## **🧩 División propuesta de Microservicios para Perfulandia SPA**

### 

#### **Diagrama base de microservicios:**

#### 

### **NOTA: Los Endpoints son sugeridos.**

### **1. Microservicio de Autenticación y Gestión de Usuarios**

Encargado de manejar las **credenciales**, el **login/logout**, la **gestión de roles** (cliente, vendedor, administrador) y datos generales del usuario.

* **Entidades**: Usuario, Rol, Token
* **Endpoints**:  
  + POST /auth/login
  + POST /auth/logout
  + POST /auth/registro
  + GET /usuarios/{id}
  + GET /usuarios/rol/{tipo}
* **Tecnologías sugeridas**: Spring Security + JWT

### **2. Microservicio de Clientes**

Gestiona información de clientes minoristas/mayoristas registrados.

* **Entidades**: Cliente, Direcciones, Historial de compras
* **Endpoints**:  
  + POST /clientes
  + GET /clientes/{id}
  + PUT /clientes/{id}
  + GET /clientes/preferencias

### **3. Microservicio de Vendedores**

Gestiona la información de los vendedores y sus operaciones.

* **Entidades**: Vendedor, SucursalAsignada, Metas
* **Endpoints**:  
  + POST /vendedores
  + GET /vendedores/{id}
  + PUT /vendedores/{id}
  + GET /vendedores/sucursal/{id}

### **4. Microservicio de Productos**

Administra el catálogo de productos disponibles.

* **Entidades**: Producto, Categoría, Marca
* **Endpoints**:  
  + GET /productos
  + POST /productos
  + GET /productos/{id}
  + GET /productos/categoria/{nombre}

### 

### **5. Microservicio de Inventario**

Administra el stock disponible por sucursal o bodega.

* *Entidades*\*: Inventario, MovimientoInventario, Almacén
* **Endpoints**:  
  + GET /inventario/producto/{id}
  + PUT /inventario/ajuste
  + POST /inventario/movimiento

### **6. Microservicio de Ventas**

Gestiona la compra y venta de productos, y la aplicación de cupones.

* **Entidades**: Venta, DetalleVenta, Cupon
* **Endpoints**:  
  + POST /ventas
  + GET /ventas/{id}
  + GET /ventas/cliente/{clienteId}
  + POST /cupones/validar

### 

### **7. Microservicio de Envíos**

Organiza y actualiza el estado de despachos a clientes.

* **Entidades**: Envío, EstadoEnvío, Transportista
* **Endpoints**:  
  + POST /envios
  + GET /envios/{id}
  + PUT /envios/{id}/estado

### **8. Microservicio de Soporte al Cliente**

Administra tickets y requerimientos de atención.

* **Entidades**: Ticket, TipoSoporte
* **Endpoints**:  
  + POST /soporte/tickets
  + GET /soporte/tickets/cliente/{id}
  + PUT /soporte/tickets/{id}/estado

### 

### **9. Microservicio de Reportes (opcional)**

Entrega métricas de gestión para la administración.

* **Entidades virtuales**: KPIs, Resúmenes de ventas, Inventario crítico
* **Endpoints**:  
  + GET /reportes/ventas/periodo
  + GET /reportes/vendedores/top
  + GET /reportes/inventario/critico

## **✅ Observaciones técnicas y pedagógicas**

* El **microservicio de autenticación** debe implementar **JWT**, y compartir el token con el resto de los servicios para proteger rutas privadas.
* Los roles (cliente, vendedor, admin) pueden estar integrados al **microservicio de usuarios**, con rutas específicas por tipo de rol.
* Cada microservicio puede tener su propio esquema de base de datos o estar conectado a un core central según nivel del curso.
* Se sugiere usar **Swagger/OpenAPI** y pruebas unitarias con **JUnit** en al menos 2 microservicios, para integrar las actividades de las Evaluaciones 2 y 3.

**Este es un modelo de datos base en MySQL que pueden analizar y completar los estudiantes:**  
  
**-- MODELO BASE DE DATOS PARA ARQUITECTURA DE MICROSERVICIOS - PERFULANDIA SPA**

**-- Este modelo base sirve como referencia para guiar el desarrollo por microservicios en Spring Boot + MySQL.**

**-- MICROSERVICIO DE AUTENTICACIÓN Y USUARIOS**

CREATE TABLE roles (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50) UNIQUE

);

CREATE TABLE usuarios (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

email VARCHAR(100),

enabled BOOLEAN DEFAULT TRUE,

rol\_id BIGINT,

FOREIGN KEY (rol\_id) REFERENCES roles(id)

);

**-- MICROSERVICIO DE CLIENTES**

CREATE TABLE clientes (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

usuario\_id BIGINT,

nombre\_completo VARCHAR(100),

rut VARCHAR(12),

telefono VARCHAR(15),

direccion TEXT,

FOREIGN KEY (usuario\_id) REFERENCES usuarios(id)

);

**-- MICROSERVICIO DE VENDEDORES**

CREATE TABLE vendedores (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

usuario\_id BIGINT,

sucursal VARCHAR(100),

meta\_mensual DECIMAL(10,2),

FOREIGN KEY (usuario\_id) REFERENCES usuarios(id)

);

**-- MICROSERVICIO DE PRODUCTOS**

CREATE TABLE productos (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

descripcion TEXT,

precio DECIMAL(10,2),

categoria VARCHAR(50),

marca VARCHAR(50)

);

**-- MICROSERVICIO DE INVENTARIO**

CREATE TABLE inventario (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

producto\_id BIGINT,

cantidad INT,

ubicacion VARCHAR(100),

FOREIGN KEY (producto\_id) REFERENCES productos(id)

);

**-- MICROSERVICIO DE VENTAS**

CREATE TABLE ventas (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

cliente\_id BIGINT,

vendedor\_id BIGINT,

fecha TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

total DECIMAL(10,2),

FOREIGN KEY (cliente\_id) REFERENCES clientes(id),

FOREIGN KEY (vendedor\_id) REFERENCES vendedores(id)

);

CREATE TABLE detalle\_venta (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

venta\_id BIGINT,

producto\_id BIGINT,

cantidad INT,

precio\_unitario DECIMAL(10,2),

FOREIGN KEY (venta\_id) REFERENCES ventas(id),

FOREIGN KEY (producto\_id) REFERENCES productos(id)

);

CREATE TABLE cupones (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

codigo VARCHAR(30),

descuento DECIMAL(5,2),

valido\_hasta DATE

);

**-- MICROSERVICIO DE ENVIOS**

CREATE TABLE envios (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

venta\_id BIGINT,

estado VARCHAR(50),

fecha\_envio DATE,

transportista VARCHAR(100),

FOREIGN KEY (venta\_id) REFERENCES ventas(id)

);

**-- MICROSERVICIO DE SOPORTE**

CREATE TABLE tickets\_soporte (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

cliente\_id BIGINT,

asunto VARCHAR(100),

mensaje TEXT,

estado VARCHAR(50),

fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (cliente\_id) REFERENCES clientes(id)

);

**-- NOTA: CADA MICROSERVICIO DEBE TENER SU PROPIO ESQUEMA / BASE DE DATOS SI SE UTILIZA DESACOPLAMIENTO ESTRICTO.**

**-- TAMBIÉN SE PUEDE IMPLEMENTAR UN API GATEWAY CON SPRING CLOUD GATEWAY O KONG PARA GESTIÓN CENTRAL DE RUTAS Y AUTENTICACIÓN.**

**Modelo Físico MySQL base.**

