

Documentación Completa de la API de Gestión de Inventario

4 de julio de 2025

Índice

1. Introducción	3
2. Categorías	3
2.1. Obtener Resumen de Categorías	3
2.2. Fusionar Categorías	3
2.3. Obtener Todas las Categorías	4
2.4. Crear Categoría	4
2.5. Obtener Categoría por ID	4
2.6. Asignar Producto a Categoría	5
2.7. Obtener Conteo de Productos por Categoría	5
2.8. Archivar Categoría	5
2.9. Actualizar Categoría	6
2.10. Eliminar Categoría	6
3. Productos	6
3.1. Obtener Productos con Stock Bajo	6
3.2. Actualizar Precios en Masa	7
3.3. Obtener Todos los Productos	7
3.4. Crear Producto	7
3.5. Obtener Producto por ID	8
3.6. Actualizar Stock de Producto	8
3.7. Archivar Producto	8
3.8. Actualizar Producto	9
3.9. Eliminar Producto	9
4. Clientes	9
4.1. Obtener Clientes con Más Ventas	9
4.2. Obtener Todos los Clientes	10
4.3. Crear Cliente	10
4.4. Obtener Cliente por ID	10
4.5. Obtener Historial de Compras de Cliente	11
4.6. Asignar Venta a Cliente	11
4.7. Actualizar Crédito de Cliente	11
4.8. Desactivar Cliente	12

4.9. Actualizar Cliente	12
4.10. Eliminar Cliente	12
5. Catálogos	13
5.1. Obtener Todos los Catálogos	13
5.2. Crear Catálogo	13
5.3. Obtener Catálogo por ID	13
5.4. Obtener Productos de un Catálogo	14
5.5. Asignar Producto a Catálogo	14
5.6. Eliminar Producto de Catálogo	14
5.7. Duplicar Catálogo	15
5.8. Actualizar Estado de Catálogo	15
5.9. Actualizar Catálogo	15
5.10. Eliminar Catálogo	16
6. Notificaciones	16
6.1. Obtener Todas las Notificaciones	16
6.2. Crear Notificación por Stock Bajo	16
6.3. Crear Notificación	17
6.4. Obtener Notificación por ID	17
6.5. Marcar Notificación como Leída	17
6.6. Actualizar Notificación	18
6.7. Eliminar Notificación	18
7. Proveedores	18
7.1. Obtener Resumen de Proveedores	18
7.2. Obtener Todos los Proveedores	19
7.3. Crear Proveedor	19
7.4. Obtener Proveedor por ID	19
7.5. Obtener Conteo de Productos por Proveedor	19
7.6. Asignar Catálogo a Proveedor	20
7.7. Actualizar Estado de Proveedor	20
7.8. Actualizar Proveedor	20
7.9. Eliminar Proveedor	21
8. Ventas	21
8.1. Obtener Todas las Ventas	21
8.2. Crear Venta con Validación	21
8.3. Obtener Ventas por Rango de Fechas	21
8.4. Crear Venta	22
8.5. Obtener Venta por ID	22
8.6. Obtener Ganancia de Venta	22
8.7. Aplicar Descuento a Venta	22
8.8. Actualizar Venta	22
8.9. Cancelar Venta	22
8.10. Eliminar Venta	23

1. Introducción

Esta documentación describe todas las URIs de la API de gestión de inventario, desarrollada con Node.js, Express y MongoDB. La API permite gestionar categorías, productos, clientes, catálogos, notificaciones, proveedores y ventas. Cada endpoint se detalla con su método HTTP, parámetros, respuestas y ejemplos de uso.

2. Categorías

2.1. Obtener Resumen de Categorías

Método y URI: GET /api/categories/summary

Descripción: Devuelve un resumen de todas las categorías con el conteo de productos por categoría.

Parámetros: Ninguno

Respuestas: ■ 200: Resumen de categorías ([categoryId, name, productCount])

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/categories/summary

Ejemplo de Respuesta: [

```
  {"categoryId": "123", "name": "Electrónica", "productCount": 10},
  {"categoryId": "456", "name": "Ropa", "productCount": 5}
```

]

2.2. Fusionar Categorías

Método y URI: POST /api/categories/merge

Descripción: Fusiona dos categorías, moviendo los productos de la categoría fuente a la categoría destino y eliminando la fuente.

Parámetros (Cuerpo): {sourceCategoryId: String, targetCategoryId: String}

Respuestas: ■ 200: Categoría destino actualizada

■ 404: Categoría no encontrada

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/categories/merge \

```
-H "Content-Type: application/json" \
```

```
-d '{"sourceCategoryId": "123", "targetCategoryId": "456"}'
```

Ejemplo de Respuesta: 200: {categoryId: "456", name: Ropa}

- 404: {error: Categoría no encontrada}

2.3. Obtener Todas las Categorías

Método y URI: GET /api/categories

Descripción: Devuelve todas las categorías.

Parámetros: Ninguno

Respuestas: ■ 200: Lista de categorías ([name])

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/categories

Ejemplo de Respuesta: [{"_id": "123", "name": "Electrónica"}, {"_id": "456", "name": "Electrónica"}]

2.4. Crear Categoría

Método y URI: POST /api/categories

Descripción: Crea una nueva categoría.

Parámetros (Cuerpo): {name: String}

Respuestas: ■ 201: Categoría creada

 ■ 400: Datos inválidos

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/categories \\\n -H "Content-Type: application/json" \\\n -d '{"name": "Electrónica"}'

Ejemplo de Respuesta: {id : "123", name : "Electrónica"}

2.5. Obtener Categoría por ID

Método y URI: GET /api/categories/:id

Descripción: Devuelve una categoría específica.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: ■ 200: Categoría encontrada

 ■ 404: Categoría no encontrada

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/categories/123

Ejemplo de Respuesta: {id : "123", name : "Electrónica"}

2.6. Asignar Producto a Categoría

Método y URI: POST /api/categories/:id/assign-product

Descripción: Asigna un producto a una categoría.

Parámetros (Ruta): id: String (ObjectId de categoría)

Parámetros (Cuerpo): {productId: String}

Respuestas:

- 200: Producto asignado
- 404: Categoría o producto no encontrado

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/categories/123/assign-product \\\n-H "Content-Type: application/json" \\\n-d '{"productId": "789"}'

Ejemplo de Respuesta: {id: "789", name: "Producto1", categoryId: "123"}

2.7. Obtener Conteo de Productos por Categoría

Método y URI: GET /api/categories/:id/product-count

Descripción: Devuelve el número de productos en una categoría.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Conteo de productos
- 404: Categoría no encontrada

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/categories/123/product-count

Ejemplo de Respuesta: {categoryId: "123", productCount: 10}

2.8. Archivar Categoría

Método y URI: PUT /api/categories/:id/archive

Descripción: Cambia el estado de una categoría a "archived".

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Categoría archivada
- 404: Categoría no encontrada

Ejemplo de Solicitud: curl -X PUT http://localhost:8000/api/categories/123/archive

Ejemplo de Respuesta: {id: "123", name: "Electrónica", status: "archived"}

2.9. Actualizar Categoría

Método y URI: PUT /api/categories/:id

Descripción: Actualiza los datos de una categoría.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {name: String}

Respuestas:

- 200: Categoría actualizada
- 404: Categoría no encontrada

Ejemplo de Solicitud:

```
curl -X PUT http://localhost:8000/api/categories/123 \
-H "Content-Type: application/json" \
-d '{"name": "Electrónica Actualizada"}'
```

Ejemplo de Respuesta: `{id: "123", name: "Electrónica Actualizada"}`

2.10. Eliminar Categoría

Método y URI: DELETE /api/categories/:id

Descripción: Elimina una categoría.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 204: Categoría eliminada
- 404: Categoría no encontrada

Ejemplo de Solicitud: `curl -X DELETE http://localhost:8000/api/categories/123`

Ejemplo de Respuesta: (Sin contenido)

3. Productos

3.1. Obtener Productos con Stock Bajo

Método y URI: GET /api/products/low-stock

Descripción: Devuelve productos con stock menor o igual a un umbral.

Parámetros (Query): threshold: Number (opcional, por defecto 10)

Respuestas:

- 200: Lista de productos con stock bajo

Ejemplo de Solicitud: `curl -X GET http://localhost:8000/api/products/low-stock?threshold=10`

Ejemplo de Respuesta:

```
[
  { "_id": "789", "name": "Producto 1", "stock": 3, "categoryId": {...} },
  { "_id": "012", "name": "Producto 2", "stock": 5, "categoryId": {...} }
]
```

3.2. Actualizar Precios en Masa

Método y URI: POST /api/products/bulk-price-update

Descripción: Aumenta los precios de productos en una categoría por un porcentaje.

Parámetros (Cuerpo): {categoryId: String, percentage: Number}

Respuestas: ■ 200: Número de productos modificados

- 400: Porcentaje inválido
- 404: Categoría no encontrada

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/products/bulk-price-update \\\n -H "Content-Type: application/json" \\\n -d '{"categoryId": "123", "percentage": 10}'

Ejemplo de Respuesta: {modifiedCount: 5}

3.3. Obtener Todos los Productos

Método y URI: GET /api/products

Descripción: Devuelve todos los productos.

Parámetros: Ninguno

Respuestas: ■ 200: Lista de productos

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/products

Ejemplo de Respuesta: [{"_id": "789", "name": "Producto 1", "categoryId": {...},

3.4. Crear Producto

Método y URI: POST /api/products

Descripción: Crea un nuevo producto.

Parámetros (Cuerpo): {name: String, categoryId: String, purchaseUnit: String, quantityIncluded: Number, saleUnit: String, purchasePrice: Number, salePrice: Number, stock: Number}

Respuestas: ■ 201: Producto creado

- 400: Datos inválidos

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/products \\\n -H "Content-Type: application/json" \\\n -d '{"name": "Producto 1", "categoryId": "123", "purchaseUnit": "kg", "quantityIncluded": 10, "saleUnit": "kg", "purchasePrice": 10, "salePrice": 12, "stock": 10}'

Ejemplo de Respuesta: {id: "789", name: "Producto1", ...}

3.5. Obtener Producto por ID

Método y URI: GET /api/products/:id

Descripción: Devuelve un producto específico.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Producto encontrado
- 404: Producto no encontrado

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/products/789

Ejemplo de Respuesta: `{id : "789", name : "Producto1", ...}`

3.6. Actualizar Stock de Producto

Método y URI: PUT /api/products/:id/stock

Descripción: Actualiza el stock de un producto.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {stock: Number}

Respuestas:

- 200: Producto actualizado
- 400: Stock negativo
- 404: Producto no encontrado

Ejemplo de Solicitud: curl -X PUT http://localhost:8000/api/products/789/stock \\\n-H "Content-Type: application/json" \\\n-d '{"stock": 50}'

Ejemplo de Respuesta: `{id : "789", name : "Producto1", stock : 50, ...}`

3.7. Archivar Producto

Método y URI: PUT /api/products/:id/archive

Descripción: Cambia el estado de un producto a "archived".

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Producto archivado
- 404: Producto no encontrado

Ejemplo de Solicitud: curl -X PUT http://localhost:8000/api/products/789/archive

Ejemplo de Respuesta: `{id : "789", name : "Producto1", status : "archived", ...}`

3.8. Actualizar Producto

Método y URI: PUT /api/products/:id

Descripción: Actualiza los datos de un producto.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {name: String, categoryId: String, purchaseUnit: String, quantityIncluded: Number, saleUnit: String, purchasePrice: Number, salePrice: Number, stock: Number}

Respuestas: ■ 200: Producto actualizado

 ■ 404: Producto no encontrado

Ejemplo de Solicitud: curl -X PUT http://localhost:8000/api/products/789 \\\n -H "Content-Type: application/json" \\\n -d '{"name": "Producto 1 Actualizado", "stock": 75}'

Ejemplo de Respuesta: {*id* : "789", *name* : "Producto1Actualizado", *stock* : 75, ...}

3.9. Eliminar Producto

Método y URI: DELETE /api/products/:id

Descripción: Elimina un producto.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: ■ 204: Producto eliminado

 ■ 404: Producto no encontrado

Ejemplo de Solicitud: curl -X DELETE http://localhost:8000/api/products/789

Ejemplo de Respuesta: (Sin contenido)

4. Clientes

4.1. Obtener Clientes con Más Ventas

Método y URI: GET /api/clients/top-sales

Descripción: Devuelve los 5 clientes con más ventas.

Parámetros: Ninguno

Respuestas: ■ 200: Lista de clientes con conteo de ventas

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/clients/top-sales

Ejemplo de Respuesta:

```
[
  {"clientId": "111", "clientName": "Juan Pérez", "totalSales": 10},
  {"clientId": "222", "clientName": "María López", "totalSales": 8}
]
```

4.2. Obtener Todos los Clientes

Método y URI: GET /api/clients

Descripción: Devuelve todos los clientes.

Parámetros: Ninguno

Respuestas: ■ 200: Lista de clientes

Ejemplo de Solicitud: `curl -X GET http://localhost:8000/api/clients`

Ejemplo de Respuesta:

```
[{"_id": "111", "fullname": "Juan Pérez", "taxid": "123456"}]
```

4.3. Crear Cliente

Método y URI: POST /api/clients

Descripción: Crea un nuevo cliente.

Parámetros (Cuerpo): {taxid: String, fullname: String, address: String, references: String, phone: String, email: String}

Respuestas: ■ 201: Cliente creado
■ 400: Datos inválidos

Ejemplo de Solicitud: `curl -X POST http://localhost:8000/api/clients \
-H "Content-Type: application/json" \
-d '{"taxid": "123456", "fullname": "Juan Pérez", "address": "Calle 1", "ref"`

Ejemplo de Respuesta:

```
{_id: "111", fullname: "JuanPrez", ...}
```

4.4. Obtener Cliente por ID

Método y URI: GET /api/clients/:id

Descripción: Devuelve un cliente específico.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: ■ 200: Cliente encontrado
■ 404: Cliente no encontrado

Ejemplo de Solicitud: `curl -X GET http://localhost:8000/api/clients/111`

Ejemplo de Respuesta:

```
{_id: "111", fullname: "JuanPrez", ...}
```

4.5. Obtener Historial de Compras de Cliente

Método y URI: GET /api/clients/:id/purchase-history

Descripción: Devuelve las ventas asociadas a un cliente.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Lista de ventas
- 404: No se encontraron compras

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/clients/111/purchase-history

Ejemplo de Respuesta: [{"_id": "333", "clientId": "111", "items": [...], "total": ...}]

4.6. Asignar Venta a Cliente

Método y URI: POST /api/clients/:id/assign-sale

Descripción: Asigna una venta a un cliente.

Parámetros (Ruta): id: String (ObjectId de cliente)

Parámetros (Cuerpo): {saleId: String}

Respuestas:

- 200: Venta asignada
- 404: Cliente o venta no encontrada

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/clients/111/assign-sale \\\n-H "Content-Type: application/json" \\\n-d '{"saleId": "333"}'

Ejemplo de Respuesta: {id: "333", clientId: "111", ...}

4.7. Actualizar Crédito de Cliente

Método y URI: PUT /api/clients/:id/credit

Descripción: Actualiza el crédito de un cliente.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {credit: Number}

Respuestas:

- 200: Cliente actualizado
- 400: Crédito negativo
- 404: Cliente no encontrado

Ejemplo de Solicitud: curl -X PUT http://localhost:8000/api/clients/111/credit \\\n-H "Content-Type: application/json" \\\n-d '{"credit": 1000}'

Ejemplo de Respuesta: {id: "111", fullname: "JuanPerez", credit: 1000, ...}

4.8. Desactivar Cliente

Método y URI: PUT /api/clients/:id/deactivate

Descripción: Cambia el estado de un cliente a inactive".

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Cliente desactivado
- 404: Cliente no encontrado

Ejemplo de Solicitud: curl -X PUT http://localhost:8000/api/clients/111/deactivate

Ejemplo de Respuesta: `{id : "111", fullname : "JuanPrez", status : "inactive", ...}`

4.9. Actualizar Cliente

Método y URI: PUT /api/clients/:id

Descripción: Actualiza los datos de un cliente.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {taxid: String, fullname: String, address: String, references: String, phone: String, email: String}

Respuestas:

- 200: Cliente actualizado
- 404: Cliente no encontrado

Ejemplo de Solicitud:

```
curl -X PUT http://localhost:8000/api/clients/111 \
-H "Content-Type: application/json" \
-d '{"fullname": "Juan Pérez Actualizado"}'
```

Ejemplo de Respuesta: `{id : "111", fullname : "JuanPrezActualizado", ...}`

4.10. Eliminar Cliente

Método y URI: DELETE /api/clients/:id

Descripción: Elimina un cliente.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 204: Cliente eliminado
- 404: Cliente no encontrado

Ejemplo de Solicitud: curl -X DELETE http://localhost:8000/api/clients/111

Ejemplo de Respuesta: (Sin contenido)

5. Catálogos

5.1. Obtener Todos los Catálogos

Método y URI: GET /api/catalogs

Descripción: Devuelve todos los catálogos.

Parámetros: Ninguno

Respuestas: ■ 200: Lista de catálogos

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/catalogs

Ejemplo de Respuesta: [{"_id": "444", "name": "Catálogo 1", "products": [...], .

5.2. Crear Catálogo

Método y URI: POST /api/catalogs

Descripción: Crea un nuevo catálogo.

ellipted for brevity...

Parámetros (Cuerpo): {name: String, filePath: String}

Respuestas: ■ 201: Catálogo creado

■ 400: Datos inválidos

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/catalogs \\\n -H "Content-Type: application/json" \\\n -d '{"name": "Catálogo 1", "filePath": "/path/to/file"}'

Ejemplo de Respuesta: {*i*d : "444", name : "Catlogo1", filePath : "/path/to/file", ...}

5.3. Obtener Catálogo por ID

Método y URI: GET /api/catalogs/:id

Descripción: Devuelve un catálogo específico.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: ■ 200: Catálogo encontrado

■ 404: Catálogo no encontrado

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/catalogs/444

Ejemplo de Respuesta: {*i*d : "444", name : "Catlogo1", ...}

5.4. Obtener Productos de un Catálogo

Método y URI: GET /api/catalogs/:id/products

Descripción: Devuelve los productos asignados a un catálogo.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Lista de productos
- 404: Catálogo no encontrado

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/catalogs/444/products

Ejemplo de Respuesta: [{"_id": "789", "name": "Producto 1", ...}]

5.5. Asignar Producto a Catálogo

Método y URI: POST /api/catalogs/:id/assign-product

Descripción: Asigna un producto a un catálogo.

Parámetros (Ruta): id: String (ObjectId de catálogo)

Parámetros (Cuerpo): {productId: String}

Respuestas:

- 200: Catálogo actualizado
- 404: Catálogo o producto no encontrado

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/catalogs/444/assign-product \\\n-H "Content-Type: application/json" \\\n-d '{"productId": "789"}'

Ejemplo de Respuesta: {id: "444", name: "Catlogo1", products: ["789"], ...}

5.6. Eliminar Producto de Catálogo

Método y URI: POST /api/catalogs/:id/remove-product

Descripción: Elimina un producto de un catálogo.

Parámetros (Ruta): id: String (ObjectId de catálogo)

Parámetros (Cuerpo): {productId: String}

Respuestas:

- 200: Catálogo actualizado
- 404: Catálogo no encontrado

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/catalogs/444/remove-product \\\n-H "Content-Type: application/json" \\\n-d '{"productId": "789"}'

Ejemplo de Respuesta: {id: "444", name: "Catlogo1", products: [], ...}

5.7. Duplicar Catálogo

Método y URI: POST /api/catalogs/:id/duplicate

Descripción: Crea una copia de un catálogo.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 201: Catálogo duplicado
- 404: Catálogo no encontrado

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/catalogs/444/duplicate

Ejemplo de Respuesta: `{id: "555", name: "Catlogo1(Copy)", ...}`

5.8. Actualizar Estado de Catálogo

Método y URI: PUT /api/catalogs/:id/status

Descripción: Actualiza el estado de un catálogo (active/inactive).

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {status: String (.active.º inactive')}

Respuestas:

- 200: Catálogo actualizado
- 400: Estado inválido
- 404: Catálogo no encontrado

Ejemplo de Solicitud:

```
curl -X PUT http://localhost:8000/api/catalogs/444/status \
-H "Content-Type: application/json" \
-d '{"status": "inactive"}
```

Ejemplo de Respuesta: `{id: "444", name: "Catlogo1", status: "inactive", ...}`

5.9. Actualizar Catálogo

Método y URI: PUT /api/catalogs/:id

Descripción: Actualiza los datos de un catálogo.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {name: String, filePath: String}

Respuestas:

- 200: Catálogo actualizado
- 404: Catálogo no encontrado

Ejemplo de Solicitud:

```
curl -X PUT http://localhost:8000/api/catalogs/444 \
-H "Content-Type: application/json" \
-d '{"name": "Catálogo 1 Actualizado"}
```

Ejemplo de Respuesta: `{id: "444", name: "Catlogo1Actualizado", ...}`

5.10. Eliminar Catálogo

Método y URI: DELETE /api/catalogs/:id

Descripción: Elimina un catálogo.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 204: Catálogo eliminado
- 404: Catálogo no encontrado

Ejemplo de Solicitud: curl -X DELETE http://localhost:8000/api/catalogs/444

Ejemplo de Respuesta: (Sin contenido)

6. Notificaciones

6.1. Obtener Todas las Notificaciones

Método y URI: GET /api/notifications

Descripción: Devuelve todas las notificaciones.

Parámetros: Ninguno

Respuestas:

- 200: Lista de notificaciones

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/notifications

Ejemplo de Respuesta: [{"_id": "666", "message": "Stock bajo", "icon": "warning"}]

6.2. Crear Notificación por Stock Bajo

Método y URI: POST /api/notifications/low-stock

Descripción: Crea notificaciones para productos con stock bajo.

Parámetros (Cuerpo): {threshold: Number (opcional, por defecto 10)}

Respuestas:

- 201: Notificaciones creadas

Ejemplo de Solicitud:

```
curl -X POST http://localhost:8000/api/notifications/low-stock \
-H "Content-Type: application/json" \
-d '{"threshold": 5}'
```

Ejemplo de Respuesta: [{"_id": "666", "message": "El producto Producto 1 tiene un inventario bajo"}]

6.3. Crear Notificación

Método y URI: POST /api/notifications

Descripción: Crea una nueva notificación.

Parámetros (Cuerpo): {icon: String, message: String, read: Boolean (opcional, por defecto false)}

Respuestas:

- 201: Notificación creada
- 400: Datos inválidos

Ejemplo de Solicitud: `curl -X POST http://localhost:8000/api/notifications \`
`-H "Content-Type: application/json" \`
`-d '{"icon": "warning", "message": "Notificación personalizada"}'`

Ejemplo de Respuesta: {*id* : "666", *icon* : "warning", *message* : "Notificación personalizada", *read* : false, ...}

6.4. Obtener Notificación por ID

Método y URI: GET /api/notifications/:id

Descripción: Devuelve una notificación específica.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Notificación encontrada
- 404: Notificación no encontrada

Ejemplo de Solicitud: `curl -X GET http://localhost:8000/api/notifications/666`

Ejemplo de Respuesta: {*id* : "666", *message* : "Stockbajo", ...}

6.5. Marcar Notificación como Leída

Método y URI: PUT /api/notifications/:id/read

Descripción: Marca una notificación como leída.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 200: Notificación actualizada
- 404: Notificación no encontrada

Ejemplo de Solicitud: `curl -X PUT http://localhost:8000/api/notifications/666/read`

Ejemplo de Respuesta: {*id* : "666", *message* : "Stockbajo", *read* : true, ...}

6.6. Actualizar Notificación

Método y URI: PUT /api/notifications/:id

Descripción: Actualiza los datos de una notificación.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {icon: String, message: String, read: Boolean}

Respuestas:

- 200: Notificación actualizada
- 404: Notificación no encontrada

Ejemplo de Solicitud:

```
curl -X PUT http://localhost:8000/api/notifications/666 \
-H "Content-Type: application/json" \
-d '{"message": "Notificación actualizada"}'
```

Ejemplo de Respuesta: `{id: "666", message: "Notificación actualizada", ...}`

6.7. Eliminar Notificación

Método y URI: DELETE /api/notifications/:id

Descripción: Elimina una notificación.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 204: Notificación eliminada
- 404: Notificación no encontrada

Ejemplo de Solicitud:

```
curl -X DELETE http://localhost:8000/api/notifications/666
```

Ejemplo de Respuesta: (Sin contenido)

7. Proveedores

7.1. Obtener Resumen de Proveedores

Método y URI: GET /api/suppliers/summary

Descripción: Devuelve un resumen de proveedores con el conteo de productos por catálogo.

Parámetros: Ninguno

Respuestas:

- 200: Resumen de proveedores

/Ejemplo de Solicitud:]

```
curl -X GET http://localhost:8000/api/suppliers/summary
```

Ejemplo de Respuesta:

```
[
  {"supplierId": "777", "name": "Proveedor 1", "productCount": 10},
  {"supplierId": "888", "name": "Proveedor 2", "productCount": 5}
]
```

7.2. Obtener Todos los Proveedores

Método y URI: GET /api/suppliers

Descripción: Devuelve todos los proveedores.

Parámetros: Ninguno

Respuestas: 7.3. Crear Proveedor

Método y URI: POST /api/suppliers

Descripción: Crea un nuevo proveedor.

Parámetros (Cuerpo): {idNumber: String, company: String, contactName: String, phone: String, bankAccount: String, bankName: String, catalogId: String}

Respuestas: ■ 201: Proveedor creado

■ 400: Datos inválidos

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/suppliers \\\n-H "Content-Type: application/json" \\\n-d '{"idNumber": "123456", "company": "Proveedor 1", "contactName": "Ana

Ejemplo de Respuesta: {id: "777", company: "Proveedor1", ...}

7.4. Obtener Proveedor por ID

Método y URI: GET /api/suppliers/:id

Descripción: Devuelve un proveedor específico.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: ■ 200: Proveedor encontrado

■ 404: Proveedor no encontrado

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/suppliers/777

Ejemplo de Respuesta: {id: "777", company: "Proveedor1", ...}

7.5. Obtener Conteo de Productos por Proveedor

Método y URI: GET /api/suppliers/:id/product-count

Descripción: Devuelve el número de productos en el catálogo de un proveedor.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: ■ 200: Conteo de productos

■ 404: Proveedor no encontrado

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/suppliers/777/product

Ejemplo de Respuesta: {supplierId: "777", productCount: 10}

7.6. Asignar Catálogo a Proveedor

Método y URI: POST /api/suppliers/:id/assign-catalog

Descripción: Asigna un catálogo a un proveedor.

Parámetros (Ruta): id: String (ObjectId de proveedor)

Parámetros (Cuerpo): {catalogId: String}

Respuestas:

- 200: Proveedor actualizado
- 404: Proveedor o catálogo no encontrado

Ejemplo de Solicitud:

```
curl -X POST http://localhost:8000/api/suppliers/777/as
-H "Content-Type: application/json" \
-d '{"catalogId": "444"}'
```

Ejemplo de Respuesta: {*d*: "777", *company*: "Proveedor1", *catalogId*: "444", ...}

7.7. Actualizar Estado de Proveedor

Método y URI: PUT /api/suppliers/:id/status

Descripción: Actualiza el estado de un proveedor (active/inactive).

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {status: String (.active.º inactive')}

Respuestas:

- 200: Proveedor actualizado
- 400: Estado inválido
- 404: Proveedor no encontrado

Ejemplo de Solicitud:

```
curl -X PUT http://localhost:8000/api/suppliers/777/sta
-H "Content-Type: application/json" \
-d '{"status": "inactive"}'
```

Ejemplo de Respuesta: {*d*: "777", *company*: "Proveedor1", *status*: "inactive", ...}

7.8. Actualizar Proveedor

Método y URI: PUT /api/suppliers/:id

Descripción: Actualiza los datos de un proveedor.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {idNumber: String, company: String, contactName: String, phone: String, bankAccount: String, bankName: String, catalogId: String}

Respuestas:

- 200: Proveedor actualizado
- 404: Proveedor no encontrado

Ejemplo de Solicitud:

```
curl -X PUT http://localhost:8000/api/suppliers/777 \
-H "Content-Type: application/json" \
-d '{"company": "Proveedor 1 Actualizado"}'
```

Ejemplo de Respuesta: {*d*: "777", *company*: "Proveedor1Actualizado", ...}

7.9. Eliminar Proveedor

Método y URI: DELETE /api/suppliers/:id

Descripción: Elimina un proveedor.

Parámetros (Ruta): id: String (ObjectId)

Respuestas:

- 204: Proveedor eliminado
- 404: Proveedor no encontrado

Ejemplo de Solicitud: curl -X DELETE http://localhost:8000/api/suppliers/777

Ejemplo de Respuesta: (Sin contenido)

8. Ventas

8.1. Obtener Todas las Ventas

Método y URI: GET /api/sales

Descripción: Devuelve todas las ventas.

Parámetros: Ninguno

Respuestas:

- 200: Lista de ventas

Ejemplo de Solicitud: curl -X GET http://localhost:8000/api/sales

Ejemplo de Respuesta: [{"_id": "333", "clientId": {...}, "items": [...], "total": ...}]

8.2. Crear Venta con Validación

Método y URI: POST /api/sales/validated

Descripción: Crea una venta con validación de stock y cliente.

Parámetros (Cuerpo): {clientId: String, items: [{productId: String, name: String, quantity: Number}]}

8.3. Obtener Ventas por Rango de Fechas

Método y URI: POST /api/sales/date-range

Descripción: Devuelve ventas en un rango de fechas con totales.

Parámetros (Cuerpo): {startDate: String (ISO8601), endDate: String (ISO8601)}

Respuestas:

- 200: Lista de ventas con totales

Ejemplo de Solicitud: curl -X POST http://localhost:8000/api/sales/date-range -H "Content-Type: application/json" \ -d '{"startDate": "2025-01-01", "endDate": "2025-07-04"}'

Ejemplo de Respuesta:

```
{
  "totalSales": 2,
  "totalRevenue": 200,
  "sales": [{"_id": "333", "total": 100, ...}, ...]
}
```

8.4. Crear Venta

Método y URI: POST /api/sales

Descripción: Crea una nueva venta sin validación de stock.

Parámetros (Cuerpo): {clientId: String, items: [{productId: String, name: String, price: Number}]}

Respuestas: 8.5. Obtener Venta por ID

Método y URI: GET /api/sales/:id

Descripción: Devuelve una venta específica.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: 8.6. Obtener Ganancia de Venta

Método y URI: GET /api/sales/:id/profit

Descripción: Calcula la ganancia de una venta.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: 8.7. Aplicar Descuento a Venta

Método y URI: PUT /api/sales/:id/discount

Descripción: Aplica un descuento porcentual a una venta.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {discount: Number (0-100)}

Respuestas: 8.8. Actualizar Venta

Método y URI: PUT /api/sales/:id

Descripción: Actualiza los datos de una venta.

Parámetros (Ruta): id: String (ObjectId)

Parámetros (Cuerpo): {clientId: String, items: [{productId: String, name: String, price: Number}]}

Respuestas: 8.9. Cancelar Venta

Método y URI: DELETE /api/sales/:id/cancel

Descripción: Cancela una venta y restaura el stock de los productos.

Parámetros (Ruta): id: String (ObjectId)

Respuestas: 8.10. Eliminar Venta**Método y URI:** DELETE /api/sales/:id**Descripción:** Elimina una venta sin restaurar stock.**Parámetros (Ruta):** id: String (ObjectId)**Respuestas:**