**Nombre: Sebastian Orlando Manchabajoy Rosero**

**ID: 899013**

**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.

**FUNCIONALIDADES**

**Pacientes:**

* Registrar e iniciar sesión.
* Buscar médicos por especialidad y disponibilidad
* Reservar. modificar o cancelar citas
* Ver historial de consultas

**Medicos:**

* Iniciar sesión como medico
* Gestionar agenda (crear, modificar o cancelar horarios)
* Ver citas programadas
* Registrar observaciones medicas

**Administradores:**

* Iniciar sesión como administrador
* Crear, editar, eliminar usuario de pacientes y médicos
* Supervisar y controlar el sistema

**COMPONENTES**

* **Gestion de usuarios:** registro de pacientes, médicos y administradores
* **Agenda medica:** horarios disponibles, gestión de citas, calendario medico
* **Reserva de citas:** búsqueda de disponibilidad, reserva, modificación y cancelación de citas
* **Gestion administrativa:** administración de usuarios y estadísticas del sistema
* **Historial medico:** visualizar citas pasadas y observaciones medicas
* **Notificaciones:** confirmación de cita, recordatorios por correo o SMS

**Diagrama

El contenido generado por IA puede ser incorrecto.DIGRAMA DE COMPONENTES**

**JUSTIFICACION**

El sistema tiene como componente central la “Gestión de Usuarios” ya que cualquier acción como reservar, revisar la agenda o acceder al historial médico, dependen de usuarios registrados y autenticados. También se separaron responsabilidades lo que favorece la modularidad y facilita futuras ampliaciones. Al tener servicios independientes se puede escalar solos módulos que lo requieran.

**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.

**FUNCIONALIDADES**

**Usuarios:**

* Registrar e iniciar sesión
* Navegar por restaurantes y menus
* Realizar pedidos y pagos
* Rastrear el estado del pedido

**Restaurante**

* Iniciar sesión como restaurante
* Crear, editar y eliminar productos (menú)
* Recibir y aceptar pedidos
* Actualizar estado del pedido

**Repartidores**

* Registarse e iniciar sesión como repartidor
* Ver pedidos asignados
* Ver ubicación del cliente
* Marcar pedido como entregado

**Administrador del sistema**

* Iniciar sesión como administrador
* Gestionar usuarios, restaurantes y repartidores
* Supervisar el rendimiento general
* Resolver inconvenientes

**COMPONENTES**

* **Pagina web:** frontend.
* **Gestion de usuarios**: registro e inicio de sesión según el rol (cliente, restaurante, repartidor, administrador).
* **Catalogo de restaurantes:** menus, productos, generar una búsqueda por tipo de productos.
* **Gestion de pedidos:** flujo completo del pedido (creacion, estado, entrega).
* **Gestion de entregas:** asignación de repartidores y seguimiento de ubicación.
* **Pagos:** procesamiento de pagos.
* **Facturacion:** generación de facturas.
* **Notificaciones:** alertas de estado del pedido, además de promociones o mensajes.

**DIAGRAMA DE COMPONENTES**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Justificación**

Se hizo un sistema centralizado donde el módulo central es la “Gestión de pedidos” el cual coordina todo el sistema, esto se decidió ya que se reduce la complejidad y permite que otros módulos se mantengan desacoplados; además se hace separación de responsabilidades donde cada componente tiene una función bien definida, lo cual sigue el principio de alta cohesión y bajo acoplamiento, lo que facilita el mantenimiento y escabilidad.

**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.

**COMPONENTES Y FUNCIONALIDADES:**

**Gestion de Usuarios:**

* Registro y administración de estudiantes y docentes
* Asignacion de permisos

**Catalogo de libros:**

* Consulta de títulos, autores y disponibilidad
* Clasificaciones por genero literario y nivel educativo

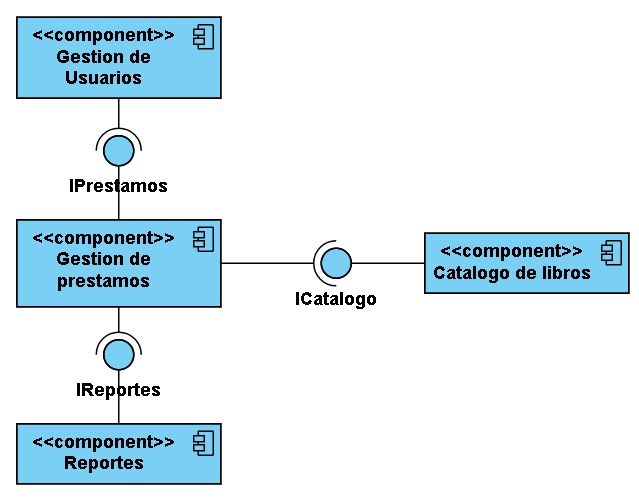
**Gestion de prestamos**

* Solicitar, renovar y devolver libros
* Control de fechas limites y sanciones

**Reportes**

* Estadisticas de uso, cuales son los libros mas prestados
* Envio de datos para administración

**DIAGRAMA DE COMPONENTES**



**JUSTIFICACIÓN**

Cada componente tiene una función clara, lo que facilita entender, desarrollar y probar cada módulo por separado, además que se vuelven componentes fáciles de escalar individualmente. También con esta estructura es sencillo añadir un API o una interfaz web.