**Código Mermaid.**

🖥️ **Diagrama de Arquitectura en Mermaid:**

graph TD

%% Clientes del sistema

Cliente[Usuario final / Administrador]

%% Capas del sistema

subgraph Capa\_de\_Presentacion

UI[Interfaz Web/Móvil]

end

subgraph Capa\_de\_Negocio

Auth[Gestión de Autenticación]

GestionProd[Gestión de Productos]

GestionInv[Gestión de Inventario]

GestionVentas[Gestión de Ventas]

Reportes[Generación de Reportes]

end

subgraph Capa\_de\_Datos

DB[(Base de Datos Relacional)]

end

%% Servicios externos

subgraph Servicios\_Externos

Logs[Servicio de Logs/Auditoría]

Backup[Servicio de Respaldo Automático]

end

%% Conexiones Cliente-Presentación

Cliente --> UI

%% Conexiones Presentación-Negocio

UI --> Auth

UI --> GestionProd

UI --> GestionInv

UI --> GestionVentas

UI --> Reportes

%% Conexiones Capa de Negocio-Datos

Auth --> DB

GestionProd --> DB

GestionInv --> DB

GestionVentas --> DB

Reportes --> DB

%% Conexiones Capa de Negocio-Servicios Externos (Requisitos no funcionales)

GestionVentas --> Logs

GestionProd --> Logs

GestionInv --> Logs

Auth --> Logs

DB --> Backup

%% Comentarios de Requerimientos No Funcionales

classDef comentario fill:#e8f5e9,stroke:#81c784,stroke-width:1px,color:#333;

RNF1[Alta disponibilidad: Servicio Backup]:::comentario

RNF2[Seguridad: Autenticación y Logs]:::comentario

RNF3[Escalabilidad: Componentes modulares independientes]:::comentario

RNF4[Rendimiento: Uso eficiente de base de datos]:::comentario

%% Ubicación comentarios

Backup --- RNF1

Logs --- RNF2

Capa\_de\_Negocio --- RNF3

DB --- RNF4

🛠️ **Diagrama de Componentes (Mermaid)**:

flowchart TD

%% Componente Principal (Aplicación)

subgraph AplicacionInventario

InterfazUsuario["📱 Interfaz Usuario Web/Móvil"]

APIBackend["🔄 API Backend"]

ModuloAuth["🔐 Módulo Autenticación y Autorización"]

ModuloGestionProd["📦 Módulo Gestión Productos"]

ModuloGestionInv["📊 Módulo Gestión Inventario"]

ModuloVentas["💳 Módulo Gestión Ventas"]

ModuloReportes["📈 Módulo Reportes"]

end

%% Componente Base de Datos

subgraph Almacenamiento

BaseDatos[(🗄️ Base de Datos Relacional)]

end

%% Servicios externos y apoyo

subgraph ServiciosExternos

ServicioLogs["📜 Servicio Logs/Auditoría"]

ServicioBackup["💾 Servicio Backup"]

end

%% Conexiones entre componentes internos

InterfazUsuario <--> APIBackend

APIBackend --> ModuloAuth

APIBackend --> ModuloGestionProd

APIBackend --> ModuloGestionInv

APIBackend --> ModuloVentas

APIBackend --> ModuloReportes

ModuloAuth --> BaseDatos

ModuloGestionProd --> BaseDatos

ModuloGestionInv --> BaseDatos

ModuloVentas --> BaseDatos

ModuloReportes --> BaseDatos

%% Conexiones con Servicios externos

APIBackend --> ServicioLogs

ModuloAuth --> ServicioLogs

ModuloVentas --> ServicioLogs

BaseDatos --> ServicioBackup

📌 **Diagrama de base de datos (Mermaid):**

erDiagram

%% TABLAS PRINCIPALES

USUARIO ||--o{ VENTA : registra

PRODUCTO ||--o{ INVENTARIO : tiene

PRODUCTO ||--o{ DETALLE\_VENTA : incluye

VENTA ||--|| DETALLE\_VENTA : contiene

CATEGORIA ||--o{ PRODUCTO : categoriza

PROVEEDOR ||--o{ PRODUCTO : suministra

%% DEFINICION DE TABLAS Y CAMPOS

USUARIO {

int id\_usuario PK

varchar nombre

varchar email

varchar password

datetime fecha\_registro

}

PRODUCTO {

int id\_producto PK

varchar nombre

text descripcion

decimal precio

int id\_categoria FK

int id\_proveedor FK

datetime fecha\_creacion

}

CATEGORIA {

int id\_categoria PK

varchar nombre

varchar descripcion

}

PROVEEDOR {

int id\_proveedor PK

varchar nombre

varchar contacto

varchar telefono

varchar email

}

INVENTARIO {

int id\_inventario PK

int id\_producto FK

int cantidad

datetime ultima\_actualizacion

}

VENTA {

int id\_venta PK

int id\_usuario FK

datetime fecha\_venta

decimal total

}

DETALLE\_VENTA {

int id\_detalle PK

int id\_venta FK

int id\_producto FK

int cantidad

decimal precio\_unitario

}

AUDITORIA\_LOGS {

int id\_log PK

varchar accion

varchar modulo

datetime fecha\_hora

int id\_usuario FK

}