

Integration PRD (Product Requirements Document)

1. Propósito

Definir el **modelo formal de integración** (contratos de APIs y eventos) para una plataforma empresarial integral compuesta por módulos independientes pero **coherentes**, con una experiencia unificada (navegación, búsqueda, archivos, permisos, auditoría, notificaciones y trazabilidad end-to-end).

Este documento especifica: - Convenciones y componentes transversales (Gateway, Event Bus, IAM, archivos, búsqueda, notificaciones, auditoría). - **Contratos por módulo**: endpoints (REST), eventos (pub/sub), esquemas de payload, requisitos de seguridad, idempotencia, versionado y compatibilidad. - Escenarios de integración y orquestación (entre módulos) usando APIs y eventos.

2. Alcance

2.1 Módulos en alcance (actuales)

- Intranet
- Comunicaciones
- Canvas
- Gestor de Tareas
- CRM
- Información de Productos
- inventario (compras + stock; sin contabilidad)
- Licitaciones (Mercado Público)
- RRHH

2.2 Integraciones previstas (futuras, fuera del desarrollo interno actual)

- Finanzas/Contabilidad/Tesorería (herramienta existente externa)
- Gestión Comercial completa (CPQ/Cotizaciones/Órdenes de Venta)
- Postventa / Servicio Técnico
- Gestión de Proyectos
- Documentos y Contratos (DMS/CLM + firma)

Nota: Aunque estén fuera del desarrollo actual, se dejan “**stubs**” de integración para no bloquear diseño.

3. Objetivos

1. **Unificación operacional**: mismos conceptos de tenant, workspace, usuario, permisos, archivos, notificaciones y búsqueda.
2. **Coherencia de datos**: una única fuente de verdad por entidad; el resto consume por referencia y read-models.
3. **Escalabilidad y resiliencia**: integración asíncrona por eventos, reintentos, DLQ, outbox.
4. **Auditabilidad**: trazabilidad completa (correlation_id) y audit log transversal.

5. **Compatibilidad:** versionado de APIs/eventos, contratos backward-compatible y deprecación controlada.

4. No Objetivos

- No se define UI/UX en detalle (solo implicancias de integración).
 - No se diseña lógica interna completa de módulos futuros (solo interfaces).
 - No se especifica infraestructura exacta (cloud/provider); sí patrones obligatorios.
-

5. Arquitectura de Integración

5.1 Componentes

5.1.1 API Gateway

Responsable de: - Autenticación y verificación de tokens. - Enrutamiento a módulos por prefijo. - Rate limiting. - Normalización de headers (correlation, request-id). - Aplicación de políticas ABAC "globales" (opcional) y delegación RBAC/ABAC a cada módulo.

Convención de paths: - `/api/{module}/v1/...` - Ej.: `/api/crm/v1/deals`, `/api/tasks/v1/tasks`

5.1.2 Event Bus

Pub/Sub interno. Requisitos: - Topics por módulo (dominio) y por tipo de evento. - Retención configurable por clasificación. - Consumidores idempotentes. - DLQ por módulo. - Esquemas en registry (o validación equivalente).

5.1.3 Platform Core Services

Servicios transversales (pueden ser módulos propios): - IAM (AuthN + sessions) y Directory - Authorization Policy (RBAC/ABAC; motor de políticas opcional) - Files/Assets - Notifications - Search - Audit Log - Integration Hub (conectores externos)

6. Convenciones Globales (Obligatorias)

6.1 Identidad, Tenancy y Seguridad

6.1.1 Token

- JWT firmado (o equivalente) con claims:
 - `tenant_id` (obligatorio)
 - `user_id` (obligatorio)
 - `roles` y/o `groups`
 - `scopes` (permisos de API)
 - `exp`, `iat`, `jti`

6.1.2 Scopes

Scopes por módulo: - `crm.read`, `crm.write` - `tasks.read`, `tasks.write` - `comms.read`, `comms.write` - `canvas.read`, `canvas.write` - `inventory.read`, `inventory.write` - `tenders.read`, `tenders.write` - `products.read`, `products.write` - `hr.read`, `hr.write` - `intranet.read`, `intranet.write` - `files.*`, `search.*`, `notifications.*`, `audit.*`

6.1.3 Clasificación de datos

Todo objeto y evento debe llevar: - `classification`: `PUBLIC_INTERNAL` | `CONFIDENTIAL` | `RESTRICTED` - `pii`: `NONE` | `LOW` | `HIGH`

Regla: objetos `RESTRICTED` o con `pii=HIGH` no se indexan globalmente sin política explícita.

6.2 Headers Estándar

- `Authorization`: `Bearer <token>`
- `X-Request-Id`: `UUID` (generado por gateway si falta)
- `X-Correlation-Id`: `UUID` (propagado cross-módulo; obligatorio en orquestaciones)
- `Idempotency-Key`: `string` (obligatorio en POST/PUT "mutadores" críticos)
- `If-Match`: `ETag` (para concurrencia optimista)
- `X-Actor-User-Id`: opcional (si el sistema actúa en nombre de usuario)

6.3 Idempotencia

6.3.1 Regla

Para endpoints mutadores: - Si llega el mismo `Idempotency-Key` + mismo actor + mismo tenant dentro de ventana (p.ej. 24h), debe devolver el **mismo resultado** (o error equivalente) sin duplicar.

6.3.2 Respuesta idempotente

- `201` (creado) o `200` (replay) con header `Idempotency-Replayed: true|false`.

6.4 Concurrencia Optimista

- Recursos modificables devuelven `ETag`.
- `PATCH/PUT` deben requerir `If-Match` salvo excepciones (comentarios, reacciones, etc.).
- Errores: `412 Precondition Failed` o `409 Conflict`.

6.5 Página, Filtrado y Orden

- Paginación: `limit` (1..200), `cursor`.
- Filtros: `filter[field]=value` o `q`.
- Orden: `sort=field:asc|desc`.

6.6 Modelo de Error Unificado

HTTP + body estándar:

```
{
  "error": {
    "code": "STRING_ENUM",
    "message": "Human readable",
    "details": { "...": "..." },
    "retryable": true,
    "correlation_id": "..."
  }
}
```

Códigos comunes: - VALIDATION_ERROR , UNAUTHORIZED , FORBIDDEN , NOT_FOUND , CONFLICT , PRECONDITION_FAILED , RATE_LIMITED , INTERNAL_ERROR , DEPENDENCY_FAILURE .

6.7 Versionado

6.7.1 APIs

- Versionado por path: /v1/ .
- Cambios breaking → /v2/ .
- Cambios no-breaking: agregar campos opcionales.

6.7.2 Eventos

- event_type incluye versión: module.entity.action.v1 .
- Cambios breaking → ...v2 y coexistencia por ventana.

7. Tipos Canónicos (Schemas Compartidos)

7.1 ObjectRef (enlaces universales)

```
{
  "tenant_id": "tnt_...",
  "object_type": "CRM DEAL | CRM CONTACT | TENDER | PRODUCT | SKU | INVENTORY PO | TASK | CONVERSATION | CANVAS BOARD | ...",
  "object_id": "...",
  "display": { "title": "...", "code": "..." },
  "url": "platform://..."
}
```

7.2 EntityLink (relaciones cross-módulo)

```
{
  "link_id": "lnk_...",
  "from": { "...ObjectRef" },
  "to": { "...ObjectRef" },
```

```
"relation_type": "RELATED_TO|DERIVED_FROM|ATTACHED_TO|DISCUSSION_OF|BLOCKS|...",
  "created_by": "usr_...",
  "created_at": "2025-12-22T...Z"
}
```

7.3 ExternalRef (IDs externos)

```
{
  "external_system": "MERCADO_PUBLICO|ERP_FINANCE|...",
  "external_id": "...",
  "metadata": { "...": "..." }
}
```

7.4 FileRef (archivo universal)

```
{
  "file_id": "fil_...",
  "name": "...",
  "mime_type": "application/pdf",
  "size_bytes": 12345,
  "hash_sha256": "...",
  "classification": "CONFIDENTIAL",
  "created_at": "..."
}
```

7.5 Event Envelope

```
{
  "event_id": "evt_...",
  "event_type": "crm.deal.created.v1",
  "schema_version": 1,
  "occurred_at": "2025-12-22T...Z",
  "tenant_id": "tnt_...",
  "actor": { "type": "USER|SYSTEM", "user_id": "usr_..." },
  "correlation_id": "cor_...",
  "subject": { "...ObjectRef" },
  "classification": "CONFIDENTIAL",
  "pii": "LOW",
  "data": { "...": "..." }
}
```

8. Contratos por Módulo

Formato por endpoint: **Método / Path** — propósito — scopes — request — response — errores — notas de idempotencia.

8.1 Platform Core — IAM & Directory

8.1.1 Endpoints

POST /api/iam/v1/sessions

- **Propósito:** crear sesión (login) o intercambiar token.
- **Scopes:** público (no requiere token) o `iam.write` según modalidad.
- **Request:** `{ "username": "...", "password": "...", "mfa_code": "..." }`
- **Response 200:**
`{ "access_token": "...", "refresh_token": "...", "expires_in": 3600 }`

POST /api/iam/v1/sessions/refresh

- **Propósito:** refresh.
- **Request:** `{ "refresh_token": "..." }`

GET /api/directory/v1/people

- **Propósito:** listar personas (read-model principal desde RRHH).
- **Scopes:** `directory.read`
- **Query:** `q`, `filter[org_unit_id]`, `filter[status]`, `limit`, `cursor`
- **Response 200:** `{ "items": [{ "person_id": "per...", "user_id": "usr...", "display_name": "...", "org_unit": { "id": "...", "name": "..." }, "email": "..." }, ...], "next_cursor": "..." }`

GET /api/directory/v1/people/{person_id}

- **Scopes:** `directory.read`

GET /api/iam/v1/users/{user_id}/effective-permissions

- **Propósito:** calcular permisos efectivos (RBAC+ABAC) para diagnóstico.
- **Scopes:** `iam.read` (admin)
- **Response:** `{ "permissions": ["..."] }`

8.1.2 Eventos

- `iam.user.created.v1`
- `iam.user.disabled.v1`
- `directory.person.upserted.v1`

Payload (`directory.person.upserted.v1` → data):

```
{
  "person_id": "per_...",
  "user_id": "usr_...",
  "display_name": "...",
  "org_unit": {"id": "ou_...", "name": "..."},
  "manager_person_id": "per_...",
  "employment_status": "ACTIVE|INACTIVE",
  "attributes": {"location": "...", "cost_center": "...", "job_title": "..."}
}
```

8.2 Platform Core — Files/Assets

8.2.1 Endpoints

POST /api/files/v1/uploads/init

- **Propósito:** iniciar carga (pre-signed URL).
- **Scopes:** files.write
- **Request:** { "name": "...", "mime_type": "...", "size_bytes": 123, "classification": "CONFIDENTIAL" }
- **Response 201:** { "file_id": "fil_...", "upload_url": "...", "headers": {...}, "expires_at": "..." }

POST /api/files/v1/uploads/{file_id}/complete

- **Propósito:** finalizar y disparar escaneo/preview.
- **Scopes:** files.write
- **Request:** { "hash_sha256": "..." }
- **Response 200:** { "file": {...FileRef}, "status": "READY|PROCESSING" }

POST /api/files/v1/files/{file_id}/links

- **Propósito:** asociar archivo a entidad.
- **Scopes:** files.write
- **Request:** { "target": {...ObjectRef}, "relation_type": "ATTACHED_TO" }
- **Response 201:** { "link": {...EntityLink} }

GET /api/files/v1/files/{file_id}

- **Scopes:** files.read

8.2.2 Eventos

- files.file.created.v1
- files.file.ready.v1
- files.file.linked.v1

8.3 Platform Core — Notifications

8.3.1 Endpoints

GET /api/notifications/v1/inbox

- **Propósito:** bandeja unificada.
- **Scopes:** notifications.read
- **Query:** filter[unread]=true|false, limit, cursor

POST /api/notifications/v1/items/{notification_id}/read

- **Scopes:** notifications.write

8.3.2 Eventos (consumidos)

- task.assigned.v1, task.mentioned.v1
- comms.mention.created.v1
- tender.deadline.reminder.v1
- inventory.approval.required.v1

Payload canónico de notificación (data):

```
{  
  "notification_id": "ntf_...",  
  "recipient_user_id": "usr_...",  
  "type": "MENTION|ASSIGNMENT|DEADLINE|STATUS_CHANGE|APPROVAL",  
  "title": "...",  
  "body": "...",  
  "target": {...ObjectRef},  
  "created_at": "...",  
  "channels": ["IN_APP", "EMAIL"],  
  "priority": "LOW|NORMAL|HIGH"  
}
```

8.4 Platform Core — Search

8.4.1 Endpoints

GET /api/search/v1/query

- **Scopes:** search.read
- **Query:** q, types, workspace_id, limit, cursor
- **Response 200:** { "items": [{ "object": {...ObjectRef}, "snippet":"...", "score":0.12 }], "next_cursor":"..." }

8.4.2 Eventos (ingesta)

Todos los módulos emiten: - search.entity.upserted.v1 - search.entity.deleted.v1

Payload (search.entity.upserted.v1 → data):

```
{  
  "object": "...ObjectRef",  
  "title": "...",  
  "body": "...",  
  "tags": [...],  
  "workspace_id": "wsp...",  
  "acl": {"visibility": "WORKSPACE|PRIVATE|ORG", "allowed_user_ids": [...], "allowed_group_ids": [...]},  
  "updated_at": "..."  
}
```

8.5 Intranet

8.5.1 Endpoints

GET /api/intranet/v1/home

- **Propósito:** composición de home (widgets: tareas asignadas, alertas, noticias).
- **Scopes:** intranet.read
- **Response:** { "widgets": [{"type": "TASKS_ASSIGNED", "data": {...}}, {"type": "TENDER_ALERTS", "data": {...}}] }

CRUD Contenidos

- POST /api/intranet/v1/pages (crear)
- GET /api/intranet/v1/pages/{page_id}
- PATCH /api/intranet/v1/pages/{page_id}
- POST /api/intranet/v1/pages/{page_id}/publish

Page (request/response):

```
{  
  "page_id": "pag...",  
  "title": "...",  
  "slug": "...",  
  "body_rich": "...",  
  "status": "DRAFT|REVIEW|PUBLISHED|ARCHIVED",  
  "tags": [...],  
  "classification": "PUBLIC_INTERNAL",  
  "audience": {"org_units": ["ou..."], "groups": ["grp..."]},  
  "created_by": "usr...",  
  "created_at": "...",  
  "updated_at": "..."  
}
```

8.5.2 Eventos

- `intranet.page.created.v1`
- `intranet.page.published.v1`
- `intranet.announcement.created.v1`

Requisito: al publicar, emitir `search.entity.upserted.v1` y/o evento propio que lo dispare.

8.6 Comunicaciones

8.6.1 Endpoints

POST /api/comms/v1/conversations

- **Propósito:** crear canal/DM.
- **Scopes:** `comms.write`
- **Idempotency:** obligatorio.
- **Request:**

```
{  
    "type": "CHANNEL | DM | GROUP_DM",  
    "workspace_id": "wsp_...",  
    "name": "#licitacion-123",  
    "topic": "...",  
    "is_private": true,  
    "members": ["usr_..."],  
    "links": [ {"target": "...ObjectRef"},  
              {"relation_type": "DISCUSSION_OF"} ]  
}
```

- **Response 201:** { "conversation_id": "cnv_..." , "workspace_id": "wsp_..." , "name": "...", "created_at": "..." }

GET /api/comms/v1/conversations

- **Scopes:** `comms.read`
- **Query:** `workspace_id`, `filter[type]`, `q`, `limit`, `cursor`

POST /api/comms/v1/conversations/{conversation_id}/messages

- **Scopes:** `comms.write`
- **Idempotency:** usar `client_message_id` (o Idempotency-Key).
- **Request:**

```
{  
    "client_message_id": "msgc_...",  
    "content": "...",  
    "format": "MARKDOWN | PLAIN",  
    "thread_root_id": "msg_..." ,
```

```

    "attachments": [{"file_id": "fil_..."}],
    "mentions": ["usr_..."],
    "links": [ {"target": "...ObjectRef"}, "relation_type": "RELATED_TO" } ]
}

```

POST /api/comms/v1/messages/{message_id}/reactions

- **Request:** { "emoji": ":thumbsup:" }

POST /api/comms/v1/conversations/{conversation_id}/members

- **Request:** { "user_id": "usr_...", "role": "MEMBER|MODERATOR" }

8.6.2 Eventos

- comms.conversation.created.v1
- comms.message.posted.v1
- comms.message.edited.v1
- comms.message.deleted.v1
- comms.mention.created.v1
- comms.member.joined.v1

Payload (comms.message.posted.v1 → data):

```
{
  "conversation_id": "cnv_...",
  "message_id": "msg_...",
  "thread_root_id": "msg_...",
  "author_user_id": "usr_...",
  "content": "...",
  "attachments": [{"file_id": "fil_..."}],
  "mentions": ["usr_..."],
  "links": [{"target": "...ObjectRef"}, "relation_type": "RELATED_TO"}],
  "created_at": "..."
}
```

8.7 Gestor de Tareas

8.7.1 Endpoints

POST /api/tasks/v1/workspaces

- **Scopes:** tasks.write
- **Request:** { "name": "...", "visibility": "PRIVATE|ORG|INVITE" }

POST /api/tasks/v1/tasks

- **Propósito:** crear tarea.
- **Scopes:** tasks.write
- **Idempotency:** obligatorio.
- **Request:**

```
{  
    "workspace_id": "wsp_...",  
    "container": {"type": "LIST|FOLDER|SPACE", "id": "..."},  
    "title": "...",  
    "description": "...",  
    "status": "TODO|IN_PROGRESS|DONE",  
    "priority": "LOW|NORMAL|HIGH|URGENT",  
    "assignees": ["usr_..."],  
    "due_at": "...",  
    "tags": [...],  
    "custom_fields": {"tender_id": "...", "deal_id": "..."},  
    "links": [{"target": "...ObjectRef"}, {"relation_type": "RELATED_TO"}]  
}
```

PATCH /api/tasks/v1/tasks/{task_id}

- **Scopes:** tasks.write
- **If-Match:** requerido.

POST /api/tasks/v1/tasks/{task_id}/comments

- **Scopes:** tasks.write
- **Request:** { "body": "...", "mentions": ["usr_..."] }

POST /api/tasks/v1/tasks/{task_id}/attachments

- **Scopes:** tasks.write
- **Request:** { "file_id": "fil_..." }

GET /api/tasks/v1/views

- **Scopes:** tasks.read
- **Query:** workspace_id, type=LIST|BOARD|CALENDAR|GANTT

8.7.2 Eventos

- task.created.v1
- task.updated.v1
- task.status_changed.v1
- task.assigned.v1
- task.comment.added.v1
- task.mentioned.v1

Payload (task.created.v1 → data):

```
{
  "task_id": "tsk_...",
  "workspace_id": "wsp_...",
  "title": "...",
  "status": "TODO",
  "assignees": ["usr_..."],
  "due_at": "...",
  "links": [{"target": "...ObjectRef", "relation_type": "RELATED_TO"}],
  "created_at": "..."
}
```

8.8 Canvas

8.8.1 Endpoints

POST /api/canvas/v1/boards

- **Scopes:** canvas.write
- **Idempotency:** obligatorio.
- **Request:**

```
{
  "workspace_id": "wsp_...",
  "title": "...",
  "visibility": "PRIVATE|WORKSPACE|ORG",
  "links": [{"target": "...ObjectRef", "relation_type": "RELATED_TO"}]
}
```

POST /api/canvas/v1/boards/{board_id}/elements

- **Scopes:** canvas.write
- **Request:**

```
{
  "type": "NOTE|DOCUMENT|LINK|FILE|SHORTCUT",
  "data": {"text": "...", "url": "...", "file_id": "fil...", "target": "...ObjectRef"}, 
  "position": {"x": 10, "y": 20},
  "size": {"w": 400, "h": 200},
  "style": {"...": "..."}
}
```

PATCH /api/canvas/v1/elements/{element_id}

- **If-Match:** requerido.

POST /api/canvas/v1/boards/{board_id}/comments

- **Request:** { "element_id": "elm_...", "body": "...", "mentions": ["usr_..."] }

POST /api/canvas/v1/boards/{board_id}/share-links

- **Propósito:** generar link compatible.
- **Request:** { "scope": "WORKSPACE|ORG", "expires_at": "..." }

POST /api/canvas/v1/boards/{board_id}/exports

- **Propósito:** exportación (PDF/PNG).
- **Request:** { "format": "PDF|PNG", "include_background": true }

8.8.2 Eventos

- canvas.board.created.v1
- canvas.element.created.v1
- canvas.element.updated.v1
- canvas.comment.added.v1
- canvas.share_link.created.v1
- canvas.export.completed.v1

Payload (canvas.export.completed.v1 → data):

```
{  
  "board_id": "brd_...",  
  "export_id": "exp_...",  
  "format": "PDF",  
  "file_id": "fil_...",  
  "status": "COMPLETED|FAILED",  
  "completed_at": "..."  
}
```

8.9 CRM

8.9.1 Endpoints

POST /api/crm/v1/companies

- **Scopes:** crm.write
- **Idempotency:** recomendado.
- **Request:**
{ "name": "...", "tax_id": "...", "industry": "...", "addresses": [...],
 "contacts": [...], "external_refs": [{"...ExternalRef"}] }

POST /api/crm/v1/contacts

- **Request:**

```
{ "first_name": "...", "last_name": "...", "email": "...", "phone": "...", "company_id": "cmp_..." }
```

POST /api/crm/v1/deals

- **Propósito:** crear oportunidad.

- **Request:**

```
{
  "title": "...",
  "company_id": "cmp_...",
  "pipeline_id": "pip_...",
  "stage_id": "stg_...",
  "value": {"amount": 0, "currency": "CLP"},
  "owner_user_id": "usr_...",
  "expected_close_at": "...",
  "external_refs": [{"...ExternalRef"}],
  "links": [{"target": {...ObjectRef}, "relation_type": "DERIVED_FROM"}]
}
```

POST /api/crm/v1/deals/{deal_id}/activities

- **Propósito:** log de actividad (llamada, mail, reunión, nota, tarea).

- **Request:**

```
{
  "type": "CALL|EMAIL|MEETING|NOTE|TASK",
  "occurred_at": "...",
  "summary": "...",
  "details": "...",
  "links": [{"target": {...ObjectRef}, "relation_type": "RELATED_TO"}]
}
```

PATCH /api/crm/v1/deals/{deal_id}

- **If-Match:** requerido.

8.9.2 Eventos

- `crm.company.created.v1`
- `crm.contact.created.v1`
- `crm.deal.created.v1`
- `crm.deal.stage_changed.v1`
- `crm.deal.closed.v1`
- `crm.activity.logged.v1`

Payload (crm.deal.created.v1 → data):

```
{
  "deal_id": "del_...",
  "title": "...",
  "company_id": "cmp_...",
  "pipeline_id": "pip_...",
  "stage_id": "stg_...",
  "owner_user_id": "usr_...",
  "external_refs": [{"...ExternalRef"}],
  "created_at": "..."
}
```

8.10 Información de Productos

8.10.1 Endpoints

POST /api/products/v1/products

- **Scopes:** products.write
- **Idempotency:** obligatorio para imports.
- **Request:**

```
{
  "name": "...",
  "brand": "...",
  "model": "...",
  "category": "...",
  "status": "DRAFT|ACTIVE|RETIRED",
  "attributes": {"...": "..."},
  "source": {"manufacturer": "...", "url": "..."},
  "documents": [{"type": "DATASHEET", "file_id": "fil_..."}]
}
```

POST /api/products/v1/products/{product_id}/documents

- **Request:** { "type": "MANUAL|DATASHEET|CERTIFICATE|EETT|FFTT", "file_id": "fil_...", "version": "1.0" }

POST /api/products/v1/templates

- **Propósito:** plantillas de generación.
- **Request:** { "template_type": "EETT|FFTT", "name": "...", "schema": {...}, "file_id": "fil_..." }

POST /api/products/v1/generation-jobs

- **Propósito:** generar EETT/FFTT.
- **Request:**

```
{
  "job_type": "GENERATE_EETT|GENERATE_FFTT",
  "product_id": "prd_...",
  "template_id": "tpl_...",
  "output_format": "PDF|DOCX",
  "context": {"tender_id": "...", "customer": "..."}
}
```

8.10.2 Eventos

- prod.product.created.v1
- prod.product.updated.v1
- prod.document.version_published.v1
- prod.specsheet.generated.v1

Payload (prod.specsheet.generated.v1 → data):

```
{
  "job_id": "job_...",
  "product_id": "prd_...",
  "template_id": "tpl_...",
  "output_file_id": "fil_...",
  "output_format": "PDF",
  "status": "COMPLETED|FAILED",
  "completed_at": "..."
}
```

8.11 Inventario

8.11.1 Endpoints

POST /api/inventory/v1/warehouses

- **Scopes:** inventory.write
- **Request:** { "name": "...", "code": "WH-01", "locations": [{"code": "A1"}] }

POST /api/inventory/v1/requisitions (PR)

- **Propósito:** crear requerimiento.
- **Idempotency:** obligatorio.
- **Request:**

```
{
  "workspace_id": "wsp_...",
  "requester_user_id": "usr_...",
  "cost_center": "...",
  "needed_by": "..."}
```

```

    "lines": [{"sku_id": "sku_..."}, {"qty": 10, "uom": "UN", "notes": "..."}],
    "links": [{"target": "...ObjectRef"}, {"relation_type": "RELATED_TO"}]
}

```

POST /api/inventory/v1/requisitions/{pr_id}/approve

- **Request:** { "decision": "APPROVE|REJECT", "reason": "..." }

POST /api/inventory/v1/rfqs

- **Request:** { "pr_id": "pr_...", "suppliers": ["sup_..."], "due_at": "..." }

POST /api/inventory/v1/pos

- **Propósito:** crear orden de compra.
- **Idempotency:** obligatorio.
- **Request:**

```

{
  "supplier_id": "sup_...",
  "currency": "CLP",
  "lines": [{"sku_id": "sku_..."}, {"qty": 10, "unit_price": 1000}],
  "shipping": {"incoterm": "...", "eta": "..."},
  "links": [{"target": "...ObjectRef"}, {"relation_type": "DERIVED_FROM"}]
}

```

POST /api/inventory/v1/grns

- **Propósito:** registrar recepción.
- **Idempotency:** obligatorio.
- **Request:**

```

{
  "po_id": "po_...",
  "warehouse_id": "wh_...",
  "received_at": "...",
  "lines": [{"sku_id": "sku_..."}, {"qty": 10, "lot": "...", "serials": ["..."]}]
}

```

POST /api/inventory/v1/ledger/entries

- **Propósito:** registrar movimiento inmutable (solo sistema o roles autorizados).
- **Request:**

```

{
  "entry_type": "RECEIPT|ISSUE|TRANSFER|ADJUSTMENT",
  "warehouse_id": "wh_...",
  "location_id": "loc_...",
  "sku_id": "sku_..."
}

```

```

    "qty_delta": 10,
    "reason": "...",
    "links":[{"target":"...ObjectRef"}, "relation_type": "RELATED_TO"]
}

```

8.11.2 Eventos

- inv.pr.created.v1
- inv.pr.approved.v1
- inv.rfq.sent.v1
- inv.po.created.v1
- inv.grn.posted.v1
- inv.ledger.posted.v1
- inv.stock.snapshot.updated.v1

Payload (inv.stock.snapshot.updated.v1 → data):

```
{
  "sku_id": "sku_...",
  "warehouse_id": "wh_...",
  "location_id": "loc_...",
  "on_hand": 100,
  "reserved": 0,
  "available": 100,
  "as_of": "..."
}
```

8.12 Licitaciones (Mercado Público)

8.12.1 Endpoints

POST /api/tenders/v1/sources/ingestions

- **Propósito:** iniciar ingesta (manual o scheduler).
- **Scopes:** tenders.write
- **Idempotency:** obligatorio.
- **Request:** { "source": "MERCADO_PUBLICO", "mode": "DELTA|FULL", "since": "..." }

GET /api/tenders/v1/tenders

- **Scopes:** tenders.read
- **Query:** filter[state], filter[closing_before], q, limit, cursor

POST /api/tenders/v1/rules

- **Propósito:** reglas de matching.
- **Request:**

```
{
  "name": "...",
  "enabled": true,
  "criteria": {"keywords": [...], "categories": [...], "buyer_ids": [...]},
  "actions": {"create_alert": true, "priority": "HIGH"}
}
```

POST /api/tenders/v1/tenders/{tender_id}/convert

- **Propósito:** convertir a oportunidad CRM + workspace.
- **Scopes:** tenders.write
- **Idempotency:** obligatorio.
- **Request:**

```
{
  "crm": {"create_deal": true, "pipeline_id": "pip_...", "stage_id": "stg_...", "owner_user_id": "usr_..."}, "collaboration": {"create_conversation": true, "create_board": true, "create_task_bundle": true}, "workspace_id": "wsp_..."
}
```

- **Response 200:** { "deal_id": "del_...", "conversation_id": "cnv_...", "board_id": "brd_...", "created_task_ids": ["tsk_..."] }

8.12.2 Eventos

- tender.source_ingested.v1
- tender.match.created.v1
- tender.opportunity.created.v1
- tender.status.changed.v1
- tender.deadline.reminder.v1

Payload (tender.match.created.v1 → data):

```
{
  "tender_id": "tnd_...",
  "rule_id": "rul_...",
  "score": 0.82,
  "reasons": ["keyword: bombas", "category: ..."],
  "priority": "HIGH",
  "created_at": ...
}
```

8.13 RRHH

8.13.1 Endpoints

POST /api/hr/v1/employees

- **Scopes:** hr.write
- **Idempotency:** obligatorio.
- **Request:**

```
{  
    "person":  
        {"first_name": "...", "last_name": "...", "national_id": "...", "email": "..."},  
        "employment": {"type": "INDEFINITE|FIXED_TERM|HONORARIOS|PRACTICE",  
                      "start_date": "...", "org_unit_id": "ou...", "job_title": "..."},  
        "manager_person_id": "per...",  
        "attributes": {"location": "...", "cost_center": "..."}  
}
```

PATCH /api/hr/v1/employees/{employee_id}

- **If-Match:** requerido.

POST /api/hr/v1/employees/{employee_id}/terminate

- **Request:** { "date": "...", "reason": "..." }

GET /api/hr/v1/org-units

- **Scopes:** hr.read

8.13.2 Eventos

- hr.employee.created.v1
- hr.employee.updated.v1
- hr.employee.terminated.v1
- hr.orgunit.updated.v1

Payload (hr.employee.created.v1 → data):

```
{  
    "employee_id": "emp...",  
    "person_id": "per...",  
    "user_id": "usr...",  
    "org_unit_id": "ou...",  
    "job_title": "...",  
    "employment_type": "INDEFINITE",  
    "start_date": "...",  
    "manager_person_id": "per...",  
    ...  
}
```

```
        "attributes": {"location":"...","cost_center":"..."}  
    }
```

9. Contratos de Integración (Orquestaciones Estándar)

9.1 Convertir licitación a ejecución (Tender → CRM + Comms + Tasks + Canvas)

Trigger: usuario ejecuta `/tenders/{id}/convert`.

Secuencia mínima (correlation_id único): 1) Tender: valida estado y ownership. 2) CRM: `POST /crm/v1/deals` con `ExternalRef(MERCADO_PUBLICO)`. 3) Comms: `POST /comms/v1/conversations` con link a Deal y Tender. 4) Tasks: `POST /tasks/v1/tasks` (bundle) con links a Deal/Tender. 5) Canvas: `POST /canvas/v1/boards` con links. 6) Publicar `EntityLink` entre todos.

Errores críticos: - Si falla CRM, abortar (no crear salas ni tareas). - Si falla Comms/Tasks/Canvas tras crear Deal, crear **tarea de remediación** y notificar.

9.2 Generación de EETT/FFT para licitación

1) Tender selecciona productos. 2) Products: `POST /generation-jobs`. 3) Al completar: evento `prod.specsheet.generated.v1` con `output_file_id`. 4) Files: link `file_id` al Tender/Deal/Board. 5) Notificación a responsable.

9.3 Reabastecimiento por oportunidad

- (Stub para futuro Comercial/Proyectos/Postventa)
- Cuando Deal cambia a "Won": emitir `crm.deal.closed.v1`.
- Consumidor inventario sugiere/crea PR (según política) o crea tareas de abastecimiento.

10. Requisitos No Funcionales (Integración)

10.1 Disponibilidad

- Core (IAM, Gateway, Files, Bus, Notifications, Search): objetivo $\geq 99.9\%$.
- Módulos: $\geq 99.5\%$.

10.2 Observabilidad

- Todos los requests deben registrar: `tenant_id`, `user_id`, `request_id`, `correlation_id`, latencias, status.

- Event consumers: métricas de lag, retries, DLQ.

10.3 Seguridad

- Cifrado en tránsito (TLS).
- Cifrado en reposo en Files.
- Masking de PII en logs.
- Auditoría de accesos a RRHH y descargas de archivos sensibles.

10.4 Compatibilidad

- Campos nuevos siempre opcionales.
 - Eventos: no eliminar campos; deprecación por versionado.
-

11. Stubs de Integración para Módulos Futuros

11.1 Finanzas Externa (ERP)

Eventos a exportar

- `inv.po.created.v1`
- `inv.grn.posted.v1`
- (futuro) `sales.invoice.issued.v1`, `sales.payment.received.v1`

Endpoint del Integration Hub

- `POST /api/integrations/v1/erp-finance/exports`

11.2 Documentos y Contratos

- Objeto canónico futuro: `CONTRACT`, `DOCUMENT_PACKAGE`.
- Integración con Files y EntityLink (sin duplicar almacenamiento).

11.3 Proyectos

- Objeto canónico futuro: `PROJECT` con workspace asociado.
- Consumir `crm.deal.closed.v1` para crear proyecto y baseline de tareas.

11.4 Postventa

- Objeto canónico futuro: `ASSET_INSTALLED`, `WORK_ORDER`, `SLA`.
 - Enlaces con Product, Inventory, CRM.
-

12. Checklist de Implementación (para ingeniería)

1) Implementar headers estándar y propagación `correlation_id`. 2) Implementar idempotencia en todos los POST críticos. 3) Definir catálogo global `object_type` y registry de schemas. 4) Montar outbox por módulo + DLQ. 5) Publicar eventos mínimos por módulo (sección 8). 6) Implementar Files como servicio único y migrar attachments. 7) Implementar Search ingestion por eventos. 8) Implementar Notification Center y enrutar eventos. 9) Implementar EntityLink/ExternalRef como tablas/servicio transversal. 10) Probar orquestación Tender→Deal→Collab end-to-end con fallas parciales.

13. Apéndice: Enumeraciones sugeridas

- `relation_type` : RELATED_TO, DERIVED_FROM, ATTACHED_TO, DISCUSSION_OF, BLOCKS, DEPENDS_ON, OWNED_BY
- `classification` : PUBLIC_INTERNAL, CONFIDENTIAL, RESTRICTED
- `pii` : NONE, LOW, HIGH
- `workspace_visibility` : PRIVATE, WORKSPACE, ORG