

Modelado de Arquitecturas de Software (C4 + UML)

1. Análisis Inicial

Requerimientos Funcionales

El sistema debe gestionar las siguientes funcionalidades:

- **Gestión de Cursos y Clases:** Creación, edición y visualización de cursos y clases para administradores y estudiantes.
- **Inscripción:** Proceso de inscripción de estudiantes con validación de prerequisites y disponibilidad.
- **Gestión de Pagos:** Generación de facturas, procesamiento de pagos en línea a través de una pasarela externa y registro de transacciones.
- **Gestión de Usuarios y Roles:** Autenticación y autorización para estudiantes, profesores y administrativos.
- **Notificaciones:** Envío de notificaciones automáticas vía correo electrónico y/o SMS.
- **Reportes:** Generación de informes sobre inscripciones, pagos y ocupación.

Atributos de Calidad Prioritarios

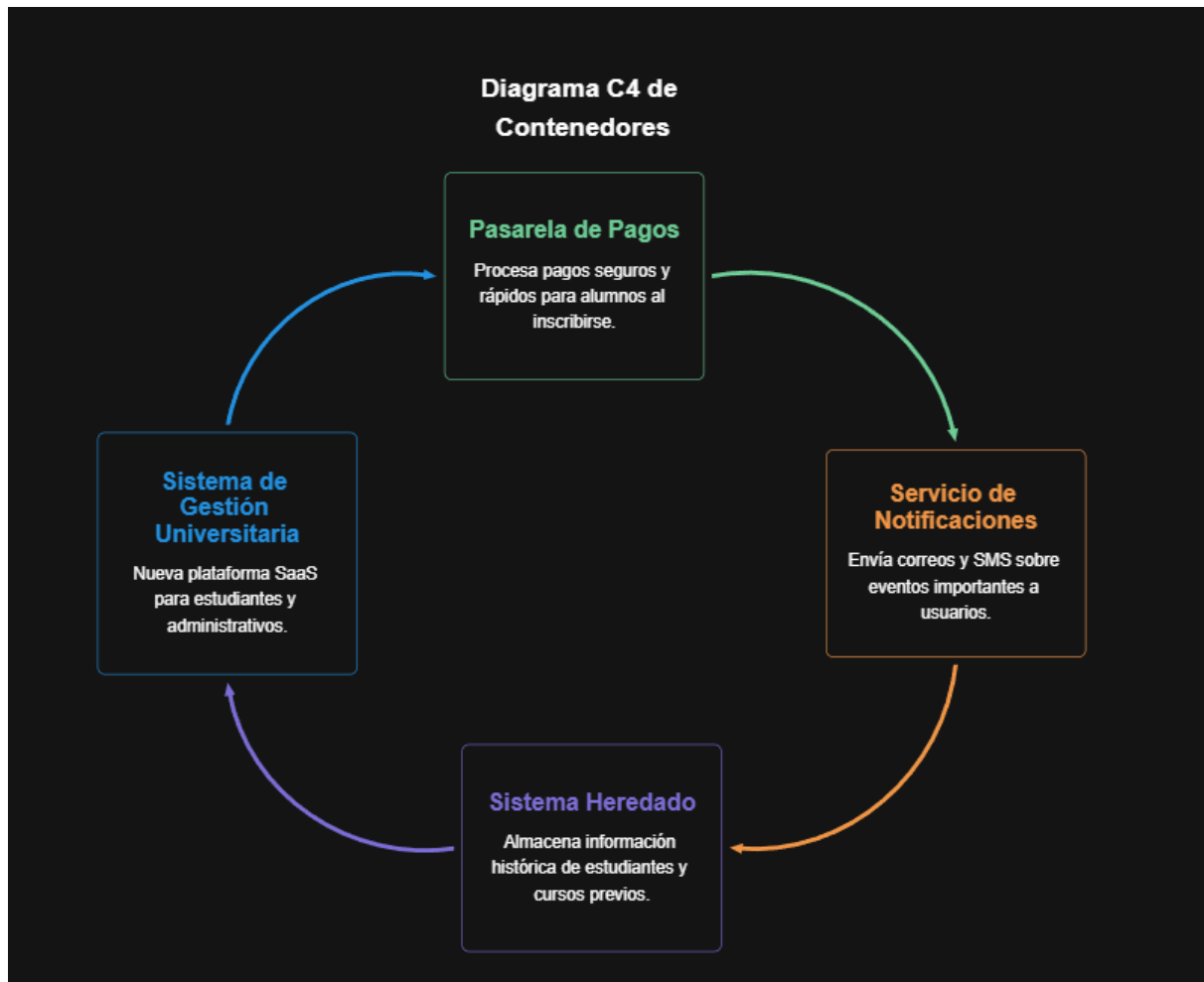
Los atributos de calidad más importantes para esta evolución de la arquitectura son:

- **Escalabilidad:** El sistema debe escalar horizontalmente para soportar picos de carga durante los periodos de inscripción.
- **Disponibilidad:** Alta disponibilidad y tolerancia a fallos para garantizar el acceso 24/7.
- **Seguridad:** Protección de datos sensibles de usuarios y pagos, cumpliendo con los estándares de seguridad de la industria.
- **Mantenibilidad:** La arquitectura modular debe facilitar el desarrollo, despliegue y actualización de funcionalidades de manera independiente.

2. Modelado Arquitectónico (Modelo C4)

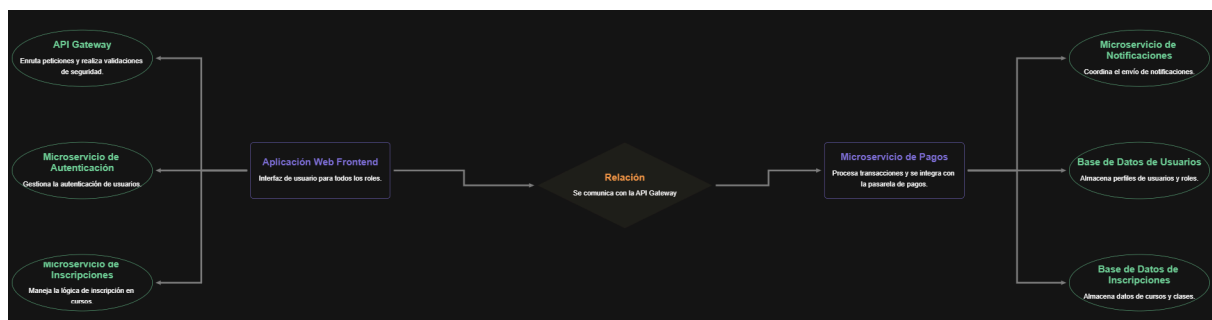
Nivel 1: Diagrama de Contexto

Muestra el sistema en el centro, con sus interacciones clave con usuarios y sistemas externos.



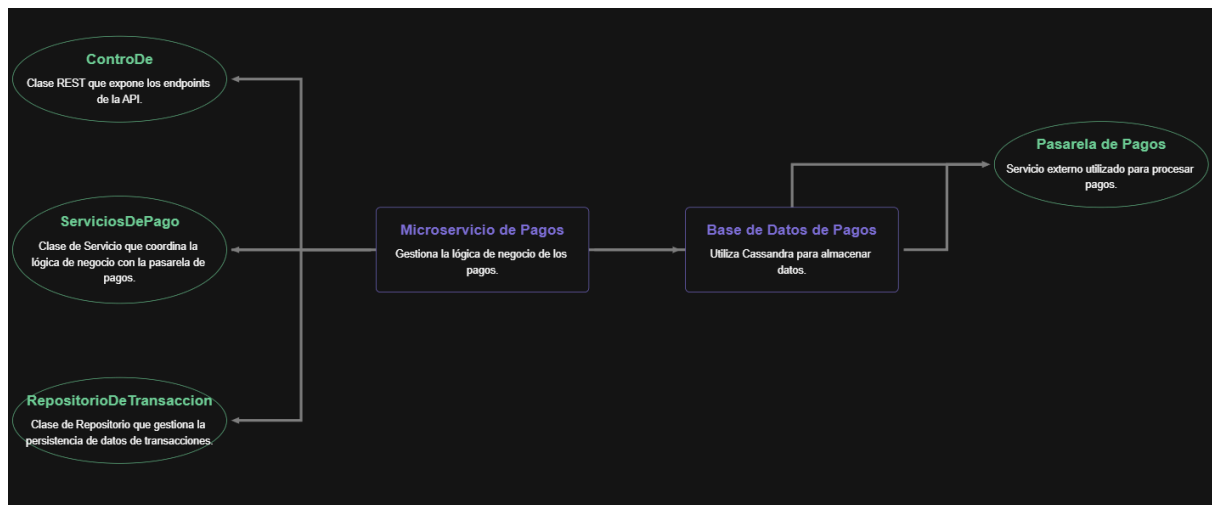
Nivel 2: Diagrama de Contenedores

Descompone el sistema en aplicaciones, bases de datos y microservicios.



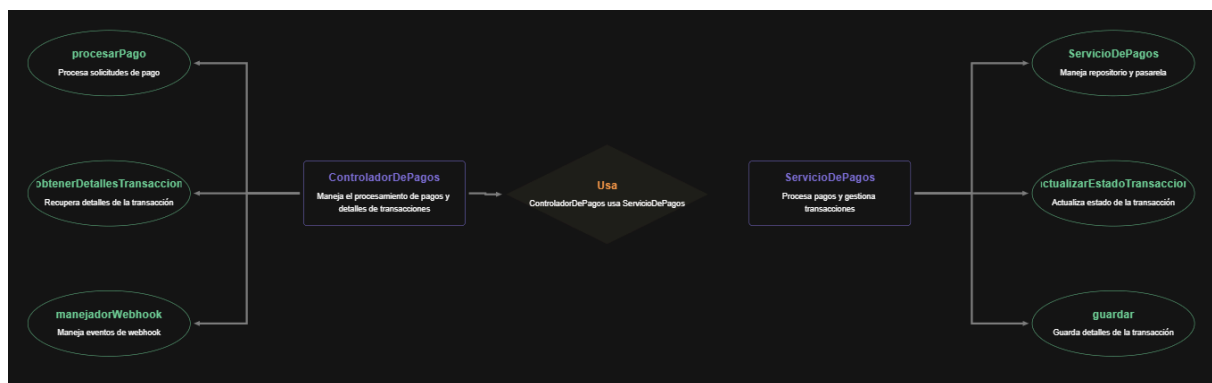
Nivel 3: Diagrama de Componentes (Microservicio de Pagos)

Detalla los componentes internos del Microservicio de Pagos.



Nivel 4: Diagrama de Código (UML de un Microservicio)

Un diagrama de clases UML para mostrar la estructura de un microservicio.



3. Diagrama de Despliegue UML

Muestra la arquitectura en un entorno de nube, usando Google Cloud Platform como ejemplo.



4. Decisiones Arquitectónicas (ADR)

ADR 001: Arquitectura de Microservicios

- **Decisión:** Migrar del sistema monolítico a una arquitectura de microservicios.
- **Justificación:** El sistema monolítico actual presenta problemas de escalabilidad y mantenimiento. La arquitectura de microservicios permitirá escalar cada componente de forma independiente para manejar picos de demanda y facilitará la entrega continua y el desarrollo ágil por equipos autónomos.

ADR 002: Persistencia Políglota

- **Decisión:** Utilizar diferentes tipos de bases de datos para cada microservicio (**polyglot persistence**).
- **Justificación:** Cada dominio de datos tiene necesidades distintas. PostgreSQL es ideal para la estructura relacional de inscripciones y cursos, MongoDB ofrece flexibilidad para los perfiles de usuario, y Cassandra es óptima para el alto volumen de escrituras del historial de pagos. Esta decisión maximiza el rendimiento y la eficiencia de cada servicio.

ADR 003: Autenticación con JSON Web Tokens (JWT)

- **Decisión:** Implementar la autenticación a través de JWT en lugar de un enfoque de sesión centralizada.

- **Justificación:** En una arquitectura de microservicios, la autenticación debe ser "sin estado" (stateless). JWT permite al API Gateway validar la autenticidad y los permisos de un usuario de manera rápida y sin tener que consultar a un servicio de autenticación en cada petición, lo que reduce la latencia y mejora la escalabilidad de todo el sistema.