**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.

**Requisitos funcionales**

**Gestión de usuarios**

* RQF-001: El sistema debe permitir registrar a los usuarios.
* RQF-002: El sistema debe permitir iniciar sesión a los usuarios.

**Paciente**

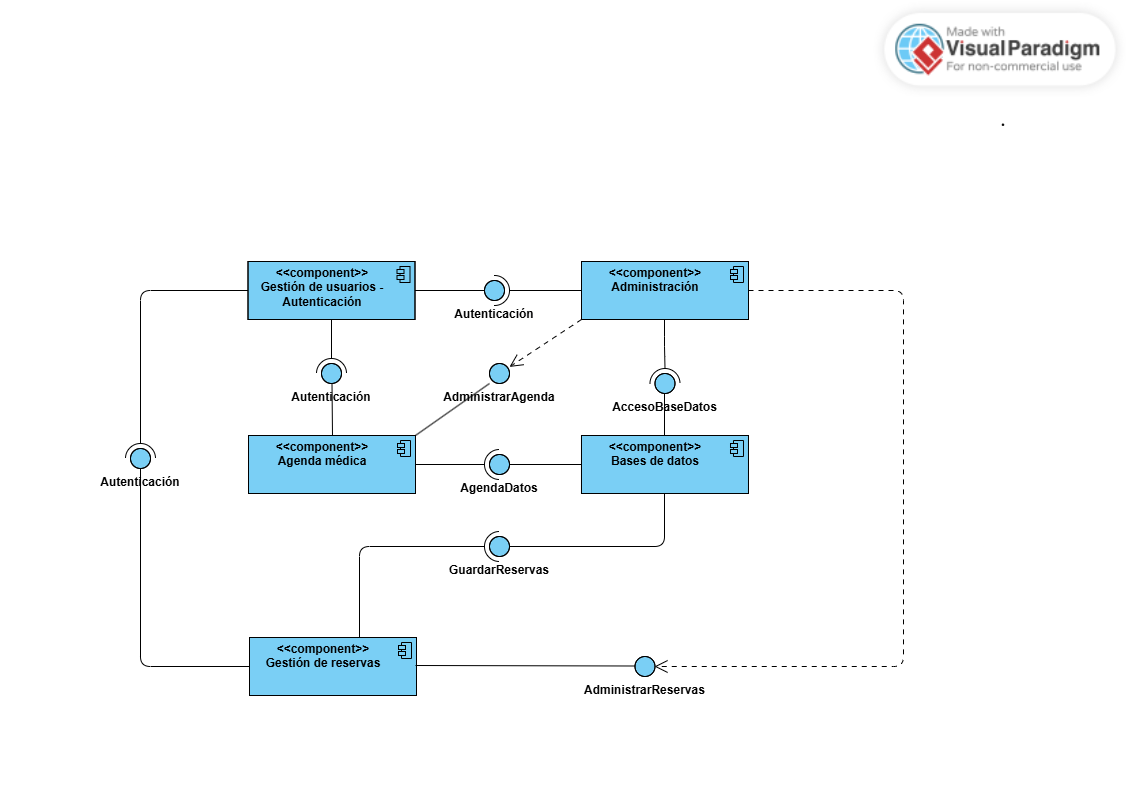
* RQF-003: El sistema debe permitir a los pacientes reservar consultas.
* RQF-004: El sistema debe permitir a los pacientes modificar su reserva.
* RQF-005: El sistema debe permitir a los pacientes cancelar su reserva.
* RQF-006: El sistema debe enviar notificaciones a los pacientes (Recordatorios).

**Médico**

* RQF-007: El sistema debe permitir a los médicos gestionar su agenda.
* RQF-008: El sistema debe enviar notificaciones a los médicos (Cambios de citas, nuevas reservas).

**Administrador**

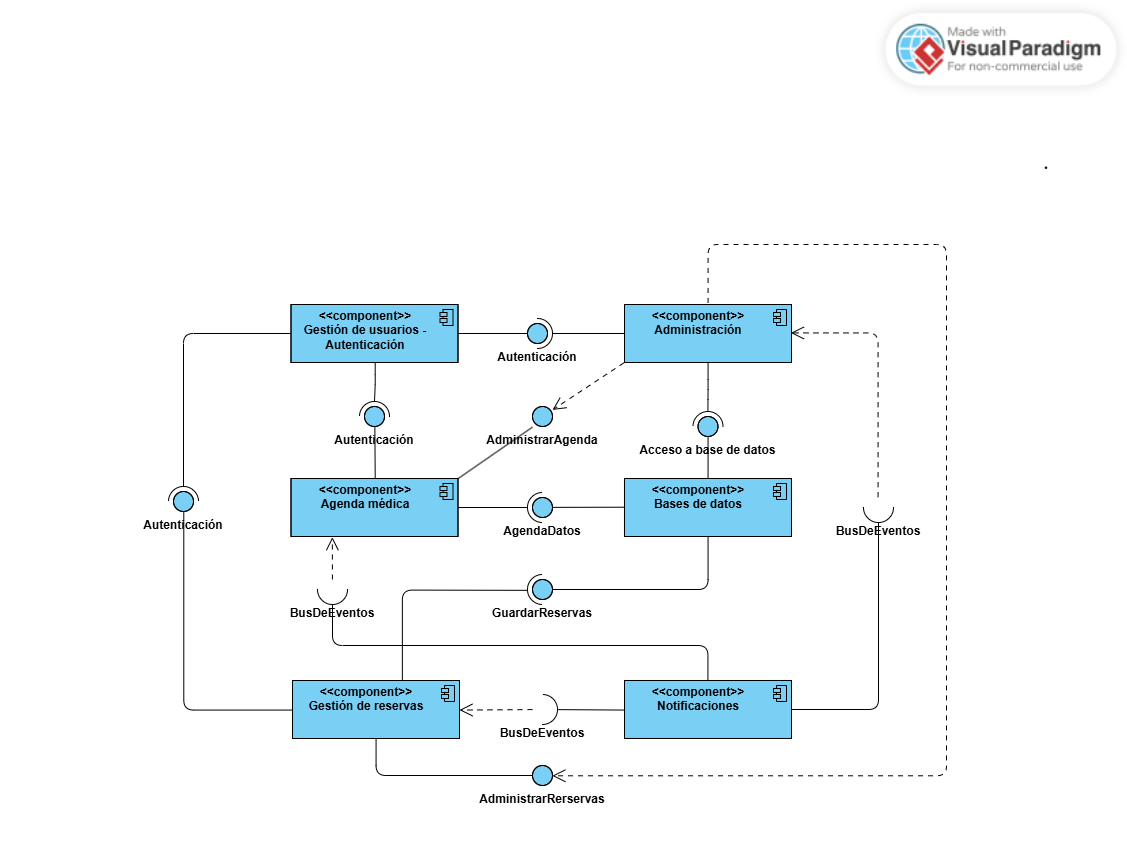
* RQF-009: El sistema debe permitir a los administradores controlar el sistema.
* RQF-010: El sistema debe enviar notificaciones a los administradores (Alertas del sistema).



**¿Por qué se eligió esta decisión de diseño?**

Uno de los factores claves aquí es en cómo se van a reservar las consultas, entonces, primero debemos asignar un componente que nos permita manejar lo que viene siendo lo de la gestión de usuarios y autenticación, esto, tanto para los pacientes, como también médicos y administradores. Ya con esto, podemos ver la función de los diferentes componentes, los cuales son: Gestión de reservas, Agenda médica, Administración y Base de datos. Cada uno de estos componentes tienen una interfaz concisa de como se van a comunicar entre sí con otros, ninguno va a tener acceso a otro cuando no sea necesario la comunicación entre ellos.

**\*Segunda Opción\***

****

En este diagrama se incluyen “Notificaciones”, pero el diseño se miraba intrincado, de igual manera se puede ver como actúa este componente con los demás componentes.

**Ejercicio 2.** Un 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.

**Gestión de usuarios**

* RQF-001: El sistema debe permitir registrar a los usuarios.
* RQF-002: El sistema debe permitir iniciar sesión a los usuarios.

**Usuarios (Compradores)**

* RQF-003: El sistema debe permitir a los usuarios pedir comida.
* RQF-004: El sistema debe permitir a los usuarios realizar pagos dentro y fuera de la app.

**Gestión de Restaurantes**

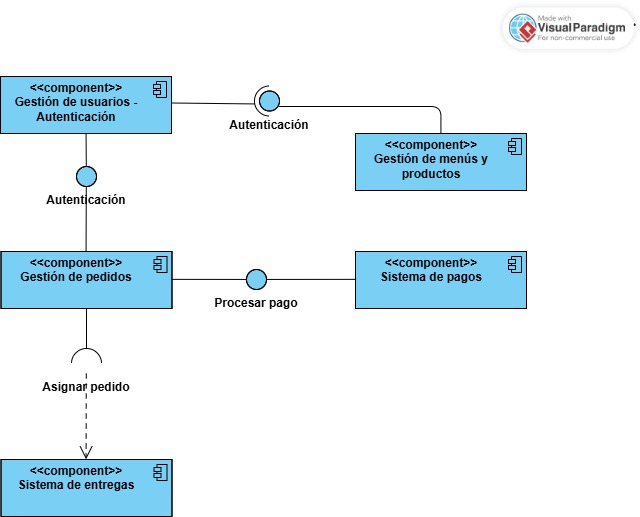
* RQF-005: El sistema debe permitir a los restaurantes gestionar su menú.
* RQF-006: El sistema debe permitir a los restaurantes recibir pedidos.

**Repartidores**

* RQF-007: El sistema debe permitir a los repartidores gestionar sus órdenes.

**Sistema**

* RQF-008: El sistema asigna órdenes a los repartidores.

****

**¿Por qué se eligió esta decisión de diseño?**

Este diagrama fue diseñado para ver cómo se conectan los diferentes componentes de una App/Web para comidas, priorizando los componentes más importantes para poder entender la funcionalidad del sistema. Este diagrama se centra en 5 componentes: Gestión de usuarios, Gestión de menús y productos, Gestión de pedidos, Sistema de pagos y Sistema de entregas, cada componente tiene una comunicación única con el otro, con interfaces que reflejan la funcionalidad entre ellos.

**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.

**Gestión de usuarios**

* RQF-001: El sistema debe permitir registrar a los usuarios.
* RQF-002: El sistema debe permitir iniciar sesión a los usuarios.

**Gestión de libros (Catálogo)**

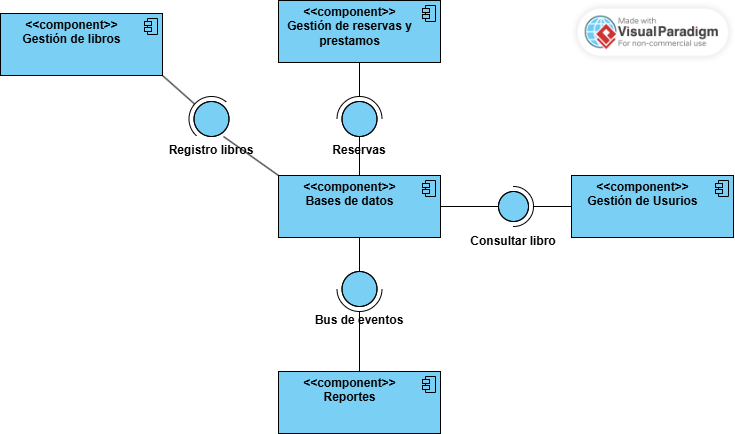
* RQF-003: El sistema debe permitir buscar libros por título, autor y categoría.
* RQF-004: El sistema debe permitir agregar libros.
* RQF-005: El sistema debe permitir eliminar libros.
* RQF-006: El sistema debe permitir modificar libros.

**Gestión de reservas y prestamos (Préstamos)**

* RQF-007: El sistema debe permitir al usuario reservar libros.
* RQF-008: El sistema debe permitir cancelar reservas.
* RQF-009: El sistema debe permitir a los bibliotecarios registrar préstamos.
* RQF-010: El sistema debe notificar sobre vencimientos a préstamos 24 horas antes de la devolución.

**Reportes**

* RQF-011: El sistema debe generar reportes sobre los libros más prestados.
* RQF-012: El sistema debe generar reportes con mora en la devolución de libros.

****

**¿Por qué se eligió esta decisión de diseño?**

El diagrama de componentes presentado ilustra la refactorización de un sistema monolítico de gestión de biblioteca en una arquitectura modular. Se han identificado cinco componentes principales: Gestión de Usuarios, responsable de la autenticación y autorización; Gestión de Reservas y Préstamos, que administra el ciclo de vida de los préstamos; Gestión de Libros, encargado de la catalogación y mantenimiento de la información bibliográfica; y Reportes, que genera informes estadísticos sobre el uso de la biblioteca. Esta modularización facilita el mantenimiento, la escalabilidad y la integración de nuevas funcionalidades, promoviendo una mayor robustez y flexibilidad del sistema.