



FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

ESCUELA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

1. Datos generales

Materia: INGENIERÍA DE SOFTWARE III
Código: ICC0036
Paralelo: A
Periodo : Septiembre-2023 a Febrero-2024
Profesor: ERAZO GARZON LENIN XAVIER
Correo electrónico: lerazo@uazuay.edu.ec

Nivel: 7

Distribución de horas.

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

Prerrequisitos:

Código: ICC0029 Materia: INGENIERÍA DE SOFTWARE II

2. Descripción y objetivos de la materia

El conocimiento de diferentes modelos de desarrollo de software existentes y su aplicabilidad. Así como técnicas de estimación, planificación, seguimiento y control de proyectos de software. Sumado a esto, técnicas de administración de configuración de productos software. Para finalmente, conocer conceptos relacionados al mantenimiento y reingeniería de software.

Está integrado directamente con Ingeniería de Software II ya que es el paso siguiente luego de conocer de análisis y diseño orientado a objetos. Así como también integrado con Ingeniería de Software IV en donde se ve la Ingeniería Software basado en componentes y calidad de software.

Para fortalecer el conocimiento de los estudiantes en cuanto a los modelos de procesos de software, así como en la gestión adecuada de proyectos, al igual que conocer de la administración de la configuración del software y el mantenimiento, reingeniería y refactorización de software.

3. Objetivos de Desarrollo Sostenible



4. Contenidos

1.01.	¿Qué es un modelo de proceso de software?
1.02.	Modelo en cascada, incremental, evolutivo (construcción de prototipos, espiral), concurrentes, unificado
1.03.1.	¿Qué es la agilidad en el software?
1.03.2.	El manifiesto ágil
1.03.3.	Programación extrema: valores XP, proceso XP, XP industrial.
1.03.4.	SCRUM: roles, product backlog, sprint, reuniones.
1.04.1.	Componentes y modelo de componentes

1.04.2.	Procesos CBSE
1.04.3.	Composición de componentes
1.04.4.	Ingeniería de software orientada a aspectos
1.05.1.	Metamodelos de sistema
1.05.3.	Arquitectura dirigida por modelos (MDA)
1.05.4.	Lenguajes de dominio específico (DSL) y transformaciones de modelos
1.06.	Cuadro comparativo de modelos de proceso de software
2.01.1.	Personal: participantes (stakeholders), líderes de equipo, equipo de software, equipos ágiles, comunicación y coordinación.
2.01.2.	Producto: ámbito del software, descomposición del problema.
2.01.3.	Proceso: fusión de producto y proceso, descomposición del proceso
2.01.4.	Proyecto: definición, gestión de proyectos, ciclo de vida de un proyecto, influencia de la organización en la gestión de proyectos
2.02.1.	Ámbito y factibilidad del software
2.02.2.	Métricas de proceso y de proyecto
2.02.3.	Estimación del tamaño, esfuerzo, tiempo y costos de proyectos de software
2.02.5.	Modelos de estimación empíricos del tamaño del software
2.02.6.	Gestión de riesgos de proyectos de software
2.02.8.	Planificación temporal de proyectos de software: estructura de desglose del trabajo (EDT), red de tareas (Pert) y cronograma (Gantt).
2.03.	Seguimiento y control de proyectos: gestión del valor ganado (EVM)
2.05.	Planificación, medición y estimación en proyectos de desarrollo ágil
3.01.	¿Qué es la ACS?. El repositorio ACS y los elementos de configuración del software
3.02.	Líneas de referencia o base. Versiones, revisiones, variantes y releases
3.03.1.	Identificación de la configuración del software
3.03.2.	Control de versiones de la configuración del software
3.03.3.	Control de cambios de la configuración del software (Procedimientos y mecanismos)
3.03.4.	Auditoría de la configuración. Reportes de estado de la configuración
3.04.	Estudio de caso sobre ACS
4.01.	Introducción a las técnicas de validación del software
4.02.01.	Definición y tipos de revisión
4.02.02.	Fases de una revisión (Planificación, orientación inicial, preparación individual, reunión de inspección, seguimiento y evaluación)
4.03.01.	Definición y principios de la prueba de software
4.03.02.	Tipos de pruebas del software (unidad, integración, sistema, aceptación, regresión)
4.03.03.	Metodología y estrategias de pruebas
4.03.04.	Diseño de casos de prueba
4.03.05.	Integración continua (pruebas automáticas)
5.01.	Mantenimiento de software. Reingeniería de software (concepto objetivos, beneficios)
5.02.1.	Modelo cíclico de Pressman: análisis de inventarios, re-estructuración de documentación, ingeniería inversa,

5. Sistema de Evaluación

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia

az. Evalúa sistemas computacionales de múltiples niveles de detalle en cuanto a abstracción,

Evidencias

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia	Evidencias
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.	
-Aplica técnicas de administración de configuración de productos software	-Evaluación escrita -Investigaciones -Proyectos
-Aplica técnicas de estimación, planificación, seguimiento y control de proyectos de software	-Evaluación escrita -Investigaciones -Proyectos
-Compara diferentes modelos de desarrollo de software existentes y su aplicabilidad.	-Evaluación escrita -Investigaciones -Proyectos
-Conoce conceptos relacionados al mantenimiento y reingeniería de software	-Evaluación escrita -Investigaciones -Proyectos

Desglose de evaluación

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
Investigaciones	Trabajo de investigación sobre modelos de proceso de desarrollo de software en formato artículo científico (sustentado).	Modelos de proceso de software	APORTE	2	Semana: 3 (02-OCT-23 al 07-OCT-23)
Evaluación escrita	Prueba capítulos 1 y 2.	Gestión de proyectos de software, Modelos de proceso de software	APORTE	4	Semana: 5 (16-OCT-23 al 21-OCT-23)
Proyectos	Desarrollo de un metamodelo para un dominio de aplicación específico.	Modelos de proceso de software	APORTE	4	Semana: 5 (16-OCT-23 al 21-OCT-23)
Evaluación escrita	Prueba capítulo 2 y 3	Administración de la configuración del software (ACS), Gestión de proyectos de software	APORTE	5	Semana: 10 (20-NOV-23 al 25-NOV-23)
Proyectos	Desarrollo del proyecto de ciclo: Ambito del software. Especificación de Requisitos del Software (ERS). Estimación del tamaño del software mediante puntos casos de uso. Plan de gestión de riesgo. Planificación temporal del proyecto.	Gestión de proyectos de software	APORTE	5	Semana: 10 (20-NOV-23 al 25-NOV-23)
Proyectos	Plan de gestión de configuración y pruebas de software del proyecto de ciclo.	Administración de la configuración del software (ACS)	APORTE	4	Semana: 13 (11-DIC-23 al 16-DIC-23)
Evaluación escrita	Prueba capítulos 3, 4 y 5.	Administración de la configuración del software (ACS), Mantenimiento y reingeniería, Verificación y validación del software	APORTE	3	Semana: 15 (al)
Investigaciones	Trabajo de investigación sobre refactorización de software.	Mantenimiento y reingeniería	APORTE	3	Semana: 15 (al)
Evaluación escrita	Toda la materia.	Administración de la configuración del software (ACS), Gestión de proyectos de software, Mantenimiento y reingeniería, Modelos de proceso de software, Verificación y validación del software	EXAMEN	10	Semana: 19-20 (21-01-2024 al 27-01-2024)
Proyectos	Desarrollo de un DSL para modelar un dominio de aplicación específico.	Modelos de proceso de software	EXAMEN	10	Semana: 19-20 (21-01-2024 al 27-01-2024)
Evaluación escrita	Toda la materia.	Administración de la configuración del software (ACS), Gestión de proyectos de software, Mantenimiento y reingeniería, Modelos de proceso de software, Verificación y validación del software	SUPLETORIO	20	Semana: 20 (al)

Metodología

Descripción	Tipo horas
Se utiliza una metodología de enseñanza – aprendizaje basada en la clase inversa, es decir previo a las clases se publicará material didáctico (presentaciones, libros, artículos, ejercicios resueltos, videos) en el campus virtual para la revisión autónoma por parte de los estudiantes. A su vez, el estudiante desarrollará trabajos fuera de clase relacionados con investigaciones sobre temas incluidos en el silabo; y, planificación e implementación de proyectos de desarrollo de software.	Autónomo
La estrategia metodológica a emplear tiene como objetivo promover una participación activa de los estudiantes dentro del proceso de enseñanza – aprendizaje. La implementación de la estrategia metodológica contempla las siguientes actividades: i) exposiciones magistrales por parte del profesor para proporcionar un marco teórico – práctico de cada uno de los temas y solventar las dudas sobre temas previamente publicados en el campus virtual; ii) planteamiento y resolución de problemas a través de proyectos grupales.; y, iii) pruebas referentes a los temas tratados, incluyendo las respectivas revisiones y retroalimentaciones por parte del profesor.	Total docencia

Criterios de evaluación

Descripción	Tipo horas
Los trabajos relacionados con investigaciones y los proyectos de software serán sustentados y se evaluarán considerando los siguientes aspectos: i) Dominio e integración de conocimientos. ii) Profundidad de la investigación y aporte personal al tema. (Fuentes de consulta). iii) Originalidad del documento/solución. iv) Cumplimiento de los requerimientos establecidos en el trabajo. v) Calidad de la sustentación (fluidez del expositor y el buen uso de ayudas audiovisuales). vi) Redacción y ortografía. vii) Puntualidad en la entrega de los trabajos.	Autónomo
Las preguntas de las pruebas serán formuladas en base a los temas tratados en clase y a los trabajos realizados por los estudiantes. La correcta conceptualización y el análisis empleado en la resolución de cada una de las preguntas serán considerados en la calificación. Serán inaceptables situaciones de plagio y copia, haciéndose acreedor el alumno a un puntaje de cero.	Total docencia

6. Referencias

Bibliografía base

Libros

Autor	Editorial	Título	Año	ISBN
Pressman R. S., Maxim B. R.	McGraw Hill	Software Engineering: A Practitioner's Approach	2015	
Sommerville, I.	Pearson	Software Engineering.	2016	
Brambilla, M., Cabot, J., Wimmer, M.	Morgan & Claypool	Model-Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering	2012	
Stephens, R.	John Wiley & Sons	Beginning Software Engineering	2015	
PROJECT MANAGEMENT INSTITUTE, INC.	PMI	GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS (GUÍA DEL PMBOK)	2008	978-1-933890-72-2
Pons, C., Giandani, R., Pérez, G.	McGraw Hill	Desarrollo de Software dirigido por modelos: Conceptos teóricos y su aplicación práctica.	2010	

Web

Autor	Título	Url
OMG	MDA Specifications	https://www.omg.org/mda/specs.htm

Software

Revista

Bibliografía de apoyo

Libros

Software

Autor	Título	Url	Versión
Obeo	Obeo Designer Community	https://www.obeodesigner.com/en/download	11.5

Revista

Autor	Volumen	Título	Año	DOI
Lenin Erazo-Garzón;	IEEE - Second	Models@runtime and Internet of Things: A	2021	https://ieeexplore.ieee.
Lenin Erazo-Garzón;	IEEE ACCESS / 10	A Domain-Specific Language for Modeling	2022	https://ieeexplore.ieee.

Docente

Director/Junta

Fecha aprobación: 11/09/2023

Estado: Aprobado