**Modularización y componetización.**

Elabore para cada uno de los siguientes enunciados:

1. Identifique las funcionalidades (requisitos funcionales)
2. Modularice el sistema que dará solución a las funcionalidades identificando componentes
3. Identifique las interfaces – conectores que permitirán comunicar los componentes
4. Agregue un análisis breve que justifique las decisiones de diseño para el modelo propuesto.
5. Socialice en clase sus propuestas de diseño.

**Ejercicio 1.** Un centro médico requiere un sistema web para que pacientes puedan reservar consultas, médicos gestionen su agenda y los administradores controlen el sistema.

**Ejercicio 2.** Una startup quiere desarrollar una aplicación móvil y web tipo Rappi o Uber Eats, donde los usuarios pueden pedir comida, los restaurantes gestionan pedidos y los repartidores hacen las entregas.

**Ejercicio 3.** Un colegio tiene un sistema heredado de biblioteca que mezcla lógica de préstamo, usuarios, catálogo y reportes en un solo módulo monolítico. El sistema debe ser **refactorizado y rediseñado** usando principios modernos de diseño.

**Desarrollo**

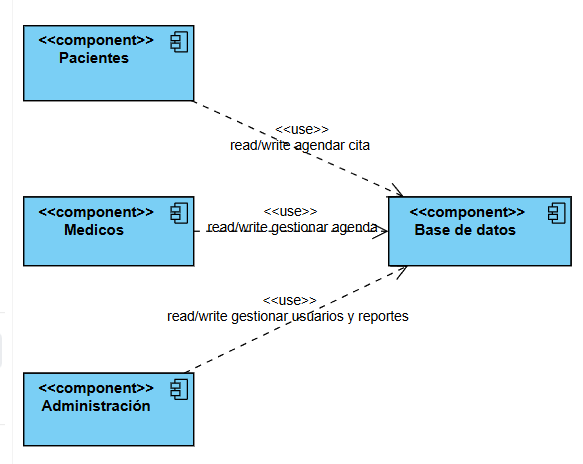
**Ejercicio 1. sistema web para centro medico**

* 1. **Funcionalidades**
* **Revisar consultas en linea**
* **Gestion de agenda para medicos**
* **Administracion del sistema por parte de los administradores** 
  1. **Componentes**
* Modulo para pacientes ( reservas )
* Modulo para medicos ( gestion de agenda )
* Modulo de administracion (control y gestion)

**1.3 . Interfaces – Conectores**

* API REST pacientes ↔ medicos ↔ administracion
* Base de datos compartida para los usuarios y los turnos
  1. **Análisis**

Este sistema se divide en tres componenetes claros los cuales tienen roles especificos lo que ivita mezclar las respinsabilidades que tiene cada uno de estos , permitiendo asi escalar funciones independientes.



**Ejercicio 2.** **App tipo Rapi/Uber Eats**

**2.1. Funcionalidades**

* Realizar pedidos
* Gestionar órdenes
* Realizar entregas

**2.2.Componentes**

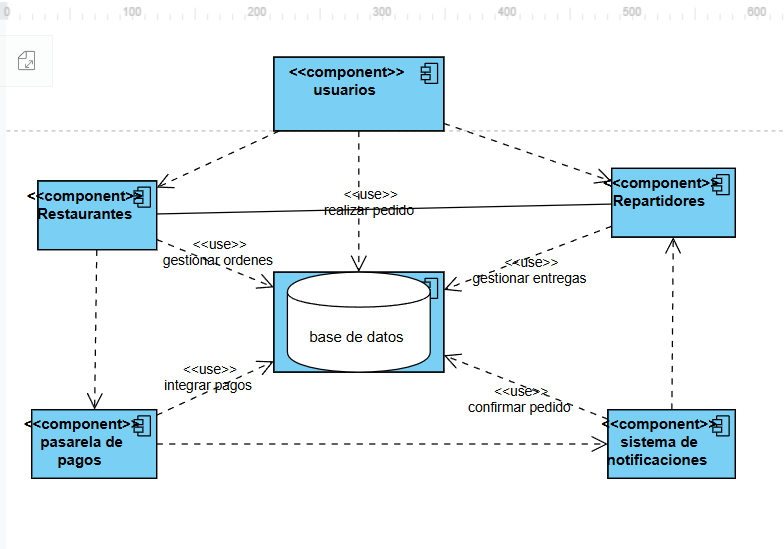
* Modulo para usuarios (app movil/web para los pedidos)
* Modulo para restaurantes (gestion del menu y pedidos)
* Modulo para repartidores (gestion entrega de pedidos )
* Modulo de pasarela de pagos (validar pagos)

**2.3. Interfaces – Conectores**

* Api movil de pedidos ↔ usuarios ↔restaurantes ↔ repartidores
* Integrar pasarela de pagos
* Sistema para notificar pedido realizado

**2.4.Análisis**

El sistema en este caso separa muy bien los roles y procesos que realizara cada componente. Esto facilita la escalabilidad de cada modulo generando asi mas clientes , restaurantes y repartidores, sin afectar las funcionalidades que cada uno dbe cumplir entre si para el buen funcionamiento del sistema y la prestacion del servicio.



**Ejercicio 3.** **Sistema de biblioteca heredado.**

**3.1. Funcionalidades**

* **Prestar libros**
* **Gestionar usuarios**
* **Catalo de libros**
* **Reportes**

**3.2. Componentes**

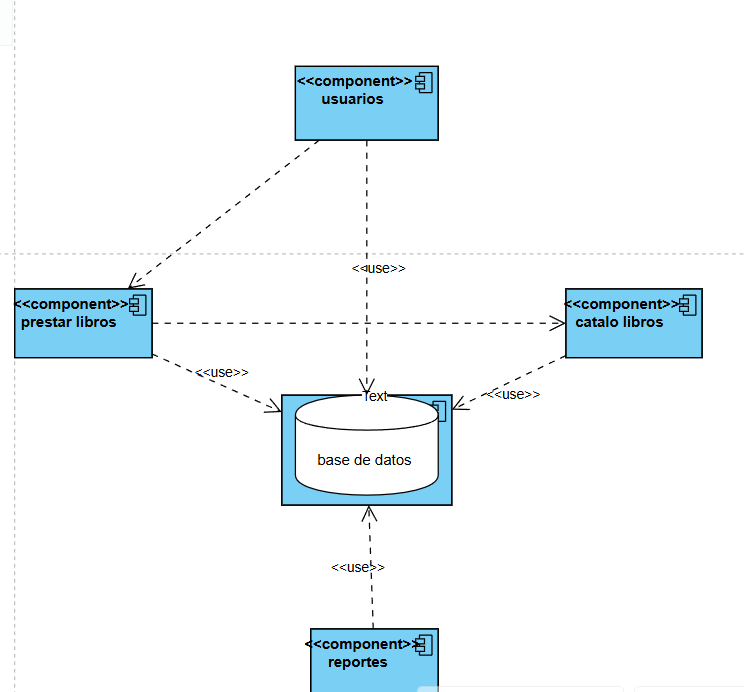
* **Modulo de prestamos**
* **Modulo de usuarios**
* **Modulo de catalo**
* **Modulo de reportes**

**3.3. Interfaces – Conectores**

* **Base de datos compartida con acceso por modulos**
* **Servicios internos para consultar disponibilidad de libros**

**3.4.Análisis**

El sistema cuenta con cuatro modulos separados, cada uno de estos esta especializado en una parte del sistema, mejorando la mantenibilidad y escalabilidad del sistema como tal . Esta modularizacion de componentes permite que a futuro se puedan adaptar microservicios para cada componente.

****