Modularización y Componentización

Nombre del estudiante : Oskar Julian Ortiz Ortiz   
Curso : 4 Semestre   
Fecha: 16/09/2025

# Introducción

La modularización y la componentización permiten dividir un sistema en partes independientes, con responsabilidades bien definidas, que se comunican mediante interfaces. Esto facilita el mantenimiento, la escalabilidad y la integración de los sistemas.

# Ejercicio 1: Centro Médico

## Funcionalidades

|  |  |
| --- | --- |
| Funcionalidad | Descripción |
| Reservar cita | El paciente puede reservar una cita validando disponibilidad. |
| Cancelar cita | El paciente cancela una cita previamente reservada. |
| Gestionar agenda | El médico gestiona horarios y disponibilidad. |

## Componentes

* - PatientPortal
* - AppointmentService
* - ScheduleService
* - DoctorPortal
* - AuthService
* - NotificationService
* - AdminService
* - EMR/PatientRecords

## Interfaces

|  |  |  |
| --- | --- | --- |
| Componente | Interfaz (API) | Métodos principales |
| AppointmentService | AppointmentAPI | CreateBooking(), CancelBooking(), GetHistory() |
| ScheduleService | ScheduleAPI | CheckAvailability(), UpdateAvailability() |
| AuthService | AuthAPI | Authenticate(), Authorize() |

## Diagrama

[Aquí insertar diagrama del ejercicio]

## Análisis y justificación

Explicación breve de las decisiones de diseño.

# Ejercicio 2: App Delivery

## Funcionalidades

|  |  |
| --- | --- |
| Funcionalidad | Descripción |
| Realizar pedido | El usuario crea un pedido desde la aplicación. |
| Pagar en línea | El usuario realiza el pago de forma digital. |
| Actualizar estado | El restaurante actualiza el estado del pedido. |
| Entrega | El repartidor acepta y confirma la entrega. |

## Componentes

* - Mobile/Web Client
* - OrderService
* - PaymentService
* - RestaurantPortal
* - DeliveryService
* - NotificationService
* - AuthService
* - CatalogService
* - GeoService/Maps

## Interfaces

|  |  |  |
| --- | --- | --- |
| Componente | Interfaz (API) | Métodos principales |
| OrderService | OrderAPI | PlaceOrder(), GetOrderStatus(), CancelOrder() |
| PaymentService | PaymentAPI | InitiatePayment(), ConfirmPayment(), Refund() |
| DeliveryService | DeliveryAPI | AssignDelivery(), UpdateLocation(), ConfirmDelivery() |
| CatalogService | CatalogAPI | SearchMenu(), GetItemDetails() |
| GeoService/Maps | GeoAPI | GetRoute(), TrackLocation() |

## Diagrama

[Aquí insertar diagrama del ejercicio]

## Análisis y justificación

Explicación breve de las decisiones de diseño.

# Ejercicio 3: Sistema de Biblioteca

## Funcionalidades

|  |  |
| --- | --- |
| Funcionalidad | Descripción |
| Buscar libro | El estudiante consulta el catálogo de libros. |
| Prestar libro | El bibliotecario registra un préstamo. |
| Devolver libro | El bibliotecario registra una devolución. |
| Generar reportes | El bibliotecario genera reportes de uso. |

## Componentes

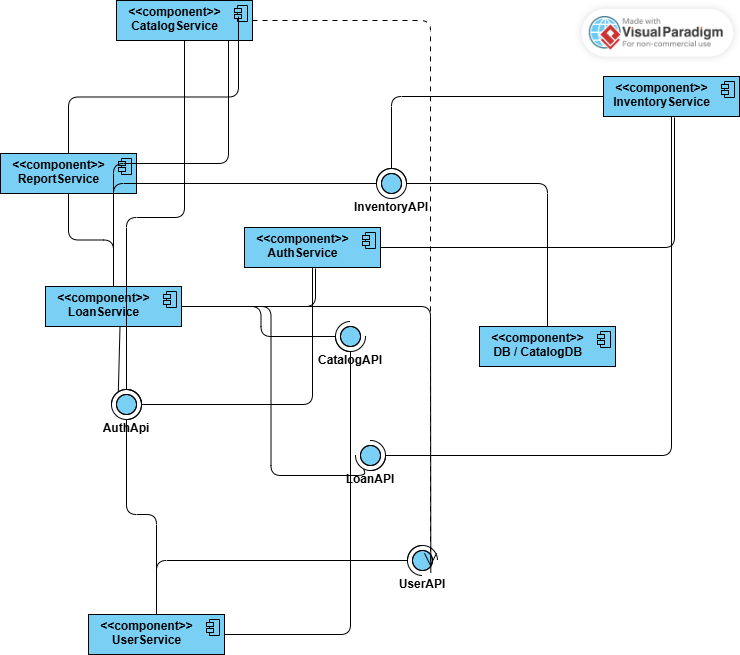
* - CatalogService
* - LoanService
* - UserService
* - InventoryService
* - ReportService
* - AuthService
* - NotificationService
* - DB/CatalogDB

## Interfaces

|  |  |  |
| --- | --- | --- |
| Componente | Interfaz (API) | Métodos principales |
| CatalogService | CatalogAPI | SearchBooks(), GetBookDetails() |
| LoanService | LoanAPI | Checkout(), Return(), ReserveBook() |
| UserService | UserAPI | GetUserInfo(), ValidateUser() |
| ReportService | ReportAPI | GenerateLoanReport(), GenerateOverdueReport() |
| InventoryService | InventoryAPI | UpdateCopyStatus() |

## Diagramas

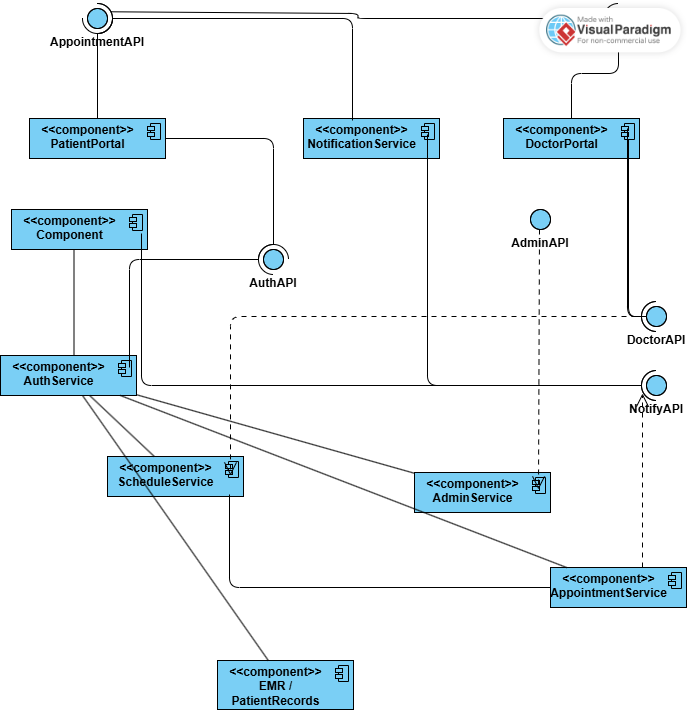
-Biblioteca



# -AppDelivery

# 

-CentroMedico



# Conclusión

La modularización y la componentización permiten que los sistemas sean más mantenibles, escalables y fáciles de integrar con servicios externos. Al separar responsabilidades en componentes independientes, se facilita la reutilización y el despliegue individual.