

Caso de negocio

La empresa Paperplast S.A.S. se dedica a la comercialización de empaques plásticos y de papel. Actualmente, enfrenta un problema en el manejo de inventarios: los registros se llevan de forma manual en hojas de cálculo, lo que ocasiona errores en las cantidades disponibles, pérdidas por faltantes o excesos de stock y dificultades para responder rápidamente a los pedidos de los clientes.

Los usuarios principales del sistema son:

- Administrador de bodega, quien necesita registrar ingresos, salidas y ajustes de inventario.
- Área comercial, que consulta la disponibilidad de productos para ofrecer respuestas rápidas a clientes.
- Gerencia, que requiere reportes confiables sobre movimientos y existencias para la toma de decisiones.

El valor esperado con el desarrollo de esta aplicación es reducir errores humanos, optimizar el control de inventario y facilitar la gestión de la información en tiempo real. Con ello se espera mejorar la eficiencia, brindar un mejor servicio al cliente y contar con un soporte confiable para la toma de decisiones estratégicas.

Historias de usuario:

HU1. Como Administrador quiero crear un producto para tenerlo en el inventario.

Given: No existe producto con el mismo código, When: registro los datos, Then: el producto queda guardado.

HU2. Como Operador quiero registrar una venta para disminuir stock.

Given: producto con stock suficiente, When: ingreso cantidad vendida, Then: stock se reduce y se registra movimiento.

HU3. Como Administrador quiero ajustar stock manualmente para corregir errores.

Given: producto existe, When: indico nueva cantidad o ajuste, Then: stock queda actualizado.

HU4. Como Operador quiero buscar producto por código o nombre para ver su disponibilidad.

Given: hay productos registrados, When: busco por texto, Then: veo lista filtrada.

HU5. Como Administrador quiero listar productos con stock bajo para reabastecerlos.

Given: umbral de stock (p. ej. 5), When: pido reporte, Then: obtengo lista de productos con cantidad \leq umbral.

HU6. Como Administrador quiero guardar y cargar datos desde un archivo JSON para persistir el inventario entre ejecuciones.

Given: inventario en memoria, When: solicito guardar, Then: se genera inventario.json.

Requerimientos de negocio

RN-01: Cada producto tiene código único.

RN-02: No se permite vender más cantidad de la que hay en stock.

RN-03: El sistema permite ajustar stock por inventario físico.

RN-04: El sistema debe permitir exportar/importar inventario.

RN-05: Reporte de stock bajo con umbral configurable.

Requerimientos funcionales

Requisitos Funcionales - 01: Crear, leer, actualizar y eliminar productos

Requisitos Funcionales - 02: Registrar venta lo cual disminuye stock.

Requisitos Funcionales - 03: Ajustar stock de forma manual

Requisitos Funcionales - 04: Buscar productos por código o nombre.

Requisitos Funcionales - 05: Listar todos los productos.

Requisitos Funcionales - 06: Guardar y cargar inventario en inventario.json.

Requisitos Funcionales - 07: Reporte de productos con stock bajo.

Requisitos Funcionales - 08: Menú de consola con opciones claras.

Requerimientos no funcionales

Requisitos no Funcionales - 01: Operaciones comunes responden < 1 s para <= 500 productos.

Requisitos no Funcionales - 02: Mensajes de error claros (ejemplo: "No hay stock suficiente").

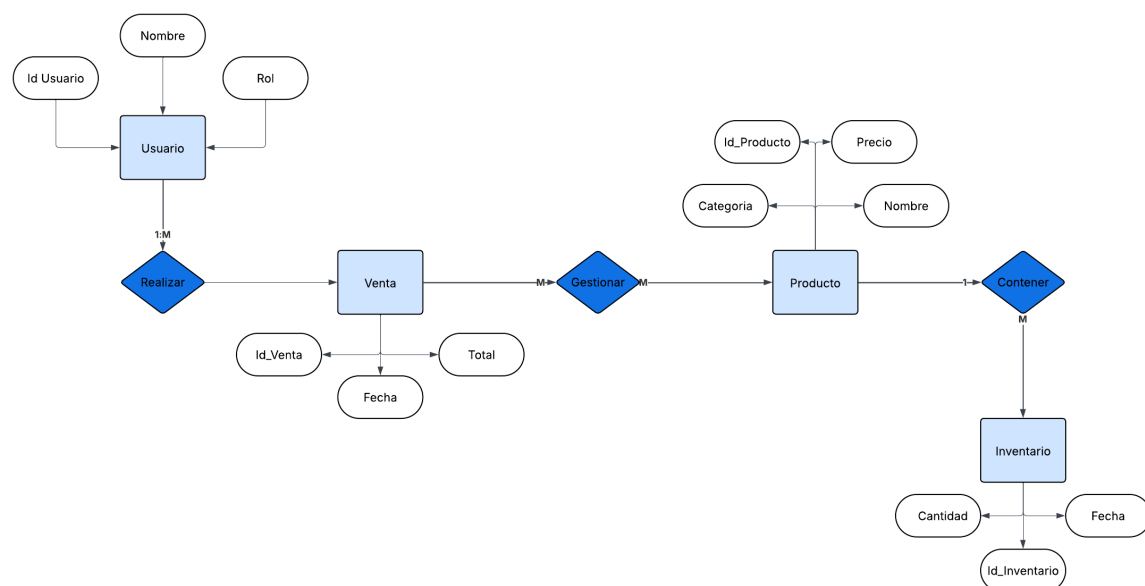
Requisitos no Funcionales - 03: Logging básico acciones principales en consola.

Requisitos no Funcionales - 04: Código legible con nombres en español simples.

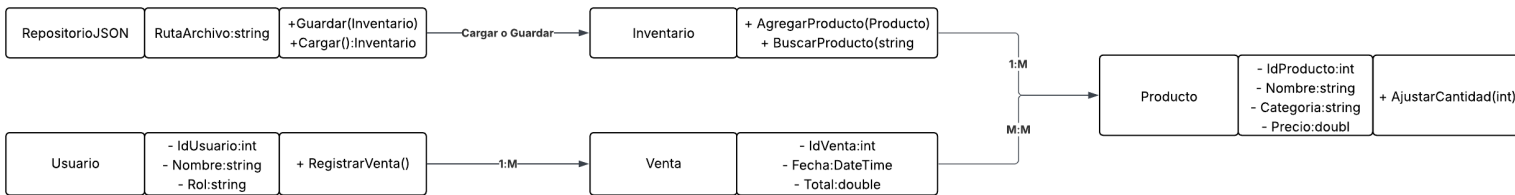
Requisitos no Funcionales - 05: Lógica separada por interfaces.

Requisitos no Funcionales - 06: Portabilidad (.NET 8, Windows/Linux).

Modelo conceptual



Diseño POO



Evidencias

Validación de instalador .NET:

```
Microsoft Windows [Versión 10.0.26100.5074]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sebastián>dotnet --version
8.0.414

C:\Users\Sebastián>|
```

Ejecutar el código:

```
Microsoft Windows [Versión 10.0.26100.5074]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sebastián>dotnet --version
8.0.414

C:\Users\Sebastián>cd Documents

C:\Users\Sebastián\Documents>cd "Programacion de Software"

C:\Users\Sebastián\Documents\Programacion de Software>cd Paperplast

C:\Users\Sebastián\Documents\Programacion de Software\Paperplast>cd Paperplast.Console

C:\Users\Sebastián\Documents\Programacion de Software\Paperplast\Paperplast.Console>dotnet run

--- Paperplast - Inventario ---
1) Crear producto  2) Vender  3) Ajustar stock  4) Buscar  5) Listar  6) Stock bajo  0) Salir
Opción: |
```

Creación de productos:

```
--- Paperplast - Inventario ---
1) Crear producto  2) Vender  3) Ajustar stock  4) Buscar  5) Listar  6) Stock bajo  0) Salir
Opción: 1
Código: 0002
Nombre: Lápiz Mirado 2
Precio: 1000
Cantidad inicial: 3
Producto creado.
```

Venta SIN stock:

```
--- Paperplast - Inventario ---
1) Crear producto  2) Vender  3) Ajustar stock  4) Buscar  5) Listar  6) Stock bajo  0) Salir
Opción: 2
Código: 0002
Cantidad vendida: 5
Error: No hay stock suficiente
```

Venta CON stock:

```
--- Paperplast - Inventario ---
1) Crear producto  2) Vender  3) Ajustar stock  4) Buscar  5) Listar  6) Stock bajo  0) Salir
Opción: 2
Código: 0002
Cantidad vendida: 2
Venta registrada.
```

Alerta Stock Bajo:

```
--- Paperplast - Inventario ---
1) Crear producto  2) Vender  3) Ajustar stock  4) Buscar  5) Listar  6) Stock bajo  0) Salir
Opción: 6
Umbral: 2
[0001] Pliego Cartulina Blanca - Cant: 0 - Precio: $ 3.000,00
[0002] Lápiz Mirado 2 - Cant: 1 - Precio: $ 1.000,00
```

Listar productos:

```
--- Paperplast - Inventario ---
1) Crear producto  2) Vender  3) Ajustar stock  4) Buscar  5) Listar  6) Stock bajo  0) Salir
Opción: 5
[0001] Pliego Cartulina Blanca - Cant: 0 - Precio: $ 3.000,00
[0002] Lápiz Mirado 2 - Cant: 1 - Precio: $ 1.000,00
[0003] Borrador - Cant: 5 - Precio: $ 500,00
```