

NOMBRE DE LA ASIGNATURA		Patrones De Diseño de Software							
NOMBRE DE LA ACTIVIDAD		Diseño de una solución de software incorporando patrones creacionales.							
TIPO DE ACTIVIDAD		Sincrónica		Asincrónica	x	Individual	x	Grupal	
TEMÁTICA REQUERIDA PARA LA ACTIVIDAD				OBJETIVOS					
Patrones creacionales y su integración con frameworks				Aplicar patrones de diseño creacionales en el desarrollo de una solución de software, integrando frameworks para lograr flexibilidad, escalabilidad y reutilización. Aplicar Patrones de estructurales para lograr para lograr sistema altamente modulares, flexibles y escalables .					
COMPETENCIAS				INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS					
<p>Resultado de Aprendizaje: Diseña soluciones de software aplicando patrones de diseño creacionales, de acuerdo a un escenario identificado, considerando su integración con frameworks de desarrollo para lograr flexibilidad y reutilización.</p> <p>Estrategia de Evaluación: Análisis crítico en el que se evidencia la correcta implementación de patrones creacionales, calidad del código e integración con frameworks de desarrollo</p>				<ul style="list-style-type: none">Material educativo y material complementario de la asignatura “Unidad 2.”Material educativo y material complementario de la asignatura “Unidad 3.” Fuentes bibliográficas del módulo.					
CONOCIMIENTOS PREVIOS REQUERIDOS									
Conceptos fundamentales de POO, UML, POO en Java, SOLID, GRASP, PATRONES DE DISEÑO CREACIONALES									
ESPECIFICACIONES DE LA ACTIVIDAD									
<p>Tareas específicas a desarrollar:</p> <p>Actividad 1: Análisis del escenario y definición de requerimientos:</p> <ul style="list-style-type: none"><u>Descripción:</u> El estudiante analiza el escenario propuesto o plantea uno propio. Define los requerimientos funcionales y no funcionales.<u>Producto esperado:</u> Descripción de caso de estudio (contexto del problema), incluyendo definición y explicación de requerimientos funcionales y no funcionales									

Nota: el problema debe orientarse a una problemática real y no ficticia, a la cual, se pretenda ofrecer una iniciativa de solución. Puede definirse en algunas de las áreas de aplicación: comercio, agricultura y ganadería, bienestar social y cuidado del medio ambiente.

Actividad 2: Diseño de la arquitectura de la solución

- Descripción: El estudiante diseña la arquitectura de la solución, justificando el tipo de arquitectura seleccionada (microservicios, hexagonal, etc), identificando los puntos donde se aplicarán los patrones creacionales (Factory Method, Abstract Factory, Singleton, Builder, Prototype) y estructurales (Adapter, Bridge, Composite, Decorator, Facade, Proxy, Flyweight) con su justificación técnica.
- Producto esperado: Diagrama de clases y componentes (arquitectura), con anotaciones sobre los patrones aplicados.

Actividad 3: Implementación de la solución

- Descripción: Descripción y justificación técnica del stack de herramienta seleccionadas para el desarrollo (lenguaje, framework, versiones, gestor dependencias, db, etc). Así mismo se espera la explicación de los mecanismos que ofrece el framework seleccionado para su integración con patrones de diseño creacionales y estructurales (anotaciones, inyección de dependencias, etc). Explicación detallada por cada patrón de diseño utilizado.
- Producto esperado: Código fuente documentado y funcional (opcional).

Actividad 4. Presentación de los resultados

Cada equipo tendrá un periodo de 25 minutos para presentar los resultados de su trabajo. En ese sentido, deberá explicar lo elaborado en cada una de las actividades desarrolladas y la demostración de la solución elaborada.

Entregables académicos:

Cada equipo deberá elaborar un informe técnico en el cual se integren los documentos entregables de cada actividad. El informe técnico debe poseer al menos la siguiente estructura:

- Análisis del problema
- Justificación de patrones seleccionados
- Arquitectura de la solución (diagrama de clases y componentes)
- Conclusiones y lecciones aprendidas.
- Acceso al código desarrollado y su documentación técnica

Criterios de Evaluación y Ponderación:

Criterio	Excelente (90%)	Bueno (70%)	Aceptable (40%)	Insuficiente (0-39%)
Aplicación del patrón creacional	Uso correcto, justificado y contextualizado de múltiples patrones.	Uso correcto de al menos dos patrones, con justificación.	Uso parcial o con errores menores en la aplicación de patrones.	Aplicación incorrecta o ausente de patrones.
Integración con frameworks	Integración fluida y adecuada con framework).	Integración funcional con algunos elementos de framework.	Uso limitado o con errores de integración.	No se evidencia integración con frameworks.
Calidad del código	Código limpio, modular, reutilizable y bien documentado.	Código funcional con buena estructura y documentación básica.	Código funcional pero con problemas de legibilidad o estructura.	Código desorganizado, sin documentación o con errores graves.

Presentación y reflexión	Explicación clara, profunda y crítica del trabajo realizado.	Explicación clara con algunos elementos críticos.	Presentación superficial o poco clara.	No se presenta o no se comprende el trabajo.	
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