

Diseño de Microservicios SOLID-SM para TaskHub

Omar Yesid Hernández Sotelo

Abril 2025

Índice

1. ProjectService	2
2. AuthService	4
3. DocumentService	5
4. NotificationService	6
5. ExternalToolsService	7
6. API Gateway	8

1. ProjectService

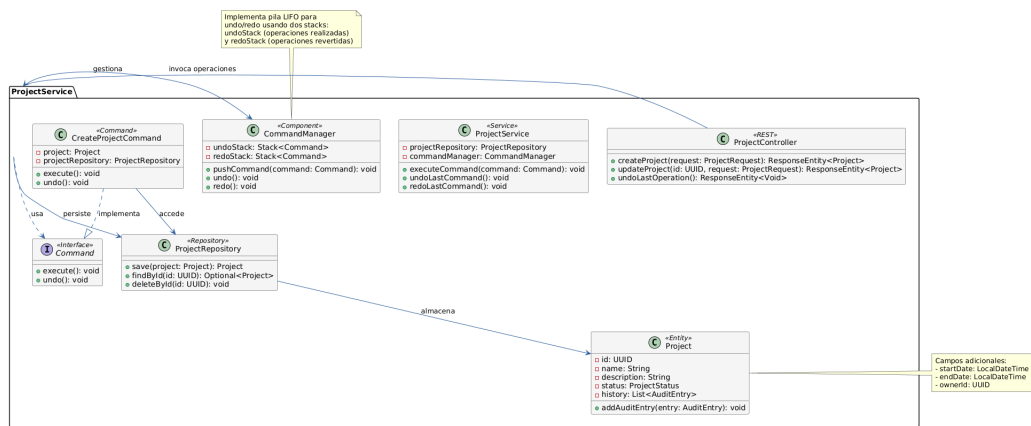


Figura 1: Caption

FASE 1: Análisis y Diseño Funcional

- **Bounded Context:** Gestión del ciclo de vida completo de proyectos.
- **Entrada Principal:** Solicitudes RESTful con autenticación JWT.
- **Funcionalidad:**
 - CRUD de proyectos
 - Operaciones undo/redo (Patrón Command)
 - Integración con DocumentService y NotificationService
- **Modelo de Negocio:** Proyectos con estados: ACTIVO, INACTIVO, COMPLETADO.
- **Especificación Funcional:** OpenAPI 3.0
- **Contratos REST:** Endpoints y esquemas JSON definidos
- **Modelo de Dominio:** Diagrama de clases con patrones de diseño

FASE 2: Diseño Técnico y de Interfaces

- **API REST:** Versionamiento con /v1/projects
- **Tecnologías:** MongoDB, JWT, RabbitMQ
- **Patrones:** Command para reversibilidad, Observer para eventos
- **Documentación:** Swagger UI + OpenAPI
- **Circuit Breaker** para manejo de errores

2. AuthService

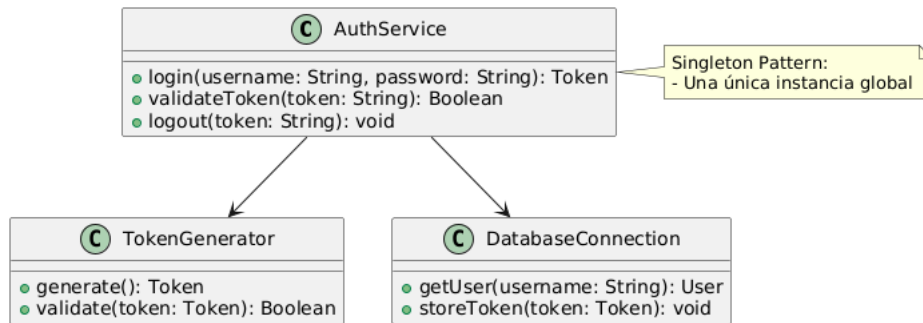


Figura 2: Diseño del Auth Service

FASE 1: Análisis y Diseño Funcional

- **Bounded Context:** Gestión de credenciales y tokens.
- **Entrada Principal:** Credenciales de usuario.
- **Funcionalidad:** login, logout, validación de tokens.
- **Modelo de Negocio:** Usuarios y roles
- **Contratos REST:** login, validateToken, logout

FASE 2: Diseño Técnico y de Interfaces

- **Tecnologías:** PostgreSQL, JWT
- **Patrón:** Singleton
- **Interfaces:** login, validateToken, logout

3. DocumentService

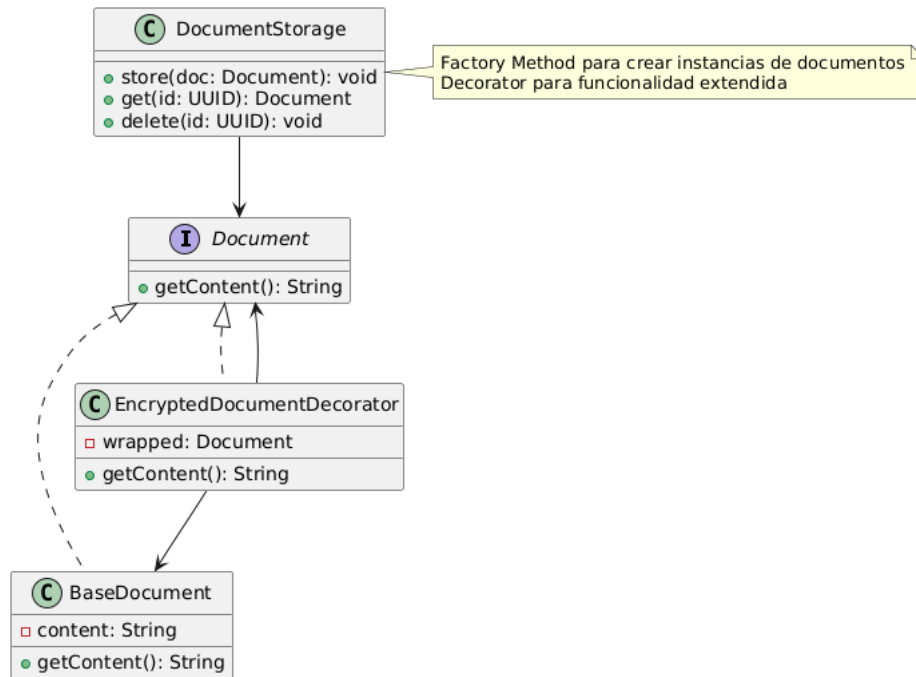


Figura 3: Diseño del Document Service

FASE 1: Análisis y Diseño Funcional

- **Bounded Context:** Manejo y versión de documentos
- **Entrada Principal:** Archivos y metadatos
- **Funcionalidad:** CRUD de documentos, integración con proyectos

FASE 2: Diseño Técnico y de Interfaces

- **Tecnologías:** MongoDB
- **Patrones:** Factory Method y Decorator
- **Interfaces:** uploadDocument, getDocument, deleteDocument

4. NotificationService

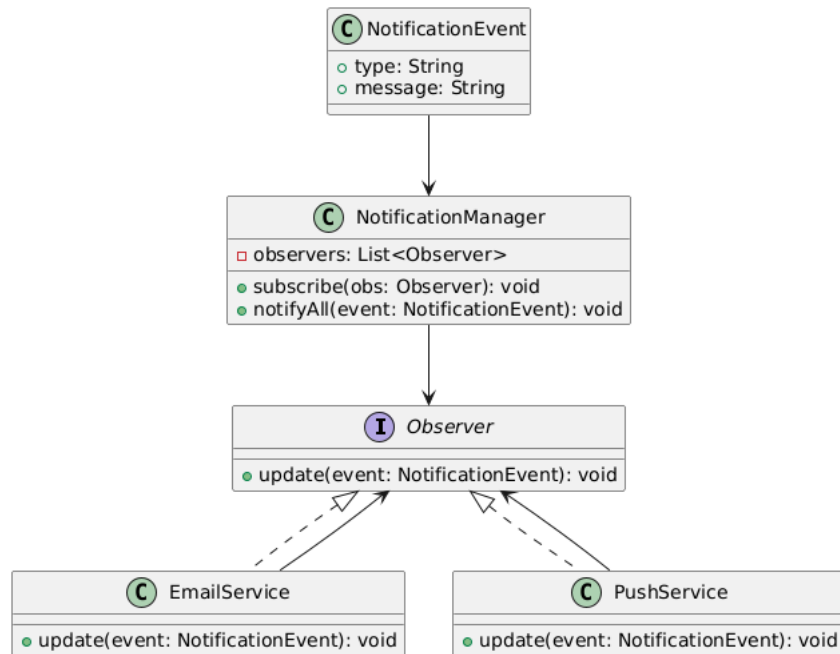


Figura 4: Diseño del Notification Service

FASE 1: Análisis y Diseño Funcional

- **Bounded Context:** Enviar notificaciones
- **Entrada Principal:** Eventos desde otros servicios
- **Funcionalidad:** Email y notificaciones push

FASE 2: Diseño Técnico y de Interfaces

- **Tecnologías:** RabbitMQ
- **Patrón:** Observer
- **Interfaces:** `sendEmailNotification`, `sendPushNotification`

5. ExternalToolsService

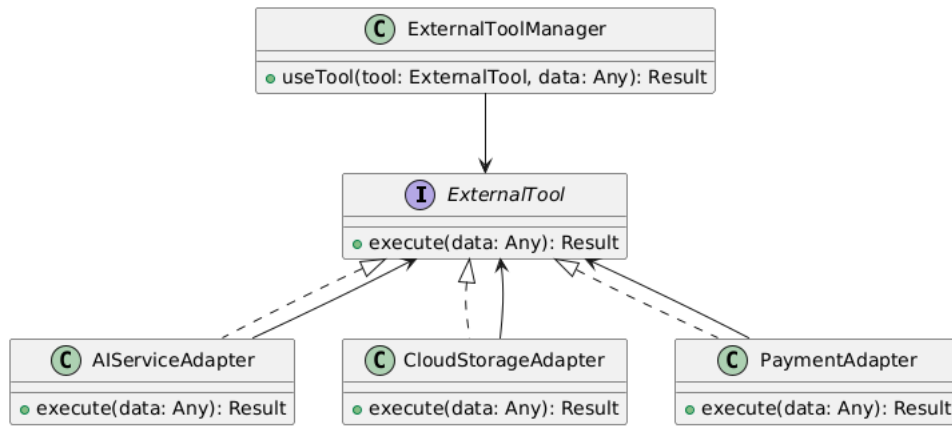


Figura 5: Diseño de external tools service

FASE 1: Análisis y Diseño Funcional

- **Bounded Context:** Integración con servicios externos
- **Funcionalidad:** IA, almacenamiento, pagos

FASE 2: Diseño Técnico y de Interfaces

- **Patrón:** Adapter
- **Interfaces:** analyzeContent, uploadFile, processPayment

6. API Gateway

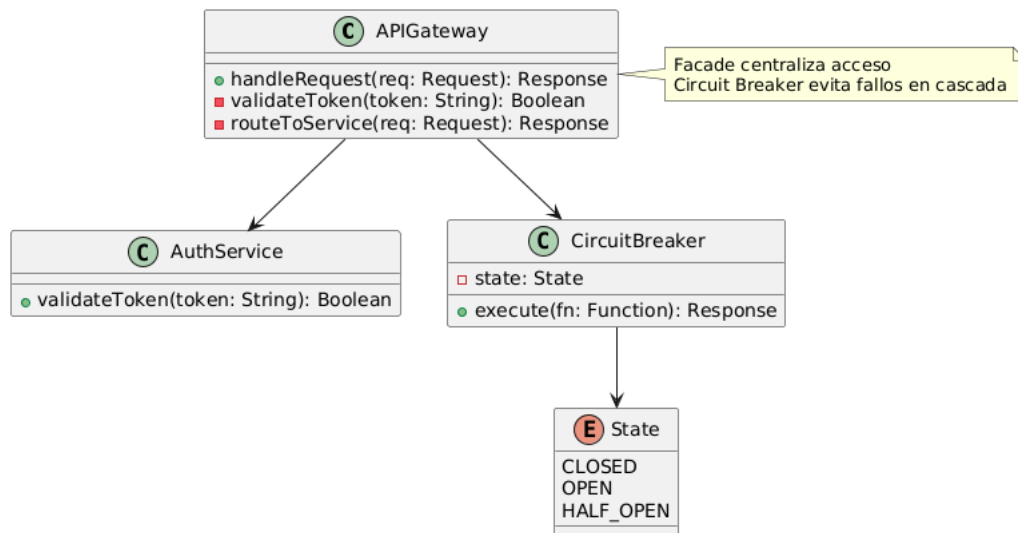


Figura 6: Diseño del API Gateway

FASE 1: Análisis y Diseño Funcional

- **Bounded Context:** Punto de entrada único
- **Funcionalidad:** Validar token, enrutar peticiones

FASE 2: Diseño Técnico y de Interfaces

- **Patrones:** Facade y Circuit Breaker
- **Interfaces:** `handleRequest`, `routeToService`, `validateToken`