



Indice

Contenido

[Introducción 2](#_heading=h.il7xqhcsl7b4)

[Modelado de Software 3](#_heading=h.wtl41xk6yf27)

[Vista Lógica: 4](#_heading=h.cno2h6hijjtn)

[Diagrama de clases 4](#_heading=h.g17inql1kydd)

[Vista de Desarrollo 5](#_heading=h.688aq9ixxjl2)

[Diagrama de paquetes 5](#_heading=h.jozl67bym2pm)

[Vista de Proceso: 6](#_heading=h.e27es8uqtz55)

[Diagrama de secuencias 6](#_heading=h.n2erv43yl8bb)

[Vista Física 7](#_heading=h.rz1b6dw2xaq0)

[Diagrama de despliegue 7](#_heading=h.cflut7b3n0dc)

[Vista de Casos de Uso 8](#_heading=h.1pzda3vo568r)

[Diagrama de Casos de uso: 8](#_heading=h.8likll8hl1r5)

[Diseño de Software 9](#_heading=h.s1bf6emha34w)

[Ciclo de Vida del Software 9](#_heading=h.i9lahevgkipx)

[Conclusión 10](#_heading=h.eikng9x4l4gm)

# 

# Introducción

Este informe detalla el desarrollo de EduHub, una plataforma educativa diseñada para organizar y ofrecer recursos académicos de manera digital. EduHub funciona como una biblioteca virtual, donde los estudiantes pueden acceder a contenido relevante para cada asignatura, generado por los docentes. Este enfoque permite una mayor concentración en el aprendizaje autónomo, brindando a los estudiantes una herramienta accesible.  
  
 El sistema se construyó bajo una arquitectura de microservicios, lo que facilita su escalabilidad y mantenimiento. Para la representación del diseño del sistema, se ha adoptado el modelo 4+1, lo que permite examinar el sistema desde diferentes perspectivas: lógica, de desarrollo, de procesos, física y de casos de uso. A través de diagramas UML, se presentan los componentes clave del sistema y sus interacciones, ofreciendo una visión clara de su estructura y funcionamiento.  
  
 En este informe también se explica la implementación de estándares de calidad en el diseño del software, los cuales son fundamentales para asegurar la robustez y fiabilidad del sistema, facilitando su integración y evolución futura.

# 

# 

# 

# 

# 

# 

# Modelado de Software

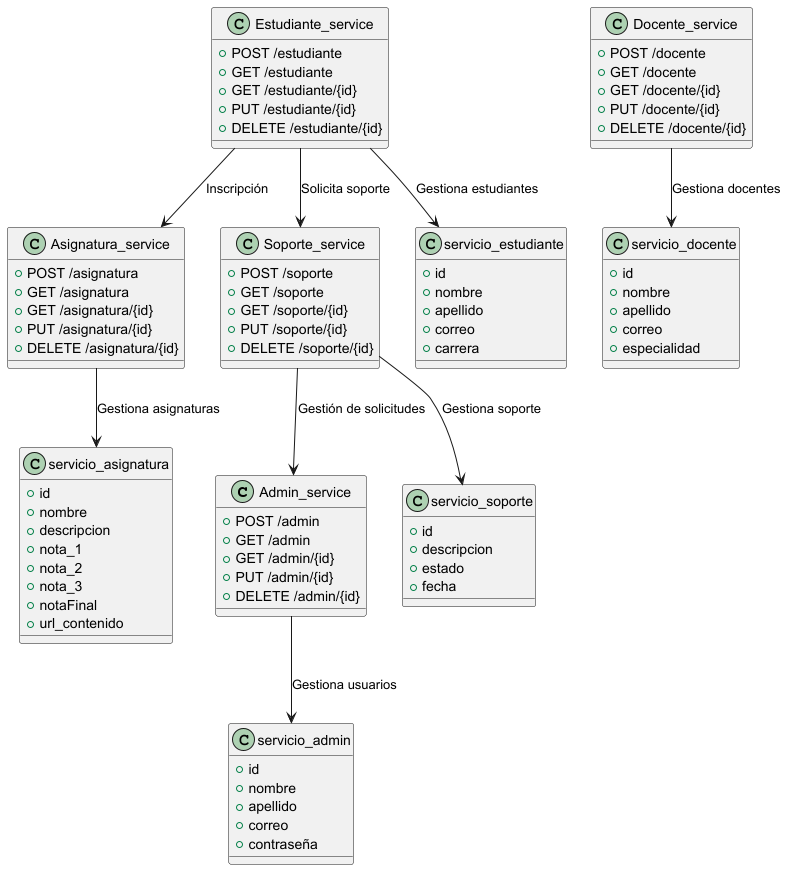
El modelado de software es una técnica esencial para representar de manera abstracta los sistemas y facilitar su comprensión, desarrollo y mantenimiento. En el caso de EduHub, se utilizaron los estándares de UML (Unified Modeling Language) para representar la arquitectura y el diseño del sistema. Los diagramas más relevantes incluyen:  
  
 Diagramas de Clases: Estos diagramas muestran la estructura estática del sistema, incluyendo las clases de los microservicios y sus relaciones. Cada microservicio cuenta con sus propias entidades, como 'Admin', 'Estudiante', 'Docente', 'Asignatura' y 'Soporte', representadas mediante clases con atributos y métodos clave.  
  
 Diagramas de Casos de Uso: Representan las interacciones entre los actores (como administradores, estudiantes y docentes) y el sistema. Estos diagramas permiten entender los requerimientos funcionales del sistema desde la perspectiva del usuario.  
  
 Diagramas de Secuencia: Muestran cómo los diferentes microservicios interactúan entre sí en tiempo de ejecución para llevar a cabo una operación, como la creación de un nuevo recurso académico o la solicitud de soporte.

El modelo 4+1 proporciona una visión integral del sistema desde cinco perspectivas clave:

## Vista Lógica:

Representa el sistema desde el punto de vista de sus componentes y clases. Los diagramas de clases se utilizan para mostrar la estructura de los microservicios, tales como 'AdminService' o 'EstudianteService', y cómo se relacionan entre sí.

### Diagrama de clases



## Vista de Desarrollo

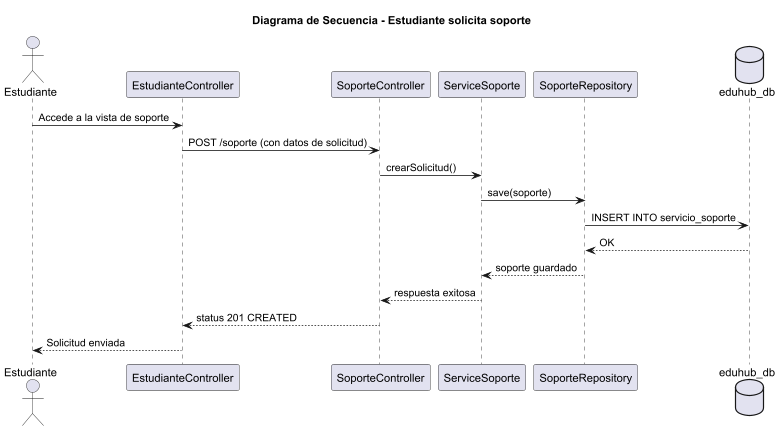
Muestra la organización del código y los paquetes dentro de cada microservicio, representados en los diagramas de paquetes. Cada microservicio tiene su propia estructura de paquetes como `controller`, `model`, `repository` y `service`, que facilitan el mantenimiento y la escalabilidad del sistema.

### Diagrama de paquetes

## Vista de Proceso:

Se enfoca en cómo interactúan los componentes del sistema en tiempo de ejecución. Los diagramas de secuencia proporcionan detalles sobre las interacciones entre los microservicios durante las operaciones del sistema, como la creación y gestión de recursos académicos.

### Diagrama de secuencias



## Vista Física

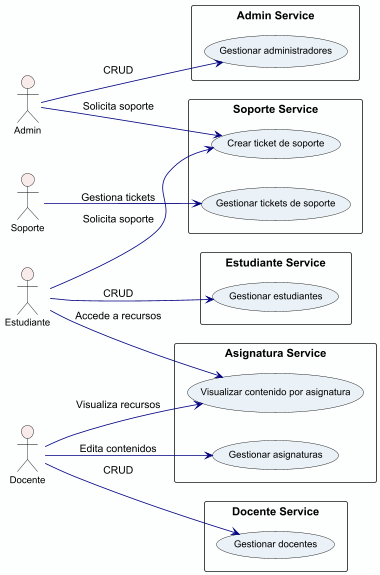
Representa la distribución de los microservicios en la infraestructura, detallando cómo se despliegan en servidores o contenedores. Este aspecto es clave para entender cómo los servicios se comunican y cómo se gestionan los recursos en la red.

### Diagrama de despliegue

## Vista de Casos de Uso

Detalla las interacciones entre los usuarios y el sistema. A través de los diagramas de casos de uso, se identifican las funcionalidades principales del sistema, tales como el registro de estudiantes, la creación de recursos académicos por los docentes y la gestión de soporte por parte de los administradores.

### Diagrama de Casos de uso:



# 

# Diseño de Software

El diseño de EduHub sigue una serie de estándares de calidad que buscan asegurar la robustez, escalabilidad y mantenibilidad del sistema. Algunos de los principales estándares aplicados son:  
  
 Principio de Responsabilidad Única: Cada microservicio tiene una única responsabilidad, lo que facilita su mantenimiento y escalabilidad. Por ejemplo, `AdminService` se encarga exclusivamente de la gestión de administradores, mientras que `EstudianteService` gestiona los estudiantes.  
  
Desacoplamiento y Modularidad: Los microservicios están desacoplados, lo que permite que puedan evolucionar y escalar de forma independiente. Cada microservicio interactúa con su propia base de datos, evitando dependencias fuertes entre ellos.  
  
Manejo de Excepciones y Validaciones: Se han implementado medidas de validación y manejo de errores para garantizar la integridad de los datos, como validaciones de entrada utilizando anotaciones en las clases de entidades.  
  
Escalabilidad: Se ha diseñado el sistema con la flexibilidad necesaria para permitir su escalabilidad horizontal en el futuro. Si bien actualmente se utiliza una base de datos única, se proyecta migrar hacia una solución distribuida con bases de datos separadas por microservicios.

# Ciclo de Vida del Software

El ciclo de vida de EduHub ha sido abordado desde su conceptualización hasta su implementación siguiendo las mejores prácticas de la ingeniería de software. Se comenzó con un análisis detallado de los requisitos, seguido de la creación de la documentación de diseño, basada en los principios de la metodología ágil Scrum. Durante la implementación, se utilizaron patrones de diseño como 4+1 para garantizar un desarrollo modular y escalable.

# Conclusión

En conclusión, el proyecto EduHub ha demostrado la efectividad de utilizar una arquitectura basada en microservicios para resolver problemas administrativos y académicos en una plataforma educativa. La aplicación de los modelos UML y el enfoque 4+1 ha proporcionado una representación clara y estructurada del sistema, facilitando su comprensión y la integración de nuevos módulos en el futuro.  
  
 El sistema está diseñado para ser robusto, escalable y fácil de mantener, con un enfoque centrado en la simplicidad y la accesibilidad de los recursos educativos. Los estándares de calidad aplicados aseguran que el sistema cumpla con altos niveles de fiabilidad y que sea fácilmente adaptable a futuras necesidades. Este enfoque no solo mejora la experiencia educativa de los usuarios, sino que también garantiza un desarrollo sostenible y colaborativo en el largo plazo.