**🧍‍♂️ AUTH SERVICE**

**Puerto:** 3000  
**Ruta base:** /auth

**🔹 Propósito**

Gestionar la **autenticación y validación de usuarios**.  
Centraliza la emisión y validación de **tokens JWT**, que son usados por todos los demás servicios.

**🔸 Endpoints**

| **Método** | **Ruta** | **Descripción** |
| --- | --- | --- |
| **POST** | /auth/register | Crea un nuevo usuario en el sistema (guarda en authdb). |
| **POST** | /auth/login | Verifica credenciales y devuelve un token JWT válido. |
| **GET** | /auth/profile | Devuelve los datos del usuario autenticado según el token. |
| **GET** | /auth/validate | Endpoint interno que valida un token JWT (usado por otros servicios). |

**🔁 Comunicación**

* Los servicios como **Orders**, **Billing** o **Reports** llaman a /auth/validate para validar el token de autenticación de un usuario antes de ejecutar acciones seguras.
* Provee identidad y roles (por ejemplo, admin o cliente).

**⚙️ Variables relevantes**

AUTH\_PORT=3000

JWT\_SECRET=supersecreto

MONGO\_URI=mongodb://admin:admin123@mongodb:27017/authdb

**📦 PRODUCTS SERVICE**

**Puerto:** 3001  
**Ruta base:** /products

**🔹 Propósito**

Gestionar los **productos** del sistema: creación, edición, eliminación y consulta.  
Provee información base a otros servicios como Inventory o Orders.

**🔸 Endpoints**

| **Método** | **Ruta** | **Descripción** |
| --- | --- | --- |
| **GET** | /products | Lista todos los productos disponibles. |
| **GET** | /products/:id | Devuelve los detalles de un producto específico. |
| **POST** | /products | Crea un nuevo producto. |
| **PUT** | /products/:id | Actualiza los datos de un producto. |
| **DELETE** | /products/:id | Elimina un producto del catálogo. |

**🔁 Comunicación**

* **Inventory Service** llama a /products/:id para validar que el producto exista antes de registrar stock.
* Puede ser consultado por **Orders Service** para obtener información de productos al crear órdenes.

**⚙️ Variables relevantes**

PRODUCTS\_PORT=3001

MONGO\_URI=mongodb://admin:admin123@mongodb:27017/productsdb

**🏬 INVENTORY SERVICE**

**Puerto:** 3002  
**Ruta base:** /inventory

**🔹 Propósito**

Administrar los **niveles de stock** de cada producto.  
Permite registrar, actualizar y consultar inventarios.

**🔸 Endpoints**

| **Método** | **Ruta** | **Descripción** |
| --- | --- | --- |
| **GET** | /inventory | Lista todos los registros de inventario. |
| **GET** | /inventory/:productId | Devuelve el stock disponible de un producto. |
| **POST** | /inventory | Crea un nuevo registro de inventario (nuevo producto en stock). |
| **PUT** | /inventory/:productId | Actualiza la cantidad de stock existente. |

**🔁 Comunicación**

* **Inventory → Products:** valida la existencia del producto con /products/:id antes de registrar stock.
* **Orders → Inventory:** consulta stock disponible al crear una orden para verificar disponibilidad.

**⚙️ Variables relevantes**

INVENTORY\_PORT=3002

PRODUCTS\_SERVICE\_URL=http://products-service:3001

MONGO\_URI=mongodb://admin:admin123@mongodb:27017/inventorydb

**🧾 ORDERS SERVICE**

**Puerto:** 3003  
**Ruta base:** /orders

**🔹 Propósito**

Gestionar las **órdenes de compra** o pedidos realizados por los usuarios.  
Integra datos de usuario, productos y stock en un solo flujo.

**🔸 Endpoints**

| **Método** | **Ruta** | **Descripción** |
| --- | --- | --- |
| **GET** | /orders | Lista todas las órdenes registradas. |
| **GET** | /orders/:id | Devuelve los detalles de una orden. |
| **POST** | /orders | Crea una nueva orden (requiere autenticación y stock disponible). |
| **PUT** | /orders/:id/status | Actualiza el estado de la orden (por ejemplo: pendiente, pagada, enviada). |

**🔁 Flujo interno**

1. **Valida token JWT** con Auth Service (/auth/validate).
2. **Verifica stock** con Inventory Service (/inventory/:productId).
3. **Registra la orden** en la base de datos local (ordersdb).

**🔁 Comunicación**

* **Orders → Auth:** valida la autenticación del usuario.
* **Orders → Inventory:** consulta disponibilidad del producto.
* **Billing → Orders:** usa la información de las órdenes para generar facturas.

**⚙️ Variables relevantes**

ORDERS\_PORT=3003

AUTH\_SERVICE\_URL=http://auth-service:3000

INVENTORY\_SERVICE\_URL=http://inventory-service:3002

MONGO\_URI=mongodb://admin:admin123@mongodb:27017/ordersdb

**💳 BILLING SERVICE**

**Puerto:** 3004  
**Ruta base:** /billing

**🔹 Propósito**

Encargado de **facturación**, generación de comprobantes o tickets.  
Depende directamente de las órdenes creadas.

**🔸 Endpoints**

| **Método** | **Ruta** | **Descripción** |
| --- | --- | --- |
| **GET** | /billing | Lista todas las facturas registradas. |
| **POST** | /billing | Genera una nueva factura a partir de una orden existente. |
| **GET** | /billing/:orderId | Devuelve la factura asociada a una orden específica. |

**🔁 Comunicación**

* **Billing → Orders:** obtiene los datos de la orden (productos, precios, usuario).
* **Billing → Auth:** obtiene los datos del usuario asociado para incluirlos en la factura.
* **Reports Service** posteriormente consulta las facturas para generar reportes de ventas.

**⚙️ Variables relevantes**

BILLING\_PORT=3004

ORDERS\_SERVICE\_URL=http://orders-service:3003

AUTH\_SERVICE\_URL=http://auth-service:3000

MONGO\_URI=mongodb://admin:admin123@mongodb:27017/billingdb

**📈 REPORTS SERVICE**

**Puerto:** 3005  
**Ruta base:** /reports

**🔹 Propósito**

Centraliza y genera **reportes analíticos** del sistema.  
Consume información de múltiples microservicios.

**🔸 Endpoints**

| **Método** | **Ruta** | **Descripción** |
| --- | --- | --- |
| **GET** | /reports/sales | Genera reporte de ventas (usa datos de Billing + Orders). |
| **GET** | /reports/inventory | Genera reporte de stock (usa datos de Inventory + Products). |
| **GET** | /reports/users | Genera reporte de usuarios registrados (usa Auth Service). |

**🔁 Comunicación**

* **Reports → Billing & Orders:** obtiene datos de ventas e ingresos.
* **Reports → Inventory & Products:** analiza stock disponible o agotado.
* **Reports → Auth:** consulta datos de usuarios activos o registrados.

**⚙️ Variables relevantes**

REPORTS\_PORT=3005

BILLING\_SERVICE\_URL=http://billing-service:3004

ORDERS\_SERVICE\_URL=http://orders-service:3003

AUTH\_SERVICE\_URL=http://auth-service:3000

MONGO\_URI=mongodb://admin:admin123@mongodb:27017/reportsdb

**🌐 API GATEWAY (Nginx)**

**🔹 Propósito**

Centraliza todas las peticiones HTTP entrantes desde el **frontend**.  
Redirige las rutas hacia el microservicio correspondiente mediante **proxy\_pass**.  
Evita exponer múltiples puertos al exterior y mejora la seguridad y mantenimiento.

**🔸 Rutas principales**

| **Ruta** | **Redirige a** |
| --- | --- |
| /auth/ | http://auth-service:3000/ |
| /products/ | http://products-service:3001/ |
| /inventory/ | http://inventory-service:3002/ |
| /orders/ | http://orders-service:3003/ |
| /billing/ | http://billing-service:3004/ |
| /reports/ | http://reports-service:3005/ |
| / | http://frontend:5173/ (interfaz unificada) |

**🔁 Comunicación entre microservicios (resumen funcional)**

**🔸 Ejemplo 1 — Flujo de pedido**

1. El **usuario** se autentica en /auth/login → obtiene un JWT.
2. Desde el **frontend**, crea una orden → /orders con token.
3. **Orders Service**:
   * Valida token en **Auth Service** /auth/validate.
   * Consulta stock en **Inventory Service** /inventory/:productId.
   * Registra orden si hay disponibilidad.
4. **Billing Service** crea factura usando los datos de **Orders** y **Auth**.
5. **Reports Service** analiza ventas y genera reportes.

**🔸 Ejemplo 2 — Comunicación indirecta**

* **Inventory → Products**: valida productos.
* **Orders → Inventory → Products**: cadena jerárquica de dependencias.
* **Reports → Billing → Orders → Auth**: reportes consolidados multi-servicio.