Taller Scrum P1 Entrega Modulo Inventario

Juan Sebastian Muñoz Melo

Docente

Carlos Eduardo Mujica Reyes

Universidad Manuela Beltran - UMB

Ingenieria

Ingenieria de Software – Arquitectura de Software

2026

Parte 1 – Modulo Inventario

1. VISIÓN (Inventario)

Para el jefe de almacén que necesita controlar existencias en tiempo real, nuestro módulo de inventario permitiria registrar y monitorear productos con trazabilidad de movimientos, a diferencia de hojas de cálculo manuales, ofreciendo control automatizado y alertas.

Vision (CRM)

Para el asesor comercial que necesita gestionar clientes y hacer seguimiento de interacciones, nuestro módulo CRM permite registrar, buscar y administrar clientes, a diferencia de registros dispersos o manuales, ofreciendo organización centralizada y trazabilidad comercial.

2. PERSONAS

1. Jefe de Almacén:Responsable del control general de inventario; Necesita reportes y alertas.
2. Operario de Inventario:Registra entradas y salidas; Necesita rapidez y validaciones claras.
3. Asesor comercial: Registra clientes; Consulta historial;Necesita búsqueda rápida.
4. Administrador

Problemática del flujo Actual:

- Se recibe mercancía.
- Se anota en Excel.

- Se actualiza manualmente stock.
- No hay validación automática.
- Reportes se hacen manuales.
- Errores frecuentes.

Problemática del flujo actual Modulo del CRM:

- Informacion de clientes dispersa.
- Duplicidad de registros
- Falta de historial de interacciones
- Ausencia de seguimiento estructurado
- Dificultad para medir el desempeño
- Riesgo de perdida de información
- Comunicación no estandarizada
- Integracion inexistente con otros modulos

Flujo Esperado (El que hace el ERP)

1. Registrar producto en sistema.
2. Registrar entrada automáticamente.
3. Sistema actualiza stock.
4. Validación automática.
5. Generación automática de movimientos.
6. Reportes en tiempo real.
7. Alertas de stock mínimo.

Flujo Esperado (El que hace el ERP) — CRM

1. **Registrar cliente en el sistema.**

   Se ingresan los datos obligatorios y el sistema valida duplicados automáticamente.

2. **Clasificación automática del estado inicial.**

   El cliente se crea como *Prospecto* o según el tipo definido.

3. **Registro estructurado de interacciones.**

   Cada llamada, reunión o correo queda almacenado en el historial del cliente.

4. **Actualización automática del estado comercial.**

   El sistema permite cambiar de Prospecto → En negociación → Cliente activo.

5. **Generación automática de historial y trazabilidad.**

   Todas las acciones quedan registradas con fecha, usuario y detalle.

6. **Consulta y filtrado en tiempo real.**

   Búsqueda por nombre, documento, estado o fecha de creación.

7. **Reportes y métricas comerciales.**

   Indicadores como:

   - Nuevos clientes
   - Tasa de conversión
   - Seguimientos pendientes

8. **Integración con facturación e inventario.**

   Visualización de compras, facturas y comportamiento del cliente.

3. Épicas Inventario:

Gestión de productos

Gestión de movimientos

Reportes y control

Epicas CRM:

Gestión de clientes

Gestion de Interacciones

Pipeline Comercial

Reportes y métricas comerciales

Integracion con otros modulos

5. HISTORIAS DE USUARIO

INV-01 — Registrar Nuevo Producto

Campo	**Detalle**
Título	Registrar nuevo producto
Historia	Como jefe de almacén quiero registrar un nuevo producto con código, descripción, unidad y stock inicial para poder gestionarlo en el sistema.
Criterios de aceptación	1. Dado que estoy en el formulario de nuevo producto cuando ingreso código, descripción, unidad y stock inicial válidos entonces el sistema guarda el producto y lo muestra en la lista. 2. Dado que el código ya existe cuando intento guardar el producto

Campo	Detalle
	entonces el sistema muestra mensaje de error y no permite duplicados. 3. Dado que dejo campos obligatorios vacíos cuando intento guardar entonces el sistema muestra validaciones correspondientes. 4. Dado que el stock inicial es negativo cuando intento registrar entonces el sistema rechaza el registro.
Notas de diseño	Formulario con validación en tiempo real, campos obligatorios marcados, botón Guardar y Cancelar, mensaje de confirmación tras registro exitoso.
Impacto arquitectónico	Se crea entidad Producto en dominio. Endpoint POST /productos. Actualiza diagrama de clases y componentes.
Datos de prueba	Código: P001, Descripción: Teclado, Unidad: Unidad, Stock: 50
Story Points	3
Prioridad	Alta
Valor	Alto
Riesgo	Bajo

INV-02 — Registrar Entrada de Stock

Campo	Detalle
Título	Registrar entrada de stock
Historia	Como operario quiero registrar una entrada de stock para actualizar las existencias tras una compra o reposición.
Criterios de aceptación	1. Dado que existe un producto registrado cuando ingreso una cantidad positiva entonces el stock aumenta correctamente.2. Dado que registro la entrada entonces el sistema guarda un movimiento tipo "entrada" con fecha y usuario.3. Dado que ingreso cantidad negativa cuando intento registrar entonces el sistema muestra error.4. Dado que el producto no existe cuando intento registrar entonces el sistema impide la operación.
Notas de diseño	Formulario con selección de producto, campo cantidad, fecha automática, confirmación visual de éxito.
Impacto arquitectónico	Se agrega entidad Movimiento. Endpoint POST /movimientos/entrada. Actualiza diagrama de secuencia.
Datos de prueba	Producto: P001, Cantidad: 20
Story Points	5
Prioridad	Alta

Campo	Detalle
Valor	Alto
Riesgo	Medio

INV-03 — Registrar Salida de Stock

Campo	Detalle
Título	Registrar salida de stock
Historia	Como operario quiero registrar una salida de stock para reflejar ventas o consumos internos.
Criterios de aceptación	1. Dado que existe un producto con stock suficiente cuando registro salida válida entonces el stock disminuye correctamente.2. Dado que la cantidad supera el stock disponible cuando intento registrar entonces el sistema muestra mensaje de error.3. Dado que registro salida entonces el sistema guarda movimiento tipo "salida" con fecha y usuario.4. Dado que ingreso cantidad negativa cuando intento registrar entonces el sistema rechaza la operación.
Notas de diseño	Formulario similar a entrada pero con indicador rojo para salida, validación visual de stock disponible.
Impacto arquitectónico	Endpoint POST /movimientos/salida. Validación en servicio InventarioService. Actualiza secuencia de salida.
Datos de prueba	Producto: P001, Stock actual: 50, Salida: 10
Story Points	5
Prioridad	Alta
Valor	Alto
Riesgo	Medio

INV-04 — Consultar Stock Actual

Campo	Detalle
Título	Consultar stock actual
Historia	Como jefe de almacén quiero consultar el stock actual por producto para tomar decisiones de reabastecimiento.
Criterios de aceptación	1. Dado que hay productos registrados cuando accedo a la vista inventario entonces veo la lista con código, descripción y stock.2. Dado que utilizo el filtro por código cuando aplico búsqueda entonces se muestran coincidencias.3. Dado que filtro por rango de stock cuando aplico entonces el sistema muestra productos dentro del rango.4. Dado que no existen productos cuando accedo entonces el sistema muestra mensaje informativo.
Notas de diseño	Tabla con paginación, buscador dinámico, filtro por rango, color rojo para stock bajo.
Impacto arquitectónico	Endpoint GET /productos con filtros opcionales. No modifica dominio. Actualiza secuencia de consulta.
Datos de prueba	Producto A: 100, Producto B: 5, Producto C: 0
Story Points	3
Prioridad	Media
Valor	Medio
Riesgo	Bajo

INV-05 — Alerta de Stock Mínimo

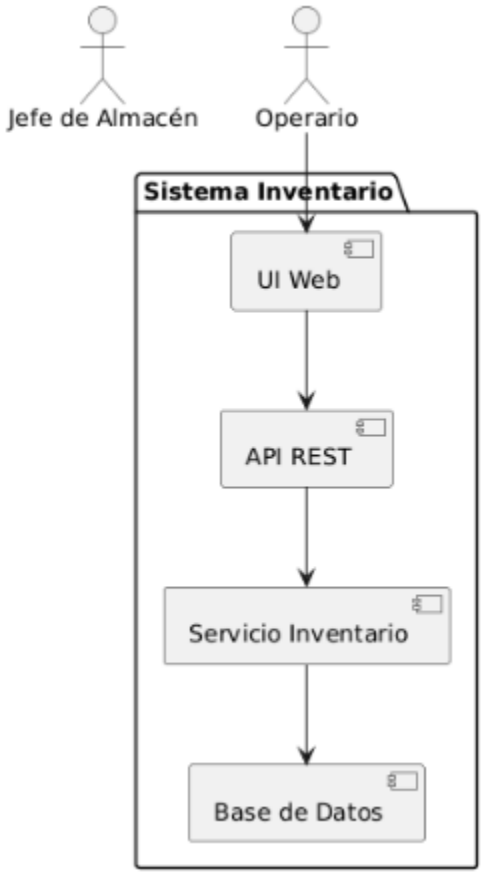
Campo	Detalle
Título	Generar alerta de stock mínimo
Historia	Como jefe de almacén quiero recibir alerta cuando el stock esté por debajo del mínimo para evitar quiebres de inventario.
Criterios de aceptación	1. Dado que un producto tiene mínimo configurado cuando el stock baja de ese nivel entonces el sistema genera alerta.2. Dado que se genera alerta entonces el sistema muestra notificación visual.3. Dado que el stock vuelve a nivel normal entonces la alerta desaparece.4. Dado que no hay productos bajo mínimo entonces no se muestran alertas.
Notas de diseño	Icono de alerta en lista, badge rojo, notificación tipo toast.
Impacto arquitectónico	Agrega atributo stockMinimo en Producto. Lógica de validación en servicio. Actualiza diagrama de clases.
Datos de prueba	Producto con mínimo 10 y stock actual 8
Story Points	8
Prioridad	Media
Valor	Medio
Riego	Medio

INV-06 — Reporte de Movimientos

Campo	Detalle
Título	Generar reporte de movimientos
Historia	Como auditor quiero generar un reporte de movimientos por rango de fechas para control y trazabilidad.
Criterios de aceptación	1. Dado que existen movimientos registrados cuando selecciono rango de fechas entonces el sistema muestra listado correspondiente.2. Dado que genero reporte entonces incluye tipo, fecha, usuario y cantidad.3. Dado que no existen movimientos en el rango entonces el sistema muestra mensaje informativo.4. Dado que solicito exportar entonces el sistema permite descargar reporte en formato PDF o CSV.
Notas de diseño	Selector de fecha inicio-fin, botón generar, tabla de resultados, botón exportar.
Impacto arquitectónico	Endpoint GET /movimientos?fechaInicio&fechaFin. Consulta optimizada en base de datos.
Datos de prueba	Fecha inicio: 01/02/2026 – Fecha fin: 15/02/2026
Story Points	5
Prioridad	Baja
Valor	Bajo
Riego	Bajo

Diagrama de componentes:

```
@startuml
actor "Jefe de Almacén"
actor Operario

package "Sistema Inventario" {
  [UI Web]
  [API REST]
  [Servicio Inventario]
  [Base de Datos]
}

Operario --> [UI Web]
[UI Web] --> [API REST]
[API REST] --> [Servicio Inventario]
[Servicio Inventario] --> [Base de Da

@enduml
```



Codigo Diagrama de componentes:

@startuml

actor "Jefe de Almacén"

actor Operario

package "Sistema Inventario" {

[UI Web]

[API REST]

[Servicio Inventario]

[Base de Datos]

}

Operario --> [UI Web]

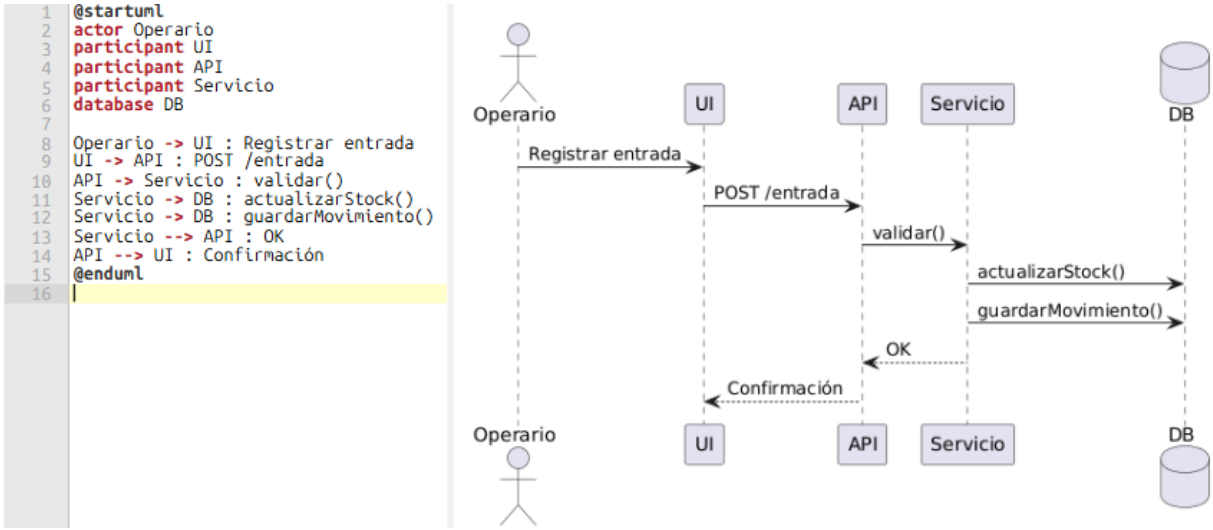
[UI Web] --> [API REST]

[API REST] --> [Servicio Inventario]

[Servicio Inventario] --> [Base de Datos]

@enduml

Diagrama de Secuencia: Entrada de Stock



Codigo diagrama de secuencia: Entrada de Stock

@startuml

actor Operario

participant UI

participant API

participant Servicio

database DB

Operario -> UI : Registrar entrada

UI -> API : POST /entrada

API -> Servicio : validar()

Servicio -> DB : actualizarStock()

Servicio -> DB : guardarMovimiento()

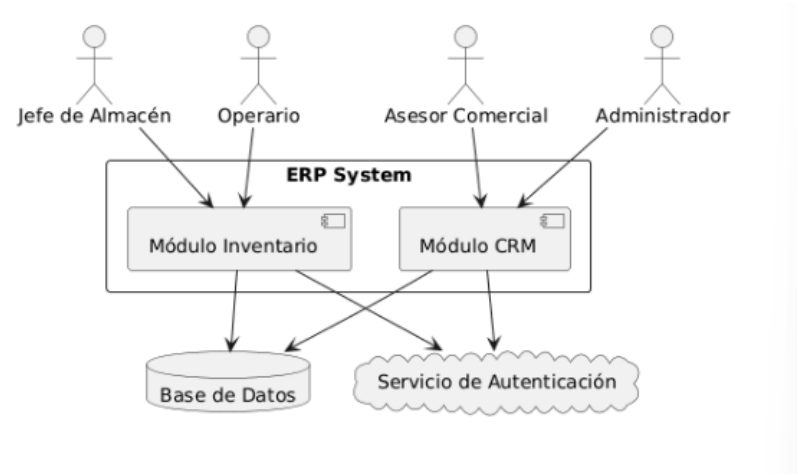
Servicio --> API : OK

API --> UI : Confirmación

@enduml

Diagrama de Contexto: Actualizado a personas:

```
@startuml
actor "Jefe de Almacén"
actor "Operario"
actor "Asesor Comercial"
actor "Administrador"

rectangle "ERP System" {
  component "Módulo Inventario"
  component "Módulo CRM"
}

database "Base de Datos"
cloud "Servicio de Autenticación"

"Jefe de Almacén" --> "Módulo Inventa
"Operario" --> "Módulo Inventario"
"Asesor Comercial" --> "Módulo CRM"
"Administrador" --> "Módulo CRM"

"Módulo Inventario" --> "Base de Dato
"Módulo CRM" --> "Base de Datos"
"Módulo Inventario" --> "Servicio de
"Módulo CRM" --> "Servicio de Autenti

@enduml
```



Codigo de diagrama de contexto: actualizado a personas

@startuml

actor "Jefe de Almacén"

actor "Operario"

actor "Asesor Comercial"

actor "Administrador"

rectangle "ERP System" {

  component "Módulo Inventario"

  component "Módulo CRM"

}

database "Base de Datos"

cloud "Servicio de Autenticación"

"Jefe de Almacén" --> "Módulo Inventario"

"Operario" --> "Módulo Inventario"

"Asesor Comercial" --> "Módulo CRM"

"Administrador" --> "Módulo CRM"

"Módulo Inventario" --> "Base de Datos"

"Módulo CRM" --> "Base de Datos"

"Módulo Inventario" --> "Servicio de Autenticación"

"Módulo CRM" --> "Servicio de Autenticación"

@enduml

Definicion del DoR:

- Historia clara
- Criterios completos
- Wireframe disponible
- Impacto arquitectónico identificado
- Estimación realizada

Definicion del DoD:

- Código implementado
- PR aprobada
- Pruebas pasadas
- Diagramas actualizados
- Demo funcional

Decisiones arquitectónicas o ADR

ADR-01 — Arquitectura por Capas

Estado: Propuesto

Contexto:
El sistema ERP requiere separación clara de responsabilidades para facilitar mantenimiento, pruebas y escalabilidad.

Decisión:
Adoptar arquitectura por capas:

- Presentación (UI)
- Aplicación (API)
- Dominio (Reglas de negocio)
- Persistencia (Acceso a datos)

Consecuencias:

- Facilita pruebas unitarias.
- Permite cambiar base de datos sin afectar dominio.
- Mejora mantenibilidad.

ADR-02 — Persistencia Relacional

Estado: Propuesto

Contexto:
Se requiere consistencia fuerte en inventario y clientes.

Decisión:
Usar base de datos relacional con integridad referencial.

Consecuencias:

- Evita inconsistencias de stock.
- Permite consultas complejas.
- Facilita generación de reportes.

ADR-03 — Estrategia de Autenticación

Estado: Propuesto

Contexto:
El sistema requiere control por roles (Operario, Jefe, Asesor, Admin).

Decisión:
Implementar autenticación basada en roles (RBAC).

Consecuencias:

- Control de acceso por módulo.
- Seguridad básica implementada.
- Escalable para futuros permisos.

SPRINT PLANNING 1

Objetivo del Sprint:

Implementar el flujo básico de inventario permitiendo registrar productos, consultar stock y registrar movimientos con validación de stock.

Selección y estimación:

Historias Seleccionadas para el Sprint 1

Historia	Descripción	Story Points
INV-01	Registrar producto	3
INV-02	Registrar movimiento	5
INV-04	Consultar stock	3

Total Sprint: 11 Story Points (Sprint de capacidad baja – media)

ID Historia	Nombre	Descripción Corta	Prioridad	Story Points	Responsable	Estado Inicial
INV-01	Registrar Producto	Crear producto con validaciones básicas	Alta	3	Pendiente	To Do
INV-02	Registrar Movimiento	Registrar entrada/salida validando stock	Alta	5	Pendiente	To Do
INV-04	Consultar Stock	Visualizar stock con filtros	Media	3	Pendiente	To Do

Planning Poker (Justificación breve)

- INV-01 → 3 SP: CRUD simple con validaciones
- INV-02 → 5 SP: Lógica de negocio + validación stock
- INV-04 → 3 SP: Consulta + filtros

Historia	Complejidad Técnica	Riesgo	Justificación	SP
INV-01	Media	Bajo	CRUD simple + validaciones	3
INV-02	Alta	Medio	Lógica de negocio + validación stock	5
INV-04	Baja	Bajo	Consulta con filtros	3

Explicacion técnica:

INV-01 — Registrar Producto

Tipo	Tarea
UI	Formulario producto
Backend	Endpoint POST /productos
Dominio	Validación campos obligatorios

Tipo	Tarea
DB	Tabla producto
Pruebas	Unit test creación
Docs	Actualizar diagrama clases

INV-02 — Movimiento

Tipo	Tarea
UI	Formulario movimiento
Backend	Endpoint POST /movimientos
Dominio	Regla: no permitir stock negativo
DB	Tabla movimiento
Pruebas	Test validación stock
Docs	Diagrama secuencia

INV-04 — Consulta

Tipo	Tarea
UI	Tabla con filtros
Backend	GET /productos
DB	Query optimizada
Pruebas	Test filtro

Distribución estimada por días para SPRINT 1:

Día	Historia en foco
1–2	INV-01
3–6	INV-02
7	INV-04
8	Pruebas + ajustes + demo

Limite WIP que se acuerda para el SPRINT 1:

Un máximo de 2 historias en progreso de manera simultanea ya que esto evitaría la saturación y como tal mejoraría el foco de atención y capacidades del equipo de trabajo.

Nota: No se debe iniciar una nueva historia hasta no cerrar una que ya este en progreso.

BURNDOWN SPRINT 1: