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

* El paciente debe poder acceder al sistema mediante un inicio de sesión.
* El paciente debe poder visualizar la disponibilidad de los médicos en tiempo real.
* El paciente debe poder reservar una cita médica, especificando la fecha, hora y el médico deseado.
* El paciente debe poder modificar o cancelar su cita, según las políticas del centro médico.
* El paciente debe recibir una confirmación de la cita por correo electrónico o SMS.
* El médico debe poder iniciar sesión en su cuenta personal.
* El médico debe poder visualizar su agenda de citas, con la opción de añadir, modificar o eliminar citas.
* El médico debe recibir notificaciones sobre nuevas reservas o cambios en sus citas programadas.
* El médico debe poder marcar su disponibilidad para nuevas citas o hacer ajustes en la agenda (por ejemplo, tiempo bloqueado).
* El médico debe poder acceder a la historia clínica del paciente durante la consulta (si está habilitada esta opción).
* El administrador debe poder gestionar las cuentas de los pacientes, médicos y otros administradores.
* El administrador debe poder añadir, editar o eliminar usuarios (pacientes, médicos, personal administrativo).
* El administrador debe poder asignar roles y permisos a los usuarios, diferenciando entre paciente, médico y administrador.
* El administrador debe poder visualizar un resumen de las citas y la actividad del sistema.
* El administrador debe poder ver todas las consultas reservadas por los pacientes y gestionarlas.
* El administrador debe poder modificar la información de las consultas (asignación de médicos, horarios, etc.).
* El administrador debe poder cancelar consultas, si es necesario.
* El administrador debe generar reportes de consultas realizadas, pacientes atendidos, y médicos disponibles.
* Los pacientes y médicos deben recibir notificaciones sobre las citas confirmadas, modificadas o canceladas.
* El sistema debe enviar recordatorios de citas a los pacientes y médicos 24 horas antes de la consulta.
* El sistema debe permitir al paciente y al médico recibir notificaciones por cambios o alertas en la agenda.
* El paciente debe poder consultar su historial médico, siempre que esté habilitado este acceso en el sistema.
* El historial médico debe incluir detalles de consultas pasadas, diagnósticos, recetas, y tratamientos realizados.
* El sistema debe permitir al paciente descargar o imprimir su historial médico si es necesario.
* El sistema debe ofrecer un acceso seguro para pacientes, médicos y administradores mediante autenticación de usuario (usuario y contraseña).
* El sistema debe permitir restablecer contraseñas en caso de olvido.
* El sistema debe tener medidas de seguridad para proteger la información personal y médica de los usuarios.

**Análisis Breve de las Decisiones de Diseño**

* **Modularización**: El sistema se divide en componentes independientes (autenticación, perfil de usuario, y gestión de citas) para facilitar el mantenimiento y escalabilidad, permitiendo actualizaciones sin afectar el resto del sistema.
* **Interfaces Claras**: Las interfaces "Logeo" y "Reserva Citas" garantizan una comunicación eficiente entre los componentes, facilitando la integración y mejorando la seguridad al mantener la autenticación separada del resto de los procesos.
* **Escalabilidad y Flexibilidad**: Este diseño modular permite añadir nuevas funcionalidades (como historial médico o notificaciones) fácilmente, sin interrumpir el sistema actual.
* **Seguridad**: Separar la autenticación de los perfiles de usuario mejora la seguridad y permite añadir nuevos métodos de autenticación sin complicar el sistema.

Grafico

Diagrama

El contenido generado por IA puede ser incorrecto.

1. 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.

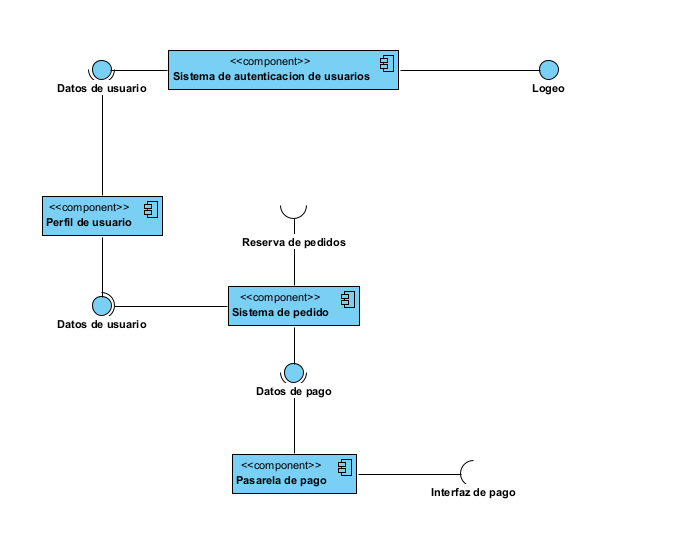
Requisitos Funcionales

* Los usuarios deben poder registrarse en la aplicación (móvil/web) con un correo electrónico o mediante redes sociales.
* +Los usuarios deben poder iniciar sesión en la aplicación con sus credenciales (correo electrónico/contraseña).
* Los usuarios deben poder ver una lista de restaurantes cercanos o disponibles.
* Los usuarios deben poder filtrar restaurantes por tipo de comida, precios o valoraciones.
* Los usuarios deben poder ver el menú completo de los restaurantes, con fotos, descripciones y precios.
* Los usuarios deben poder seleccionar productos del menú y agregarlos al carrito.
* Los usuarios deben poder modificar la cantidad de los productos en el carrito.
* Los usuarios deben poder aplicar cupones de descuento si están disponibles.
* Los usuarios deben poder pagar el pedido a través de diversos métodos de pago (tarjetas de crédito/débito, PayPal, etc.).
* Los usuarios deben recibir una confirmación del pago y del pedido.
* Los usuarios deben poder ver el estado de su pedido en tiempo real (por ejemplo: en preparación, en tránsito, entregado).
* Los usuarios deben recibir una notificación cuando el pedido esté en camino y cuando haya sido entregado.
* Los usuarios deben poder calificar y dejar comentarios sobre el restaurante y el servicio de entrega después de recibir su pedido.
* Los restaurantes deben poder registrarse en la plataforma y completar su perfil (nombre, ubicación, información de contacto).
* Los restaurantes deben poder agregar, eliminar o modificar los productos de su menú.
* Los restaurantes deben poder gestionar los precios, descripciones y fotos de los productos.
* Los restaurantes deben poder ver los pedidos que reciben en tiempo real.
* Los restaurantes deben poder cambiar el estado del pedido (por ejemplo: en preparación, listo para entregar).
* Los restaurantes deben poder configurar sus horarios de apertura y cierre.
* Los restaurantes deben poder configurar los tiempos estimados de entrega de los pedidos.

**Análisis Breve para el Ejercicio #2**

* **Modularización**: El sistema se divide en módulos independientes para **clientes**, **restaurantes**, **repartidores** y **administradores**, lo que facilita la actualización de funcionalidades y la integración de nuevos servicios sin afectar las partes existentes del sistema. Esto mejora la **gestión** y la **eficiencia operativa**.
* **Interacción y comunicación**: La comunicación entre los módulos se realiza mediante **HTTP/REST** y **notificaciones en tiempo real**, lo que garantiza que los usuarios, restaurantes y repartidores estén informados en todo momento sobre el estado de los pedidos. Esta interacción fluida mejora la **experiencia del usuario** y optimiza el rendimiento del sistema
* **Sistema de pagos y seguridad**: El sistema de **pagos seguros** es esencial para garantizar transacciones fiables. La plataforma debe aceptar métodos como tarjetas de crédito/débito y servicios de pago como **PayPal**. Además, se implementarán estrictas medidas de **seguridad** para proteger la información personal y financiera de todos los actores involucrados.
* **Optimización para repartidores**: El diseño incluye la integración de un sistema de **navegación en tiempo real** para los repartidores, lo que optimiza las rutas de entrega y mejora la **eficiencia operativa**. Esto no solo reduce el tiempo de entrega, sino que también mejora la **satisfacción del cliente**.
* **Escalabilidad**: El sistema está diseñado para crecer con facilidad. Los nuevos servicios, como la integración de más restaurantes, nuevas funcionalidades para los usuarios o incluso nuevos tipos de usuarios (por ejemplo, comercios adicionales), pueden ser implementados sin reestructurar el sistema principal.
* **Flexibilidad y mantenimiento**: La arquitectura modular facilita la integración de nuevas funcionalidades, como nuevas formas de pago o características de análisis de datos, lo que permite que el sistema se mantenga actualizado y adaptado a las necesidades del mercado.

Grafico



1. 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

 **Registro e inicio de sesión de usuarios**: Los usuarios deben poder registrarse en el sistema y acceder mediante un inicio de sesión utilizando credenciales (nombre de usuario y contraseña).

 **Gestión de usuarios**: Los administradores deben poder gestionar las cuentas de los usuarios (crear, editar, eliminar). También deben poder asignar roles (por ejemplo, estudiante, profesor, administrador) con permisos específicos.

 **Gestión de catálogo de libros**: Los administradores deben poder añadir, editar o eliminar libros del catálogo, incluyendo detalles como título, autor, género, disponibilidad, y ubicación en la biblioteca.

 **Préstamos de libros**: Los usuarios deben poder solicitar préstamos de libros disponibles en el catálogo. Los préstamos deben tener un período definido y el sistema debe permitir renovaciones, si es aplicable.

 **Control de fechas de vencimiento y renovaciones**: El sistema debe permitir a los usuarios y administradores consultar las fechas de vencimiento de los préstamos y realizar renovaciones si no hay reservas pendientes para el libro.

 **Devolución de libros**: Los usuarios deben poder registrar la devolución de los libros en el sistema, lo que actualiza la disponibilidad del libro en el catálogo.

 **Historial de préstamos**: Los usuarios deben poder ver un historial de todos los libros que han prestado, incluyendo fechas de préstamo, vencimiento y devolución.

 **Notificación de vencimiento**: El sistema debe enviar notificaciones a los usuarios antes de que sus préstamos lleguen a su fecha de vencimiento.

 **Reservas de libros**: Los usuarios deben poder reservar libros que están actualmente prestados, y el sistema debe notificarles cuando el libro esté disponible para préstamo.

 **Gestión de reportes**: Los administradores deben poder generar reportes sobre el estado de los libros, número de préstamos, libros más populares, etc. Los reportes deben ser exportables en formatos como CSV o PDF.

 **Búsqueda avanzada de libros**: Los usuarios deben poder buscar libros en el catálogo utilizando filtros como título, autor, género, disponibilidad, etc.

 **Interfaz de usuario intuitiva**: El sistema debe proporcionar una interfaz fácil de usar para la gestión del catálogo, préstamos, usuarios, y reportes.

 **Seguridad y acceso controlado**: El sistema debe tener control de acceso basado en roles para garantizar que los administradores, usuarios y personal tengan acceso solo a las funcionalidades y datos necesarios según su rol.

Grafico

Diagrama

El contenido generado por IA puede ser incorrecto.