



Universidad Tecnológica de Tehuacán

Grado y Grupo: 7°B

Materia: Arquitectura de Software

Equipo:

- Irvin Isael Martínez Alejo
- Osbaldo Álvarez Martínez
- Ahabram Moreno Vasquez
- Felix Ivan Garcia Flores
- Kevin Rojas Martínez

Fecha: 21 / 09 / 2025



Universidad Tecnológica de Tehuacán

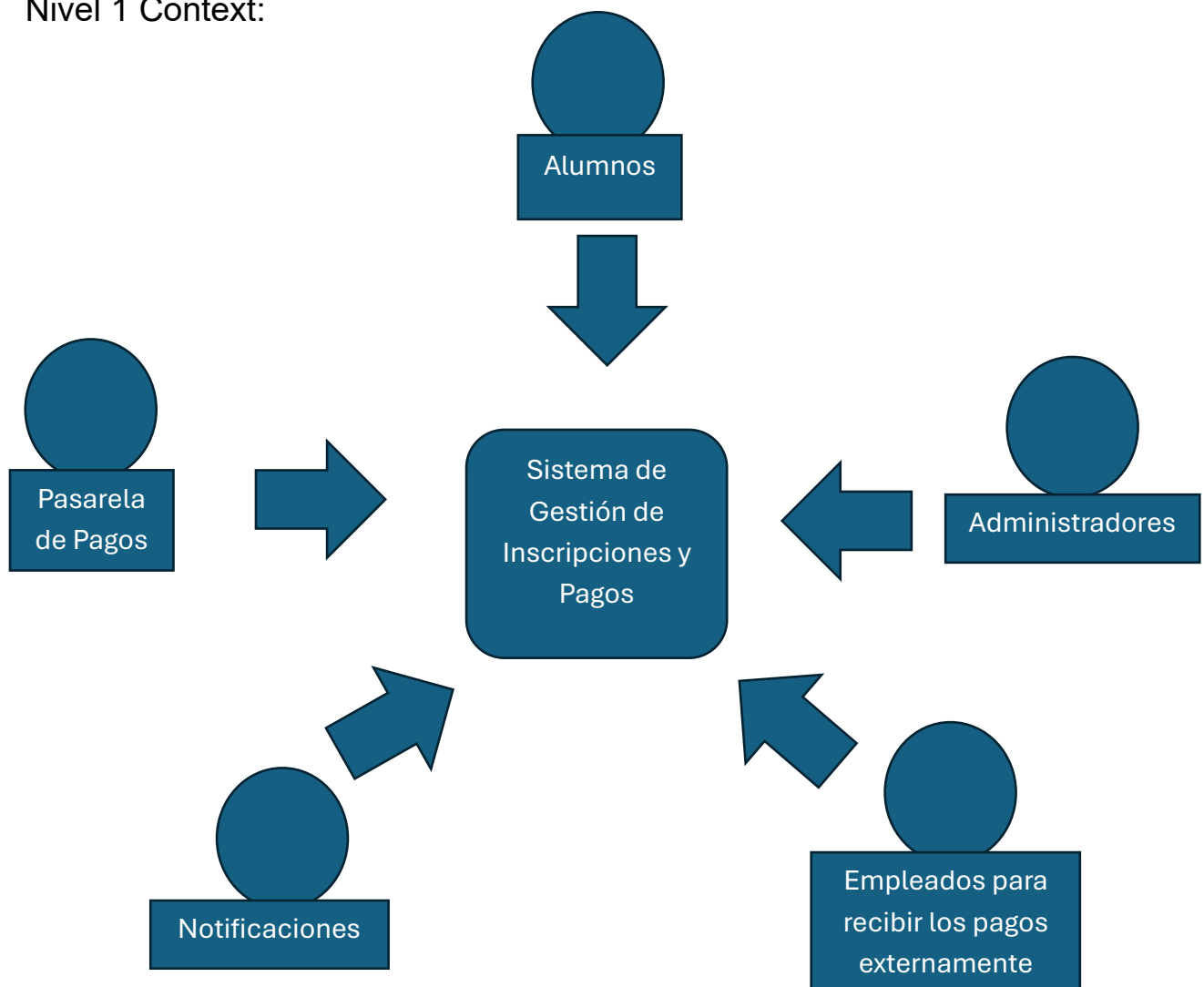


## Actividad Semana 3: Modelado de Arquitecturas de Software (C4 + UML)

La universidad quiere rediseñar su sistema de gestión de inscripciones y pagos en línea. Actualmente es un sistema monolítico que presenta problemas de escalabilidad, disponibilidad y mantenibilidad. Se requiere evolucionarlo a una arquitectura SaaS escalable, con microservicios y microfrontends.

Diagramas C4 Completas: Alumnos, Administradores, Pasarela de Pagos, notificaciones, empleados para recibir los pagos (Oxxo,Banco Azteca,Electra,Bodega Aurrera,)

Nivel 1 Context:

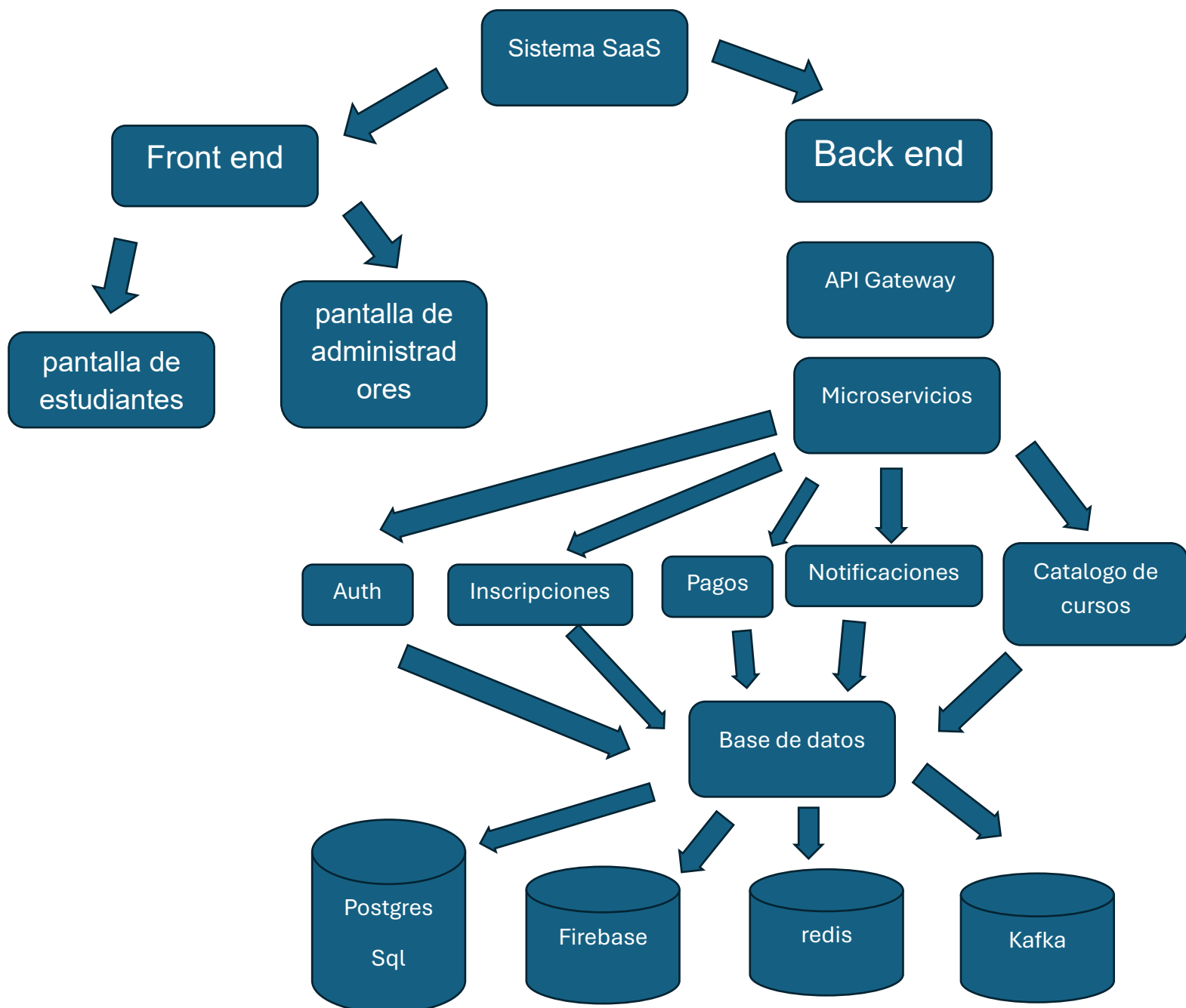


Nivel 2 Container:

Aplicaciones de Front-end: Portal de pantalla de estudiantes y portal de pantalla de administradores.

Aplicaciones de Back-end: Api Gateway, Microservicios de: inscripciones, pagos, notificaciones, pagos, catalogo de cursos y de auth.

Base de datos: Firebase,Postgres Sql, redis y Kafka



### Nivel 3 Components:

Api rest que expone los pagos, los pagos id

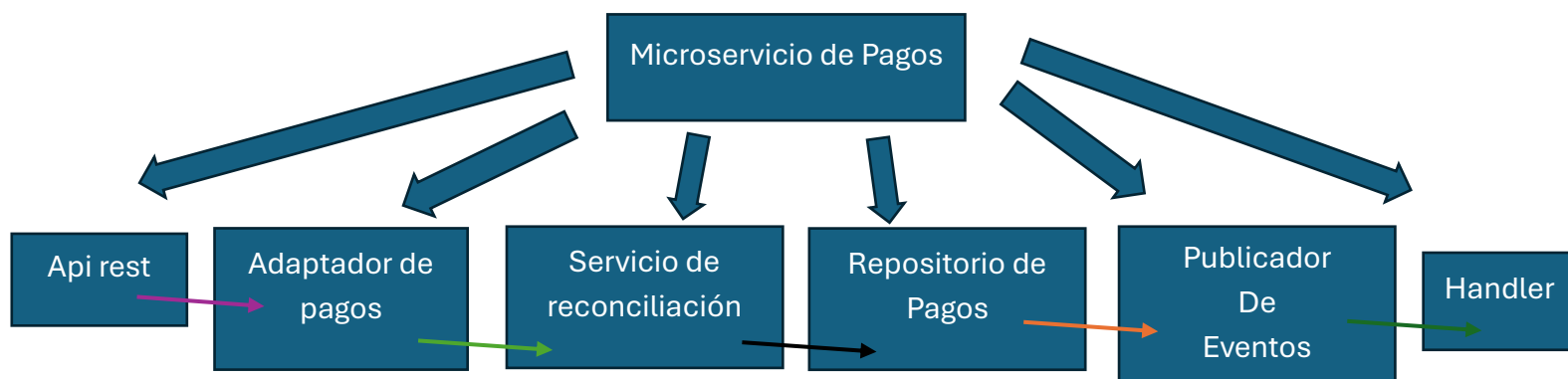
Adaptador de pasarelas

Servicio de reconciliación del estado de los pagos

Repositorio de pagos

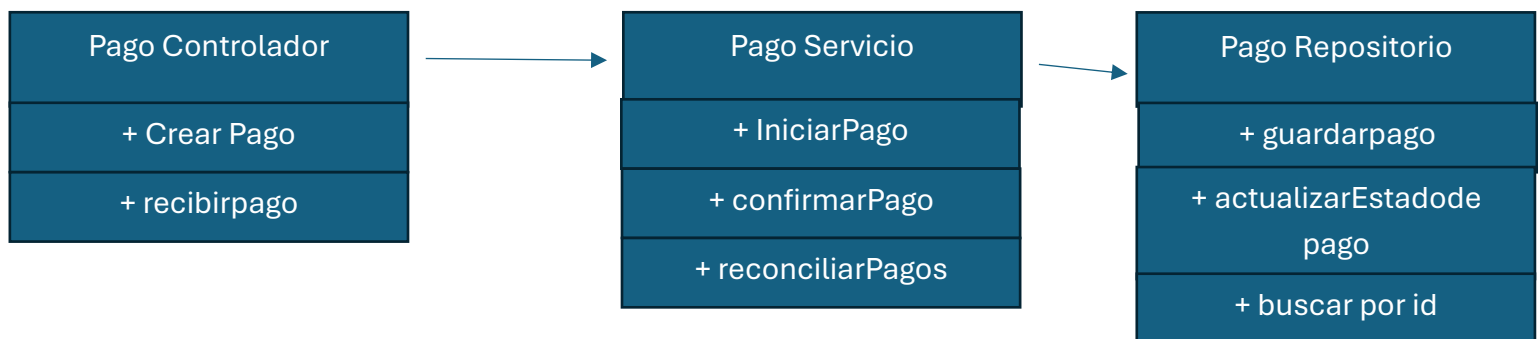
Publicador de eventos de pago creado, pagoconfirmado

Un handler donde recibe las confirmaciones de pasarela



### 4 Code:

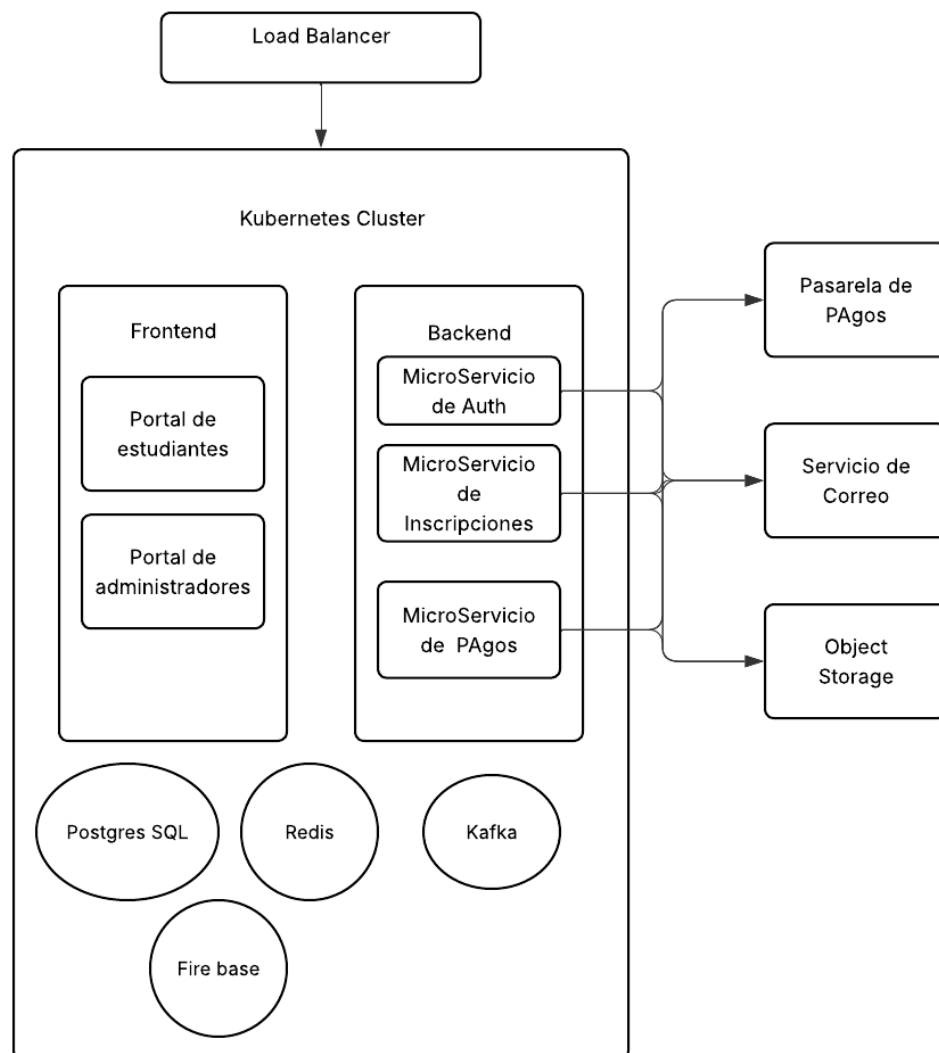
#### Diagrama UML Mircroservicio de Pagos:



## Diagrama UML de Despliegue:

Link de Lucidspark: [https://lucid.app/lucidchart/b4589131-939d-44e5-a439-d8841bddd42/edit?viewport\\_loc=-194%2C137%2C2944%2C1264%2C.Q4MUjXso07N&invitationId=invb0c7ee34-2b73-4eb4-931c-265674ce9ecb](https://lucid.app/lucidchart/b4589131-939d-44e5-a439-d8841bddd42/edit?viewport_loc=-194%2C137%2C2944%2C1264%2C.Q4MUjXso07N&invitationId=invb0c7ee34-2b73-4eb4-931c-265674ce9ecb)

## Diagrama UML de Despliegue



## **ADR 3 decisiones Clave**

### **ADR1 Migración a Microservicios vs Mantener Monolito**

El sistema actual se planteaba como un monolito, lo que complica la escalabilidad, el mantenimiento y los despliegues. Se requiere una solución que permita evolucionar y escalar módulos de manera independiente. Adoptar una arquitectura de microservicios, dividiendo el sistema en dominios clave (auth, inscripciones, pagos, notificaciones, catálogo).

Las ventajas serian:

Escalabilidad independiente por módulo.

Despliegues desacoplados con menor riesgo.

Desventajas:

Necesidad de orquestador y monitoreo distribuido, como complejidad operativa mayor

### **ADR2 Base de Datos Distribuida vs Centralizada**

El sistema maneja múltiples dominios con diferentes necesidades de datos. Una única base central podría generar cuellos de botella y problemas de rendimiento. Implementar el patrón Database per Service, cada microservicio tendrá su propia base como para los postgres para inscripciones y pagos, Redis para cacheo y Kafka para eventos

Las ventajas serian:

Aislamiento y autonomía por servicio

Escalabilidad y seguridad mejoradas

Desventajas:

Consistencia eventual entre dominios, como las inscripciones y los pagos

### ADR3 Autenticación JWT vs OAuth2/OpenID Connect

El sistema debe autenticar estudiantes y administrativos, además de integrarse en un futuro con SSO institucional o proveedores externos como Google y Microsoft. Adoptar OAuth2/OpenID Connect, utilizando JWT como access tokens

Las ventajas serian:

Tokens stateless, menor carga en servidores

Soporte a SSO e integraciones externas

Desventajas:

Complejidad en la gestión y revocación de tokens