**Modularización y componetización.**

Elabore para cada uno de los siguientes enunciados:

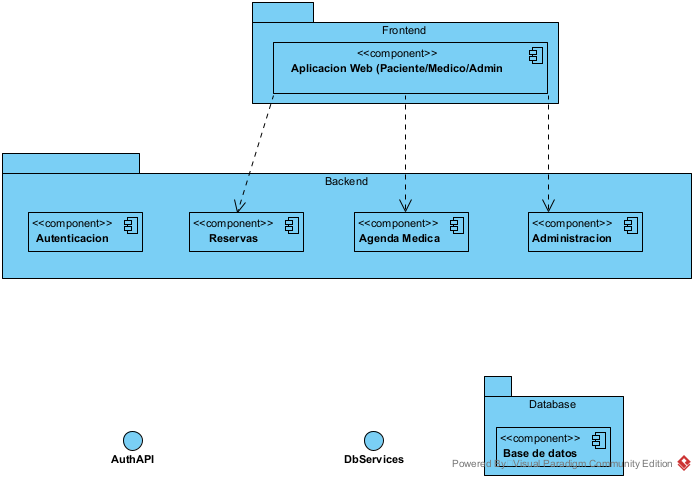
1. Identifique las funcionalidades (requisitos funcionales)
2. Modularice el sistema que dará solución a las funcionalidades identificando componentes
3. Identifique las interfaces – conectores que permitirán comunicar los componentes
4. Agregue un análisis breve que justifique las decisiones de diseño 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**

1. El sistema debe permitir a los pacientes reservar consultas médicas en línea.
2. El sistema debe permitir a los médicos gestionar su agenda (agregar, modificar y cancelar citas).
3. El sistema debe permitir a los administradores controlar el sistema (usuarios, permisos, configuración).
4. El sistema debe ofrecer autenticación de usuarios para garantizar la seguridad de accesos.
5. El sistema debe contar con un almacenamiento centralizado para guardar la información de pacientes, médicos y citas.

|  |  |
| --- | --- |
| **Funcionalidad** | **Componente** |
| Reservar consulta | Módulo Reservas |
| Gestionar agenda | Módulo Agenda Médica |
| Control del sistema | Módulo Administración |
| Autenticación de usuarios | Módulo Autenticación |
| Almacenamiento de datos | Base de Datos |



**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.

**Requisitos funcionales**

1. La aplicación debe permitir a los usuarios pedir comida desde el móvil o web.
2. La aplicación debe permitir a los restaurantes gestionar su menú y pedidos en tiempo real.
3. La aplicación debe permitir a los repartidores gestionar las entregas y actualizar estados.
4. El sistema debe ofrecer un módulo de autenticación para usuarios, restaurantes y repartidores.
5. La aplicación debe contar con un sistema para procesar pagos en línea.
6. El sistema debe disponer de un almacenamiento centralizado que guarde usuarios, pedidos, menús y transacciones.

|  |  |
| --- | --- |
| **Funcionalidad** | **Componente** |
| Pedir comida (usuarios) | Módulo Pedidos |
| Gestionar menú y pedidos | Módulo Restaurantes |
| Entregas | Módulo Repartidores |
| Autenticación | Módulo Autenticación |
| Procesar pagos | Pasarela de Pagos |
| Almacenamiento de datos | Base de Datos |

Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

**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.

Requisitos funcionales

1. El sistema debe permitir el préstamo y devolución de libros a los usuarios.
2. El sistema debe permitir la gestión de usuarios (registro, consulta, actualización).
3. El sistema debe permitir la consulta del catálogo de libros en línea.
4. El sistema debe permitir la generación de reportes de préstamos, usuarios y catálogo.
5. El sistema debe ofrecer un módulo de autenticación para garantizar el acceso seguro.
6. El sistema debe contar con un almacenamiento centralizado para la información de usuarios, libros y préstamos.

|  |  |
| --- | --- |
| **Funcionalidad** | **Componente** |
| Préstamo de libros | Módulo Préstamos |
| Gestión de usuarios | Módulo Usuarios |
| Consulta de catálogo | Módulo Catálogo |
| Generación de reportes | Módulo Reportes |
| Autenticación | Módulo Autenticación |
| Almacenamiento | Base de Datos |