilidad (inventory_status=quarantine).
- RTV postea movimiento de salida.

5.7. Adjuntos, comentarios y notificaciones

- **Attachment**(attachment_id, object_type, object_id, file_meta, uploaded_by, uploaded_at)
- **Comment**(comment_id, object_type, object_id, user_id, text, created_at)
- **Notification**(notif_id, user_id, type, payload_json, read_at_optional)

6. Flujos de negocio (end-to-end) con estados y validaciones

6.1. Flujo A — Necesidad → PR → Aprobación

1. Usuario crea PR (draft) con líneas SKU.
2. Valida: SKU activo, UoM válida, qty > 0, bodega definida.
3. Usuario envía PR (submitted).
4. Motor de aprobaciones crea ApprovalInstance según reglas.
5. Aprobadores deciden: approve/reject.
6. Si approved: PR pasa a approved.

Reglas

- No permitir edición de líneas en estados submitted/approved, salvo mediante “solicitud de cambio” (MVP: permitir edición solo si vuelve a draft).

6.2. Flujo B — PR → RFQ → Cotizaciones → Adjudicación

1. Comprador crea RFQ desde PR approved.
2. Selecciona proveedores (mín. 1) y fecha límite.

3. Envía RFQ (email/link) y registra estado por proveedor.
4. Registra cotizaciones por proveedor.
5. Comparador muestra ranking por: precio, lead time, cumplimiento histórico, condiciones.
6. Comprador adjudica y convierte a PO.

Validaciones

- Moneda consistente por quote; conversión opcional.
- Cotización expirada no puede adjudicarse sin override.

6.3. Flujo C — PO → Recepción → Putaway → Stock disponible

1. PO en sent/confirmed.
2. Operador genera GRN contra PO.
3. Recibe por línea: qty recibida y tracking (lote/serie si aplica).
4. Sistema registra discrepancias.
5. Al postear GRN:
6. Actualiza qty_received de PO line.
7. Crea InventoryLedgerEntry de entrada a ubicación de receiving.
8. Si política de calidad activa: estado quarantine; si no, available.
9. Genera PutawayTask.
10. Putaway mueve stock a ubicación destino creando ILE de transferencia.

Reglas

- GRN posted es idempotente (no duplicar movimientos si se reintenta).
- No permitir putaway de qty mayor a recibido.

6.4. Flujo D — Transferencias y ajustes

- Transferencia: requiere origen/destino, qty, tracking.
- Ajuste: requiere motivo, aprobación si umbral.
- Todo genera ILE.

6.5. Flujo E — Conteo cíclico

1. Sistema genera tareas según plan o manual.
2. Operador cuenta y registra.
3. Si varianza alta → recount.
4. Aprobación → postear ajuste.

6.6. Flujo F — Reposición

1. Motor calcula sugerencias diariamente o bajo demanda.
2. Planificador revisa tablero de sugerencias.
3. Aprueba y genera PR o PO directa (según permiso/política).

7. Requisitos funcionales detallados (FR)

Cada requisito incluye: descripción, entradas, salidas, reglas y errores.

7.1. Maestros

FR-MD-01 — Gestión de SKU

- Entradas: sku_code, name, category, base_uom, tracking_mode, atributos.
- Salidas: SKU persistido, audit log.
- Reglas: sku_code único; tracking_mode determina campos obligatorios en recepciones.
- Errores: duplicado, UoM inválida, categoría inexistente.

FR-MD-02 — Estructura de almacén

- Crear bodegas, zonas y ubicaciones; asignar tipos y códigos.
- Reglas: códigos únicos por nivel; ubicación receiving obligatoria por bodega.

FR-MD-03 — Proveedores y catálogo

- CRUD de proveedor; asociación SKU-proveedor con MOQ/múltiplos/lead time/precio.
- Reglas: MOQ ≥ 0 ; múltiplos > 0 .

7.2. Solicitudes de compra (PR)

FR-PR-01 — Crear PR

- Permite múltiples líneas.
- Adjuntos y comentarios.

FR-PR-02 — Aprobación

- Reglas configurables (JSON) por monto/categoría.
- Estados y trazabilidad.

7.3. RFQ

FR-RFQ-01 — Crear/Enviar RFQ

- Selección de proveedores.
- Plantilla de email; registro de envío y respuesta.

FR-RFQ-02 — Registrar cotizaciones y comparar

- Comparador con ranking.
- Permite adjudicar parcial por línea (vNext: split awards).

7.4. Órdenes de compra (PO)

FR-PO-01 — Crear/Editar PO

- Desde adjudicación o manual.
- Versionado cuando cambia qty/price/fechas.

FR-PO-02 — Seguimiento

- Alertas por atraso.

- Confirmación de fecha prometida.

7.5. Recepción

FR-REC-01 — Crear GRN desde PO

- Recepción parcial.
- Discrepancias.

FR-REC-02 — Postear GRN

- Genera ILE entradas.
- Actualiza PO.
- Si calidad activa: quarantine.

7.6. Putaway

FR-WMS-01 — Ejecutar putaway

- Desde receiving a storage.
- Genera ILE transferencia.

7.7. Movimientos y ajustes

FR-INV-01 — Transferencia

- Validación de saldo disponible.

FR-INV-02 — Ajuste

- Requiere motivo; aprobación por umbral.
- Genera ILE.

7.8. Conteos

FR-COUNT-01 — Crear tarea de conteo

- Manual o por plan.

FR-COUNT-02 — Cerrar conteo

- Recount y aprobación.
- Ajuste generado opcional.

7.9. Reposición

FR-PLAN-01 — Configurar políticas

- min/max/safety/lead time.

FR-PLAN-02 — Generar sugerencias

- Incluye in_transit y reserved.

7.10. Calidad y devoluciones

FR-QA-01 — Cuarentena y liberación

- Bloquea disponibilidad.

FR-RTV-01 — Devolución a proveedor

- Crea documento RTV y movimientos.

7.11. Reportes

FR-RPT-01 — Stock

- On hand por dimensiones.

FR-RPT-02 — Cobertura/quiebre

- Días de cobertura = $\text{on_hand} / \text{consumo_promedio}$ (MVP: consumo manual o histórico simple si hay salidas).

FR-RPT-03 — Proveedores

- OTD, lead time real.

7.12. Integraciones

FR-INT-01 — API

- CRUD de maestros.
- Lectura de documentos.
- Posteo de eventos.

FR-INT-02 — Export

- Exportar PO/GRN/RTV/Ajustes.

8. Reglas transversales

8.1. Estados y transiciones (tabla conceptual)

- PR: draft → submitted → approved/rejected → sourced → closed
- RFQ: draft → sent → closed
- Quote: received → evaluated → awarded/declined
- PO: draft → sent → confirmed → partially_received → received → closed/cancelled
- GRN: draft → posted → cancelled
- PutawayTask: open → in_progress → done/cancelled
- CountTask: open → counting → recount_needed → approved → closed

8.2. Idempotencia

Operaciones “postear” deben incluir:

- idempotency_key por request
- verificación de duplicados por (source_type, source_id, hash)

8.3. Control de saldo negativo

Configurable por bodega:

- strict_no_negative: bloquea
- allow_negative: permite con auditoría

8.4. Costos operativos (sin contabilidad)

- Guardar unit_price en PO.
 - Calcular costo promedio operativo opcional.
 - No generar valorización contable.
-

9. Interfaz de usuario (IU) y pantallas mínimas

9.1. Navegación principal

- Dashboard
- Compras: PR, RFQ, PO
- Bodega: Recepciones, Putaway, Transferencias, Ajustes
- Conteos
- Reposición
- Maestros
- Reportes
- Admin/Integraciones

9.2. Pantallas clave

1. **Listado PR** + filtros + estados + acciones.
 2. **Detalle PR**: líneas, historial, aprobaciones, adjuntos.
 3. **RFQ**: proveedores, envío, cotizaciones y comparador.
 4. **PO**: detalle, versión, estado, recepciones asociadas.
 5. **Recepción (GRN)**: captura rápida, tracking, discrepancias.
 6. **Putaway**: tareas, origen/destino, confirmación.
 7. **Kardex**: movimientos por SKU y filtros.
 8. **Conteo**: tarea, captura, varianzas.
 9. **Reposición**: sugerencias y generación PR/PO.
 10. **Maestros**: SKU, proveedores, bodegas.
-

10. Reportes y KPIs (definiciones)

- **Exactitud inventario** = $1 - (|variance| / expected)$ agregado.

- **OTD proveedor** = entregas a tiempo / entregas totales (según promised_date).
 - **Lead time real** = received_at - po_confirmed_at.
 - **Cobertura** = on_hand / consumo_promedio.
 - **Aging**: días desde último movimiento.
-

11. Requisitos de QA y pruebas

11.1. Matriz de pruebas por flujo

- PR→RFQ→PO
- PO→GRN→Putaway→Stock
- Transferencias
- Ajustes con aprobación
- Conteos con recount
- Reposición con in_transit

11.2. Invariantes (propiedades a testear)

- Stock derivado del ledger coincide con snapshots.
 - No duplicación de ILE en reintentos.
 - Transiciones de estado válidas.
 - Trazabilidad completa de documentos.
-

12. Telemetría y observabilidad

- Logs estructurados por entidad.
 - Métricas: tiempo de posteo, errores por endpoint, colas.
 - Auditoría accesible por UI.
-

13. Entregables del MVP

1. Backend API + DB.
 2. UI web.
 3. Importador de maestros y saldos.
 4. Reportes esenciales.
 5. Auditoría y RBAC.
 6. Export básico.
-

14. Backlog y fases

14.1. MVP (prioridad alta)

- Maestros mínimos
- PR + aprobaciones
- RFQ + cotizaciones + comparador

- PO + versionado
- GRN + posteo a ledger
- Putaway simple
- Movimientos + ajustes
- Conteo cíclico
- Reposición min/max
- Reportes base

14.2. MVP+1

- App móvil/escáner
 - Putaway dirigido
 - Reservas/picking si hay salidas
 - Calidad avanzada y muestreo
 - Portal proveedor
-

15. Anexos

15.1. Códigos de motivo (Reason codes) sugeridos

- RECEIPT_PO, TRANSFER, ADJUSTMENT_GAIN, ADJUSTMENT_LOSS, RTV, SCRAP, QUARANTINE, RELEASE

15.2. Catálogo de errores estándar

- VALIDATION_ERROR
- PERMISSION_DENIED
- INVALID_STATE_TRANSITION
- INSUFFICIENT_STOCK
- DUPLICATE_POST
- INTEGRATION_FAILURE