



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: A
Periodo : Marzo-2024 a Junio-2024
Profesor: ORTEGA CHASI PATRICIA MARGARITA
Correo electrónico: portega@uazuay.edu.ec

Nivel: 6

Distribución de horas.

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

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.

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.

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. Objetivos de Desarrollo Sostenible



4. Contenidos

01	Modelado orientado a objetos
1.01.	Fundamentos del modelo de objetos. Análisis orientado a objetos OOA y Diseño orientado a objetos OOD
1.02.	Elementos del modelo de objetos (abstracción, encapsulación, modularidad, jerarquía, tipo, concurrencia y persistencia)
02	Lenguaje de modelado unificado (UML)
2.01.	Clases
2.01.1	Notación de clases y relaciones (asociación, especialización, agregación y composición)

2.01.2	Conceptos avanzados: clases parametrizadas, nombres de extremos, calificadores, clases asociativas
2.01.3	Restricciones, OCL
2.01.4	Taller de modelado de diagramas de clases
2.02.	Paquetes
2.02.1	Notación de paquetes
2.02.2	Visibilidad de elementos, relaciones de dependencia, importar, acceso.
2.02.3	Taller de modelado de diagramas de paquete
2.03.	Componentes
2.03.1	Notación de componentes e interfaces
2.03.2	Realización y estructura interna del componente
2.03.3	Taller de modelado de diagramas de componentes
2.04.	Modelado diagramas de despliegue
2.04.1.	Notación de nodos y artefactos
2.04.2.	Taller de modelado de diagramas de despliegue
2.05.	Modelado diagramas de actividad
2.05.1.	Notación de nodos inicio y fin, actividades, nodos de decisión, bifurcación y unión, particiones (swimlines), flujos o transiciones
2.05.2.	Taller de modelado de diagramas de actividad
2.06.	Diagramas de máquina de estado
2.06.1.	Notación de estados iniciales, finales, simples, transiciones, eventos
2.06.2.	Conceptos avanzados: actividades de estado, controlar transiciones, estados compuestos y anidados
2.06.3	Taller de modelado de diagramas de máquina de estado
2.07.	Diagramas de secuencia
2.07.1.	Notación de objeto, líneas de vida, mensajes
2.07.2.	Conceptos avanzados: tiempos de espera, mensajes reflexivos y recursivos, constructores de control (interaction use, alt, loop)
2.07.3	Taller de modelado de diagramas de secuencia
03	Patrones de diseño
3.01.	Definición y elementos de un patrón de diseño
3.02.	Organización del catálogo de patrones de diseño
3.03.	¿Cómo seleccionar y usar un patrón de diseño?
3.04.	Estudio de casos y resolución de problemas mediante patrones de diseño. (2 horas)
04	Diseño arquitectónico
4.01.	Conceptos e importancia de la arquitectura de software
4.02.	Vistas arquitectónicas
4.03.	Estructura, elemento, sistema, subsistema, modulo, componentes, conectores y relaciones, acoplamiento, cohesión, complejidad
4.04.	Estilos arquitectónicos: centrada en datos, flujos de datos, máquinas virtuales, llamada - retorno, componentes independientes
4.05.	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

5. Sistema de Evaluación

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia

Evidencias

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 más específico de manera encadenada

-Evaluación escrita
-Proyectos
-Trabajos prácticos -
productos

Desglose de evaluación

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
Evaluación escrita	Examen práctico sobre los temas tratados	Modelado orientado a objetos	APORTE	5	Semana: 4 (18-MAR-24 al 23-MAR-24)
Trabajos prácticos - productos	Tareas asignadas sobre los temas tratados en clase	Modelado orientado a objetos	APORTE	5	Semana: 4 (18-MAR-24 al 23-MAR-24)
Evaluación escrita	Examen práctico sobre los temas tratados	Lenguaje de modelado unificado (UML)	APORTE	5	Semana: 8 (15-ABR-24 al 20-ABR-24)
Trabajos prácticos - productos	Tareas asignadas sobre los temas tratados en clase	Lenguaje de modelado unificado (UML)	APORTE	5	Semana: 8 (15-ABR-24 al 20-ABR-24)
Evaluación escrita	Examen práctico sobre los temas tratados	Patrones de diseño	APORTE	5	Semana: 12 (13-MAY-24 al 18-MAY-24)
Trabajos prácticos - productos	Tareas asignadas sobre los temas tratados en clase	Patrones de diseño	APORTE	5	Semana: 12 (13-MAY-24 al 18-MAY-24)
Evaluación escrita	Examen escrito, incluye casos prácticos	Diseño arquitectónico, Lenguaje de modelado unificado (UML), Modelado orientado a objetos, Patrones de diseño	EXAMEN	10	Semana: 16 (10-JUN-24 al 11-JUN-24)
Proyectos	Proyecto	Diseño arquitectónico, Lenguaje de modelado unificado (UML), Modelado orientado a objetos, Patrones de diseño	EXAMEN	10	Semana: 16 (10-JUN-24 al 11-JUN-24)
Evaluación escrita	Examen escrito, incluye casos prácticos	Diseño arquitectónico, Lenguaje de modelado unificado (UML), Modelado orientado a objetos, Patrones de diseño	SUPLETORIO	20	Semana: 19-20 (al)

Metodología

Descripción	Tipo horas
Se requiere que los estudiantes revisen el material de manera previa a cada clase. Adicionalmente, se espera que en estas horas autónomas los estudiantes profundicen los temas de clase a través de las actividades y bibliografía sugeridas. Para esta asignatura, los estudiantes requieren completar entre 4 y 5 horas de trabajo autónomo cada semana.	Autónomo
Los materiales del curso y envíos de trabajos se realizarán a través del Campus Virtual. Se fomentará el diálogo y la comunicación como medio de compartir conocimientos y examinar supuestos y creencias. Como instructor, publicaré artículos, notas de clase, enlaces y aspectos destacados en las diversas formas que el Campus Virtual permite. La comunicación se realizará a través del Campus Virtual o por medio del correo electrónico institucional. Las actividades propuestas fomentan el aprendizaje activo. Se utilizará una variedad de metodologías incluyendo presentaciones, debates, trabajo en grupo, preguntas y respuestas, demostraciones, sesiones prácticas, ejercicios en grupos pequeños, etc.	Total docencia

Criterios de evaluación

Descripción	Tipo horas
<p>Las actividades autónomas programadas se realizarán a través de trabajos prácticos, lecturas, ejercicios, etc. Los actividades serán realizadas de manera individual o en grupo cuando se indique.</p> <p>Se evaluará la presentación, el apego técnico, el alcance y nivel de detalle de la investigación, y el aporte personal de los alumnos. En todos los trabajos escritos, se evaluará la ortografía y la correcta redacción con una penalización de máximo el 20% de la calificación.</p> <p>Con el objetivo de promover la equidad y la integridad en el curso, se establece la siguiente política con respecto al uso de sistemas de inteligencia artificial (IA) en trabajos y evaluaciones. Los estudiantes no están autorizados a utilizar IA para generar texto en tareas, exámenes, proyectos o cualquier otra actividad evaluada relacionada con el proceso de diseño de software. Esto incluye la generación de soluciones de diseño, documentación técnica, modelos UML, diagramas, entre otros. El uso de IA de esta manera puede afectar negativamente el desarrollo de habilidades críticas de diseño y análisis propias del proceso de ingeniería de software. No obstante, se permite el uso de IA como una herramienta de apoyo para investigación y preparación, siempre y cuando los estudiantes generen y elaboren sus propias soluciones de diseño. Los estudiantes deben ser conscientes de las limitaciones y riesgos asociados con el uso de IA en el proceso de diseño de software, y evaluar críticamente la relevancia y calidad de la información generada. Cualquier infracción a esta política se considerará una violación del código de conducta académica del curso y será sujeta a las consecuencias correspondientes. El uso de herramientas de IA generativa debe estar debidamente documentado y citado para cualquier trabajo presentado en este curso. Si existen dudas o inquietudes sobre el uso adecuado de IA, se recomienda consultar al instructor del curso para obtener orientación adicional.</p>	Autónomo
<p>Las evaluaciones se realizarán a través de trabajos prácticos, talleres de trabajo en clase y un proyecto que equivale a 10 puntos del examen final y un examen sobre 10 puntos.</p> <ul style="list-style-type: none">· Los trabajos serán realizados en grupos de estudiantes que serán definidos de manera aleatoria por el profesor.· Los trabajos versaran sobre los temas tratados en clase y se evaluarán tanto por su contenido como por las presentaciones orales que realizarán los alumnos en clase.· La calificación de cada parcial se realizará con base en los trabajos presentados por los alumnos. <p>Los alumnos deberán seleccionar un dominio de software de su interés y construir distintos los modelos que se estudien a lo largo del ciclo. En estos trabajos se evaluará, el alcance y su nivel de detalle, el apego a los estándares de modelado a ser utilizados y la relación entre los distintos modelos construidos. En todas las sustentaciones orales se evaluará la diagramación de la presentación, la calidad de la presentación y el apego técnico.</p>	Total docencia

6. 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
Grossard, 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., Sommerlad P., Stal M	John Wiley & Sons	Pattern-Oriented Software Architecture Volume 1: A System of Patterns	2001	

Web

Software

Revista

Bibliografía de apoyo

Libros

Web

Software

Revista

Docente

Director/Junta

Fecha aprobación: **20/02/2024**

Estado: **Aprobado**