

Ingeniería de Software II

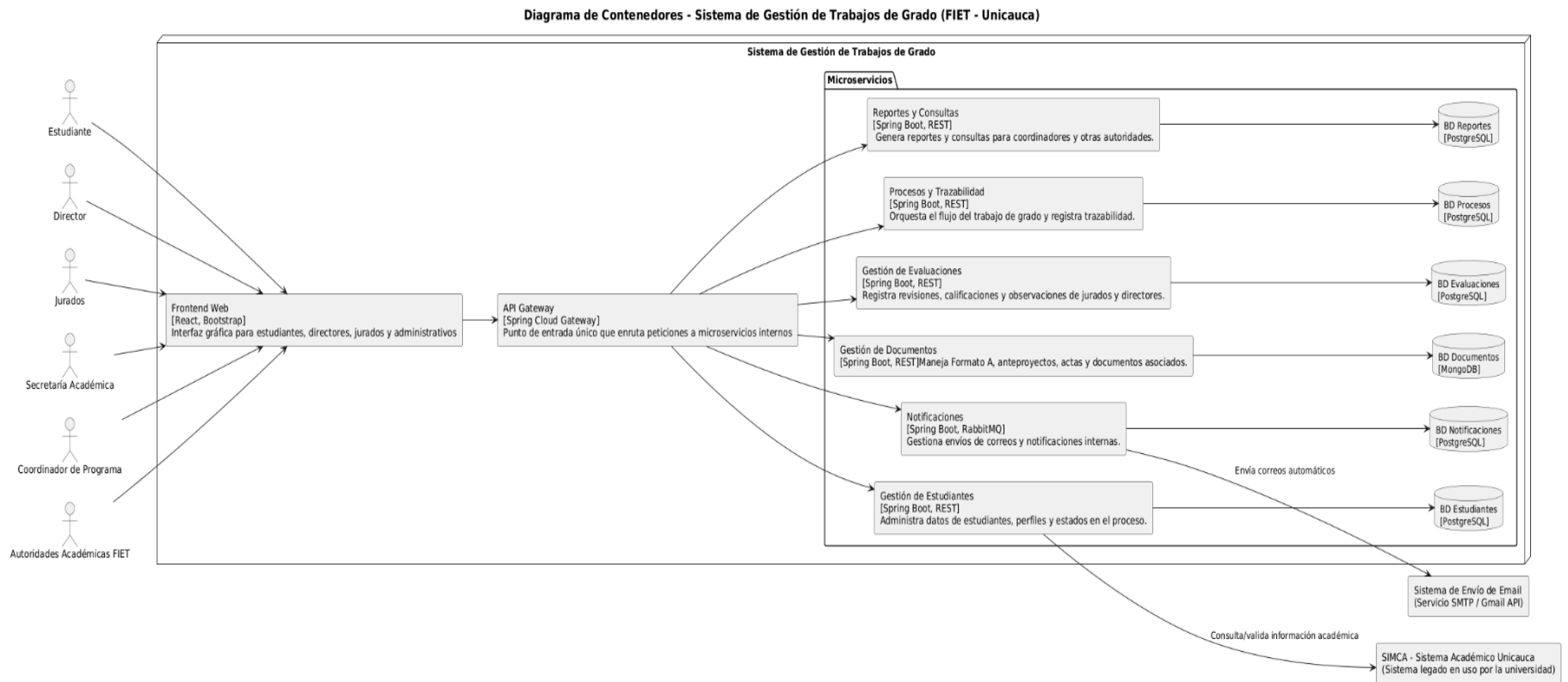
Teoría y Laboratorio

Entregables segundo corte 2025.2

Entregables

En grupos definidos del primer corte, hacer una refactorización de la iteración 1 al proyecto de curso de Gestión del Proceso de Trabajo de Grado para la FIET, implementando una solución distribuida con arquitectura de **microservicios** y orientada a eventos para garantizar la **escalabilidad**. Además, se deben aplicar los patrones de diseño vistos. La Figura 1, muestra un diagrama de contenedores con una posible solución utilizando microservicios (tomarlo como guía, cada grupo debe identificar los microservicios acordes a los requisitos).

Figura 1: Diagrama de contenedores (Modelo C4), con una posible solución de microservicios



A nivel de requisitos funcionales, se debe agregar dos nuevas historias y el envío de notificaciones (en color azul). A continuación, se describen los requerimientos a entregar.

Requisitos funcionales

Se deben implementar los siguientes requisitos funcionales de alto valor para el cliente:

1. **Yo como docente necesito** registrarme en el sistema de Gestión de Trabajos de grado **para** iniciar el flujo de un proyecto de grado, comenzando con la presentación del formato A. *Contexto:* Los datos que se deben ingresar son: nombres, apellidos, celular (dato opcional), programa al que pertenece (Ingeniería de Sistemas, Ingeniería Electrónica y Telecomunicaciones, Automática industrial, Tecnología en Telemática), email institucional y contraseña (mínimo 6 caracteres, debe tener al menos un dígito, al menos un carácter especial y al menos una mayúscula).
2. **Yo como docente necesito** subir un el formato A **para** comenzar el proceso de proyecto de grado. *Contexto:* el docente, una vez iniciada sesión, debe diligenciar un formulario con los datos: Título del proyecto de grado, modalidad (investigación, práctica profesional), fecha actual, director del proyecto de grado, codirector del proyecto de grado, objetivo general, objetivos específicos, archivo PDF a adjuntar. Cuando se trata de una modalidad de Práctica Profesional, el formato A debe tener al final, la carta de aceptación de la empresa. **NOTA: Una vez enviado el formato A, el sistema debe enviar una notificación asíncrona al email del coordinador.**
3. **Yo como** coordinador de programa **necesito** evaluar un formato A **para** aprobar, rechazar y dejar observaciones. *Contexto:* El sistema debe cargar al coordinador un listado de proyectos y su estado. Una vez evaluado el formato A, el sistema debe enviar **una notificación asíncrona mediante** un correo electrónico a los docentes y estudiantes implicados (se puede simular el envío con un *logger*), informando que se hizo una evaluación.
4. **Yo como docente necesito** subir una nueva versión del formato A cuando hubo una evaluación de rechazado **para** continuar con el proceso de proyecto de grado. *Contexto:* el requisito se parece al requisito 2, solo que lleva el conteo del número del intento (2,3). Después de un tercer intento, el proyecto es rechazado definitivamente y el estudiante debe empezar un nuevo proyecto desde cero. **NOTA: Una vez enviado el formato A (la nueva versión), el sistema debe enviar una notificación asíncrona al email del coordinador.**

5. **Yo como estudiante necesito** entrar a la plataforma y ver el estado de mi proyecto de grado. Contexto: los estados podrían ser algo como: en primera evaluación, formato A, en segunda evaluación formato A, en tercera evaluación formato A, aceptado formato A y rechazado formato A.
6. **Yo como docente necesito** subir el anteproyecto **para** continuar con el proceso de proyecto de grado. *Contexto:* el docente, una vez aprobado el Formato A del proyecto, puede subir el anteproyecto para ser evaluado por jefatura. El sistema debe guardar la fecha. *NOTA:* Una vez enviado el anteproyecto, el sistema debe enviar una notificación asíncrona al email del jefe de departamento.
7. **Yo como jefe de departamento necesito** ver los anteproyectos que han sido subidos por los docentes **para** luego asignar dos evaluadores del departamento de sistemas. Contexto: la asignación de evaluadores se hará en otra historia de usuario, aquí es solamente listar.

Rúbrica de evaluación

Criterio	Excelente (5 pts)	Bueno (4 pts)	Aceptable (3 pts)	Insuficiente (0, 1 pts)	Peso (%)
Cumplimiento de los requisitos funcionales	Implementa completamente los requisitos funcionales del segundo corte	Los requisitos se implementan con pequeñas limitaciones o errores.	Los requisitos se implementan parcialmente o con errores funcionales importantes.	Los requisitos no se implementan o no cumplen las necesidades del cliente.	30%
Cumplimiento de los requisitos NO funcionales de modificabilidad y escalabilidad	Se aplican correctamente los principios de diseño SOLID, SEIS patrones de diseño y patrón de	Se aplican los principios SOLID y patrones de diseño con pequeñas limitaciones o	Se aplican los principios SOLID y patrones de diseño con limitaciones o errores	No se aplican los principios SOLID y patrones de diseño.	30%

	microservicios.	errores	importantes		
Documentación de arquitectura	Documento claro, estructurado, con diagramas y detalles técnicos relevantes. Repositorio Git bien organizado y documentado	Documento y repositorio adecuados, pero con pequeños aspectos mejorables.	Documento y repositorio, con aspectos importantes a mejorar.	No hay documento de la arquitectura y repositorio Git.	20%
Pruebas unitarias automatizadas	Hay pruebas unitarias a todas las clases del dominio (entidades y servicios) correctamente implementadas	Hay pruebas unitarias con pequeñas limitaciones o errores	Hay pruebas unitarias con limitaciones o errores importantes	No hay pruebas unitarias	20%

PROTOCOLO DE SUSTENTACIÓN

Para la sustentación del primer entregable del proyecto de curso, cada equipo debe elaborar un video en YouTube. Importante que **todos los integrantes** del equipo participen en la sustentación. A continuación, el protocolo de sustentación:

1. Mostrar brevemente las historias de usuario implementadas en la segunda iteración: **1 minuto**
2. Mostrar el escenario de calidad de escalabilidad: **1 minuto**
3. Mostrar el diagrama de contextos (bounded context): **1 minuto**
4. Mostrar la arquitectura y diseño de software (usar el modelo C4 y UML): **2 minutos**
5. Mostrar cómo se implementaron los 6 patrones de diseño GoF: **1 minuto**
6. Mostrar las pruebas unitarias automatizadas: **1 minuto**
7. Mostrar el repositorio Git con los commits de todos los integrantes del equipo y el tablero de tareas del primer sprint Scrum: **1 minuto**
8. Mostrar el software funcional y algunos aspectos claves de la codificación que evidencien la implementación de la arquitectura: **8 minutos**

NOTA: Los grupos que se pasen de estos tiempos, serán penalizados con -1 punto.