

Ingeniería de Software II

Teoría y Laboratorio

Entregables primer corte 2025.2

Entregables

En grupos de 3 a 5 estudiantes, hacer una implementación del proyecto de curso de Gestión del Proceso de Trabajo de Grado para la FIET. La aplicación debe estar implementada en java Desktop con una arquitectura monolítica en 3 capas. Además, usar el micro-patrón MVC. Utilizar principios de diseño SOLID y algunos patrones de diseño GoF para garantizar la **modificabilidad** de la aplicación. Se debe implementar al menos un conjunto de historias de usuario de alto valor para el cliente.

A continuación, los requisitos funcionales que se deben entregar para el primer corte.

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.

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 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.
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.

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 primer 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	Se aplican correctamente los principios de diseño SOLID y patrones de diseño.	Se aplican los principios SOLID y patrones de diseño con pequeñas	Se aplican los principios SOLID y patrones de diseño con limitaciones o	No se aplican los principios SOLID y patrones de diseño.	30%

		limitaciones o errores	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 las historias de usuario con sus criterios de aceptación, prototipos de la interfaz gráfica de usuario y test de usabilidad: **2 minutos**
2. Mostrar los atributos de calidad relevantes para la iteración: **1 minutos**
3. Mostrar la arquitectura y diseño de software (usar el modelo C4 y UML): **2 minutos**
4. Mostrar las pruebas unitarias automatizadas: **1 minuto**
5. Mostrar el repositorio git con los commits de todos los integrantes del equipo y el tablero de tareas del primer sprint Scrum: **1 minuto**
6. Mostrar el software funcional y algunos aspectos claves de la codificación que evidencien la implementación de la arquitectura: **5 minutos**

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