

NOMBRE DE LA ASIGNATURA		Patrones De Diseño de Software							
NOMBRE DE LA ACTIVIDAD		Fundamentos Avanzados de Diseño Orientado a Objetos							
TIPO DE ACTIVIDAD		Sincrónica		Asincrónica	x	Individual	x	Grupal	
TEMÁTICA REQUERIDA PARA LA ACTIVIDAD				OBJETIVOS					
Unidad 1. Fundamentos avanzados de diseño orientado a objetos				Fortalecer la capacidad de analizar un sistema de software existente con problemas de diseño evidentes, para identificar violaciones a los principios SOLID y GRASP, y proponer un rediseño estructural utilizando diagramas UML que demuestre la aplicación correcta de estos principios.					
COMPETENCIAS				INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS					
<p>Resultado de Aprendizaje: Evalúa el nivel de aplicación de los principios de diseño en una solución de software, proponiendo mejoras estructurales basadas en UML.</p> <p>Estrategia de Evaluación: Análisis crítico de un sistema orientado a objetos existentes y rediseñado en UML incorporando los principios de diseño.</p>				<ul style="list-style-type: none">Material educativo y material complementario de la asignatura “Unidad 1.”Fuentes bibliográficas del módulo.					
CONOCIMIENTOS PREVIOS REQUERIDOS									
Conceptos fundamentales de POO, UML, POO en Java, SOLID, GRASP									
ESPECIFICACIONES DE LA ACTIVIDAD									
<p>Caso de estudio:</p> <p>En el repositorio https://github.com/UpcProgramingII/DesignPatterns-Ingsoft-2025-II.git, en el directorio “1Unidad 1\Actividad cierre\code” encontrara la primera versión de un API REST para Sistema de Gestión de Biblioteca Digital desarrollado por un grupo de ingenieros con poca experiencia. En el código elaborado se evidencian múltiples violaciones a principios de diseños, entre las cuales encontramos:</p> <ul style="list-style-type: none">Clases con múltiples responsabilidadesAlto acoplamiento entre componentesViolación del principio abierto/cerradoFalta de abstracción adecuadaProblemas de cohesión									

Junto con su equipo de trabajo, se le ha encomendado realizar un análisis exhaustivo del código, que les permita identificar violaciones a los principios SOLID, y con base a sus hallazgos proponer un rediseño estructural utilizando diagramas UML con la aplicación correcta de estos principios.

Tareas específicas a desarrollar:

- **Análisis funcional**
 - Identificar casos de uso principales
 - Documentar flujos de trabajo críticos
 - Crear diagrama de casos de uso inicial (entregable)
- **Análisis estructural**
 - Mapear clases existentes y sus relaciones
 - Identificar patrones arquitectónicos utilizados
 - Crear diagrama de clases del estado actual del modelo de dominio - (ingeniería inversa) (entregable)
 - Crear diagrama de componentes de la arquitectura actual (entregable)
- **Identificación de Problemas de Diseño**
 - Identificar violaciones sistemáticamente
 - Consolidar y documentar los hallazgos encontrados por el equipo (la documentación debe ser clara y precisa, indicar la clase, la línea de código, que principio está mal aplicado y sus posibles consecuencias). Puede agrupar los hallazgos por principios. (Entregable)
- **Planificación del rediseño**
 - Aplicar de principios de diseño para resolver problemas identificados
 - Diseño de nuevas abstracciones e interfaces (si lo considera necesario)
 - Nuevo diagrama de clases aplicando principios (entregable).
 - Nuevo Diagrama de componentes y dependencias (entregable).
 - Análisis y documentación de beneficios esperados (entregable)

Documento entregable – Resumen ejecutivo:

Se debe entregar un documento pdf de máximo 10 páginas que incluya las siguientes secciones:

1. Previa (Presentación)
2. Análisis funcional y estructural:
 - Diagrama de casos de uso
 - Diagrama de clase del modelo de dominio
 - Diagrama de arquitectura (componentes)
3. Análisis crítico consolidado (hallazgos de problemas de diseño):
 - Problemas de diseño identificados agrupados por cada principio (al menos 2 por cada principio)
4. Propuesta de rediseño detallado:
 - Diagrama de clase del modelo de dominio con mejoras
 - Diagrama de arquitectura con cambios (si los hay)
 - Documentación de cambios realizados (o por realizar) y los beneficios esperados

Criterios de Evaluación y Ponderación:

Criterio	Peso	Descripción
Criterio 1: Identificación correcta de violaciones a principios de diseño	35%	Capacidad para detectar y documentar violaciones a principios SOLID y GRASP
Criterio 2: Justificación técnica de las mejoras propuestas	30%	Calidad de la argumentación técnica y coherencia de las soluciones propuestas.
Criterio 3: Uso adecuado de diagramas UML y coherencia en el rediseño	20%	Corrección técnica de los diagramas UML y consistencia del nuevo diseño.
Criterio 4: Documentación y socialización	15%	Calidad del documento final entregado (organización, estructura, ortografía, normas, calidad de imágenes), dominio del tema

**RECOMENDACIONES /
OBSERVACIONES**

Para el diseño UML del Diagrama de clases se sugiere utilizar cualquiera de las siguientes herramientas: StartUML, PlantUML, Draw.io, Visual Paradigma.

Elaboro: Ing. Jairo Seoanes, Msc Ingeniería de Sistemas y Computación