Ingeniería de Software IV

Diseño de Software

Modularización y Componentización

Santiago Bustos López

Ingeniería de Software IV

Diseño de Software – Modularización y Componentización

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.

**Requerimientos del sistema:**

**RQM1:** El sistema debe permitir un registro con los datos personales del usuario.

**RQM2:** El sistema debe permitir iniciar sesión al usuario.

**RQM3:** El usuario paciente debe poder consultar la disponibilidad del médico, teniendo en cuenta la fecha y hora.

**RQM4:** El usuario paciente puede hacer una reservación de su cita medica .

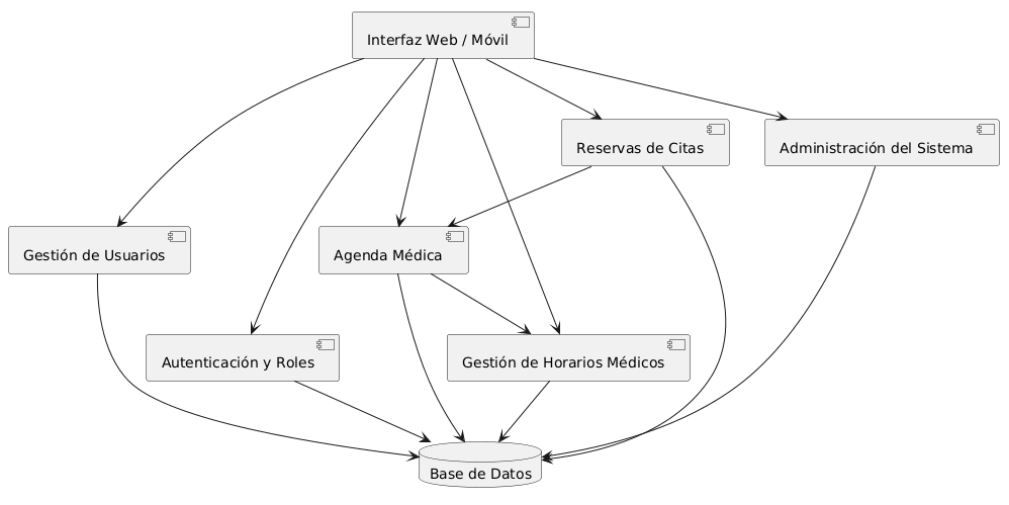
**RQM5:** El usuario medico podrá gestionar la disponibilidad de su horario.

**RQM6:** Los administradores controlen el sistema.

**Módulos del sistema:**

* **Módulo de Gestión de Usuarios**: registro de pacientes, médicos y administradores.
* **Módulo de Autenticación y Roles**: login y control de permisos según el tipo de usuario.
* **Módulo de Agenda Médica**: consulta de horarios y disponibilidad.
* **Módulo de Reservas de Citas**: gestiona solicitudes y confirmaciones.
* **Módulo de Gestión de Horarios Médicos**: definir disponibilidad, modificar horarios.
* **Módulo de Administración del Sistema**: gestión global de usuarios, citas y configuración.

**Interfaces:**

****

**Análisis:**

El sistema se diseñó de manera modular para separar responsabilidades y facilitar el mantenimiento.  
Cada requisito se asocia a un módulo específico: gestión de usuarios y autenticación para la seguridad, agenda y reservas para el control de citas, horarios para los médicos y un módulo de administración para la gestión global.  
La base de datos central asegura la consistencia de la información (usuarios, citas, horarios).  
Las interfaces claras entre módulos permiten la comunicación fluida entre pacientes, médicos y administradores.  
Este diseño modular favorece la escalabilidad (agregar nuevas funciones en el futuro, como recordatorios por correo) y la flexibilidad (mejorar un módulo sin afectar los demás).

**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.

**Requerimientos Funcionales:**

**RQMF1:** El usuario puede registrar una cuenta con sus datos personales.

**RQMF2:** El usuario cliente pude iniciar sesión.

**RQMF3:** El usuario tipo cliente puede agregar uno o más números de teléfono.

**RQMF4:** El usuario tipo cliente puede buscar el restaurante del cual dese pedir su comida.

**RQMF5:** El usuario tipo cliente puede agregar una o mas localizaciones para la entrega del pedido.

**RQMF6:** El usuario tipo cliente debe poder registrar un método de pago virtual.

**RQMF7:** El usuario tipo cliente debe poder realizar pedidos.

**RQMF8:** El usuario tipo restaurante debe gestionar los pedidos.

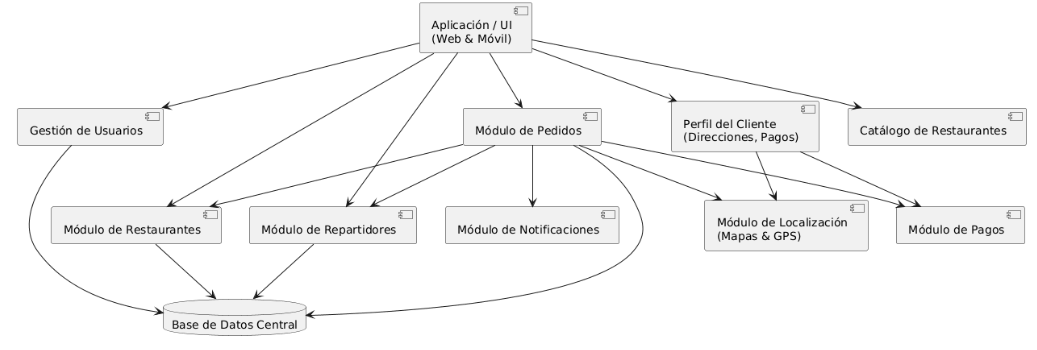
**RQMF9:** El usuario tipo repartidor debe poder visualizar la ubicación del restaurante y del cliente.

**RQMF10:** El usuario tipo repartidor debe poder registrar que el pedido ha sido recogido y entregado.

**Módulos del Sistema:**

* **Módulo de gestión de usuarios:** Manejo de clientes, restaurantes y repartidores (registro, login, actualización de datos).
* **Módulo de gestión del perfil:** Gestión de teléfonos, direcciones y métodos de pago.
* **Módulo de catálogo de restaurantes:** Permite buscar restaurantes y consultar menús
* **Módulo de pedidos:** Orquestador del proceso de pedido
* **Módulo de gestión de pedidos:** Seguimiento de pedidos en curso, registro de recogida y entrega.
* **Módulo de pagos:** Validación y procesamiento de pagos virtuales.
* **Módulo de notificaciones:** Envía notificaciones a clientes, restaurantes y repartidores.

**Interfaces:**



**Análisis:**

El sistema se modularizó para garantizar claridad, escalabilidad y mantenimiento sencillo.  
Cada módulo cumple una función específica (usuarios, pedidos, pagos, localización, etc.), lo que evita dependencias innecesarias y facilita reemplazar o mejorar componentes sin afectar todo el sistema.  
La base de datos central asegura consistencia en la información, mientras que las interfaces bien definidas permiten comunicación clara entre módulos y la app.  
Este diseño favorece la extensibilidad (añadir nuevas funciones fácilmente) y la escalabilidad, ya que los módulos críticos como pedidos o pagos pueden crecer de forma independiente.

**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.

**Requerimientos funcionales:**

**RQMF1: Control de préstamos:** administración de préstamos y devoluciones, seguimiento de fechas de entrega, renovaciones y aplicación de sanciones por retraso.

**RQMF2:** **Gestión del catálogo:** permite registrar, actualizar, eliminar y consultar libros y materiales disponibles en la biblioteca.

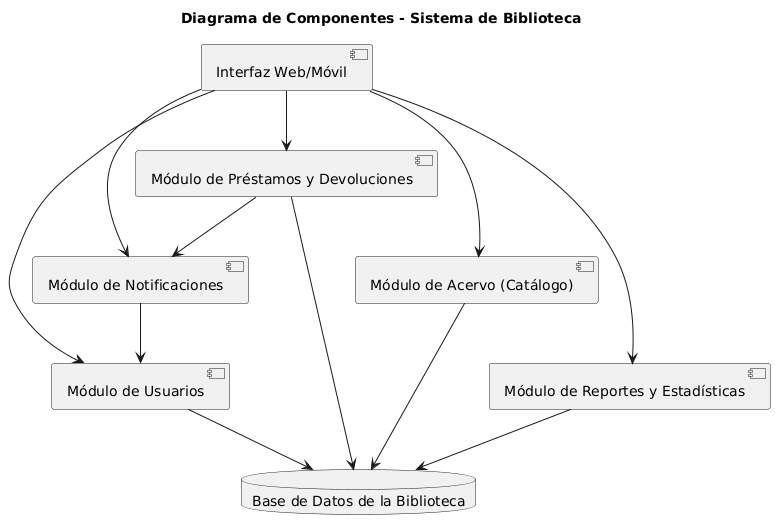
**RQMF3: Sistema de notificaciones:** envío de recordatorios sobre fechas de entrega, multas pendientes y difusión de novedades o eventos de la biblioteca.

**RQMF4: Generación de reportes:** elaboración de informes sobre circulación de libros, usuarios con mayor actividad y materiales más solicitados.

**Módulos del sistema:**

* **Módulo de Usuarios:** administración de datos de usuarios, perfiles, historial de préstamos y roles.
* **Módulo de Catálogo:** manejo del inventario de libros y materiales, con funciones de alta, baja, modificación y búsqueda por criterios.
* **Módulo de Préstamos y Devoluciones:** registro de operaciones de préstamo, devolución y renovación, con cálculo automático de penalizaciones.
* **Módulo de Reportes y Estadísticas:** generación de informes de uso, tendencias de lectura y métricas de gestión bibliotecaria.
* **Módulo de Notificaciones:** envío automatizado de alertas sobre devoluciones, multas y novedades de interés para los usuarios.

**Interfaces:**



**Análisis:**

El sistema se diseñó de forma modular para separar responsabilidades y simplificar el mantenimiento. Cada requisito se tradujo en un módulo independiente: usuarios, catálogo, préstamos, reportes y notificaciones.  
La base de datos central asegura la integridad de la información (usuarios, libros, préstamos), mientras que las interfaces claras permiten la interacción entre los módulos sin dependencias directas.  
Este diseño facilita la escalabilidad (por ejemplo, añadir nuevos servicios como reservas en línea) y la flexibilidad, ya que se pueden actualizar módulos específicos (como notificaciones) sin afectar al resto del sistema.