

SÍLABO



FACULTAD DE INGENIERÍA

MARZO 2024-AGOSTO 2024

NOMBRE DE LA ASIGNATURA		CÓDIGO:	18562
ANÁLISIS Y DISEÑO DE SOFTWARE - GRUPO: 1			
CARRERA	COMPUTACION REDISEÑO		
CICLO O SEMESTRE	CUARTO NIVEL	EJE DE FORMACIÓN	PROFESIONALES, PRAXIS PROFESIONAL
CRÉDITOS DE LA ASIGNATURA	4	MODALIDAD:	PRESENCIAL

CARGA HORARIA

COMPONENTES DEL APRENDIZAJE	Horas / Semana	Horas / Periodo Académico
APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	4.0	64.0
APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	2.0	32.0
APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	2.0	32.0
APRENDIZAJE AUTÓNOMO (AA)	4.0	64.0
Total Horas:	12.0	192.0

PROFESOR(ES) RESPONSABLE(S):

PONCE VASQUEZ DIEGO ARTURO - (D.P.)	(diego.ponce@ucuenca.edu.ec)	PRINCIPAL
VEGA ZAMORA OSWALDO FRANCISCO - (O.V.)	(francisco.vegaz@ucuenca.edu.ec)	

DESCRIPCIÓN DE LA ASIGNATURA:

Resumen descriptivo en torno al propósito, la estrategia metodológica y el contenido fundamental de la asignatura.

La asignatura presenta a los estudiantes los métodos de análisis, diseño y desarrollo de software. Se presenta los métodos de desarrollo, patrones de diseño, modelos de desarrollo, y principales arquitecturas del software en el contexto de la programación orientada a objetos.

REQUISITOS DE LA ASIGNATURA

Esta asignatura no tiene co-requisitos

PRE-REQUISITOS	
Asignatura	Código
PROGRAMACIÓN II: ESTRUCTURA DE DATOS Y ANÁLISIS DE ALGORITMOS	17952

OBJETIVO(S) DE LA ASIGNATURA:

Objetivos general y específicos de la asignatura en relación al Perfil de salida de la carrera.

Objetivo general: Conocer el proceso de desarrollo de software, el análisis y diseño de software, la importancia de los patrones de diseño en el proceso de desarrollo orientado a objetos, así como la tendencia actual en cuanto a metodologías.

Objetivos específicos:

1. Conocer la importancia del análisis y diseño de software en el proceso de desarrollo orientado a objetos.

2. Conocer los diferentes métodos de desarrollo de software.
3. Conocer y Aplicar UML y PUDS.
4. Conocer y aplicar los patrones de diseño a casos prácticos.
5. Conocer y aplicar los métodos ágiles.
6. Conocer los modelos de software.
7. Conocer las distintas arquitecturas del software, y los contextos de desarrollo.

LOGRO DE LOS RESULTADOS DE APRENDIZAJE, INDICADOR(ES) Y ESTRATEGIA(S) DE EVALUACIÓN

Resultados o Logros de Aprendizaje (RdA's) de la Unidad de Organización Curricular (UOC) correspondiente, Indicadores y Estrategias de Evaluación de la Asignatura, tomando como referencia el Perfil de salida (PdS) y la Organización Curricular (OC) del Proyecto de Carrera (PdC).

RESULTADOS O LOGROS DE APRENDIZAJE	INDICADORES	ESTRATEGIAS DE EVALUACIÓN
RdA1. Conoce desde una perspectiva histórica la evolución de los métodos de desarrollo.	• Conoce los términos y las fuerzas en el surgimiento y evolución de los métodos de desarrollo de software.	• Conoce y entiende las razones de la evolución de la Ingeniería del Software hasta el estado actual.
RdA2. Conoce los métodos de desarrollo, sus ventajas y desventajas y aplicación.	• Sabe y aplica correctamente los diferentes métodos de desarrollo de software, conoce sus fases y aplicación de cada uno.	• Conoce las metodologías, sus ventajas y desventajas y cuando debe utilizar una u otra.
RdA3. Conoce y aplica el diseño basado en interfaces.	• Conoce el concepto y utilidad de las Interfaces y aplica correctamente.	• Conoce el diseño de interfaces, su utilidad y aplicación práctica.
RdA4. Conoce y aplica los patrones de diseño de software.	• Conoce y aplica correctamente los patrones de diseño.	• Conoce los patrones de diseño de software, identifica cuando aplicarlos, los utiliza acertadamente en casos prácticos.
RdA5. Crea programas utilizando modelos y basados en patrones de diseño.	• Traduce un diagrama de clases basado en patrones a un programa orientado a objetos.	• Aplica patrones de diseño en forma acertada en el desarrollo de software.
RdA6. Entiende y aplica los métodos ágiles.	• Conoce y aplica correctamente los métodos ágiles en el desarrollo de Software.	• Conoce la filosofía ágil y la aplicación de los métodos Extreme y SCRUM. • Conoce, aplica y utiliza los métodos de desarrollo basado en pruebas y en modelos.

CONTENIDOS, SESIONES Y ACTIVIDADES DE APRENDIZAJE

Título de la Unidad, sub -unidades, nro. de sesión y actividades para los componentes de aprendizaje.

SUB-UNIDADES	Nro. SESIÓN	COMPONENTE DE APRENDIZAJE	ACTIVIDADES DE APRENDIZAJE	
1. 0. PRESENTACIÓN DEL SÍLABO, EVALUACIÓN Y DIAGNOSTICO.				
1. Presentación del Sílabo, evaluación y diagn+oastico.	1	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Presentación del sílabo, examen y diagnóstico. (D.P.)	2 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Aprendizaje práctico experimental. (D.P.)	1 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	Sílabo. (D.P.)	2 horas
		APRENDIZAJE AUTÓNOMO (AA)	examen (D.P.)	1 horas
2. 1. EVOLUCIÓN HISTÓRICA DEL A&D DE SOFTWARE. IMPORTANCIA DE LA INGENIERÍA DE REQUERIMIENTOS.				

SUB-UNIDADES	Nro. SESIÓN	COMPONENTE DE APRENDIZAJE	ACTIVIDADES DE APRENDIZAJE	
1. Evolución histórica del desarrollo de software desde las primeras décadas hasta nuestros días. 2. Importancia de la estimación de tiempos, presupuesto y recursos de un proyecto de ingeniería del software. 3. Evolución de los métodos de desarrollo de softwarwe. 4. Importancia de la especificación de requisitos.	1	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Tarea en clase (D.P.)	1 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	tutoria (D.P.)	1 horas
	2	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	4 horas
		APRENDIZAJE AUTÓNOMO (AA)	Trabajo (D.P.)	4 horas
3. 2. MÉTODOS DE ANÁLISIS & DISEÑO				
1. Método en Cascada. 2. Método en V. 3. Método en espiral. 4. Métodos Iterativos 5. Misceláneos.	2	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Tarea en Clase (D.P.)	1 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	tutoria (D.P.)	1 horas
	4	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	8 horas
	6	APRENDIZAJE AUTÓNOMO (AA)	Trabajo (D.P.)	6 horas
4. 3. UML UNIFIED MODELLING LANGUAGE, LENGUAJE UNIFICADO DE MODELADO.				
1. UML, Fundamentos. 2. Ciclo de desarrollo UML. 3. Fases del ciclo de desarrollo UML. 4. Artefactos, Diagramas, Paritcipantes. 5. Herramientas.	3	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Tarea en clase (D.P.)	4 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	tutoria (D.P.)	4 horas
	6	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	6 horas
	9	APRENDIZAJE AUTÓNOMO (AA)	Trabajo (D.P.)	12 horas
5. 4. PUDS PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.				
1. Proceso unificado del Dersarrollo de Software. 2. Ciclos, Fases, Productos. 3. Estudio de un caso didáctico. 4. Aplicación práctica del PUDS.	4	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Tarea en clase (D.P.)	6 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	tutoria (D.P.)	4 horas
	8	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	6 horas
	12	APRENDIZAJE AUTÓNOMO (AA)	trabajo (D.P.)	12 horas
6. 5. FUNDAMENTOS DE PROGRAMACION ORIENTADA A OBJETOS: HERENCIA, POLIMORFISMO, ENCAPSULAMIENTO. FUNDAMENTOS DEL LENGUAJE DE PROGRAMACIÓN JAVA.				

SUB-UNIDADES	Nro. SESIÓN	COMPONENTE DE APRENDIZAJE	ACTIVIDADES DE APRENDIZAJE	
1. El paradigma de la programación orientada a objetos. 2. Conceptos de Objeto, herencia encapsulamiento y polimorfismo. 3. Concepto de clase e interfaz	13	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	4 horas
		APRENDIZAJE AUTÓNOMO (AA)	Tutorial de Programacion en Java desde Cero (D.P.)	8 horas
	14	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	tutoria poo en Java. (D.P.)	1 horas
	15	APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	Representacion en lenguaje UML. (D.P.)	4 horas
7. 6.PATRONES DE DISEÑO DE SOFTWARE. ERICK GAMMA Y LA BANDA DE LOS 4.				
1. Que son los patrones de diseño de software y cual es su importancia. 2. Presentación de los principales patrones de diseño del software y su implementación en código Java. 3. Presentación de los patrones de diseño estructurales, Creacionales y de Comportamiento. 4. Realización de un caso práctico.	15	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Aplicación de los patrones de diseño en el A y D de un proyecto de la vida real (D.P.)	2 horas
	16	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Programacion de patrones de diseño (D.P.)	8 horas
		APRENDIZAJE AUTÓNOMO (AA)	tarea sobre patrones de diseño y su aplicación. (D.P.)	4 horas
	17	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Taller de patrones de diseño (D.P.)	6 horas
	18	APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	Taller programacion de patrones de diseño (D.P.)	4 horas
8. 7. MÉTODOS AGILES: EXTREME Y SCRUM.				
1. Fundamentos de los métodos Agiles. El manifiesto Agil. 2. Programación Extrema. 3. Metodología de programación SCRUM.	6	APRENDIZAJE AUTÓNOMO (AA)	Trabajo (D.P.)	4 horas
	7	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	tarea en clase (D.P.)	4 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	tutoria (D.P.)	3 horas
	14	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	8 horas
	16	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Trabajo (D.P.)	6 horas
9. 8.ARQUITECTURAS DE SOFTWARE.				

SUB-UNIDADES	Nro. SESIÓN	COMPONENTE DE APRENDIZAJE	ACTIVIDADES DE APRENDIZAJE	
1. Principales Arquitecturas del Software. 2. Método que se aplican dependiendo del contexto y arquitectura.	4	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Tareas en clase (D.P.)	4 horas
	7	APRENDIZAJE AUTÓNOMO (AA)	trabajo (D.P.)	4 horas
	8	APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	tutoría (D.P.)	3 horas
	14	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	4 horas
	18	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Trabajo (D.P.)	2 horas
10. 9. TEST DRIVEN DEVELOPMENT TDD.				
1. Metodología Basada en Pruebas. 2. Clases de Pruebas 3. El ciclo de desarrollo basado en pruebas.	5	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Tareas en Clase (D.P.)	1 horas
	7	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	clase (D.P.)	4 horas
	9	APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	Tutoría (D.P.)	3 horas
	20	APRENDIZAJE AUTÓNOMO (AA)	Trabajo (D.P.)	5 horas
11. 10. MODEL DRIVEN DEVELOPMENT MDD.				
1. Modelos de Desarrollo 2. Desarrollo basado en modelos. 3. Principales modelos y su aplicación	6	APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	Tarea en clase (D.P.)	1 horas
	10	APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	tutoría (D.P.)	3 horas
	16	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	Clase (D.P.)	2 horas
	22	APRENDIZAJE AUTÓNOMO (AA)	Trabajo (D.P.)	4 horas
		APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	64 horas	
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	32 horas	
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - TUTORÍA (APE/T)	32 horas	
		APRENDIZAJE AUTÓNOMO (AA)	64 horas	
		Total Planificación:	192 horas	

RECURSOS O MEDIOS PARA EL APRENDIZAJE

Equipos, materiales, instrumentos tecnológicos, reactivos, entre otros, que serán utilizados durante el desarrollo de la asignatura.

• El ambiente de trabajo requiere del uso de un proyector de computadora, laboratorio de computadores personales, utilización de las herramientas Eclipse para Java, UML, PUDS, herramientas gráficas y entorno de desarrollo de aplicaciones con acceso a Internet.

Se hace uso de una amplia bibliografía, herramientas de software y tutoriales en línea para reforzar en aprendizaje en aula.

CRITERIOS PARA LA ACREDITACIÓN DE LA ASIGNATURA

Parámetros de acreditación, tomando como referencia los Resultados de Aprendizaje (RdA's), indicadores y criterios de evaluación planteados y en base a la normativa de evaluación y calificaciones vigente en la Universidad de Cuenca y Consejo de Educación Superior (CES).

CRITERIO GENERAL DE ACREDITACIÓN	PUNTAJE
PRUEBAS	20
TRABAJOS	20
TAREAS EN CLASE	10
EXAMENES	50
TOTAL:	100

	DETALLE DE CRITERIOS DE ACREDITACIÓN	PUNTAJE / CRITERIO GENERAL	
C94	APROVECHAMIENTO I		
	Trabajo autónomo, se promedia con una prueba sobre el tema.	10	TRABAJOS
	prueba 1er bimestre	10	PRUEBAS
	tareas en clase 1er bimestre	5	TAREAS EN CLASE
C95	INTERCICLO		
	Interciclo	20	EXAMENES
C96	APROVECHAMIENTO II		
	Trabajo autónomo 2do bimestre, se promedia con una prueba sobre el tema.	10	TRABAJOS
	Prueba 2do bimestre	10	PRUEBAS
	tareas en clase 2do bimestre	5	TAREAS EN CLASE
C97	FINAL		
	Examen Final	30	EXAMENES
C98	SUSPENSIÓN		
	Total:	100	

TEXTOS U OTRAS REFERENCIAS REQUERIDAS PARA EL APRENDIZAJE DE LA ASIGNATURA

Libros, revistas, bases digitales, periódicos, direcciones de Internet y demás fuentes de información, pertinentes y actuales.

BÁSICA

1. Sommerville Ian, Ingeniería de Software, 7ma Edición, 2005, Addison Wesley.
2. UML, Jacobson, Rumbaugh, Booch, Addison Wesley.
3. Erich Gamma, Patrones de Diseño, 2003, Addison Wesley.
4. Raoul Gabriel Urma - Richard Warburton, (2020). Real World Software Development, O'Reilly. ISBN 978-1-491-96717-1.

COMPLEMENTARIA

1. Budd Timothy, Object-Oriented Programming, 3th, 2002, Addison Wesley
2. R.. Wirfs-Brock, B. Wilkerson, L. Wiener, Designing Object Oriented Software, 1990, Prentice Hall.
3. Pressman R., Ingeniería de Software. Un enfoque practico, 2006, McGraw Hill.
4. Java in a Nutshell, 5th, 2005, O'Reilly Media

5. Sierra K., Bates B., Head First Java, 2006, O'Reilly Media.
6. Eddie Burris, Programming in the Large with Design Patterns,
7. Erick y Elisabeth Freeman, Head First Design Patterns.

Docente: PONCE VASQUEZ DIEGO ARTURO

Finalizado: 13/3/2024

Director: VEINTIMILLA REYES JAIME EDUARDO

Publicado: 27/3/2024