Actividad:

Diagrama de componentes

Nombre:

David Alejandro Salas

Materia:

Diseño de software

Docente:

Miguel Tovar

San juan de pasto

17 de septiembre del 2025

Cuarto semestre

Ingeniería en Software

**Ejercicio 1: Sistema Web Centro Médico**

((Sistema Web Centro Médico))

Pacientes

RF01: Registro y autenticación

RF02: Búsqueda de médicos

RF03: Reserva de citas

RF04: Historial de citas

RF05: Cancelación/reprogramación

RF06: Recordatorios y notificaciones

Médicos

RF07: Autenticación de médicos

RF08: Gestión de agenda

RF09: Visualización de citas

RF10: Confirmación/cancelación

RF11: Historial clínico básico

RF12: Bloqueo de horarios

Administradores

RF13: Gestión de usuarios

RF14: Configuración de especialidades

RF15: Reportes y estadísticas

RF16: Horarios institucionales

RF17: Monitoreo del sistema

**Ejercicio 2: Aplicación de Delivery**

Usuarios/Clientes

RF01: Registro y autenticación

RF02: Búsqueda de restaurantes

RF03: Visualización de menús

RF04: Carrito de compras

RF05: Pedidos y pagos

RF06: Seguimiento en tiempo real

RF07: Calificaciones y reseñas

RF08: Historial de pedidos

Restaurantes

RF09: Registro y perfil

RF10: Gestión de menús

RF11: Gestión de pedidos

RF12: Disponibilidad de productos

RF13: Confirmación de tiempos

RF14: Dashboard de ventas

Repartidores

RF15: Registro y autenticación

RF16: Recepción de pedidos

RF17: Rutas optimizadas

RF18: Recolección y entrega

RF19: Disponibilidad

RF20: Historial de entregas

Sistema

RF21: Pagos y comisiones

RF22: Geolocalización

RF23: Notificaciones push

RF24: Cálculo de envío

RF25: Soporte y chat

Ejercicio 3: Sistema de Biblioteca (Refactorizado)

((Sistema Biblioteca))

Gestión de Usuarios

RF01: Registro de estudiantes/profesores

RF02: Autenticación y perfiles

RF03: Gestión de privilegios

Gestión de Catálogo

RF04: CRUD de libros

RF05: Búsqueda y filtrado

RF06: Categorización y etiquetado

RF07: Control de inventario

Gestión de Préstamos

RF08: Solicitud de préstamos

RF09: Validación de disponibilidad

RF10: Renovaciones y devoluciones

RF11: Multas y sanciones

RF12: Reservas de libros

Reportes

RF13: Reportes de préstamos

RF14: Estadísticas de uso

RF15: Reportes de multas

RF16: Popularidad de libros

# Ejercicio 1 – Centro Médico

## 2. Modularización (Componentes)

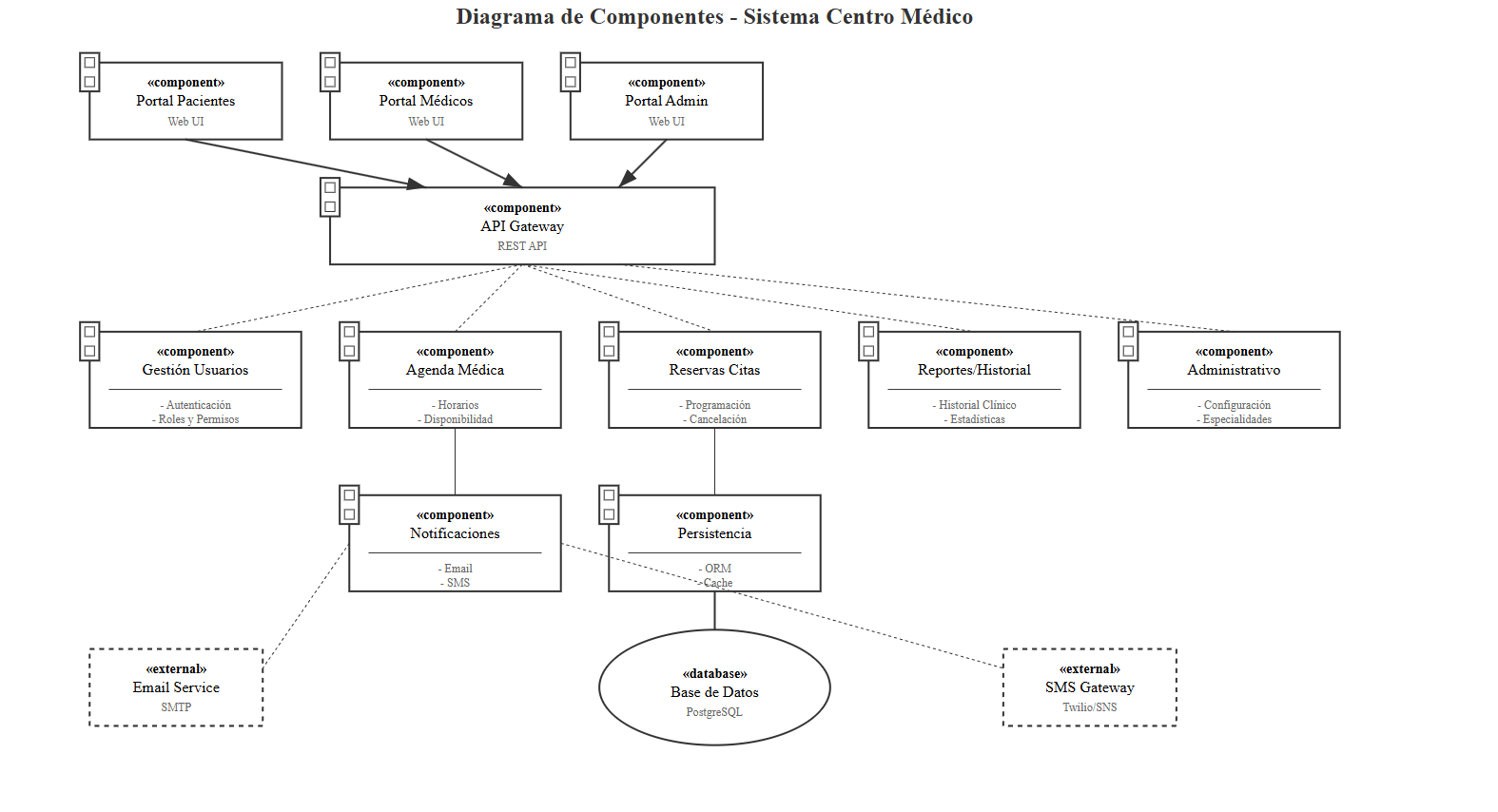
- Módulo de Gestión de Usuarios: Registro, autenticación, administración de roles y permisos.  
- Módulo de Agenda Médica: Creación, asignación y actualización de horarios de los médicos.  
- Módulo de Reservas de Citas: Solicitud, confirmación, cancelación y reprogramación de consultas.  
- Módulo Administrativo: Gestión de usuarios internos, especialidades médicas y parámetros del sistema.  
- Módulo de Reportes e Historial Clínico: Generación de reportes médicos, acceso a historial de citas y evolución clínica.  
- Módulo de Notificaciones: Envío de recordatorios y alertas vía correo electrónico o SMS.  
- Módulo de Persistencia: Base de datos centralizada con control de acceso y respaldo automático.

## 3. Interfaces – Conectores

- API REST: Comunicación entre frontend, backend y aplicaciones móviles.  
- ORM/SQL: Persistencia y consultas de datos estructurados.  
- Servicios externos: Conectores para envío de correos, SMS y posibles integraciones con aseguradoras.

## 4. Análisis de diseño

La modularización se diseñó en torno a los principales roles del sistema: pacientes, médicos y administradores. Esta estrategia promueve alta cohesión dentro de cada módulo y bajo acoplamiento entre ellos, lo cual incrementa la escalabilidad y simplifica el mantenimiento. El módulo de notificaciones se mantiene independiente para facilitar cambios de proveedor sin afectar el resto del sistema. Además, la separación entre agenda, reservas y administración permite evolucionar el sistema hacia futuras integraciones con apps móviles o telemedicina.



# Ejercicio 2 – Plataforma de Delivery (estilo Rappi/Uber Eats)

## 2. Modularización (Componentes)

- Módulo de Gestión de Usuarios: Registro, autenticación y perfil de clientes, restaurantes y repartidores.  
- Módulo de Pedidos: Creación, seguimiento y actualización de órdenes en tiempo real.  
- Módulo de Restaurantes: Administración de menús, disponibilidad y precios.  
- Módulo de Repartidores: Asignación de pedidos, rutas y seguimiento de entregas.  
- Módulo de Pagos: Procesamiento seguro de transacciones electrónicas.  
- Módulo de Notificaciones: Envío de alertas y actualizaciones en tiempo real (push/email).  
- Módulo Administrativo: Gestión global del sistema, promociones, métricas y soporte.  
- Módulo de Persistencia: Base de datos optimizada para consultas masivas y operaciones simultáneas.

## 3. Interfaces – Conectores

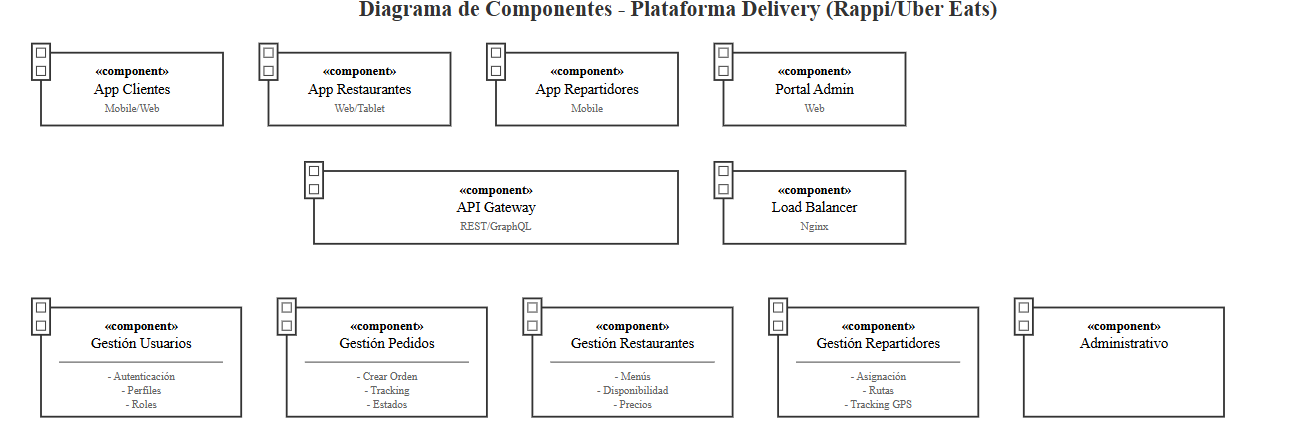
- API REST: Comunicación entre clientes, restaurantes, repartidores y backend.  
- Pasarela de pagos: Integración con sistemas de pago seguros (ej: PayU, Stripe).  
- Conector BD: Manejo de datos con ORM y consultas optimizadas.  
- Servicios externos: Notificaciones push/SMS y geolocalización.

## 4. Análisis de diseño

El diseño separa los procesos críticos (pedidos, pagos, logística de repartidores) en módulos específicos, garantizando alta disponibilidad y escalabilidad. El módulo de pagos se mantiene aislado por seguridad, mientras que las notificaciones en tiempo real optimizan la experiencia del usuario. Esta arquitectura facilita la incorporación de nuevas funciones como cupones, programas de fidelización o integraciones con marketplaces externos.

## 5. Socialización

Se presentará un diagrama de arquitectura, mostrando cómo interactúan clientes, restaurantes y repartidores mediante el flujo de pedidos. Se resaltará la importancia de un módulo de pagos seguro y la relevancia de las notificaciones instantáneas para la satisfacción del usuario.



# Ejercicio 3 – Biblioteca Escolar (Refactorización)

## 2. Modularización (Componentes)

- Módulo de Autenticación: Inicio de sesión seguro, control de accesos y roles (alumno, profesor, bibliotecario).  
- Módulo de Gestión de Usuarios: Información personal, historial de préstamos y sanciones.  
- Módulo de Catálogo: Registro, búsqueda y clasificación de materiales bibliográficos.  
- Módulo de Préstamos y Devoluciones: Administración de préstamos, renovaciones y devoluciones.  
- Módulo de Reportes y Estadísticas: Métricas de uso, materiales más solicitados y gestión de inventario.  
- Módulo de Persistencia: Base de datos con control de integridad y respaldo periódico.

## 3. Interfaces – Conectores

- API REST: Conexión entre frontend y backend.  
- Conector BD (ORM/SQL): Consultas y gestión de datos.  
- Servicios internos: Comunicación entre préstamos, catálogo y usuarios.

## 4. Análisis de diseño

El sistema inicial era monolítico, lo que dificultaba mantenimiento y escalabilidad. La refactorización propone separar en módulos independientes para mejorar flexibilidad y evolución del sistema. Ahora, cada componente puede actualizarse o escalarse sin afectar el resto.

## 5. Socialización

Se expondrá un diagrama de arquitectura modular, destacando la transición de un sistema monolítico a uno basado en componentes. Se enfatizarán las ventajas: mantenibilidad, flexibilidad y capacidad de crecimiento.

