

ASIGNATURA:	PATRONES DE DISEÑO DE SOFTWARE
Código:	TS5A4
Créditos Académicos:	4

Justificación	Dentro de la formación de un Tecnólogo en Desarrollo de Software, conocer patrones de diseño y saber cómo utilizarlos representa una herramienta importante en el proceso de desarrollo de un software ya que contemplan bases y estrategias en el análisis y diseño del mismo. Cuando se inicia la elaboración de un diseño de un sistema complejo y con calidad se torna difícil y en ocasiones con poca calidad, los patrones de diseño de software permiten fragmentar el diseño, reutilizar ideas y ayuda a complementar otras ideas así volviendo más óptimo y con mejor calidad el diseño a realizar. Por tal motivo es importante impartir a los estudiantes un fundamento sólido en cuanto al conocimiento de la teoría de los patrones de diseño y aprendan a aplicar los conocimientos adquiridos en las sesiones prácticas y así puedan enfrentar el reto de nuevos desarrollos tecnológicos.
Objetivo general	Proporcionar, a los futuros Tecnólogos, fundamentos teóricos y principios básicos en el diseño de software y la utilización de los patrones de diseño.
Objetivos Específicos	1. Capacitar al estudiante en el proceso de análisis y diseño de un software. 2. Impartir al estudiante la utilización adecuada de diagramas UML como apoyo en el análisis y diseño de un software. 3. Estudiar los 3 tipos de patrones de diseño (creacionales, estructurales y de comportamiento). 4. Estudiar los patrones de diseño e identificar el uso adecuado. 5. Orientar al estudiante la utilización de los patrones en el desarrollo de un diseño de software.

	6. Orientar al estudiante en finalizar un proceso de desarrollo de software desde el análisis, diseño e implementación. 7. Instruir al estudiante en cómo afrontar algunas complicaciones del paso de análisis a diseño y de diseño a implementación.
Metodología	Se expondrán los temas a tratar en clase utilizando material visual, utilizando como apoyo diapositiva, documentación y memorias contenidas en la nube donde el estudiante tendrá acceso constante. En las clases se presenta una atención principal en guiar y resolver dudas en cada uno de los temas a tratar, y así, debatir diferentes situaciones comunes que se generan en el análisis y diseño de un software. Cada tema a tratar se indica la teoría y desarrolla debates, talleres y ejercicios de análisis para así poner en práctica y afianzar los conceptos impartidos.
Competencias Genéricas	Actitudinales: El estudiante debe desarrollar actitudes que lo caractericen por su: - Facilidad para el trabajo en equipo. - Responsabilidad y cumplimiento. - Compromiso en la calidad de sus productos. - Capacidad para la toma de decisión. - Iniciativa y automotivación. Meta cognoscitiva: Se busca que el estudiante desarrolle: Capacidad investigativa y autónoma en el aprendizaje y apropiación de temas relacionados con el contenido de la asignatura.
Competencias específicas	Conceptuales: El estudiante estará en capacidad de: ➤ Aplicar los conceptos de análisis y diseño de software de acuerdo a unos requerimientos. ➤ Aplicar los conceptos de patrones de diseño en la creación de un diseño de software. ➤ Utilizar un diseño de software para la implementación de un software según los requerimientos del mismo.

	Procedimentales: Se enfoca al estudiante: ➤ Realizar un análisis previo de los requerimientos identificando el problema a solucionar. ➤ Identificar y fragmentar los diferentes componentes así enmarcar el alcance de una solicitud de software. ➤ Aplicar conocimiento en desarrollar un diseño de software utilizando patrones de diseño en casos reales. ➤ La organización de su trabajo.
Estrategias de aprendizaje	Las técnicas docentes que se van a utilizar son: 1. Clases magistrales. 2. Asesorías colectivas e individuales. 3. Debates y análisis de casos en grupo. 4. Talleres y ejercicios para afianzar conceptos. 5. Difusión de información y material de apoyo mediante correo electrónico y otras herramientas web que sean pertinentes. 6. Desarrollo de un diseño e implementación de un software.

Contenido de la asignatura	
Unidad 1	INTRODUCCIÓN CONCEPTOS BÁSICOS Paradigma orientada objetos Diagramas UML ¿Qué es un patrón de diseño?
Unidad 2	ANÁLISIS DE REQUERIMIENTOS ¿Qué es análisis y diseño? ¿Qué son el análisis y diseño orientados a objetos? Identificar y plantear un alcance de los requerimientos
Unidad 3	PATRONES DE CREACIÓN Abstract Factory Builder Patterns Factory Method Prototype Singleton

Unidad 4	PATRONES ESTRUCTURALES Adapter Bridge Composite Decorator Facade Flyweight Proxy
Unidad 5	PATRONES DE COMPORTAMIENTO Chain of responsibility Command Interpreter Iterator Mediator Memento Observer Strategy Template method Visitor
Unidad 6	IMPLEMENTACIÓN DE DISEÑO Unificar componentes fragmentados en un solo diseño Análisis de diseño al pasar a la implementación Implementación y pruebas estándar del software.

Textos Guía	• Craig Larman, UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado, 2ª. Edición, Pearson Educación, 2003 • Design Patterns Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Pearson Education 1994
--------------------	---

Referencia	Bibliografía
1	Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra., Head First Design Patterns, O'Reilly, 2008
2	Laurent Debrauwer, Patrones de diseño para C#: Los 23 modelos de diseño: descripción y soluciones ilustradas en UML 2 y C#, Ediciones ENI, 2012

3	Laurent Debrauwer , Patrones de diseño en Java: los 23 modelos de diseño: descripción y solución ilustradas en UML 2 y Java, Ediciones ENI, 2013
4	Vaskaran Sarcar , Design Patterns in C#: A Hands-on Guide with Real-world Examples, Apress, 2020
5	Alexandre Malavasi , Implementing Design Patterns in C# and .NET 5, BPB Publications, 2021
6	Praseed Pai, Shine Xavier , .NET Design Patterns, Packt Publishing, 2017
7	Yang Hu , Fácil Aprendizaje Patrones de Diseño Python: Cree un Código Orientado a Objetos Mejor y Reutilizable, Independently Published, 2020
8	Carlos Santana Roldán , React 17 Design Patterns and Best Practices, 3 Edition, Packt Publishing, 2021
9	Junade Ali , Mastering PHP Design Patterns, Packt Publishing, 2016