

Endpoints Lógicos para el API de Smart Economart

Basado en el diagrama ER del sistema "Smart Economart" (una app para gestionar inventario, pedidos, recetas y control de calidad en una escuela de cocina), esta es una propuesta de endpoints RESTful. Asumo un API basado en HTTP, y que cada entidad soporta operaciones CRUD básicas (Create, Read, Update, Delete).

He extendido los CRUD con endpoints lógicos derivados de las relaciones y necesidades del cliente:

- **Necesidades clave:** Generación de pedidos, registro de usos, gestión de proveedores, inventario y reportes. Enfoque en trazabilidad (e.g., stock, temperaturas, desperdicios), optimización de costos (presupuesto limitado), y usabilidad educativa (e.g., alertas, reportes por clase). Endpoints incluyen filtros, paginación (`?page=1&limit=10`), búsqueda (`?search=term`) y agregaciones para reportes.

Los endpoints usan convención REST:

- **GET:** Listar/leer (con filtros).
- **POST:** Crear.
- **PUT/PATCH:** Actualizar (PUT completo, PATCH parcial).
- **DELETE:** Eliminar.

Agrupé por entidad para claridad, con ejemplos de paths base como `/api/v1/`.

1. USUARIOS (Gestión de usuarios: autenticación, roles)

- GET /usuarios: Listar usuarios (filtro por rol: `?rol=profesor`).
- GET /usuarios/{id}: Detalle de usuario.
- POST /usuarios: Crear usuario.
- PUT /usuarios/{id}: Actualizar usuario.
- DELETE /usuarios/{id}: Eliminar usuario.
- **Lógicos adicionales:**
 - POST /auth/login: Autenticación (devuelve token).
 - GET /usuarios/me: Perfil actual.

2. PROVEEDORES (Gestión de proveedores externos para compras)

- GET /proveedores: Listar proveedores (filtro por calificación: `?calificacion>4`).
- GET /proveedores/{id}: Detalle de proveedor.
- POST /proveedores: Crear proveedor.
- PUT /proveedores/{id}: Actualizar proveedor.
- DELETE /proveedores/{id}: Eliminar proveedor.
- **Lógicos adicionales:**
 - GET /proveedores/reportes/rendimiento: Reporte de calificaciones/tiempos de entrega.

3. ALBARANES (Registro de recepciones de productos)

- GET /albaranes: Listar albaranes (filtro por fecha/proveedor: `?fecha=2025-09-01&proveedor=1`).
- GET /albaranes/{id}: Detalle de albarán (incluye detalles).

- POST /albaranes: Crear albarán.
- PUT /albaranes/{id}: Actualizar albarán (e.g., marcar concordancia).
- DELETE /albaranes/{id}: Eliminar albarán.
- **Lógicos adicionales:**
 - GET /albaranes/{id}/detalles: Sub-endpoint para detalles.

4. DETALLES_ALBARANES (Detalles por ítem en recepciones; no CRUD directo, sino vía albarán)

- POST /albaranes/{id}/detalles: Agregar detalle a albarán.
- PUT /detalles-albaranes/{id}: Actualizar detalle (e.g., cantidad recibida).
- DELETE /detalles-albaranes/{id}: Eliminar detalle.
- **Lógicos adicionales:**
 - GET /albaranes/{id}/detalles: Listar detalles de un albarán.

5. IMPUESTOS (Tipos de impuestos; entidad estática)

- GET /impuestos: Listar impuestos.
- POST /impuestos: Crear impuesto.
- PUT /impuestos/{id}: Actualizar porcentaje.
- DELETE /impuestos/{id}: Eliminar.
- **Lógicos adicionales:**
 - Ninguno.

6. UNIDADES (Unidades de medida; estática)

- GET /unidades: Listar unidades (e.g., kg, unidades).
- POST /unidades: Crear unidad.
- PUT /unidades/{id}: Actualizar.
- DELETE /unidades/{id}: Eliminar.
- **Lógicos adicionales:**
 - Ninguno.

7. PEDIDOS_INTERNO (Pedidos internos)

- GET /pedidos-interno: Listar pedidos (filtro por estado/usuario: ?estado=pendiente&usuario=1).
- GET /pedidos-interno/{id}: Detalle de pedido (incluye detalles).
- POST /pedidos-interno: Crear pedido.
- PUT /pedidos-interno/{id}: Actualizar (e.g., cambiar estado).
- DELETE /pedidos-interno/{id}: Cancelar pedido.
- **Lógicos adicionales:**
 - GET /pedidos-interno/consolidados: Lista consolidada por día.

8. DETALLES_PEDIDOS_INTERNO (Detalles de pedidos; vía pedido)

- POST /pedidos-interno/{id}/detalles: Agregar ítem a pedido.
- PUT /detalles-pedidos-interno/{id}: Actualizar cantidad.
- DELETE /detalles-pedidos-interno/{id}: Eliminar ítem.

- **Lógicos adicionales:**
 - GET /pedidos-interno/{id}/detalles: Listar ítems.

9. USOS_SOBRANTES (Registro de uso y sobrantes post-pedido)

- GET /usos-sobrantes: Listar usos (filtro por pedido: ?pedido=1).
- GET /usos-sobrantes/{id}: Detalle.
- POST /usos-sobrantes: Crear registro.
- PUT /usos-sobrantes/{id}: Actualizar cantidades.
- DELETE /usos-sobrantes/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /usos-sobrantes/reportes/desperdicio: Reporte de sobrantes por producto.

10. BAJAS (Registro de descartes)

- GET /bajas: Listar bajas (filtro por motivo/fecha).
- GET /bajas/{id}: Detalle.
- POST /bajas: Registrar baja.
- PUT /bajas/{id}: Actualizar.
- DELETE /bajas/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /bajas/reportes: Reporte de costos por motivo.

11. INVENTARIO (Stock actual)

- GET /inventario: Listar stock (filtro por producto/caducidad: ?caducidad<2025-10-01).
- GET /inventario/{id}: Detalle de lote.
- POST /inventario: Agregar entrada (vía albarán).
- PUT /inventario/{id}: Actualizar stock (e.g., ajustar cantidad).
- DELETE /inventario/{id}: Eliminar lote.
- **Lógicos adicionales:**
 - GET /inventario/alertas: Alertas de stock bajo/caducidad.

12. PRODUCTOS (Catálogo de productos)

- GET /productos: Listar productos (filtro por categoría/alergenicos: ?categoria=frutas).
- GET /productos/{id}: Detalle.
- POST /productos: Crear producto.
- PUT /productos/{id}: Actualizar (e.g., coste_base).
- DELETE /productos/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /productos/busqueda: Búsqueda avanzada (código_barras o nombre).

13. RENDIMIENTOS (Cálculos de rendimiento)

- GET /rendimientos: Listar (filtro por producto/fecha).
- GET /rendimientos/{id}: Detalle.
- POST /rendimientos: Registrar rendimiento.

- PUT /rendimientos/{id}: Actualizar.
- DELETE /rendimientos/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /rendimientos/promedio/{producto_id}: Promedio histórico.

14. RECETAS (Recetas educativas)

- GET /recetas: Listar (filtro por dificultad/nivel: `?dificultad=fácil`).
- GET /recetas/{id}: Detalle (incluye ingredientes).
- POST /recetas: Crear receta.
- PUT /recetas/{id}: Actualizar.
- DELETE /recetas/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /recetas/{id}/coste: Cálculo derivado de coste_por_porcion.

15. DETALLES_RECETAS (Ingredientes por receta; vía receta)

- POST /recetas/{id}/detalles: Agregar ingrediente.
- PUT /detalles-recetas/{id}: Actualizar cantidad.
- DELETE /detalles-recetas/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /recetas/{id}/detalles: Listar ingredientes.

16. LOG_TEMPERATURAS (Monitoreo de temperaturas)

- GET /log-temperaturas: Listar (filtro por inventario/fecha).
- GET /log-temperaturas/{id}: Detalle.
- POST /log-temperaturas: Registrar temperatura.
- PUT /log-temperaturas/{id}: Actualizar.
- DELETE /log-temperaturas/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /log-temperaturas/alertas: Alertas por umbrales excedidos.

17. MOVIMIENTOS (Historial de stock)

- GET /movimientos: Listar (filtro por tipo/fecha).
- GET /movimientos/{id}: Detalle.
- POST /movimientos: Registrar movimiento.
- PUT /movimientos/{id}: Actualizar.
- DELETE /movimientos/{id}: Eliminar.
- **Lógicos adicionales:**
 - GET /movimientos/auditoria/{producto_id}: Auditoría por producto.

Consideraciones Generales

- **Paginación/Filtros:** Todos los GET list usan `?page=&limit=&search=`.
- **Reportes Globales:** Añade `/reportes/inventario` (agregado de stock/costos), `/reportes/desperdicios` (de BAJAS/USOS).

- **Integraciones:** Endpoints como POST albarán actualizan INVENTARIO/MOVIMIENTOS automáticamente.