UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
ESCUELA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

1. Datos generales

Materia: INGENIERÍA DE SOFTWARE II

Código: ICC0029

Paralelo:

Periodo : Marzo-2021 a Julio-2021

Profesor: ORTEGA CHASI PATRICIA MARGARITA

Correo electrónico portega@uazuay.edu.ec

Docencia	Práctico	Autónomo: 56		Total horas
		Sistemas de tutorías	Autónomo	
48	16	16	40	120

Prerrequisitos:

Código: ICC0022 Materia: INGENIERÍA DE SOFTWARE I

2. Descripción y objetivos de la materia

Esta materia es uno de los pilares de la carrera, pues en su dominio se refleja el diseño de sistemas robustos y de fácil mantenimiento. La materia de Ingeniería de Software II es parte del área de software en la que se integran asignaturas como Ingeniería de Software I, III y IV, las cuáles se fundamentan en la aplicación de métodos, procedimientos y técnicas necesarios para la especificación de requerimientos de software, la planificación, gestión y control de proyectos de desarrollo de software; además de la especificación, evaluación y aseguramiento de la calidad del software.

Este curso cubrirá los principios y conceptos del modelado orientado a objetos, el estudio de los principales diagramas de estructura, comportamiento e interacción de UML, así como el diseño de software basado en patrones a nivel de objetos y arquitectura.

La materia de Ingeniería de Software II contribuye en la formación integral de los futuros profesionales, brindándoles un marco teórico – práctico para que a partir de la especificación de los requerimientos puedan profundizar en el análisis y diseño del software con un enfoque orientado a objetos, utilizando los diagramas de estructura, comportamiento e interacción de UML. Adicionalmente, el estudiante conocerá y comprenderá la importancia del uso de patrones de diseño en el proceso de desarrollo de software.

3. Contenidos

1.	**Modelado orientado a objetos**
1.01.	Fundamentos del modelo de objetos. Análisis orientado a objetos OOA (1 horas)
1.02.	Diseño orientado a objetos OOD. Beneficio del modelo de objetos (1 horas)
1.03.	Elementos del modelo de objetos (abstracción, encapsulación, modularidad, jerarquía, tipo, concurrencia y persistencia) (2 horas)
2.	**Clases y objetos**
2.01.	Definición de objeto. Estado,comportamiento e identidad del objeto (1 horas)
2.02.	Definición de clase. Atributos (propiedades), métodos (operaciones) y mensajes (1 horas)
2.03.	Atributos (propiedades), métodos (operaciones) y mensajes (1 horas)
2.04.	Relaciones entre clases (generalización - herencia, asociación, agregación, composición, dependencia) (2 horas)
2.05.	Polimorfismo (1 horas)
3.	**Lenguaje de modelado unificado (UML)**
3.01.	Introducción a UML. Modelos a nivel conceptual, lógico y físico (0 horas)
3.02.	Clases (0 horas)
3.02.1	Notación de clases y relaciones (asociación, especialización, agregación y composición) (2 horas)
3.02.2	Conceptos avanzados: clases parametrizadas, nombres de extremos, calificadores, restricciones, OCL, clases asociativas, notas (4 horas)
3.02.3	Taller de modelado de diagramas de clases (2 horas)
3.03.	Paquetes (0 horas)
3.03.1	Notación de paquetes (1 horas)
3.03.2	Visibilidad de elementos, relaciones de dependencia, importar, acceso. (2 horas)

3.03.3	Taller de modelado de diagramas de paquete (1 horas)
3.04.	Componentes (0 horas)
3.04.1	Notación de componentes e interfaces (1 horas)
3.04.2	Realización y estructura interna del componente (2 horas)
3.04.3	Taller de modelado de diagramas de componentes (2 horas)
3.05.	Modelado diagramas de despliegue (0 horas)
3.05.1.	Notación de nodos y artefactos (2 horas)
3.05.2.	Taller de modelado de diagramas de despliegue (1 horas)
3.06.	Modelado diagramas de actividad (0 horas)
3.06.1.	Notación de nodos inicio y fin, actividades, nodos de decisión, bifurcación y unión, particiones (swimlines), flujos o transiciones (2 horas)
3.06.2.	Taller de modelado de diagramas de actividad (2 horas)
3.07.	Diagramas de máquina de estado (0 horas)
3.07.1.	Notación de estados iniciales, finales, simples, transiciones, eventos (1 horas)
3.07.2.	Conceptos avanzados: actividades de estado, controlar transiciones, estados compuestos y anidados (2 horas)
3.07.3	Taller de modelado de diagramas de máquina de estado (2 horas)
3.08.	Diagramas de secuencia (0 horas)
3.08.1.	Notación de objeto, líneas de vida, mensajes (1 horas)
3.08.2.	Conceptos avanzados: tiempos de espera, mensajes reflexivos y recursivos, constructores de control (interaction use, alt, loop) (3 horas)
3.08.3	Taller de modelado de diagramas de secuencia (2 horas)
4.	**Patrones de diseño**
4.01.	Definición y elementos de un patrón de diseño (1 horas)
4.02.	Organización del catálogo de patrones de diseño (1 horas)
4.03.	¿Cómo seleccionar y usar un patrón de diseño? (2 horas)
4.04.	Estudio de casos y resolución de problemas mediante patrones de diseño. (2 horas)
4.05.	Estudio de casos y resolución de problemas mediante patrones de diseño. (2 horas)
5.	**Diseño arquitectónico**
5.01.	Conceptos e importancia de la arquitectura de software (1 horas)
5.02.	Vistas arquitectónicas (1 horas)
5.03.	Estructura, elemento, sistema, subsistema, modulo, componentes, conectores y relaciones, acoplamiento, cohesión, complejidad (2 horas)
5.04.	Aproximaciones al diseño: Top-Down, Bottom-Up. "Greenfield" vs "Brownfield" sistemas de software (1 horas)
5.06.	Estilos arquitectónicos: centrada en datos, flujos de datos, máquinas virtuales, llamada - retorno, componentes independientes (2 horas)
5.07.	Estudio de casos y resolución de problemas mediante patrones arquitectónicos: modelo-vista-controlador, tubería&filtro (pipe&filter), repositorio (blackboard), broker, capas, orientada a servicios, dirigida por eventos (Presentación-abstracción-control), microservicios (2 horas)
5.08.	Estudio de casos y resolución de problemas mediante patrones arquitectónicos: modelo-vista-controlador, tubería&filtro (pipe&filter), repositorio (blackboard), broker, capas, orientada a servicios, dirigida por eventos (Presentación-abstracción-control), microservicios (2 horas)

4. Sistema de Evaluación

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia	Evidencias
az. Evalúa sistemas computacionales de múltiples niveles de detalle en cuanto a abstracción, complejidad, cambio evolutivo y principios generales, trascendiendo detalles de implementación de los componentes y contextualizando la estructura de los sistemas informáticos y los procesos implicados en su construcción y análisis.	
-Analiza y diseña sistemas aplicando conceptos de orientación a objetos	-Evaluación escrita -Proyectos -Trabajos prácticos - productos
-Relaciona los distintos artefactos de tal forma que el problema se resuelva partiendo de lo más general a lo mas específico de manera encadenada	-Proyectos -Trabajos prácticos - productos

Desglose de evaluación

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
Evaluación escrita	Resolver cuestionarios y ejercicios sobre conceptos básicos del modelado de objetos/diagrama de clases.		APORTE DESEMPEÑO	3	Semana: 4 (05/04/21 al 10/04/21)
Evaluación escrita	Resolver cuestionarios y ejercicios sobre diagrama de componentes		APORTE DESEMPEÑO	3	Semana: 9 (10/05/21 al 15/05/21)
Evaluación escrita	Resolver cuestionarios y ejercicios sobre diagramas de actividad y secuencias		APORTE DESEMPEÑO	4	Semana: 12 (31/05/21 al 05/06/21)
Proyectos	Proyecto final de modelado de un sistema usando UML.		EXAMEN FINAL ASINCRÓNICO	10	Semana: 17-18 (05-07-2021 al 18-07-2021)
Evaluación escrita	Examen con preguntas de conceptos y aplicación de modelado de objetos y UML, incluirá ejercicios de aplicación.		EXAMEN FINAL SINCRÓNICO	10	Semana: 17-18 (05-07-2021 al 18-07-2021)
Proyectos	Proyecto final de modelado de un sistema usando UML.		SUPLETORIO ASINCRÓNICO	10	Semana: 17-18 (05-07-2021 al 18-07-2021)
Evaluación escrita	Examen con preguntas de conceptos y aplicación de modelado de objetos y UML, incluirá ejercicios de aplicación.		SUPLETORIO SINCRÓNICO	10	Semana: 17-18 (05-07-2021 al 18-07-2021)

Metodología

Criterios de Evaluación

5. Referencias

Bibliografía base

Libros

Autor	Editorial	Título	Año	ISBN
Pressman R. S.	McGraw Hill.	Ingeniería de Software: Un enfoque práctico	2010	
BOOCH,G., ROBERT A. MAKSIMCHUK, MICHAEL W. ENGLE	Pearson Education	OBJECT-ORIENTED ANALYSIS AND DESIGN WITH APPLICATIONS	2007	0-201-89551-X
Pressman R. S., Maxim B. R.	McGraw Hill	Software Engineering: A Practitioner's Approach	2015	
Sommerville, I.	Pearson	Software Engineering.	2016	
Stephens, R.	John Wiley & Sons	Beginning Software Engineering	2015	
SOMMERVILLE, IAN.	Pearson Addison Wesley	INGENIERÍA DE SOFTWARE	2005	978-6-07-320603-7
LARMAN, CRAIG	Prentice Hall	UML Y PATRONES INTRODUCCIÓN AL ANÁLISIS Y DESEMPEÑO ORIENTADO A OBJETOS	2004	8420534382
Groussard, T., Debrauwer, L.	Editorial ENI	Patrones de Diseño y Java		
Bass, L., Clements, P., Kazman, R.	Addison-Wesley	Software Architecture in Practice	2012	
Gamma E., Helm R., Johnson, R., Vlissides, J.	Addison-Wesley	Design patterns: elements of Reusable Object-Oriented Software	1994	
Buschmann F., Meunier R., Rohnert H.,	John Wiley & Sons	Pattern-Oriented Software Architecture Volume 1: A System of	2001	

Autor	Editorial	Título	Año	ISBN
Sommerlad P., Stal M		Patterns		

Web

Software

Bibliografía de apoyo

Libros

Web

Software

Docente

Director/Junta

Fecha aprobación: **12/03/2021**

Estado: **Aprobado**